



Cisco NCS 5000 シリーズ ルータ、IOS XR リリース 6.0.x 向けインターフェイスおよびハードウェアコンポーネントのコンフィギュレーションガイド

初版：2015 年 12 月 23 日

最終更新：年 月 日

シスコシステムズ合同会社

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

<http://www.cisco.com/jp>

お問い合わせ先：シスコ コンタクトセンター

0120-092-255（フリーコール、携帯・PHS含む）

電話受付時間：平日 10:00～12:00、13:00～17:00

<http://www.cisco.com/jp/go/contactcenter/>

【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

はじめに vii

マニュアルの変更履歴 vii

マニュアルの入手方法およびテクニカル サポート vii

物理インターフェイスのプリコンフィギュレーション 1

物理インターフェイスのプリコンフィギュレーションの概要 2

物理インターフェイスのプリコンフィギュレーションの前提条件 3

インターフェイスのプリコンフィギュレーションを行う利点 3

物理インターフェイスのプリコンフィギュレーションを行う方法 3

物理インターフェイスのプリコンフィギュレーションに関する情報 6

 インターフェイス プリコンフィギュレーション コマンドの使用法 6

管理イーサネット インターフェイスの高度な設定および変更 7

管理イーサネット インターフェイスの設定の前提条件 7

高度な管理イーサネット インターフェイス設定の実行 8

 管理イーサネット インターフェイスの設定 8

 管理イーサネット インターフェイスのデュプレックス モードの設定 11

 管理イーサネット インターフェイスの速度の設定 12

 管理イーサネット インターフェイスの MAC アドレスの変更 14

 管理イーサネット インターフェイス設定の確認 15

管理イーサネット インターフェイスの設定に関する情報 16

 デフォルト インターフェイス設定 16

イーサネット インターフェイスの設定 17

イーサネット インターフェイスの設定 17

ギガビット イーサネット インターフェイスの設定 17

L2VPN イーサネット ポートの設定 22

イーサネットの設定に関する情報 25

 ギガビット イーサネットおよび 10 ギガビット イーサネットのデフォルト設定値 25

| | |
|--|-----------|
| イーサネット インターフェイスのリンクのオートネゴシエーション | 26 |
| イーサネット MTU | 27 |
| イーサネット インターフェイスでのフロー制御 | 28 |
| 802.1Q VLAN | 28 |
| VRRP | 28 |
| HSRP | 28 |
| サブインターフェイス | 29 |
| レイヤ 2 サブインターフェイス (EFP) の拡張パフォーマンス モニタリ ング | 32 |
| その他のパフォーマンス管理の機能拡張 | 33 |
| リンク バンドルの設定 | 35 |
| イーサネット リンク バンドルの機能および互換性のある特性 | 36 |
| イーサネット リンク バンドルの設定 | 37 |
| イーサネット リンク バンドルでの EFP ロード バランシングの設定 | 42 |
| イーサネット リンク バンドル上の VLAN | 44 |
| VLAN バンドルの設定 | 45 |
| 46 | |
| リンク バンドルの設定に関する情報 | 50 |
| LACP を通じたリンク集約 | 50 |
| IEEE 802.3ad 規格 | 51 |
| ロード バランシング | 51 |
| リンク バンドルの設定の概要 | 52 |
| RP スイッチオーバー時のノンストップ フォワーディング | 53 |
| リンク スイッチオーバー | 53 |
| 仮想ループバックおよびヌル インターフェイスの設定 | 55 |
| 仮想インターフェイスの設定に関する情報 | 55 |
| 仮想ループバック インターフェイスの概要 | 55 |
| 仮想インターフェイスの設定の前提条件 | 56 |
| 仮想ループバック インターフェイスの設定 | 56 |
| ヌル インターフェイスの概要 | 58 |
| ヌル インターフェイスの設定 | 59 |
| 仮想 IPV4 インターフェイスの設定 | 60 |
| 802.1Q VLAN インターフェイスの設定 | 63 |

| | |
|-------------------------------|----|
| 802.1Q VLAN インターフェイスの設定方法 | 64 |
| 802.1Q VLAN サブインターフェイスの設定 | 64 |
| 802.1Q VLAN サブインターフェイスの削除 | 67 |
| 802.1Q VLAN インターフェイスの設定に関する情報 | 68 |
| サブインターフェイス | 68 |
| サブインターフェイス MTU | 69 |
| EFP | 69 |
| VLAN でのレイヤ 2 VPN | 69 |



はじめに

Cisco NCS 5000 シリーズルータ向けインターフェイスおよびハードウェア コンポーネントのコンフィギュレーションガイドでは、ルータ インターフェイスおよびハードウェア設定に関連する情報と手順について説明します。

ここでは、次の内容について説明します。

- [マニュアルの変更履歴](#), [vii ページ](#)
- [マニュアルの入手方法およびテクニカル サポート](#), [vii ページ](#)

マニュアルの変更履歴

表 1 に、初版後、このマニュアルに加えられた技術的な変更の履歴を示します。

表 1: マニュアルの変更履歴

| 日付 | Summary |
|-------------|------------|
| 2015 年 12 月 | このマニュアルの初版 |

マニュアルの入手方法およびテクニカル サポート

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

『*What's New in Cisco Product Documentation*』はシスコの新規および改訂版の技術マニュアルの一覧を提供するもので、RSS フィードとして購読できます。また、リーダー アプリケーションを使用すると、コンテンツがデスクトップに直接配信されるようになります。RSS フィードは無料のサービスです。



第 1 章

物理インターフェイスのプリコンフィギュレーション

このモジュールでは、物理インターフェイスのプリコンフィギュレーションについて説明します。

プリコンフィギュレーションは、次のタイプのインターフェイスやコントローラでサポートされます。

- ギガビットイーサネット
- 10 ギガビットイーサネット
- 100 ギガビットイーサネット
- 管理イーサネット

プリコンフィギュレーションによって、モジュラ サービス カードをルータへの装着前に設定できます。カードを装着すると、ただちに設定されます。プリコンフィギュレーション情報は、通常の方法で設定されたインターフェイスの場合とは異なり、別のシステム データベース ツリー（ルート プロセッサ上のプリコンフィギュレーション ディレクトリ）に作成されます。

検証機能が動作するのはモジュラ サービス カード上に限られるため、モジュラ サービス カードが存在していなければ検証できないプリコンフィギュレーション データもあります。このようなプリコンフィギュレーション データは、モジュラ サービス カードを装着し、検証機能が起動したときに検証されます。設定がプリコンフィギュレーション 領域からアクティブ 領域にコピーされるとときにエラーが検出されると、設定は拒否されます。



(注) 1 ギガビットイーサネット インターフェイスはサポートされていません。10 ギガビットイーサネット インターフェイスは、光タイプに基づいて 1 GigE モードで動作できます。



(注) プリコンフィギュレーションを実行できるのは物理インターフェイスだけです。

- [物理インターフェイスのプリコンフィギュレーションの概要, 2 ページ](#)
- [物理インターフェイスのプリコンフィギュレーションの前提条件, 3 ページ](#)
- [インターフェイスのプリコンフィギュレーションを行う利点, 3 ページ](#)
- [物理インターフェイスのプリコンフィギュレーションを行う方法, 3 ページ](#)
- [物理インターフェイスのプリコンフィギュレーションに関する情報, 6 ページ](#)

物理インターフェイスのプリコンフィギュレーションの概要

プリコンフィギュレーションは、インターフェイスがシステムに存在しないうちにインターフェイスを設定する作業です。プリコンフィギュレーションされたインターフェイスは、位置（ラック/スロット/モジュール）が一致するインターフェイスが実際にルータに装着されるまで検証または適用されません。適切なモジュラサービスカードが装着され、インターフェイスが作成されると、事前に作成された設定情報が検証され、問題がなければ、ただちにルータの実行コンフィギュレーションに適用されます。



(注) 適切なモジュラサービスカードを装着するときには、適切な **show** コマンドを使用してプリコンフィギュレーションの内容を検証してください。

プリコンフィギュレーション済みの状態にあるインターフェイスを表示するには、**show run** コマンドを使用します。



(注) カードを装着し、インターフェイスをアップ状態にするときに、想定される設定と実際にプリコンフィギュレーションされたインターフェイスを比較できるように、サイトプランニングガイドにプリコンフィギュレーション情報を記入することをお勧めします。



ヒント プリコンフィギュレーションを実行コンフィギュレーションファイルに保存するには、**commit best-effort** コマンドを使用します。**commit best-effort** コマンドは、ターゲット コンフィギュレーションと実行コンフィギュレーションを結合し、有効な設定だけをコミットします（ベストエフォート）。セマンティックエラーにより一部の設定が適用されないこともありますが、その場合でも有効な設定はアップ状態になります。

物理インターフェイスのプリコンフィギュレーションの前提条件

物理インターフェイスのプリコンフィギュレーションを実行する前に、次の条件が満たされていることを確認します。

- プリコンフィギュレーションドライバおよびファイルがインストールされている必要があります。プリコンフィギュレーションドライバがインストールされていなくても物理インターフェイスのプリコンフィギュレーションを行える場合もありますが、ルータ上で有効なインターフェイス名の文字列を提供するインターフェイス定義ファイルを設定するには、プリコンフィギュレーションファイルが必要です。

インターフェイスのプリコンフィギュレーションを行う利点

プリコンフィギュレーションによって、新しいカードをシステムに追加するときのダウンタイムが短縮されます。プリコンフィギュレーションを行うと、新しいモジュラサービスカードが即座に設定され、カードのブートアップ中も動作します。

プリコンフィギュレーションを行うもう1つの利点は、モジュラサービスカードの交換時に、カードを取り外した後でも、以前の設定を表示し、変更できることです。

物理インターフェイスのプリコンフィギュレーションを行う方法

ここでは、インターフェイスの最も基本的なプリコンフィギュレーションについてのみ説明します。

手順の概要

1. **configure**
2. **interface preconfigure***type interface-path-id*
3. 次のいずれかのコマンドを使用します。
 - **ipv4 address***ip-address subnet-mask*
 - **ipv4 address***ip-address/prefix*
4. 追加のインターフェイスパラメータを設定します。詳細については、設定するインターフェイスのタイプに対応する、このマニュアルの設定の章を参照してください。
5. **end** または **commit***best-effort*
6. **show running-config**

手順の詳細

ステップ 1 **configure**

例：

```
RP/0/RP0/CPU0:router#configure
グローバル コンフィギュレーション モードを開始します。
```

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

例：

```
RP/0/RP0/CPU0:router(config)# interface preconfigure HundredGigE 0/0/1/0
インターフェイスのインターフェイス プリコンフィギュレーション モードを開始します。このモードでは、type でサポート対象のインターフェイスタイプのうちどれを設定するかを指定し、interface-path-id でインターフェイスの場所を rack/slot/module/port 表記で指定します。
```

ステップ 3 次のいずれかのコマンドを使用します。

- **ipv4 address***ip-address subnet-mask*
- **ipv4 address***ip-address/prefix*

例：

```
RP/0/RP0/CPU0:router(config-if-pre)# ipv4 address 192.168.1.2/32
IP アドレスとマスクをインターフェイスに割り当てます。
```

ステップ 4 追加のインターフェイスパラメータを設定します。詳細については、設定するインターフェイスのタイプに対応する、このマニュアルの設定の章を参照してください。

ステップ 5 **end** または **commit***best-effort*

例：

```
RP/0/RP0/CPU0:router(config-if-pre)# end
```

または

```
RP/0/RP0/CPU0:router(config-if-pre)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。Uncommitted changes found, commit them before exiting (yes/no/cancel)?
- **Yes** と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
- **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
- **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを継続するには、**commit best-effort** コマンドを使用します。**commit best-effort** コマンドは、ターゲットコンフィギュレーションと実行コンフィギュレーションを結合し、有効な変更だけをコミットします（ベストエフォート）。セマンティックエラーが原因で、一部の設定変更は失敗する場合があります。

ステップ 6 show running-config

例：

```
RP/0/RP0/CPU0:router# show running-config
```

(任意) 現在ルータで使用されている設定情報を表示します。

次に、基本的なイーサネットインターフェイスのプリコンフィギュレーションを行う例を示します。

```
RP/0/RP0/CPU0:router# configure
```

```
RP/0/RP0/CPU0:router(config)# interface preconfigure HundredGigE 0/0/1/0
```

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.1.2/32
```

```
RP/0/RP0/CPU0:router(config-if-pre)# commit
```

物理インターフェイスのプリコンフィギュレーションに関する情報

インターフェイスのプリコンフィギュレーションを行うには、次の概念を理解している必要があります。

インターフェイス プリコンフィギュレーション コマンドの使用法

システムにまだ存在しないインターフェイスのプリコンフィギュレーションを行うには、グローバル コンフィギュレーション モードで `interface preconfigure` コマンドを使用します。

`interface preconfigure` コマンドによって、ルータはインターフェイス コンフィギュレーション モードに移行します。ユーザは、使用可能なすべてのコマンドを追加できます。プリコンフィギュレーションされたインターフェイス用に登録された検証機能により、設定が検証されます。ユーザが `end` コマンドを入力するか、それに対応する `exit` コマンドまたはグローバルコンフィギュレーションモード コマンドを入力すると、プリコンフィギュレーションが完了します。



(注) モジュラ サービス カードを装着しなければ検証できない設定もあります。

新たにプリコンフィギュレーションされたインターフェイスには **no shutdown** コマンドを入力しないでください。このコマンドの **no** 形式は既存の設定を削除するものであり、この場合は既存の設定が存在しないからです。

ユーザがプリコンフィギュレーション時に指定する名前は、作成するインターフェイスの名前と一致する必要があります。インターフェイス名が一致しない場合、インターフェイスの作成時にプリコンフィギュレーションを適用できません。インターフェイス名は、ルータがサポートし、対応するドライバがインストール済みのインターフェイス タイプから始めます。ただし、スロット、ポート、サブインターフェイス番号、およびチャネルインターフェイス番号の情報は検証できません。



(注) すでに存在し、設定されているインターフェイス名（または `e0/3/0/0` のような省略形）は指定できません。



第 2 章

管理イーサネット インターフェイスの高度な設定および変更

このモジュールでは、管理イーサネット インターフェイスの設定について説明します。

Telnet を使用して LAN IP アドレスを介してルータにアクセスする前に、管理イーサネット インターフェイスを設定し、Telnet サーバをイネーブルにしておく必要があります。



(注) 物理層インターフェイスモジュール (PLIM) と管理イーサネットインターフェイスポート間のフォワーディングは、デフォルトではディセーブルに設定されています。PLIM ポートと管理イーサネットインターフェイスポート間での転送をイネーブルにするには、`rp mgmtethernet forwarding` コマンドを使用します。



(注) システムの管理イーサネット インターフェイスはデフォルトで表示されますが、これらのインターフェイスを使用してルータにアクセスしたり、簡易ネットワーク管理プロトコル (SNMP)、Common Object Request Broker Architecture (CORBA)、HTTP、Extensible Markup Language (XML)、TFTP、Telnet、コマンドラインインターフェイス (CLI) などのプロトコルやアプリケーションを使用したりするには、インターフェイスを設定する必要があります。

- [管理イーサネット インターフェイスの設定の前提条件, 7 ページ](#)
- [高度な管理イーサネット インターフェイス設定の実行, 8 ページ](#)
- [管理イーサネット インターフェイスの設定に関する情報, 16 ページ](#)

管理イーサネット インターフェイスの設定の前提条件

この章で説明する管理イーサネット インターフェイスの設定手順を実行する前に、次に示す作業が実施されており、条件を満たしていることを確認する必要があります。

- 管理イーサネットインターフェイスの初期設定は実行済みです。
- 汎用インターフェイス名の仕様である *rack/slot/module/port* の適用方法を理解しています。



(注) トランスペアレントスイッチオーバーの場合、アクティブおよびスタンバイの管理イーサネットインターフェイスが両方とも、物理的に同じ LAN またはスイッチに接続されている必要があります。

高度な管理イーサネットインターフェイス設定の実行

この項では、次の手順について説明します。

管理イーサネットインターフェイスの設定

管理イーサネットインターフェイスを設定するには、次の作業を行います。この手順では、管理イーサネットインターフェイスに必要な最小限の設定について説明します。

MTU は、管理イーサネットインターフェイスに設定できません。デフォルト値は 1514 バイトです。

手順の概要

1. **configure**
2. **interface MgmtEth***interface-path-id*
3. **ipv4 address***ip-address mask*
4. **mtu***bytes*
5. **no shutdown**
6. **end** または **commit**
7. **show interfaces MgmtEth** *interface-path-id*

手順の詳細

ステップ 1 **configure**

例：

```
RP/0/RP0/CPU0:router# configure
グローバル コンフィギュレーション モードを開始します。
```

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

例：

```
RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0
```

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

この例では、スロット 0 に装着された RSP カードのポート 0 を示しています。

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

例：

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 1.76.18.150/16 255.255.0.0
```

IP アドレスとサブネット マスクをインターフェイスに割り当てます。

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

ステップ 4 **mtu***bytes*

例：

```
RP/0/RP0/CPU0:router(config-if)# mtu 1488
```

(任意) インターフェイスの最大伝送単位 (MTU) バイト値を設定します。デフォルト値は 1514 です。

- デフォルトは 1514 バイトです。
- 管理イーサネット インターフェイス インターフェイスの **mtu** 値は 64 ~ 1514 バイトの範囲です。

ステップ 5 **no shutdown**

例：

```
RP/0/RP0/CPU0:router(config-if)# no shutdown
```

shutdown 設定を削除します。その結果、インターフェイスに強制されていた管理上のダウン状態が解除され、アップ状態またはダウン状態に移行できるようになります。

ステップ 6 **end** または **commit**

例：

```
RP/0/RP0/CPU0:router(config-if)# end
```

または

```
RP/0/RP0/CPU0:router(config-if)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?
[cancel]:
```

- **yes** と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
- **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
- **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ステップ 7 show interfaces MgmtEth interface-path-id

例：

```
RP/0/RP0/CPU0:router# show interfaces MgmtEth 0/RP0/CPU0/0
```

(任意) ルータ上のインターフェイスに関する統計情報を表示します。

次に、RP での管理イーサネットインターフェイスの高度な設定とその確認を行う例を示します。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:router(config)# ipv4 address 1.76.18.150/16 255.255.0.0
RP/0/RP0/CPU0:router(config-if)# speed 100
RP/0/RP0/CPU0:router(config-if)# duplex full
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# commit
RP/0/RP0/CPU0:router:Mar 26 01:09:28.685 :ifmgr[190]:%LINK-3-UPDOWN :Interface
MgmtEth0/RP0/CPU0/0, changed state to Up
RP/0/RP0/CPU0:router(config-if)# end

RP/0/RP0/CPU0:router# show interfaces MgmtEth 0/RP0/CPU0/0

MgmtEth0/RP0/CPU0/0 is up, line protocol is up
  Interface state transitions: 3
  Hardware is Management Ethernet, address is 1005.cad8.4354 (bia 1005.cad8.4354)
  Internet address is 1.76.18.150/16
  MTU 1488 bytes, BW 1000000 Kbit (Max: 1000000 Kbit)
    reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 1000Mb/s, 1000BASE-T, link type is autonegotiation
  loopback not set,
  Last link flapped 00:00:59
```

```
ARP type ARPA, ARP timeout 04:00:00
Last input 00:00:00, output 00:00:02
Last clearing of "show interface" counters never
5 minute input rate 4000 bits/sec, 3 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 21826 packets input, 4987886 bytes, 0 total input drops
 0 drops for unrecognized upper-level protocol
Received 12450 broadcast packets, 8800 multicast packets
 0 runts, 0 giants, 0 throttles, 0 parity
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
1192 packets output, 217483 bytes, 0 total output drops
Output 0 broadcast packets, 0 multicast packets
0 output errors, 0 underruns, 0 applique, 0 resets
0 output buffer failures, 0 output buffers swapped out
3 carrier transitions
```

```
RP/0/RP0/CPU0:router# show running-config interface MgmtEth 0/RP0/CPU0/0
```

```
interface MgmtEth0/RP0/CPU0/0
mtu 1488
ipv4 address 1.76.18.150 255.255.0.0
ipv6 address 2002::14c:125a/128
ipv6 enable
!
```

管理イーサネットインターフェイスのデュプレックスモードの設定

RPに対応した管理イーサネットインターフェイスのデュプレックスモードを設定するには、次の作業を行います。

手順の概要

1. **configure**
2. **interface MgmtEth***interface-path-id*
3. **duplex**[full | half]
4. **end** または **commit**

手順の詳細

ステップ 1 **configure**

例：

```
RP/0/RP0/CPU0:router# configure
グローバル コンフィギュレーション モードを開始します。
```

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

例：

```
RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0
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、管理イーサネットインターフェイスの名前とインスタンスを指定します。
```

ステップ 3 **duplex**[full | half]

例：

```
RP/0/RP0/CPU0:router(config-if)# duplex full
```

インターフェイスのデュプレックスモードを設定します。有効なオプションは **full** または **half** です。

- (注)
- システムをオートネゴシエーションされたデュプレックス操作に戻すには、**no duplex** コマンドを使用します。

ステップ 4 end または commit

例：

```
RP/0/RP0/CPU0:router(config-if)# end
```

または

```
RP/0/RP0/CPU0:router(config-if)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?  
[cancel]:
```

- **yes** と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
- **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻りません。変更はコミットされません。
- **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

管理イーサネットインターフェイスの速度の設定

RPに対応した管理イーサネットインターフェイスの速度を設定するには、次の作業を行います。

手順の概要

1. **configure**
2. **interface MgmtEthinterface-path-id**
3. **speed {10 | 100 | }**
4. **end** または **commit**

手順の詳細

ステップ 1 **configure**

例：

```
RP/0/RP0/CPU0:router# configure
```

グローバル コンフィギュレーション モードを開始します。

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

例：

```
RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0
```

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、管理イーサネット インターフェイスの名前とインスタンスを指定します。

ステップ 3 **speed {10 | 100 |}**

例：

```
RP/0/RP0/CPU0:router(config-if)# speed 100
```

インターフェイス速度 **speed** パラメータを設定します。

有効なオプションは、10 Mbps または 100 Mbps です。

- (注)
- デフォルトの管理イーサネット インターフェイス速度はオートネゴシエーションされません。
 - システムをオートネゴシエーションされたデフォルトの速度に戻すには、**no speed** コマンドを使用します。

ステップ 4 **end** または **commit**

例：

```
RP/0/RP0/CPU0:router(config-if)# end
```

または

```
RP/0/RP0/CPU0:router(config-if)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?  
[cancel]:
```

- **yes** と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
- **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻りません。変更はコミットされません。

- `cancel` と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを継続するには、`commit` コマンドを使用します。

管理イーサネット インターフェイスの MAC アドレスの変更

RPに対応した管理イーサネットインターフェイスのMAC層アドレスを設定するには、次の作業を行います。

手順の概要

1. `configure`
2. `interface MgmtEthinterface-path-id`
3. `mac-addressaddress`
4. `end` または `commit`

手順の詳細

ステップ 1 `configure`

例：

```
RP/0/RP0/CPU0:router# configure
グローバル コンフィギュレーション モードを開始します。
```

ステップ 2 `interface MgmtEthinterface-path-id`

例：

```
RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0
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、管理イーサネット インターフェイスの名前とインスタンスを指定します。
```

ステップ 3 `mac-addressaddress`

例：

```
RP/0/RP0/CPU0:router(config-if)# mac-address 0001.2468.ABCD
管理イーサネット インターフェイスのMAC層アドレスを設定します。
```

- (注) • デバイスをデフォルトのMACアドレスに戻すには、`no mac-address` コマンドを使用します。

ステップ4 end または commit

例：

```
RP/0/RP0/CPU0:router(config-if)# end  
または
```

```
RP/0/RP0/CPU0:router(config-if)# commit  
設定変更を保存します。
```

- **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?  
[cancel]:
```

- **yes** と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
- **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
- **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

管理イーサネットインターフェイス設定の確認

管理イーサネットインターフェイスの設定変更を確認するには、次の作業を行います。

手順の概要

1. **show interfaces MgmtEthinterface-path-id**
2. **show running-config interface MgmtEthinterface-path-id**

手順の詳細

ステップ1 show interfaces MgmtEthinterface-path-id

例：

```
RP/0/RP0/CPU0:router# show interfaces MgmtEth 0/RP0/CPU0/0  
管理イーサネットインターフェイス設定を表示します。
```

ステップ 2 show running-config interface MgmtEthinterface-path-id

例：

```
RP/0/RP0/CPU0:router# show running-config interface MgmtEth 0/RP0/CPU0/0
```

実行コンフィギュレーションを表示します。

管理イーサネットインターフェイスの設定に関する情報

管理イーサネットインターフェイスを設定するには、次の概念について理解している必要があります。

デフォルト インターフェイス設定

次の表に、デフォルトの管理イーサネットインターフェイス設定を示します。これらの設定は、手動設定により変更できます。デフォルト設定は、**show running-config** コマンド出力には表示されません。

表 2: 管理イーサネットインターフェイスのデフォルト設定

| パラメータ | デフォルト値 | 設定ファイルのエントリ |
|--------------|---|--|
| 速度 (Mbps 単位) | 速度はオートネゴシエーションされます。 | speed [10 100 1000] システムをオートネゴシエーションされた速度に戻すには、 no speed [10 100 1000] コマンドを使用します。 |
| デュプレックスモード | デュプレックスモードはオートネゴシエーションされません。 | duplex {full half} システムをオートネゴシエーションされたデュプレックス操作に戻すには、必要に応じて no duplex {full half} コマンドを使用します。 |
| MAC address | MAC アドレスは、ハードウェアに組み込みのアドレス (BIA) から読み取られます。 | mac-address address デバイスをデフォルトの MAC アドレスに戻すには、 no mac-address コマンドを使用します。 |



第 3 章

イーサネット インターフェイスの設定

このモジュールでは、イーサネット インターフェイスの設定について説明します。

10 ギガビット、40 ギガビット、100 ギガビットの分散型イーサネット アーキテクチャは、ネットワークに拡張性とパフォーマンスをもたらすとともに、サービスプロバイダーが高密度で高帯域幅のネットワークング ソリューションを提供できるようにします。これらのソリューションは、コア ルータやエッジ ルータ、レイヤ 2 および レイヤ 3 スイッチなど、POP 内の他のシステムとルータを相互接続するように設計されています。

- [イーサネット インターフェイスの設定, 17 ページ](#)
- [ギガビット イーサネット インターフェイスの設定, 17 ページ](#)
- [L2VPN イーサネット ポートの設定, 22 ページ](#)
- [イーサネットの設定に関する情報, 25 ページ](#)

イーサネット インターフェイスの設定

このモジュールでは、イーサネット インターフェイスの設定について説明します。

10 ギガビット、40 ギガビット、100 ギガビットの分散型イーサネット アーキテクチャは、ネットワークに拡張性とパフォーマンスをもたらすとともに、サービスプロバイダーが高密度で高帯域幅のネットワークング ソリューションを提供できるようにします。これらのソリューションは、コア ルータやエッジ ルータ、レイヤ 2 および レイヤ 3 スイッチなど、POP 内の他のシステムとルータを相互接続するように設計されています。

ギガビット イーサネット インターフェイスの設定

基本的なイーサネット インターフェイス設定を作成するには、次の手順を実行します。

手順の概要

1. **show version**
2. **show interfaces [HundredGigE | TenGigE]interface-path-id**
3. **configure**
4. **interface[HundredGigE | TenGigE]interface-path-id**
5. **ipv4 addressip-address mask**
6. **flow-control{bidirectional| egress| ingress}**
7. **mtubytes**
8. **mac-addressvalue1.value2.value3**
9. **negotiation auto**
10. **no shutdown**
11. **end** または **commit**
12. **show interfaces [HundredGigE | TenGigE]interface-path-id**

手順の詳細

ステップ 1 show version

例 :

```
RP/0/RP0/CPU0:router# show version
```

(任意) 現在のソフトウェア バージョンを表示します。また、ルータがモジュラ サービス カードを認識していることを確認する場合にも使用できます。

ステップ 2 show interfaces [HundredGigE | TenGigE]interface-path-id

例 :

```
RP/0/RP0/CPU0:router# show interface HundredGigE 0/0/1/1
```

(任意) 設定済みのインターフェイスを表示し、各インターフェイスポートのステータスを確認します。

このステップで使用できるインターフェイスの種類は次のとおりです。

- HundredGigE
- TenGigE

ステップ 3 configure

例 :

```
RP/0/RP0/CPU0: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 4 interface[HundredGigE | TenGigE]interface-path-id

例：

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE 0/0/1/1
```

インターフェイス コンフィギュレーション モードを開始し、イーサネット インターフェイス名と *rack/slot/module/port* 表記を指定します。このステップで使用できるインターフェイスの種類は次のとおりです。

- HundredGigE
- TenGigE

(注) • この例は、モジュラ サービス カード スロット 1 の 8 ポート 10 ギガビット イーサネット インターフェイスです。

ステップ 5 `ipv4 address ip-address mask`

例：

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
```

IP アドレスとサブネット マスクをインターフェイスに割り当てます。

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

ステップ 6 `flow-control {bidirectional | egress | ingress}`

例：

```
RP/0/RP0/CPU0:router(config-if)# flow control ingress
```

(任意) フロー制御のポーズ フレームの送信および処理をイネーブルにします。

- **egress** : 出力でフロー制御のポーズ フレームの送信をイネーブルにします。
- **ingress** : 入力で受信したポーズ フレームの処理をイネーブルにします。
- **bidirectional** : 出力でフロー制御のポーズ フレームの送信をイネーブルにし、入力で受信したポーズ フレームの処理をイネーブルにします。

ステップ 7 `mtu bytes`

例：

```
RP/0/RP0/CPU0:router(config-if)# mtu 1448
```

(任意) インターフェイスの MTU 値を設定します。

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

ステップ 8 mac-addressvalue1.value2.value3

例：

```
RP/0/RP0/CPU0:router(config-if)# mac address 0001.2468.ABCD
```

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

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

ステップ 9 negotiation auto

例：

```
RP/0/RP0/CPU0:router(config-if)# negotiation auto
```

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

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

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

ステップ 10 no shutdown

例：

```
RP/0/RP0/CPU0:router(config-if)# no shutdown
```

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

ステップ 11 end または commit

例：

```
RP/0/RP0/CPU0:router(config-if)# end
```

または

```
RP/0/RP0/CPU0:router(config-if)# commit
設定変更を保存します。
```

- **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?
[cancel]:
```

- **yes** と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
- **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
- **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ステップ 12 show interfaces [HundredGigE | TenGigE]interface-path-id

例：

```
RP/0/RP0/CPU0:router# show interfaces HundredGigE 0/0/1/1
(任意) ルータ上のインターフェイスに関する統計情報を表示します。
```

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

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface HundredGigE 0/0/1/1
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RP0/CPU0:router(config-if)# flow-control ingress
RP/0/RP0/CPU0:router(config-if)# mtu 1448
RP/0/RP0/CPU0:router(config-if)# mac-address 0001.2468.ABCD
RP/0/RP0/CPU0:router(config-if)# no shutdown
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes

RP/0/RP0/CPU0:router# show interfaces HundredGigE 0/5/0/24
HundredGigE0/5/0/24 is up, line protocol is up
  Interface state transitions: 1
  Hardware is HundredGigE, address is 6219.8864.e330 (bia 6219.8864.e330)
  Internet address is 3.24.1.1/24
  MTU 9216 bytes, BW 100000000 Kbit (Max: 100000000 Kbit)
    reliability 255/255, txload 3/255, rxload 3/255
  Encapsulation ARPA,
  Full-duplex, 100000Mb/s, link type is force-up
  output flow control is off, input flow control is off
  Carrier delay (up) is 10 msec
```

```

loopback not set,
Last link flapped 10:05:07
ARP type ARPA, ARP timeout 04:00:00
Last input 00:08:56, output 00:00:00
Last clearing of "show interface" counters never
5 minute input rate 1258567000 bits/sec, 1484160 packets/sec
5 minute output rate 1258584000 bits/sec, 1484160 packets/sec
 228290765840 packets input, 27293508436038 bytes, 0 total input drops
 0 drops for unrecognized upper-level protocol
Received 15 broadcast packets, 45 multicast packets
 0 runts, 0 giants, 0 throttles, 0 parity
 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
212467849449 packets output, 25733664696650 bytes, 0 total output drops
Output 23 broadcast packets, 15732 multicast packets
39 output errors, 0 underruns, 0 applique, 0 resets
 0 output buffer failures, 0 output buffers swapped out
 0 carrier transitions

```

L2VPN イーサネット ポートの設定

L2VPN イーサネット ポートを設定するには、次の手順を実行します。



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

ポイントツーポイントの擬似回線 `xconnect` を設定するには、『Implementing MPLS Layer 2 VPNs module of the L2VPN and Ethernet Services Configuration Guide』を参照してください。

レイヤ 2 サービス ポリシー、たとえば Quality of Service (QoS) をイーサネット インターフェイスに追加するには、該当するソフトウェアのコンフィギュレーションガイドを参照してください。

手順の概要

1. `configure`
2. `interface[HundredGigE|TenGigE]interface-path-id`
3. `l2transport`
4. `l2protocol cpsv {tunnel | reverse-tunnel}`
5. `l2protocol {cdp | pvst | stp | vtp} {[forward| tunnel][experimentalbits]|drop}`
6. `end` または `commit`
7. `show interfaces [HundredGigE |TenGigE]interface-path-id`

手順の詳細

ステップ 1 `configure`

例 :

```
RP/0/RP0/CPU0:router# configure
```

グローバル コンフィギュレーション モードを開始します。

ステップ 2 interface[HundredGigE| TenGigE]interface-path-id

例 :

```
RP/0/RP0/CPU0:router(config)# interface
HundredGigE 0/0/1/0
```

インターフェイス コンフィギュレーション モードを開始し、イーサネット インターフェイス名と *rack/slot/module/port* 表記を指定します。このステップで使用できるインターフェイスの種類は次のとおりです。

- HundredGigE
- TenGigE

ステップ 3 l2transport

例 :

```
RP/0/RP0/CPU0:router(config-if)# l2transport
```

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

ステップ 4 l2protocol cpsv {tunnel | reverse-tunnel}

例 :

```
RP/0/RP0/CPU0:router(config-if-l2)# l2protocol cpsv tunnel
```

プロトコル CDP、PVST+、STP、VTP のイーサネット インターフェイスでのレイヤ 2 プロトコル トンネリングとプロトコル データ ユニット (PDU) フィルタリングを設定します。

- **tunnel** : インターフェイスに入るときのフレームの L2PT カプセル化と、インターフェイスから出るときのフレームのカプセル化解除を指定します。
- **reverse-tunnel** : インターフェイスから出るときのフレームの L2PT カプセル化と、インターフェイスに入るときのフレームのカプセル化解除を指定します。

ステップ 5 l2protocol{cdp | pvst | stp | vtp} {[forward| tunnel][experimentalbits][drop]}

例 :

```
RP/0/RP0/CPU0:router(config-if-l2)# l2protocol stp tunnel
```

インターフェイスで、フィルタリングするレイヤ 2 プロトコル トンネリングおよびプロトコル データ ユニット (PDU) を設定します。

設定可能なプロトコルおよびオプションは次のとおりです。

- **cdp** : Cisco Discovery Protocol (CDP) のトンネリングおよびデータ ユニットのパラメータ。

- **pvst** : VLAN スパニングツリー プロトコルのトンネリングおよびデータ ユニットのパラメータを設定します。
- **stp** : スパニングツリー プロトコルのトンネリングおよびデータ ユニットのパラメータ。
- **vtp** : VLAN Trunk Protocol のトンネリングおよびデータ ユニットのパラメータ。
- **tunnel** : (任意) 指定されたプロトコルに関連付けられたパケットのトンネリングを行います。
- **experimentalbits**: (任意) 指定されたプロトコルの MPLS EXP ビットを変更します。
- **drop** : (任意) 指定されたプロトコルに関連するパケットをドロップします。

ステップ 6 end または commit

例 :

```
RP/0/RP0/CPU0:router(config-if-12)# end
または
```

```
RP/0/RP0/CPU0:router(config-if-12)# commit
設定変更を保存します。
```

- **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?
[cancel]:
```

- **yes** と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
- **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
- **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ステップ 7 show interfaces [HundredGigE |TenGigE]interface-path-id

例 :

```
RP/0/RP0/CPU0:router# show interfaces HundredGigE 0/0/1/1
(任意) ルータ上のインターフェイスに関する統計情報を表示します。
```

AC のポイントツーポイント疑似回線 xconnect を設定するには、次のマニュアルを参照してください。

- 『L2VPN and Ethernet Services Configuration Guide』
- 『VPN and Ethernet Services Command Reference』

次に、イーサネットインターフェイスでレイヤ 2 VPN AC を設定する例を示します。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface HundredGigE 0/0/1/1
RP/0/RP0/CPU0:router(config-if)# l2transport
RP/0/RP0/CPU0:router(config-if-l2)# l2protocol tunnel
RP/0/RP0/CPU0:router(config-if-l2)# commit
```

イーサネットの設定に関する情報

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

ここでは、次の情報について説明します。

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

次の表に、10 ギガビットイーサネットまたは 100 ギガビットイーサネットのモジュラ サービスカードおよび関連付けられた PLIM でインターフェイスをイネーブルにしたときに表示される、デフォルトのインターフェイス設定パラメータを示します。



(注)

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

表 3: 100 ギガビットイーサネット モジュラ サービス カードのデフォルト設定値

| パラメータ | 設定ファイルのエントリ | デフォルト値 |
|----------------|-----------------------|-----------------------|
| MAC accounting | mac-accounting | off |
| フロー制御 | flow-control | egress on ingress off |

| パラメータ | 設定ファイルのエントリ | デフォルト値 |
|-------------|--------------------|--|
| MTU | mtu | <ul style="list-style-type: none"> • 1514 バイト (通常のフレーム) • 1518 バイト (802.1Q タグ付きフレーム) • 1522 バイト (Q-in-Q フレーム) |
| MAC address | mac address | ハードウェア BIA (バーンドインアドレス) |

イーサネット インターフェイスのリンクのオートネゴシエーション

リンクのオートネゴシエーションによって、リンク セグメントを共有するデバイスは、最高のパフォーマンスモードの相互運用で自動的に設定されます。イーサネット インターフェイスでリンクのオートネゴシエーションをイネーブルにするには、インターフェイス コンフィギュレーション モードで **negotiation auto** コマンドを使用します。ラインカードのイーサネット インターフェイスで、リンクのオートネゴシエーションはデフォルトでディセーブルです。



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

次の表に、速度モードのさまざまな組み合わせ別のシステム パフォーマンスを示します。指定されたコマンドによってこのとおりにシステムが動作するには、インターフェイス上でオートネゴシエーションが設定済みであることが条件となります。

表 4: *duplex* コマンドと *speed* コマンドの関係

| duplex コマンド | speed コマンド | |
|--------------------|-------------------|-------------------------------------|
| full-duplex | no speed | 強制的に全二重モードが指定され、速度はオートネゴシエーションされます。 |
| full-duplex | speed 1000 | 強制的に全二重モードと 1000 Mbps が指定されます。 |
| full-duplex | speed 100 | 強制的に全二重モードと 100 Mbps が指定されます。 |

| duplex コマンド | speed コマンド | |
|-------------|------------|-------------------------------------|
| full-duplex | speed 10 | 強制的に全二重モードと 10 Mbps が指定されます。 |
| half-duplex | no speed | 強制的に半二重モードが指定され、速度はオートネゴシエーションされます。 |
| half-duplex | speed 1000 | 強制的に半二重モードと 1000 Mbps が指定されます。 |
| half-duplex | speed 100 | 強制的に半二重モードと 100 Mbps が指定されます。 |
| half-duplex | speed 10 | 強制的に半二重モードと 10 Mbps が指定されます。 |

イーサネット MTU

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

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

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



(注) IPv6 はフラグメンテーションをサポートしません。

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

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

イーサネット インターフェイスでのフロー制御

10 ギガビット イーサネット インターフェイスでのフロー制御は、フロー制御ポーズ フレームを定期的に送信する処理で構成されます。この処理は、標準の管理インターフェイスで使用される通常の全二重および半二重のフロー制御とは根本的に異なります。フロー制御は、入トラフィックについてのみアクティブ化または非アクティブ化することができます。出トラフィックについては自動的に実装されます。

802.1Q VLAN

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

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

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

VRRP

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

HSRP

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

HSRPは、特定の状況でIPトラフィックを中断しない切り替えをサポートし、ホストからは単一のルータを使用しているように見え、使用している第1ホップのルータに障害が発生した場合で

も接続を維持できるように設計されています。つまり、HSRPは、発信元のホストが第1ホップのルータのIPアドレスを動的に取得できない場合でも、第1ホップのルータの障害に対処できません。複数のルータがHSRPに参加し、連携して単一の仮想ルータであるように見せます。HSRPによって、確実に単一のルータが仮想ルータの代わりにパケットを転送します。エンドホストがそのパケットを仮想ルータに転送します。

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

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

サブインターフェイス

Cisco IOS XR ソフトウェアでは、デフォルトでインターフェイスはメインインターフェイスになっています。メインインターフェイスは、VLAN トランキングのコンテキストでのトランクという単語の用法と混同しないように、トランク インターフェイスとも呼ばれます。

3種類のトランク インターフェイスがあります。

- 物理
- Bundle

物理インターフェイスはルータがカードとその物理インターフェイスを認識する際に自動的に作成されます。ただし、バンドルインターフェイスは自動作成されません。これらはユーザに設定されたときに作成されます。

次の設定例は、作成されるトランク インターフェイスの例です。

- `interface HundredGigabitethernet 0/5/0/0`
- `interface bundle-ether 1`

サブインターフェイスとは、トランク インターフェイスの下に作成される論理インターフェイスです。

サブインターフェイスを作成するには、最初にトランク インターフェイスを指定する必要があります。サブインターフェイスは、この下に配置されます。バンドル インターフェイスについては、バンドルインターフェイスがまだ存在していない場合は作成する必要があります。これで、その下にサブインターフェイスを作成できるようになります。

作成するサブインターフェイスにサブインターフェイス番号を割り当てます。サブインターフェイス番号は、ゼロ以上の正の整数でなければなりません。1つのトランク インターフェイスの下の各サブインターフェイスに一意的な値が必要です。

サブインターフェイス番号は、連続している必要はなく、数値順でなくてもかまいません。たとえば、1つのトランク インターフェイスの下で次のサブインターフェイス番号を指定できます。

1001、0、97、96、100000

サブインターフェイスは、1個のトランクの下に同じサブインターフェイス番号を設定できません。

次の例では、スロット5のカードにトランク インターフェイス **HundredGigE 0/5/0/0** があります。この下に、サブインターフェイス **HundredGigE 0/5/0/0.0** が作成されます。

```
RP/0/RP0/CPU0:router# conf
Mon Sep 21 11:12:11.722 EDT
RP/0/RP0/CPU0:router(config)# interface HundredGigE 0/5/0/0.0
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RP0/CPU0:router(config-subif)# commit

RP/0/RP0/CPU0:Sep 21 11:12:34.819 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
committed by user 'root'. Use 'show configuration commit changes 1000000152' to view the
changes.

RP/0/RP0/CPU0:router(config-subif)# end

RP/0/RP0/CPU0:Sep 21 11:12:35.633 : config[65794]: %MGBL-SYS-5-CONFIG_I : Configured from
console by root
RP/0/RP0/CPU0:router#
```

show run コマンドは、トランク インターフェイスを最初に表示し、次に昇順の数値順にサブインターフェイスを表示します。

```
RP/0/RP0/CPU0:router# show run | begin HundredGigE 0/5/0/0
Mon Sep 21 11:15:42.654 EDT
Building configuration...
interface GigabitEthernet0/5/0/0
 shutdown
 !
interface GigabitEthernet0/5/0/0.0
 encapsulation dot1q 100
 !
interface GigabitEthernet0/5/0/1
 shutdown
 !
```

サブインターフェイスが初めて作成されたときは、ルータはそのインターフェイスがトランク インターフェイスと交換可能であると認識します（いくつかの例外があります）。新しいサブインターフェイスの設定をさらに行った後で、**show interface** コマンドを実行すると、そのサブインターフェイスが一意のカウンタとともに表示されます。

次に、トランク インターフェイス **HundredGigE 0/5/0/0** を表示出力し、その後にサブインターフェイス **HundredGigE 0/5/0/0.0** を表示出力する例を示します。

```
RP/0/RP0/CPU0:router# show interface HundredGigE 0/5/0/0
Mon Sep 21 11:12:51.068 EDT
GigabitEthernet0/5/0/0 is administratively down, line protocol is administratively down.
 Interface state transitions: 0
 Hardware is GigabitEthernet, address is 0024.f71b.0ca8 (bia 0024.f71b.0ca8)
 Internet address is Unknown
 MTU 1514 bytes, BW 1000000 Kbit
   reliability 255/255, txload 0/255, rxload 0/255
 Encapsulation 802.1Q Virtual LAN,
 Full-duplex, 1000Mb/s, SFXD, link type is force-up
 output flow control is off, input flow control is off
 loopback not set,
 ARP type ARPA, ARP timeout 04:00:00
 Last input never, output never
 Last clearing of "show interface" counters never
```

```

5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  0 packets input, 0 bytes, 0 total input drops
  0 drops for unrecognized upper-level protocol
Received 0 broadcast packets, 0 multicast packets
  0 runts, 0 giants, 0 throttles, 0 parity
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 packets output, 0 bytes, 0 total output drops
Output 0 broadcast packets, 0 multicast packets
  0 output errors, 0 underruns, 0 applique, 0 resets
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions

```

```

RP/0/RP0/CPU0:router# show interface HundredGigE 0/5/0/0.0
Mon Sep 21 11:12:55.657 EDT
GigabitEthernet0/5/0/0.0 is administratively down, line protocol is administratively down.
  Interface state transitions: 0
  Hardware is VLAN sub-interface(s), address is 0024.f71b.0ca8
  Internet address is Unknown
  MTU 1518 bytes, BW 1000000 Kbit
    reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation 802.1Q Virtual LAN, VLAN Id 100, loopback not set,
  ARP type ARPA, ARP timeout 04:00:00
  Last input never, output never
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 bytes, 0 total input drops
    0 drops for unrecognized upper-level protocol
  Received 0 broadcast packets, 0 multicast packets
    0 packets output, 0 bytes, 0 total output drops
  Output 0 broadcast packets, 0 multicast packets

```

この例では、2つのインターフェイスが同時に作成されます。最初にバンドル トランク インターフェイスが作成され、その後でサブインターフェイスがトランクに追加されます。

```

RP/0/RP0/CPU0:router# conf
Mon Sep 21 10:57:31.736 EDT
RP/0/RP0/CPU0:router(config)# interface Bundle-Ether1
RP/0/RP0/CPU0:router(config-if)# no shut
RP/0/RP0/CPU0:router(config-if)# interface bundle-Ether1.0
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RP0/CPU0:router(config-subif)# commit
RP/0/RP0/CPU0:Sep 21 10:58:15.305 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT : C
onfiguration committed by user 'root'. Use 'show configuration commit changes 10
00000149' to view the changes.
RP/0/RP0/CPU0:router# show run | begin Bundle-Ether1
Mon Sep 21 10:59:31.317 EDT
Building configuration..
interface Bundle-Ether1
!
interface Bundle-Ether1.0
 encapsulation dot1q 100
!

```

no interface コマンドを使用してサブインターフェイスを削除します。

```

RP/0/RP0/CPU0:router#
RP/0/RP0/CPU0:router# show run | begin HundredGigE0/5/0/0
Mon Sep 21 11:42:27.100 EDT
Building configuration...
interface GigabitEthernet0/5/0/0
 negotiation auto
!
interface HundredGigE0/5/0/0.0
 encapsulation dot1q 100
!
interface HundredGigE0/5/0/1
 shutdown
!
RP/0/RP0/CPU0:router# conf

```

```

Mon Sep 21 11:42:32.374 EDT
RP/0/RP0/CPU0:router(config)# no interface GigabitEthernet0/5/0/0.0
RP/0/RP0/CPU0:router(config)# commit
RP/0/RP0/CPU0:Sep 21 11:42:47.237 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT : Configuration
committed by user 'root'. Use 'show configuration commit changes 1000000159' to view the
changes.
RP/0/RP0/CPU0:router(config)# end
RP/0/RP0/CPU0:Sep 21 11:42:50.278 : config[65794]: %MGBL-SYS-5-CONFIG_I : Configured from
console by root
RP/0/RP0/CPU0:router# show run | begin GigabitEthernet0/5/0/0
Mon Sep 21 11:42:57.262 EDT
Building configuration...
interface HundredGigE0/5/0/0
 negotiation auto
!
interface HundredGigE0/5/0/1
 shutdown
!

```

レイヤ2サブインターフェイス（EFP）の拡張パフォーマンス モニタリング

ここでは、レイヤ2 インターフェイス カウンタの新しいサポートの概要について説明します。

パフォーマンス統計情報収集の新しいエンティティをサポートし、次のコマンドでレイヤ2 インターフェイス上に表示するために **interface basic-counters** キーワードが追加されました。

- **performance-mgmt statistics interface basic-counters**
- **performance-mgmt threshold interface basic-counters**
- **performance-mgmt apply statistics interface basic-counters**
- **performance-mgmt apply threshold interface basic-counters**
- **performance-mgmt apply monitor interface basic-counters**
- **show performance-mgmt monitor interface basic-counters**
- **show performance-mgmt statistics interface basic-counters**

performance-mgmt threshold interface basic-counters コマンドは、**show performance-mgmt statistics interface basic-counters** および **show performance-mgmt monitor interface basic-counters** コマンドでも表示される、レイヤ2 統計情報の属性値をサポートします。

| 属性 | 説明 |
|------------------|------------------------------|
| InOctets | 受信したバイト (64 ビット) |
| InPackets | 受信したパケット (64 ビット) |
| InputQueueDrops | 入力キューのドロップ (64 ビット) |
| InputTotalDrops | インバウンドの廃棄された適正なパケット (64 ビット) |
| InputTotalErrors | インバウンドの廃棄された不正なパケット (64 ビット) |
| OutOctets | 送信したバイト (64 ビット) |

| 属性 | 説明 |
|-------------------|-------------------------------|
| OutPackets | 送信したパケット (64 ビット) |
| OutputQueueDrops | 出力キューのドロップ (64 ビット) |
| OutputTotalDrops | アウトバウンドの廃棄された適正なパケット (64 ビット) |
| OutputTotalErrors | アウトバウンドの廃棄された不正なパケット (64 ビット) |

その他のパフォーマンス管理の機能拡張

Cisco IOS XR ソフトウェアには、次の追加のパフォーマンス管理の拡張機能が含まれています。

- **performance-mgmt statistics interface** コマンドの新しい **history-persistent** キーワードオプションを使用して、パフォーマンス統計情報の新しいプロセスの再起動やルートプロセッサ (RP) のフェールオーバーを通してパフォーマンス管理の履歴統計情報を保持できます。
- **performance-mgmt resources dump local** コマンドを使用して、ローカル ファイルにパフォーマンス管理統計情報を保存できます。
- 一致する文字列を指定する複数の正規表現インデックスを含む正規表現グループ (**performance-mgmt regular-expression** コマンド) の定義で、パフォーマンス管理インスタンスをフィルタリングできます。 **performance-mgmt statistics interface** または **performance-mgmt thresholds** インターフェイス コマンドで、1つまたは複数の統計情報またはしきい値テンプレートに、定義された正規表現グループを適用します。



第 4 章

リンクバンドルの設定

リンクバンドル機能を使用すると、複数のポイントツーポイントリンクを1つの論理リンクにグループ化して、2台のルータ間により高い双方向帯域幅、冗長性とロードバランシングを提供できます。仮想インターフェイスは、バンドルリンクに割り当てられます。コンポーネントリンクは仮想インターフェイスに動的に追加および削除できます。

仮想インターフェイスは、IPアドレスやリンクバンドルで使用されるその他のソフトウェア機能を設定できる、単一のインターフェイスとして扱われます。リンクバンドルに送信されたパケットは、バンドル内のリンクの1つに転送されます。

リンクバンドルは、まとめてバンドルされて単一のリンクとして機能するポートのグループにすぎません。リンクバンドルの利点は次のとおりです。

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

1つのバンドル内の個別リンクは、すべて同じタイプと同じ速度でなければなりません。

Cisco IOS XR ソフトウェアは、次に示すイーサネットインターフェイスのバンドル形成方法をサポートします。

- IEEE 802.3ad : バンドル内のすべてのメンバーリンクの互換性を確保するため、Link Aggregation Control Protocol (LACP) を採用した標準テクノロジー。互換性がないリンクや障害になったリンクは、バンドルから自動的に削除されます。
- イーサチャンネル : ユーザがリンクを設定してバンドルに追加できるシスコの専用テクノロジー。バンドル内のリンクに互換性があるかどうかを確認するための仕組みはありません。
- [イーサネットリンクバンドルの機能および互換性のある特性, 36 ページ](#)
- [イーサネットリンクバンドルの設定, 37 ページ](#)

- [イーサネットリンクバンドルでの EFP ロード バランシングの設定, 42 ページ](#)
- [イーサネットリンクバンドル上の VLAN, 44 ページ](#)
- [VLAN バンドルの設定, 45 ページ](#)
- [リンクバンドルの設定に関する情報, 50 ページ](#)

イーサネットリンクバンドルの機能および互換性のある特性

次のリストに、イーサネットリンクバンドルのプロパティと制約事項を示します。

- LACP (Link Aggregation Control Protocol) を使用するかにかかわらず、すべてのタイプのイーサネットインターフェイスをバンドルできます。
- イーサネットリンクバンドルは最大 64 の物理リンクをサポートできます。64 本を超えるリンクをバンドルに追加した場合は、そのリンクのうち 64 本だけが distributing 状態になり、残りのリンクは待機状態になります。
- 単一のルータは最大 63 のバンドルをサポートできます。
- 異なる速度が混在するバンドルでは、異なる帯域幅のメンバリンクを単一のバンドル内のアクティブメンバとして設定できます。バンドルメンバの帯域幅の比率は 10 を超えることはできません。また、バンドルの重みの合計は 64 を超えることはできません。たとえば、100 Gbps リンクと 10 Gbps リンクはバンドル内でアクティブメンバとして設定できます。異なる速度が混在するバンドルでは、以下も可能です。
 - 帯域幅の重み付けに基づくメンバリンクのロードバランシング。
 - すべてのユニキャストフローに比例したロードバランシングのサポート。
 - すべての VPLS フラッドイングとレイヤ 2 マルチキャストフローの帯域幅重み付けに基づくロードバランシングのサポート。
- 異なる速度が混在するバンドルでは、バンドルの重みの合計がバンドル内のメンバ数よりも大きくなります。これは、重みは最小アクティブ数の帯域幅を表しているためです。
- 各バンドルメンバの重みは、帯域幅が最も低いメンバに対する帯域幅の比率です。バンドルの重みの合計は、各バンドルメンバの重みか、または相対帯域幅の合計です。バンドルメンバの重みが 1 以上で 10 以下であるため、混在バンドルの場合のバンドル内のリンクの総メンバは 64 未満になります。
- 物理層とリンク層の設定は、バンドルの個々のメンバーリンクに対して実行します。
- ネットワーク層プロトコルおよび上位層のアプリケーションの設定は、バンドル自体に対して実行します。
- IPv4 および IPv6 アドレッシングがイーサネットリンクバンドル上でサポートされます。

- バンドルは、管理上イネーブルまたはディセーブルにできます。
- バンドル内のそれぞれのリンクは、管理上イネーブルまたはディセーブルにできます。
- イーサネットリンクバンドルは、イーサネットチャンネルと同様の方法で作成され、両方のエンドシステムで同じコンフィギュレーションを入力します。
- バンドルに対して設定されたMACアドレスは、そのバンドル内の各リンクのMACアドレスになります。
- LACPが設定されている場合、バンドル内の各リンクでは、異なるメンバに対して異なるキープアライブ周期を設定できます。
- ロードバランシング（メンバーリンク間のデータの分散）は、パケットではなくフロー単位で実行されます。データはバンドル対するそのリンクの帯域幅に比例して、リンクに配信されます。
- QoSがサポートされており、各バンドルメンバーに均等に適用されます。
- CDPキープアライブやHDLCキープアライブなどのリンクレイヤプロトコルは、バンドル内の各リンク上で独立して動作します。
- 1つのバンドル内のすべてのリンクは、同じ2台のシステム上で終端する必要があります。
- バンドルされたインターフェイスはポイントツーポイントです。
- リンクがバンドル内で **distributing** 状態になるには、その前にアップ状態なる必要があります。
- バンドルインターフェイスには、物理リンクとVLANサブインターフェイスのみを含めることができます。トンネルは、バンドルのメンバにできません。
- マルチキャストトラフィックは、バンドルのメンバー上でロードバランスされます。特定のフローに対し、内部処理によってメンバリンクが選択され、そのフローのすべてのトラフィックがそのメンバ上で送信されます。

イーサネットリンクバンドルの設定

ここでは、イーサネットリンクバンドルの設定方法について説明します。



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

手順の概要

1. **configure**
2. **interface Bundle-Ether***bundle-id*
3. **ipv4 address***ipv4-address mask*
4. **bundle minimum-active bandwidth***kbps*
5. **bundle minimum-active links***links*
6. **bundle maximum-active links***links*[**hot-standby**]
7. **lacp fast-switchover**
8. **exit**
9. **interface** { | **TenGigE**} *interface-path-id*
10. **bundle id***bundle-id* [**mode** {**active** | **on** | **passive**}]
11. **bundle port-priority***priority*
12. **no shutdown**
13. **exit**
14. **bundle id***bundle-id* [**mode**{ **active** | **passive** | **on**}] **no shutdown***exit*
15. **end** または **commit**
16. **exit**
17. **exit**
18. 接続のリモートエンドでステップ 1 から 15 を実行します。
19. **show bundle** **Bundle-Ether***bundle-id*
20. **show lacp bundle** **Bundle-Ether***bundle-id*

手順の詳細

ステップ 1 **configure**

例 :

```
RP/0/RP0/CPU0:router# configure
グローバル コンフィギュレーション モードを開始します。
```

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

例 :

```
RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3
```

指定したバンドル ID を使用して新しいイーサネットリンクバンドルを作成します。指定できる範囲は 1 ~ 65535 です。

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

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

例：

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0
```

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

- (注) • IPアドレスが必要なのは、レイヤ3のバンドルインターフェイスのみです。

ステップ 4 `bundle minimum-active bandwidth`*kbps*

例：

```
RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 580000
```

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

ステップ 5 `bundle minimum-active links`*links*

例：

```
RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2
```

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

ステップ 6 `bundle maximum-active links`*links[hot-standby]*

例：

```
RP/0/RP0/CPU0:router(config-if)# bundle maximum-active links 1 hot-standby
```

(任意) バンドルで 1:1 保護回線を実装します。これにより、バンドル内で最も優先順位が高いリンクがアクティブになり、2番目に優先順位が高いリンクがスタンバイになります。また、アクティブおよびスタンバイのLACP対応のリンクの間でのスイッチオーバーが、専用の最適化に従って実装されることを指定します。

- (注) • アクティブおよびスタンバイリンクの優先順位は、**bundle port-priority** コマンドの値で決まります。

ステップ 7 `lACP fast-switchover`

例：

```
RP/0/RP0/CPU0:router:router(config-if)# lacp fast-switchover
```

(任意) LACP が動作するメンバーリンクを持つバンドル上で 1:1 のリンク保護をイネーブルにすると (**bundle maximum-active links** コマンドの値に 1 を設定)、LACP 状態マシンの `wait-while` タイマーをディセーブルにできます。このタイマーをディセーブルにすると、スタンバイモードのバンドルメンバーリンクで、正常状態のネゴシエーションが高速になるため、障害になったアクティブリンクからスタンバイリンクへのスイッチオーバーが高速になります。

ステップ 8 `exit`

例：

```
RP/0/RP0/CPU0:router(config-if)# exit
```

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

ステップ 9 `interface` { | **TenGigE** } `interface-path-id`

例：

```
RP/0/RP0/CPU0:router(config)# interface TenGigE 1/0/0/HundredGigE 0/0/1/1
```

指定したインターフェイスに対してインターフェイスコンフィギュレーションモードを開始します。

GigabitEthernet キーワードまたは **TenGigE** キーワードを入力して、インターフェイスの種類を指定します。`interface-path-id` 引数には、`rack/slot/module` 形式でノード ID を指定します。

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

例：

```
RP/0/RP0/CPU0:router(config-if)# bundle-id 3
```

指定したバンドルにリンクを追加します。

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

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

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

ステップ 11 `bundle port-priority` `priority`

例：

```
RP/0/RP0/CPU0:router(config-if)# bundle port-priority 1
```

(任意) **bundle maximum-active links** コマンドに 1 を設定する場合、アクティブリンクの優先順位を最も高くし (最も小さい値)、スタンバイリンクの優先順位を 2 番目に高く (次に小さい値) する必要があります。たとえば、アクティブリンクの優先順位を 1 に設定し、スタンバイリンクの優先順位を 2 に設定します。

ステップ 12 `no shutdown`

例：

```
RP/0/RP0/CPU0:router(config-if)# no shutdown
```

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

ステップ 13 `exit`

例：

```
RP/0/RP0/CPU0:router(config-if)# exit
```

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

ステップ 14 **bundle id** *bundle-id* [**mode** {**active** | **passive** | **on**}] **no shutdown** **exit**

例：

```
RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/2/1
```

```
RP/0/RP0/CPU0:router(config-if)# bundle id 3
```

```
RP/0/RP0/CPU0:router(config-if)# bundle port-priority 2
```

```
RP/0/RP0/CPU0:router(config-if)# no shutdown
```

```
RP/0/RP0/CPU0:router(config-if)# exit
```

```
RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/2/3
```

```
RP/0/RP0/CPU0:router(config-if)# bundle id 3
```

```
RP/0/RP0/CPU0:router(config-if)# no shutdown
```

```
RP/0/RP0/CPU0:router(config-if)# exit
```

(任意) バンドルにさらにリンクを追加するには、ステップ 8 から 11 を繰り返します。

ステップ 15 **end** または **commit**

例：

```
RP/0/RP0/CPU0:router(config-if)# end
```

または

```
RP/0/RP0/CPU0:router(config-if)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?
```

```
[cancel]:
```

- **yes** と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
- **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻りません。変更はコミットされません。
- **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ステップ 16 exit

例：

```
RP/0/RP0/CPU0:router(config-if)# exit
```

インターフェイス コンフィギュレーション モードを終了します。

ステップ 17 exit

例：

```
RP/0/RP0/CPU0:router(config)# exit
```

グローバル コンフィギュレーション モードを終了します。

ステップ 18 接続のリモートエンドでステップ 1 から 15 を実行します。
リンクバンドルの他端をアップ状態にします。

ステップ 19 show bundle Bundle-Etherbundle-id

例：

```
RP/0/RP0/CPU0:router# show bundle Bundle-Ether 3
```

(任意) 指定したイーサネットリンクバンドルに関する情報を表示します。

ステップ 20 show lacp bundle Bundle-Etherbundle-id

例：

```
RP/0/RP0/CPU0:router# show lacp bundle  
Bundle-Ether 3
```

(任意) LACP ポートとそのピアに関する詳細情報を表示します。

イーサネットリンクバンドルでの EFP ロードバランシングの設定

ここでは、イーサネットリンクバンドルでイーサネットフローポイント (EFP) ロードバランシングを設定する情報を説明します。

デフォルトでは、イーサネットフローポイント (EFP) ロードバランシングはイネーブルです。ただし、バンドルの固定メンバのすべての出力トラフィックを、同じ物理メンバリンクを介して送信されるように設定できます。この設定は、レイヤ 2 転送 (**l2transport**) をイネーブルにしたイーサネットバンドルサブインターフェイスでしか使用できません。



(注) バンドルのアクティブメンバが変更されると、バンドルへのトラフィックは、設定値と一致するハッシュ値を持つ別の物理リンクにマッピングされる場合があります。

手順の概要

1. **configure**
2. **hw-module load-balance bundle l2-service l3-params**
3. **interface Bundle-Etherbundle-idl2transport**
4. **bundle load-balance hashhash-value [auto]**
5. **end** または **commit**

手順の詳細

ステップ 1 **configure**

例：

```
RP/0/RP0/CPU0:router# configure
```

グローバル コンフィギュレーション モードを開始します。

ステップ 2 **hw-module load-balance bundle l2-service l3-params**

例：

```
RP/0/RP0/CPU0:router(config)# hw-module load-balance bundle l2-service l3-params
```

(任意) レイヤ 2 リンクバンドルでのレイヤ 3 ロードバランシングをイネーブルにします。

ステップ 3 **interface Bundle-Etherbundle-idl2transport**

例：

```
RP/0/RP0/CPU0:router(config)# interface Bundle-Ether 3 l2transport
```

指定したバンドル ID を使用し、レイヤ 2 転送をイネーブルにして、新しいイーサネットリンクバンドルを作成します。

指定できる範囲は 1 ~ 65535 です。

ステップ 4 **bundle load-balance hashhash-value [auto]**

例：

```
RP/0/RP0/CPU0:router(config-subif)# bundle load-balancing hash 1
```

または

```
RP/0/RP0/CPU0:router(config-subif)# bundle load-balancing hash auto
```

バンドルの固定メンバのすべての出力トラフィックを、同じ物理メンバリンクを通過するように設定します。

- **hash-value** : このバンドルのすべての出力トラフィックが通過する物理メンバリンクを指定する数値。値は 1 ~ 8 です。
- **auto** : このバンドルのすべての出力トラフィックが通過する物理メンバリンクが自動的に選択されます。

ステップ 5 end または commit

例：

```
RP/0/RP0/CPU0:router(config-if)# end
または
```

```
RP/0/RP0/CPU0:router(config-if)# commit
設定変更を保存します。
```

- **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。Uncommitted changes found, commit them before exiting (yes/no/cancel)?
- **yes** と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
- **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
- **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

次に、バンドルの固定メンバのすべての出力トラフィックが、同じ物理メンバリンクを介して自動的に送信されるように設定する例を示します。

```
RP/0/RP0/CPU0:router# configuration terminal
RP/0/RP0/CPU0:router(config)# interface bundle-ether 1.1 l2transport
RP/0/RP0/CPU0:router(config-subif)#bundle load-balancing hash auto
RP/0/RP0/CPU0:router(config-subif)#
```

次に、バンドルの固定メンバのすべての出力トラフィックが、指定した物理メンバリンクを介して送信されるように設定する例を示します。

```
RP/0/RP0/CPU0:router# configuration terminal
RP/0/RP0/CPU0:router(config)# interface bundle-ether 1.1 l2transport
RP/0/RP0/CPU0:router(config-subif)#bundle load-balancing hash 1
RP/0/RP0/CPU0:router(config-subif)#
```

イーサネット リンク バンドル上の VLAN

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

- 各バンドルに許可される VLAN の最大数は、4096 です。
- 各ルータに許可されるバンドル VLAN の最大数は、4000 です。



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

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

interface Bundle-Ether*interface-bundle-id.subinterface*

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

VLAN サブインターフェイスでは、イーサネット フロー ポイント (EFP) およびレイヤ 3 サービスなどの複数のレイヤ 2 フレーム タイプおよびサービスをサポートできます。

レイヤ 2 EFP は次のように設定します。

```
interface bundle-ether instance.subinterface l2transport. encapsulation dot1q xxxxx
```

レイヤ 3 VLAN サブインターフェイスは次のように設定します。

```
interface bundle-ether instance.subinterface, encapsulation dot1q xxxxx
```



(注) レイヤ 2 およびレイヤ 3 インターフェイス間の違いは、**l2transport** キーワードです。両方のタイプのインターフェイスが **dot1q encapsulation** を使用します。

VLAN バンドルの設定

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

手順の概要

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

手順の詳細

ステップ 1 イーサネット バンドルを作成します。

ステップ 2 VLAN サブインターフェイスを作成し、イーサネット バンドルに割り当てます。

ステップ 3 イーサネット リンクをイーサネット バンドルに割り当てます。

これらの作業について、以降の手順で詳しく説明します。



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

手順の概要

1. **configure**
2. **interface Bundle-Ether***bundle-id*
3. **ipv4 address***ipv4-address mask*
4. **bundle minimum-active bandwidth***kbps*
5. **bundle minimum-active links***links*
6. **bundle maximum-active links***links[hot-standby]*
7. **exit**
8. **interface Bundle-Ether***bundle-id.vlan-id*
9. **encapsulation dot1q**
10. **ipv4 address***ipv4-address mask*
11. **no shutdown**
12. **exit**
13. ステップ 2 で作成したバンドルにさらに VLAN を追加するには、ステップ 9 から 12 を繰り返します
14. **end** または **commit**
15. **exit**
16. **exit**
17. **configure**
18. **interface {GigabitEthernet | TenGigE}***interface-path-id*
19. **lACP fast-switchover**

手順の詳細

ステップ 1 **configure**

例：

```
RP/0/RP0/CPU0:router# configure
グローバル コンフィギュレーション モードを開始します。
```

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

例：

```
RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3
```

新しいイーサネット リンク バンドルを作成し名前を付与します。

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

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

例：

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0
```

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

ステップ4 **bundle minimum-active bandwidth***kbps*

例：

```
RP/0/RP0/CPU0:router(config-if)# bundle minimum-active bandwidth 580000
```

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

ステップ5 **bundle minimum-active links***links*

例：

```
RP/0/RP0/CPU0:router(config-if)# bundle minimum-active links 2
```

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

ステップ6 **bundle maximum-active links***links[hot-standby]*

例：

```
RP/0/RP0/CPU0:router(config-if)# bundle maximum-active links 1 hot-standby
```

(任意) バンドルで 1:1 のリンク保護を実装します。これにより、バンドル内で最も優先順位が高いリンクがアクティブになり、2 番目に優先順位が高いリンクがスタンバイになります。また、アクティブおよびスタンバイの LACP 対応のリンクの間でのスイッチオーバーが、専用の最適化に従って実装されることを指定します。

(注) アクティブおよびスタンバイ リンクの優先順位は、**bundle port-priority** コマンドの値で決まります。

ステップ7 **exit**

例：

```
RP/0/RP0/CPU0:router(config-if)# exit
```

インターフェイス コンフィギュレーション サブモードを終了します。

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

例：

```
RP/0/RP0/CPU0:router#(config)# interface Bundle-Ether 3.1
```

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

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

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

(注) *.vlan-id* 引数を **interface Bundle-Etherbundle-id** コマンドに追加すると、サブインターフェイスコンフィギュレーションモードが開始されます。

ステップ 9 encapsulation dot1q

例：

```
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 100, untagged
```

インターフェイスのレイヤ 2 カプセル化を設定します。

(注) **dot1q vlan** コマンドは、Cisco ASR 9000 シリーズルータでは **encapsulation dot1q** コマンドに置き換えられます。引き続き、下位互換性のために使用可能ですが、レイヤ 3 インターフェイスだけが対象です。

ステップ 10 ipv4 address ipv4-address mask

例：

```
RP/0/RP0/CPU0:router#(config-subif)# ipv4 address 10.1.2.3/24
```

IP アドレスおよびサブネットマスクをサブインターフェイスに割り当てます。

ステップ 11 no shutdown

例：

```
RP/0/RP0/CPU0:router#(config-subif)# no shutdown
```

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

ステップ 12 exit

例：

```
RP/0/RP0/CPU0:router(config-subif)# exit
```

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

ステップ 13 ステップ 2 で作成したバンドルにさらに VLAN を追加するには、ステップ 9 から 12 を繰り返します (任意) バンドルにさらにサブインターフェイスを追加します。

ステップ 14 end または commit

例：

```
RP/0/RP0/CPU0:router(config-subif)# end
```

または

```
RP/0/RP0/CPU0:router(config-subif)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。

```
Uncommitted changes found, commit them before
exiting(yes/no/cancel)?
[cancel]:
```

-**yes** と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。

-**no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。

-**cancel** と入力すると、ルータは現在のコンフィギュレーションセッションで継続されます。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。

- 実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ステップ 15 exit

例：

```
RP/0/RP0/CPU0:router(config-subif)# end
```

インターフェイス コンフィギュレーション モードを終了します。

ステップ 16 exit

例：

```
RP/0/RP0/CPU0:router(config)# exit
```

グローバル コンフィギュレーション モードを終了します。

ステップ 17 configure

例：

```
RP/0/RP0/CPU0:router # configure
```

グローバル コンフィギュレーション モードを開始します。

ステップ 18 interface {GigabitEthernet | TenGigE} interface-path-id

例：

```
RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 1/0/0/0
```

バンドルに追加するイーサネット インターフェイスのインターフェイス コンフィギュレーション モードを開始します。

GigabitEthernet キーワードまたは **TenGigE** キーワードを入力して、インターフェイスの種類を指定します。*interface-path-id* 引数には、*rack/slot/module* 形式でノード ID を指定します。

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

ステップ 19 lacp fast-switchover

例：

```
RP/0/RP0/CPU0:router(config-if)# lacp fast-switchover
```

(任意) LACP が動作するメンバー リンクを持つバンドル上で 1:1 のリンク保護をイネーブルにすると (**bundle maximum-active links** コマンドの値に 1 を設定)、LACP 状態マシンの **wait-while** タイマーをディセーブルにできます。このタイマーをディセーブルにすると、スタンバイ モードのバンドルメンバーリンクで、正常状態のネゴシエーションが高速になるため、障害になったアクティブリンクからスタンバイリンクへのスイッチオーバーが高速になります。

リンクバンドルの設定に関する情報

リンクバンドルを設定するには、次の概念について理解する必要があります。

LACP を通じたリンク集約

異なるモジュラ サービス カードおよび同じサービス カード内の SPA 上のインターフェイスを集約することで、冗長性が提供され、インターフェイスまたはモジュラサービスカードで障害が発生したときに、トラフィックをすばやく他のメンバー リンクにリダイレクトできます。

オプションの Link Aggregation Control Protocol (LACP) は IEEE 802 規格で定義されています。LACP では、2 台の直接接続されたシステム (ピア) 間で通信し、バンドルメンバーの互換性が確認されます。ピアは、別のルータまたはスイッチのいずれかです。LACP は、リンクバンドルの動作状態を監視し、次のことを確認します。

- すべてのリンクが同じ 2 台のシステム上で終端していること。
- 両方のシステムがリンクを同じバンドルの一部と見なしていること。
- すべてのリンクがピア上で適切に設定されていること

LACP で送信されるフレームの内容は、ローカルポート状態と、ローカルから見たパートナーシステムの状態です。これらのフレームが解析され、両方のシステムが同調していることが確認されます。

IEEE 802.3ad 規格

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

バンドルメンバーとして設定された各リンクに対し、リンクバンドルの各エンドをホストするシステム間で、次の情報が交換されます。

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

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

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

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

バンドルに追加されている最初のリンクの MAC アドレスがバンドル自体の MAC アドレスになります。そのリンク (バンドルに追加されている最初のリンク) がバンドルから削除されるか、ユーザが別の MAC アドレスを設定するまで、この MAC アドレスが使用されます。バンドルの MAC アドレスは、バンドルトラフィックを通過させる際にすべてのメンバーリンクによって使用されます。バンドルに対して設定されたすべてのユニキャストアドレスまたはマルチキャストアドレスも、すべてのメンバー リンクで設定されます。



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

ロードバランシング

ロードバランシングは、ルータのレイヤ 3 ルーティング情報に基づいて、複数のリンクにトラフィックを分配する転送メカニズムです。フローごとのロードバランシングは、バンドルのすべてのリンクでサポートされます。この方法では、ルータが、ハッシュ計算で決定されたバンドル内のリンクの 1 つを経由してパケットを配信することによって、ロードシェアリングが実行されます。ハッシュ計算は特定のパラメータに基づいたリンク選択のアルゴリズムです。

標準のハッシュ計算は、次のパラメータを使用する 3 タプルハッシングです。

- IP 送信元アドレス

- IP 宛先アドレス
- ルータ ID

レイヤ 3 およびレイヤ 4 のパラメータに基づいて 7 タプル ハッシングも設定できます。

- IP 送信元アドレス
- IP 宛先アドレス
- ルータ ID
- 入力インターフェイス
- IP プロトコル
- レイヤ 4 送信元ポート
- レイヤ 4 宛先ポート

フローごとのロードバランシングと 3 タプルハッシングをイネーブルにすると、特定の送信元と宛先のペア間のすべてのパケットは、使用可能なリンクが複数あるにもかかわらず同じリンクを通過します。フローごとのロードバランシングは、特定の送信元と宛先ペアのパケットが順序どおりに到達できるようにします。



- (注) マルチキャストトラフィックの場合、入力転送はファブリック マルチキャストグループ ID (FGID) に基づきます。バンドル上の出力転送はバンドルのロードバランシングに基づきます。

リンクバンドルの設定の概要

リンクバンドルの設定の一般的な概要を次のステップで示します。リンクをバンドルに追加する前に、リンクから以前のネットワーク層コンフィギュレーションをすべてクリアする必要があります。ことに注意してください。

- 1 グローバルコンフィギュレーションモードで、リンクバンドルを作成します。イーサネットリンクバンドルを作成するには、**interface Bundle-Ether** コマンドを入力します。
- 2 **ipv4 address** コマンドを使用して、IP アドレスとサブネットマスクを仮想インターフェイスに割り当てます。
- 3 インターフェイスコンフィギュレーションサブモードで **bundle id** コマンドを使用し、ステップ 1 で作成したバンドルにインターフェイスを追加します。
1 つのバンドルに最大 64 個のリンクを追加できます。
- 4 バンドルに対してオプションで 1:1 のリンク保護を実装できます。そのためには、**bundle maximum-active links** コマンドに 1 を設定します。この設定を行うと、バンドルで優先順位が最も高いリンクがアクティブになり、優先順位が 2 番目に高いリンクがスタンバイになります。

(リンクのプライオリティは**bundle port-priority** コマンドの値に基づきます)。アクティブリンクに障害が発生した場合は、スタンバイリンクがすぐにアクティブリンクになります。



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

RP スイッチオーバー時のノンストップフォワーディング

Cisco IOS XR ソフトウェアは、アクティブとスタンバイのペアの RP カード間でのスイッチオーバー時のノンストップフォワーディングをサポートしています。ノンストップフォワーディングを使用すると、スイッチオーバーが発生したときにリンクバンドルの状態が変化しません。

たとえば、アクティブな RP に障害が発生した場合、スタンバイ RP が動作可能になります。障害が発生した RP の設定、ノード状態、チェックポイントデータがスタンバイ RP に複製されます。スタンバイ RP がアクティブ RP になった時に、バンドルされたインターフェイスがすべて存在しています。



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

リンク スイッチオーバー

デフォルトでは、バンドル内の最大 64 のリンクがアクティブにトラフィックを転送できます。バンドル内の 1 つのメンバーリンクが障害になると、トラフィックは動作可能な残りのメンバーリンクにリダイレクトされます。

バンドルに対してオプションで 1:1 のリンク保護を実装できます。そのためには、**bundle maximum-active links** コマンドに 1 を設定します。そうすることで、1 つのアクティブリンクと 1 つ以上の専用のスタンバイリンクが指定されます。アクティブリンクが障害になるとスイッチオーバーが発生し、スタンバイリンクがすぐにアクティブになり、中断のないトラフィックが保証されます。

アクティブリンクとスタンバイリンクで LACP が動作している場合、IEEE 規格に基づくスイッチオーバー (デフォルト) か、専用の高速な最適化されたスイッチオーバーを選択できます。アクティブリンクとスタンバイリンクで LACP が動作していない場合、専用の最適化されたスイッチオーバー オプションが使用されます。

使用するスイッチオーバーの種類にかかわらず、**wait-while** タイマーをディセーブルにできます。これにより、スタンバイリンクの状態ネゴシエーションが高速になり、障害になったアクティブリンクからスタンバイリンクへのスイッチオーバーが高速になります。

そのためには、**lACP fast-switchover** コマンドを使用します。



第 5 章

仮想ループバックおよびヌルインターフェイスの設定

このモジュールでは、ループバックおよびヌルインターフェイスの設定について説明します。ループバック インターフェイスとヌルインターフェイスは、仮想インターフェイスと見なされます。

仮想インターフェイスは、ルータ内部の論理パケットスイッチング エンティティです。仮想インターフェイスにはグローバル スコープがありますが、関連付けられた位置はありません。代替として、仮想インターフェイスは名前のあとにグローバルに一意的な数字による ID を持ちます。たとえば、Loopback 0、Loopback 1、Loopback 99999 です。この ID は仮想インターフェイスのタイプごとに固有であるため、Loopback 0 と Null 0 の両方を持つことができ、全体として固有な文字列の名前を形成します。

ループバック インターフェイスとヌルインターフェイスのコントロールプレーンは、アクティブ ルート スイッチ プロセッサ (RSP) 上に存在します。設定およびコントロールプレーンは、スタンバイ RSP 上にミラーリングされ、フェールオーバーが発生した場合には、仮想インターフェイスがそれまでのスタンバイに移り、このスタンバイが新たにアクティブ RSP となります。

- [仮想インターフェイスの設定に関する情報, 55 ページ](#)

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を設定するには、次の概念を理解している必要があります。

仮想ループバック インターフェイスの概要

仮想ループバックインターフェイスは、常にアップ状態にあるシングルエンドポイントを持つ仮想インターフェイスです。仮想ループバック インターフェイスで転送されるパケットは、ただちに同じインターフェイスによって受信されます。ループバック インターフェイスは物理インターフェイスをエミュレートします。

Cisco IOS XR ソフトウェアでは、仮想ループバック インターフェイスが次の機能を実行します。

- ループバック インターフェイスは、ルーティング プロトコルセッションの終端アドレスとして設定することができます。これにより、アウトバウンドインターフェイスがダウンしても、ルーティング プロトコルセッションをアップ状態に維持することができます。
- ルータ IP スタックが適切に動作していることを確認するには、ループバック インターフェイスに対して ping を実行します。

他のルータまたはアクセスサーバが仮想ループバック インターフェイスにアクセスを試みるようなアプリケーションでは、ルーティングプロトコルを設定して、ループバックアドレスに割り当てられるサブネットを分散させる必要があります。

ループバックインターフェイスにルーティングされたパケットは、ルータまたはアクセスサーバに再ルーティングされ、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ものの、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。

仮想インターフェイスの設定の前提条件

適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。このコマンドリファレンスには、各コマンドに必要なタスク IDが含まれます。ユーザグループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA管理者に連絡してください。

仮想ループバック インターフェイスの設定

ここでは、基本的なループバック インターフェイスの設定手順について説明します。

[Restrictions (機能制限)]

ループバックインターフェイスのIPアドレスは、ネットワーク上のすべてのルータ間で固有である必要があります。このIPアドレスは、ルータ上の他のインターフェイスでは使用できません。また、ネットワーク上のいかなるルータのインターフェイスでも使用できません。

手順の概要

1. **configure**
2. **interface loopbackinstance**
3. **ipv4 addressip-address**
4. **end** または **commit**
5. **show interfacestype instance**

手順の詳細

ステップ 1 **configure**

例：

```
RP/0/RP0/CPU0:router# configure
```

グローバル コンフィギュレーション モードを開始します。

ステップ 2 **interface loopbackinstance**

例：

```
RP/0/RP0/CPU0:router#(config)# interface Loopback 3
```

インターフェイス コンフィギュレーション モードを開始して、新しいループバック インターフェイスの名前を指定します。

ステップ 3 **ipv4 addressip-address**

例：

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 100.100.100.69/32
```

ipv4 address コンフィギュレーション コマンドを使用して、仮想ループバック インターフェイスに IP アドレスおよびサブネット マスクを割り当てます。

ステップ 4 **end** または **commit**

例：

```
RP/0/RP0/CPU0:router(config-if)# end
```

または

```
RP/0/RP0/CPU0:router(config-if)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?  
[cancel]:
```

- **yes** と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
- **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻りません。変更はコミットされません。
- **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ステップ 5 `show interfacestype instance`

例：

```
RP/0/RP0/CPU0:router# show interfaces Loopback0
```

(任意) ループバック インターフェイスの設定を表示します。

次に、ループバック インターフェイスを設定する例を示します。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface Loopback0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 100.100.100.69/32
RP/0/RP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
RP/0/RP0/CPU0:router# show interfaces Loopback0

Loopback0 is up, line protocol is up
  Interface state transitions: 1
  Hardware is Loopback interface(s)
  Internet address is 100.100.100.69/32
  MTU 1500 bytes, BW 0 Kbit
    reliability Unknown, txload Unknown, rxload Unknown
  Encapsulation Loopback, loopback not set,
  Last link flapped 01:57:47
  Last input Unknown, output Unknown
  Last clearing of "show interface" counters Unknown
  Input/output data rate is disabled.
```

ヌルインターフェイスの概要

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装置と同様に機能します。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、トラフィックをフィルタリングするための代替的な方法として使用できます。不要なネットワークトラフィックをヌルインターフェイスに送ることによって、アクセスリストを使用する場合に伴うオーバーヘッドを回避できます。

ヌルインターフェイスに指定できるインターフェイス コンフィギュレーション コマンドは **ipv4 unreachable** コマンドのみです。 **ipv4 unreachable** コマンドを使用した場合、ソフトウェアは、認識できないプロトコルが使用されている自分宛の非ブロードキャストパケットを受信すると、インターネット制御メッセージプロトコル (ICMP) プロトコル到達不能メッセージを送信元に送信します。宛先アドレスまでのルートが不明なため最終的な宛先に配信できないデータグラムを受信した場合、ソフトウェアはそのデータグラムの発信者に ICMP ホスト到達不能メッセージで応答します。デフォルトでは、 **ipv4 unreachable** コマンドはイネーブルになっています。ICMP にプロトコル到達不能を送信させない場合は、 **ipv4 icmp unreachable disable** コマンドを使用して設定する必要があります。

ブート プロセス時にデフォルトで Null 0 インターフェイスが作成されます。このインターフェイスは削除できません。このインターフェイスに **ipv4 unreachable** コマンドを設定することは可能ですが、このインターフェイスは送られてきたすべてのパケットを廃棄するだけなので、ほとんどの設定は不要です。

Null 0 インターフェイスを表示するには、 **show interfaces null0** コマンドを使用します。

ヌルインターフェイスの設定

ここでは、基本的なヌルインターフェイスの設定手順について説明します。

手順の概要

1. **configure**
2. **interface null 0**
3. **end** または **commit**
4. **show interfaces null 0**

手順の詳細

ステップ 1 **configure**

例：

```
RP/0/RP0/CPU0:router# configure
グローバル コンフィギュレーション モードを開始します。
```

ステップ 2 **interface null 0**

例：

```
RP/0/RP0/CPU0:router#(config)# interface null 0
null0 インターフェイス コンフィギュレーション モードを開始します。
```

ステップ 3 **end** または **commit**

例：

```
RP/0/RP0/CPU0:router(config-null0)# end
または
```

```
RP/0/RP0/CPU0:router(config-null0)# commit
設定変更を保存します。
```

- **no** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。

```
Uncommitted changes found, commit them before
exiting(yes/no/cancel)?
[cancel]:
```

- **yes** と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
- **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。

- **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ステップ 4 show interfaces null 0

例：

```
RP/0/RP0/CPU0:router# show interfaces null 0
ヌルインターフェイスの設定を確認します。
```

次に、ヌルインターフェイスを設定する例を示します。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface Null 0
RP/0/RP0/CPU0:router(config-null0)# ipv4 unreachable
RP/0/RP0/CPU0:router(config-null0)# end
Uncommitted changes found, commit them? [yes]: yes
RP/0/RP0/CPU0:router# show interfaces Null 0

Null0 is up, line protocol is up
Interface state transitions: 1
Hardware is Null interface
Internet address is Unknown
MTU 1500 bytes, BW 0 Kbit
reliability 255/255, txload Unknown, rxload Unknown
Encapsulation Null, loopback not set,
Last link flapped 4d20h
Last input never, output never
Last clearing of "show interface" counters 05:42:04
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 total input drops
0 drops for unrecognized upper-level protocol
Received 0 broadcast packets, 0 multicast packets
0 packets output, 0 bytes, 0 total output drops
Output 0 broadcast packets, 0 multicast packets
```

仮想 IPv4 インターフェイスの設定

ここでは、IPv4 仮想インターフェイスの設定手順について説明します。

手順の概要

1. **configure**
2. **ipv4 address virtual addressipv4-address subnetmask**
3. **end** または **commit**

手順の詳細

ステップ 1 **configure**

例：

```
RP/0/RP0/CPU0:router# configure
```

グローバル コンフィギュレーション モードを開始します。

ステップ 2 **ipv4 address virtual address ipv4-address subnetmask**

例：

```
RP/0/RP0/CPU0:router(config)# ipv4 virtual address 10.3.32.154/8
```

管理イーサネット インターフェイスの IPv4 仮想アドレスを定義します。

ステップ 3 **end** または **commit**

例：

```
RP/0/RP0/CPU0:router(config-null0)# end
```

または

```
RP/0/RP0/CPU0:router(config-null0)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。

```
Uncommitted changes found, commit them before
exiting(yes/no/cancel)?
[cancel]:
```

- **yes** と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
- **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻りません。変更はコミットされません。
- **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
- 実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

次に、仮想 IPv4 インターフェイスを設定する例を示します。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ipv4 virtual address 10.3.32.154/8
RP/0/RP0/CPU0:router(config-null0)# commit
```




第 6 章

802.1Q VLAN インターフェイスの設定

VLAN とは、実際は異なる LAN セグメント上のデバイスでも、同じセグメントで接続している場合と同様に通信できるように設定された、1 つまたは複数の LAN 上にあるデバイスのグループです。VLAN は、物理接続ではなく論理グループに基づいているため、ユーザ、ホスト管理、帯域幅割り当て、リソースの最適化に非常に柔軟に対応します。

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

802.1Q 仕様は、イーサネットフレームに VLAN メンバーシップ情報を挿入する標準方式を確立します。Cisco NCS 5000 シリーズルータは、10 ギガビットイーサネットインターフェイスおよび 100 ギガビットイーサネットインターフェイス上で VLAN のサブインターフェイスの設定をサポートします。

802.1Q タグ付きフレーム

IEEE 802.1Q タグベースの VLAN は、MAC ヘッダーの特別なタグを使用し、ブリッジでのフレームの VLAN メンバーシップを識別できます。このタグは、VLAN および Quality of Service (QoS) のプライオリティの識別に使用されます。VLAN ID は、フレームを特定の VLAN に関連付けて、スイッチがネットワークでフレームを処理する必要があるという情報を提供します。タグ付きフレームは、タグなしフレームよりも 4 バイト長く、イーサネットフレームの Type および Length フィールドにある 2 バイトの Tag Protocol Identifier (TPID) フィールドと、イーサネットフレームの Source Address フィールドの後ろから始まる 2 バイトの Tag Control Information (TCI) が含まれます。

802.1Q タグ付きフレームの詳細については、『*L2VPN and Ethernet Services Configuration Guide for Cisco NCS 5000 Series Routers*』の「*References for Carrier Ethernet Model*」の項を参照してください。

- [802.1Q VLAN インターフェイスの設定方法, 64 ページ](#)
- [802.1Q VLAN インターフェイスの設定に関する情報, 68 ページ](#)

802.1Q VLAN インターフェイスの設定方法

ここでは、次の手順について説明します。

802.1Q VLAN サブインターフェイスの設定

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

手順の概要

1. **configure**
2. **interface{HundredGigE | TenGigE | Bundle-Ether}interface-path-id.subinterface**
3. **encapsulation dot1q**
4. **ipv4 addressip-address mask**
5. **exit**
6. ステップ 2 ~ 5 を繰り返し、残りの VLAN サブインターフェイスを定義します。
7. **end** または **commit**
8. **show ethernet trunk bundle-etherinstance**
9. **show vlan trunks [brief] [locationinstance] [{GigabitEthernet | TenGigE | Bundle-Ether} | interface-path-id] [summary]**

手順の詳細

ステップ 1 **configure**

例：

```
RP/0/RP0/CPU0:router# configure
グローバル コンフィギュレーション モードを開始します。
```

ステップ 2 **interface{HundredGigE | TenGigE | Bundle-Ether}interface-path-id.subinterface**

例：

```
RP/0/RP0/CPU0:router(config)# interface TenGigE 0/2/0/4.10
サブインターフェイス コンフィギュレーション モードを開始し、インターフェイス タイプ、ロケーション、サブインターフェイス番号を指定します。
```

- **interface-path-id** 引数を、次のいずれかのインスタンスに置き換えます。
- 物理イーサネットインターフェイスインスタンスまたはイーサネットバンドルインスタンス。名前表記は *rack/slot/module/port* で、値の間のスラッシュは表記の一部として必要です。

- イーサネット バンドル インスタンス。範囲は 1 ～ 65535 です。
- *subinterface* 引数をサブインターフェイスの値に置き換えます。範囲は 0 ～ 4095 です。
- 名前表記は *interface-path-id.subinterface* で、表記の一部として引数をピリオドで区切る必要があります。

ステップ 3 encapsulation dot1q

例：

```
RP/0/RP0/CPU0:router(config-subif)# encapsulation dot1q 100, untagged
```

インターフェイスのレイヤ 2 カプセル化を設定します。

ステップ 4 ipv4 address ip-address mask

例：

```
RP/0/RP0/CPU0:router(config-subif)# ipv4 address 178.18.169.23/24
```

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

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

ステップ 5 exit

例：

```
RP/0/RP0/CPU0:router(config-subif)# exit
```

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

- **exit** コマンドは、明示的に指定する必要はありません。

ステップ 6 ステップ 2 ～ 5 を繰り返し、残りの VLAN サブインターフェイスを定義します。

ステップ 7 end または commit

例：

```
RP/0/RP0/CPU0:router(config)# end
```

または

```
RP/0/RP0/CPU0:router(config)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?
[cancel]:
```

-**yes** と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。

-**no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。

-**cancel** と入力すると、ルータは現在のコンフィギュレーションセッションで継続されます。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。

- 実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ステップ 8 show ethernet trunk bundle-etherinstance

例：

```
RP/0/RP0/CPU0:router# show ethernet trunk bundle-ether 5
```

(任意) インターフェイス コンフィギュレーションを表示します。

イーサネット バンドル インスタンスの範囲は 1 ~ 65535 です。

ステップ 9 show vlan trunks [brief] [locationinstance] [{GigabitEthernet | TenGigE | Bundle-Ether | } interface-path-id] [summary]

例：

```
RP/0/RP0/CPU0:router# show vlan trunk summary
```

(任意) 各 VLAN トランク インターフェイスに関するサマリー情報を表示します。

- キーワードの意味は次のとおりです。
- **brief** : 簡潔なサマリーを表示します。
- **summary** : 完全なサマリーを表示します。
- **location** : 指定したポート上の VLAN トランク インターフェイスに関する情報を表示します。
- **interface** : 指定したインターフェイスまたはサブインターフェイスに関する情報を表示します。

802.1Q VLAN サブインターフェイスの削除

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

手順の概要

1. **configure**
2. **no interface {HundredGigE | TenGigE | Bundle-Ether} interface-path-id.subinterface**
3. ステップ 2 を繰り返し、その他の VLAN サブインターフェイスを削除します。
4. **end** または **commit**

手順の詳細

ステップ 1 **configure**

例：

```
RP/0/RP0/CPU0:router# configure
```

グローバル コンフィギュレーション モードを開始します。

ステップ 2 **no interface {HundredGigE | TenGigE | Bundle-Ether} interface-path-id.subinterface**

例：

```
RP/0/RP0/CPU0:router(config)# no interface TenGigE 0/2/0/4.10
```

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

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

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

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

ステップ 4 **end** または **commit**

例：

```
RP/0/RP0/CPU0:router(config)# end
```

または

```
RP/0/RP0/CPU0:router(config)# commit
```

設定変更を保存します。

- **end** コマンドを実行すると、次に示す変更のコミットを求めるプロンプトが表示されます。

```
Uncommitted changes found, commit them before exiting(yes/no/cancel)?
[cancel]:
```

-**yes** と入力すると、実行コンフィギュレーションファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。

-**no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。

-**cancel** と入力すると、ルータは現在のコンフィギュレーションセッションで継続されます。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。

- 実行コンフィギュレーションファイルに設定変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

802.1Q VLAN インターフェイスの設定に関する情報

802.1Q VLAN インターフェイスを設定するには、次の概念を理解している必要があります。

サブインターフェイス

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

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

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

サブインターフェイス MTU

サブインターフェイスの最大伝送単位 (MTU) は、物理インターフェイスから継承されます。これには、802.1Q VLAN タグに許可されている追加の 4 バイトも含まれます。イーサネット MTU およびイーサネット インターフェイスでのフロー制御の詳細については、『*L2VPN and Ethernet Services Configuration Guide for Cisco NCS 5000 Series Routers*』の「*References for Carrier Ethernet Model*」の項を参照してください。

EFP

イーサネット フロー ポイント (EFP) は、抽象的なルータのアーキテクチャを説明する Metro Ethernet Forum (MEF) の用語です。EFP は VLAN カプセル化を使用した レイヤ 2 サブインターフェイスによって実装されます。用語 EFP は VLAN タグ付き L2 サブインターフェイスと同義的に使用されます。EFP の詳細については、『*L2VPN and Ethernet Services Configuration Guide for Cisco NCS 5000 Series Routers*』の「*Carrier Ethernet Model*」の章を参照してください。

VLAN でのレイヤ 2 VPN

レイヤ 2 バーチャルプライベート ネットワーク (L2VPN) 機能を利用すると、サービス プロバイダー (SP) は、地理的に離れたカスタマー サイトにレイヤ 2 サービスを提供できるようになります。

VLAN 接続回線 (AC) を設定するための設定モデルは、基本の VLAN の設定に使用するモデルに類似しています。ユーザはまず VLAN サブインターフェイスを作成し、次にサブインターフェイス コンフィギュレーション モードで VLAN を設定します。AC を作成するには、**interface** コマンド文字列に **l2transport** キーワードを含めて、そのインターフェイスがレイヤ 2 インターフェイスであることを指定する必要があります。

VLAN AC は、これらの L2VPN 操作のモードをサポートします。

- 基本の Dot1Q AC : AC は、特定の VLAN タグで送受信されるすべてのフレームに対応します。
- QinQ AC : AC は、特定の外部 VLAN タグおよび特定の内部 VLAN タグで送受信されるすべてのフレームに対応します。QinQ は、2つのタグのスタックを使用する Dot1Q の拡張です。

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

VLAN 上のレイヤ 2 VPN およびそれらの設定の詳細については、『*L2VPN and Ethernet Services Configuration Guide for Cisco NCS 5000 Series Routers*』の「*Implementing Point-to-Point Layer 2 Services*」の章を参照してください。

