インターフェイスおよびハードウェア コンポーネント コンフィギュレーション ガイド、Cisco IOS XE Release 3S（ASR 1000）

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

【注意】シスコ製品をご使用になる前に、安全上の注意（**www.cisco.com/jp/go/safety_warning/**）をご確認ください。本書は、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。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

目 次

第 1 章

IPv6 を介した mGRE トンネル サポート

mGRE トンネルは、サービス プロバイダーがコア インフラストラクチャに IPv6 を導入できるように設定されます。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 最新の注意事項と機能情報については、プラットフォームおよびソフトウェア リリースのバグ検索ツールとリリース ノートを参照してください。 このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表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シスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

IPv6 を介した mGRE トンネル サポートに関する情報

IPv6 を介した mGRE サポート

DMVPN の複数のサイトが IPv6 によって相互接続されています。 単一の論理 mGRE トンネル インターフェイスが、ある VPN サイトを別の VPN サイトに相互接続します。 IPv6 サブネットは、トンネル インターフェイスをさまざまな VPN サイトの他のトンネル インターフェイスに接続し

ます。 VPN サイトを接続しているすべてのトンネル インターフェイスが、論理 IPv6 サブネットでホストとして機能します。 この構造は、トンネル オーバーレイ ネットワークと呼ばれます。

IPv6 を介した mGRE トンネル サポート

サービス プロバイダーがコア インフラストラクチャに IPv6 を導入できるように、IPv6 を介したマルチポイント総称ルーティング カプセル化（mGRE）トンネルがサポートされます。 Dynamic Multipoint Virtual Private Network（DMVPN）のユーザは、ローカル ネットワークで IPv4 または IPv6 のいずれかを実行できるため、オーバーレイ エンドポイントは IPv4 または IPv6 のいずれかになります。 IPv6 トランスポート エンドポイントでは、オーバーレイ エンドポイントは IPv4 または IPv6 プライベート ネットワーク アドレスのいずれかになります。

GRE には、パッセンジャ プロトコルを識別するプロトコル フィールドが含まれています。 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 フィールドが含まれていない場合は、トンネルが IS-IS パケットまたは IPv6 パケットを伝送していたかどうかは識別できません。 GRE 内で IS-IS および IPv6 をトンネル化するには、GRE プロトコル フィールドが必要です。

その他の関連資料

関連資料

関連項目	マニュアル タイトル
IPv6 アドレッシングと接続	『*IPv6 Configuration Guide*』
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
IPv6 コマンド	『Cisco IOS IPv6 Command Reference』
Cisco IOS IPv6 機能	『Cisco IOS IPv6 Feature Mapping』

標準および RFC

標準/RFC	タイトル
IPv6 に関する RFC	『*IPv6 RFCs*』

MIB

MIB	MIB のリンク
	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 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/cisco/web/support/index.html

IPv6 を介した mGRE トンネル サポートの機能情報

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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

表 *1*：*IPv6* を介した *mGRE* トンネル サポートの機能情報

機能名	リリース	機能情報
IPv6 を介した mGRE トンネル サポート	15.2(1)T XE Release 3.8S	mGRE トンネルは、サービス プロバイダーがコア インフラストラクチャに IPv6 を導入できるように設定されます。

第 2 章

IPv6 IP トンネルを介する IP

IPv6 は IPv6 トンネルを介して IP をサポートします。サポート内容は次のとおりです。

- IPv6 トラフィックに対する総称ルーティング カプセル化（GRE）IPv4 トンネル 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。 GRE トンネルは、2 つのエッジ デバイス間またはエッジ デバイスとエンド システム間に定期的でセキュアな通信を必要とする安定した接続のために主に使用されます。 エッジ デバイスとエンド システムは、デュアル スタック実装である必要があります。
- IPv6 トランスポートを介した GRE サポート：GRE には、パッセンジャ プロトコルを識別するプロトコル フィールドが含まれています。 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。
- IPv6 トンネルを介した VRF-Aware IPv4/IPv6：仮想ルーティングおよび転送（VRF）Aware トンネルは、信頼できないコア ネットワークまたは別のインフラストラクチャ（IPv4 または IPv6）を備えたコア ネットワークで区切られたカスタマー ネットワークに接続するために使用されます。

IPv6 トンネルを介した IP に関する情報

IPv6 トラフィック用の GRE IPv4 トンネル 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。 GRE トンネルは、手動で設定された IPv6 トンネルと同様、リンクごとに個別のトンネルが設定された 2 つのポイント間のリンクです。 これらのトンネルは、特定のパッセンジャまたはトランスポート プロトコルに結合されていませんが、この場合、GRE を使用するパッセンジャ プロトコルとして IPv6 を伝送し、トランスポート プロトコルとして IPv4 または IPv6 を伝送します。

GRE トンネルは、2 つのエッジ デバイス間またはエッジ デバイスとエンド システム間に定期的でセキュアな通信を必要とする安定した接続のために主に使用されます。エッジデバイスとエンド システムは、デュアル スタック実装である必要があります。

IPv6 トランスポートを介した GRE サポート

GRE には、パッセンジャ プロトコルを識別するプロトコル フィールドが含まれています。 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 フィールドが含まれていない場合は、トンネルが IS-IS パケットまたは IPv6 パケットを伝送していたかどうかは識別できません。 GRE プロトコル フィールドでは、GRE 内に IS-IS および IPv6 をトンネリングすることを推奨します。

IPv6 トンネルを介した IP の設定方法

次の作業では、IPv6 トンネルを設定する方法について説明します。 IPv6 または IPv4 パケットは、このトンネルで転送できます。

GRE IPv6 トンネルの設定

IPv6 ネットワーク上で GRE トンネルを設定するには、次の作業を実行します。 GRE トンネルは、IPv6 ネットワーク層を介して送出し、IPv6 トンネルを介して IPv6 パケットと IPv4 パケットを転送するように設定できます。

はじめる前に

GRE IPv6 トンネルが設定されている場合、IPv6 アドレスは、トンネル送信元およびトンネル宛先に割り当てられます。 トンネル インターフェイスは、IPv4 または IPv6 アドレスのいずれかにすることができます（このことは、以降の作業では示されていません）。 設定されたトンネルの両

端にあるホストまたはデバイスは、IPv4 プロトコル スタックと IPv6 プロトコル スタックの両方をサポートしている必要があります。

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. **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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • 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 tunnel** *tunnel-number* 例： `Device(config)# interface tunnel 0`	トンネル 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： `Device(config-if)# tunnel source ethernet 0`	送信元 IPv6 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。 • インターフェイスのタイプと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、『IPv6 Command Reference』を参照してください。
ステップ 5	**tunnel destination** *ipv6-address* 例： `Device(config-if)# tunnel destination 2001:0DB8:0C18:2::300`	トンネル インターフェイスの宛先 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、『IPv6 Command Reference』を参照してください。

	コマンドまたはアクション	目的
ステップ 6	**tunnel mode gre ipv6** 例： `Device(config-if)# tunnel mode gre ipv6`	GRE IPv6 トンネルを指定します。 （注） **tunnel mode gre ipv6** コマンドでは、GRE をトンネルインターフェイスのカプセル化プロトコルとして指定します。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、『IPv6 Command Reference』を参照してください。
ステップ 7	**end** 例： `Device(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。

IPv6 トンネルを介した IP の設定例

例：IPv6 トンネルを介した IPv6

例：CE1 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet0/0
 no ipv6 address
 ipv6 address 2001:DB8:2:1::1/64
 no shutdown
 exit
!

ipv6 route 2001:DB8:2:5::/64 2001:DB8:2:1::2
ipv6 route 2001:DB8:2:9::/64 2001:DB8:2:1::2
!
```

例：PE1 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Tunnel0
 no ipv6 address
 ipv6 address 2001:DB8:2:9::1/64
 tunnel source 2001:DB8:2:2::1
 tunnel mode ipv6
 tunnel destination 2001:DB8:2:4::2
 exit
!
```

```
interface Ethernet0/0
 no ipv6 address
 ipv6 address 2001:DB8:2:1::2/64
 no shutdown
 exit
!
!
interface Ethernet1/1
 no ipv6 address
 ipv6 address 2001:DB8:2:2::1/64
 no shutdown
 exit
!

ipv6 route 2001:DB8:2:3::/64 2001:DB8:2:2::2
ipv6 route 2001:DB8:2:4::/64 2001:DB8:2:2::2
ipv6 route 2001:DB8:2:5::/64 Tunnel0 2001:DB8:2:9::2
```

例：PE2 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Tunnel0
 no ipv6 address
 ipv6 address 2001:DB8:2:9::2/64
 tunnel source 2001:DB8:2:4::2
 tunnel mode ipv6
 tunnel destination 2001:DB8:2:2::1
 exit
!
interface Ethernet0/0
 no ipv6 address
 ipv6 address 2001:DB8:2:5::1/64
 no shutdown
 exit
!
interface Ethernet0/1
 no ipv6 address
 ipv6 address 2001:DB8:2:4::2/64
 no shutdown
 exit
!
!
ipv6 route 2001:DB8:2:2::/64 2001:DB8:2:4::1
ipv6 route 2001:DB8:2:3::/64 2001:DB8:2:4::1
ipv6 route 2001:DB8:2:1::/64 Tunnel0 2001:DB8:2:9::1
```

例：CE2 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet0/0
 no ipv6 address
 ipv6 address 2001:DB8:2:5::2/64
 no shutdown
 exit
!
ipv6 route 2001:DB8:2:1::/64 2001:DB8:2:5::1
ipv6 route 2001:DB8:2:9::/64 2001:DB8:2:5::1
!
```

例：コア デバイス 1 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet1/0
 no ipv6 address
 no shutdown
 ipv6 address 2001:DB8:2:3::1/64
 exit
!
interface Ethernet1/1
 no ipv6 address
 ipv6 address 2001:DB8:2:2::2/64
 no shutdown
 exit
!
ipv6 route 2001:DB8:2:4::/64 2001:DB8:2:3::2
```

例：コア デバイス 2 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet0/1
 no ip address
 ipv6 address 2001:DB8:2:4::1/64
 no shutdown
 exit
!
interface Ethernet1/0
 no ip address
 ipv6 address 2001:DB8:2:3::2/64
 no shutdown
 exit
!
ipv6 route 2001:DB8:2:2::/64 2001:DB8:2:3::1
```

例：IPv6 トンネルを介した IPv4

例：CE1 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet0/0
 no ip address
 ip address 192.168.1.1 255.255.255.0
 no shutdown
 exit
!
ip route 192.168.5.0 255.255.255.0 192.168.1.2
ip route 192.168.9.0 255.255.255.0 192.168.1.2
!
```

例：PE1 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Tunnel0
 no ip address
 ip address 192.168.9.1 255.255.255.0
 tunnel source 2001:DB8:2:2::1
 tunnel destination 2001:DB8:2:4::2
 tunnel mode ipv6
 exit
!
interface Ethernet0/0
 no ip address
 ip address 192.168.1.2 255.255.255.0
 no shutdown
 exit
!
!
interface Ethernet1/1
 no ipv6 address
 ipv6 address 2001:DB8:2:2::1/64
 no shutdown
 exit
!

ipv6 route 2001:DB8:2:3::/64 2001:DB8:2:2::2
ipv6 route 2001:DB8:2:4::/64 2001:DB8:2:2::2
ip route 192.168.5.0 255.255.255.0 Tunnel 0 192.168.9.2
```

例：PE2 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Tunnel0
 no ip address
 ip address 192.168.9.2 255.255.255.0
 tunnel source 2001:DB8:2:4::2
 tunnel destination 2001:DB8:2:2::1
 tunnel mode ipv6
 exit
!
interface Ethernet0/0
 no ip address
 ip address 192.168.5.1 255.255.255.0
 no shutdown
 exit
!
interface Ethernet0/1
 no ipv6 address
 ipv6 address 2001:DB8:2:4::2/64
 no shutdown
 exit
!
!
ipv6 route 2001:DB8:2:2::/64 2001:DB8:2:4::1
ipv6 route 2001:DB8:2:3::/64 2001:DB8:2:4::1
ip route 192.168.1.0 255.255.255.0 Tunnel 0 192.168.9.1
```

例：CE2 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet0/0
 no ip address
 ip address 192.168.5.2 255.255.255.0
 no shutdown
 exit
!
ip route 192.168.1.0 255.255.255.0 192.168.1.2
ip route 192.168.9.0 255.255.255.0 192.168.1.2

!
```

例：コア デバイス 1 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet1/0
 no ipv6 address
 no shutdown
 ipv6 address 2001:DB8:2:3::1/64
 exit
!
interface Ethernet1/1
 no ipv6 address
 ipv6 address 2001:DB8:2:2::2/64
 no shutdown
 exit
!
ipv6 route 2001:DB8:2:4::/64 2001:DB8:2:3::2
```

例：コア デバイス 2 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet0/1
 no ip address
 ipv6 address 2001:DB8:2:4::1/64
 no shutdown
 exit
!
interface Ethernet1/0
 no ip address
 ipv6 address 2001:DB8:2:3::2/64
 no shutdown
 exit
!
ipv6 route 2001:DB8:2:2::/64 2001:DB8:2:3::1
```

その他の関連資料

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Command List, All Releases』
IPv6 コマンド	『Cisco IOS IPv6 Command Reference』
IPv6 アドレッシングと接続	『*IPv6 Configuration Guide*』
Cisco IOS IPv6 機能	『Cisco IOS IPv6 Feature Mapping』

標準および RFC

標準/RFC	タイトル
IPv6 に関する RFC	*IPv6 RFC*

シスコのテクニカル サポート

説明	リンク
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 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/cisco/web/support/index.html

IPv6 トンネルを介した IP 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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

表 *2*：*IPv6* トンネルを介した *IP* の機能情報

機能名	リリース	機能情報
IPv6 IP トンネルを介する IP	12.2(30)S 12.2(33)SRA 12.3(7)T 12.4 12.4(2)T 15.0(1)S Cisco IOS XE Release 2.1 15.1(1)SY Cisco IOS XE Release 3.2SE	IPv6 トンネルを介した IP の機能がサポートされます。 次のコマンドが導入または変更されました。**tunnel destination**、**tunnel mode ipv6**、**tunnel mode gre ipv6**、**tunnel source**

第 3 章

IPv4 トンネルを介して手動設定した IPv6

この機能は、IPv4 トンネルを介して手動設定した IPv6 に対するサポートを提供します。 手動で設定されたトンネルは、IPv4 バックボーンを介した 2 つの IPv6 ドメイン間の固定リンクに相当します。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 最新の注意事項と機能情報については、プラットフォームおよびソフトウェア リリースのバグ検索ツールとリリース ノートを参照してください。 このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表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シスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

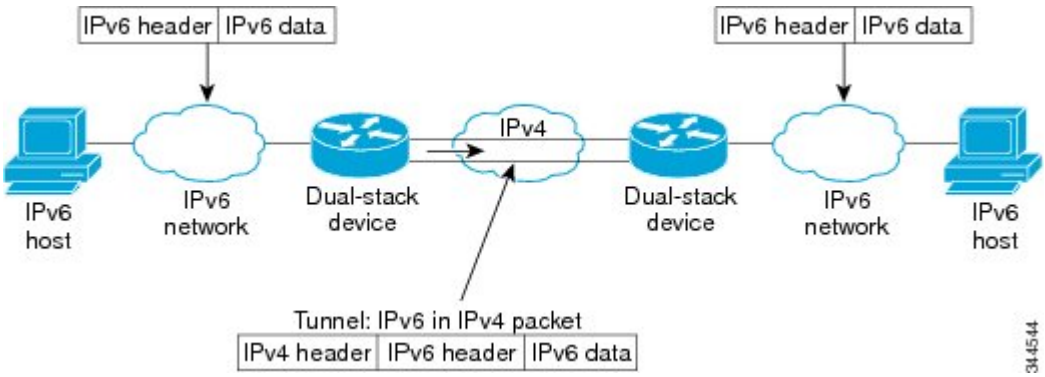
IPv4 トンネルを介して手動設定した IPv6 に関する情報

オーバーレイ トンネル for IPv6

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以下の図）へ伝送します オーバーレイ トンネルを使用することで、孤立した IPv6 ネットワークと通信できます。このとき、孤立した複数の IPv6 ネットワーク間にある IPv4 インフラストラクチャをアップグレードする必要はありません。 オーバーレイ トンネルは、境界デバイス間、または境界デバイスとホスト間に設定できますが、両方のエンドポイントが IPv4 プロトコル スタックと IPv6 プロトコル スタックの両方をサポートしている必要があります。 IPv6 では、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。

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

図 *1*：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最大伝送単位（MTU）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 と IPv6 の両方のプロトコル スタック、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への移行方法と見なす必要があります。

以下の表は、IPv4 ネットワーク上での IPv6 パケットの伝送にどのトンネル タイプを設定すればよいかを決定する場合に役立ちます。

表 *3*：*IPv4* ネットワーク上で *IPv6* パケットを伝送するトンネル タイプの推奨される使用方法

トンネリング タイプ	推奨される使用方法	使用方法
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	IPv6 パケットだけを伝送できます。
GRE および IPv4 互換	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	IPv6、コネクションレス型ネットワーク サービス（CLNS）、およびその他の多数のタイプのパケットを伝送できます。
IPv4- 互換機	ポイントツーマルチポイント トンネル	::/96 プレフィックスを使用します。 このトンネル タイプの使用は推奨しません。
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	サイトでは、2002::/16 プレフィックスからのアドレスを使用します。
6RD	IPv6 サービスは、IPv4 に IPv6 のカプセル化を使用することで IPv4 ネットワーク上のユーザに提供されます。	プレフィックスは、SP 自身のアドレス ブロックから割り当てることができます。
ISATAP	サイト内のシステム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	サイトでは、任意の IPv6 ユニキャスト アドレスを使用できます。

個々のトンネルタイプについて、このマニュアルで詳しく説明しています。 実装する特定のトンネル タイプに関する情報を確認および理解することを推奨します。 必要なトンネル タイプに精

通している場合は、以下の表で、有用と思われるトンネル設定パラメータの概要を参照してください。

表 *4*：トンネリング タイプ別のトンネル設定パラメータ

トンネリングタイプ	トンネル設定パラメータ			
トンネル モード	トンネルの送信元	トンネルの宛先	インターフェイスプレフィックスまたはアドレス	
手動	ipv6ip	IPv4アドレス、またはIPv4が設定されたインターフェイスへの参照。	IPv4 アドレス。	IPv6 アドレス。
GRE/IPv4	gre ip		IPv4 アドレス。	IPv6 アドレス。
IPv4- 互換機	ipv6ip auto-tunnel		不要。これらはすべて、ポイントツーマルチポイントのトンネリングタイプです。IPv4宛先アドレスは、パケット単位で、IPv6宛先から計算されます。	不要。インターフェイスアドレスは、*::tunnel-source*/96として生成されます。
6to4	ipv6ip 6to4			IPv6 アドレス。プレフィックスは、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。
6RD	ipv6ip 6rd			IPv6 アドレス。
ISATAP	ipv6ip isatap			変更された eui-64 形式での IPv6 プレフィックス。IPv6アドレスは、プレフィックスおよびトンネル送信元 IPv4 アドレスから生成されます。

手動で設定された IPv6 トンネル

手動で設定されたトンネルは、IPv4 バックボーンを介した 2 つの IPv6 ドメイン間の固定リンクに相当します。 主に、2 つのエッジ デバイス間またはエンド システムとエッジ デバイス間に定期的でセキュアな通信を必要とする安定した接続のために、またはリモート IPv6 ネットワークへの接続のために使用されます。

IPv6 アドレスは、トンネル インターフェイス上で手動で設定され、手動で設定された IPv4 アドレスは、トンネル送信元およびトンネル宛先に割り当てられます。 設定されたトンネルの両端にあるホストまたはデバイスは、IPv4 プロトコル スタックと IPv6 プロトコル スタックの両方をサポートしている必要があります。 手動で設定されたトンネルは、境界デバイス間または境界デバイスとホスト間で設定できます。 シスコ エクスプレス フォワーディング スイッチングは、手動で設定された IPv6 トンネルに使用できます。または、シスコ エクスプレス フォワーディング スイッチングは、プロセス スイッチングが必要な場合はディセーブルにできます。

IPv4 トンネルを介して手動設定した IPv6 を有効にする方法

手動 IPv6 トンネルの設定

はじめる前に

手動で設定された IPv6 トンネルでは、IPv6 アドレスは、トンネル インターフェイス上で設定され、手動で設定された IPv4 アドレスは、トンネル送信元およびトンネル宛先に割り当てられます。設定されたトンネルの両端にあるホストまたはデバイスは、IPv4 プロトコル スタックと IPv6 プロトコル スタックの両方をサポートしている必要があります。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. 次のいずれかのコマンドを入力します。
   - **ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*}
   - **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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • 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 tunnel** *tunnel-number* 例： `Device(config)# interface tunnel 0`	トンネル 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	次のいずれかのコマンドを入力します。 • **ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*} • **ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**] 例： `Device(config-if)# ipv6 address 3ffe:b00:c18:1::3/127`	インターフェイスに割り当てられている IPv6 ネットワークを指定し、インターフェイスで IPv6 処理をイネーブルにします。 • **eui-64** キーワードを指定すると、ソフトウェアは、インターフェイスの IPv6 アドレスを設定し、アドレスの下位 64 ビットの EUI-64 インターフェイス ID を使用してインターフェイスで IPv6 処理を有効にします。 （注） IPv6 アドレスの設定の詳細については、「Implementing IPv6 Addressing and Basic Connectivity」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： `Device(config-if)# tunnel source gigabitethernet 0/0/0`	トンネル インターフェイスの送信元 IPv4 アドレスまたは送信元インターフェイス タイプと番号を指定します。 • 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されている必要があります。
ステップ 6	**tunnel destination** *ip-address* 例： `Device(config-if)# tunnel destination 192.168.30.1`	トンネル インターフェイスの宛先 IPv4 アドレスまたはホスト名を指定します。

	コマンドまたはアクション	目的
ステップ 7	**tunnel mode ipv6ip** 例： `Device(config-if)# tunnel mode ipv6ip`	手動 IPv6 トンネルを指定します。 （注） **tunnel mode ipv6ip** コマンドでは、IPv6 をパッセンジャ プロトコルとして指定し、IPv4 を手動 IPv6 トンネル用の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： `Device(config-if)# end`	特権 EXEC モードに戻ります。

IPv4 トンネルを介して手動設定した IPv6 の設定例

例：手動 IPv6 トンネルの設定

ルータ A とルータ B 間で手動 IPv6 トンネルを設定する例を次に示します。 この例では、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 が、グローバル IPv6 アドレスを使用して手動で設定されます。 トンネル送信元およびトンネル宛先のアドレスについても、手動で設定されます。

ルータ A の設定

```
interface gigabitethernet 0/0/0
 ip address 192.168.99.1 255.255.255.0
interface tunnel 0
 ipv6 address 3ffe:b00:c18:1::3/127
 tunnel source gigabitethernet 0/0/0
 tunnel destination 2001:DB8:1111:2222::1/64
 tunnel mode ipv6ip
```

ルータ B の設定

```
interface gigabitethernet 0/0/0
 ip address 192.168.30.1 255.255.255.0
interface tunnel 0
 ipv6 address 3ffe:b00:c18:1::2/127
 tunnel source gigabitethernet 0/0/0
 tunnel destination 2001:DB8:1111:2222::2/64
 tunnel mode ipv6ip
```

例：GRE IPv4 トンネルを介した IPv6

例：CE1 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Ethernet0/0
 no ip address
 ipv6 address 2001:DB8:2:1::1/64
 no shutdown
 exit
!
!
ipv6 route 2001:DB8:2:2::/64 2001:DB8:2:1::2
ipv6 route 2001:DB8:2:4::/64 2001:DB8:2:1::2
!
```

例：PE1 の設定

```
ipv6 unicast-routing
ipv6 cef
!
interface Tunnel0
 no ip address
 ipv6 address 2001:DB8:2:4::1/64
 tunnel source 10.22.22.22
 tunnel destination 10.44.44.44
 exit
!
interface Ethernet0/0
 no ip address
 ipv6 address 2001:DB8:2:1::2/64
 no shutdown
 exit
!
interface Ethernet1/1
 no ip address
 ip address 10.22.22.22 255.255.255.0
 no shutdown
 exit
!
ip route 10.44.44.0 255.255.255.0 10.22.22.23
ipv6 route 2001:DB8:2:2::/64 Tunnel0 2001:DB8:2:4::2
```

例：PE2 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
interface Tunnel0
 no ipv6 address
 ipv6 address 2001:DB8:2:4::2/64
 tunnel source 10.44.44.44
 tunnel destination 10.22.22.22
 exit
!
interface Ethernet0/0 no ipv6 address
 ipv6 address 2001:DB8:2:2::1/64
 no shutdown
 exit
```

```
!
interface Ethernet1/0
 no ip address
 ip address 10.44.44.44 255.255.255.0
 no shutdown
 exit
!
ip route 10.22.22.0 255.255.255.0 10.44.44.43
!
ipv6 route 2001:DB8:2:1::/64 Tunnel0 2001:DB8:2:4::1
!
```

例：CE2 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
!
interface Ethernet0/0
 no ipv6 address
 ipv6 address 2001:DB8:2:2::2/64
 no shutdown
 exit
!
!
ipv6 route 2001:DB8:2:1::/64 2001:DB8:2:2::1
ipv6 route 2001:DB8:2:4::/64 2001:DB8:2:2::1
!
```

例：デバイス X の設定

```
!
interface Ethernet1/0
 no ip address
 ip address 10.44.44.43 255.255.255.0
 no shutdown
 exit
!
interface Ethernet1/1
 no ip address
 ip address 10.22.22.23 255.255.255.0
 no shutdown
 exit
!
```

例：トンネル設定の確認

CE1 から

```
Device# ping ipv6 2001:db8:2:2::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:2:2::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/9/43 ms

Device# ping ipv6 2001:db8:2:2::2 source 2001:db8:2:1::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:2:2::2, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:2:1::1
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

PE1 から

```
Device# show tunnel interface

Tunnel0
   Mode:GRE/IP, Destination 10.44.44.44, Source 10.22.22.22
   IP transport: output interface Ethernet1/1 next hop 10.22.22.23
   Application ID 1: unspecified
   Linestate - current up
   Internal linestate - current up, evaluated up
   Tunnel Source Flags: Local
   Transport IPv4 Header DF bit cleared
   OCE: IP tunnel decap
   Provider: interface Tu0, prot 47
     Performs protocol check [47]
     Protocol Handler: GRE: opt 0x0
       ptype: ipv4 [ipv4 dispatcher: punt]
       ptype: ipv6 [ipv6 dispatcher: from if Tu0]
       ptype: mpls [mpls dispatcher: drop]
       ptype: otv [mpls dispatcher: drop]
       ptype: generic [mpls dispatcher: drop]
There are 0 tunnels running over the EON IP protocol
There are 0 tunnels running over the IPinIP protocol
There are 0 tunnels running over the NOSIP protocol
There are 0 tunnels running over the IPv6inIP protocol
There are 0 tunnels running over the RBSCP/IP protocol

Device# show ip route 10.44.44.44

Routing entry for 10.44.44.0/24
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
  * 10.22.22.23
      Route metric is 0, traffic share count is 1

Device# debug ipv6 icmp

  ICMP Packet debugging is on
*Jan  1 10:57:37.882: ICMPv6: Sent R-Advert, Src=FE80::A8BB:CCFF:FE00:5200, Dst=FF02::1
*Jan  1 11:00:18.634: ICMPv6: Received R-Advert, Src=FE80::A8BB:CCFF:FE00:5200,Dst=FF02::1
```

その他の関連資料

関連資料

関連項目	マニュアル タイトル
IPv6 アドレッシングと接続	『*IPv6 Configuration Guide*』
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
IPv6 コマンド	『Cisco IOS IPv6 Command Reference』
Cisco IOS IPv6 機能	『Cisco IOS IPv6 Feature Mapping』

標準および RFC

標準/RFC	タイトル
IPv6 に関する RFC	『*IPv6 RFCs*』

MIB

MIB	MIB のリンク
	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 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/cisco/web/support/index.html

IPv4 トンネルを介して手動設定した IPv6 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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

表 *5*：*IPv4* トンネルを介して手動設定した *IPv6* の機能情報

機能名	リリース	機能情報
IPv6 トンネリング：IPv4 トンネルを介して手動設定したIPv6	Cisco IOS XE Release 2.1	手動で設定されたトンネルは、IPv4バックボーンを介した2つの IPv6 ドメイン間の固定リンクに相当します。 次のコマンドが導入または変更されました。**tunnel destination**、**tunnel ipv6ip**、**tunnel source**

第 4 章

物理インターフェイスの設定

Cisco ASR 1000 シリーズアグリゲーション サービス ルータは、ギガビット イーサネット、Packet over SONET（POS）、およびシリアル共有ポート アダプタ（SPA）インターフェイスなどさまざまな種類の物理（ハードウェア）インターフェイスをサポートします。 ハードウェアに関する技術的な説明およびインターフェイスのインストール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インストールおよび設定マニュアルを参照してください。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 最新の注意事項と機能情報については、プラットフォームおよびソフトウェア リリースのバグ検索ツールとリリース ノートを参照してください。 このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表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シスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

設定情報

- ギガビット イーサネットの管理イーサネット インターフェイスの使い方については、次の URL で『*Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide*』の「Using the Management Ethernet Interface」の章を参照してください。

http://www.cisco.com/en/US/docs/routers/asr1000/configuration/guide/chassis/asrswcfg.html

- Cisco ASR 1000 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る SPA インターフェイス プロセッサ（SIP）および SPA の設定およびトラブルシューティングについては、次の URL の『*Cisco ASR 1000 Series Aggregation Services Routers SIP and SPA Software Configuration Guide*』を参照してください。

http://cisco.com/en/US/docs/interfaces_modules/shared_port_adapters/configuration/ASR1000/ASRspasw.html

コマンド リファレンス情報

- インターフェイスの設定に使用するコマンドの詳細については、次の URL の『*Cisco IOS Interface and Hardware Component Command Reference*』に記載されています。

http://www.cisco.com/en/US/docs/ios/interface/command/reference/ir_book.html

- それ以外の Cisco IOS XE コマンドの詳細については、http://tools.cisco.com/Support/CLILookup にある Command Lookup Tool を使用するか、http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html にある『*Cisco IOS Master Command List, All Releases*』を参照してください。

第 5 章

仮想インターフェイスの設定

仮想インターフェイスは、ユーザが Cisco IOS XE コマンドを使用してネットワーキング デバイスのメモリに作成したソフトウェア ベースのインターフェイスです。 仮想インターフェイスには、100BASE-T ファスト イーサネット ネットワーク インターフェイス カード上の RJ-45 メス型ポートなどの、ハードウェア コンポーネントはありません。 このモジュールでは、Cisco IOS XE ソフトウェアを使用して設定できる 4 つの一般的な種類の仮想（論理）インターフェイスについて説明します。

- ループバック インターフェイス
- ヌル インターフェイス
- サブインターフェイス
- トンネル インターフェイス

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 最新の注意事項と機能情報については、プラットフォームおよびソフトウェア リリースのバグ検索ツールとリリース ノートを参照してください。 このモジュールに記

載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表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シスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

仮想インターフェイスの設定の前提条件

ネットワークで仮想インターフェイスを使用する前に、いくつかの物理（ハードウェア）インターフェイスを設定する必要があります。また、仮想インターフェイスを使用するネットワーキングデバイス間で通信できる必要があります。

仮想インターフェイスの設定に関する情報

仮想インターフェイス

仮想インターフェイスは、物理インターフェイスと関連付けられていないネットワーク インターフェイスです。 物理インターフェイスには、イーサネット ケーブル上の RJ-45 オス型コネクタなど、何らかの物理的な要素が存在する必要があります。 仮想インターフェイスは、ソフトウェア上にのみ存在します。物理的な要素はありません。 仮想インターフェイス名の後に、数値 ID を使用して、個々の仮想インターフェイスを指定する必要があります。 たとえば、loopback 0、tunnel 1、fastethernet 0/0/0.1 などです。 この ID は名前文字列全体を一意にするために仮想インターフェイスの種類ごとに異なります。たとえば、loopback 0 インターフェイスと null 0 インターフェイスは共存できますが、単一のネットワーキング デバイスで 2 つの loopback 0 インターフェイスは共存できません。

Cisco IOS XE ソフトウェアは次の 4 つの種類の仮想インターフェイスをサポートします。

- ループバック
- ヌル
- サブインターフェイス
- トンネル

仮想インターフェイスの利点

ループバック インターフェイスによって、IP アドレスまたは IPX アドレスなどのレイヤ 3 アドレスを割り当てることができる安定したインターフェイスが用意されます。 このアドレスは、ネットワーキング デバイスが NetFlow や Cisco Discovery Protocol（CDP）などのプロトコルのデータをネットワークの別のデバイスに送信する必要があり、その受信側デバイスにネットワーキングデバイスからの同じ送信元 IP アドレスを常に認識させたい場合に送信元アドレスとして設定でき

ます。通常の状態では、ネットワーキングデバイスによって生成されるパケットによって、アウトバウンドインターフェイスからのIPアドレスがそのパケットの送信元アドレスとして使用されたり、ネットワーキングデバイスから受信ホストへの複数の等コストパスがあるネットワークでは、各パケットによって異なるアウトバウンドインターフェイスが使用されることがあるため、ネットワークに複数の等コスト パスがある場合、これは問題になります。

ヌルインターフェイスによって、アクセスリストの使用に関連するオーバーヘッドなしでフィルタ処理の代替方式が用意されます。 たとえば、宛先ネットワークへのトラフィックをインターフェイス外へ送信できないよう、アウトバウンド アクセス リストを作成する代わりに、ヌル インターフェイスを指す宛先ネットワークに対してスタティック ルートを作成できます。

サブインターフェイスによって、共通の物理インターフェイスが共用される場合でも、IPルーティング プロトコルでは、各リモート ネットワーキング デバイスへのネットワーク接続を別々の物理インターフェイスとして認識されるよう、物理インターフェイスを複数のインターフェイスに仮想的に分割する方式として、サブインターフェイスが考案されました。 サブインターフェイスの最初の使用例の 1 つは、フレーム リレー WAN 上でのスプリット ホライズンの問題を解決することです。

トンネリング（別のプロトコルでのトラフィックのカプセル化）は、次のようないくつかの場合に、役に立ちます。

- 1 つのプロトコル バックボーンを介してマルチプロトコル ローカル ネットワークをイネーブルにする場合
- RIP バージョン 1、AppleTalk など、ホップ カウントが限定されているプロトコルが使用されているネットワークで、回避策を用意する場合
- 隣接していないサブネットワークを接続する場合
- WAN を介してバーチャル プライベート ネットワークを使用できるようにする場合

ループバック インターフェイス

ループバック インターフェイスと呼ばれるソフトウェアのみのインターフェイスを指定して、物理インターフェイスをエミュレートできます。 ループバック インターフェイスは、すべてのプラットフォームでサポートされます。 ループバック インターフェイスは、**no shutdown** コマンドを使用後、**shutdown** コマンドでディセーブルにするまでアップ（アクティブ）の状態にあるCiscoルータ上の仮想インターフェイスです。 サブインターフェイスと異なり、ループバックインターフェイスは、どの物理インターフェイスの状態にも依存しません。

ループバック インターフェイスを一度イネーブルにすると、シャットダウンするまでアップのままのため、ループバックインターフェイスは安定していると見なすことができます。 ネットワーキング デバイスで、いずれの物理インターフェイスのステータスにも依存しない参照先として 1 つのアドレスが必要な場合、IP アドレスなどのレイヤ 3 アドレスの割り当てには、ループバックインターフェイスが適しています。 これに適した例は、ループバック インターフェイスの IP アドレスを、ネットワーキング デバイスのドメイン ネーム システム（DNS）のホストアドレスのIPアドレスとして使用する場合です。 ルータを管理するときに、任意の時点で使用可能となる可能性があるインターフェイスIPアドレスが確かではないため、ループバックインターフェイスが

使用できるようになる前に、ネットワーク管理者は割り当てられたIPアドレスがあるルータ上の各インターフェイスのDNSホスト エントリを設定する必要があります。 次に示されたルータAのインターフェイス設定とDNSエントリの例では、各インターフェイスに1つのDNSエントリが設定されています。

ループバック前のルータ A のインターフェイス設定

```
GigabitEthernet0 10.10.10.1 255.255.255.0
GigabitEthernet1 10.10.11.1 255.255.255.0
GigabitEthernet2 10.10.12.1 255.255.255.0
GigabitEthernet3 10.10.13.1 255.255.255.0
GigabitEthernet4 10.10.14.1 255.255.255.0
GigabitEthernet5 10.10.15.1 255.255.255.0
```

ループバック前のルータ A の DNS エントリ

```
RouterA   IN  A  10.10.10.1
          IN  A  10.10.11.1
          IN  A  10.10.12.1
          IN  A  10.10.13.1
          IN  A  10.10.14.1
          IN  A  10.10.15.1
```

ネットワーキング デバイスのインターフェイスは、障害が発生したり、メンテナンスのために運転を停止する場合があります。 ルータAのいずれかのルータに障害が発生するか、アウト オブ サービスの場合、別のネットワーキング デバイスからそのインターフェイスにアクセスできません。ループバック インターフェイスを使用してネットワーキング デバイスを設定し、ネットワーク全体でアドバタイズされるIPアドレスをデバイスに割り当てる場合、ネットワーキング デバイスにIPトラフィックを送受信できるネットワーク インターフェイスが少なくとも1つある限り、このIPアドレスを使用してネットワーキング デバイスに到達できます。 ループバック インターフェイス設定後のルータ A でのインターフェイス設定とDNSエントリの例では、その物理インターフェイスのいずれかを介してルータに到達するために、1つのDNSエントリのみがあることがわかります。

ループバック後のルータ A のインターフェイス設定

```
Loopback 172.16.78.1 255.255.255.0
GigabitEthernet0 10.10.10.1 255.255.255.0
GigabitEthernet1 10.10.11.1 255.255.255.0
GigabitEthernet2 10.10.12.1 255.255.255.0
GigabitEthernet3 10.10.13.1 255.255.255.0
GigabitEthernet4 10.10.14.1 255.255.255.0
GigabitEthernet5 10.10.15.1 255.255.255.0
```

ループバック後のルータ A の DNS エントリ

```
RouterA   IN  A  172.16.78.1
```

ループバック インターフェイスに設定されているIPアドレス172.16.78.1は、ルータによって生成されるパケットの送信元アドレスとして使用でき、ネットワーキング管理アプリケーションおよびルーティング プロトコルに転送されます。 このループバック インターフェイスは、明示的にシャットダウンされない限り、常に到達可能です。

ループバック インターフェイスは、Open Shortest Path First（OSPF）またはボーダー ゲートウェイ プロトコル（BGP）セッションの終端アドレスとして使用できます。 すべての他のインターフェ

イスがダウンしている場合、ループバックインターフェイスは、デバイスのコンソールポートからその補助ポートへの Telnet セッションを確立する場合にも使用できます。 他のルータまたはアクセスサーバがこのループバックインターフェイスに到達しようとしているアプリケーションでは、ループバックアドレスに割り当てられたサブネットを配信するよう、ルーティングプロトコルを設定する必要があります。

ループバックインターフェイスにルーティングされたIPパケットは、ルータまたはアクセスサーバに再びルーティングされ、ローカルで処理されます。ループバックインターフェイス外にルーティングされるがループバックインターフェイス宛てで送信されないIPパケットは、ドロップされます。 これらの 2 つの状況では、ループバック インターフェイスはヌル インターフェイスのように動作できます。

ループバック インターフェイスとループバック モード

ループバック インターフェイスは安定した発信元インターフェイスを実現するもので、IP ルーティングプロトコルがループバックインターフェイスに割り当てられたサブネットをアドバタイズする限り、発信元インターフェイスに割り当てられたIPアドレスがいつでも到達可能になるようにします。 ただし、ループバックモードを使用して、ビット喪失またはデータ破損などのWAN（シリアル）リンクでの問題がテストされ、診断されます。 この考えは、トラフィックの送信元のデバイスに同じインターフェイスがバックアウトされるインターフェイスで受信したデータパケットが戻されるよう、ループを設定することです。 送信されたときと同じ条件でデータパケットが戻されることをチェックすることによって、ループバック モードを使用して問題のトラブルシューティングが行われます。 データ パケットでのエラーは、WAN インフラストラクチャでの問題を示します。多くの種類のシリアルインターフェイスには、インターフェイスまたはコントローラ コンフィギュレーション モードで入力される独自の形式のループバック コマンド構文があります。

ヌル インターフェイス

ヌルインターフェイスは、ループバックインターフェイスに類似した仮想ネットワークインターフェイスです。ループバックインターフェイスへのトラフィックは、ルータ自体に宛てて送信される一方で、ヌル インターフェイスに送信されるトラフィックは廃棄されます。 このインターフェイスは常にアップで、トラフィックの転送や受信はできません。カプセル化は常に失敗します。 ヌル インターフェイスは、ほとんどのオペレーティング システムで使用可能なヌル デバイスと同様に機能します。

ヌルインターフェイスは、不必要なネットワークトラフィックを廃棄する、オーバーヘッドが低い方式として使用されます。 たとえば、ネットワーク ユーザが、特定の IP サブネットに到達できないようにする場合、ネットワーキングデバイスのヌルインターフェイスを指すサブネットに対して、スタティック IP ルートを作成できます。 スタティック IP ルートの使用は、IP アクセスリストを使用する場合よりも、ネットワーキング デバイスでより少ない CPU が消費されます。また、スタティック ルート設定は、インターフェイス コンフィギュレーション モードではなくグローバル コンフィギュレーション モードで行われるため、IP アクセス リストよりも簡単に設定できます。

ヌル インターフェイスは、1 つのアドレスで設定できない場合があります。 Null 0 と示されるヌルインターフェイスがネクストホップであるスタティック ルートを設定することによってのみ、このインターフェイスにトラフィックを送信できます。 ネクスト ホップをヌル インターフェイスに設定する 1 つの例は、BGP を介してアナウンスメントできるように集約ネットワークにルートを設定する場合や、セキュリティなどの目的で、特定のアドレス範囲へのトラフィックが、ルータを介して送信されないようにする場合です。

ルータには、常に 1 つのヌルインターフェイスがあります。デフォルトでは、ヌルインターフェイスに送信されるパケットによって、パケットの発信元 IP アドレスにインターネット制御メッセージ プロトコル（ICMP）の到達不能メッセージを送信することによって、ルータが応答します。 これらの応答を送信するか、パケットを静かにドロップするかのいずれかが行われるよう、ルータを設定できます。

サブインターフェイス

サブインターフェイスは、物理インターフェイスに関連付けられます。 サブインターフェイスが関連付けられている物理インターフェイスがイネーブルにされると、サブインターフェイスがイネーブルになり、物理インターフェイスがシャットダウンされると、サブインターフェイスはディセーブルになります。

（注） サブインターフェイスは、関連付けられている物理インターフェイスとは別に、イネーブルにしたりシャットダウンしたりすることができます。 ただし、シャットダウンされた物理インターフェイスのサブインターフェイスは、イネーブルにできません。

物理インターフェイスを、IP サブネットなどの固有のレイヤ 3 ネットワーク アドレスを割り当てることができる複数の仮想インターフェイスに分割することによって、サブインターフェイスが作成されます。 サブインターフェイスの最初の使用例の 1 つは、フレーム リレー WAN 上でのスプリット ホライズンの問題を解決することです。 スプリット ホライズンは、IP サブネットが認識されたのと同じ物理インターフェイスからアドバタイズされない、RIP などの IP ルーティング プロトコルに関連付けられた動作です。 スプリットホライズンは、IP ネットワークでのルーティングのループを防ぐために実装されました。 ネットワーク接続の両側のネットワーキング デバイスによって、お互いに同じ IP ルートがアドバタイズされるときに、ルーティングのループが発生します。 多くのネットワーキング デバイスのデフォルト動作はスプリット ホライズンを実装することが目的であるため、1 つの物理インターフェイスを介して複数のリモート ネットワーキング デバイスに接続されたフレーム リレー マルチポイント ネットワーク インターフェイスでは、スプリット ホライズンが問題となりました。これは、同じ物理インターフェイスを介しても到達可能であった他のデバイスにインターフェイスがバック アウトされるインターフェイスを介して認識された IP ルートは、ネットワーキング デバイスによってアドバタイズされないことを意味します。 サブインターフェイスによって、共通の物理インターフェイスが共用される場合でも、IP ルーティング プロトコルでは、各リモート ネットワーキング デバイスへのネットワーク接続を別々の物理インターフェイスとして認識されるよう、物理インターフェイスを複数のインターフェイスに仮想的に分割する方式として、サブインターフェイスが考案されました。 TCP/IP では、デフォルトで、スプリットホライズンの制限がディセーブルにされるようになりましたが、AppleTalk や IPX などのプロトコルは、引き続きスプリット ホライズンによる制約を受けます。

サブインターフェイスは、Hardware Interface Descriptor（IDB）、ピリオド記号、プレフィックスで固有の番号の順で構成されるプレフィックスによって指定されます。 フルサブインターフェイス番号は、ネットワーキング デバイスに対して固有である必要があります。 たとえば、GigabitEthernet インターフェイス 0/0/0 の最初のサブインターフェイスの名前は GigabitEthernet 0/0/0.1 となります。ここで .1 は最初のサブインターフェイスを示します。

トンネル インターフェイス

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「passenger」プロトコルまたは「transport」プロトコルには関連付けられていませんが、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の実装に必要なサービスが提供されるように設計されたアーキテクチャです。

提供する必要がある接続によって、トンネル インターフェイスを実装するいくつかの方法があります。 トンネルの 1 つの共通の使用目的は、IPX がサポートされないネットワークでのデバイスを介した IPX などのネットワーク プロトコルに、データ トラフィックを運ぶことです。 たとえば、ネットワークのコアではなくエッジにあるサイトで、ネットワークによって IPX が使用される場合、ネットワークのコアを介して IP の IPX をトンネリングすることによって、ネットワークエッジにある IPX サイトに接続できます。

ソフトウェアを使用して利用できるさまざまな種類のトンネリング技術の詳細については、『*Cisco IOS XE Interface and Hardware Component Configuration Guide*』の「Implementing Tunnels」モジュールを参照してください。

仮想インターフェイスの設定方法

ループバック インターフェイスの設定

この作業では、ループバック インターフェイスを設定する方法について説明します。ループバック インターフェイスを一度イネーブルにすると、シャットダウンするまでアップのままのため、ループバック インターフェイスは安定していると見なすことができます。 このため、ループバック インターフェイスはネットワーキング デバイスのいずれの物理インターフェイスの状態にも影響を受けない参照先として使用する単一アドレスが必要なときに IP アドレスなどのレイヤ 3 アドレスを割り当てる場合に理想的です。

はじめる前に

ループバック インターフェイスの IP アドレスは、固有である必要があり、別のインターフェイスによっては使用されません。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface loopback** *number*
4. **ip address** *ip-address mask* [**secondary**]
5. **end**
6. **show interfaces loopback** *number*
7. **exit**

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。 • 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： `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 loopback** *number* 例： `Router(config)# interface loopback 0`	ループバック インターフェイスを指定し、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。 • 作成または設定するループバック インターフェイスの数を指定する場合は、*number* 引数を使用します。 （注） 作成可能なループバック インターフェイスの数に制限はありません。
ステップ 4	**ip address** *ip-address mask* [**secondary**] 例： `Router(config-if)# ip address 10.20.1.2 255.255.255.0`	ループバック インターフェイスの IP アドレスを指定し、インターフェイス上で IP の処理をイネーブルにします。 • ループバック アドレスのサブネットを指定する場合は、*ip-address* 引数と *mask* 引数を使用します。
ステップ 5	**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。

	コマンドまたはアクション	目的
ステップ 6	**show interfaces loopback** *number* 例： `Router# show interfaces loopback 0`	（任意）ループバック インターフェイスに関する詳細情報を表示します。 • ある特定のループバック インターフェイスに関する情報を表示する場合は、*number* 引数を使用します。 （注） この例では、このタスクに適用可能な構文だけが使用されています。 詳細については、『Cisco IOS Interface and Hardware Component Command Reference』を参照してください。
ステップ 7	**exit** 例： `Router# exit`	特権 EXEC モードを終了します。

例

次に、**show interfaces loopback** コマンドからの出力例を示します。

```
Router# show interfaces loopback
Loopback0 is up, line protocol is up
  Hardware is Loopback
  Internet address is 10.20.1.2/24
  MTU 1514 bytes, BW 8000000 Kbit, DLY 5000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation LOOPBACK, loopback not set
  Last input never, output never, 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
     0 packets input, 0 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
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

ヌル インターフェイスの設定

この作業では、ヌル インターフェイスを設定する方法について説明します。 ヌル インターフェイスによって、トラフィックのフィルタ処理のためのアクセスコントロールリストの代替方式が提供されます。 すべての不要なトラフィックはヌル インターフェイス宛てにすることができます。ヌル インターフェイスではトラフィックの送受信を行えず、そのトラフィックがカプセル化されます。

ヌル インターフェイスに指定できる唯一のインターフェイス コンフィギュレーション コマンドは、**no ip unreachables** コマンドです。

ヌル インターフェイスからの ICMP 到達不能メッセージ

ヌル インターフェイスに送信したパケットに対する応答で、ICMP 到達不能メッセージの送信を無効にするには、インターフェイス コンフィギュレーション モードで **no ip unreachables** コマンドを使用します。ヌルインターフェイスに送信したパケットに対する応答で、ICMP到達不能メッセージの送信を再度有効にするには、インターフェイス コンフィギュレーション モードで **ip unreachables** コマンドを使用します。

デバイスには 1 個のヌル インターフェイスのみを設定できます。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface null** *number*
4. **no ip unreachables**
5. **end**
6. **show interfaces null** [*number*] [**accounting**]

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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • 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 null** *number* 例： `Device(config)# interface null 0`	ヌル インターフェイスと番号を指定し、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。 • 引数 number は、常に 0 です。

	コマンドまたはアクション	目的
ステップ 4	**no ip unreachables** 例： `Device(config-if)# no ip unreachables`	インターフェイス上で ICMP 到達不能メッセージの生成を防ぎます。 • このコマンドは、すべてのタイプの ICMP 到達不能メッセージに影響を及ぼします。
ステップ 5	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show interfaces null** [*number*] [**accounting**] 例： `Device# show interfaces null 0`	（任意）ヌル インターフェイスに関する詳細情報を表示します。 • ヌルインターフェイスでは、引数 *number* は常に 0 です。 （注） この例では、このタスクに適用可能な構文だけが使用されています。 詳細については、『Cisco IOS Interface and Hardware Component Command Reference』を参照してください。

例

次に、**show interfaces null** コマンドからの出力例を示します。

```
Device# show interfaces null

Null0 is up, line protocol is up
  Hardware is Unknown
  MTU 1500 bytes, BW 10000000 Kbit, DLY 0 usec,
     reliability 0/255, txload 0/255, rxload 0/255
  Encapsulation ARPA, loopback not set
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

サブインターフェイスの設定

この作業では、サブインターフェイスを設定する方法について説明します。 サブインターフェイスは、関連付けられている物理インターフェイスとは別に、イネーブルにしたりシャットダウン

したりすることができます。 ただし、シャットダウンされた物理インターフェイスのサブインターフェイスは、イネーブルにできません。

はじめる前に

インターフェイスの IP アドレスは、固有である必要があり、別のインターフェイスによっては使用されません。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *type number.subinterface-number*
4. **ip address** *ip-address mask* [**secondary**]
5. **end**
6. **show interfaces** *type number.subinterface-number*
7. **exit**

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。 • 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： `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.subinterface-number* 例： `Router(config)# interface GigabitEthernet 2/3.5`	インターフェイス タイプ、インターフェイス番号、およびサブインターフェイス番号を指定し、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	**ip address** *ip-address mask* [**secondary**] 例： `Router(config-if)# ip address 209.165.200.225 255.255.255.224`	インターフェイスの IP アドレスを指定し、インターフェイス上で IP の処理をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 5	**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。
ステップ 6	**show interfaces** *type number.subinterface-number* 例： `Router# show interfaces GigabitEthernet 2/3.5`	（任意）インターフェイスに関する詳細情報を表示します。
ステップ 7	**exit** 例： `Router# exit`	特権 EXEC モードを終了します。

例

次に、**show interfaces** コマンドの出力例を示します。

```
Router# show interfaces GigabitEthernet 2/3.5
GigabitEthernet2/3.5432 is down, line protocol is down (notconnect)
  Hardware is c7600 1Gb 802.3, address is 001b.0de6.c100 (bia 001b.0de6.c100)
  Description: *sample*
  Internet address is 10.11.12.13/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID  2339.
  ARP type: ARPA, ARP Timeout 04:00:00
  Keepalive set (10 sec)
  Last clearing of "show interface" counters never
```

仮想インターフェイスの設定例

ループバック インターフェイスの設定例

次に、ループバック インターフェイス loopback 0 の設定シーケンスを示します。

```
interface loopback 0
 ip address 209.165.200.225 255.255.255.0
 end
```

ヌル インターフェイスの設定例

次に、ヌル インターフェイスの設定シーケンスおよび ICMP 到達不能メッセージをドロップする例を示します。ヌルインターフェイスに送信されるすべてのパケットはドロップされ、また、この例では、通常はヌル インターフェイスに送信されたパケットに対する応答として送信される ICMP メッセージがドロップされます。

```
interface null 0
 no ip unreachables
 end
```

サブインターフェイスの設定例

次に、サブインターフェイスの設定シーケンスを示します。

```
interface GigabitEthernet 2/3.5
 description *sample*
 encapsulation dot1Q 2339
 ip address 209.165.200.225 255.255.255.224
end
```

次の作業

- ネットワークでトンネルを導入する場合は、『*Cisco IOS XE Interface and Hardware Component Configuration Guide*』の「Implementing Tunnels」モジュールを参照してください。
- ネットワークで物理（ハードウェア）インターフェイス（ギガビットイーサネットやシリアルインターフェイスなど）を導入する場合は、『*Cisco IOS XE Interface and Hardware Component Configuration Guide*』の「Configuring Physical Interfaces」モジュールを参照してください。

その他の関連資料

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
インターフェイス コマンド：コマンド構文の詳細、コマンドモード、デフォルト設定、コマンド履歴、使用に関する注意事項および例	『*Cisco IOS Interface and Hardware Component Command Reference*』
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*』

関連項目	マニュアル タイトル
BGP でのループバック インターフェイスの使用方法を示した設定例	『Sample Configuration for iBGP and eBGP With or Without a Loopback Address』

標準

標準	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 のリンク
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 ソフトウェア リリース、およびフィーチャ セットの MIB を検索してダウンロードする場合は、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
新しい RFC または変更された RFC はサポートされていません。また、既存の RFC に対するサポートに変更はありません。	--

シスコのテクニカル サポート

説明	リンク
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 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/cisco/web/support/index.html

第 6 章

トンネルの実装

このモジュールでは、トンネリング技術のさまざまなタイプを示します。 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、設定の詳細や設定例を示しています。 テクノロジー固有のコマンドを使用して多数のトンネリング技術が実装されており、テクノロジーに関して対応するモジュールへのリンクが示されています。

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、設定のために簡単なインターフェイス提供するための仮想インターフェイスとして実装されます。 トンネル インターフェイスは、特定の「passenger」プロトコルまたは「transport」プロトコルには関連付けられていませんが、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の実装に必要なサービスを提供するアーキテクチャです。

（注） Cisco ASR 1000 シリーズ アグリゲーション サービス ルータは、VPN ルーティング/転送（VRF）Aware の総称ルーティング カプセル化（GRE）トンネル キープ アライブ機能をサポートします。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 最新の注意事項と機能情報については、プラットフォームおよびソフトウェア リリースのバグ検索ツールとリリース ノートを参照してください。 このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表を参照してください。

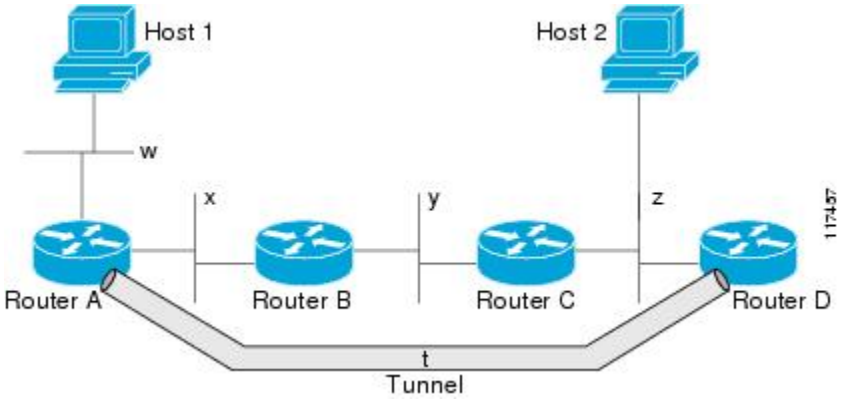
プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。 Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

トンネル実装の制約事項

- トンネル プロトコルにファイアウォールとアクセス コントロール リスト（ACL）チェックのパススルーを許可することが必要です。
- 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可能性があります。
- トンネルは単一のホップ リンクに似ており、ルーティング プロトコルはマルチホップ物理パスを経由するトンネルを優先することがあります。 トンネルは単一ホップ リンクであるかどうかに関係なく、マルチ ホップ リンクよりも低速パスを通過する場合があります。 トンネルは、実際に通過するリンクと同様に堅牢で高速であったり、信頼性が低く低速であったりします。 ホップ カウントだけに基づいて決定を行うルーティング プロトコルは、物理リンクのセットを経由するトンネルを優先することが多くなります。 トンネルは、1 つのホップのポイントツーポイント リンクで、パスのコストが最も低いように思われますが、代替物理トポロジと比較した場合、遅延の観点から見ると実際にはコストがかかる場合があります。 たとえば、以下の図に示すトポロジでは、ホスト 1 からのパケットは、w、x、y、z の 4 つのパスを使用する代わりに、トンネルのホップ 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 を通過してホスト 2 へ送信されると考えられます。 ただし実際には、

トンネルを通って送信されるパケットは、ルータ A、B、C を通過して、さらにルータ D まで移動してからルータ C に戻る必要があります。

図 *2*：トンネルに関する注意事項：ホップ カウント



- ルーティングが正しく設定されていない場合、トンネルに再帰ルーティングの問題がある可能性があります。 トンネル宛先への最良パスはトンネル自身です。そのため再帰ルーティングによってトンネル 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、常にコントロール プレーン ルーティングをトンネル ルーティングとは別個にします。
  - 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（ルーティング ループがないかどうか監視）をオーバーライドします。

次のエラーは、再帰ルーティングがトンネルの宛先にあるときに表示されます。

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

トンネル実装に関する情報

トンネリングとカプセル化

トンネルがどのように動作するかを理解するには、カプセル化とトンネリングの概念を区別できる必要があります。 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。 開放型システム間相互接続（OSI）参照モデルは、ネットワークの機能について説明します。1 個のホスト（PC など）からネットワーク上の別のホストにデータ パケットを送信するには、カプセル化を使用して、プロトコルスタックの各レイヤで、データ パケットの前にヘッダーを降順で追加します。 ヘッダーには、現在のレイヤのすぐ上にある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ことが必要です。 ネッ

トワークの受信側でパケットがプロトコル スタックを上っていくと、カプセル化された各ヘッダーは逆順に削除されます。

トンネリングでは、別のプロトコル内の 1 つのプロトコルからデータ パケットをカプセル化し、外部ネットワーク上のパケットを転送します。 トンネリングは、カプセル化とは異なり、より低いレイヤのプロトコルと同じレベルのレイヤのプロトコルが、トンネルを通って送信されるようにします。 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。 トンネリングは、次の 3 種類の主要コンポーネントから構成されます。

- 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。 たとえば、IPv4 プロトコルや IPv6 プロトコルなどです。
- キャリア プロトコル：カプセル化するプロトコル。 たとえば、総称ルーティング カプセル化（GRE）やマルチプロトコル ラベル スイッチング（MPLS）などです。
- トランスポート プロトコル：カプセル化したプロトコルを伝送するプロトコル。 主なトランスポート プロトコルは IP です。

Tunnel ToS（ToS）

タイプ オブ サービス（ToS）により、ネットワーク トラフィックをトンネリングして、すべてのパケットを同じ ToS バイト値に分類できます。 ToS バイト値および存続可能時間（TTL）のホップ カウント値は、ルータの IP トンネル インターフェイス向けのトンネル パケットのカプセル化 IP ヘッダーに設定できます。 Tunnel ToS 機能は、シスコ エクスプレス フォワーディング（旧称：CEF）、高速スイッチング、およびプロセス スイッチングでサポートされます。

ToS および TTL バイト値は、RFC 791 で定義されています。 RFC 791 に定義されているとおり、RFC 2474 および RFC 2780 では ToS バイトの使用を廃止しています。 RFC 791 では、ToS バイトのビット 6 と 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 バイト値を使用し、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。

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 トンネリングとも呼ばれています。

EoMPLS は、長距離のレイヤ 2 拡張に非常に有効です。 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 ラベル スイッチド パス（LSP）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。 カプセル化後、元のデータペイロードと独立したIPヘッダーであった部分は、GREペイロードと呼ばれます。 プロトコルの種類の情報と再計算されたチェックサムを示すために、GREヘッダーがパケットに追加されます。 GREヘッダーの前にIPヘッダーも追加されます。 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、レイヤ2 VPNおよびレイヤ3 VPN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、パケットがトンネル宛先エンドポイントに到着した時点で、新しいIPヘッダーとGREヘッダーがパケットから削除され、その後はパケットの最終宛先への転送に元のIPヘッダーが使用されます。

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、プロバイダーエッジ（PE）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。

プロバイダー エッジからプロバイダー エッジ総称ルーティング カプセル化へのトンネル

プロバイダー エッジからプロバイダー エッジ（PE）GREへのトンネルのシナリオでは、お客様は、コアのどの部分もMPLSへ移行せずに、EoMPLSおよび基本MPLS VPNサービスを提供することを選択します。 したがって、MPLSトラフィックのGREトンネリングはPE間で行われます。

プロバイダーからプロバイダー総称ルーティング カプセル化へのトンネル

プロバイダーからプロバイダー（P）GREへのトンネルのシナリオでは、プロバイダー エッジ（PE）ルータとPルータの間でマルチプロトコル ラベル スイッチング（MPLS）が有効ですが、ネットワーク コアでは非MPLS認識ルータまたはIP暗号化ボックスが使用できます。 このシナリオでは、Pルータ間で、MPLSラベル付きパケットのGREトンネリングが実行されます。

プロバイダー エッジからプロバイダー総称ルーティング カプセル化へのトンネル

プロバイダーエッジからプロバイダーGREへのトンネルのシナリオでは、ネットワークにMPLS認識のP-to-Pノードがあります。 GREトンネリングは、PE-to-Pの非MPLSネットワーク セグメント間で行われます。

総称ルーティング カプセル化に固有の機能

導入シナリオに関して次の設定と情報を理解しておく必要があります。

- 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）に対する双方向フォワーディング検出（BFD）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、ギガビット、Packet over SONET（POS）、TenGigabit 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

Ethernet over MPLS に固有の機能

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

マルチプロトコル ラベル スイッチングの仮想プライベート ネットワークに固有の機能

- IPv4 VRF での PE ロールのサポート
- すべての PE to カスタマー エッジ（CE）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
- マルチプロトコル ラベル スイッチング（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、「例：EoMPLS over GRE の設定」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』マニュアルを参照してください。

パス MTU ディスカバリ

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インターネット制御メッセージ プロトコル（ICMP）メッセージが返されます。 このメッセージには、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）と、パケットがドロップされる原因となったリンクの MTU が含まれています。

（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で PMTUD を有効にするには、**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 Quality of Service（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してさまざまな 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）のトンネル インターフェイスへの直接適用や、Modular QoS CLI（MQC）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、専用アクセス レート（CAR）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、重み付け均等化キューイング（WFQ）、重み付けランダム早期検出（WRED）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、ユーザがトンネル インターフェイス上またはクリプト マップ上で QoS の事前分類機能を適用する際に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）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 インターフェイス上に一部の QoS 機能を導入する方法については、32 ページの「トンネル インターフェイスでの QoS オプションの設定：例, （65 ページ）」の項を参照してください。

トンネルの実装方法

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

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。

手順の詳細

ステップ 1 パッセンジャ プロトコルを決定します。 パッセンジャ プロトコルはカプセル化の対象となるプロトコルです。

ステップ 2 必要に応じて、**tunnel mode** コマンド キーワードを決定します。
次の表に **tunnel mode** コマンドで使用する適切なキーワードを設定する例を示します。

表 *6*：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	Distance Vector Multicast Routing Protocol（DVMRP）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** キーワードおよび **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** キーワードおよび **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** キーワードを使用します。 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、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。
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 トンネルを指定できます。
mpls	トラフィック エンジニアリング（TE）トンネルの設定に **mpls** キーワードを使用します。

IPv4 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。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。トンネルの片方の端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ドロップされます。

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でも送信できます。

はじめる前に

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。 ハードウェアに関する技術的な説明およびインターフェイスのインストール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インストールおよび設定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** | **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。 • 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： `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。 • トンネルを設定するには、*type* 引数に **tunnel** を使用します。
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 パラメータにすぎないため、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* 引数と *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。 機能のこの組み合わせ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*}	トンネル送信元を設定します。

	コマンドまたはアクション	目的
	例： `Router(config-if)# tunnel source GigabitEthernet 0/0/0`	• 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場合は、*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 IP アドレスと宛先の 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 10.0.2.1`	トンネル宛先を設定します。 • ホストの宛先の名前を指定するには、*hostname* 引数を使用します。 • ホストの宛先の IP アドレスを指定する場合は、*ip-address* 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 8	**tunnel key** *key-number* 例： `Router(config-if)# tunnel key 1000`	（任意）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** | **multipoint**} 例： `Device(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。 • マルチポイント GRE（mGRE）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： `Device(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 ソフトウェアは DF ビ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 を持っている必要があります。

	コマンドまたはアクション	目的
		• IPv6 パケットに対しては、**ipv6 mtu** コマンドを使用します。 （注） **tunnel path-mtu-discovery** コマンドが有効になっている場合、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： `Device(config-if)# ip tcp mss 250`	（任意）ルータ上で生成または終了する TCP 接続に対して、最大セグメント サイズ（MSS）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： `Device(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 PMTUD を有効にします。 • トンネルインターフェイスでPMTUDがイネーブルの場合、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： `Device(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」の項に進みます。

6to4 トンネルの設定

はじめる前に

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、プレフィックス 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 トンネルの両端の境界ルータは、IPv4 プロトコル スタックと IPv6 プロトコル スタックの両方をサポートしている必要があります。

（注） IPv4 互換トンネル 1 つだけの設定、および 6to4 IPv6 トンネル 1 つだけの設定が、1 台のルータ上でサポートされます。 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、シスコは、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 ストリームからのパケット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ために、トンネルの送信元しか使用できないためです。 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。

手動で設定された IPv6 トンネルの場合、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 宛先が両方とも定義されているため、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*
9. **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。 • 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： `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。

	コマンドまたはアクション	目的
ステップ 3	**interface tunnel** *tunnel-number* 例： `Router(config)# interface tunnel 0`	トンネル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た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 2002::/16 プレフィックスに続く 32 ビットは、トンネル送信元に割り当てられたIPv4アドレスに対応します。 （注） IPv6アドレスの設定の詳細については、「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`	トンネル インターフェイスの送信元 IPv4 アドレスまたは送信元インターフェイス タイプと番号を指定します。 （注） **tunnel source** コマンドで指定した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必要があります。
ステップ 6	**tunnel mode ipv6ip 6to4** 例： `Router(config-if)# tunnel mode ipv6ip 6to4`	6to4 アドレスを使用する IPv6 オーバーレイ トンネルを指定します。
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`	指定したトンネルインターフェイスへのスタティック 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場合は、6to4 トンネル インターフェイスに IPv6 6to4 プレフィックス 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** コマンドで指定したトンネル番号は、**interface tunnel** コマンドで指定したトンネル番号と同じである必要があります。

	コマンドまたはアクション	目的
ステップ 9	**end** 例： `Router(config)# end`	グローバル 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」の項に進みます。

トンネルの設定と動作の確認

以下の手順にある **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。 パスワードを入力します（要求された場合）。

例：
```
Device> enable
```

ステップ 2 **show interfaces tunnel** *number* [**accounting**]
2 台のルータがトンネルのエンドポイントとして設定されます。デバイス A では、IPv4 アドレスが 10.0.0.1、IPv6 プレフィックスが 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、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。 デバイス B では、IPv4 アドレスが 10.0.0.2、IPv6 プレフィックスが 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。

トンネル送信元およびトンネル宛先のアドレスが設定されていることを確認するには、**show interfaces tunnel** コマンドをデバイス A で使用します。

例：

```
Device A# 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チェックするには、デバイス A で **ping** コマンドを使用します。

例：

```
DeviceA# 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 アドレスへのルートが存在することを確認するには、**show ip route** コマンドを次のように使用します。

例：

```
DeviceA# 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 アドレスに到着できることを確認するには、**ping** コマンドをデバイス A で使用します。

（注） フィルタリングが原因で、**ping** コマンドを使用してリモートエンドポイントアドレスに到着できない場合がありますが、トンネル トラフィックは依然としてその宛先に到着している場合があります。

例：

```
DeviceA# 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着可能であることを確認するには、デバイス A で **ping** コマンドを再び使用します。 この手順の前半のフィルタリングに関する説明も、この例に適用されます。

例：

```
DeviceA# ping 2001:0DB8:1111:2222::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繰り返すことができます。

トンネル実装の設定例

例：GRE IPv4 トンネルの設定

GRE トンネリングの単純な設定例を次に示します。 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
```

例：EoMPLS over GRE の設定

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
Router B Configuration
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 2/1/1
 mtu 9216
```

```
 no ip address
 xconnect 209.165.200.237 200 encapsulation mpls
!
interface TenGigabitEthernet 2/3/1
 mtu 9216
 no ip address
!
interface TenGigabitEthernet 2/3.11/1
 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
```

トンネル インターフェイスでの 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、MQC コマンドを使用して同じシェーピング ポリシーをトンネル インターフェイスに適用する方法を示しています。

```
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。論理インターフェイス（この例に挙げているトンネルインターフェイス）では本来、輻輳状態はサポートされておらず、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はサポートされていません。代わりに、階層型ポリシーを適用します。**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
```

クラスベースシェーピングを適用する「親」またはトップレベルの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比率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
```

その他の関連資料

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	マニュアル タイトル
すべての Cisco IOS XE コマンド	『Cisco IOS Master Command List, All Releases』
トンネルコマンド：コマンド構文の詳細、コマンドモード、デフォルト設定、コマンド履歴、使用に関する注意事項および例	『*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、デフォルト設定、コマンド履歴、使用に関する注意事項および例	『*Cisco IOS IPv6 Command Reference*』

関連項目	マニュアル タイトル
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*』
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*』
VRF 対応ダイナミック マルチポイント VPN（DMVPN）の設定例	『*Cisco IOS XE Security Configuration Guide: Secure Connectivity*』の「Dynamic Multipoint VPN (DMVPN)」コンフィギュレーション モジュール

標準/RFC

標準	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。	--
RFC 791	インターネット プロトコル
RFC 1191	パス *MTU* ディスカバリ
RFC 1323	『*TCP Extensions for High Performance*』
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*』

標準	タイトル
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
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 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/cisco/web/support/index.html

トンネル実装の機能情報

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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

表 7: トンネル実装の機能情報

機能名	リリース	機能情報
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。
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	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により、任意の VPN VRF テーブルに属するようにトンネルの送信元と宛先を設定できます。 次のコマンドが導入または変更されました。**tunnel vrf**

機能名	リリース	機能情報
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でも送信できます。 この機能により、コマンド **keepalive**（**トンネル インターフェイス**）が導入されました。
IPv6 IP トンネルを介する IP	Cisco IOS XE Release 2.4	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。
IP トンネル：SSO	Cisco IOS XE Release 3.6	ハイアベイラビリティのサポートが、IP トンネルに追加されました。 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。

機能名	リリース	機能情報
Tunnel ToS（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カプセル化 IP 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、高速スイッチング、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。 この機能により、**show interfaces tunnel**、**tunnel tos**、**tunnel** および **ttl** の各コマンドが導入または変更されました。

第 7 章

トンネルのルート選択

トンネルのルート選択機能により、ルーティング テーブルのサブセットを使用してトンネル転送をルーティングできます。 トンネルの宛先へのコストが等しいルートが複数ある場合、通常のトンネル転送動作は、ランダムに選択された使用可能なルートのいずれかを使用することになります。 トンネルのルート選択機能により、トンネル転送の発信インターフェイスを明示的に設定できます。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 最新の注意事項と機能情報については、プラットフォームおよびソフトウェア リリースのバグ検索ツールとリリース ノートを参照してください。 このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表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シスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

トンネルのルート選択の前提条件

トンネル インターフェイスが選択されていることが前提です。

トンネルのルート選択の制約事項

この機能は、次のトンネル モードのみでサポートされます。

- 総称ルーティング カプセル化（GRE）
- GRE マルチポイント
- IP in IP
- モバイル ユーザ データグラム プロトコル（UDP）

この機能は、トンネル転送が GRE マルチポイント トンネルの場合はトンネルではサポートされません。

サポートされている構成

```
interface tunnel 0
  tunnel mode gre multipoint
  tunnel route-via tunnel 1
interface tunnel 1
   tunnel mode gre ip
```

サポートされない設定

```
interface tunnel 0
  tunnel mode gre multipoint
  tunnel route-via tunnel 1
interface tunnel 1
  tunnel mode gre multipoint
```

トンネルのルート選択に関する情報

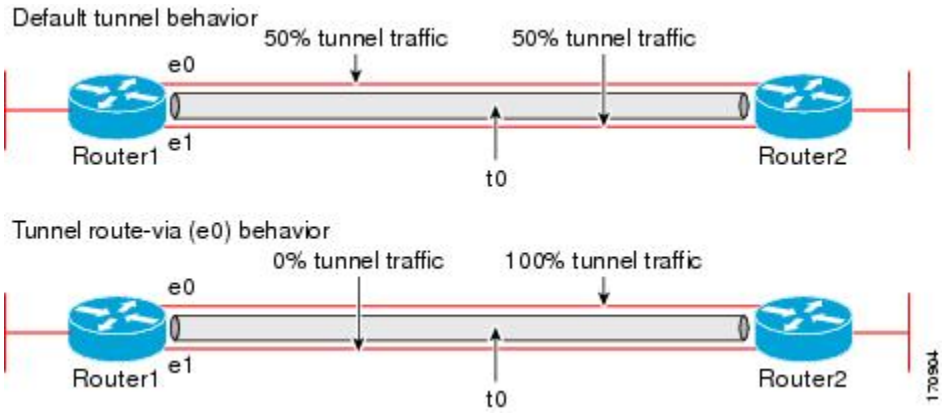
トンネル転送動作

トンネルのルート選択機能により、トンネル転送の発信インターフェイスを指定してルーティング テーブルのサブセットを使用し、トンネル転送をルーティングすることができます。

トンネルのルート選択機能は、トンネル転送に関するポリシーベースルーティングの実装と同じではありません。 トンネルのルート選択機能では、ルート テーブルのサブセットを 1 つだけ使用してトラフィックを転送できますが、ルーティング ループをネットワークに導入することはできません。

以下の図では、トンネルのルート選択動作とデフォルトのトンネル動作との比較を示します。

図 3：トンネルのルート選択トラフィック



トンネルのルート選択の設定方法

トンネルのルート選択の設定

次の手順を実行して、トンネル転送の発信インターフェイスを指定し、ルーティング テーブルのサブセットを使用してトンネル転送のルーティングを行います。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface tunnel** *interface-number*
4. **tunnel route-via** *interface-type interface-number* {**mandatory** | **preferred**}
5. **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。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
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** *interface-number* 例： `Router(config)# interface tunnel 0`	トンネルインターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**tunnel route-via** *interface-type interface-number* {**mandatory** | **preferred**} 例： `Router(config-if)# tunnel route-via ethernet0 mandatory`	トンネル転送が使用する発信インターフェイスを指定します。
ステップ 5	**end** 例： `Router(config-if)# end`	特権 EXEC モードに戻ります。

トラブルシューティングのヒント

設定のトラブルシューティングを行うには、特権 EXEC モードで **debug tunnel route-via** コマンドを使用します。 **tunnel route-via** コマンドでルーティング テーブルのサブセットを使用してトンネル転送の明示的ルーティングを行った後に実行した、**debug tunnel route-via** コマンドからの出力例を次に示します。

```
Router# debug tunnel route-via
Tunnel route-via debugging is on
Router#
*May 23 08:40:53.707: TUN-VIA: Tunnel0 candidate route-via Ethernet0/0, next hop 10.73.2.1
*May 23 08:40:53.707: TUN-VIA: Tunnel0 route-via action is forward
*May 23 08:41:03.719: TUN-VIA: Tunnel0 candidate route-via Ethernet0/0, next hop 10.73.2.1
*May 23 08:41:03.719: TUN-VIA: Tunnel0 route-via action is forward
Router# undebug tunnel route-via
Tunnel route-via debugging is off
```

次の作業

トンネルのルート選択の設定を確認できます。 設定を確認するには、特権 EXEC モードで **show interfaces tunnel** コマンドを使用します。 トンネル転送の発信インターフェイスを指定すること

で、ルーティング テーブルのサブセットを使用してトンネル転送をルーティングする例を次に示します。

```
Router# show running-config interface tunnel 0
Building configuration...
Current configuration : 147 bytes
!
interface Tunnel0
 ip unnumbered Loopback0
 tunnel source Loopback0
 tunnel destination 10.73.0.102
 tunnel route-via Ethernet0 preferred
end
Router# show interfaces tunnel 0 | include route-via
Tunnel route-via feature is on [Ethernet0, preferred]
```

トンネルのルート選択の設定例

トンネルのルート選択の設定例

イーサネットインターフェイス0を優先的な発信転送インターフェイスとして使用するようTunnel 0を設定する例を次に示します。イーサネット インターフェイス 0 からトンネルの宛先までのルートが存在する場合は、トンネル 0 インターフェイスを使用してルータを終了するトラフィックは、イーサネット インターフェイス 0 から送信されます。イーサネット インターフェイス 0 からのルートが存在しない場合は、トラフィックは、トンネルのルート選択機能が設定されていないかのように転送されます。

tunnel route-via *interface-type interface-number* **mandatory** コマンドが設定されており、そのインターフェイスを使用しているトンネルの宛先へのルートが存在しない場合、ポイントツーポイント トンネル インターフェイスはダウン状態になります。

```
Router> enable
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface tunnel 0
Router(config-if)# tunnel route-via ethernet0 preferred
Router(config-if)# end
Router# show running-config interface tunnel 0
Building configuration...
Current configuration : 147 bytes
!
interface Tunnel0
 ip unnumbered Loopback0
 tunnel source Loopback0
 tunnel destination 10.73.0.102
 tunnel route-via Ethernet0 preferred
end
```

その他の関連資料

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Command List, All Releases』
インターフェイス コマンド：**define interface-range**、**interface range**、および **interface vlan**	『Cisco IOS Interface and Hardware Component Command Reference』
インターフェイス コマンド：**show running-config**	『Cisco IOS Configuration Fundamentals Command Reference』

シスコのテクニカル サポート

説明	リンク
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 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/cisco/web/support/index.html

トンネルのルート選択の機能情報

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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

表 *8*：トンネルのルート選択の機能情報

機能名	リリース	機能情報
トンネルのルート選択	12.4(11)T 15.0(1)M Cisco IOS Release 3.9S	トンネルのルート選択機能により、ルーティング テーブルのサブセットを使用してトンネル転送をルーティングできます。トンネルの宛先へのコストが等しいルートが複数ある場合、通常のトンネル転送動作は、ランダムに選択された使用可能なルートのいずれかを使用することになります。トンネルのルート選択機能により、トンネル転送の発信インターフェイスを明示的に設定できます。 この機能により、**debug tunnel route-via**、**tunnel route-via**、**show interfaces tunnel** の各コマンドが導入または変更されました。

第 8 章

MPLS VPN over mGRE

MPLS VPN over mGRE 機能では、IP 専用ネットワークによって接続されている各ネットワーク間でマルチプロトコル ラベル スイッチング（MPLS）接続を可能にすることによって、通信事業者がMPLSをサポートしていなければならないという要件を克服しています。 これにより、MPLS ラベル スイッチド パス（LSP）が、総称ルーティング カプセル化（GRE）トンネルを使用して、ルーティング エリア、自律システム、およびインターネット サービス プロバイダー（ISP）を横断することが可能になります。 マルチポイント GRE（mGRE）を介して MPLS VPN を設定すると、標準ベースの IP コアを使用して、レイヤ 3（L3）プロバイダー エッジ（PE）ベースのバーチャル プライベート ネットワーク（VPN）サービスを導入できます。 これにより、オーバーレイ方式を使用しないで VPN サービスをプロビジョニングできます。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 最新の注意事項と機能情報については、プラットフォームおよびソフトウェア リリースのバグ検索ツールとリリース ノートを参照してください。 このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表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シスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

MPLS VPN over mGRE の前提条件

mGRE トンネルを使用して MPLS VPN を設定する前に、MPLS VPN が設定されていて、正しく動作していることを確認してください。 MPLS VPN の設定に関する詳細については、「Configuring MPLS Layer 3 VPNs」モジュールを参照してください。

MPLS VPN over mGRE の制約事項

- トンネルリングされたタグ トラフィックは、MPLS VPN over mGRE がサポートされているラインカードを介してルータに入る必要があります。
- 各 PE ルータでサポートされるトンネル コンフィギュレーションは 1 つだけです。
- MPLS VPN over mGRE では、VPN 間におけるマルチキャスト トラフィックの転送はサポートされていません。
- GRE トンネルの宛先アドレスおよび送信元アドレスが mGRE と同じである場合、トンネルによってルートキャッシュが切り替えられます。
- フラグメンテーションが必要なパケットによって、ルートキャッシュが切り替えられます。
- L3VPN プロファイルをいったん削除して後で戻す場合、**clear ip bgp soft** コマンドを使用して、ボーダー ゲートウェイ プロトコル（BGP）をクリアする必要があります。
- mGRE トンネルが作成されると、ダミー トンネルも作成されます。
- BGP コンフィギュレーションのアップデート元で使用されるループバックまたは IP アドレスは、L3VPN プロファイルの送信元と同じである必要があります。
- mGRE は、ステートフル スイッチオーバー（SSO）には対応していません。 ただし、mGRE と SSO の両方が共存します。
- mGRE とマルチキャスト配信ツリー（MDT）トンネルを同一のループバック アドレスを使用して設定できません。

MPLS VPN over mGRE 機能の制限事項は、次のとおりです。

- ハードウェア内で、すべての GRE オプションがサポートされているわけではありません（GRE 拡張ヘッダーや GRE キーなど）。
- トンネル上では、複数の同一 VLAN（インターネット制御メッセージ プロトコル（ICMP）リダイレクト）のチェックはサポートされていません。
- トンネル上では、ユニキャスト リバース パス転送（uRPF）や BGP ポリシー アカウントなどの機能はサポートされていません。

MPLS VPN over mGRE について

mGRE トンネルを設定して、IP バックボーンをオーバーレイするマルチポイント トンネル ネットワークを作成できます。 このオーバーレイによって、VPN トラフィックを転送するために各 PE ルータ同士が接続されます。

さらに、MPLS VPN を mGRE を介して設定すると、標準ベースの IP コアを使用して、L3 PE ベースの VPN サービスを導入できます。 これにより、オーバーレイ方式を使用しないで VPN サービスをプロビジョニングできます。 MPLS VPN over mGRE が設定されると、システムでは、PE 間の VPN ラベル IPv4 および IPv6 パケットのカプセル化に IPv4 ベースの mGRE トンネルが使用されます。 MPLS VPN over mGRE トンネルを配置するには、VRF インスタンスを作成し、L3 VPN カプセル化をイネーブルおよび設定し、ルートマップをアプリケーションテンプレートにリンクし、アップデートがルート マップを介してフィルタ処理されるように BGP VPNv4 と VPNv6 の交換を設定します。

MPLS VPN over mGRE

GRE とは、ポイントツーポイント トンネリング プロトコルの 1 つであり、2 つのピアがトンネルのエンドポイントとなります。 GRE は、ネットワーク層のパケットを IP トンネリング パケット内にカプセル化するように設計されています。 mGRE は、GRE と類似したプロトコルですが、トンネルの片方は単一のエンドポイントで、それがトンネルのもう片方にある複数のエンドポイントに接続されています。 mGRE トンネルによって、同じ VPN に接続された各支社間が共通のリンクを使用できるようになります。 mGRE は、ポイントツーマルチポイント モデルなので、各 MPLS VPN PE デバイスを相互接続するうえでフル メッシュ構造の GRE トンネルは不要です。

MPLS は、広く採用されている VPN インターネット アーキテクチャです。 MPLS では、ネットワーク内のすべてのコア ルータで MPLS がサポートされていることが必要です。 この機能は、サービス プロバイダーがバックボーン キャリアを使用して接続を提供しているネットワークで有用です。

MPLS VPN over mGRE 機能では、IP 専用ネットワークによって接続されている各ネットワーク間で MPLS 接続を可能にすることによって、通信事業者が MPLS をサポートしていなければならないという要件を克服しています。 これにより、MPLS LSP が、GRE トンネルを使用して、ルーティング エリア、自律システム、および ISP を横断することが可能になります。

MPLS VPN を mGRE を介して設定すると、標準ベースの IP コアを使用して、L3 PE ベースの VPN サービスを導入できます。 これにより、LSP やラベル配布プロトコル（LDP）を使用しないで VPN サービスをプロビジョニングできます。 システムでは、PE 間の VPN ラベル IPv4 および IPv6 パケットのカプセル化に IPv4 ベースの mGRE トンネルが使用されます。

また、MPLS VPN over mGRE 機能によって、既存の MPLS VPN LSP カプセル化テクノロジーを、MPLS VPN over mGRE と同時に導入し、特定のトラフィックをルーティングするために使用されるカプセル化方式が自動的に決定されるようにすることも可能です。 入力 PE ルータによって、パケットがリモート PE ルータに送信されるときに使用されるカプセル化テクノロジーが決定されます。

ここでは、MPLS VPN over mGRE 機能に関する次の項目について説明します。

ルート マップ

デフォルトでは、VPN トラフィックの送信に LSP が使用されます。 MPLS VPN over mGRE 機能では、ユーザ定義のルート マップが使用されて、mGRE トンネルを介して到達可能な VPN プレフィックスと、LSP を使用して到達可能な VPN プレフィックスが決定されます。 ルート マップは、VPNv4 および VPNv6 アドレス ファミリのアドバタイズメントに適用されます。 ルート マップでは、VPN トラフィックのカプセル化方式の決定に Next Hop Tunnel Table が使用されます。

mGRE トンネルを介してトラフィックをルーティングするため、mGRE トンネル内でトラフィックをカプセル化することによって到達されるすべてのネクスト ホップを示す代替アドレス空間が自動的に作成されます。 mGRE トンネルを使用する特定のルートを設定するには、ユーザが、そのルートのエントリをルートマップに追加します。 その新しいエントリによって、代替アドレス空間に対して、そのルートのネットワーク層到着可能性情報（NLRI）が再マッピングされます。 あるルートのルート マップ内に再マッピング エントリが存在しない場合、そのルート上のトラフィックは LSP を介して転送されます。

ユーザが MPLS VPN over mGRE を設定すると、代替アドレス空間が自動的にプロビジョニングされ、通常の場合、トンネルカプセル化 Virtual Routing and Forwarding（VRF）インスタンス内に保持されます。 アドレス空間を介して到達可能なトラフィックが確実にすべて mGRE トンネル内でカプセル化されるように、トンネル外への単一のデフォルト ルートが自動的にインストールされます。 また、ルート マップ上にデフォルト トンネルも自動的に作成されます。 ユーザは、このデフォルト ルート マップを、適切な BGP アップデートに対応付けることが可能です。

トンネル エンドポイントの検出およびフォワーディング

MPLS VPN over mGRE 機能が正しく機能するように、システム内のリモート PE が検出でき、それらのリモート PC のトンネル フォワーディング情報が作成できるようにする必要があります。 また、リモート PE が無効となったことが検出され、その PE のトンネル フォワーディング情報が削除されるようにする必要もあります。

入力 PE によって BGP を介して VPN アドバタイズメントが受信される場合、その入力 PE によってルート ターゲット属性（VRF に入力されます）および、アドバタイズメントからの MPLS VPN ラベルが使用され、その結果、プレフィックスと適切なお客様が関連付けられます。 入力されたルートのネクスト ホップが、アドバタイズメントの NLRI に設定されます。

アドバタイズされたプレフィックスには、システム内のリモート PE に関する情報が（NLRI の形式で）格納され、PE では、この情報が使用されて、NLRI がアクティブまたは非アクティブになったときシステムに通知されます。 システムでは、この通知が使用されて、PE フォワーディング情報がアップデートされます。

システムによって、新しいリモート PE の通知が受信されると、Tunnel Endpoint Database にその情報が追加され、これを契機として、トンネル インターフェイスに関連付けられた隣接が作成されます。 この隣接の説明として、カプセル化に関する情報、およびカプセル化されたパケットを新しいリモート PE に送信するために実行される必要のあるその他の処理に関する情報が記述されています。

この隣接情報は、トンネル カプセル化 VRF に入力されます。 ユーザが（ルート マップを使用して）VRF 内のルートに VPN NLRI を再マッピングすると、その NLRI が隣接に対してリンクされ、その結果、VPN がトンネルにリンクされます。

トンネルの非カプセル化

MPLS VPN over mGRE 機能を使用するトンネル インターフェイスからのパケットを入力 PE が受信すると、その PE によってパケットが非カプセル化され、VPN ラベル タグ付きパケットが作成されて、MPLS Forwarding（MFI）コードにそのパケットが送信されます。

トンネルの送信元

MPLS VPN over mGRE 機能では、大量のエンドポイント（リモート PE）を持つシステムの設定に、mGRE トンネルとして設定された単一のトンネルが使用されます。 トンネルカプセル化パケットの送信元を特定するために、システムによってトンネル送信元情報が使用されます。

送信（入力）PE では、VPN パケットがトンネルに送信されるときのトンネル宛先は NLRI です。受信（出力）PE では、トンネル送信元は、mGRE トンネルでカプセル化されたパケットが受信されるアドレスです。 そのため、出力 PE では、パケットの宛先がローカル PE からの NLRI と一致している必要があります。

IPv6 VPN

アドバタイジング PE ルータのアドレスが IPv6 である場合、（PE 間のネットワークには関係なく）NLRI のアドレスも IPv6 である必要があります。 各 PE 間のネットワークが IPv4 ベースである場合、::FFFF:IPv4-PE-address という形式の IPv4 射影アドレスが使用されて、アドバタイジング PE の IPv6 アドレスが作成されます。 受信 PE によって、VPN タグ IPv6 プレフィックスのネクスト ホップが、IPv6 NLRI に埋め込まれた IPv4 アドレスに設定されます。 これにより、PE によって、VPNv4 トラフィックをマッピングするのと同じ方法で、VPNv6 トラフィックを LSP または mGRE トンネルにリンクすることが可能になります。

PE によって VPNv6 アップデートが受信されると、そのアップデートが IPv6 ルート マップに適用されます。 MPLS VPN over mGRE 機能では、Tunnel_Encap VRF におけるネクスト ホップ情報の設定に IPv6 ルート マップが使用されます。

MPLS VPN over mGRE の設定方法

L3VPN カプセル化プロファイルの設定

ここでは、L3VPN カプセル化プロファイルを設定する方法を説明します。

（注） この設定では、IPv6、MPLS、IP、およびレイヤ2トンネルプロトコルバージョン3（L2TPv3）のような転送プロトコルも使用できます。

手順の概要

1. **enable**
2. **configure terminal**
3. **l3vpn encapsulation ip** *profile-name*
4. **transport ipv4 [source** *interface-type interface-number* **]**
5. **protocol gre [key** *gre-key* **]**
6. **end**
7. **show l3vpn encapsulation ip** *profile-name*

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。 • 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： `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	**l3vpn encapsulation ip** *profile-name* 例： `Router(config)# l3vpn encapsulation ip tunnel encap`	L3 VPN カプセル化コンフィギュレーション モードを開始し、トンネルを作成します。
ステップ4	**transport ipv4 [source** *interface-type interface-number* **]** 例： `Router(config-l3vpn-encap-ip)# transport ipv4 source loopback 0`	（任意）IPv4 送信元モードを指定して、送信元インターフェイスを定義します。 • **transport ipv4 source** *interface-type interface-number* コマンドを使用する場合、指定した送信元アドレスが、PE によってアドバタイズされた BGP アップデートにおけるネクストホップとして使用されていることを確認します。

	コマンドまたはアクション	目的
		• このコマンドを使用しない場合、**bgp update source** または **bgp next-hop** コマンドが、トンネル送信元として自動的に使用されます。
ステップ 5	**protocol gre [key** *gre-key* **]** 例： `Router(config-l3vpn-encap-ip)# protocol gre key 1234`	GRE をトンネル モードとして指定し、GRE キーを設定します。
ステップ 6	**end** 例： `Router(config-l3vpn-encap-ip)# end`	L3 VPN カプセル化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。
ステップ 7	**show l3vpn encapsulation ip** *profile-name* 例： `Router# show l3vpn encapsulation ip tunnel encap`	（任意）プロファイルの状態および基本となるトンネル インターフェイスを表示します。

BGP およびルート マップの設定

BGP およびルート マップを設定するには、次の作業を実行します。 次の手順では、ルート マップをアプリケーションテンプレートにリンクし、アップデートがルートマップを介してフィルタ処理されるように BGP VPNv4 と VPNv6 の交換を設定することも可能です。

手順の概要

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **bgp log-neighbor-changes**
5. **neighbor** *ip-address* **remote-as** *as-number*
6. **neighbor** *ip-address* **update-source** *interface name*
7. **address-family ipv4**
8. **no synchronization**
9. **redistribute connected**
10. **neighbor** *ip-address* **activate**
11. **no auto-summary**
12. **exit**
13. **address-family vpnv4**
14. **neighbor** *ip-address* **activate**
15. **neighbor** *ip-address* **send-community both**
16. **neighbor** *ip-address* **route-map** *map-name* **in**
17. **exit**
18. **address-family vpnv6**
19. **neighbor** *ip-address* **activate**
20. **neighbor** *ip-address* **send-community both**
21. **neighbor** *ip-address* **route-map** *map-name* **in**
22. **exit**
23. **route-map** *map-tag* **permit** *position*
24. **set ip next-hop encapsulate l3vpn** *profile-name*
25. **set ipv6 next-hop encapsulate l3vpn** *profile-name*
26. **exit**
27. **exit**

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。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
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	**router bgp** *as-number* 例： `Router(config)# router bgp 100`	他の BGP ルータに接続されたルータを特定する自律システムの番号を指定し、転送されるルーティング情報にタグ付けし、ルータ コンフィギュレーション モードを開始します。
ステップ 4	**bgp log-neighbor-changes** 例： `Router(config-router)# bgp log-neighbor-changes`	BGP ネイバー リセットのロギングをイネーブルにします。
ステップ 5	**neighbor** *ip-address* **remote-as** *as-number* 例： `Router(config-router)# neighbor 209.165.200.225 remote-as 100`	BGP ネイバー テーブルまたはマルチプロトコル BGP ネイバー テーブルにエントリを追加します。
ステップ 6	**neighbor** *ip-address* **update-source** *interface name* 例： `Router(config-router)# neighbor 209.165.200.225 update-source loopback 0`	BGP セッションが、TCP 接続の動作インターフェイスを使用できるようにします。
ステップ 7	**address-family ipv4** 例： `Router(config-router)# address-family ipv4`	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、IPv4 アドレス プレフィックスを使用するルーティング セッションを設定します。
ステップ 8	**no synchronization** 例： `Router(config-router-af)# no synchronization`	IGP を待たずにネットワーク ルートをアドバタイズするよう、Cisco ソフトウェアをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 9	**redistribute connected** 例： `Router(config-router-af)# redistribute connected`	1 つのルーティング ドメインから別のルーティング ドメインにルートを再配布し、送信元プロトコルによって認識されたルート、および、送信元プロトコルが実行されているインターフェイスを介して接続されているプレフィックスを、ターゲット プロトコルで再配布できるようにします。
ステップ 10	**neighbor** *ip-address* **activate** 例： `Router(config-router-af)# neighbor 209.165.200.225 activate`	BGP ネイバーとの情報交換をイネーブルにします。
ステップ 11	**no auto-summary** 例： `Router(config-router-af)# no auto-summary`	自動サマライズをディセーブルにし、サブプレフィックス ルーティング情報をクラスフル ネットワーク境界間で送信します。
ステップ 12	**exit** 例： `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了します。
ステップ 13	**address-family vpnv4** 例： `Router(config-router)# address-family vpnv4`	アドレス ファミリ コンフィギュレーション モードを開始して、標準 VPNv4 アドレス プレフィックスを使用する、BGP などのルーティング セッションを設定します。
ステップ 14	**neighbor** *ip-address* **activate** 例： `Router(config-router-af)# neighbor 209.165.200.225 activate`	BGP ネイバーとの情報交換をイネーブルにします。
ステップ 15	**neighbor** *ip-address* **send-community both** 例： `Router(config-router-af)# neighbor 209.165.200.225 send-community both`	標準コミュニティと拡張コミュニティの両方の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 16	**neighbor** *ip-address* **route-map** *map-name* **in**	名前付きルート マップを受信ルートに適用します。

	コマンドまたはアクション	目的
	例： `Router(config-router-af)# neighbor 209.165.200.225 route-map SELECT_UPDATE_FOR_L3VPN in`	
ステップ 17	**exit** 例： `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了します。
ステップ 18	**address-family vpnv6** 例： `Router(config-router)# address-family vpnv6`	アドレス ファミリ コンフィギュレーション モードを開始して、VPNv6 アドレス プレフィックスを使用する、BGP などのルーティング セッションを設定します。
ステップ 19	**neighbor** *ip-address* **activate** 例： `Router(config-router-af)# neighbor 209.165.200.252 activate`	BGP ネイバーとの情報交換をイネーブルにします。
ステップ 20	**neighbor** *ip-address* **send-community both** 例： `Router(config-router-af)# neighbor 209.165.200.252 send-community both`	標準コミュニティと拡張コミュニティの両方の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 21	**neighbor** *ip-address* **route-map** *map-name* **in** 例： `Router(config-router-af)# neighbor 209.165.200.252 route-map SELECT_UPDATE_FOR_L3VPN in`	名前付きルート マップを受信ルートに適用します。
ステップ 22	**exit** 例： `Router(config-router-af)# exit`	アドレス ファミリ コンフィギュレーション モードを終了します。

	コマンドまたはアクション	目的
ステップ 23	**route-map** *map-tag* **permit** *position* 例： `Router(config-router)# route-map SELECT_UPDATE_FOR_L3VPN permit 10`	ルート マップ コンフィギュレーション モードを開始し、1 つのルーティング プロトコルから別のルーティング プロトコルへルートを再配布する条件を定義します。 • **redistribute** ルータ コンフィギュレーション コマンドによって、指定されたマップ タグが使用され、このルートマップが参照されます。 複数のルートマップで同じマップ タグ名を共有できます。 • このルート マップの一致基準が満たされている場合は、set アクションの制御に従ってルートが再配布されます。 • 一致基準が満たされないと、同じマップ タグを持つ次のルート マップが検査されます。 あるルートが、同じ名前を共有するルート マップ セットの一致基準のいずれをも満たさない場合、そのセットによる再配布は行われません。 • *position* 引数は、同じ名前で設定済みのルート マップのリストに新しいルート マップが入る位置を示します。
ステップ 24	**set ip next-hop encapsulate l3vpn** *profile-name* 例： `Router(config-route-map)# set ip next-hop encapsulate l3vpn my profile`	ルート マップの match 句を渡す出力 IPv4 パケットは、トンネルのカプセル化のため、VRF に送信されます。
ステップ 25	**set ipv6 next-hop encapsulate l3vpn** *profile-name* 例： `Router(config-route-map)# set ip next-hop encapsulate l3vpn tunnel encap`	ルート マップの match 句を渡す出力 IPv6 パケットは、トンネルのカプセル化のため、VRF に送信されます。
ステップ 26	**exit** 例： `Router(config-route-map)# exit`	ルート マップ 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 モードを開始します。

	コマンドまたはアクション	目的
ステップ 27	**exit** 例： `Router(config)# exit`	グローバル コンフィギュレーション モードを終了します。

MPLS VPN over mGRE の設定例

MPLS VPN over mGRE 設定の確認例

設定が正しく動作していることを確認する例を次に示します。

シスコ エクスプレス フォワーディング（CEF）スイッチング

CEF スイッチングが想定どおりに動作しているかどうかを確認します。

```
Router# show ip cef vrf Customer_A tunnel 0

209.165.200.250
/24
    nexthop 209.165.200.251 Tunnel0 label 16
```

エンドポイントの作成

トンネルのエンドポイントが作成されているかどうかを確認します。

```
Router# show tunnel endpoints tunnel 0

 Tunnel0 running in multi-GRE/IP mode
 Endpoint transport 209.165.200.251 Refcount 3 Base 0x2AE93F0 Create Time 00:00:42
   overlay 209.165.200.254 Refcount 2 Parent 0x2AE93F0 Create Time 00:00:42
```

隣接

対応する隣接が作成されているかどうかを確認します。

```
Router# show adjacency tunnel 0
   Protocol Interface                 Address
  IP       Tunnel0                   209.165.200.251(4)
  TAG      Tunnel0                   209.165.200.251(3)
```

プロファイルの状態

show l3vpn encapsulation *profile-name* コマンドを使用して、アプリケーションの基本的な状態に関する情報を取得できます。 このコマンドの出力には、基本となるトンネルの詳細が表示されます。

```
Router# show l3vpn encapsulation ip tunnel encap
 Profile: tunnel encap
 transport ipv4 source Auto: Loopback0
 protocol gre
   Tunnel Tunnel0 Created [OK]
   Tunnel Linestate [OK]
   Tunnel Transport Source (Auto) Loopback0 [OK]
```

MPLS VPN over mGRE のシーケンス設定例

次に、MPLS VPN over mGRE の設定シーケンスの例を示します。

```
vrf definition Customer A
 rd 100:110
 route-target export 100:1000
 route-target import 100:1000
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
!
ip cef
!
ipv6 unicast-routing
ipv6 cef
!
!
l3vpn encapsulation ip sample profile name
 transport source loopback 0
 protocol gre key 1234
!
!
 interface Loopback0
  ip address 209.165.200.252 255.255.255.224
  ip router isis
!
interface Serial2/0
 vrf forwarding Customer A
 ip address 209.165.200.253 255.255.255.224
 ipv6 address 3FFE:1001::/64 eui-64
 no fair-queue
 serial restart-delay 0
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 209.165.200.254 remote-as 100
 neighbor 209.165.200.254 update-source Loopback0
 !
 address-family ipv4
  no synchronization
  redistribute connected
  neighbor 209.165.200.254 activate
  no auto-summary
 exit-address-family
 !
 address-family vpnv4
  neighbor 209.165.200.254 activate
  neighbor 209.165.200.254 send-community both
  neighbor 209.165.200.254 route-map SELECT_UPDATE_FOR_L3VPN in
```

```
 exit-address-family
 !
 address-family vpnv6
  neighbor 209.165.200.254 activate
  neighbor 209.165.200.254 send-community both
  neighbor 209.165.200.254 route-map SELECT_UPDATE_FOR_L3VPN in
 exit-address-family
 !
 address-family ipv4 vrf Customer A
  no synchronization
  redistribute connected
 exit-address-family
 !
 address-family ipv6 vrf Customer A
  redistribute connected
  no synchronization
 exit-address-family
!
!
route-map SELECT_UPDATE_FOR_L3VPN permit 10
set ip next-hop encapsulate sample profile name
set ipv6 next-hop encapsulate sample profile name
```

その他の関連資料

関連資料

関連項目	マニュアル タイトル
MPLS レイヤ 3 VPNs の設定	『*Cisco IOS XE Multiprotocol Label Switching Configuration Guide*』
シスコ エクスプレス フォワーディング	『*Cisco IOS XE IP Switching Configuration Guide*』
総称ルーティング カプセル化	『*Cisco IOS XE Interface and Hardware Component Configuration Guide*』

標準

標準	タイトル
なし	--

MIB

MIB	MIB のリンク
IETF-PPVPN-MPLS-VPN-MIB	選択したプラットフォーム、Cisco ソフトウェア リリース、およびフィーチャ セットの MIB を検索してダウンロードする場合は、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 2547	『*BGP/MPLS VPNs*』
RFC 2784	『*Generic Routing Encapsulation (GRE)*』
RFC 2890	『*Key Sequence Number Extensions to GRE*』
RFC 4023	『Encapsulating MPLS in IP or Generic Routing Encapsulation』
RFC 4364	『*BGP/MPLS IP Virtual Private Networks (VPNs)*』

シスコのテクニカル サポート

説明	リンク
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 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/cisco/web/support/index.html

MPLS VPN over mGRE 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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

表 *9*：*MPLS VPN over mGRE* の機能情報

機能名	リリース	機能情報
MPLS VPN over mGRE	Cisco IOS XE Release 3.1S	この機能では、mGRE を介した MPLS レイヤ 3 VPN トラフィックの搬送がサポートされています。 この機能では、コマンド l3vpn encapsulation ip、protocol gre、show l3vpn encapsulation ip、transport ipv4、set ip next-hop、set ipv6 next-hop が導入または変更されています。

インターフェイスおよびハードウェア コンポーネント コンフィギュレーション ガイド、**Cisco IOS XE Release 3S**（**ASR 1000**）

第 9 章

IP トンネル MIB

このモジュールでは、インターフェイスおよびハードウェア コンポーネントで使用する MIB について説明します。 IP トンネル MIB 機能は、RFC 4087『IP Tunnel MIB』に示されているすべての IPv4 および IPv6 関連のトンネルを管理する汎用 MIB を提供します。 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。 シスコは、IPv4 および IPv6 環境に対してインターネット技術特別調査委員会（IETF）によって指定されたさまざまなトンネリング メカニズムを実装しています。 トンネルの管理には、さまざまな MIB を使用できます。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 最新の注意事項と機能情報については、プラットフォームおよびソフトウェア リリースのバグ検索ツールとリリース ノートを参照してください。 このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表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シスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

IP トンネル MIB の前提条件

IP トンネル MIB 機能を使用するルータに簡易ネットワーク管理プロトコル（SNMP）を設定します。 詳細については、SNMP を使用するためのルータの設定, （103 ページ）を参照してください。 SNMP サーバの設定の詳細については、『Cisco IOS Network Management Configuration Guide』の「Configuring SNMP Support」の章を参照してください。

IP トンネル MIB の制約事項

IP トンネル MIB 機能は、**interface tunnel** コマンドを使用して作成できるトンネルだけをサポートします。 IP トンネル MIB 機能は、レイヤ 2 トンネル プロトコル（L2TP）、Point-to-Point Tunneling Protocol（PPTP）、マルチプロトコル ラベル スイッチング（MPLS）のトンネルをサポートしません。

IP トンネル MIB の概要

IP トンネル MIB の利点

ネットワークの品質向上

IP トンネル装置が改善されることで、ネットワークの品質が向上し、より高いサービスを提供できます。 ネットワークの品質が向上すると、サービス プロバイダーは信頼性の高いサービスを提供できます。

信頼性の向上

IP トンネル MIB により、ネットワーク管理システムのユーザはインベントリを設定し、IP トンネルのアクティビティに関する通知を受信できます。

IP トンネル MIB は、RFC 3291 で定義されている IPv4 および IPv6 ネットワーク層をサポートし、Cisco IOS ソフトウェアに実装されている IP トンネルの管理に使用されます。

IP トンネル MIB は、すべてのトンネル タイプとともにトンネル作成および廃棄機能をサポートします。

シスコ デバイス以外のデバイスとの相互運用性

IP トンネル MIB は、サードパーティ ベンダー製を含む主要ネットワーク管理システムと相互運用ができます。

IP トンネル MIB でサポートされる MIB オブジェクト

IP トンネル MIB 機能によってサポートされる MIB オブジェクトは次のとおりです。 MIB オブジェクトの使用の詳細については、RFC 4087『IP Tunnel MIB』を参照してください。

表 *10* : *IP* トンネル *MIB* でサポートされるオブジェクト

MIB オブジェクト	説明
tunnelIfEntry	特定の設定済みトンネルに関する情報が含まれます。 このオブジェクトの値を設定するには、**interface tunnel** コマンドを使用します。
tunnelIfEncapsMethod	トンネルで使用されるカプセル化方式。 このオブジェクトの値を設定するには、**tunnel mode** コマンドを使用します。
tunnelIfHopLimit	外部 IP ヘッダーで使用する IPv4 存続可能時間（TTL）または IPv6 ホップ制限を定義します。 このオブジェクトの値を設定するには、**tunnel ttl** コマンドを使用します。
tunnelIfSecurity	外部 IP ヘッダーを保護するためにトンネルで使用されます。 値 ipsec は、認証、暗号化、またはその両方にトンネル エンドポイント間で IPsec が使用されることを示します。
tunnelIfTOS	外部 IP ヘッダーの IPv4 タイプ オブ サービス（ToS）または IPv6 トラフィック クラスの上位 6 ビット（ディファレンシエーテッド サービス コードポイント）を設定するためにトンネルで使用されます。 このオブジェクトの値を設定するには、**tunnel tos** コマンドを使用します。
tunnelIfFlowLabel	IPv6 フロー ラベル値を設定するために使用されます。 このオブジェクトは IPv6 上のトンネルでサポートされます。 このオブジェクトのデフォルト値は 0 です。
tunnelIfAddressType	対応する tunelIfLocalInetAddress オブジェクトおよび tunnelIfRemoteInetAddress オブジェクトのアドレスのタイプを示します。 このオブジェクトは、コマンドライン インターフェイス（CLI）を使用して個別に設定できません。

MIB オブジェクト	説明
tunnelIfLocalInetAddress	トンネルのローカル エンドポイントのアドレス（外部 IP ヘッダーで使用される送信元アドレス）。 アドレスが不明な場合、この値は IPv4 では 0.0.0.0、IPv6 では :: です。 このオブジェクトのアドレス タイプは tunnelIfAddressType で指定します。 このオブジェクトの値を設定するには、**tunnel source** コマンドを使用します。
tunnelIfRemoteInetAddress	トンネルのリモート エンドポイントのアドレス（外部 IP ヘッダーで使用される宛先アドレス）。 アドレスが不明な場合、またはトンネルがポイントツーポイント リンクではない場合（たとえば、6-to-4 トンネル）、この値は IPv4 上のトンネルでは 0.0.0.0、IPv6 上のトンネルでは :: です。 このオブジェクトのアドレス タイプは tunnelIfAddressType で指定します。 このオブジェクトの値を設定するには、**tunnel destination** コマンドを使用します。
tunnelIfEncapsLimit	このノードでカプセル化されるパケットに対して許可される追加カプセル化の最大数を示します。 -1 は制限がないことを示します（パケット サイズの結果を除く）。
tunnelInetConfigEntry	特定の設定済みトンネルに関する情報が含まれます。 マルチポイント トンネル、および IPv4 では 0.0.0.0、IPv6 では :: のリモート inet アドレスを持つトンネルのエントリは 1 つだけです。 MIB を使用して作成するのは、Generic Routing Encapsulation（GRE）/IP トンネルおよび GRE/IPv6 トンネルだけです。
tunnelInetConfigIfIndex	トンネル インターフェイスに対応する ifIndex の値を示します。 0 はアクティブ状態では無効であり、インターフェイス インデックスがまだ割り当てられていないことを意味します。
tunnelInetConfigStatus	MIB テーブルのテーブル エントリの作成または削除に使用します。 このオブジェクトの値を設定するには、**interface tunnel** を使用します。
tunnelInetConfigStorageType	ストレージ タイプを示します。 不揮発性ストレージ値だけがサポートされます。

SNMP の設定方法および IP トンネル MIB の使用方法

SNMP を使用するためのルータの設定

（注） ここで説明する作業の中には、ルータに設定パラメータを設定し、ルータの MIB オブジェクトから値を読み取るために使用する SNMP CLI 構文の例が含まれているものがあります。 これらの SNMP CLI 構文の例は、パブリック ドメイン SNMP ツールを使用して Linux ワークステーションから取られています。 ご使用のワークステーションによっては SNMP CLI 構文が異なる場合があります。ネットワーク管理ワークステーションの正しい構文については、SNMP ツールに付属のマニュアルを参照してください。

IP トンネル MIB 機能を使用する前に、SNMP をサポートするためにルータを設定する必要があります。 ルータの SNMP をイネーブルにするには、この作業を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **snmp-server community** *string1* **ro**
4. **snmp-server community** *string2* **rw**
5. **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。 • 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： `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。

	コマンドまたはアクション	目的
ステップ 3	**snmp-server community** *string1* **ro** 例： `Router(config)# snmp-server community public ro`	SNMP へのアクセスを許可するコミュニティ アクセス ストリングを設定します。 • *string1* 引数は、1 ～ 32 文字の英数字で構成されるコミュニティ ストリングで、パスワードのように機能して SNMP プロトコルへのアクセスを許可します。 コミュニティ ストリングに空白は使用できません。 • **ro** キーワードは、読み取り専用アクセスを指定します。 このストリングを使用する SNMP 管理ステーションは MIB オブジェクトを取得できます。 （注） この例の SNMP コミュニティ読み取り専用（RO）ストリングは public です。 ご使用の設定では、この値にこれより複雑な構文を使用する必要があります。
ステップ 4	**snmp-server community** *string2* **rw** 例： `Router(config)# snmp-server community private rw`	SNMP へのアクセスを許可するコミュニティ アクセス ストリングを設定します。 • *string2* 引数は、1 ～ 32 文字の英数字で、パスワードのように機能して SNMP プロトコルへのアクセスを許可します。 コミュニティ ストリングに空白は使用できません。 • **rw** キーワードは、読み取りと書き込みアクセスを指定します。 このストリングを使用する SNMP 管理ステーションは、MIB オブジェクトを取得して修正できます。 （注） この例の SNMP コミュニティ読み取り/書き込み（RW）ストリングは private です。 ご使用の設定では、この値にこれより複雑な構文を使用する必要があります。
ステップ 5	**end** 例： `Router(config)# end`	現在の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

IP トンネル MIB を実装するには、トンネルを設定する必要があります。 トンネルの設定については、『Cisco IOS Interface and Hardware Component Configuration Guide』の「Implementing Tunnels」の章を参照してください。

SNMP を介した IP トンネル MIB の設定に関連する問題をデバッグまたはトラブルシューティングするには、debug snmp tunnel-mib コマンドを使用します。 このコマンドの詳細については、『Cisco IOS Interface and Hardware Component Command Reference』を参照してください。

その他の関連資料

関連資料

関連項目	マニュアル タイトル
SNMP コマンド、コマンド構文の詳細、コマンドリファレンス、コマンド履歴、デフォルト設定、使用に関する注意事項および例	『*Cisco IOS Network Management Command Reference*』
SNMP サポートの設定	『*Cisco IOS Network Management Configuration Guide*』
トンネルの実装	『*Cisco IOS Interface and Hardware Component Configuration Guide*』

標準

標準	タイトル
この機能でサポートされる新規の標準または変更された標準はありません。また、既存の標準のサポートは変更されていません。	--

MIB

MIB	MIB のリンク
『IP Tunnel MIB』	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

RFC

RFC	タイトル
RFC 4087	『IP Tunnel MIB』

シスコのテクニカル サポート

説明	リンク
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 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/cisco/web/support/index.html

トンネル MIB 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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

表 *11* : *IP* トンネル *MIB* の機能情報

機能名	リリース	機能情報
IP トンネル MIB	12.2(33)SRB 12.2(1st)SY 12.2(44)SG 12.2(33)SRD 15.0(1)M Cisco IOS XE 3.1.0SG Cisco IOS XE Release 3.9S	IP トンネル MIB は、RFC 4087『IP Tunnel MIB』に示されているすべての IPv4 および IPv6 関連のトンネルを管理する汎用 MIB です。

第 10 章

IF-MIB

このモジュールでは、インターフェイスおよびハードウェア コンポーネントで使用する MIB について説明します。 IF-MIB は、RFC 2863、インターフェイス グループ MIB および CISCO-IFEXTENSION-MIB で定義されているすべてのテーブルをサポートします。 この MIB によって、インターフェイス MIB オブジェクトにクエリーを送信する機能が提供され、返される情報は、SNMP コンテキストがマップされる仮想プライベート ネットワーク（VPN）ルーティング/転送（VRF）インスタンスに限定されます。 特定のホストに送信する必要のある通知を制限するコンテキストで、通知ホストを設定することもできます。

IF-MIB により、VRF 環境では、コンテキスト対応パケット情報がサポートされます。 クライアントから IF-MIB に保存されている情報に選択アクセスが可能になるよう、コンテキストが必要な VRF 環境が VPN に適用されます。 特定の VRF に属するクライアントでは、その VRF にのみ属している IF-MIB からのインターフェイスに関する情報にアクセスできます。 クライアントで、特定のコンテキストに関連付けられているインターフェイスからの情報を取得しようとするときには、そのクライアントでは、コンテキストにのみ属している情報にアクセスでき、権限のない情報は参照できません。

このマニュアルでは、サブインターフェイスでのインターフェイス グループ MIB の拡張機能、および、Cisco IOS ソフトウェアの IF-MIB のシスコでの実装における RFC 2233 の準拠について、説明します。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 最新の注意事項と機能情報については、プラットフォームおよびソフトウェア リリースのバグ検索ツールとリリース ノートを参照してください。 このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表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シスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

IF-MIB の使用に関する前提条件

このマニュアルで説明しているインターフェイス グループMIBおよびイーサネットに類似したインターフェイス MIB を使用するには、使用するシステムに SNMP を設定する必要があります。ネットワークのパフォーマンスのモニタリングには、Cisco IOS または CiscoWorks のようなネットワーク管理システム（NMS）の使用が想定されています。 これらのトピックについての詳細は、「関連資料」の項に記載されているマニュアルか、または、使用しているネットワーク管理アプリケーションに付属しているマニュアルを参照してください。

IF-MIB に関する情報

IF-MIB は RFC 2233 に準拠し、サブインターフェイスの SNMP サポートが提供されます。 さらに、SNMP を設定すると、linkUp トラップまたは linkDown トラップの既存のシスコの実装または IETF 規格と互換性がある IF-MIB のいずれかを使用できます。 linkUp トラップおよび linkDown トラップに関する情報については、RFC 2233 を参照してください。

Cisco IOS Release 12.1(2)T/12.0(21)S3 からは、**snmp-server trap link ietf** コマンドを使用し、新しい RFC 2233 IETF 規格に基づいた実装を使用して、ルータを設定できます。 このコマンドによって、サブインターフェイスの通知サポートがイネーブルにされます。以前のシスコ実装の linkUp/linkDown トラップを使用することを選択した場合は、これを使用し続けられるよう、デフォルトではディセーブルにされています。

ただし、以前のシスコ オブジェクトの定義を使用する場合、サブインターフェイスの linkUp/linkDown トラップにある *locIfReason* オブジェクトでは、任意の値が使用されることに、注意してください。 これは、*locIfReason* オブジェクトが OLD-CISCO-INTERFACES-MIB.my を使用する現在のシスコ実装のサブインターフェイスに定義されていないためです。

この機能をイネーブルにしない場合、リンク トラップの varbind リストは、{ifIndex, ifDescr, ifType, locIfReason} で構成されます。 **snmp-server trap link ietf** コマンドを使用してこの機能をイネーブルにした後では、varbind リストは、{inIndex, ifAdminStatus, ifOperStatus, if Descr, ifType} で構成されます。 *locIfReason* オブジェクトも、そのオブジェクトにとって意味のある情報が取得できるかどうかによって、条件付きで、このリストに含まれます。 設定されたサブインターフェイスで

は、取得可能な情報が生成されます。 非 HWIDB インターフェイスでは、*locIfReason* に対して定義されている値はありませんので、トラップ メッセージからは省略されます。

IF-MIB モジュールに対する他のアップデートも、RFC2233 に準拠するように行われています。これらの変更には、ifCounterDiscontinuityTime オブジェクトの追加、および、ifTableLastChange に対する基本サポートの追加が、含まれます。 アップデートされた活性挿抜（OIR）ドライバは、ifTableLastChange の全機能のサポートが行われる将来のリリースに含められる予定です。

IF-MIB の利点

RFC 2233 に準拠

IF-MIB の機能拡張によって、Cisco IOS では、RFC 2233 がサポートされます。 このリリースの前は、Cisco IOS では、RFC 1573 のみがサポートされていました。

サブインターフェイスでの linkUp/linkDown トラップの生成

IF-MIB の機能拡張によって、サブインターフェイスの linkUp および linkDown の SNMP トラップが、正しくサポートされます。一方で、影響が及ぼされないユーザは、以前のシスコ実行を使用して操作を続行できます。

コンテキスト対応 IF-MIB

コンテキスト対応 IF-MIB によって、インターフェイス MIB オブジェクトにクエリーを送信する機能が提供され、返される情報は、SNMP コンテキストがマップされる VRF に限定されます。 特定のホストに送信する必要のある通知を制限するコンテキストで、通知ホストを設定することもできます。

VPN 環境では、異なるインターフェイスは異なる VRF インスタンスに属します。 VRF インスタンスは、SNMP コンテキストに固有に関連付けできます。 コンテキスト対応 IF-MIB では、VRF インスタンスにマップされている指定されたコンテキストが含まれている SNMP 要求を受信すると、コンテキストに関連付けられている VRF に属しているインターフェイスに関連する情報のみが取得されます。

IP ヘルパー アドレスの取得

IF-MIB により、各インターフェイスに設定されているすべてのヘルパー アドレスを取得できます。

SNMP の IETF-Compliant リンク トラップをイネーブルにする方法

IF-MIB の設定は、システムではオプションで、デフォルトではディセーブルにされています。 設定する際には、SNMP の IETF 準拠リンク トラップを有効にする必要があります。 SNMP

linkUp/linkDown トラップの新しいオブジェクト リストの使用をイネーブルにするには、次の作業を実行します。特権 EXEC モードを開始し、次のコマンドを使用します。

手順の概要

1. **enable**
2. **configure terminal**
3. **snmp-server trap link ietf**
4. **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。 • 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： `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	**snmp-server trap link ietf** 例： `Router(config)# snmp-server trap link ietf`	RFC 2233 に準拠する SNMP トラップを有効にします。
ステップ 4	**end** 例： `Router(config)# end`	現在のコンフィギュレーション セッションを終了して、特権 EXEC モードに戻ります。

次の作業

SNMP の IETF 準拠リンク トラップの確認

コマンドが実行中のコンフィギュレーション ファイルにあることを確認するには、特権 EXEC モードで **more system:running-config** コマンドを使用します。

トラブルシューティングのヒント

トラブルシューティングのためにリアルタイムでの SNMP トラップ アクティビティをモニタするには、SNMP debug コマンド（**debug snmp packet** コマンドなど）を使用します。 SNMP debug コマンドのマニュアルについては、http://www.cisco.com/en/US/docs/ios/debug/command/reference/db_book.html にある Cisco.com、または、シスコの Documentation CD-ROM で使用可能な、リリース 12.4 の『*Cisco IOS Debug Command Reference*』を参照してください。

SNMP の IETF-Compliant リンク トラップをイネーブルにする例

次に、IETF 準拠実装を有効にする前の SNMP 関連出力、これを有効が場合のコンフィギュレーション セッション、およびコンフィギュレーション後に変更された出力を示します。

```
Router#
more system:running config
. . .
snmp-server engineID local 00000009000000A1616C2056
snmp-server community public RO
snmp-server community private RW
. . .
Router#
conf term

Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
snmp-server trap link ietf

Router(config)#
end
Router#
 more system:running config
. . .
snmp-server engineID local 00000009000000A1616C2056
snmp-server community public RO
snmp-server community private RW
snmp-server trap link ietf
. . .
```

特定のインターフェイスのリンク トラップをイネーブルまたはディセーブルにするには、次の操作を実行します。

```
7609_supBXL_45(config-if)#snmp trap link-status ?
  permit  Permit the following capability
  <cr>
7609_supBXL_45(config-if)#
```

スイッチオーバー中に linkUp/linkDown トラップをイネーブルにするには、次の操作を実行します。

```
7609_supBXL_45(config)#snmp-server trap link ?
  ietf        Use IETF standard for SNMP traps
  switchover  Enable link up/down traps during switchover
```

SNMP の設定方法および IF-MIB の使用方法

SNMP を使用するためのルータの設定

SNMP を使用した IF-MIB 機能を使用する前に、SNMP がサポートされるようルータを設定する必要があります。

（注） ここで説明する作業の中には、ルータに設定パラメータを設定し、ルータの MIB オブジェクトから値を読み取るために使用する SNMP CLI 構文の例が含まれているものがあります。 これらの SNMP CLI 構文の例は、パブリック ドメイン SNMP ツールを使用して Linux ワークステーションから取られています。 ご使用のワークステーションによっては SNMP CLI 構文が異なる場合があります。 ネットワーク管理ワークステーションの正しい構文については、SNMP ツールに付属の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. **snmp-server community** *string1* **ro**
4. **snmp-server community** *string2* **rw**
5. **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。 • 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： `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。

	コマンドまたはアクション	目的
ステップ 3	**snmp-server community** *string1* **ro** 例： `Router(config)# snmp-server community public ro`	SNMP へのアクセスを許可するコミュニティ アクセス ストリングを設定します。 • *string1* 引数は、1 ～ 32 文字の英数字で構成されるコミュニティ ストリングで、パスワードのように機能して SNMP プロトコルへのアクセスを許可します。 コミュニティ ストリングに空白は使用できません。 • **ro** キーワードは、読み取り専用アクセスを指定します。 このストリングを使用する SNMP 管理ステーションは MIB オブジェクトを取得できます。 （注） この例の SNMP コミュニティ読み取り専用（RO）ストリングは public です。 ご使用の設定では、この値にこれより複雑な構文を使用する必要があります。
ステップ 4	**snmp-server community** *string2* **rw** 例： `Router(config)# snmp-server community private rw`	SNMP へのアクセスを許可するコミュニティ アクセス ストリングを設定します。 • *string2* 引数は、1 ～ 32 文字の英数字で、パスワードのように機能して SNMP プロトコルへのアクセスを許可します。 コミュニティ ストリングに空白は使用できません。 • **rw** キーワードは、読み取りと書き込みアクセスを指定します。 このストリングを使用する SNMP 管理ステーションは、MIB オブジェクトを取得して修正できます。 （注） この例の SNMP コミュニティ読み取り/書き込み（RW）ストリングは private です。 ご使用の設定では、この値にこれより複雑な構文を使用する必要があります。
ステップ 5	**end** 例： `Router(config)# end`	現在の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

IF-MIB を実装するには、トンネルを設定する必要があります。 トンネルの設定については、このガイドの「トンネルの実装」の章を参照してください。

その他の関連資料

関連資料

関連項目	マニュアル タイトル
IPv6 アドレッシングと接続	『*IPv6 Configuration Guide*』
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
IPv6 コマンド	『Cisco IOS IPv6 Command Reference』
Cisco IOS IPv6 機能	『Cisco IOS IPv6 Feature Mapping』

標準および RFC

標準/RFC	タイトル
IPv6 に関する RFC	『*IPv6 RFCs*』

MIB

MIB	MIB のリンク
	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 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/cisco/web/support/index.html

IF-MIB 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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

表 *12* : *IF-MIB* の機能情報

機能名	リリース	機能情報
IF-MIB	12.1(2)T 12.0(21)S3 12.3(2)T 12.0(24)S 12.2(2)SXI 12.2(33)SB Cisco IOS Release 3.9S	ルータは、RFC 2233 IETF 規格ベースの実装を使用して設定できます。 IF-MIB により、サブインターフェイスの通知サポートがイネーブルにされます。 LinkUp/Down トラップは、リンクのアップまたはダウン時に生成されます。 この機能は、ifAdminStatus と ifOperStatus を含むように LinkUp/Down トラップ情報を更新します。 IF-MIB により IP ヘルパー アドレスがサポートされ、各インターフェイスに設定されているすべての IP ヘルパー アドレスを取得できます。 インターフェイス MIB オブジェクトにクエリーを送信する機能が提供され、返される情報は、SNMP コンテキストがマップされる VRF に限定されます。 特定のホストに送信する必要のある通知を制限するコンテキストで、通知ホストを設定することもできます。

第 11 章

同期イーサネット（SyncE）ESMC と SSM

このモジュールでは、同期ステータス メッセージ（SSM）とイーサネット同期メッセージ チャネル（ESMC）について説明し、さらにSyncE機能での簡易ネットワーク管理プロトコル（SNMP）トラップの生成について説明します。

サービス プロバイダー ネットワークで、Synchronous Optical Networking（SONET）と Synchronous Digital Hierarchy（SDH）機器を段階的に置き換えるイーサネット機器を使用する場合、ポート経由で高品質なクロック同期を提供するためには周波数を同期化することが必要です。

同期イーサネット（SyncE）により、物理レベルで必要な同期化が実現します。 SyncE では、イーサネット リンクは SONET/SDH と同じ方法で、高品質なストラタム 1 の追跡可能なクロック信号とビット クロックのタイミングをとることで同期化されます。 処理メッセージが SyncE リンクを維持し、ノードが最も信頼性に優れた送信元から常にタイミングを得るようにします。

SyncE は、イーサネット ポート経由のクロック周波数を同期化します。 クロック情報を伝送するための通信チャネルは SONET/SDH では SSM、SyncE では ESMC です。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 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。 この章に記載されている機能の詳細、および各機能がサポートされているリリースのリストについては、同期イーサネット（SyncE）ESMC と SSM の機能情報, （131 ページ） 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。

同期イーサネット（SyncE）ESMC と SSM の前提条件

最初に SyncE 設定用にネットワーク クロックを設定する必要があります。 ネットワーク クロックの自動同期を有効にする必要があります。 **network-clock-select** コマンドおよび **network-clock-participate** コマンドがコンフィギュレーションにないことを確認して、SyncE 設定を続行してください。

同期イーサネット（SyncE）ESMC と SSM に関する制限事項

- **network-clock synchronization ssm option** コマンドを使用するには、次の条件が必要です。
  - 入力源がコンフィギュレーションに存在しない。
  - ネットワーク クロックの品質レベルがコンフィギュレーションに存在しない。
  - どの同期イーサネット インターフェイスにもネットワーク クロック ソースの品質が設定されていない。
- **network-clock synchronization ssm option** コマンドが、コンフィギュレーションで **network-clock eec** コマンドと互換性がある必要があります。
- **esmc process** コマンドと **synchronous mode** コマンドは、SyncE 対応インターフェイスがルータに設置されている場合にだけ使用することができます。

同期イーサネット（SyncE）ESMC と SSM に関する情報

同期イーサネット（SyncE）ESMC と SSM

パケット ネットワークを使用するユーザは、時分割多重（TDM）回線で複数のリモート ネットワーク要素（NE）にタイミングを提供することは難しいでしょう。 SyncE機能は、パケット ネットワークを介してリモート NE に有効なタイミングを提供することにより、この問題を解決することができます。 SyncE はイーサネット物理レイヤを活用してリモート サイトに周波数を送信します。 SyncE の機能性と正確性は、その物理レイヤの特性により、SONET/SDH ネットワークに

類似しています。 SyncE は ESMC を使用して最善のクロック ソースのトレーサビリティを提供し、タイミング ソースを正しく定義し、タイミング ループを回避することができます。

SONET/SDH は、メッセージの転送で SONET/SDH オーバーヘッド フレームの 2 つの S バイトから 4 ビットを使用します。 イーサネットは、メッセージの転送で IEEE 802.3 構成固有の低速プロトコルに基づく ESMC に依存します。 同期パス上の各 NE は SyncE をサポートし、SyncE はパスの周波数を効果的に提供します。 SyncE は相対時間（位相整列など）も絶対時間（時刻）もサポートしません。

SyncE は、既知で共通の精密周波数基準の周波数の配布をイーサネット物理レイヤ ネットワーク レベルで提供します。 SyncE で使用するクロックは、SONET/SDH 同期ネットワークで使用されるクロックと互換性があります。 ネットワーク同期を行う場合は、出力クロックのパフォーマンスを備えた同期ネットワーク接続を経由するネットワークから同期情報が送信されます。 クロック情報を伝送するための通信チャネルは、SONET/SDH では同期ステータス メッセージ（SSM）、SyncE ではイーサネット同期メッセージ チャネル（ESMC）です。

ESMC は同期証跡のタイミング品質を識別する品質レベル（QL）ID を伝送します。 QL-TLV の QL 値は、SONET および SDH SSM に定義した QL 値と同じです。 ネットワークの送信中に SSM QL によって提供される情報により、最も信頼できるソースから適切なタイミングでノードを取得することができるようになり、タイミング ループが回避されます。 ESMC は同期選択アルゴリズムとともに使用されます。イーサネット ネットワークはすべてのリンクまたはすべての場所で同期している必要がないため、ESMC チャネルはこのサービスを提供します。 ESMC は、標準イーサネット ヘッダーから構成されます。ヘッダーの内容は、構成固有の低速プロトコル、ITU-T OUI、固有の ITU-T サブタイプ、ESMC 固有のヘッダー、フラグ フィールド、およびタイプ、長さ、値（TLV）構造です。 フラグと TLV を使用することにより、SyncE リンクと関連するタイミングの変更の管理体制が向上します。 Cisco 7600 シリーズ ルータの同期イーサネット サポートの詳細については、『Cisco 7600 Series Ethernet Services Plus (ES+) and Ethernet Services Plus T (ES+T) Line Card Configuration Guide』を参照してください。

同期イーサネット（SyncE）ESMC と SSM の設定方法

SyncE の設定

ESMC と SSM を使用して SyncE を設定するには、この作業を実行します。

手順の概要

1. **enable**
2. **network-clock set l ockout** {**external** *slot* / *card* / *port*[**10m**| **2m**| **t1**{**sf** | **esf** | **d4**}] | **interface** *type slot* / *port*}
3. **network-clock clear lockout** {**external** *slot* / *card* / *port* [**10m**| **2m**| **t1** {**sf** | **esf** | **d4**}] | **interface** *type slot* / *port*}
4. **network-clock switch force { external** *slot* / *card* / *port* [**10m** | **2m**] | **t0** | **t1** {**sf** | **esf** | **d4**} **t0** | **internal { external** *slot* / *card* / *port*[**10m** | **2m**] | **t0}** | **interface** *type slot* / *port* **external** *slot* / *card* / *port* [**10m** | **2m**] | **t0** }
5. **network-clock switch manual { interface** *type slot* /port *{* **external** *slot* / *card* / *port* [**10m** | **2m**] | **t0** } | **external** *slot* / *card* / *port*{**10m** | **2m** | **t0** | **t1** {**sf** | **esf** | **d4**} | **internal { external** *slot* / *card* / *port*[**10m** | **2m**] | **t0}** *}*
6. **network-clock clear switch** {**t0** | **external** *slot* / *card* / *port* [**10m** | **2m**]}
7. **configure terminal**
8. **network-clock synchronization automatic**
9. **network-clock synchronization ssm option** {**1**| **2**{**GEN1**| **GEN2**}}
10. **network-clock input-source** *priority* {**external** *slot* / *card* / *port* [**10m** | **2m** | **t1** {**sf** | **esf** | **d4**}] | **interface** *type slot* / *port*}
11. **network-clock synchronization mode ql-enabled**
12. **network-clock hold-off** {**0**| *milliseconds*}
13. **network-clock wait-to-restore** *seconds*
14. **esmc process**
15. **network-clock external** *slot* / *card* / *port* **hold-off** {**0** | *milliseconds*}
16. **network-clock quality-level** {**tx**| **rx**} *value* {**interface** *type slot* / *port* | **external** *slot* / *card* / *port* [**10m** | **2m** | **t1** {**sf** | **esf** | **d4**}]
17. **network-clock output-source {line** | **system**} *priority interface type slot* / *port* **external** *slot* / *card* / *port*[**10m** | **2m** | **t1**{**sf** | **esf** | **d4**} **]**
18. **interface** *type number*
19. **synchronous mode**
20. **esmc mode** [**ql-disabled**| **tx**| **rx**] *value*
21. **network-clock source quality-level** *value* {**tx** | **rx**}
22. **network-clock hold-off** {**0** | *milliseconds*}
23. **network-clock wait-to-restore** *seconds*
24. **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。 • パスワードを入力します（要求された場合）。
ステップ 2	**network-clock set l ockout** {**external** *slot* / *card* / *port*[**10m**	**2m**	**t1**{**sf**	**esf**	**d4**}]	**interface** *type slot* / *port*} 例： `Router# network-clock set lockout GigabitEthernet7/1`	入力のロックアウト状態を「オン」に設定します。これで、入力は選択プロセスで使用できないと見なされます。
ステップ 3	**network-clock clear lockout** {**external** *slot* / *card* / *port* [**10m**	**2m**	**t1** {**sf**	**esf**	**d4**}]	**interface** *type slot* / *port*} 例： `Router# network-clock clear lockout GigabitEthernet7/1`	入力のロックアウト状態を「オフ」に設定します。これで、入力は選択プロセスで使用できると見なされます。
ステップ 4	**network-clock switch force { external** *slot* / *card* / *port* [**10m**	**2m**]	**t0**	**t1** {**sf**	**esf**	**d4**} **t0**	**internal { external** *slot* / *card* / *port*[**10m**	**2m**]	**t0**}	**interface** *type slot* / *port* **external** *slot* / *card* / *port* [**10m**	**2m**]	**t0** } 例： `Router# network-clock switch force interface GigabitEthernet 7/1 t0`	同期元が有効かつロックアウトされていない場合、現在選択されている同期元をオーバーライドします。強制スイッチ コマンドで選択された送信元が無効またはロックアウトされている場合、強制スイッチ コマンドは□動的に拒否されます。
ステップ 5	**network-clock switch manual { interface** *type slot* /port *{* **external** *slot* / *card* / *port* [**10m**	**2m**]	**t0** }	**external** *slot* / *card* / *port*{**10m**	**2m**	**t0**	**t1** {**sf**	**esf**	**d4**}	**internal { external** *slot* / *card* / *port*[**10m**	**2m**]	**t0**} *}* 例： `Router# network-clock switch manual interface GigabitEthernet 7/1 t0`	同期元インターフェイスが有効かつロックアウトされていない場合に同期元を選択します。 以前割り当てられた同期元のプライオリティをオーバーライドするには、手動スイッチが使用されます。

	コマンドまたはアクション	目的
ステップ 6	**network-clock clear switch** {**t0** | **external** *slot* / *card* / *port* [**10m** | **2m**]} 例： `Router# network-clock clear switch t0`	強制スイッチ コマンドと手動スイッチ コマンドをクリアします。 インターフェイスが指定されていない場合、選択された強制/手動インターフェイスは自動的にクリアされます。
ステップ 7	**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 8	**network-clock synchronization automatic** 例： `Router(config)# network-clock synchronization automatic`	ネットワーク クロック選択アルゴリズムをイネーブルにします。 このコマンドを実行すると、シスコ固有のネットワーク クロック プロセスが無効になり、G.781 ベースの自動クロック選択プロセスが有効になります。
ステップ 9	**network-clock synchronization ssm option** {**1**| **2**{**GEN1**| **GEN2**}} 例： `Router(config)# network-clock synchronization ssm option 2 GEN2`	ルータが同期ネットワークで動作するように設定します。 • オプション 1 は、ヨーロッパ向けに設計された同期ネットワークを示します。 768 ビットは、デフォルト値です。 • オプション2は、米国向けに設計された同期ネットワークを示します。
ステップ 10	**network-clock input-source** *priority* {**external** *slot* / *card* / *port* [**10m** | **2m** | **t1** {**sf** | **esf** | **d4**}] | **interface** *type slot* / *port*} 例： `Router(config)# network-clock input-source 1 interface GigabitEthernet 7/1`	クロック ソース ライン、外部タイミング入力インターフェイス、GPS インターフェイス、またはシステムの入力クロックとしてのパケット ベースのタイミング再生クロックとして設定されるインターフェイスの選択を有効にします。 インターフェイスは、SyncE またはチャネライズド SONET の場合があります。
ステップ 11	**network-clock synchronization mode ql-enabled** 例： `Router(config)# network-clock synchronization mode ql-enabled`	自動選択プロセスの ql-enabled モードを設定します。 • QL はデフォルトで無効になっています。 • ql-enabled モードは SSM の送信に同期インターフェイスが使用可能な場合にのみ使用できます。

	コマンドまたはアクション	目的
ステップ 12	**network-clock hold-off** {**0**| *milliseconds*} 例： `Router(config)# network-clock hold-off 0`	（任意）インターフェイスの hold-off タイマーを設定します。
ステップ 13	**network-clock wait-to-restore** *seconds* 例： `Router(config)# network-clock wait-to-restore 70`	（任意）SyncE インターフェイスの wait-to-restore タイマーを設定します。
ステップ 14	**esmc process** 例： `Router(config)# esmc process`	ESMC プロセスを有効にします。
ステップ 15	**network-clock external** *slot* / *card* / *port* **hold-off** {**0** | *milliseconds*} 例： `Router(config)# network-clock external 0/1/0 hold-off 0`	インターフェイスの hold-off タイマーをオーバーライドします。
ステップ 16	**network-clock quality-level** {**tx**| **rx**} *value* {**interface** *type slot* / *port* | **external** *slot* / *card* / *port* [**10m** | **2m** | **t1** {**sf** | **esf** | **d4**}] 例： `Router(config)# network-clock quality-level rx QL-STU GigabitEthernet 0/0/0`	回線または外部タイミング入出力に対して QL 値を強制します。
ステップ 17	**network-clock output-source {line** | **system**} *priority interface type slot* / *port* **external** *slot* / *card* / *port*[**10m** | **2m** | **t1**{**sf** | **esf** | **d4**} **]** 例： `Router(config)# network-clock output-source line 1 GigabitEthernet1/2 external 0/0/1 10m`	外部タイミング入力インターフェイスから外部タイミング出力インターフェイスに信号を送信します。
ステップ 18	**interface** *type number* 例： `Router(config)# interface GigabitEthernet 0/0`	インターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 19	**synchronous mode** 例： `Router(config-if)# synchronous mode`	イーサネット インターフェイスを同期モードに設定し、インターフェイスで ESMC プロセスおよび QL プロセスを自動的に有効にします。
ステップ 20	**esmc mode** [**ql-disabled**| **tx**| **rx**] *value* 例： `Router(config-if)# esmc mode rx QL-STU`	（任意）インターフェイスで ESMC プロセスを有効にします。
ステップ 21	**network-clock source quality-level** *value* {**tx**	**rx**} 例： `Router(config-if)# network-clock source quality-level QL-ST4 tx`	（任意）ローカル クロック選択プロセスに強制 QL 値を提供します。
ステップ 22	**network-clock hold-off** {**0**	*milliseconds*} 例： `Router(config-if)# network-clock hold-off 0`	（任意）インターフェイスの hold-off タイマーを設定します。
ステップ 23	**network-clock wait-to-restore** *seconds* 例： 例： `Router(config-if)# network-clock wait-to-restore 70`	（任意）SyncE インターフェイスの wait-to-restore タイマーを設定します。
ステップ 24	**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。

SyncE イベントでの SNMP トラップの有効化と無効化

簡易ネットワーク管理プロトコル（SNMP）トラップは、非送信請求情報についてネットワーク管理システム（NMS）に SNMP エージェントが通知するように定義されます。 SNMP トラップは、重大な SyncE イベントがデバイスで発生した場合に NMS に通知します。 SNMP トラップが

SyncE 設定で有効になっている場合、SNMP エージェント コードが SyncE イベント用の SyncE トラップを生成します。

SyncE イベントの SNMP トラップの有効化と無効化を実行するには、次の作業を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps netsync**
4. **no snmp-server enable traps netsync**
5. **end**
6. **show running-config all | include traps**

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。 • 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： `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	**snmp-server enable traps netsync** 例： `Router(config)# snmp-server enable traps netsync`	SyncE トラップを有効にします。
ステップ 4	**no snmp-server enable traps netsync** 例： `Router(config)#` **no snmp-server enable traps netsync**	（任意）SyncE トラップを無効にします。
ステップ 5	**end** 例： `Router(config)# end`	グローバル コンフィギュレーション モードを終了します。

	コマンドまたはアクション	目的
ステップ 6	**show running-config all | include traps** 例： `Router# show running-config all | include trap`	（任意）ルータ上で有効な SyncE トラップを表示します。

同期イーサネット（SyncE）ESMC と SSM の設定例

同期イーサネット（SyncE）ESMC と SSM の例

次に、SyncE 設定シーケンスの例を示します（2 個の SyncE インターフェイスと 2 個の外部インターフェイスを備えたインターフェイスを設定しています）。

```
Interface GigabitEthernet0/0/0
    synchronous mode
    clock source line
    network-clock wait-to-restore 720
!
Interface GigabitEthernet1/0/0
    synchronous mode
    clock source line
!
network-clock synchronization automatic
network-clock input-source 1 external 0/0/0 2m
network-clock input-source 2 external 1/0/0 2m
network-clock output-source line 1 interface GigabitEthernet0/0/0 external 0/0/0 2m
network-clock output-source line 1 interface GigabitEthernet1/0/0 external 1/0/0 2m
```

次に、ESMC が有効になっているかどうかを確認する例を示します。

```
Router# show esmc

Interface: GigabitEthernet0/0/0
Administrative configurations:
  Mode: Synchronous
  ESMC TX: Enable
  ESMC RX : Enable
  QL RX configured : NA
  QL TX configured : NA
Operational status:
  Port status: UP
  QL Receive: QL-SSU-B
  ESMC Information rate : 1 packet/second
  ESMC Expiry: 5 second
```

次に、ネットワーク クロック同期詳細情報の表示例を示します。

```
Router# show network-clock synchronization detail

Automatic selection process : Enable
Equipment Clock : 2048 (EEC-Option1)
Clock Mode : QL-Enable
ESMC : Disabled
```

```
SSM Option : 1
T0 : Internal
Hold-off (global) : 300 ms
Wait-to-restore (global) : 300 sec
Revertive : No
Force Switch: FALSE
Manual Switch: FALSE
Number of synchronization sources: 1
Secondary src: Ethernet0/0
Slots disabled 0x0
Monitor source(s):  Ethernet0/0
Selected QL: QL-SEC
sm(netsync_ql_dis NETCLK_QL_ENABLE), running yes, state 1A
Last transition recorded: (begin)-> 1A (ql_mode_enable)-> 1A (src_added)-> 1A


Nominated Interfaces

 Interface            SigType     Mode/QL      Prio  QL_IN     ESMC Tx   ESMC Rx
*Internal             NA          NA/Dis       251   QL-SEC    NA        NA
 Et0/0                NA          Sync/En      2     QL-DNU    -         -

Interface:
-----------------------------------------------
Local Interface: Internal
Signal Type: NA
Mode: NA(Ql-enabled)
SSM Tx: Disable
SSM Rx: Disable
Priority: 251
QL Receive: QL-SEC
QL Receive Configured: -
QL Receive Overrided: -
QL Transmit: -
QL Transmit Configured: -
Hold-off: 0
Wait-to-restore: 0
Lock Out: FALSE
Signal Fail: FALSE
Alarms: FALSE
Slot Disabled: FALSE

Local Interface: Et0/0
Signal Type: NA
Mode: Synchronous(Ql-enabled)
ESMC Tx: Enable
ESMC Rx: Enable
Priority: 2
QL Receive: QL-DNU
QL Receive Configured: -
QL Receive Overrided: -
QL Transmit: -
QL Transmit Configured: -
Hold-off: 300
Wait-to-restore: 300
Lock Out: FALSE
Signal Fail: FALSE
Alarms: FALSE
Slot Disabled: FALSE
Dont Use: FALSE
Configured Priority: 2
Force Switch: FALSE
Manual Switch: FALSE
Manual Switch In progress: FALSE
Holdoff_cfg: FALSE
Wtr_cfg: FALSE
Reason for alarm flag: 0
Msw in progress: FALSE
Intf_sig_nv: 0
Hold off Timer: Stopped
```

```
Wait to restore Timer: Stopped
Switchover Timer: Stopped
ESMC Tx Timer: Stopped
ESMC Rx Timer: Stopped
Tsm Delay Timer: Stopped
```

SyncE イベントでの SNMP トラップの有効化と無効化の例

次に、SyncE イベントでの SNMP トラップの有効化と無効化の方法を示します。

```
Router > enable
Router # configure terminal
Router(config)# snmp-server enable traps netsync
Router (config)# no snmp-server enable traps netsync
Router (config)# end
Router# show running-config all| include traps
snmp-server enable traps flowmon
snmp-server enable traps sonet
snmp-server enable traps netsync
```

その他の関連資料

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
インターフェイスおよびハードウェア コンポーネント コンフィギュレーション コマンド	『*Cisco IOS Interface and Hardware Component Command Reference*』
Cisco 7600 同期イーサネット	Cisco 7600 Series Ethernet Services Plus (ES+) and Ethernet Services Plus T (ES+T) Line Card Configuration Guide

標準

標準	タイトル
ITU-T G.8262	同期イーサネット機器のスレーブ クロック（*EEC*）タイミング特性
ITU-T G.8264	パケット ネットワークを介したタイミングの配布
ITU-T G.781	同期レイヤ機能

MIB

MIB	MIB のリンク
CISCO-NETSYNC-MIB	選択したプラットフォーム、Cisco ソフトウェア リリース、およびフィーチャ セットの MIB を検索してダウンロードする場合は、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
なし	--

シスコのテクニカル サポート

説明	リンク
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 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/cisco/web/support/index.html

同期イーサネット（SyncE）ESMC と SSM 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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

表 *13*：同期イーサネット（*SyncE*）：*ESMC* と *SSM* の機能情報

機能名	リリース	機能情報
SyncE 機能での SNMP トラップの生成	15.1(2)S Cisco IOS XE Release 3.8S	この機能は、任意の非送信請求情報について NMS に通知できるように SyncE に SNMP トラップを設定する方法を説明します。 この機能により、次のコマンドが導入または変更されました。 **no snmp-server enable traps netsync**、**show running-config all| include trap**、**snmp-server enable traps netsync**
同期イーサネット（SyncE）：ESMC と SSM	15.0(1)S Cisco IOS XE Release 3.8S	この機能は、SyncE が品質レベルの選択を伴うイーサネットポート経由のクロック周波数を同期化できるように、ESMC と SSM 制御プロトコルをサポートします。 **esmc mode ql-disabled**、**esmc process**、**show esmc**、**show interfaces accounting** の各コマンドが、この機能によって導入または変更されました。

第 12 章

1+1 SR-APS Without Bridging

自動保護スイッチング（APS）機能は、リンクの冗長性を提供し、回線障害の発生時に Packet over SONET（POS）回線のスイッチオーバーを可能にします。この機能は多くの場合、Synchronous Optical Networking（SONET）装置を通信装置に接続する際に必要となります。 Single Router（SR）APS 機能では、保護インターフェイスと現用インターフェイスの両方が同じルータ上にある必要があります。

APS は、現用 POS インターフェイスのバックアップとして、SONET ネットワーク内の保護 POS インターフェイスを使用するメカニズムです。 現用インターフェイスに障害が発生した場合、保護インターフェイスが即座にそのトラフィック負荷を引き継ぎます。 設定に基づいて、2つの回線は同じルータで終端できます。 保護メカニズムには、双方向接続による 1+1 アーキテクチャが含まれています。 ブリッジングとは、ユーザ データを現用インターフェイスと保護インターフェイスの両方に送信することを意味します。 非ブリッジングの場合は、現用インターフェイスだけにユーザ データが送信されます。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 最新の注意事項と機能情報については、プラットフォームおよびソフトウェア リリースのバグ検索ツールとリリース ノートを参照してください。 このモジュールに記

載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表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シスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

1+1 SR-APS Without Bridging の前提条件

インターフェイスのIPアドレスとともに、最初に現用インターフェイスを設定します。この設定により、APS の設定中に保護インターフェイスがアクティブ回線になるのを防止します。 保護インターフェイスが誤って最初に設定され、アクティブになった場合は、**shut** コマンドまたは **no shut** コマンドを使用して現用インターフェイスをアクティブにすることができます。

1+1 SR-APS Without Bridging の制約事項

- 保護インターフェイスと現用インターフェイスは、まったく同じに設定する必要があります。 2 つのインターフェイスの設定が異なっていても、警告メッセージは表示されません。
- 保護インターフェイスと現用インターフェイスの設定が同じでない場合、APS ペア（保護インターフェイスと現用インターフェイス）の動作は予測できません。
- 活性挿抜（OIR）時、または共有ポートアダプタ（SPA）やキャリア カード（CC）の破損時には、50 ミリ秒以内の APS スイッチオーバーはサポートされません。
- APS の切り替えがルート プロセッサ（RP）またはフォワーディング プレーン（FP）のハイアベイラビリティ（HA）と同時に行われる場合は、50 ミリ秒以内である必要はありません。

1+1 SR-APS Without Bridging に関する情報

1+1 SR-APS Without Bridging

APS 機能は、リンクの冗長性を提供し、回線障害の発生時に POS 回線のスイッチオーバーを可能にします。この機能は多くの場合、SONET 装置を通信装置に接続する際に必要となります。SR-APS機能では、保護インターフェイスと現用インターフェイスの両方が同じルータ上にある必要があります。

APS は、現用 POS インターフェイスのバックアップとして、SONET ネットワーク内の保護 POS インターフェイスを使用するメカニズムです。 現用インターフェイスに障害が発生した場合、保護インターフェイスが即座にそのトラフィック負荷を引き継ぎます。 設定に基づいて、2 つの回線は同じルータで終端できます。 保護メカニズムには、双方向接続による 1+1 アーキテクチャが含まれています。

1+1 アーキテクチャでは、1 つの現用インターフェイス（回線）と 1 つの保護インターフェイスが存在し、送信側からのペイロードと同じペイロードが両方の受信側に送信されます。 受信側は、使用する必要のあるインターフェイスを決定します。 SONET フレーム内の回線オーバーヘッド（LOH）バイト（K1 および K2）は、ステータスと処理の両方を示します。 あるインターフェイスがダウンした場合、または K1/K2 バイトが変化した場合、APS は通常のインターフェイス設定メッセージを使用して保護インターフェイスを起動します。

ブリッジングとは、ユーザ データを現用インターフェイスと保護インターフェイスの両方に送信することを意味します。非ブリッジングの場合は、現用インターフェイスにのみユーザデータが送信されます。現用インターフェイスがアクティブインターフェイスになるように設定する必要があります。 Cisco ASR 1000 シリーズ ルータ（ASR1000）は、非ブリッジングだけをサポートしています。

非ブリッジングでは、ASR1000（APS をイネーブルに設定）はリモート エンドに信号を送信します。 ASR 1000 は（K1/K2 バイト以外の）信号を現用インターフェイスにのみ送信し、保護インターフェイスには送信しません。 K1/K2 バイトは保護インターフェイスにのみ送信されます。 一方、ASR 1000 をブリッジング APS 対応のデバイスに接続することは可能です。つまり、デバイスは ASR1000 の現用インターフェイスと保護インターフェイスの両方に同じ信号を送信します。ただし ASR 1000 は（K1/K2 バイト以外の）ユーザ データをデバイスの現用インターフェイスにのみ送信します。 K1/K2 バイトは保護インターフェイスに送信されます。

SR-APS では、現用インターフェイスと保護インターフェイスの間で Protect Group Protocol（PGP）が使用されます。 保護インターフェイスの APS 設定には、PGP を使用して現用インターフェイスと通信するために、同一ルータ上のループバック インターフェイスの IP アドレスを含める必要があります。 PGP を使用することで、チャネル信号の劣化または損失、または手動介入が発生した場合に、POS インターフェイスを切り替えることができます。 双方向モードでは、受信および送信チャネルはペアとして切り替わります。

双方向 APS では、ローカル接続とリモート接続が、データ パス用に選択される入力インターフェイスをネゴシエートします。 出力インターフェイスのトラフィックは、現用インターフェイスと保護インターフェイスのいずれにも送信されません。

1+1 SR-APS Without Bridging の設定方法

APS 動作および保護インターフェイスの設定

APS 現用および保護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. **enable**
2. **configure terminal**
3. **interface pos** *slot/sub-slot/port*
4. **aps working** *circuit-number*
5. **aps protect** *circuit-number ip-address*
6. **end**
7. **show controllers pos**
8. **show interfaces pos**
9. **show aps**

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。 • 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： `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 pos** *slot/sub-slot/port* 例： `Router(config)# interface pos 2/0/0`	動作インターフェイスとして設定する POS インターフェイスを指定し、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	**aps working** *circuit-number* 例： `Router(config-if)# aps working 1`	インターフェイスを現用インターフェイスとして設定します。
ステップ 5	**aps protect** *circuit-number ip-address* 例： `Router(config-if)# aps protect 1 209.165.200.224`	インターフェイスを保護インターフェイスとして設定します。現用インターフェイスを含む同一ルータ上のループバック インターフェイスの IP アドレスを指定します。

	コマンドまたはアクション	目的
ステップ 6	**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。
ステップ 7	**show controllers pos** 例： `Router(config)# show controllers pos`	インターフェイスが正しく設定されていることを確認できるように、POS コントローラに関する情報を表示します。
ステップ 8	**show interfaces pos** 例： `Router(config)# show interfaces pos`	設定されたインターフェイスに関する情報を表示します。
ステップ 9	**show aps** 例： `Router(config)# show aps`	設定されたルータの APS に関する情報を表示します。

その他の APS オプションの設定

その他の APS オプションを設定するには、次の作業を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface pos** *slot/sub-slot/port*
4. **aps force** *circuit-number*
5. **aps group** *group-number*
6. **aps lockout** *circuit-number*
7. **aps manual** *circuit-number*
8. **aps revert** *minutes*
9. **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。 • 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： `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 pos** *slot/sub-slot/port* 例： `Router(config)# interface pos 2/0/0`	動作インターフェイスとして設定する POS インターフェイスを指定し、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	**aps force** *circuit-number* 例： `Router(config-if)# aps force 1`	（任意）同等かより高いプライオリティを持つ要求が実行されている場合を除いて、指定された回線を保護インターフェイスに手動で切り替えます。
ステップ 5	**aps group** *group-number* 例： `Router(config-if)# aps group 20`	（任意）ルータ上で複数の保護インターフェイスまたは現用インターフェイス グループをサポートできるようにします。
ステップ 6	**aps lockout** *circuit-number* 例： `Router(config-if)# aps lockout 1`	（任意）動作インターフェイスが保護インターフェイスに切り替わるのを防ぎます。
ステップ 7	**aps manual** *circuit-number* 例： `Router(config-if)# aps manual 1`	（任意）同等かより高いプライオリティを持つ要求が実行されている場合を除いて、回線を保護インターフェイスに手動で切り替えます。
ステップ 8	**aps revert** *minutes* 例： `Router(config-if)# aps revert 3`	（任意）動作インターフェイスが使用可能になった後、保護インターフェイスから動作インターフェイスへの自動切り替えをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 9	**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。

APS のモニタリングとメンテナンス

APS のモニタおよび保守を行うには、次の作業を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **show controllers pos**
4. **show interfaces pos**
5. **show aps**

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。 • 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： `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	**show controllers pos** 例： `Router(config)# show controllers pos`	インターフェイスが正しく設定されていることを確認できるように、POS コントローラに関する情報を表示します。

	コマンドまたはアクション	目的
ステップ 4	**show interfaces pos** 例： `Router(config)# show interfaces pos`	設定されたインターフェイスに関する情報を表示します。
ステップ 5	**show aps** 例： `Router(config)# show aps`	設定されたルータの APS に関する情報を表示します。

SONET アラーム レポーティングの設定

レポートされる SONET アラームのしきい値およびタイプを設定するには、次のコマンドのいずれかを使用します。 ここで示すコマンドは任意です。 現在のビット エラー レート（BER）しきい値の設定を表示、または SONET アラームのレポートを表示するには、**show controllers pos** コマンドを使用します。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface pos** *slot/sub-slot/port*
4. **pos threshold** {**b1-tca** | **b2-tca** | **b3-tca** | **sd-ber** | **sf-ber**} *rate*
5. **pos report** {**b1-tca** | **b2-tca** | **b3-tca** | **lais** | **lrdi** | **pais** | **plop** | **prdi** | **rdool** | **sd-ber** | **sf-ber** | **slof** | **slos**}
6. **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。 • 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： `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。

	コマンドまたはアクション	目的
ステップ 3	**interface pos** *slot/sub-slot/port* 例： `Router(config)# interface pos 2/0/0`	動作インターフェイスとして設定する POS インターフェイスを指定し、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	**pos threshold** {**b1-tca** | **b2-tca** | **b3-tca** | **sd-ber** | **sf-ber**} *rate* 例： `Router(config-if)# pos threshold b1-tca 4`	（任意）信号障害（SF）、信号劣化（SD）、またはしきい値超過アラーム（TCA）の BER しきい値を設定します。
ステップ 5	**pos report** {**b1-tca** | **b2-tca** | **b3-tca** | **lais** | **lrdi** | **pais** | **plop** | **prdi** | **rdool** | **sd-ber** | **sf-ber** | **slof** | **slos**} 例： `Router(config-if)# pos report b2-tca`	（任意）選択された SONET アラームのレポーティングをイネーブルにします。
ステップ 6	**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。

APS スイッチオーバー トリガーとしての LAIS の設定

現用インターフェイスを管理シャットダウン状態にすると、**pos ais-shut** の有無にかかわらずスイッチオーバーが発生します。 インターフェイスで **pos ais-shut** を有効にすると、インターフェイスは管理シャットダウンのリモートエンドに回線アラーム検出信号（LAIS）アラームを送信します。LAIS アラームにより、スイッチオーバーが少し速くなります。 **carrier-delay msec** *milliseconds* コマンドおよび **ppp timeout retry seconds** [*milliseconds*] コマンドも、APS スイッチオーバーの発生を高速化するために使用されます。

carrier-delay msec *milliseconds* コマンドは、POS インターフェイスのリンク ダウン イベント処理を遅らせます。 たとえば、キャリア遅延を 50 ミリ秒（ms）に設定した場合、ルータは 50 ミリ秒以内にクリアされるすべてのリンク ダウン イベントを無視します。 リンクがダウンした場合、50 ミリ秒間は APS スイッチオーバーが発生しません。 デフォルトのキャリア遅延は 2 秒で、APS スイッチオーバーはリンクのダウン後 2 秒間発生しません。 したがって、スイッチオーバーを高速化するためにキャリア遅延が 50 ミリ秒に設定されます。

ppp timeout retry seconds [*milliseconds*] コマンドは、指定した時間に PPP リトライ タイムアウトを設定します。 たとえば、タイムアウト リトライを 200 ミリ秒に設定した場合、ルータは APS スイッチオーバーによる信号停止を検出後、200 ミリ秒で PPP リンクを確立しようと試みます。デフォルトのリトライ タイムアウトである 2 秒を使用した場合は、APS スイッチオーバーの 2 秒後に PPP リンクが確立されます。 したがって、スイッチオーバーを高速化するために PPP タイムアウト リトライが 50 ミリ秒に設定されます。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface pos** *slot/sub-slot/port*
4. **pos ais-shut**
5. **carrier-delay msec** *milliseconds*
6. **ppp timeout retry seconds** [*milliseconds*]
7. **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。 • 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： `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 pos** *slot/sub-slot/port* 例： `Router(config)# interface pos 2/0/0`	動作インターフェイスとして設定する POS インターフェイスを指定し、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	**pos ais-shut** 例： `Router(config-if)# pos ais-shut`	インターフェイスの管理シャットダウン時に回線アラーム検出信号（LAIS）アラームを送信します。

	コマンドまたはアクション	目的
ステップ 5	**carrier-delay msec** *milliseconds* 例： `Router(config-if)# carrier-delay msec 50`	POS インターフェイスのリンク ダウン イベント処理を遅らせて、APS スイッチオーバーを高速化します。
ステップ 6	**ppp timeout retry seconds** [*milliseconds*] 例： `Router(config-if)# ppp timeout retry 0 200`	PPP ネゴシエーション時の応答に対する最大待ち時間を設定して、APS スイッチオーバーを高速化します。
ステップ 7	**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。

1+1 SR-APS Without Bridging の設定例

1+1 SR-APS Without Bridging の設定例

次の例では、1+1 SR-APS の設定シーケンスを示します。

```
interface loopback 1
ip address 1.1.1.1 255.255.255.0
interface pos 2/0/0
     aps group 1
     aps working 1
     pos ais-shut
     end
interface pos 3/0/0
     aps group 1
     aps protect 1 1.1.1.1
     pos ais-shut
     end
```

次の例では、現用インターフェイスを持つルータに設定された APS の出力例を示します。

```
Router# show aps
POS2/1/1 APS Group 0: protect channel 0 (Inactive)
        Working channel 1 at 10.0.1.1 (Enabled)
        bidirectional, revertive (60 seconds)
        PGP timers (default): hello time=1; hold time=3
                hello fail revert time=120
        SONET framing; SONET APS signalling by default
        Received K1K2: 0x00 0x05
                No Request (Null)
        Transmitted K1K2: 0x00 0x05
```

```
                No Request (Null)
        Remote APS configuration: (null)
POS2/1/0 APS Group 0: working channel 1 (Active)
        Protect at 10.0.1.1
        PGP timers (from protect): hello time=1; hold time=3
        SONET framing
        Remote APS configuration: (null)
```

次の例では、POS コントローラの表示を示します。

```
Router# show controller pos 2/1/0
POS2/1/0
SECTION
  LOF = 0          LOS    = 1                            BIP(B1) = 0
LINE
  AIS = 2          RDI    = 2          FEBE = 14         BIP(B2) = 0
PATH
  AIS = 2          RDI    = 2          FEBE = 4          BIP(B3) = 6
  PLM = 0          UNEQ   = 0          TIM  = 0          TIU     = 0
  LOP = 1          NEWPTR = 2          PSE  = 0          NSE     = 0
Active Defects: None
Active Alarms:  None
Alarm reporting enabled for: SF SLOS SLOF B1-TCA B2-TCA PLOP B3-TCA
Framing: SONET
APS
 working (active)
  COAPS = 13          PSBF = 0
  State: PSBF_state = False
  Rx(K1/K2): 00/00  Tx(K1/K2): 00/00
  Rx Synchronization Status S1 = 00
  S1S0 = 00, C2 = CF
  Remote aps status (none); Reflected local aps status (none)
CLOCK RECOVERY
  RDOOL = 0
  State: RDOOL_state = False
PATH TRACE BUFFER: STABLE
  Remote hostname : SPA-APS2
  Remote interface: POS2/2/0
  Remote IP addr  : 10.1.1.1
  Remote Rx(K1/K2): 00/00  Tx(K1/K2): 00/00
BER thresholds:  SF = 10e-3  SD = 10e-6
TCA thresholds:  B1 = 10e-6  B2 = 10e-6  B3 = 10e-6
Clock source:  internal
```

次の例では、POS インターフェイスの設定情報および統計情報を示します。

```
Router# show interface pos 2/1/0
POS2/1/0 is up, line protocol is up  (APS working - active)
  Hardware is SPA-4XOC12-POS
  Internet address is 10.1.1.2/24
  MTU 4470 bytes, BW 155000 Kbit/sec, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Scramble disabled
  Last input 00:00:02, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/375/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
     102477 packets input, 2459448 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 4 giants, 0 throttles 0 parity
     4 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     102486 packets output, 2459934 bytes, 0 underruns
     0 output errors, 0 applique, 2 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
     10 carrier transitions
```

その他の関連資料

関連資料

関連項目	マニュアル タイトル
APS コマンド	『*Cisco IOS Interface and Hardware Component Command Reference*』

標準

標準	タイトル
なし	--

MIB

MIB	MIB のリンク
なし	選択したプラットフォーム、Cisco ソフトウェア リリース、およびフィーチャ セットの MIB を検索してダウンロードする場合は、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
なし	

シスコのテクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンラインリソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/cisco/web/support/index.html

1+1 SR-APS Without Bridging 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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

表 *14*：*1+1 SR-APS Without Bridging* の機能情報

機能名	リリース	機能情報
1+1 SR-APS Without Bridging	Cisco IOS XE Release 3.1S	この機能では、1+1 Single Router APS Without Bridging のサポートが提供されます。 この機能で導入または変更されたコマンドはありません。

第 13 章

IPv6 Rapid Deployment

IPv6 Rapid Deployment 機能により、サービス プロバイダーは、IPv4 による IPv6 のカプセル化を使用して、自身の IPv4 ネットワーク上でユニキャスト IPv6 サービスをお客様に提供できます。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 最新の注意事項と機能情報については、プラットフォームおよびソフトウェア リリースのバグ検索ツールとリリース ノートを参照してください。 このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表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シスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

IPv6 Rapid Deployment に関する情報

IPv6 Rapid Deployment トンネル

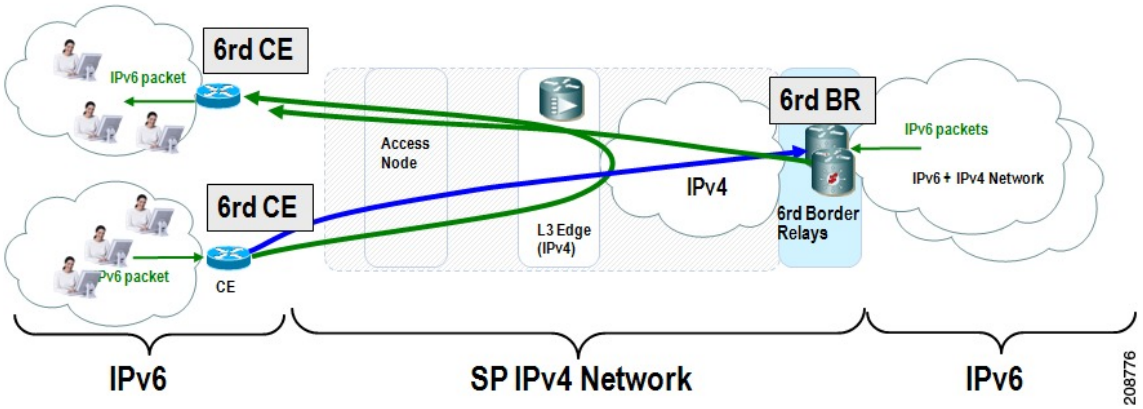
6RD機能は、6to4機能を拡張したものです。 6RD機能により、サービスプロバイダー（SP）は、IPv4 による IPv6 のカプセル化を使用して、自身の IPv4 ネットワーク上でユニキャスト IPv6 サービスをお客様に提供できます。

6RD と 6to4 トンネリングの主な違いは次のとおりです。

- 6RDはアドレスに2002::/16プレフィックスを割り当てる必要はありません。したがって、プレフィックスはSPの自身のアドレス ブロックから割り当てることができます。 この機能により、6RDの動作ドメインをSPネットワーク内にすることができます。 カスタマー サイトと 6RD 対応 SP ネットワークに接続された一般 IPv6 インターネットの観点から提供される IPv6 サービスは、ネイティブ IPv6 と同等です。
- IPv4 宛先の 32 ビットすべてを IPv6 ペイロード ヘッダーで伝送する必要はありません。 IPv4 宛先は、ペイロード ヘッダー内にあるビット データとルータ上の情報を組み合わせて求められます。 さらに、IPv4 アドレスは、6to4 の場合とは異なり、IPv6 ヘッダー内での位置が固定ではありません。

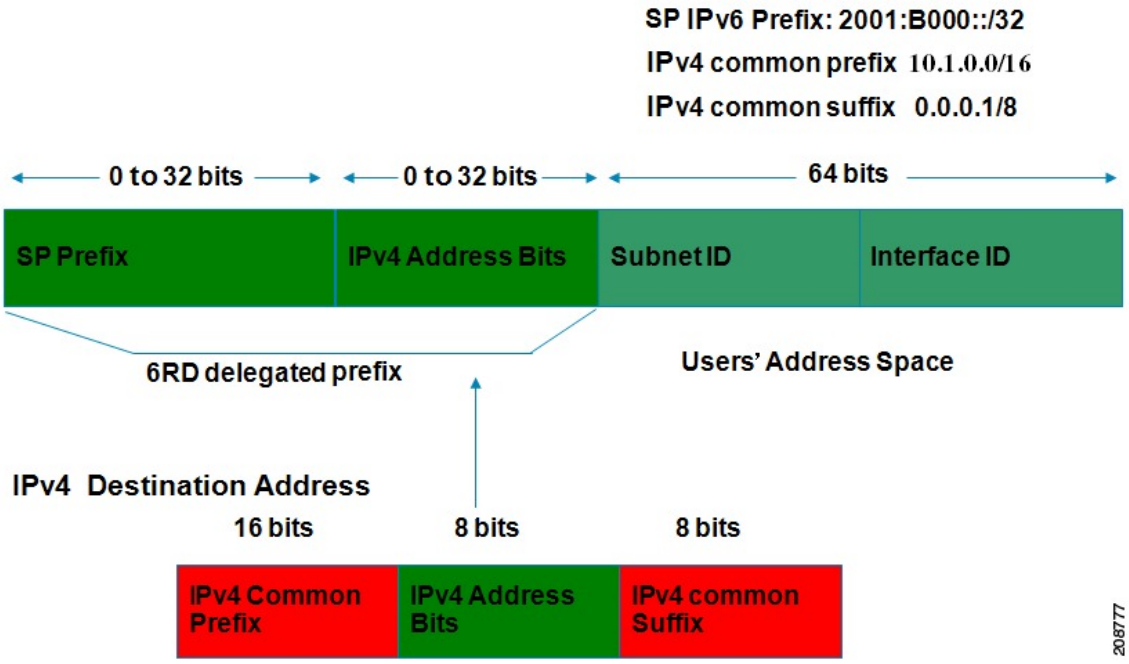
6RD SP プレフィックスは、次の図に示すように IPv6 導入の SP によって選択されました。 6RD 委任プレフィックスは SP プレフィックスと IPv4 アドレス ビットから取得され、CE によってサイト内のホストに使用されます。

図 *4*：*6RD* の展開



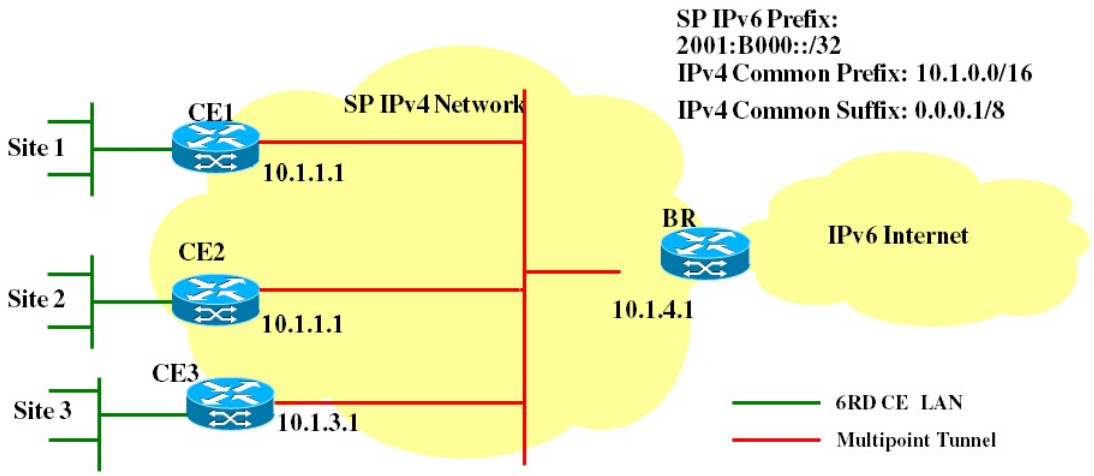
次に、6RD プレフィックス委任がどのように動作するかを示します。

図 *5* : *6RD* プレフィックス委任の説明



次に、6RD プレフィックス委任のトポロジを示します。

図 *6* : *6RD* プレフィックス委任と説明



SP Prefix	2001:B000::/32
IPv4 Common Prefix	10.1.0.0/16
IPv4 Common Suffix	0.0.0.1/8
CE1: Delegated 6RD prefix	2001:B000:0100::/40
CE2: Delegated 6RD prefix	2001:B000:0200::/40
BR: Delegated 6RD prefix	2001:B000:0400::/40
CE1 (IPv4) tunnel transport source	10.1.1.1
CE2 (IPv4) tunnel transport source	10.1.2.1
BR (IPv4) tunnel transport source	10.1.4.1

208778

IPv6 Rapid Deployment の設定方法

6RD トンネルの設定

手順の概要

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **tunnel source** {*ip-address*| *interface-t ype interface-number*}
5. **tunnel mode ipv6ip** [**6rd** | **6to4** | **auto-tunnel** | **isatap**]
6. **tunnel 6rd prefix** *ipv6-prefix* / *prefix-length*
7. **tunnel 6rd ipv4** {**prefix-length** *length*} {**suffix-length** *length*}

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。 • 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： `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** {*ip-address*| *interface-t ype interface-number*} 例： `Router(config-if)# tunnel source loopback 1`	トンネルインターフェイスの送信元インターフェイスのタイプおよび番号を指定します。

	コマンドまたはアクション	目的
ステップ 5	**tunnel mode ipv6ip** [**6rd** | **6to4** | **auto-tunnel** | **isatap**] 例： `Router(config-if)# tunnel mode ipv6ip 6rd`	スタティック IPv6 トンネル インターフェイスを設定します。 • **auto-tunnel** キーワードは、Cisco ASR 1000 シリーズ ルータではサポートされません。
ステップ 6	**tunnel 6rd prefix** *ipv6-prefix* / *prefix-length* 例： `Router(config-if)# tunnel 6rd prefix 2001:B000::/32`	IPv6 rapid 6RD トンネル上で共通の IPv6 プレフィックスを指定します。
ステップ 7	**tunnel 6rd ipv4** {**prefix-length** *length*} {**suffix-length** *length*} 例： `Router(config-if)# tunnel 6rd ipv4 prefix-length 16 suffix 8`	ドメイン内のすべての 6RD ルータに共通の IPv4 トランスポートアドレスのプレフィックス長およびサフィクス長を指定します。

IPv6 Rapid Deployment の設定例

例：6RD トンネルの設定

次の例では、6RD トンネルの実行コンフィギュレーションとそれに対応する **show tunnel 6rd** コマンドの出力を示します。

```
interface Tunnel1
 ipv6 address 2001:B000:100::1/32
 tunnel source loopback 1
 tunnel mode ipv6ip 6rd
 tunnel 6rd prefix 2001:B000::/32
 tunnel 6rd ipv4 prefix-len 16 suffix-len 8
end
Router# show tunnel 6rd tunnel 1
Interface Tunnel1:
  Tunnel Source: 10.1.1.1
  6RD: Operational, V6 Prefix: 2001:B000::/32
  V4 Common Prefix Length: 16, Value: 10.1.0.0
  V4 Common Suffix Length: 8, Value: 0.0.0.1
```

その他の関連資料

関連資料

関連項目	マニュアル タイトル
IPv6 アドレッシングと接続	『*IPv6 Configuration Guide*』
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
IPv6 コマンド	『Cisco IOS IPv6 Command Reference』
Cisco IOS IPv6 機能	『Cisco IOS IPv6 Feature Mapping』

標準および RFC

標準/RFC	タイトル
IPv6 に関する RFC	『*IPv6 RFCs*』

MIB

MIB	MIB のリンク
	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 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/cisco/web/support/index.html

IPv6 Rapid Deployment 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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

表 *15*：*IPv6 Rapid Deployment* の機能情報

機能名	リリース	機能情報
IP トンネリング：6RD IPv6 Rapid Deployment	Cisco IOS XE Release 3.1S	6RD 機能により、サービス プロバイダーは、IPv4 による IPv6 のカプセル化を使用して、自身の IPv4 ネットワーク上でユニキャスト IPv6 サービスをお客様に提供できます。 **tunnel 6rd ipv4**、**tunnel 6rd prefix**、**tunnel mode ipv6ip**、**tunnel source** の各コマンドが導入または変更されました。

第 14 章

IPv6 自動 6to4 トンネル

この機能は、IPv6 自動 6to4 トンネルに対するサポートを提供します。 自動 6to4 トンネルを使用すると、孤立した IPv6 ドメインを、IPv4 ネットワークを介してリモート IPv6 ネットワークに接続できます。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 最新の注意事項と機能情報については、プラットフォームおよびソフトウェア リリースのバグ検索ツールとリリース ノートを参照してください。 このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表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シスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

IPv6 自動 6to4 トンネルに関する情報

自動 6to4 トンネル

自動 6to4 トンネルを使用すると、孤立した IPv6 ドメインを、IPv4 ネットワークを介してリモート IPv6 ネットワークに接続できます。 自動 6to4 トンネルと、手動で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点です。 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非ブロードキャストマルチアクセス（NBMA）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た IPv4 アドレスは、自動トンネルのもう一方のエンドを検出するために使用されます。

自動 6to4 トンネルは、孤立した IPv6 ネットワーク内の境界ルータに設定できます。これにより、IPv4 インフラストラクチャを介した別の IPv6 ネットワーク内の境界ルータへのパケット単位のトンネルが作成されます。 トンネル宛先は、プレフィックス 2002::/16 で始まる IPv6 アドレス（形式は 2002:*border-router-IPv4-address* ::/48）から抽出される、境界ルータの IPv4 アドレスによって決定されます。 埋め込まれた IPv4 アドレスのあとには、サイト内のネットワークへの番号付けに使用できる 16 ビットが続きます。 6to4 トンネルの両端の境界ルータは、IPv4 プロトコル スタックと IPv6 プロトコル スタックの両方をサポートしている必要があります。 6to4 トンネルは、境界ルータ間または境界ルータとホスト間に設定されます。

6to4 トンネルの最も単純な展開シナリオは、複数の IPv6 サイトを相互接続することです。各 IPv6 サイトには、共有 IPv4 ネットワークへの 1 つ以上の接続があります。 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場合があります。 主な要件は、各サイトがグローバルに一意な IPv4 アドレスを持っていることです。Cisco ソフトウェアでは、このアドレスを使用して、グローバルに一意な 6to4/48 IPv6 プレフィックスを構成します。 他のトンネリング メカニズムと同様に、ホスト名を IPv4 と IPv6 両方の IP アドレスにマッピングするドメイン ネーム システム（DNS）によって、アプリケーションは必要なアドレスを選択できます。

IPv6 自動 6to4 トンネルの設定方法

自動 6to4 トンネルの設定

はじめる前に

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、プレフィックス 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 トンネルの両端の境界ルータは、IPv4 プロトコル スタックと IPv6 プロトコル スタックの両方をサポートしている必要があります。

（注） IPv4 互換トンネル 1 つだけの設定、および 6to4 IPv6 トンネル 1 つだけの設定が、1 台のルータ上でサポートされます。 同じルータ上でこれら両方のトンネル タイプを設定する場合は、これらのタイプが同じトンネル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 互換トンネルがインターフェイスを共有できない理由は、両方が NBMA「ポイントツーマルチポイント」アクセス リンクであり、多重化パケット ストリームからのパケットを着信インターフェイスの単一パケット ストリームに整理するにはトンネル送信元だけを使用できる点です。 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 互換トンネルの両方が同じ送信元インターフェイスを共有する場合、ルータは、着信パケットの割り当て先となる IPv6 トンネル インターフェイスを特定できません。

手動で設定された IPv6 トンネルの場合、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 宛先が両方とも定義されているため、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-address* / *prefix-length* | *prefix-name sub-bits*/*prefix-length*
5. **tunnel source** {*ip-address*| *interface-t ype interface-number*}
6. **tunnel mode ipv6ip** [**6rd** | **6to4** | **auto-tunnel** | **isatap**
7. **exit**
8. **ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix* / *prefix-length*{*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*]

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。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
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 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-address* / *prefix-length* | *prefix-name sub-bits*/*prefix-length* 例： `Router(config-if)# ipv6 address 3ffe:b00:c18:1::3/127`	インターフェイスに割り当てられている IPv6 ネットワークを指定し、インターフェイスで IPv6 処理をイネーブルにします。
ステップ 5	**tunnel source** {*ip-address*| *interface-t ype interface-number*} 例： `Router(config-if)# tunnel source loopback 1`	トンネル インターフェイスの送信元インターフェイスのタイプおよび番号を指定します。
ステップ 6	**tunnel mode ipv6ip** [**6rd** | **6to4** | **auto-tunnel** | **isatap** 例： `Router(config-if)# tunnel mode ipv6ip 6rd`	スタティック IPv6 トンネル インターフェイスを設定します。 • **auto-tunnel** キーワードは、Cisco ASR 1000 シリーズ ルータではサポートされません。
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 モードを開始します。
ステップ 8	**ipv6 route** [**vrf** *vrf-name*] *ipv6-prefix* / *prefix-length*{*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*]	指定したトンネル インターフェイスに IPv6 6to4 プレフィックス 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場合は、6to4 トンネル インターフェイスに IPv6 6to4 プレフィックス 2002::/16 のスタティック ルートを設定する必要があります。

	コマンドまたはアクション	目的
	[*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*] 例： `Router(config)# ipv6 route 2002::/16 tunnel 0`	• **ipv6 route** コマンドで指定したトンネル番号は、**interface tunnel** コマンドで指定したトンネル番号と同じである必要があります。

IPv6 自動 6to4 トンネルの設定例

例：6to4 トンネルの設定

次の例では、孤立した IPv6 ネットワーク内の境界ルータ上に 6to4 トンネルを設定します。 IPv4 アドレスは 192.168.99.1 であり、IPv6 プレフィックス 2002:c0a8:6301::/48 に変換されます。 IPv6 プレフィックスは、トンネル インターフェイス用として 2002:c0a8:6301::/64 にサブネット化されます。つまり、最初の IPv6 ネットワークは 2002:c0a8:6301:1::/64、2 番めの IPv6 ネットワークは 2002:c0a8:6301:2::/64 になります。 スタティック ルートによって、IPv6 プレフィックス 2002::/16 のその他のすべてのトラフィックは、自動トンネリングのためにトンネル インターフェイス 0 に送信されます。

```
interface GigabitEthernet0/0/0
 description IPv4 uplink
 ip address 192.168.99.1 255.255.255.0
!
interface GigabitEthernet1/0/0
 description IPv6 local network 1
 ipv6 address 2002:c0a8:6301:1::1/64
!
interface GigabitEthernet2/0/0
 description IPv6 local network 2
 ipv6 address 2002:c0a8:6301:2::1/64
!
interface Tunnel0
 description IPv6 uplink
 no ip address
 ipv6 address 2002:c0a8:6301::1/64
 tunnel source GigabitEthernet0/0/0
 tunnel mode ipv6ip 6to4
!
ipv6 route 2002::/16 tunnel 0
```

その他の関連資料

関連資料

関連項目	マニュアル タイトル
IPv6 アドレッシングと接続	『*IPv6 Configuration Guide*』
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
IPv6 コマンド	『Cisco IOS IPv6 Command Reference』
Cisco IOS IPv6 機能	『Cisco IOS IPv6 Feature Mapping』

標準および RFC

標準/RFC	タイトル
IPv6 に関する RFC	『*IPv6 RFCs*』

MIB

MIB	MIB のリンク
	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 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/cisco/web/support/index.html

IPv6 自動 6to4 トンネルの機能情報

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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

表 *16*：*IPv6* 自動 *6to4* トンネルの機能情報

機能名	リリース	機能情報
IPv6 トンネリング：自動 6to4 トンネル	Cisco IOS XE Release 2.1	自動 6to4 トンネルを使用すると、孤立した IPv6 ドメインを、IPv4 ネットワークを介してリモート IPv6 ネットワークに接続できます。 **tunnel mode ipv6ip**、**tunnel source** の各コマンドが導入または変更されました。

第 15 章

IPv4 GRE トンネルを介する IPv6

GRE トンネルは、2 つのポイント間のリンクであり、リンクごとに個別のトンネルがあります。これらのトンネルは、特定のパッセンジャまたはトランスポート プロトコルに結合されていませんが、この場合、GRE を使用するパッセンジャ プロトコルとして IPv6 を伝送し、トランスポート プロトコルとして IPv4 または IPv6 を伝送します。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 最新の注意事項と機能情報については、プラットフォームおよびソフトウェア リリースのバグ検索ツールとリリース ノートを参照してください。 このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表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シスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

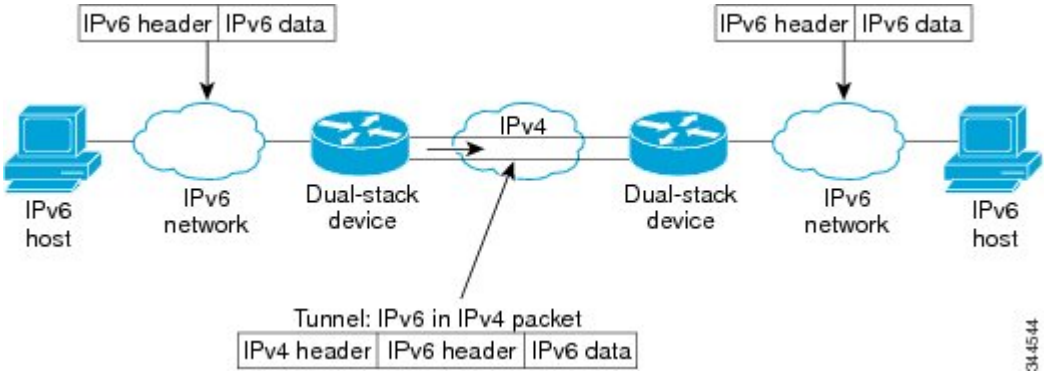
IPv4 GRE トンネルを介する IPv6 に関する情報

オーバーレイ トンネル for IPv6

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以下の図）へ伝送します オーバーレイ トンネルを使用することで、孤立した IPv6 ネットワークと通信できます。このとき、孤立した複数の IPv6 ネットワーク間にある IPv4 インフラストラクチャをアップグレードする必要はありません。 オーバーレイ トンネルは、境界デバイス間、または境界デバイスとホスト間に設定できますが、両方のエンドポイントが IPv4 プロトコル スタックと IPv6 プロトコル スタックの両方をサポートしている必要があります。 IPv6 では、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。

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

図 7: 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最大伝送単位（MTU）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 と IPv6 の両方のプロトコル スタック、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への移行方法と見なす必要があります。

以下の表は、IPv4 ネットワーク上での IPv6 パケットの伝送にどのトンネル タイプを設定すればよいかを決定する場合に役立ちます。

表 *17*：*IPv4* ネットワーク上で *IPv6* パケットを伝送するトンネル タイプの推奨される使用方法

トンネリング タイプ	推奨される使用方法	使用方法
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	IPv6 パケットだけを伝送できます。
GRE および IPv4 互換	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	IPv6、コネクションレス型ネットワーク サービス（CLNS）、およびその他の多数のタイプのパケットを伝送できます。
IPv4- 互換機	ポイントツーマルチポイント トンネル	::/96 プレフィックスを使用します。 このトンネル タイプの使用は推奨しません。
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	サイトでは、2002::/16 プレフィックスからのアドレスを使用します。
6RD	IPv6 サービスは、IPv4 に IPv6 のカプセル化を使用することで IPv4 ネットワーク上のユーザに提供されます。	プレフィックスは、SP 自身のアドレス ブロックから割り当てることができます。
ISATAP	サイト内のシステム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	サイトでは、任意の IPv6 ユニキャスト アドレスを使用できます。

個々のトンネルタイプについて、このマニュアルで詳しく説明しています。 実装する特定のトンネル タイプに関する情報を確認および理解することを推奨します。 必要なトンネル タイプに精

通している場合は、以下の表で、有用と思われるトンネル設定パラメータの概要を参照してください。

表 *18*：トンネリング タイプ別のトンネル設定パラメータ

トンネリングタイプ	トンネル設定パラメータ			
トンネル モード	トンネルの送信元	トンネルの宛先	インターフェイスプレフィックスまたはアドレス	
手動	ipv6ip	IPv4アドレス、またはIPv4が設定されたインターフェイスへの参照。	IPv4 アドレス。	IPv6 アドレス。
GRE/IPv4	gre ip		IPv4 アドレス。	IPv6 アドレス。
IPv4- 互換機	ipv6ip auto-tunnel		不要。これらはすべて、ポイントツーマルチポイントのトンネリングタイプです。IPv4宛先アドレスは、パケット単位で、IPv6宛先から計算されます。	不要。インターフェイスアドレスは、*::tunnel-source*/96として生成されます。
6to4	ipv6ip 6to4			IPv6 アドレス。プレフィックスは、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。
6RD	ipv6ip 6rd			IPv6 アドレス。
ISATAP	ipv6ip isatap			変更された eui-64 形式での IPv6 プレフィックス。IPv6アドレスは、プレフィックスおよびトンネル送信元 IPv4 アドレスから生成されます。

IPv6 トラフィック用の GRE IPv4 トンネル 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。 GRE トンネルは、手動で設定された IPv6 トンネルと同様、リンクごとに個別のトンネルが設定された 2 つのポイント間のリンクです。 これらのトンネルは、特定のパッセンジャまたはトランスポート プロトコルに結合されていませんが、この場合、GRE を使用するパッセンジャ プロトコルとして IPv6 を伝送し、トランスポート プロトコルとして IPv4 または IPv6 を伝送します。

GRE トンネルは、2 つのエッジ デバイス間またはエッジ デバイスとエンド システム間に定期的でセキュアな通信を必要とする安定した接続のために主に使用されます。 エッジデバイスとエンド システムは、デュアル スタック実装である必要があります。

IPv4 GRE トンネルを介した IPv6 の実装方法

GRE/IPv6 トンネルの設定

GRE トンネルは、IPv6 ネットワーク層を介して送出し、IPv6 トンネルで IPv4 パケットと IPv6 パケットを転送するように設定できます。

はじめる前に

GRE IPv6 トンネルが設定されている場合、IPv6 アドレスは、トンネル送信元およびトンネル宛先に割り当てられます。 トンネル インターフェイスは、割り当て済みの IPv4 アドレスまたは IPv6 アドレスを持つことができます（ここでは説明していません）。 設定されたトンネルの両端にあるホストまたはデバイスは、IPv4 プロトコル スタックと IPv6 プロトコル スタックの両方をサポートしている必要があります。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. 次のいずれかのコマンドを入力します。
   - **ipv6 address** {*ipv6-address*/*prefix-length* | *prefix-name sub-bits*/*prefix-length*}
   - **ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**]
5. **tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*}
6. **tunnel destination** {*hostname* | *ip-address* | *ipv6-address*}
7. **tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **gre multipoint** | **gre ipv6** | **ipip** [**decapsulate-any**] | **iptalk** | **ipv6** | **mpls** | **nos**}
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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • 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 tunnel** *tunnel-number* 例： `Device(config)# interface tunnel 0`	トンネル 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	次のいずれかのコマンドを入力します。 • **ipv6 address** {*ipv6-address*/*prefix-length*	*prefix-name sub-bits*/*prefix-length*} • **ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**]	インターフェイスに割り当てられている IPv6 ネットワークを指定し、インターフェイスで IPv6 処理をイネーブルにします。 • **eui-64** キーワードを指定すると、ソフトウェアは、インターフェイスの IPv6 アドレスを設定し、アドレスの下位 64 ビットの EUI-64 インターフェイス

	コマンドまたはアクション	目的
	例： `Device(config-if)# ipv6 address 3ffe:b00:c18:1::3/127`	ID を使用してインターフェイスで IPv6 処理を有効にします。
ステップ 5	**tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*} 例： `Device(config-if)# tunnel source gigabitethernet 0/0/0`	送信元 IPv4 アドレス、IPv6 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されている必要があります。
ステップ 6	**tunnel destination** {*hostname* | *ip-address* | *ipv6-address*} 例： `Device(config-if)# tunnel destination 2001:DB8:1111:2222::1/64`	トンネル インターフェイスの宛先 IPv4 アドレス、IPv6 アドレスまたはホスト名を指定します。
ステップ 7	**tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **gre multipoint** | **gre ipv6** | **ipip** [**decapsulate-any**] | **iptalk** | **ipv6** | **mpls** | **nos**} 例： `Device(config-if)# tunnel mode gre ipv6`	GRE IPv6 トンネルを指定します。 （注） **tunnel mode gre ipv6** コマンドでは、GRE をトンネルのカプセル化プロトコルとして指定します。
ステップ 8	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

IPv4 GRE トンネルを介した IPv6 の設定例

IS-IS および IPv6 トラフィックを実行する GRE トンネルの例

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 の設定

```
ipv6 unicast-routing
clns routing
!
interface tunnel 0
 no ip address
 ipv6 address 3ffe:b00:c18:1::3/127
 ipv6 router isis
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 2001:DB8:1111:2222::1/64
 tunnel mode gre ipv6
!
interface GigabitEthernet0/0/0
 ip address 10.0.0.1 255.255.255.0
!
router isis
 net 49.0000.0000.000a.00
```

ルータ B の設定

```
ipv6 unicast-routing
clns routing
!
interface tunnel 0
 no ip address
 ipv6 address 3ffe:b00:c18:1::2/127
 ipv6 router isis
 tunnel source GigabitEthernet 0/0/0
 tunnel destination 2001:DB8:1111:2222::2/64
 tunnel mode gre ipv6
!
interface GigabitEthernet0/0/0
 ip address 10.0.0.2 255.255.255.0
!
router isis
 net 49.0000.0000.000b.00
 address-family ipv6
 redistribute static
 exit-address-family
```

例：IPv6 トンネルのトンネル宛先アドレス

```
Router(config
)
#
interface Tunnel0
Router(config
-if)
#
ipv6 address 2001:1:1::1/48
Router(config
-if)
#
tunnel source GigabitEthernet 0/0/0
Router(config
-if)
#
tunnel destination 10.0.0.2
Router(config
-if)
#
tunnel mode gre ipv6
Router(config
-if)
```

```
#
exit
!
Router(config
)
#
interface GigabitEthernet0/0/0
Router(config
-if)
#
ip address 10.0.0.1 255.255.255.0
Router(config
-if)
#
exit
!
Router(config
)
#
ipv6 unicast-routing
Router(config
)
#
router isis

Router(config
)
#
net 49.0000.0000.000a.00
```

その他の関連資料

関連資料

関連項目	マニュアル タイトル
IPv6 アドレッシングと接続	『*IPv6 Configuration Guide*』
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
IPv6 コマンド	『Cisco IOS IPv6 Command Reference』
Cisco IOS IPv6 機能	『Cisco IOS IPv6 Feature Mapping』

標準および RFC

標準/RFC	タイトル
IPv6 に関する RFC	『*IPv6 RFCs*』

MIB

MIB	MIB のリンク
	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 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/cisco/web/support/index.html

IPv4 GRE トンネルを介する IPv6 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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

表 *19* : *IPv4 GRE* トンネルを介する *IPv6* の機能情報

機能名	リリース	機能情報
IPv4 GRE トンネルを介する IPv6	Cisco IOS XE Release 2.1	GRE トンネルは、2 つのポイント間のリンクであり、リンクごとに個別のトンネルがあります。 これらのトンネルは、特定のパッセンジャまたはトランスポート プロトコルに結合されていませんが、この場合、GRE を使用するパッセンジャ プロトコルとして IPv6 を伝送し、トランスポート プロトコルとして IPv4 または IPv6 を伝送します。 **tunnel destination**、**tunnel mode ipv6ip**、**tunnel source** の各コマンドが導入または変更されました。

第 16 章

GRE IPv6 トンネル

GRE IPv6 トンネル機能は、他のプロトコルから IPv6 ネットワークを介したパケット配信を有効にして、グローバルにルーティングされた IPv6 アドレスのパブリック ネットワークを介したプライベート ネットワーク間で、IPv6 パケットのルーティングが可能になります。 総称ルーティング カプセル化（GRE）は、ブロードキャストおよびマルチキャスト トラフィック（マルチキャスト ストリーミングまたはルーティング プロトコル）をカプセル化するメリットを提供するユニキャスト プロトコルであるか、その他の非 IP プロトコルで、IPsec によって保護されています。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 最新の注意事項と機能情報については、プラットフォームおよびソフトウェア リリースのバグ検索ツールとリリース ノートを参照してください。 このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表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シスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

GRE IPv6 トンネルの制約事項

- GRE トンネル キープアライブ パケットはサポートされません。
- マルチポイント GRE（mGRE）IPv6 トンネリングはサポートされていません。
- 仮想ルーティングおよび転送（VRF）ではトンネル トランスポートのサポートが限られています。 VRF での限定サポートは、トンネル保護を使用しない Ipv6 ポイントツーポイント GRE に適用できます。

GRE IPv6 トンネルに関する情報

GRE IPv6 トンネルの概要

GRE IPv6 トンネル機能は、他のプロトコルから IPv6 ネットワークを介したパケット配信を有効にして、グローバルにルーティングされたIPv6アドレスのパブリック ネットワークを介したプライベート ネットワーク間で、IPv6 パケットのルーティングが可能になります。

ポイントツーポイント GRE トンネルでは、各トンネル インターフェイスは、設定時にトンネル送信元IPv6アドレスおよびトンネル宛先のIPv6アドレスを必要とします。すべてのパケットは、外部 IPv6 ヘッダーと GRE ヘッダーでカプセル化されます。

GRE IPv6 トンネル保護

GRE IPv6 トンネル保護により、デバイスをセキュリティ ゲートウェイとして動作させ、他のセキュリティ ゲートウェイ デバイス間に IPsec トンネルを確立し、トラフィックがパブリック IPv6 インターネットから送信される場合に内部ネットワークからのトラフィックに対する暗号化IPsec保護を提供できます。 GRE IPv6 トンネル保護機能は、IPv4 GRE トンネル保護を使用したセキュリティ ゲートウェイ モデルと似ています。

GRE IPv6 トンネルの設定方法

GRE IPv6 トンネルの設定

IPv6ネットワーク上でGREトンネルを設定するには、次の作業を実行します。 GREトンネルは、IPv6 ネットワーク層を介して送出し、IPv6 トンネルを介して IPv6 パケットと IPv4 パケットを転送するように設定できます。

はじめる前に

GRE IPv6 トンネルが設定されている場合、IPv6 アドレスは、トンネル送信元およびトンネル宛先に割り当てられます。 トンネル インターフェイスは、IPv4 または IPv6 アドレスのいずれかにすることができます（このことは、以降の作業では示されていません）。 設定されたトンネルの両端にあるホストまたはデバイスは、IPv4 プロトコル スタックと IPv6 プロトコル スタックの両方をサポートしている必要があります。

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. **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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • 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 tunnel** *tunnel-number* 例： `Device(config)# interface tunnel 0`	トンネル 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： `Device(config-if)# tunnel source ethernet 0`	送信元 IPv6 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。 • インターフェイスのタイプと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、『IPv6 Command Reference』を参照してください。

	コマンドまたはアクション	目的
ステップ 5	**tunnel destination** *ipv6-address* 例： `Device(config-if)# tunnel destination 2001:0DB8:0C18:2::300`	トンネル インターフェイスの宛先 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、『IPv6 Command Reference』を参照してください。
ステップ 6	**tunnel mode gre ipv6** 例： `Device(config-if)# tunnel mode gre ipv6`	GRE IPv6 トンネルを指定します。 （注） **tunnel mode gre ipv6** コマンドでは、GRE をトンネル インターフェイスのカプセル化プロトコルとして指定します。 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、『IPv6 Command Reference』を参照してください。
ステップ 7	**end** 例： `Device(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。

GRE IPv6 トンネル保護設定

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. **tunnel protection ipsec profile** *profile-name*
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： `Device> enable`	特権 EXEC モードをイネーブルにします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
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 tunnel** *tunnel-number* 例： `Device(config)# interface tunnel 0`	トンネル 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： `Device(config-if)# tunnel source ethernet 0`	送信元 IPv6 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。 • インターフェイスのタイプと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、『IPv6 Command Reference』を参照してください。
ステップ 5	**tunnel destination** *ipv6-address* 例： `Device(config-if)# tunnel destination 2001:0DB8:0C18:2::300`	トンネル インターフェイスの宛先 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、『IPv6 Command Reference』を参照してください。
ステップ 6	**tunnel mode gre ipv6** 例： `Device(config-if)# tunnel mode gre ipv6`	GRE IPv6 トンネルを指定します。 （注） **tunnel mode gre ipv6** コマンドでは、GRE をトンネル インターフェイスのカプセル化プロトコルとして指定します。 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、『IPv6 Command Reference』を参照してください。
ステップ 7	**tunnel protection ipsec profile** *profile-name* 例： `Device(config-if)# tunnel protection ipsec profile ipsec-profile`	トンネル インターフェイスを IPsec プロファイルに関連付けます。 （注） *profile-name* 引数の場合、グローバル コンフィギュレーション モードで設定された IPsec プロファイルを指定します。
ステップ 8	**end** 例： `Device(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。

GRE IPv6 トンネルの設定例

例：GRE IPv6 トンネルの設定

IPv6 トランスポートで GRE トンネルを設定する方法の例を次に示します。 この例では、イーサネット 0/0 は IPv6 アドレスを備えており、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 Tunnel0
 ip address 10.1.1.1 255.255.255.0
 ip router isis
 tunnel source Ethernet0/0
 tunnel destination 2001:DB8:1111:2222::1
 tunnel mode gre ipv6
!
interface Ethernet0/0
 no ip address
 ipv6 address 2001:DB8:1111:1111::1/64
!
router isis
 net 49.0001.0000.0000.000a.00
```

例：GRE IPv6 トンネル保護の設定

次に、GRE IPv6 トンネル インターフェイスに IPsec プロファイル「ipsec-profile」を関連付ける方法の例を示します。 IPSec プロファイルは、**crypto ipsec profile** コマンドを使用して設定します。

```
crypto ipsec profile ipsec-profile
 set transform-set ipsec-profile
!
interface Tunnel1
 ip address 192.168.1.1 255.255.255.252
 tunnel source FastEthernet2/0
 tunnel destination 10.13.7.67
 tunnel protection ipsec profile ipsec-profile
```

その他の関連資料

関連資料

関連項目	マニュアル タイトル
Cisco IOS コマンド	『Master Commands List, All Releases』

関連項目	マニュアル タイトル
トンネル コマンド：コマンド構文の詳細、コマンド モード、デフォルト設定、コマンド履歴、使用に関する注意事項および例	『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、デフォルト設定、コマンド履歴、使用に関する注意事項および例	『IPv6 Command Reference』

シスコのテクニカル サポート

説明	リンク
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 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/cisco/web/support/index.html

GRE IPv6 トンネルの機能情報

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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

表 *20*：*GRE IPv6* トンネルの機能情報

機能名	リリース	機能情報
GRE IPv6 トンネル	Cisco IOS XE Release 3.7S	GRE IPv6 トンネル機能は、他のプロトコルから IPv6 ネットワークを介したパケット配信を有効にして、グローバルにルーティングされた IPv6 アドレスのパブリック ネットワークを介したプライベートネットワーク間で、IPv6 パケットのルーティングが可能になります。

第 17 章

IPv6 用の ISATAP トンネル サポート

ISATAP は、基礎となる IPv4 ネットワークを IPv6 の 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。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 最新の注意事項と機能情報については、プラットフォームおよびソフトウェア リリースのバグ検索ツールとリリース ノートを参照してください。 このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表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シスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

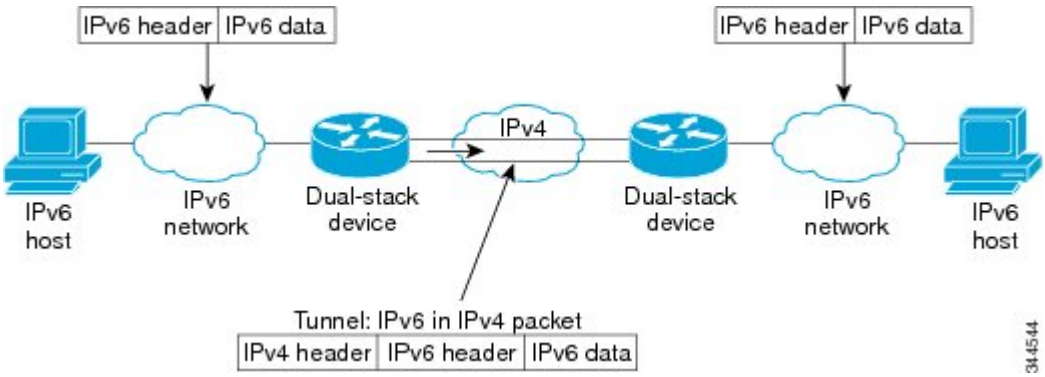
IPv6 用の ISATAP トンネル サポートに関する情報

オーバーレイ トンネル for IPv6

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以下の図）へ伝送します オーバーレイ トンネルを使用することで、孤立した IPv6 ネットワークと通信できます。このとき、孤立した複数の IPv6 ネットワーク間にある IPv4 インフラストラクチャをアップグレードする必要はありません。オーバーレイ トンネルは、境界デバイス間、または境界デバイスとホスト間に設定できますが、両方のエンドポイントが IPv4 プロトコル スタックと IPv6 プロトコル スタックの両方をサポートしている必要があります。 IPv6 では、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。

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

図 *8* : 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最大伝送単位（MTU）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 と IPv6 の両方のプロトコル スタック、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への移行方法と見なす必要があります。

以下の表は、IPv4 ネットワーク上での IPv6 パケットの伝送にどのトンネル タイプを設定すればよいかを決定する場合に役立ちます。

表 *21*：*IPv4* ネットワーク上で *IPv6* パケットを伝送するトンネル タイプの推奨される使用方法

トンネリング タイプ	推奨される使用方法	使用方法
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	IPv6 パケットだけを伝送できます。
GRE および IPv4 互換	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	IPv6、コネクションレス型ネットワーク サービス（CLNS）、およびその他の多数のタイプのパケットを伝送できます。
IPv4- 互換機	ポイントツーマルチポイント トンネル	::/96 プレフィックスを使用します。 このトンネル タイプの使用は推奨しません。
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	サイトでは、2002::/16 プレフィックスからのアドレスを使用します。
6RD	IPv6 サービスは、IPv4 に IPv6 のカプセル化を使用することで IPv4 ネットワーク上のユーザに提供されます。	プレフィックスは、SP 自身のアドレス ブロックから割り当てることができます。
ISATAP	サイト内のシステム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	サイトでは、任意の IPv6 ユニキャスト アドレスを使用できます。

個々のトンネルタイプについて、このマニュアルで詳しく説明しています。 実装する特定のトンネル タイプに関する情報を確認および理解することを推奨します。 必要なトンネル タイプに精

通している場合は、以下の表で、有用と思われるトンネル設定パラメータの概要を参照してください。

表 *22*：トンネリング タイプ別のトンネル設定パラメータ

トンネリング タイプ	トンネル設定パラメータ			
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 プレフィックスまたはアドレス	
手動	ipv6ip	IPv4 アドレス、または IPv4 が設定されたインターフェイスへの参照。	IPv4 アドレス。	IPv6 アドレス。
GRE/IPv4	gre ip		IPv4 アドレス。	IPv6 アドレス。
IPv4- 互換機	ipv6ip auto-tunnel		不要。これらはすべて、ポイントツーマルチポイントのトンネリング タイプです。IPv4 宛先アドレスは、パケット単位で、IPv6 宛先から計算されます。	不要。インターフェイス アドレスは、*::tunnel-source*/96 として生成されます。
6to4	ipv6ip 6to4			IPv6 アドレス。プレフィックスは、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。
6RD	ipv6ip 6rd			IPv6 アドレス。
ISATAP	ipv6ip isatap			変更された eui-64 形式での IPv6 プレフィックス。IPv6 アドレスは、プレフィックスおよびトンネル送信元 IPv4 アドレスから生成されます。

ISATAP トンネル

ISATAP は、基礎となる IPv4 ネットワークを IPv6 の 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（スパース 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 トンネルを使用すると、サイト内の個々の IPv4 または IPv6 デュアル スタック ホストは、基本的には 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することで、同じ仮想リンク上のこうした他のホストと通信できます。

ISATAP ルータは、標準のルータ アドバタイズメント ネットワーク設定サポートを ISATAP サイトに提供します。 この機能によって、クライアントは、ギガビットイーサネットまたはファストイーサネットに接続されている場合と同様に、クライアント自身を自動的に設定できます。 また、サイト外の接続を提供するように設定することもできます。 ISATAP では、リンク ローカルまたはグローバル（6to4 プレフィックスを含む）な任意のユニキャスト IPv6 プレフィックス（/64）で構成される、適切に定義された IPv6 アドレス形式を使用します。これにより、IPv6 ルーティングをローカルに、またはインターネット上で実行できます。 IPv4 アドレスは、IPv6 アドレスの最後の 32 ビットに符号化され、自動 IPv6-in-IPv4 トンネリングを可能にします。

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 では、64 ビットの IPv6 プレフィックスおよび 64 ビットのインターフェイス ID が含まれているユニキャスト アドレスを使用します。 インターフェイス ID は、アドレスが IPv6 ISATAP アドレスであることを示すために最初の 32 ビットに値 000:5EFE が含まれる、変更された EUI-64 形式で作成されます。 次の表に、ISATAP アドレス形式を示します。

表 *23*：*IPv6 ISATAP* のアドレス形式

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 プレフィックス	0000:5EFE	ISATAP リンクの IPv4 アドレス

上記の表に示すように、ISATAP アドレスは、IPv6 プレフィックスと ISATAP インターフェイス ID で構成されます。 インターフェイス ID には、基礎となる IPv4 リンクの IPv4 アドレスが含まれています。 次の例では、プレフィックスが 2001:DB8:1234:5678::/64 で、埋め込まれた IPv4 アドレスが 10.173.129.8 である場合、実際の ISATAP アドレスがどのようになるかを示します。 ISATAP アドレスでは、IPv4 アドレスは 16 進数の 0AAD:8108 として表現されます。

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

IPv6 用の ISATAP トンネル サポートの設定方法

ISATAP トンネルの設定

はじめる前に

ISATAP トンネルの設定で使用される **tunnel source** コマンドは、設定済みの IPv4 アドレスを持つインターフェイスをポイントする必要があります。 アドバタイズされた ISATAP IPv6 アドレスおよび（1 つまたは複数の）プレフィックスは、ネイティブ IPv6 インターフェイス用として設定されます。 IPv6 トンネル インターフェイスは、インターフェイス ID 内の最後の 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、変更された EUI-64 アドレスを使用して設定されている必要があります。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **ipv6 address** {*ipv6-address / prefix-length* | *prefix-name sub-bits*/*prefix-length*
5. **no ipv6 nd ra suppress**
6. **tunnel source** {*ip-address*| *interface-type interface-number*}
7. **tunnel mode ipv6ip** [**6rd** | **6to4** | **auto-tunnel** | **isatap**

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。 • 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： `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 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-address* / *prefix-length* | *prefix-name sub-bits*/*prefix-length* 例： `Router(config-if)# ipv6 address 2001:DB8:6301::/64 eui-64`	インターフェイスに割り当てられた 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。
ステップ 5	**no ipv6 nd ra suppress** 例： `Router(config-if)# no ipv6 nd ra suppress`	IPv6 ルータ アドバタイズメントの送信は、トンネル インターフェイス上ではデフォルトでディセーブルになっています。このコマンドによって、IPv6 ルータ アドバタイズメントの送信が再度イネーブルになり、クライアントの自動設定が可能になります。
ステップ 6	**tunnel source** {*ip-address*| *interface-type interface-number*} 例： `Router(config-if)# tunnel source gigabitethernet 1/0/1`	トンネル インターフェイスの送信元インターフェイスのタイプおよび番号を指定します。 （注） **tunnel source** コマンドで指定した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必要があります。
ステップ 7	**tunnel mode ipv6ip** [**6rd** | **6to4** | **auto-tunnel** | **isatap** 例： `Router(config-if)# tunnel mode ipv6ip isatap`	ISATAP アドレスを使用する IPv6 オーバーレイ トンネルを指定します。 • **auto-tunnel** キーワードは、Cisco ASR 1000 シリーズ ルータではサポートされません。

IPv6 に対する ISATAP トンネル サポートの設定例

例：ISATAP トンネルの設定

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。クライアントの自動設定を可能にするために、ルータ アドバタイズメントがイネーブルになっています。

```
ipv6 unicast-routing
interface tunnel 1
 tunnel source Gigabitethernet 0/0/0
 tunnel mode ipv6ip isatap
 ipv6 address 2001:DB8::/64 eui-64
 no ipv6 nd ra suppress
 exit
```

その他の関連資料

関連資料

関連項目	マニュアル タイトル
IPv6 アドレッシングと接続	『*Cisco IOS IPv6 Configuration Guide*』
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』
IPv6 コマンド	『Cisco IOS IPv6 Command Reference』
Cisco IOS IPv6 機能	『Cisco IOS IPv6 Feature Mapping』

標準/RFC	タイトル
IPv6 に関する RFC	『*IPv6 RFCs*』

標準および RFC

シスコのテクニカル サポート

説明	リンク
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 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/cisco/web/support/index.html

IPv6 に対する ISATAP トンネル サポートの機能情報

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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

表 *24*：*IPv6* に対する *ISATAP* トンネル サポートの機能情報

機能名	リリース	機能情報
IPv6 用の ISATAP トンネル サポート	Cisco IOS XE Release 2.1	ISATAP は、基礎となる IPv4 ネットワークを IPv6 の 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。 **ipv6 nd ra suppress**、**tunnel mode ipv6ip**、**tunnel source** の各コマンドが導入または変更されました。

第 18 章

VRF-Aware トンネル

仮想ルーティングおよびフォワーディング（VRF）Aware トンネルは、信頼できないコア ネットワークまたは別のインフラストラクチャ（IPv4 または IPv6）を備えたコア ネットワークで区切られたカスタマー ネットワークに接続するために使用されます。

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。 最新の注意事項と機能情報については、プラットフォームおよびソフトウェア リリースのバグ検索ツールとリリース ノートを参照してください。 このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表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シスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

VRF-Aware トンネルの前提条件

- カスタマーエッジネットワークを設定する必要があります。「トンネリング用のカスタマーエッジ ネットワークの設定」の項を参照してください。
- カスタマーを設定し、VRF を転送する必要があります。「VRF インスタンスの定義」の項を参照してください。

VRF-Aware トンネルに関する情報

トンネルの IP 送信元および宛先の VRF メンバーシップ

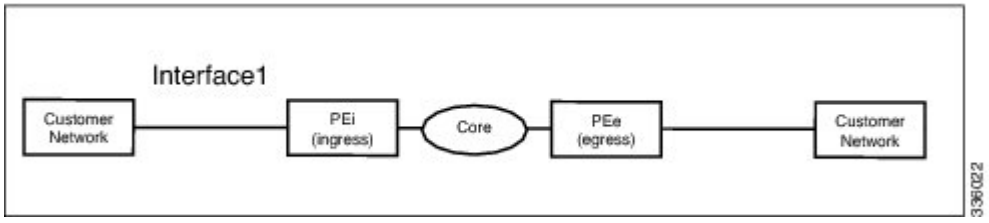
任意の VPN ルーティング/転送（VRF）テーブルに属するようにトンネルの送信元と宛先を設定できます。 VRF テーブルには、各 VPN のルーティング データが保管されます。 VRF テーブルでは、ネットワーク アクセス サーバ（NAS）に接続されているカスタマー サイトの VPN メンバーシップを定義します。 各 VRF テーブルは、IP ルーティング テーブル、派生したシスコ エクスプレス フォワーディング テーブル、およびルーティング テーブルに含まれる情報を制御するガイドラインおよびルーティング プロトコル パラメータから構成されます。

任意の VRF またはグローバル テーブルに属するようにトンネルの送信元と宛先を設定できます。トンネルは、トンネルの宛先へのルートが定義されていない場合は無効になります。

VRF-Aware トンネル

仮想ルーティングおよびフォワーディング（VRF）Aware トンネルは、信頼できない IPv4 コア ネットワークまたは IPv6 コア ネットワークで区切られたカスタマー ネットワークに接続するために使用されます。

図 *9*：*VRF-Aware* トンネル



上記のトポロジでは、トンネルがコア ネットワークに設定されます。プロバイダーエッジ（PE）デバイス PEi は、インターフェイス 1 に到着したパケットのトンネル ヘッドです。 PE デバイス PEe は、インターフェイス 1 に到着したパケットのトンネル テールです。

インターフェイス 1 に設定された VRF はカスタマー VRF です。 インターフェイス 1 を介して到着したパケットは、この VRF を使用してルーティングされます。 トンネルから出たパケットはこの VRF に転送されます。 カスタマー VRF によるルーティングは、内部 IP パケット ルーティングと呼ばれます。

tunnel vrf コマンドを使用して設定された VRF はトランスポート VRF です。 トランスポート VRF は、カプセル化されたペイロードに適用され、トンネル エンドポイントを調べるために使用される VRF です。 この VRF は、トンネルがパケットを送信する際に経由する物理インターフェイスに関連付けられている VRF と同じです。 トランスポート VRF によるルーティングは、外部 IP パケット ルーティングと呼ばれます。

トンネル エンドポイントは、グローバル ルーティング テーブルからのアドレスか、設定済みのトランスポート VRF テーブルからのアドレスとして設定できます。

IPv6 トンネルを介した VRF-Aware IPv6

仮想ルーティングおよび転送（VRF）Aware IPv6 トンネルを信頼できない IPv6 インフラストラクチャ内に作成することにより、このインフラストラクチャ上に IPv6 パケットを転送することができます。 これらのトンネルは、VRF テーブルまたはグローバル ルーティング テーブルにエンドポイントを持つことができます。 使用されるトンネルのモードは、**tunnel mode gre ipv6** と **tunnel mode ipv6** です。

IPv6 トンネルを介した VRF-Aware IPv4

仮想ルーティングおよび転送（VRF）Aware IPv4 トンネルを信頼できない IPv6 インフラストラクチャ内に作成することにより、このインフラストラクチャ上に IPv4 パケットを転送することができます。 これらのトンネルは、VRF テーブルまたはグローバル ルーティング テーブルにエンドポイントを持つことができます。 使用されるトンネルのモードは、**tunnel mode gre ipv6** と **tunnel mode ipv6** です。

IPv4 トンネルを介した VRF-Aware IPv6

仮想ルーティングおよび転送（VRF）Aware IPv6 トンネルを信頼できない IPv4 インフラストラクチャ内に作成することにより、このインフラストラクチャ上に IPv6 パケットを転送することができます。 これらのトンネルは、VRF テーブルまたはグローバル ルーティング テーブルにエンドポイントを持つことができます。 使用されるトンネルのモードは、**tunnel mode gre ipv4**（デフォルトのモード）と **tunnel mode ipv4** です。

VRF-Aware IPv6 トンネルの設定方法

VRF-Aware トンネルを設定するには、次の手順を実行する必要があります。

1 **カスタマー VRF とトランスポート VRF を定義します**：トンネルが VRF-Aware の場合はカスタマー VRF を定義します。 トンネル エンドポイントを VRF に設定する必要がある場合は、トランスポート VRF を定義します。 「VRF インスタンスの定義」の項を参照してください。

2 **ネットワークをセットアップします**：関連するインターフェイスを設定し、関連するルートを設定します。 有効なルートが PE デバイス間とカスタマーのネットワークの間にあることを確認します。

3 **PE デバイス間にトンネルを設定します**：「VRF-Aware トンネルの設定」の項を参照してください。

   1 **トンネル アドレスを設定します**
   2 **トンネル送信元を設定します**：これは、PE デバイス上のインターフェイスです。
   3 **トンネルの宛先を設定します**：これは、他の PE デバイスのトンネルの送信元です。 トンネルの正しい設定には、トンネルの宛先が ping コマンドで PE デバイスから到達可能でなければなりません（有効なルートが、トンネルの宛先に存在する必要があります）。
   4 **トンネル モードを設定します**

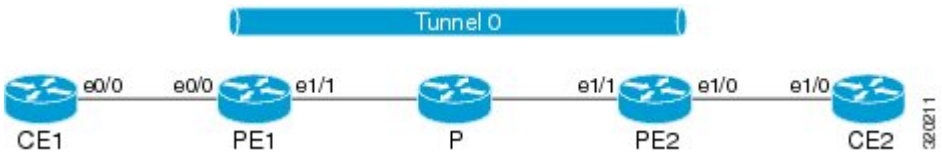
4 **カスタマーエッジ ネットワークを設定します。**「トンネリング用のカスタマーエッジ ネットワークの設定」の項を参照してください。

5 **トンネルを使用してスタティック ルートを設定します**：設定済みのトンネルを使用してリモート CE ネットワークに PE デバイス上のルートを設定します。

VRF-Aware トンネルの設定

このタスクでは、次のイメージに示すように、PE1 と PE2 間にトンネルを設定します。 両方の PE デバイス、PE1 と PE2 について設定作業を繰り返す必要があります。

図 *10*：*VRF-Aware* トンネルの設定



手順の概要

1. **interface** *type number*
2. **vrf forwarding** *transport-vrf-name*
3. • **ip address** *ip-address mask* または
   • **ipv6 address** *ipv6-address*/*prefix-length*
4. **exit**
5. プロバイダー エッジ デバイス間にスタティック ルートを設定します。
6. **interface tunnel** *number*
7. **vrf forwarding** *customer-vrf-name*
8. • **ip address** *ip-address mask* または
   • **ipv6 address** *ipv6-address*/*prefix-length*
9. **tunnel source** *interface-type interface-number*
10. **tunnel destination** [*ip-address* | *ipv6-address*]
11. **tunnel vrf** *transport-vrf-name*
12. **tunnel mode** {**aurp | cayman | dvmrp | eon | gre | gre multipoint | gre ipv6 | ipip [decapsulate-any] | ipsec ipv4 | iptalk | ipv6 | ipsec ipv6 | mpls | nos | rbscp**}
13. **exit**
14. • **ip route** [**vrf** *vrf-name*] *prefix mask interface-type interface-number* [*next-hop-ip-address*] または
    • **ipv6 route** [**vrf** *vrf-name*] *destination-ipv6-prefix interface-type interface-number* [*next-hop-ipv6-address*]
15. **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	**interface** *type number* 例： `Device(config)# interface ethernet 1/1`	トンネルの送信元として使用するインターフェイスを設定します。
ステップ 2	**vrf forwarding** *transport-vrf-name* 例： `Device(config-if)# vrf forwarding red`	（任意）トンネルとトランスポート VRF を関連付けます。 （注） この手順は、トンネル エンドポイントがグローバル ルーティング テーブルに設定されている場合には不要です。

	コマンドまたはアクション	目的

ステップ 3	• **ip address** *ip-address mask* または
	• **ipv6 address** *ipv6-address*/*prefix-length*
	
	例：
	`Device(config-if)# ip address 10.22.22.22 255.255.255.255`
	または
	`Device(config-if)# ipv6 address 2001:DB8:3::1/64`
		トンネルの送信元インターフェイスの IP アドレスを設定します。
		• PE1 について、この手順で設定されたアドレスは、トンネル エンドポイントまたはトンネルの宛先として使用されます。一方、PE2 にトンネルを設定する場合はこの逆となります。
		• このアドレスは、グローバル ルーティング テーブルまたは VRF に設定されている場合があります。

ステップ 4	**exit** 例： `Device(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 5	プロバイダー エッジ デバイス間にスタティック ルートを設定します。	プロバイダー エッジ デバイスは、**ping** コマンドまたは **ping vrf** コマンドを使用して到達可能です。
ステップ 6	**interface tunnel** *number* 例： `Device(config)# interface tunnel 0`	トンネル インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。 PE2 に同じトンネルを設定する必要があります。
ステップ 7	**vrf forwarding** *customer-vrf-name* 例： `Device(config-if)# vrf forwarding green`	（任意）トンネルとカスタマー VRF を関連付けます。 • トンネルから出たパケットはこの VRF（内部 IP パケット）に転送されます。 （注） このステップは、VRF-Aware トンネルでのみ必要です。

ステップ 8	• **ip address** *ip-address mask* または
	• **ipv6 address** *ipv6-address*/*prefix-length*
	
	例：
	`Device(config-if)# ip address 10.4.1.1 255.255.255.0`
	または
	`Device(config-if)# ipv6 address 2001:DB8:3::1/64`
		トンネルの IPv4 アドレスまたは IPv6 アドレスを設定します。
		• このアドレスは、ネクスト ホップ アドレスとしてスタティック ルートの設定時に使用されます。 PE1 と PE2 に同じネットワーク内のアドレスがあることを確認します。

	コマンドまたはアクション	目的
ステップ 9	**tunnel source** *interface-type interface-number* 例： `Device(config-if)# tunnel source ethernet 1/1`	トンネルインターフェイスの送信元アドレスを設定します。
ステップ 10	**tunnel destination** [*ip-address* | *ipv6-address*] 例： `Device(config-if)# tunnel destination 10.44.44.44`	（任意）トンネル インターフェイスの宛先を指定します。 • PE2 デバイスのトンネル送信元アドレスは PE1 のトンネル宛先アドレスとして使用されます。また、この逆の使用も可能です。 • IPv6 インフラストラクチャが 2 つの PE デバイス間にある場合は、IPv6 アドレスを使用します。 IPv4 インフラストラクチャが 2 つの PE デバイス間にある場合は、IPv4 アドレス（IPv4 トンネルを経由する IPv6）を使用します。
ステップ 11	**tunnel vrf** *transport-vrf-name* 例： `Device(config-if)# tunnel vrf red`	（任意）トンネルとトランスポート VRF を関連付けます。 • この VRF は、トンネルがパケット（外部 IP パケット ルーティング）を送信する際に経由する物理インターフェイスに関連付けられている VRF と同じです。 （注） この手順は、トンネルエンドポイントがグローバル ルーティング テーブルに設定されている場合には不要です。
ステップ 12	**tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **gre multipoint** | **gre ipv6** | **ipip** **[decapsulate-any]** | **ipsec ipv4** | **iptalk** | **ipv6** | **ipsec ipv6** | **mpls** | **nos** | **rbscp**} 例： `Device(config-if)# tunnel mode ipv6`	（任意）トンネル インターフェイスのカプセル化モードを設定します。 （注） トンネル モードが GRE IPv4 の場合は、これがデフォルト モードであるため、この手順は不要です。
ステップ 13	**exit** 例： `Device(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 モードに入ります。
ステップ 14	• **ip route** [**vrf** *vrf-name*] *prefix mask interface-type interface-number* [*next-hop-ip-address*] または	設定されたトンネルを使用してリモート カスタマー ネットワークへのスタティック ルートを確立します。 • ネクスト ホップとしてトンネル アドレスを使用します。

	コマンドまたはアクション	目的

	• **ipv6 route** [**vrf** *vrf-name*] *destination-ipv6-prefix interface-type interface-number* [*next-hop-ipv6-address*]
	
	例：
	`Device(config)# ip route 10.44.44.0 255.255.255.0 10.22.22.23`
	`Device(config)# ip route vrf red 10.44.44.0 255.255.255.0 10.22.22.23`
	または
	`Device(config)# ipv6 route 2001:DB8:2:2::/64 2001:DB8:2:1::2`
	`Device(config)# ipv6 route vrf green 2001:DB8:2:2::/64 2001:DB8:2:1::2`
		• PE1 では、ネットワーク PE2-CE2 へのスタティック ルートを設定します。PE2 では、ネットワーク PE1-CE1 へのスタティック ルートを設定します。

ステップ 15	**end** 例： `Device(config)# end`	グローバル 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

IPv6 トンネルを確認します。 「VRF-Aware トンネルの確認 」を参照してください。

VRF インスタンスの定義

仮想ルーティングおよび転送（VRF）Aware デバイスを作成し、VRF-Aware トンネルを設定するには、このタスクを実行します。

手順の概要

1. **vrf definition** *vrf-name*
2. **rd** *route-distinguisher*
3. **route-target export** *route-target-ext-community*
4. **route-target import** *route-target-ext-community*
5. **address-family {ipv4 | ipv6}**
6. **exit-address-family**
7. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**vrf definition** *vrf-name* 例： `Device(config)# vrf definition green`	VRF ルーティング テーブル インスタンスを定義するための IP VRF 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	**rd** *route-distinguisher* 例： `Device(config-vrf)# rd 1:1`	VRFインスタンスのルート識別子（RD）を指定します。
ステップ 3	**route-target export** *route-target-ext-community* 例： `Device(config-vrf)# route-target export 1:1`	ターゲット VPN 拡張コミュニティにルーティング情報をエクスポートします。
ステップ 4	**route-target import** *route-target-ext-community* 例： `Device(config-vrf)# route-target import 1:1`	ターゲット VPN 拡張コミュニティにルーティング情報をインポートします。
ステップ 5	**address-family {ipv4 | ipv6}** 例： `Device(config-vrf)# address-family ipv6`	VRF 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、標準 IPv4 または IPv6 アドレス プレフィックスを使用するルーティング セッションを設定します。
ステップ 6	**exit-address-family** 例： `Device(config-vrf-af)# exit-address-family`	VRF 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、IP VRF コンフィギュレーション モードを開始します。
ステップ 7	**exit** 例： `Device(config-vrf)# exit`	IP VRF 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 モードを開始します。

トンネリング用のカスタマー エッジ ネットワークの設定

カスタマー エッジ（CE）ネットワークを設定するには、このタスクを実行します。 この設定では、CE ネットワークは、プロバイダー エッジ（PE）デバイスに接続された CE デバイスを含む

ネットワークです。 PE1 と CE1 を接続し、PE2 と CE2 を接続します。 アドレスは適切に設定する必要があります。

はじめる前に

カスタマー VRF を定義するには、「VRF インスタンスの定義」の項を参照してください。

手順の概要

1. **interface** *type number*
2. **vrf forwarding** *customer-vrf-name*
3. • **ip address** *ip-address mask* または
   • **ipv6 address** *ipv6-address/prefix-length*
4. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**interface** *type number* 例： `Device(config)# interface Ethernet 0/0`	インターフェイス タイプを設定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 2	**vrf forwarding** *customer-vrf-name* 例： `Device(config-if)# vrf forwarding green`	（任意）VRF インスタンスまたは仮想ネットワークをトンネルと関連付けます。 （注） この手順は、インターフェイスを VRF に関連付ける必要がある場合にのみ必要です。

ステップ 3	• **ip address** *ip-address mask* または
	• **ipv6 address** *ipv6-address/prefix-length*
	
	例：
	`Device(config-if)# ip address 10.22.22.22 255.255.255.0`
	または
	`Device(config-if)# ipv6 address 2001:DB8:1::1/64`
		インターフェイスのアドレスを設定します。
		• PE デバイスに接続された CE デバイスが同じネットワーク上にあることを確認します。

ステップ 4	**exit** 例： `Device(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 モードに入ります。

VRF-Aware トンネルの確認

仮想ルーティングおよび転送（VRF）Aware トンネルを確認するには、次のコマンドを使用します。

手順の概要

1. **show tunnel interface**
2. **show ip route** *ip-address*
3. **show ip route vrf** *vrf-name ip-address*
4. **ping ipv6** *ipv6-address* **source** *ipv6-address*
5. **ping vrf** *vrf-name ipv6-address* **source** *ipv6-address*
6. **debug ipv6 icmp**

手順の詳細

ステップ 1 **show tunnel interface**

このコマンドは、すべてのトンネル インターフェイスに関する詳細情報を表示します。

例：

次は、総称ルーティング カプセル化（GRE）トンネル モードのプロバイダー エッジ（PE）からの出力例です。

```
Device# show tunnel interface

Tunnel0
   Mode:GRE/IP, Destination 10.44.44.44, Source Loopback2
   IP transport: output interface Ethernet1/0 next hop 10.0.0.2,
   Tunnel header destination 10.44.44.44
   Application ID 1: unspecified
   Linestate - current up, cached up
   Internal linestate - current up, evaluated up
```

例：

次は IPv6/IP トンネル モードの PE デバイスからの出力例です。

```
Device# show tunnel interface

Tunnel0
   Mode:IPv6/IP, Destination 44.44.44.44, Source Loopback2
   IP transport: output interface Ethernet1/0 next hop 2.0.0.2,
   Tunnel header destination 44.44.44.44
   Application ID 1: unspecified
   Linestate - current up, cached up
   Internal linestate - current up, evaluated up
```

出力が表示され、トンネル モードが確認できます。

ステップ 2 **show ip route** *ip-address*

このコマンドは、トンネルの宛先アドレスに詳細なルーティング情報を表示します。

例：
次は、グローバル ルーティング テーブルでのトンネル エンドポイントの PE デバイスからの出力例です。

```
Device# show ip route 10.44.44.44

  Routing entry for 10.44.44.44/32
  Known via "ospf 1", distance 110, metric 21, type intra area
  Last update from 10.0.0.2 on Ethernet1/0, 01:10:25 ago
  Routing Descriptor Blocks:
  * 10.0.0.2, from 10.44.44.44, 01:10:25 ago, via Ethernet1/0
      Route metric is 21, traffic share count is 1
```

次は、VRF テーブルにトンネル エンドポイントを持つ PE デバイスからの出力例です。

```
Device# show ip route 10.44.44.44

% Network not in table
```

出力が表示され、トンネルの宛先がグローバル ルーティング テーブルにあるかどうかを確認できます。

ステップ 3 **show ip route vrf** *vrf-name ip-address*
このコマンドは、宛先 IP アドレスに詳細なルーティング情報を表示します。

例：
次は、PE1 からの出力例です。

```
Device# show ip route vrf green 10.4.4.4

Routing entry for 10.4.4.4/32
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
  * 10.0.0.2, via Ethernet1/0
      Route metric is 0, traffic share count is 1
```

トンネル宛先アドレス 10.4.4.4 は、グローバル ルーティング テーブルにありません。

ステップ 4 **ping ipv6** *ipv6-address* **source** *ipv6-address*
このコマンドは、2 つのデバイス間の接続の状態を表示します。

例：
次は、カスタマー エッジ（CE）デバイス CE1 において CE2 に対して発行した **ping** コマンドの出力例です。

```
Device# ping ipv6 2001:DB8:2::1 source 2001:DB8:1::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:2::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:1::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/2/4 ms
```

ステップ 5 **ping vrf** *vrf-name ipv6-address* **source** *ipv6-address*
VRF-ping は VPN 接続をテストします。

例：

次は、CE2 に対して出された **ping vrf** コマンドによる CE1 からの出力例です。

```
Device# ping vrf green ipv6 2001:DB8:2::1 source 2001:DB8:1::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:2::1, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:1::2%green
!!!!!
```

表示された出力が成功を示している場合、VPN が正しく設定されています。

ステップ 6 **debug ipv6 icmp**

このコマンドは、IPv6 インターネット制御メッセージ プロトコル（ICMP）トランザクションのデバッグ メッセージを表示します。

例：

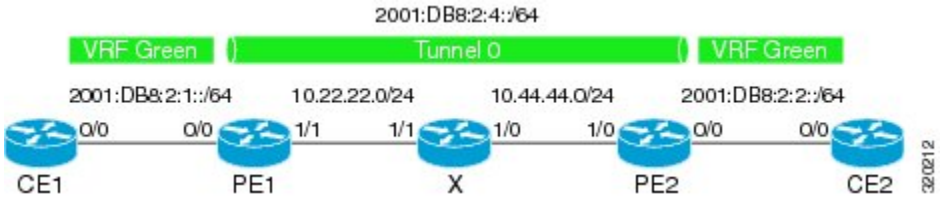
次にサンプル出力を示します。

```
Device# debug ipv6 icmp

  ICMP Packet debugging is on

*Apr  6 14:08:10.743: ICMPv6: Received echo request, Src=2001:DB8:1::2, Dst=2001:DB8:2::1
*Apr  6 14:08:10.743: ICMPv6: Sent echo reply, Src=2001:DB8:2::1, Dst=2001:DB8:1::2
...
```

表示された出力が成功を示している場合、VPN が正しく設定されています。

VRF-Aware トンネルの設定例

例：VRF-Aware トンネルの設定（グローバル ルーティング テーブルでのトンネル エンドポイント）



例：CE1 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
vrf definition green
rd 1:1
```

```
route-target export 1:1
route-target import 1:1
address-family ipv6
exit-address-family
exit
!
interface Ethernet0/0
 vrf forwarding green
 no ip address
 ipv6 address 2001:DB8:2:1::1/64
 no shutdown
 exit
!
!
ipv6 route vrf green 2001:DB8:2:2::/64 2001:DB8:2:1::2
ipv6 route vrf green 2001:DB8:2:4::/64 2001:DB8:2:1::2
!
```

例：PE1 の設定

```
ipv6 unicast-routing
ipv6 cef
!
vrf definition green
rd 1:1
route-target export 1:1
route-target import 1:1
address-family ipv6
exit-address-family
exit
!
interface Tunnel0
 no ip address
 vrf forwarding green
 ipv6 address 2001:DB8:2:4::1/64
 tunnel source 10.22.22.22
 tunnel destination 10.44.44.44
 exit
!
interface Ethernet0/0
 vrf forwarding green
 no ip address
 ipv6 address 2001:DB8:2:1::2/64
 no shutdown
 exit
!
interface Ethernet1/1
 no ip address
 ip address 10.22.22.22 255.255.255.0
 no shutdown
 exit
!
ip route 10.44.44.0 255.255.255.0 10.22.22.23
ipv6 route vrf green 2001:DB8:2:2::/64 Tunnel0 2001:DB8:2:4::2
```

例：PE2 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
vrf definition green
rd 1:1
route-target export 1:1
route-target import 1:1
address-family ipv6
exit-address-family
exit
```

```
!
interface Tunnel0
 vrf forwarding green
 no ipv6 address
 ipv6 address 2001:DB8:2:4::2/64
 tunnel source 10.44.44.44
 tunnel destination 10.22.22.22
 exit
!
interface Ethernet0/0
 vrf forwarding green
 no ipv6 address
 ipv6 address 2001:DB8:2:2::1/64
 no shutdown
 exit
!
interface Ethernet1/0
 no ip address
 ip address 10.44.44.44 255.255.255.0
 no shutdown
 exit
!
ip route 10.22.22.0 255.255.255.0 10.44.44.43
!
ipv6 route vrf green 2001:DB8:2:1::/64 Tunnel0 2001:DB8:2:4::1
!
```

例：CE2 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
vrf definition green
rd 1:1
route-target export 1:1
route-target import 1:1
address-family ipv6
exit-address-family
exit
!
interface Ethernet0/0
 vrf forwarding green
 no ipv6 address
 ipv6 address 2001:DB8:2:2::2/64
 no shutdown
 exit
!
!
ipv6 route vrf green 2001:DB8:2:1::/64 2001:DB8:2:2::1
ipv6 route vrf green 2001:DB8:2:4::/64 2001:DB8:2:2::1
!
```

例：デバイス X の設定

```
!
interface Ethernet1/0
 no ip address
 ip address 10.44.44.43 255.255.255.0
 no shutdown
 exit
!
interface Ethernet1/1
 no ip address
 ip address 10.22.22.23 255.255.255.0
 no shutdown
 exit
```

```
!
```

例：トンネル設定の確認

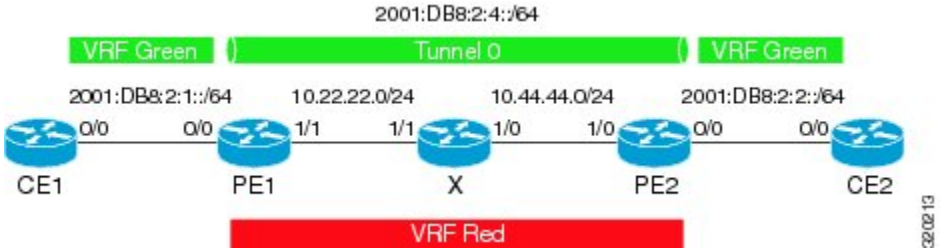
CE1 から

```
Device# ping vrf green ipv6 2001:db8:2:2::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:2:2::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/6 ms

Device# ping vrf green ipv6 2001:db8:2:2::2 source 2001:db8:2:1::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:2:2::2, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:2:1::1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

PE1 から

```
Device# show tunnel interface

Tunnel0
   Mode:GRE/IP, Destination 10.44.44.44, Source 10.22.22.22
   IP transport: output interface Ethernet1/1 next hop 10.22.22.23
   Application ID 1: unspecified
   Linestate - current up
   Internal linestate - current up, evaluated up
   Tunnel Source Flags: Local
   Transport IPv4 Header DF bit cleared
   OCE: IP tunnel decap
   Provider: interface Tu0, prot 47
     Performs protocol check [47]
     Protocol Handler: GRE: opt 0x0
       ptype: ipv4 [ipv4 dispatcher: punt]
       ptype: ipv6 [ipv6 dispatcher: from if Tu0]
       ptype: mpls [mpls dispatcher: drop]
       ptype: otv [mpls dispatcher: drop]
       ptype: generic [mpls dispatcher: drop]
There are 0 tunnels running over the EON IP protocol
There are 0 tunnels running over the IPinIP protocol
There are 0 tunnels running over the NOSIP protocol
There are 0 tunnels running over the IPv6inIP protocol
There are 0 tunnels running over the RBSCP/IP protocol


Device# show ip route 10.44.44.44

Routing entry for 10.44.44.0/24
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
  * 10.22.22.23
      Route metric is 0, traffic share count is 1

Device# debug ipv6 icmp

  ICMP Packet debugging is on
*Jan  1 10:57:37.882: ICMPv6: Sent R-Advert, Src=FE80::A8BB:CCFF:FE00:5200, Dst=FF02::1
*Jan  1 11:00:18.634: ICMPv6: Received R-Advert, Src=FE80::A8BB:CCFF:FE00:5200,Dst=FF02::1
```

例：VRF-Aware トンネルの設定（VRF でのトンネル エンドポイント）



例：CE1 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
vrf definition green
rd 1:1
route-target export 1:1
route-target import 1:1
address-family ipv6
exit-address-family
exit
!
interface Ethernet0/0
 vrf forwarding green
 no ip address
 ipv6 address 2001:DB8:2:1::1/64
 no shutdown
 exit
!
!
ipv6 route vrf green 2001:DB8:2:2::/64 2001:DB8:2:1::2
ipv6 route vrf green 2001:DB8:2:4::/64 2001:DB8:2:1::2
!
```

例：PE1 の設定

```
ipv6 unicast-routing
ipv6 cef
!
vrf definition green
rd 1:1
route-target export 1:1
route-target import 1:1
address-family ipv6
exit-address-family
exit
!
vrf definition red
rd 2:2
route-target export 2:2
route-target import 2:2
address-family ipv4
exit-address-family
exit
!
interface Tunnel0
 no ip address
 vrf forwarding green
 ipv6 address 2001:DB8:2:4::1/64
 tunnel source 10.22.22.22
```

```
 tunnel destination 10.44.44.44
 tunnel vrf red
 exit
!
interface Ethernet0/0
 vrf forwarding green
 no ip address
 ipv6 address 2001:DB8:2:1::2/64
 no shutdown
 exit
!
interface Ethernet1/1
 vrf forwarding red
 no ip address
 ip address 10.22.22.22 255.255.255.0
 no shutdown
 exit
!
ip route vrf red 10.44.44.0 255.255.255.0 10.22.22.23
ipv6 route vrf green 2001:DB8:2:2::/64 Tunnel0 2001:DB8:2:4::2
```

例：PE2 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
vrf definition green
rd 1:1
route-target export 1:1
route-target import 1:1
address-family ipv6
exit-address-family
exit
!
vrf definition red
rd 2:2
route-target export 2:2
route-target import 2:2
address-family ipv4
exit-address-family
exit
!
interface Tunnel0
 vrf forwarding green
 no ipv6 address
 ipv6 address 2001:DB8:2:4::2/64
 tunnel source 10.44.44.44
 tunnel destination 10.22.22.22
 tunnel vrf red
 exit
!
interface Ethernet0/0
 vrf forwarding green
 no ipv6 address
 ipv6 address 2001:DB8:2:2::1/64
 no shutdown
 exit
!
interface Ethernet1/0
 vrf forwarding red
 no ip address
 ip address 10.44.44.44 255.255.255.0
 no shutdown
 exit
!
ip route vrf red 10.22.22.0 255.255.255.0 10.44.44.43
!
ipv6 route vrf green 2001:DB8:2:1::/64 Tunnel0 2001:DB8:2:4::1
```

```
!
```

例：CE2 の設定

```
!
ipv6 unicast-routing
ipv6 cef
!
vrf definition green
rd 1:1
route-target export 1:1
route-target import 1:1
address-family ipv6
exit-address-family
exit
!
interface Ethernet0/0
 vrf forwarding green
 no ipv6 address
 ipv6 address 2001:DB8:2:2::2/64
 no shutdown
 exit
!
!
ipv6 route vrf green 2001:DB8:2:1::/64 2001:DB8:2:2::1
ipv6 route vrf green 2001:DB8:2:4::/64 2001:DB8:2:2::1
!
```

例：デバイス X の設定

```
!
interface Ethernet1/0
 vrf forwarding red
 no ip address
 ip address 10.44.44.43 255.255.255.0
 no shutdown
 exit
!
interface Ethernet1/1
 vrf forwarding red
 no ip address
 ip address 10.22.22.23 255.255.255.0
 no shutdown
 exit
!
```

例：トンネル設定の確認

CE1 から

```
Device# ping vrf green ipv6 2001:db8:2:2::2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:2:2::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/6 ms

Device# ping vrf green ipv6 2001:db8:2:2::2 source 2001:db8:2:1::1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:2:2::2, timeout is 2 seconds:
Packet sent with a source address of 2001:DB8:2:1::1
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

PE1 から

```
Device# show tunnel interface

Tunnel0
   Mode:GRE/IP, Destination 10.44.44.44, Source 10.22.22.22
   IP transport: output interface Ethernet1/1 next hop 10.22.22.23
   Application ID 1: unspecified
   Linestate - current up
   Internal linestate - current up, evaluated up
   Tunnel Source Flags: Local
   Transport IPv4 Header DF bit cleared
   OCE: IP tunnel decap
   Provider: interface Tu0, prot 47
     Performs protocol check [47]
     Protocol Handler: GRE: opt 0x0
       ptype: ipv4 [ipv4 dispatcher: punt]
       ptype: ipv6 [ipv6 dispatcher: from if Tu0]
       ptype: mpls [mpls dispatcher: drop]
       ptype: otv [mpls dispatcher: drop]
       ptype: generic [mpls dispatcher: drop]
There are 0 tunnels running over the EON IP protocol
There are 0 tunnels running over the IPinIP protocol
There are 0 tunnels running over the NOSIP protocol
There are 0 tunnels running over the IPv6inIP protocol
There are 0 tunnels running over the RBSCP/IP protocol

Device# show ip route 10.44.44.44

% Network not in table

Device# show ip route vrf red 10.44.44.44

Routing Table: red
Routing entry for 10.44.44.0/24
  Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
  * 10.22.22.23
      Route metric is 0, traffic share count is 1

Device# debug ipv6 icmp

  ICMP Packet debugging is on
*Jan  1 10:57:37.882: ICMPv6: Sent R-Advert, Src=FE80::A8BB:CCFF:FE00:5200, Dst=FF02::1
*Jan  1 11:00:18.634: ICMPv6: Received R-Advert, Src=FE80::A8BB:CCFF:FE00:5200,Dst=FF02::1
```

その他の関連資料

関連資料

関連項目	マニュアル タイトル
IPv6 アドレッシングと接続	『*Cisco IOS IPv6 Configuration Guide*』
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』

関連項目	マニュアル タイトル
IPv6 コマンド	『Cisco IOS IPv6 Command Reference』
Cisco IOS IPv6 機能	『Cisco IOS IPv6 Feature Mapping』

標準/RFC	タイトル
IPv6 に関する RFC	『*IPv6 RFCs*』

標準および RFC

シスコのテクニカル サポート

説明	リンク
シスコのサポートおよびドキュメンテーション Web サイトでは、ダウンロード可能なマニュアル、ソフトウェア、ツールなどのオンライン リソースを提供しています。 これらのリソースは、ソフトウェアをインストールして設定したり、シスコの製品やテクノロジーに関する技術的問題を解決したりするために使用してください。 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/cisco/web/support/index.html

VRF-Aware トンネルの機能情報

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。

プラットフォームのサポートおよびシスコ 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 にアクセスするには、www.cisco.com/go/cfn に移動します。 Cisco.com のアカウントは必要ありません。

表 *25* : *VRF-Aware* トンネルの機能情報

機能名	リリース	機能情報
VRF-Aware トンネル	Cisco IOS XE Release 3.8S	仮想ルーティングおよびフォワーディング（VRF）Aware トンネルは、信頼できないコアネットワークまたは別のインフラストラクチャ（IPv4 または IPv6）を備えたコア ネットワークで区切られたカスタマーネットワークに接続するために使用されます。 **tunnel vrf** コマンドは、IPv6 トランスポートをサポートするように変更されました。

VRF-Aware トンネルの前提条件

- カスタマーエッジネットワークを設定する必要があります。「トンネリング用のカスタマーエッジ ネットワークの設定」の項を参照してください。
- カスタマーを設定し、VRF を転送する必要があります。「VRF インスタンスの定義」の項を参照してください。