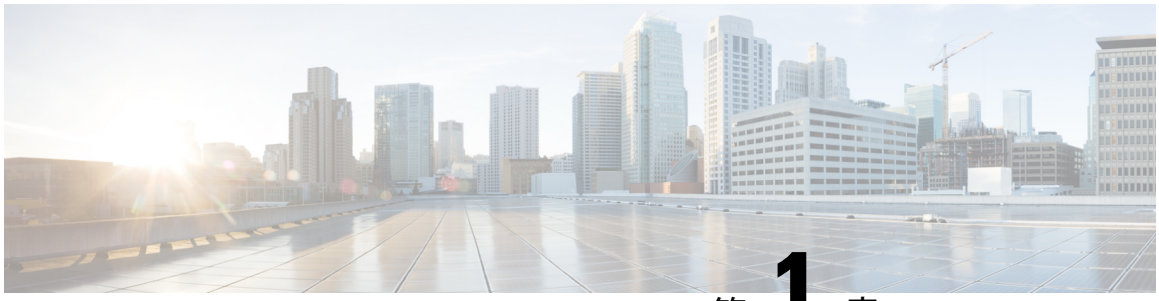
Cisco NCS 560 シリーズ ルータ（IOS XR リリース 6.6.x）インターフェイスおよびハードウェア コンポーネント コンフィギュレーション ガイド

初版：2019 年 5 月 30 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/

目 次



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。

- 1 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、ルータへの装着前にライン カードを設定できます。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ライン カード上に限られるため、ライン 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、ライン 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きにエラーが検出されると、設定は拒否されます。

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。適切なラインカードカードが装着され、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確認され、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） 適切なライン 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確認してください。

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 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、新しいライン カードが即座に設定され、ライン 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、ライン 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

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：

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 192.168.1.2/31
```

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もあります。

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。

例

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)#
```

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.1.2/31
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 コマンドの使用方法

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。

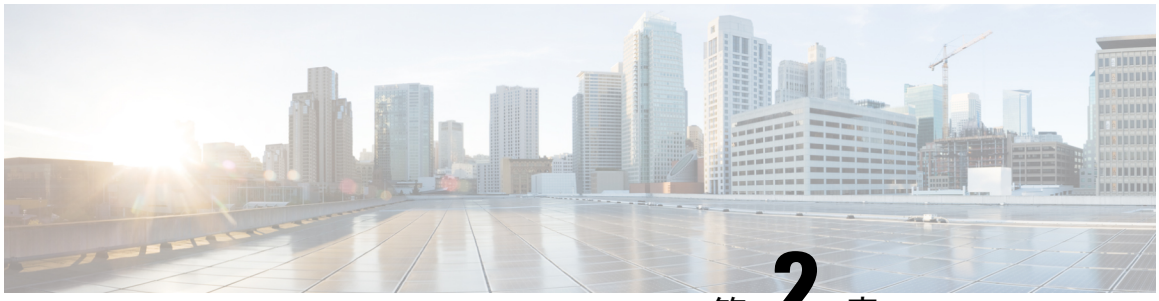
interface preconfigure コマンドによって、ルータはインターフェイス コンフィギュレーション モードに移行します。ユーザは、使用可能なすべてのコマンドを追加できます。プリコンフィギュレーションされたインターフェイス用に登録された検証機能により、設定が検証されます。ユーザが end コマンドを入力するか、それに対応する exit コマンドまたはグローバル コンフィギュレーション モード コマンドを入力すると、プリコンフィギュレーションが完了します。

（注） ライン 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 Hu0/3/0/0 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設定

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） 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存在しますが、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）、HTTP、拡張マークアップ言語（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これらのインターフェイスを設定する必要があります。

（注） ハイ アベイラビリティ設定では、アクティブ RP インターフェイスがシャットダウンされると、スタンバイ RP または仮想 RP が稼働している場合でもゲートウェイへの ping が失敗します。RSP4 は、スタンバイ RP 管理インターフェイスからのパケットの挿入をサポートしていません。

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。

手順

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 にインストールされた RP 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 (or)
ipv4 address 1.76.18.150 255.255.0.0
```

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、255.255.0.0 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、/16 は、マスクの最初の 16 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。

例

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
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 1.76.18.150/16
 ipv6 address 2002::14c:125a/64
 ipv6 enable
!
```

次に、送信元アドレスを使用した RP 上の管理イーサネット インターフェイスの VRF 設定と確認の例を示します。

```
RP/0/RP0/CPU0:router# show run interface MgmtEth 0/RP0/CPU0/0
interface MgmtEth0/RP0/CPU0/0
 vrf httpupload
 ipv4 address 10.8.67.20 255.255.0.0
 ipv6 address 2001:10:8:67::20/48
!

RP/0/RP0/CPU0:router# show run http
Wed Jan 30 14:58:53.458 UTC
http client vrf httpupload
http client source-interface ipv4 MgmtEth0/RP0/CPU0/0

RP/0/RP0/CPU0:router# show run vrf
Wed Jan 30 14:59:00.014 UTC
vrf httpupload
!
```

管理インターフェイスでの IPv6 ステートレス アドレス自動設定

管理インターフェイス上で IPv6 ステートレス自動設定を有効にするには、次のタスクを実行します。

手順

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 にインストールされた RP カードのポート 0 を示しています。

ステップ 3 **ipv6 address autoconfig**

例：

```
RP/0/RP0/CPU0:router(config-if)# ipv6 address autoconfig
```

管理ポート上の IPv6 ステートレス アドレス自動設定を有効にします。

ステップ 4 **show ipv6 interfaces** *interface-path-id*

例：

```
RP/0/RP0/CPU0:router# show ipv6 interfaces gigabitEthernet 0/0/0/0
```

（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。

例

この例では、次のように表示されます。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface MgmtEth 0/RP0/CPU0/0
RP/0/RP0/CPU0:router(config)# ipv6 address autoconfig
RP/0/RP0/CPU0:router# show ipv6 interfaces gigabitEthernet 0/0/0/0

Fri Nov  4 16:48:14.372 IST
GigabitEthernet0/2/0/0 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
  IPv6 is enabled, link-local address is fe80::d1:1eff:fe2b:baf
  Global unicast address(es):
    5::d1:1eff:fe2b:baf [AUTO CONFIGURED], subnet is 5::/64  <<<<<<  auto configured
address
  Joined group address(es): ff02::1:ff2b:baf ff02::2 ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND cache entry limit is 1000000000
  ND advertised retransmit interval is 0 milliseconds
  Hosts use stateless autoconfig for addresses.
  Outgoing access list is not set
  Inbound  common access list is not set, access list is not set
  Table Id is 0xe0800000
  Complete protocol adjacency: 0
  Complete glean adjacency: 0
  Incomplete protocol adjacency: 0
  Incomplete glean adjacency: 0
  Dropped protocol request: 0
  Dropped glean request: 0
```

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

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 **mac-address** *address*

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** アドレス 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

ステップ 1 **show interfaces MgmtEth** *interface-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 MgmtEth** *interface-path-id*

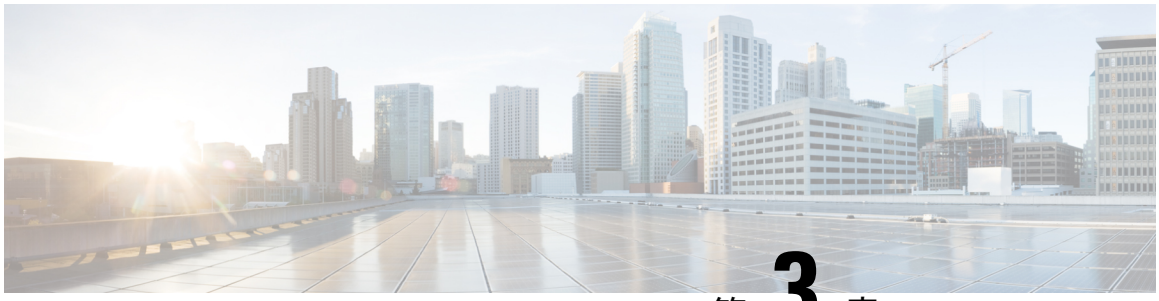
例：

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

実行設定を表示します。

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。



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。

分散型 100 メガビット、1 ギガビット、10 ギガビット、100 ギガビットのイーサネット 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キング 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。

制約事項

ルータはスタティック MAC アドレスの設定をサポートしていません。

物理イーサネット 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

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、ルータがライン 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** |] *interface-path-id*

例：

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

（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。

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
0/0/1/0
```

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。

- 1GigE
- 10GigE
- 100GigE

（注）
- この例は、ライン カード スロット 1 にある 100 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 **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。
- 100 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 7 **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 8 **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 9 **show interfaces** [**HundredGigE | TenGigE**] *interface-path-id*

例：

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

（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。

例

次に、100 ギガビット イーサネットのライン 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 0/7/0/0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224

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

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 interface HundredGigE 0/7/0/0
HundredGigE0/7/0/0 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


RP/0/RP0/CPU0:router# show running-config interface HundredGigE 0/0/1/0

interface HundredGigE0/7/0/0
 mtu 9216
 service-policy input linerate
```

```
 service-policy output elinerate
 ipv4 address 3.24.1.1 255.255.255.0
 ipv6 address 3:24:1::1/64
 flow ipv4 monitor perfv4 sampler fsm ingress
!
```

イーサネットの設定に関する情報

ここでは、次の情報について説明します。

1 ギガビット、10 ギガビット、100 ギガビット イーサネットのデフォルト設定値

次の表に、1 ギガビット、10 ギガビット、100 ギガビット イーサネット ライン カード上でインターフェイスがイネーブルになっているときに存在する、デフォルトのインターフェイス設定パラメータについて説明します。

（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。インターフェイスでのデフォルトは**no shutdown** です。ライン カードを初めて装着したときに、プリコンフィギュレーションが行われていない場合、コンフィギュレーション 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 *1 : 100* ギガビット イーサネット ライン 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
MTU	**mtu**	• 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 アドレス	**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 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。

LLDP

Cisco Discovery Protocol（CDP）は、すべてのシスコ デバイス（ルータ、ブリッジ、アクセス サーバ、およびスイッチ）のレイヤ2（データ リンク層）上で動作するデバイス検出プロトコルです。ネットワーク管理アプリケーションは CDP を使用することにより、ネットワーク接続されている他のシスコ デバイスを自動的に検出し、識別できます。

非シスコデバイスをサポートし、他のデバイス間の相互運用性を確保するために、IEEE 802.1AB LLDP もサポートしています。LLDP は、ネットワーク デバイスがネットワーク上の他のデバイスに自分の情報をアドバタイズするために使用するネイバー探索プロトコルです。このプロトコルはデータ リンク層で動作するため、異なるネットワーク層プロトコルが稼動する 2 つのシステムで互いの情報を学習できます。

LLDP のグローバルな有効化

ルータ上で LLDP を実行するには、グローバルにイネーブルにする必要があります。LLDP をグローバルにイネーブルにすると、LLDP をサポートするすべてのインターフェイスが、送受信の両方の動作に対して自動的にイネーブルになります。

受信または送信動作をディセーブルにするには、インターフェイスでこのデフォルト動作を上書きできます。インターフェイスに対する LLDP の受信動作または送信動作を選択的に無効にする方法の詳細については、「インターフェイスでの *LLDP* の受信動作および送信動作の無効化」の項を参照してください。

次の表に、設定可能なグローバル属性を示します。

属性	デフォルト	範囲	説明
Holdtime	120	0 ～ 65535	パケットで送信される保留時間（秒単位）を指定します。
Reinit	2	2-5	任意のインターフェイスでLLDPの初期化を実行するための遅延（秒単位）
Timer	30	5 ～ 65534	LLDPパケットが送信されるレートを指定します（秒単位）。

LLDP をグローバルにイネーブルにするには、次の手順を実行します。

1. `RP/0/RP0/CPU0:router # configure`
2. `RP/0/RP0/CPU0:router(config) #lldp`
3. `end or commit`

実行コンフィギュレーション

```
RP/0/RP0/CPU0:turin-5#show run lldp
Fri Dec 15 20:36:49.132 UTC
lldp
!

RP/0/RP0/CPU0:turin-5#show lldp neighbors
Fri Dec 15 20:29:53.763 UTC
Capability codes:
        (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
        (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID       Local Intf          Hold-time  Capability     Port ID
SW-NOSTG-I11-PUB.cis Mg0/RP0/CPU0/0      120        N/A            Fa0/28

Total entries displayed: 1

RP/0/RP0/CPU0:turin-5#show lldp neighbors mgmtEth 0/RP0/CPU0/0
Fri Dec 15 20:30:54.736 UTC
Capability codes:
        (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
        (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID       Local Intf          Hold-time  Capability     Port ID
SW-NOSTG-I11-PUB.cis Mg0/RP0/CPU0/0      120        N/A            Fa0/28

Total entries displayed: 1
```

インターフェイスごとの LLDP の有効化

LLDP をグローバルにイネーブルにすると、LLDP をサポートするすべてのインターフェイスが、送受信の両方の動作に対して自動的にイネーブルになります。ただし、インターフェイスごとに LLDP をイネーブルにするには、次の設定手順を実行します。

1. `RP/0/RP0/CPU0:ios(config)# int gigabitEthernet 0/2/0/0`
2. `RP/0/RP0/CPU0:ios(config-if)# no sh`
3. `RP/0/RP0/CPU0:ios(config-if)#commit`
4. `RP/0/RP0/CPU0:ios(config-if)#lldp ?`
5. `RP/0/RP0/CPU0:ios(config-if)#lldp enable`
6. `RP/0/RP0/CPU0:ios(config-if)#commit`

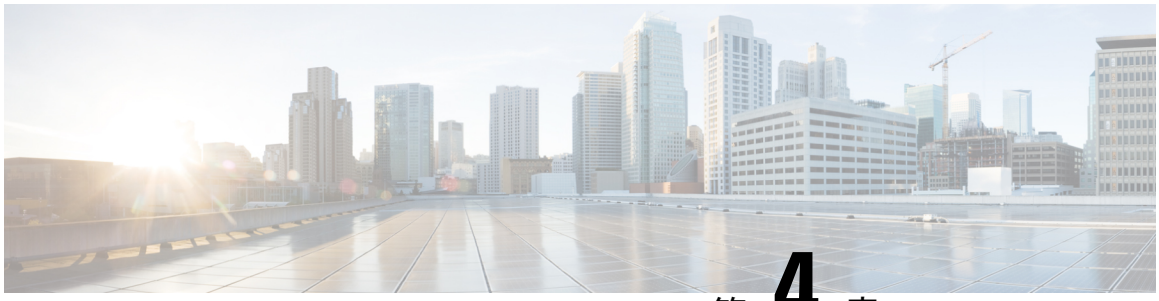
実行コンフィギュレーション

```
RP/0/RP0/CPU0:ios#sh running-config
Wed Jun 27 12:40:21.274 IST
Building configuration...
!! IOS XR Configuration 0.0.0
!! Last configuration change at Wed Jun 27 00:59:29 2018 by UNKNOWN
!
interface GigabitEthernet0/0/0/0
 shutdown
!
interface GigabitEthernet0/0/0/1
 shutdown
!
interface GigabitEthernet0/0/0/2
 shutdown
!
interface GigabitEthernet0/0/0/3
 Shutdown
!
interface GigabitEthernet0/0/0/4
 shutdown
!
interface GigabitEthernet0/0/0/5
 shutdown
!
end
```

確認

```
Verifying the config
==================
RP/0/RP0/CPU0:ios#sh lldp interface <===== LLDP enabled only on GigEth0/0/0/3
Wed Jun 27 12:43:26.252 IST


GigabitEthernet0/0/0/3:
        Tx: enabled
        Rx: enabled
        Tx state: IDLE
        Rx state: WAIT FOR FRAME
RP/0/RP0/CPU0:ios#
```

```
RP/0/RP0/CPU0:ios# show lldp neighbors
Wed Jun 27 12:44:38.977 IST
Capability codes:
        (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
        (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other

Device ID       Local Intf          Hold-time  Capability     Port ID
ios             Gi0/0/0/3           120        R               Gi0/0/0/3       <======
LLDP enabled only on GigEth0/0/0/3 and neighborship seen for the same.

Total entries displayed: 1

RP/0/RP0/CPU0:ios#
```



第 4 章

イーサネット OAM の設定

このモジュールでは、イーサネットの運用管理および保守（OAM）の設定について説明します。

イーサネット OAM 設定の機能履歴

リリース	変更内容

イーサネット OAM の設定に関する情報

イーサネット OAM 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。

イーサネット リンク OAM

メトロエリアネットワーク（MAN）またはワイドエリアネットワーク（WAN）テクノロジーとしてのイーサネットでは、運用管理および保守（OAM）機能の実装によって大きな恩恵が得られます。イーサネット リンク OAM 機能を使用すると、サービス プロバイダーは MAN や WAN での接続の品質をモニタできます。サービス プロバイダーは、特定のイベントをモニタし、ができます。イーサネット リンク OAM は単一の物理リンクで動作し、そのリンクの片側または両側をモニタするように設定できます。

イーサネット リンク OAM は次のように設定できます。

- リンク OAM プロファイルを設定し、このプロファイルを複数のインターフェイスのパラメータの設定に使用できます。
- リンク OAM は、インターフェイス上で直接設定できます。

インターフェイスでリンク OAM プロファイルも使用している場合、プロファイルで設定された特定のパラメータは、インターフェイスで直接別の値を設定することで上書きできます。

EOAM プロファイルにより、複数のインターフェイスで EOAM 機能を設定するプロセスが容易になります。イーサネット OAM プロファイルおよびそのすべての機能は、他のインターフェイスから参照でき、他のインターフェイスでそのイーサネット OAM プロファイルの機能を継承できます。

個々のイーサネット リンク OAM 機能は、1 つのプロファイルに含めることなく、個々のインターフェイスで設定できます。このような場合、個別に設定される機能は、プロファイルの機能よりも常に優先されます。

カスタム EOAM の設定を行う望ましい方法は、イーサネット コンフィギュレーション モードで、EOAMプロファイルを作成し、個別のインターフェイスまたは複数のインターフェイスにアタッチすることです。

次の標準的なイーサネット リンク OAM 機能が、ルータでサポートされています。

イーサネット CFM

イーサネット接続障害管理（CFM）はサービス レベル OAM プロトコルの 1 つで、VLAN ごとにエンドツーエンドのイーサネット サービスをモニタリングおよびトラブルシューティングするためのツールとなります。これには、予防的な接続モニタリング、障害検証、および障害分離の機能が含まれています。CFM は標準的なイーサネット フレームを使用し、イーサネット サービス フレームを転送できる物理メディア上で実行できます。単一の物理リンクに制限される他のほとんどのイーサネット プロトコルとは異なり、CFM フレームは、エンドツーエンドのイーサネット ネットワーク上で送信できます。

CFM は、次の 2 つの規格で定義されています。

- IEEE 802.1ag：CFM プロトコルのコア機能を定義しています。
- ITU-T Y.1731：IEEE 802.1ag の機能との互換性を維持しながら再定義し、一部の追加機能を定義しています。

イーサネット CFM は、ITU-T Y.1731 の次の機能をサポートしています。

- ETH-CC、ETH-RDI、ETH-LB、ETH-LT：これらは IEEE 802.1ag で定義されている、対応する機能と同じです。

（注） Y.1731 で定義されている手順ではなく、IEEE 802.1ag で定義されたリンクトレース レスポンダ手順が使用されます。ただし、相互運用できます。

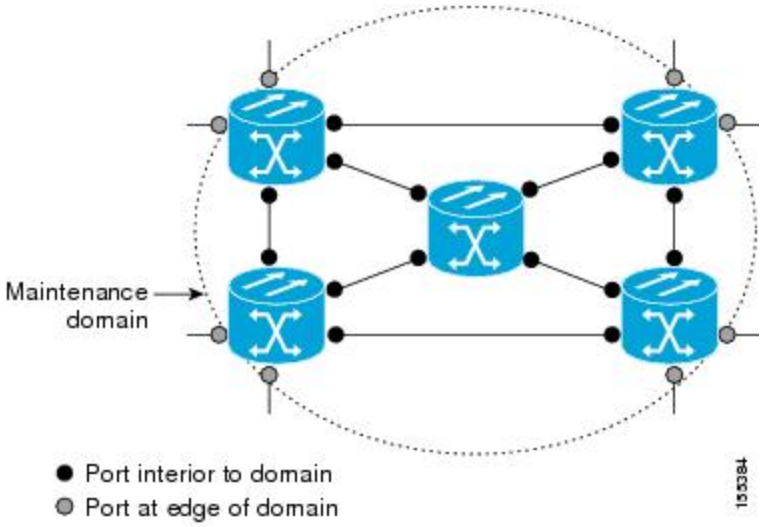
- ETH-AIS：ETH-LCK メッセージの受信もサポートされます。

CFM メンテナンス モデルの仕組みを理解するには、次の概念および機能を理解する必要があります。

メンテナンス ドメイン

メンテナンス ドメインは、ネットワークの管理を目的とした管理空間のことです。ドメインは、単一のエンティティによって所有および運用され、次の図に示すように、インターフェイスのセット（セット内部とセット境界のインターフェイス）によって定義されます。

図 *1 : CFM* メンテナンス ドメイン



メンテナンス ドメインは、そのドメイン内にプロビジョニングされているブリッジ ポートで定義されます。ドメインは、管理者が、0 ～ 7 の範囲でメンテナンス レベルを割り当てます。ドメインのレベルは、複数のドメインの階層関係の定義に役立ちます。

CFM メンテナンス ドメインは、さまざまな組織が、同じネットワークで CFM を個別に使用できます。たとえば、カスタマーにサービスを提供するサービス プロバイダーだとします。そのサービスを提供するために、ネットワークのセグメントで他に 2 人のオペレータを使用します。この環境では、CFM を次のように使用できます。

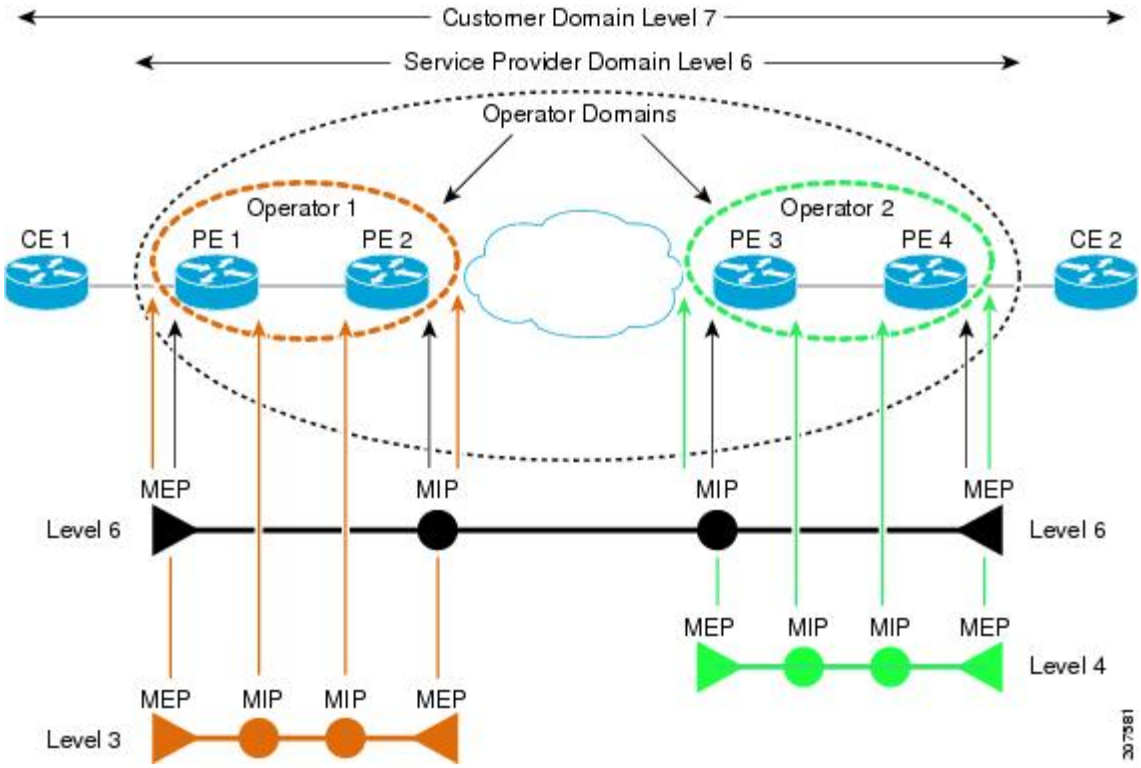
- カスタマーは、ネットワーク全体の接続の確認と管理に CE デバイス間の CFM を使用できます。
- サービス プロバイダーは、提供するサービスの確認と管理に PE デバイス間の CFM を使用できます。
- 各オペレータは、ネットワーク内の接続の確認と管理にオペレータ ネットワーク内の CFM を使用できます。

各組織は別の CFM メンテナンス ドメインを使用します。

次の図に、ネットワーク内の異なるレベルのメンテナンス ドメインの例を示します。

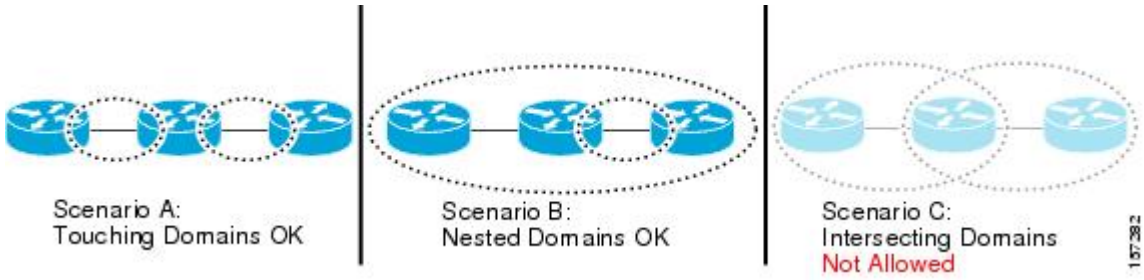
（注） CFM の図の表記規則は、三角形が MEP を表し、MEP が CFM フレームを送信する方向を指します。円は MIP を表します。MEP および MIP の詳細については、71 ページの「メンテナンス ポイント」の項を参照してください。

図 *2:* ネットワーク上のさまざまな *CFM* メンテナンス ドメイン



各ドメインの CFM フレームが相互に干渉しないようにするために、各ドメインは 0 ～ 7 のメンテナンス レベルが割り当てられます。ドメインがネストされている場合、この例のように、包含しているドメインは、包含されているドメインより上のレベルが必要です。この場合、ドメイン レベルは、関係する組織の間でネゴシエートする必要があります。メンテナンス レベルは、ドメインに関連するすべての CFM フレームで伝送されます。

CFM メンテナンス ドメイン同士が隣り合うことやネストは可能ですが、交わることはできません。次の図に、隣り合うドメインとネストされたドメインでサポートされる構造とサポートされていないドメインの交点を示します。



サービス

CFM サービスは、組織がネットワーク内の接続に応じて CFM メンテナンス ドメインを分割することができます。たとえば、ネットワークがいくつかの仮想 LAN（VLAN）に分割されている場合、CFM サービスはそれぞれに作成されます。CFM は、各サービスに個別に実行できます。1 つのサービスに関連する CFM フレームが他のサービスで受信できないように、CFM サービスはネットワーク トポロジに合わせることが重要です。たとえば、サービス プロバイダーは、カスタマーごとにそのカスタマー エンド ポイント間の接続を確認し、管理するために個別の CFM サービスを利用することがあります。

CFM サービスは、メンテナンス ドメインに常に関連付けられ、メンテナンス ドメイン内で動作するため、そのドメインのメンテナンス レベルに関連付けられます。サービス関連のすべての CFM フレームは、対応するドメインのメンテナンス レベルを伝送します。

（注） CFM サービスは、IEEE 802.1ag ではメンテナンス アソシエーションと、ITU-T Y.1731 ではメンテナンス エンティティ グループと呼ばれます。

メンテナンス ポイント

CFM メンテナンス ポイント（MP）は、特定のインターフェイス上の特定の CFM サービスのインスタンスです。CFM はインターフェイスに CFM メンテナンス ポイントが存在する場合だけインターフェイスで動作します。そうでない場合、CFM フレームは、インターフェイスを介して透過的に転送されます。

メンテナンス ポイントは、特定の CFM サービスに常に関連付けられるため、特定のレベルの特定のメンテナンス ドメインに関連付けられます。メンテナンス ポイントは、関連するメンテナンス ドメインと同じレベルの CFM フレームを一般的に処理するだけです。下位メンテナンス レベルのフレームは通常ドロップされますが、上位のメンテナンス レベルのフレームは常に透過的に転送されます。これは、69 ページの「メンテナンス ドメイン」の項で説明するメンテナンス ドメイン階層の適用に役立ち、特定ドメインの CFM フレームがドメインの境界を越えてリークできないようにします。

MP には次の 2 種類があります。

- メンテナンス エンド ポイント（MEP）：ドメインのエッジに作成されます。メンテナンス エンドポイント（MEP）は、ドメイン内の特定のサービスのメンバで、CFM フレームを送信および受信する役割があります。これらは定期的に連続性チェック メッセージを送

信し、ドメイン内の他の MEP から同様のメッセージを受信します。また、管理者の要求に応じて traceroute メッセージやループバック メッセージも送信します。MEP は、CFM メッセージをドメイン内に制限する役割があります。

- メンテナンス中間ポイント（MIP）：ドメインの途中に作成されます。MEP とは異なり、MIP は独自のレベルで CFM フレームを転送できます。

MIP の作成

MEP とは異なり、MIP は各インターフェイスで明示的に設定されていません。MIP は、CFM 802.1ag 規格で指定されたアルゴリズムに従って自動的に作成されます。アルゴリズムは、簡単にいえば、次のように各インターフェイスに対して作用します。

- インターフェイスのブリッジ ドメインまたは相互接続を検出し、そのブリッジ ドメインまたは相互接続に関連するすべてのサービスに、MIP の自動作成を考慮します。
- インターフェイスの最上位レベルの MEP レベルを検出します。上記で考慮されるサービスの中で最上位の MEP レベルより上であり、最もレベルの低いドメインのサービスが選択されます。インターフェイスに MEP がない場合、最下位レベルのドメインのサービスが選択されます。
- 選択したサービス用の MIP の自動作成の設定（**mip auto-create** コマンド）は、MIP を作成する必要があるかどうかを判断するために検査されます。

（注） サービスに対する MIP の自動作成ポリシーの設定は、このサービスに対して MIP が自動的に作成されることを保証するわけではありません。ポリシーは、そのサービスがアルゴリズムで最初に選択されている場合に考慮されるだけです。

MEP と CFM 処理の概要

ドメインの境界は、ブリッジまたはホストではなくインターフェイスです。したがって、MEP は 2 つのカテゴリに分割できます。

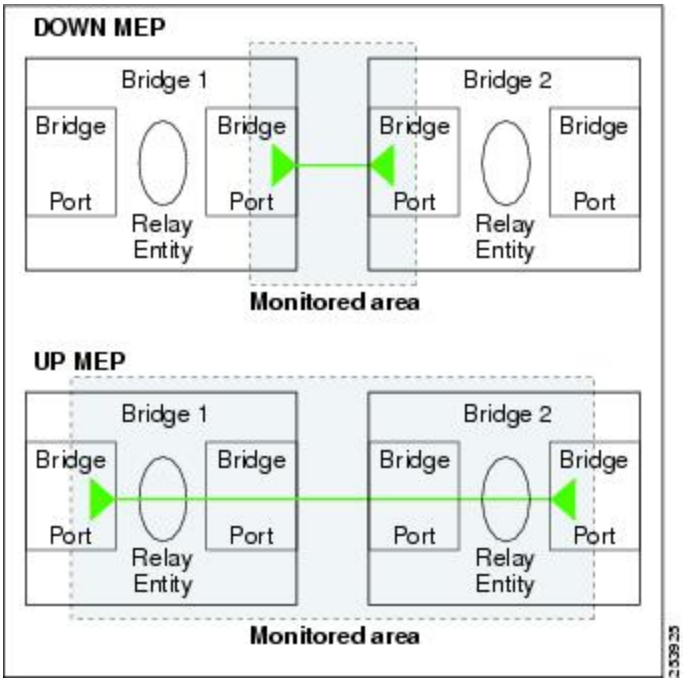
- ダウン MEP：CFM フレームを、それを設定したインターフェイスから送信し、そのインターフェイス上で受信された CFM フレームを処理します。ダウン MEP は AIS メッセージを上位（相互接続の方向）に送信します。
- アップ MEP：MEP が設定されているインターフェイスで受信したものとして、ブリッジ リレー機能にフレームを送信します。これらは、その他のインターフェイスで受信済みであり、MEP が設定されているインターフェイスから送信されるものとしてブリッジ リレー機能によってスイッチングされた CFM フレームを処理します。アップ MEP は AIS メッセージを下位（回線方向）に送信します。ただし、AIS パケットは、MEP と同じインターフェイスで設定された MIP が存在する場合に MIP レベルで送信されるだけです。

（注）

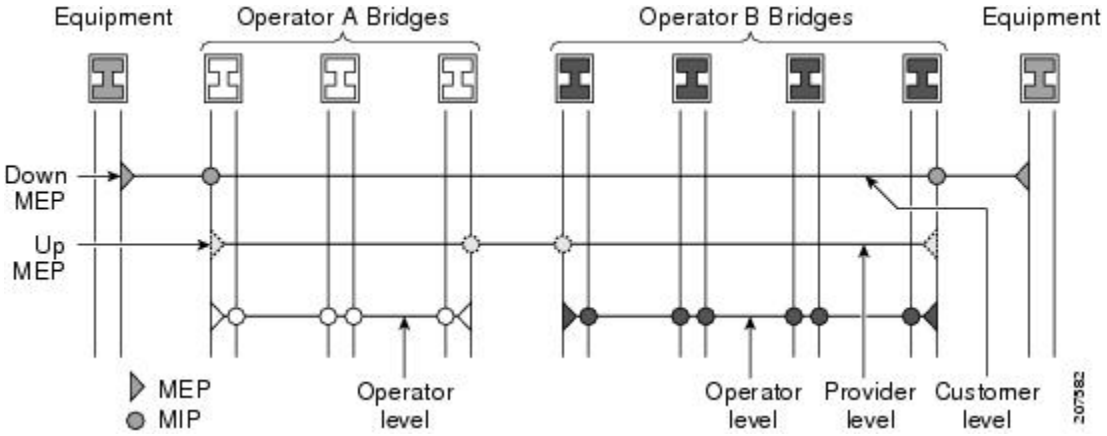
- 用語のダウン *MEP* およびアップ *MEP* は、IEEE 802.1ag と ITU-T Y.1731 規格で定義され、CFM フレームが MEP から送信される方向を指します。これらの用語を MEP の動作ステータスと混同しないでください。
- NCS540 は「ダウン MEP レベル＜アップ MEP レベル」設定のみをサポートします。

次の図に、ダウン MEP とアップ MEP のモニタ対象領域を示します。

図 *3*: ダウン *MEP* とアップ *MEP* のモニタ対象領域



次の図に、さまざまなレベルのメンテナンス ポイントを示します。ドメインはネストできますが交差できないため（図 3 を参照）、低いレベルの MEP は、より高いレベルの MEP または MIP と常に対応します。また、どのインターフェイスにも MIP を 1 つだけ使用できます。これは通常、MEP がないインターフェイスに存在する最下位ドメインに作成されます。



ブリッジ リレー機能からフレームを送受信するため、MIP とアップ MEP はスイッチド（レイヤ 2）インターフェイスにだけ存在できます。ダウン MEP はスイッチド（レイヤ 2）またはルーテッド（レイヤ 3）インターフェイスに作成できます。

MEP が作成されるインターフェイスがスパニングツリー プロトコル（STP）によってブロックされた場合、MEP は正常に動作し続けます。つまり、MEP の指示に従って、MEP レベルで CFM フレームの送受信は続行します。MEP は MEP レベルで CFM フレームの転送を許可しないため、STP ブロックが維持されます。

MIP でもインターフェイスが STP ブロックされた場合、そのレベルで CFM フレームを受信し続け、受信したフレームに応答できます。ただし、MIP は、インターフェイスがブロックされている場合、MIP レベルの CFM フレームを転送できません。

（注） CFM メンテナンス レベルの個別のセットが、VLAN タグがフレームにプッシュされるたびに作成されます。したがって、追加のタグをプッシュするインターフェイスで CFM フレームが受信された場合、フレームがネットワークの一部を「トンネル」するように、トンネル内のどの MP でも、それが同じレベルの場合であっても CFM フレームは処理されません。たとえば、1 つの VLAN タグと一致するカプセル化が指定されたインターフェイスで CFM MP が作成されている場合、そのインターフェイスで受信された 2 つの VLAN タグを持つ CFM フレームは、CFM レベルにかかわらず透過的に転送されます。

CFM プロトコル メッセージ

CFM プロトコルは、目的の異なる複数のメッセージ タイプで構成されます。すべての CFM メッセージは、CFM EtherType を使用し、適用先ドメインの CFM メンテナンス レベルを伝送します。

ここでは、次の CFM メッセージについて説明します。

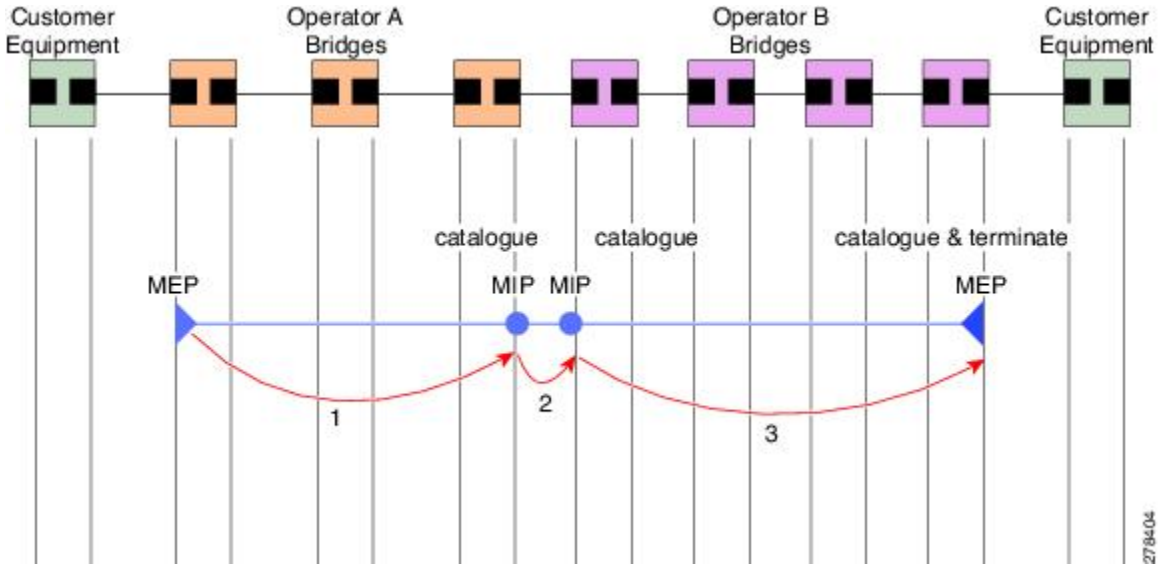
連続性チェック（IEEE 802.1ag および ITU-T Y.1731）

連続性チェック メッセージ（CCM）は、サービス内のすべての MEP 間で定期的に交換される「ハートビート」メッセージです。各 MEP はマルチキャスト CCM を送信し、サービス内の

他のすべての MEP から CCM を受信します。これらはピア *MEP* と呼ばれます。これで、各 MEP がピア MEP を検出し、両者間の接続が確立されていることを確認できます。

MIP は、CCM も受信します。MIP は、その情報を使用して、リンクトレースに応答する場合に使用する MAC 学習データベースを構築します。リンクトレースの詳細については、リンクトレース（IEEE 802.1ag と ITU-T Y.1731）を参照してください。

図 *4*:連続性チェック メッセージのフロー



サービス内の MEP すべてが同じ間隔で CCM を送信する必要があります。IEEE 802.1ag では、使用可能な 7 種類の間隔が定義されています。

- 3.3 ミリ秒
- 10 ミリ秒
- 100 ミリ秒
- 1 秒
- 10 秒
- 1 分
- 10 分

MEP は、ある数の CCM が失われた場合、ピア MEP のうちのいずれかの接続の切断を検出します。これは、CCM 間隔で指定された、一定数の CCM が予期されるのに十分な時間を経過すると発生します。この数値は、損失しきい値と呼ばれ、通常は 3 に設定されます。

CFM は、レイヤ 2 転送機能が有効になっているインターフェイス上でのみサポートされています。

CCM メッセージは、サービス内のさまざまな障害の検出を可能にするさまざまな情報を伝送します。次の情報が含まれます。

- 送信側 MEP のドメインに対して設定された ID。これは、メンテナンス ドメイン ID（MDID）と呼ばれます。
- 送信側 MEP のサービスに対して設定されている ID。これは短い MA 名（SMAN）と呼ばれます。MDID と SMAN を合わせて、メンテナンス アソシエーション ID（MAID）を構成します。MAID は、サービス内の各 MEP で同一に設定する必要があります。
- 次に、時間間隔が 1 分未満のときにセッションでサポートされている MAID のタイプに関する制約事項を示します。MAID はオフロードされた MEP 上で 2 つのタイプの形式をサポートしています。
  - ドメイン名なしの形式
    - MD 名の形式 = 1-NoDomainName
    - MA 名の短い形式 = 3 ～ 2 バイトの整数値
    - MA 名の短い形式 = 2 - 固定長
    - 短い MA 名 = 2 バイトの整数
  - 1731 MAID 形式
    - MD 名の形式 = 1-NoDomainName
    - MA 名の形式（MEGID 形式）= 32
    - MEGID 長 = 13 - 固定長
    - MEGID（ICCCode）= 6 バイト
    - MEGID（UMC）= 7 バイト
    - ITU キャリア コード（ICC）：さまざまな設定可能な ICC コード数 - 15（NPU あたり）
    - 一意の MEG ID コード（UMC） - 4

メンテナンス アソシエーション識別子（MAID）は、メンテナンス ドメイン識別子（MDID）と短い MA 名（SMAN）で構成されます。MDID は**ヌル**値のみをサポートし、SMAN は ITU キャリア コード（ICC）または数値のみをサポートします。その他の値はサポートされていません。

ドメイン ID をヌルに設定する例：**ethernet cfm domain SMB level 3 id null**

SMAN の設定例：**ethernet cfm domain SMB level 3 id null service 901234AB xconnect group 99999 p2p 99999 id number 1**

次の表に、MDID および SMAN でサポートされている値とパラメータの概要を示します。この表では、ハードウェア オフロード機能での MAID 制限についてのみ詳しく説明します。ソフトウェア オフロードまたはオフロードされていない MEP には MAID の制限はありません。

Cisco NCS 5500 シリーズのルータでは、ハードウェア オフロード セッションの場合、ドメイン ID に「id null」を明示的に設定する必要があります。

フォーマット	MDID	SMAN	サポート	備考
	×	2 バイト整数	対応	最大 2000 エントリ
	なし	13 バイト ICC コード（6 バイト）および UMC（7 バイト）	対応	最大 15 個の一意の ICC 最大 4K の UMC 値
48 バイトの文字列ベース	MDID と SMAN で 1 ～ 48 バイト		なし	最も一般的に使用

- MEP（MEP ID）に対して設定された数値 ID。サービス内の各 MEP は異なる MEP ID で設定する必要があります。
- ダイナミック リモート MEP は、間隔が 1 分未満の MEP ではサポートされていません。そのようなすべての MEP には MEP CrossCheck を設定する必要があります。
- シーケンス番号は、間隔が 1 分未満の MEP ではサポートされていません。
- リモート障害表示（RDI）。各 MEP で送信する CCM には、受信している CCM に関連する障害を検出した場合これが含まれます。これは、障害がサービス内のどこかで検出されたことを、サービス内のすべての MEP に通知します。
- CCM が送信される間隔。
- CCM Tx/Rx 統計カウンタは、間隔が 1 分未満の MEP ではサポートされていません。
- 送信者 TLV とシスコ独自のTLV は、間隔が 1 分未満の MEP ではサポートされていません。
- MEPが動作しているインターフェイスのステータス。たとえば、インターフェイスがアップ状態、ダウン状態、STP ブロックされているかどうかなど。

（注） インターフェイスのステータス（アップまたはダウン）をインターフェイスでの MEP の方向（アップ MEP/ダウン MEP）と混同しないでください。

次の障害は、受信した CCM から検出できます。

- 間隔の不一致：受信した CCM の CCM 間隔は、MEP が CCM を送信する間隔に一致しません。

- レベルの不一致：MEP は MEP 独自のレベルよりも下のメンテナンス レベルを伝送する CCM を受信しました。
- ループ：MEP が動作しているインターフェイスの MAC アドレスと同じ送信元 MAC アドレスで CCM が受信されています。
- 設定エラー：受信側 MEP 用に設定された MEP ID と同じ MEP ID で CCM が受信されています。
- 相互接続：ローカルに設定されたと MAID と一致しない MAID で CCM が受信されています。通常は 1 つのサービスからの CCM が他のサービスにリークするなど、ネットワーク内の VLAN の誤設定を示します。
- ピア インターフェイス ダウン：ピアのインターフェイスがダウンしていることを示す CCM が受信されています。
- リモート障害表示：リモート障害表示を伝送する CCM が受信されています。

（注） MEP が送信している CCM にリモート障害表示を含めるのは、この障害によるものではありません。

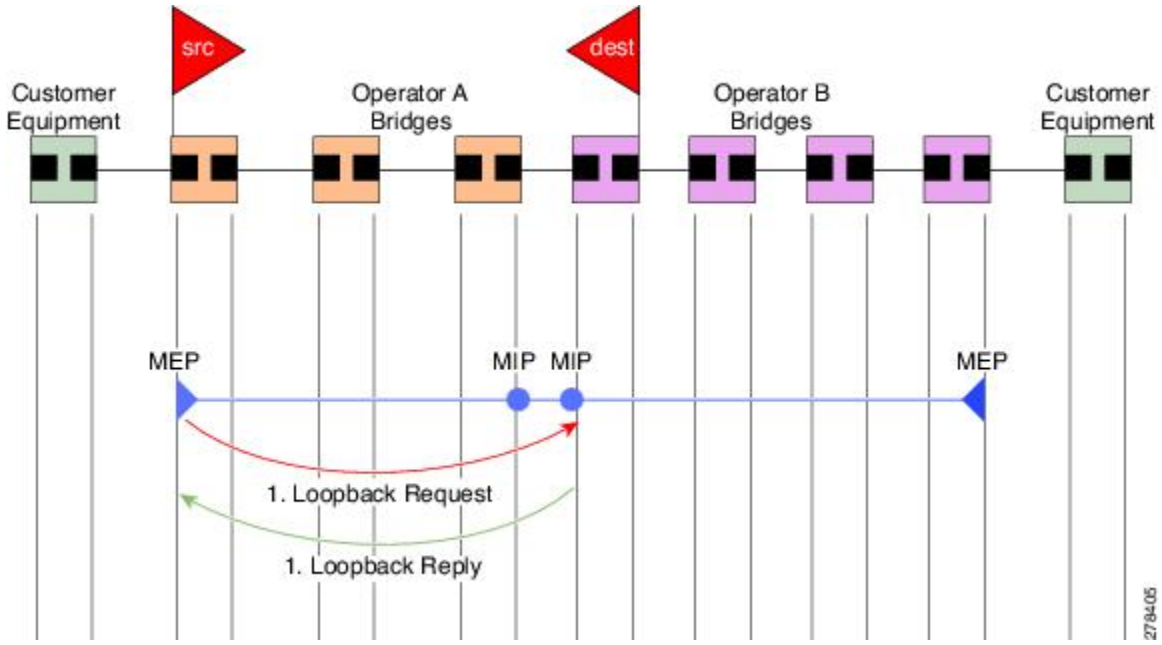
シーケンス外の CCM は、各ピア MEP から受信した CCM のシーケンス番号のモニタリングによっても検出できます。ただし、これは CCM 障害とは見なされません。

ループバック（IEEE 802.1ag と ITU-T Y.1731）

ループバック メッセージ（LBM）およびループバック応答（LBR）は、ローカル MEP と特定のリモート MP の間の接続を確認するために使用されます。管理者の要求に応じて、ローカル MEP はリモート MP にユニキャスト LBM を送信します。各 LBM を受信すると、ターゲット メンテナンス ポイントは、発信元 MEP に LBR を返します。ループバックは、宛先が到達可能かどうかを示します。パスのホップバイホップ検出はできません。ICMP エコー（ping）と概念は似ています。ループバック メッセージがユニキャスト アドレス宛てに送信されるため、メンテナンス レベルを監視している間は通常のデータ トラフィックと同様に転送されます。発信インターフェイスが（ブリッジの転送データベースで）認識されている場合、ループバックが到達する各デバイスで、フレームがそのインターフェイス上で送信されます。発信インターフェイスが認識されていない場合、メッセージはすべてのインターフェイス上でフラッディングされます。

次の図に、MEP と MIP 間の CFM ループバック メッセージ フローの例を示します。

図 *5 : loopback* メッセージ



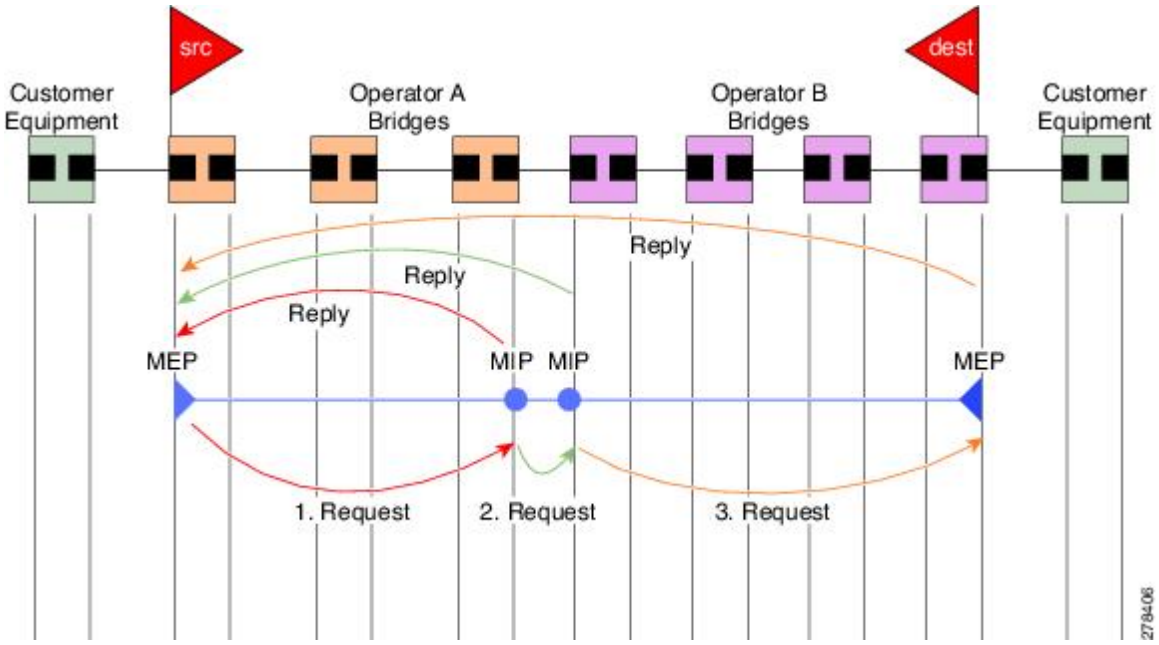
ループバック メッセージは、ユーザが指定したデータでパディングできます。これでデータ破損をネットワークで検出できます。また、順序外のフレームの検出を可能にするシーケンス番号を伝送します。

リンクトレース（IEEE 802.1ag と ITU-T Y.1731）

リンクトレース メッセージ（LTM）およびリンクトレース応答（LTR）は、ユニキャスト宛先 MAC アドレスへのパス（ホップバイホップ）を追跡するために使用されます。オペレータの要求に応じて、ローカル MEP は LTM を送信します。メンテナンス ポイントが存在する各ホップが、発信元 MEP に LTR を返します。これで、管理者がパスに関する接続データを検出できるようになります。メカニズムが異なりますが、IP traceroute と概念は似ています。CFM リンクトレースはパスの各 MP によって転送される単一 LTM を使用しますが、IP traceroute では連続するプローブが送信されます。LTM はマルチキャストであり、フレーム内のデータとしてユニキャスト ターゲット MAC アドレスを伝送します。これらは、メンテナンス ポイントが存在する各ホップで代行受信され、ターゲット MAC アドレスへのユニキャスト パスを検出するために再送信またはドロップされます。

次の図に、MEP と MIP 間の CFM リンクトレース メッセージ フローの例を示します。

図 *6:* リンクトレース メッセージ フロー



リンクトレース メカニズムは、ネットワーク障害後も有用な情報を提供するように設計されています。これは、たとえば連続性の喪失が検出された後などに、障害を見つけるために使用できます。そのためには、各 MP は CCM 学習データベースを維持します。これは、CCM の受信を介したインターフェイスに、受信した各 CCM の送信元 MAC アドレスをマッピングします。これは一般的なブリッジ MAC 学習データベースと似ていますが、CCM だけに基づいていて、分単位というよりは、ほぼ日単位で非常にゆっくりとタイム アウトになる点は除きます。

（注） IEEE 802.1ag で、CCM 学習データベースは MIP CCM データベースと呼ばれます。ただし、MIP と MEP の両方に適用されます。

IEEE 802.1ag では、MP が LTM メッセージを受信すると、次の手順を使用して応答を送信するかどうかを決定します。

1. LTM のターゲット MAC アドレスは、ブリッジ MAC 学習テーブルで検索します。MAC アドレスが認識されており、出力インターフェイスがわかると、LTR が送信されます。
2. MAC アドレスがブリッジ MAC 学習テーブルにない場合は、CCM 学習データベースで検索します。存在する場合、LTR が送信されます。
3. MAC アドレスがない場合、LTR は送信されません（LTM は転送されません）。

ネットワークにターゲット MAC が以前から存在しない場合、リンクトレース動作の結果は得られません。

（注） IEEE 802.1ag と ITU-T Y.1731 はわずかに異なるリンクトレース メカニズムを定義します。特に、CCM 学習データベースの使用と LTM メッセージに応答するための前述のアルゴリズムは IEEE 802.1ag に固有です。IEEE 802.1ag でも LTR に含めることができる追加情報を指定しています。違いに関係なく、2 種類のメカニズムを相互運用できます。

設定可能なロギング

CFM が syslog に対するさまざまな条件のロギングをサポートしています。ロギングは、サービスごとに次の条件が発生した場合に独立してイネーブルにできます。

- 新しいピア MEP が検出されるか、ピア MEP との連続性の喪失が生じる。
- CCM 障害状態への変更が検出される。
- クロスチェックの「missing」または「unexpected」の条件が検出される。
- AIS 状態が検出された（AIS メッセージを受信）またはクリアされた（AIS メッセージを受信しなくなる）。
- EFD を使用してインターフェイスをシャットダウンしたか、アップ状態に戻った。

CFM の柔軟な VLAN タギング

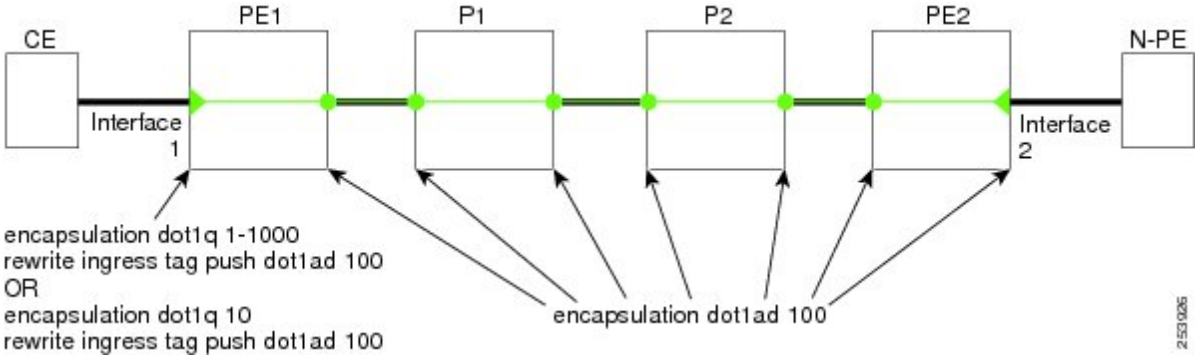
CFM 機能の柔軟な VLAN タギングでは、リモート デバイスで CFM パケットとして適切に処理されるように CFM パケットを正しい VLAN タグ付きで送信できるようにします。パケットがエッジ ルータで受信された場合、ヘッダーのタグの数によって CFM パケットまたはデータ パケットとして処理されます。システムはパケットのタグ数に基づいて CFM パケットとデータ パケットを区別し、パケットのタグ数に基づいて適切なパスにパケットを転送します。

CFM フレームは、設定されたカプセル化とタグの再書き込み動作で定義されたとおりに、インターフェイスで対応するカスタマー データ トラフィックと同じ VLAN タグを付けて通常送信されます。同様に、受信したフレームは、設定されたカプセル化とタグの再書き込み設定で定義されたとおりに正しい数のタグがある場合は CFM フレームとして扱われ、この数値を超えるタグがある場合はデータ フレーム（つまり、透過的に転送される）として扱われます。

ほとんどの場合、同じサービスを通過するデータ トラフィックとまったく同じ方法で CFM フレームが扱われるため、この動作は必要に応じたものです。ただし、複数のカスタマー VLAN が 1 つのマルチポイント プロバイダー サービス上で多重化するシナリオでは（たとえば、N:1 バンドル）、別の動作が望ましい場合があります。

次の図に、CFM を使用し複数の VLAN を持つネットワークの例を示します。

図 *7:* 複数の *VLAN* と *CFM* のサービス プロバイダー ネットワーク



次の図に、S-VLANタグがサービスデリミタとして使用される、プロバイダーのアクセスネットワークを示します。PE1 はカスタマーと対し、PE2 はコア方向のアクセス ネットワークのエッジにあります。N:1 バンドルを使用するので、C-VLAN タグの範囲にインターフェイスのカプセル化が一致します。これは潜在的に全範囲であり、総数:1 バンドルになります。単一 C-VLAN のみを一致させる使用例もありますが、それでも S-VLAN はサービス デリミタとして使用されます。これは、IEEE モデルにより沿ったものですが、プロバイダーは 4094 個のサービスに制限されます。

CFM は、アクセス ネットワークの各エンドに MEP があり、ネットワーク内のボックスに MIP（ネイティブイーサネットの場合）があるネットワークで使用されます。通常は、CFM フレームは 2 個の VLAN タグを使用して、PE1 のアップ MEP によって送信され、カスタマー データ トラフィックを照合します。コア インターフェイスおよび PE2 の MEP では、これらのインターフェイスは S-VLAN タグでのみ一致するため、カスタマー データ トラフィックであるかのように CFM フレームが転送されることを意味します。したがって、PE1 の MEP が送信する CFM フレームは他の MP では認識されません。

柔軟な VLAN タギングはアップ MEP で送受信された CFM フレームのカプセル化を変更します。柔軟な VLAN タギングは、プロバイダー サービスを表す S-VLAN タグだけを付けて PE1 の MEP からフレームが送信されます。このようにすると、コア インターフェイスは CFM フレームとしてフレームを処理し、CFM フレームが MIP と PE2 の MEP によって認識されます。同様に、PE1 の MEP は、PE2 の MEP から受信したことを示す 1 つのタグだけが付いた受信フレームを処理する必要があります。

アップ MEP からの CFM パケットが適切なパスに正しくルーティングされるように、**tags** コマンドを使用してタグをドメイン サービスの特定の番号に設定できます。現在、タグは 1 に設定できるだけです。

イーサネット OAM の設定方法

ここでは、次の設定手順を説明します。

イーサネット CFM の設定

（注） CFM は以下ではサポートされません。

- L3 インターフェイスおよびサブインターフェイス
- バンドル メンバー ポート
- EVPN-FXC
- ブリッジ ドメイン
- VPLS

CFM メンテナンス ドメインの設定

CFM メンテナンス ドメインを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**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	**ethernet cfm** 例： `RP/0/RP0/CPU0:router(config)# ethernet cfm`	イーサネット接続障害管理（CFM）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	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]] 例： `RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	すべてのドメイン設定用コンテナを作成して名前を付け、CFM ドメイン コンフィギュレーション モードを開始します。 レベルを指定する必要があります。 **id** はメンテナンス ドメイン識別子（MDID）で、CFM フレームのメンテナンス アソシエーション識別子（MAID）の最初の部分として使用されます。MDID が指定されていない場合、ドメイン名は MDID としてデフォルトで使用されます。

	コマンドまたはアクション	目的
ステップ 4	**traceroute cache hold-time** *minutes* **size** *entries* 例： `RP/0/RP0/CPU0:router(config-cfm)# traceroute cache hold-time 1 size 3000`	（任意）traceroute キャッシュ エントリの最大制限または traceroute キャッシュ エントリを保持する最大時間限度を設定します。デフォルトは 100 分、100 エントリです。
ステップ 5	**end** または **commit** 例： `RP/0/RP0/CPU0:router(config-cfm-dmn)# commit`	設定変更を保存します。 • **end** コマンドを使用すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 • **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 • **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに設定変更を保存し、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。

CFM メンテナンス ドメインのサービスの設定

メンテナンス ドメインの CFM サービスを最大 2,000 個設定できます。CFM メンテナンス ドメインのサービスを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**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	**ethernet cfm** 例： `RP/0/RP0/CPU0:router(config)# ethernet cfm`	イーサネット CFM 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	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]] 例： `RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	すべてのドメイン設定用コンテナを特定のメンテナンス レベルで作成し、CFM ドメイン コンフィギュレーション モードを開始します。 **id** は、メンテナンス ドメイン識別子（MDID）で、CFM フレームのメンテナンス アソシエーション ID（MAID）の最初の部分として使用されます。MDID が指定されていない場合、ドメイン名は MDID としてデフォルトで使用されます。
ステップ 4	**service** *service-name* {**down-meps** | **xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*] | [[**number** *number*] 例： `RP/0/RP0/CPU0:router(config-cfm-dmn)# service xconnect group X1`	サービスを設定し、ドメインに関連付け、CFM ドメイン サービス コンフィギュレーション モードを開始します。サービスをダウン MEP に対してだけ使用することを指定するか、または MIP およびアップ MEP が作成されるブリッジ ドメインに関連付けることができます。 **id** は短い MA 名を設定します。
ステップ 5	**end** または **commit** 例： `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit`	設定変更を保存します。 • **end** コマンドを使用すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • **yes** と入力すると、実行コンフィギュレーション ファイルに変更が

	コマンドまたはアクション	目的
		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。

CFM サービスの連続性チェックの有効化および設定

CFM サービスの連続性チェックを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**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	**ethernet cfm** 例： `RP/0/RP0/CPU0:router(config)# ethernet cfm`	イーサネット接続障害管理（CFM）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	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]] 例： `RP/0/RP0/CPU0:router(config-cfm)#`	すべてのドメイン設定用コンテナを作成して名前を付け、CFM ドメイン コンフィギュレーション モードを開始します。 レベルを指定する必要があります。

	コマンドまたはアクション	目的
	`domain Domain_One level 1 id string D1`	**id** は、メンテナンス ドメイン識別子（MDID）で、CFM フレームのメンテナンス アソシエーション ID（MAID）の最初の部分として使用されます。MDID が指定されていない場合、ドメイン名は MDID としてデフォルトで使用されます。
ステップ 4	**service** *service-name* {**down-meps** | **xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*] | [[**number** *number*] **例**： `RP/0/RP0/CPU0:router(config-cfm-dmn)# service xconnect group X1`	サービスを設定し、ドメインに関連付け、CFM ドメイン サービス コンフィギュレーション モードを開始します。サービスをダウン MEP に対してだけ使用することを指定するか、または MIP およびアップ MEP が作成されるブリッジ ドメインまたはクロスコネクトに関連付けることができます。 **id** は短い MA 名を設定します。
ステップ 5	**continuity-check interval** *time* [**loss-threshold** *threshold*] **例**： `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check interval 100m loss-threshold 10`	（任意）連続性チェックをイネーブルにし、CCM が送信される間隔を指定するか、または MEP のダウンを宣言するタイミングを示すしきい値の制限を設定します。
ステップ 6	**continuity-check archive hold-time** *minutes* **例**： `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check archive hold-time 100`	（任意）パケットがタイム アウトした後、ピア MEP に関する情報を保存する期間を設定します。
ステップ 7	**continuity-check loss auto-traceroute** **例**： `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check loss auto-traceroute`	（任意）MEP のダウンが宣言されたときの traceroute の自動トリガーを設定します。
ステップ 8	**end** または **commit** **例**： `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit`	設定変更を保存します。 • **end** コマンドを使用すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before`

	コマンドまたはアクション	目的
		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。

CFM サービスの自動 MIP 作成の設定

MIP を作成するためのアルゴリズムの詳細については、「**MIP の作成**」の項を参照してください。

CFM サービスの自動 MIP 作成を設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**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	**ethernet cfm** **例**： `RP/0/RP0/CPU0:router# ethernet cfm`	イーサネット接続障害管理（CFM）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	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]] 例： `RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	すべてのドメイン設定用コンテナを作成して名前を付け、CFM ドメイン コンフィギュレーション モードを開始します。 レベルを指定する必要があります。1 分未満の間隔の MEPS でサポートされているオプションは **id [null]** のみです。 **id** は、メンテナンス ドメイン識別子（MDID）で、CFM フレームのメンテナンス アソシエーション ID（MAID）の最初の部分として使用されます。MDID が指定されていない場合、ドメイン名は MDID としてデフォルトで使用されます。
ステップ 4	**service** *service-name* {**down-meps** | **xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based***icc-string umc-string*] | [**number** *number*] 例： `RP/0/RP0/CPU0:router(config-cfm-dmn)# service xconnect group X1`	サービスを設定し、ドメインに関連付け、CFM ドメイン サービス コンフィギュレーション モードを開始します。サービスをダウン MEP に対してだけ使用することを指定するか、または MIP およびアップ MEP が作成されるブリッジ ドメインに関連付けることができます。 **id** は短い MA 名を設定します。
ステップ 5	**mip auto-create** {**all** | **lower-mep-only**} {**ccm-learning**} 例： `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# mip auto-create all ccm-learning`	（任意）ブリッジ ドメインでの MIP の自動作成をイネーブルにします。**ccm-learning** オプションを使用してこのサービスで作成した MIP の CCM 学習を有効にします。これは、100 ミリ秒以上の比較的長い CCM 間隔を持つサービスでのみ使用してください。デフォルトでは、MIP での CCM 学習は無効になっています。
ステップ 6	**end** または **commit** 例： `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit`	設定変更を保存します。 • **end** コマンドを使用すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`

	コマンドまたはアクション	目的
		• **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。

CFM サービスの MEP でのクロスチェックの設定

CFM サービスの MEP でのクロスチェックを設定し、MEP の予想されるセットを指定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**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	**ethernet cfm** **例：** `RP/0/RP0/CPU0:router# ethernet cfm`	イーサネット接続障害管理（CFM）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	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]] **例：**	すべてのドメイン設定用コンテナを作成して名前を付け、CFM ドメイン 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
	`RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	レベルを指定する必要があります。 **id** は、メンテナンス ドメイン識別子（MDID）で、CFM フレームのメンテナンス アソシエーション ID（MAID）の最初の部分として使用されます。MDID が指定されていない場合、ドメイン名は MDID としてデフォルトで使用されます。
ステップ 4	**service** *service-name* { **bridge group** *bridge-domain-group* **bridge-domain** *bridge-domain-name* | **down-meps** | **xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*] | [**string** *text*] | [**number** *number*] | [**vlan-id** *id-number*] | [**vpn-id** *oui-vpnid*]] 例： `RP/0/RP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1`	サービスを設定し、ドメインに関連付け、CFM ドメイン サービス コンフィギュレーション モードを開始します。サービスをダウン MEP に対してだけ使用することを指定するか、または MIP およびアップ MEP が作成されるブリッジ ドメインまたはクロスコネクトに関連付けることができます。 **id** は短い MA 名を設定します。
ステップ 5	**mep crosscheck** 例： `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# mep crosscheck mep-id 10`	CFM MEP クロスチェック コンフィギュレーション モードを開始します。
ステップ 6	**mep-id** *mep-id-number* [**mac-address** *mac-address*] 例： `RP/0/RP0/CPU0:router(config-cfm-xcheck)# mep-id 10`	MEP でのクロスチェックをイネーブルにします。 （注） • クロスチェックの MEP の予想されるセットに含める各 MEP に対してこのコマンドを繰り返します。
ステップ 7	**end** または **commit** 例： `RP/0/RP0/CPU0:router(config-cfm-xcheck)# commit`	設定変更を保存します。 • **end** コマンドを使用すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`

	コマンドまたはアクション	目的
		• **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。

CFM サービスのその他のオプションの設定

CFM サービスのその他のオプションを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**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	**ethernet cfm** 例： `RP/0/RP0/CPU0:router# ethernet cfm`	イーサネット接続障害管理（CFM）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	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]] 例：	すべてのドメイン設定用コンテナを作成して名前を付け、CFM ドメイン 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
	`RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	レベルを指定する必要があります。 **id** は、メンテナンス ドメイン識別子（MDID）で、CFM フレームのメンテナンス アソシエーション ID（MAID）の最初の部分として使用されます。MDID が指定されていない場合、ドメイン名は MDID としてデフォルトで使用されます。
ステップ 4	**service** *service-name* { **bridge group** *bridge-domain-group* **bridge-domain** *bridge-domain-name* | **down-meps | xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*] | [**string** *text*] | [**number** *number*] | [**vlan-id** *id-number*] | [**vpn-id** *oui-vpnid*]] 例： `RP/0/RP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1`	サービスを設定し、ドメインに関連付け、CFM ドメイン サービス コンフィギュレーション モードを開始します。サービスをダウン MEP に対してだけ使用することを指定するか、または MIP およびアップ MEP が作成されるブリッジ ドメインまたはクロスコネクトに関連付けることができます。 **id** は短い MA 名を設定します。
ステップ 5	**maximum-meps** *number* 例： `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# maximum-meps 1000`	（任意）データベースに記録されるピア MEP の数を制限する、ネットワーク上の MEP の最大数（2 ～ 8190）を設定します。
ステップ 6	**log** {**ais**|**continuity-check errors**|**continuity-check mep changes**|**crosscheck errors**|**efd**} 例： `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log continuity-check errors`	（任意）特定の種類のイベントのロギングをイネーブルにします。
ステップ 7	**end** または **commit** 例： `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit`	設定変更を保存します。 • **end** コマンドを使用すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • **yes** と入力すると、実行コンフィギュレーション ファイルに変更が

	コマンドまたはアクション	目的
		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。

CFM MEP の設定

手順

	コマンドまたはアクション	目的
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 TenGigE 0/0/0/1`	MEP を作成するイーサネット インターフェイスのタイプ。**HundredGigE** または **TenGigE** と物理インターフェイスまたは仮想インターフェイスを入力します。 （注） • ルータに現在設定されているすべてのインターフェイスのリストを表示するには、**show interfaces** コマンドを使用します。

	コマンドまたはアクション	目的
ステップ 3	**interface** {**HundredGigE	TenGigE	Bundle-Ether**} *interface-path-id.subinterface* 例： `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/1`	MEP を作成するイーサネット インターフェイスのタイプ。**HundredGigE, TenGigE,** または **Bundle-Ether** と物理インターフェイスまたは仮想インターフェイスを入力し、その後にサブインターフェイス パス ID を入力します。 名前表記は、*interface-path-id.subinterface* です。表記の一部としてサブインターフェイス値の前にピリオドが必要です。
ステップ 4	**vrf vrf-name** 例： `RP/0/RP0/CPU0:router(config-if)# vrf vrf_A`	VRF インスタンスを設定し、VRF 設定モードを開始します。
ステップ 5	**interface** {**HundredGigE	TenGigE**} *interface-path-id* 例： `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/1`	MEP を作成するイーサネット インターフェイスのタイプ。**HundredGigE** または **TenGigE** と物理ンターフェイスまたは仮想インターフェイスを入力します。 （注） • ルータに現在設定されているすべてのインターフェイスのリストを表示するには、**show interfaces** コマンドを使用します。
ステップ 6	**ethernet cfm** 例： `RP/0/RP0/CPU0:router(config-if)# ethernet cfm`	インターフェイス イーサネット CFM 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	**mep domain** *domain-name* **service** *service-name* **mep-id** *id-number* 例： `RP/0/RP0/CPU0:router(config-if-cfm)# mep domain Dm1 service Sv1 mep-id 1`	インターフェイスのメンテナンス エンドポイント（MEP）を作成し、インターフェイス CFM MEP 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	**cos** *cos* 例： `RP/0/RP0/CPU0:router(config-if-cfm-mep)# cos 7`	（任意）インターフェイスで MEP が生成するすべての CFM パケットのサービス クラス（CoS）（0 ～ 7）を設定します。設定しない場合、CoS はイーサネット インターフェイスから継承されます。

	コマンドまたはアクション	目的
		（注） イーサネット インターフェイスの場合、CoS は VLAN タグ内のフィールドとして伝送されます。したがって、CoS は、パケットが VLAN タグで送信されるインターフェイスにのみ適用されます。VLAN カプセル化を設定しないインターフェイス上で MEP に **cos (CFM)** コマンドを実行しても無視されます。
ステップ 9	**end** または **commit** 例： `RP/0/RP0/CPU0:router(config-if-cfm-mep)# commit`	設定変更を保存します。 • **end** コマンドを使用すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 • **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 • **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに設定変更を保存し、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。

Y.1731 AIS の設定

ここでは、次のステップの手順について説明します。

CFM ドメイン サービスの AIS の設定

CFM ドメイン サービスのアラーム表示信号（AIS）の送信を設定し、AIS のロギングを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**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	**ethernet cfm** 例： `RP/0/RP0/CPU0:router(config)# ethernet cfm`	イーサネット CFM 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	**domain** *name* **level** *level* 例： `RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1`	ドメインおよびドメイン レベルを指定します。
ステップ 4	**service** *name* **bridge group** *name* **bridge-domain** *name* 例： `RP/0/RP0/CPU0:router(config-cfm-dmn)# service S1 bridge group BG1 bridge-domain BD2`	サービス、ブリッジ グループとブリッジ ドメインを指定します。
ステップ 5	**service** *name* **xconnect group** *xconnect-group-name* **p2p** *xconnect-name* 例： `RP/0/RP0/CPU0:router(config-cfm-dmn)# service S1 bridge group BG1 bridge-domain BD2`	サービスとクロスコネクト グループおよび名前を指定します。

	コマンドまたはアクション	目的
ステップ 6	**ais transmission** [**interval** {**1s**|**1m**}][**cos** *cos*] 例： `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# ais transmission interval 1m cos 7`	接続障害管理（CFM）ドメイン サービスのアラーム表示信号（AIS）の送信を設定します。
ステップ 7	**log ais** 例： `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log ais`	接続障害管理（CFM）ドメイン サービスの AIS ロギングを、AIS または LCK パケットを受信したときに示すように設定します。
ステップ 8	**end** または **commit** 例： `RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# 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。

CFM インターフェイス上での AIS の設定

CFM インターフェイスで AIS を設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**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 gigabitethernet** *interface-path-id* 例： `RP/0/RP0/CPU0:router# interface TenGigE 0/0/0/2`	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	**ethernet cfm** 例： `RP/0/RP0/CPU0:router(config)# ethernet cfm`	イーサネット CFM 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	**ais transmission up interval 1m cos** *cos* 例： `RP/0/RP0/CPU0:router(config-if-cfm)# ais transmission up interval 1m cos 7`	接続障害管理（CFM）インターフェイスのアラーム表示信号（AIS）の送信を設定します。
ステップ 5	**end** または **commit** 例： `RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# 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

	コマンドまたはアクション	目的
		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。

CFM の柔軟な VLAN タギングの設定

CFM パケット内のタグの数を、CFM ドメイン サービスに設定するには、次の手順を使用します。

手順

	コマンドまたはアクション	目的
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	**ethernet cfm** 例： `RP/0/RP0/CPU0:router(config)# ethernet cfm`	イーサネット CFM 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	**domain** *name* **level** *level* 例： `RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1`	ドメインおよびドメイン レベルを指定します。
ステップ 4	**service** *name* **bridge group** *name* **bridge-domain** *name* 例： `RP/0/RP0/CPU0:router(config-cfm-dmn)#`	サービス、ブリッジ グループとブリッジ ドメインを指定します。

	コマンドまたはアクション	目的
	`service S2 bridge group BG1 bridge-domain BD2`	
ステップ 5	**tags** *number* 例： `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# tags 1`	CFM パケット内のタグの数を指定します。現在、有効値は 1 だけです。
ステップ 6	**end** または **commit** 例： `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# 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。

CFM 設定の確認

CFM 設定を確認するには、次のコマンドを 1 つ以上使用します。

show ethernet cfm configuration-errors [**domain** *domain-name*] [**interface** *interface-path-id*]	設定されたCFM動作がアクティブになるのを妨げているエラー、および発生した警告に関する情報を表示します。
show ethernet cfm local maintenance-points domain *name* [**service** *name]* | **interface** *type interface-path-id*] [**mep** | **mip**]	ローカル メンテナンス ポイントのリストを表示します。

（注） CMF を設定した後、エラーメッセージ「*cfmd[317]: %L2-CFM-5-CCM_ERROR_CCMS_MISSED : Some received CCMs have not been counted by the CCM error counters*」が表示される場合があります。このエラー メッセージは、機能上の影響はなく、対処する必要はありません。

トラブルシューティングのヒント

CFM ネットワーク内の問題をトラブルシートするには、次のステップを実行します。

手順

ステップ 1 問題のある MEP への接続を確認するには、次の例に示すように **ping ethernet cfm** コマンドを使用します。

```
RP/0/RP0/CPU0:router# ping ethernet cfm domain D1 service S1 mep-id 16
source
interface TenGigE 0/0/0/1

Type escape sequence to abort.
Sending 5 CFM Loopbacks, timeout is 2 seconds -
Domain foo (level 2), Service foo
Source: MEP ID 1, interface TenGigE0/0/0/1
Target: 0001.0002.0003 (MEP ID 16):
  Running (5s) ...
Success rate is 60.0 percent (3/5), round-trip min/avg/max = 1251/1349/1402 ms
Out-of-sequence: 0.0 percent (0/3)
Bad data: 0.0 percent (0/3)
Received packet rate: 1.4 pps
```

ステップ 2 **ping ethernet cfm** コマンドの結果にピア MEP への接続の問題が示されている場合は、**traceroute ethernet cfm** コマンドを使用し、次の例に示すように問題の場所をさらに分離できるようにします。

```
RP/0/RP0/CPU0:router# traceroute ethernet cfm domain D1 service S1 mep-id
16 source interface TenGigE 0/0/0/2

Traceroutes in domain D1 (level 4), service S1
Source: MEP-ID 1, interface TenGigE0/0/0/2
================================================================================
Traceroute at 2009-05-18 12:09:10 to 0001.0203.0402,
TTL 64, Trans ID 2:
```

```
Hop Hostname/Last            Ingress MAC/name       Egress MAC/Name        Relay
--- ----------------------- ---------------------- ---------------------- -----
  1 ios                     0001.0203.0400 [Down]                         FDB
      0000-0001.0203.0400   TenGigE0/0/0/2
  2 abc                                            0001.0203.0401 [Ok]    FDB
      ios                                          Not present
  3 bcd                     0001.0203.0402 [Ok]                           Hit
      abc                   TenGigE0/0
Replies dropped: 0
```

ターゲットが MEP の場合は、最後のホップの Relay フィールドに「Hit」と表示されていることを確認してください。これは、ピア MEP への接続を確認するためです。

Relay フィールドに「MPDB」と表示されているホップがある場合は、ターゲット MAC アドレスがそのホップのブリッジ MAC 学習テーブルで見つからなかったため、結果として、CCM 学習に依存しています。この結果は正常な状況で生じているが、問題を示している可能性があります。**traceroute ethernet cfm** コマンドを使用する前に **ping ethernet cfm** を使用した場合は、MAC アドレスが学習されている必要があります。その場合に「MPDB」が出現したときは、ネットワークのそのポイントでの問題を示しています。

Unidirectional Link Detection Protocol（単方向リンク検出プロトコル）

単方向リンク検出（UDLD）は、イーサネットリンク（ポイントツーポイントと共有メディアの両方のリンクが含まれます）をモニタリングするためのシングルホップ物理リンクプロトコルです。これは、物理リンク層で検出されないリンクの問題を検出するための、シスコ独自のプロトコルです。このプロトコルの対象は、非バンドル ファイバ リンクを使用するときの配線エラーです。このようなリンクでは、1 つのポートの送信接続と受信接続の間に不一致が存在することがあります。

制限事項

- UDLD は、他の低速プロトコルと同様に、l2vpn を通じてトンネルされません。
- UDLD は SPAN 送信元ポートまたは宛先ポートでイネーブルにしないでください。

UDLD の動作

UDLD は、隣接デバイス間でプロトコル パケットを交換することによって動作しています。UDLD を動作させるには、リンク上の両方のデバイスが UDLD をサポートしており、それぞれのポートで有効にする必要があります。

UDLD が設定されたポートで、最初の PROBE メッセージが送信されます。UDLD が PROBE メッセージを受信した後は、定期的に ECHO（hello）メッセージが送信されます。どちらのメッセージにも送信元とそのポートが明示されており、そのポートでのプロトコル動作パラ

メータに関する情報も格納されています。また、ローカル デバイスがそのポートでネイバー デバイスからデバイスとポートの ID を受け取った場合は、その ID も格納されています。同様に各デバイスは、自身が接続されている場所、およびネイバーが接続されている場所を認識します。

この情報を使用すると、障害や誤配線状態を検出できます。このプロトコルの動作にはエージング メカニズムが組み込まれており、ネイバーからの情報が定期的に更新されない場合は、最終的にタイムアウトとなります。このメカニズムは、障害検出にも使用できます。

FLUSH メッセージは、あるポートで UDLD がディセーブルになっていることを示すのに使用されます。この結果、ローカル デバイスはピアのネイバー キャッシュから削除され、これによってエージング アウトが回避されます。

問題が検出された場合は、影響を受けるインターフェイスが UDLD によってディセーブルになり、ユーザへの通知も送信されます。これは、トラフィック損失以外のネットワークの問題を回避するためです。たとえばループのような、STP によって検出されず、防止もできない問題です。

障害検出のタイプ

UDLD では、次のタイプの障害を検出できます。

- 送信障害：ローカル ポートからピア デバイスへのパケット送信に失敗したが、そのピアからのパケット受信は続いている場合です。このような障害の原因は、物理リンクの障害（レイヤ 1 での単方向リンク障害の通知がメディアでサポートされていない）や、ローカルまたはピア デバイスでのパケット パス障害です。
- 誤配線障害：ローカル デバイスの、あるポートの受信側と送信側がそれぞれ異なるピア ポートに接続されている場合です（接続先が同じデバイスか、異なるデバイスかを問わない）。これは、光ファイバ ポートの接続に非バンドル ファイバを使用する場合に発生することがあります。
- ループバック障害：あるポートの受信側と送信側が相互に接続され、ループバック状態が作られている場合です。これは、意図的な動作モードのこともありますが（ある種のテスト目的）、これに該当する場合は UDLD を使用しないでください。
- 受信障害：このプロトコルにはハートビートも含まれており、ネゴシエートされた間隔でピア デバイスに送信されます。したがって、ハートビートの欠落を調べると、リンクの受信側の障害（インターフェイスの状態変更を引き起こさないもの）を検出できます。この原因としては、単方向リンクで発生した障害が受信側だけに影響していることや、リンクで発生した双方向の障害が考えられます。この検出を可能にするには、ピア デバイスによって確実に、定期的にパケットが送信される必要があります。このような理由から、UDLD プロトコルには 2 つの設定可能な動作モードがあり、ハートビート タイムアウト時の動作はこのモードによって決まります。これらのモードについては、UDLD の動作モード （63 ページ）の項を参照してください。

UDLD の動作モード

UDLD は次のモードで動作可能です。

- **通常モード**：このモードでは、受信側の障害が検出された場合はユーザに通知が送信され、それ以上のアクションは行われません。
- **アグレッシブ モード**：このモードでは、受信エラーが検出された場合はユーザに通知が送信され、影響を受けるポートがディセーブルになります。

（注） 通常モードとアグレッシブ モードでの動作の違いは、ネイバー タイムアウトの場合にのみ示されます。他のすべてのケースでは、通常モードまたはアグレッシブ モードに関係なく、単方向リンクが検出されると、システム エラーによってリンクが無効になります。

UDLD のエージング メカニズム

ここで示すのは、受信障害状態のときのシナリオです。UDLD 情報のエージング アウトが発生するのは、UDLD が動作しているポートにおいて、保留時間が経過してもネイバー ポートから UDLD パケットが受信されないときです。ポートの保留時間はリモート ポートによって決まり、リモート側のメッセージ間隔によって異なります。メッセージ間隔が短ければ短いほど、保留時間が短くなって検出が速くなります。保留時間は、Cisco IOS XR ソフトウェアのメッセージ間隔の 3 倍です。

UDLD 情報のエージング アウトは、ポートでのエラー率が高いときに起きることがあり、その原因としては物理的な問題やデュプレックスのミスマッチがあります。この場合のパケットドロップは、リンクが単方向であることを意味するものではないので、通常モードの UDLD では、そのようなリンクがディセーブルになることはありません。

検出時間を適切に設定するには、正しいメッセージ間隔を選択することが重要です。転送ループが作成される前に単方向リンクを検出できる程度に、メッセージ間隔を短くしてください。デフォルトのメッセージ間隔は 60 秒です。検出時間は、メッセージ間隔のおよそ 3 倍です。したがって、デフォルトの UDLD タイマーを使用するときは、UDLD によるリンクのタイムアウトが STP のエージング タイムよりも前に起きることはありません。

ステート マシン

UDLD では、2 種類の有限状態マシン（FSM）が使用されます。これらは一般的に、「ステート マシン」と呼ばれます。メイン FSM は、プロトコルの動作のすべての段階を扱い、検出 FSM は、ポートのステータスを判断する段階だけを扱います。

メイン FSM

メイン FSM の状態は、次のいずれかとなります。

- **Init**：プロトコルが初期化中です。
- **UDLD inactive**：ポートがダウンしているか、UDLD がディセーブルです。
- **Linkup**：ポートが稼働中であり、UDLD はネイバーの検出中です。
- **Detection**：新しいネイバーからの hello メッセージを受信済みであり、ポートのステータスを特定するための検出 FSM が実行中です。
- **Advertisement**：検出 FSM の実行が完了しており、ポートが正常に動作していると判断されました。定期的に hello が送信され、ネイバーからの hello がモニタリングされます。
- **Port shutdown**：検出 FSM が障害を検出したか、すべてのネイバーがタイムアウトし（アグレッシブ モードのとき）、その結果としてポートがディセーブルにされました。

検出 FSM

検出 FSM の状態は、次のいずれかとなります。

- **Unknown**：検出がまだ実行されていないか、UDLD がディセーブルになっています。
- **Unidirectional detected**：ネイバーがローカルデバイスを認識していないことが理由の単方向リンク状態が検出されました。ポートはディセーブルになります。
- **Tx/Rx loop**：ポート自身の ID が格納された TLV の受信によってループバック状態が検出されました。ポートはディセーブルになります。
- **Neighbor mismatch**：誤配線が検出されました。これは、ローカル デバイスが認識していない他のデバイスをネイバーが認識している状態です。ポートはディセーブルになります。
- **Bidirectional detected**：UDLD hello メッセージの交換が両方向で正常に終了しました。ポートは正しく動作しています。

Y.1731 パフォーマンス モニタリング

Y.1731 パフォーマンス モニタリング（PM）では、イーサネットのフレーム遅延、フレーム遅延変動、フレーム損失、フレーム スループット測定など、標準的なイーサネット PM 機能が提供されます。これらの測定は ITU-T Y-1731 標準で規定され、メトロ イーサネット フォーラム（MEF）標準グループによって認定されています。

NCS 540 は次をサポートしています。

- 双方向遅延測定（DM）

- 合成損失測定（SLM）

双方向遅延測定

イーサネット フレームの遅延測定を使用して、フレーム遅延とクレーム遅延変動を測定します。システムは、遅延測定メッセージ（DMM）メソッドを使用してイーサネットのフレーム遅延を測定します。

双方向遅延測定の設定に関する制約事項

双方向遅延測定を設定する際は、ここに記載するガイドラインと制約事項に従ってください。

- 一方向 DMM では、Y.1731 PM はサポートされていません。

双方向遅延測定の設定

双方向遅延測定を設定するには、次のステップを実行します。

Router> enable

RP/0/RP0/CPU0:router # **configure terminal**

RP/0/RP0/CPU0:router **(config)** # **ip sla 1101**

RP/0/RP0/CPU0:router **(config-ip-sla)** # **ethernet y1731 delay DMM domain customer vlan 100 mpaid 3101 cos 1 source mpid 4101**

RP/0/RP0/CPU0:router **(config-sla-y1731-delay)** # **aggregate interval 30**

RP/0/RP0/CPU0:router **(config-sla-y1731-delay)** # **exit**

/ * Schedule two-way delay measurement * /

RP/0/RP0/CPU0:router **(config)** # **ip sla schedule 1101 life forever start-time now**

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

CFM 遅延測定のオンデマンド イーサネット SLA 動作の設定

CFM 遅延測定のオンデマンド イーサネット SLA 動作を設定するには、特権 EXEC コンフィギュレーション モードで次のコマンドを使用します。

RP/0/RP0/CPU0:router #

ethernet sla on-demand operation type cfm-synthetic-loss-measurement probe domain D1 source interface TenGigE 0/0/0/0 target mac-address 2.3.4

ethernet sla on-demand operation type cfm-synthetic-loss-measurement probe domain D1 source interface TenGigE 0/0/0/0 target mac-address 2.3.4

実行コンフィギュレーション

```
P/0/RP0/CPU0:ios#show ethernet cfm peer meps
Mon Sep 11 12:09:44.534 UTC
Flags:
> - Ok                          I - Wrong interval
R - Remote Defect received      V - Wrong level
L - Loop (our MAC received)     T - Timed out
C - Config (our ID received)    M - Missing (cross-check)
X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)
* - Multiple errors received    S - Standby

Domain UP6 (level 6), Service s6
Up MEP on FortyGigE0/0/1/2.1 MEP-ID 1
================================================================================
St     ID MAC Address    Port    Up/Downtime   CcmRcvd SeqErr   RDI Error
-- ----- -------------- ------- ------------ --------- ------ ----- -----
 >  4001 70e4.227c.2865 Up      00:01:27             0      0     0     0

Domain DOWN0 (level 0), Service s10
Down MEP on TenGigE0/0/0/10.1 MEP-ID 2001
================================================================================
St     ID MAC Address    Port    Up/Downtime   CcmRcvd SeqErr   RDI Error
-- ----- -------------- ------- ------------ --------- ------ ----- -----
 >  6001 70e4.227c.287a Up      00:02:11             0      0     0     0
RP/0/RP0/CPU0:ios#
RP/0/RP0/CPU0:ios#
RP/0/RP0/CPU0:ios#show running-config
Mon Sep 11 12:10:18.467 UTC
Building configuration...
!! IOS XR Configuration version = 6.4.1.14I
!! Last configuration change at Mon Sep 11 12:08:16 2017 by root
!
logging console disable
telnet vrf default ipv4 server max-servers 10
username root
 group root-lr
 group cisco-support
 secret 5 $1$QJT3$94M5/wK5J0v/lpAu/wz31/
!
line console
 exec-timeout 0 0
!
ethernet cfm
 domain UP6 level 6 id null
  service s6 xconnect group g1 p2p p1 id number 6
   mip auto-create all ccm-learning
   continuity-check interval 1s
   mep crosscheck
    mep-id 4001
   !
  !
 !
 domain DOWN0 level 0 id null
  service s10 down-meps id number 10
   continuity-check interval 1s
   mep crosscheck
    mep-id 6001
   !
  !
 !
!
interface MgmtEth0/RP0/CPU0/0
 shutdown
```

```
!
interface TenGigE0/0/0/0
 shutdown
!
interface TenGigE0/0/0/1
 shutdown
!
interface TenGigE0/0/0/2
 shutdown
!
interface TenGigE0/0/0/3
 shutdown
!
interface TenGigE0/0/0/4
 shutdown
!
interface TenGigE0/0/0/5
 shutdown
!
interface TenGigE0/0/0/6
 shutdown
!
interface TenGigE0/0/0/7
 shutdown
!
interface TenGigE0/0/0/8
 shutdown
!
interface TenGigE0/0/0/9
 shutdown
!
interface TenGigE0/0/0/10.1 l2transport
 encapsulation dot1q 1
 ethernet cfm
  mep domain DOWN0 service s10 mep-id 2001
  !
 !
!
interface TenGigE0/0/0/11
 shutdown
!
interface TenGigE0/0/0/12
 shutdown
!
interface TenGigE0/0/0/13
 shutdown
!
interface TenGigE0/0/0/14
 shutdown
!
interface TenGigE0/0/0/15
 shutdown
!
interface TenGigE0/0/0/16
 shutdown
!
interface TenGigE0/0/0/17
 shutdown
!
interface TenGigE0/0/0/18
 shutdown
!
interface TenGigE0/0/0/19
 shutdown
```

```
!
interface TenGigE0/0/0/20
 shutdown
!
interface TenGigE0/0/0/21
 shutdown
!
interface TenGigE0/0/0/22
 shutdown
!
interface TenGigE0/0/0/23
 shutdown
!
interface TenGigE0/0/0/24
 shutdown
!
interface TenGigE0/0/0/25
 shutdown
!
interface TenGigE0/0/0/26
 shutdown
!
interface TenGigE0/0/0/27
 shutdown
!
interface TenGigE0/0/0/28
 shutdown
!
interface TenGigE0/0/0/29
 shutdown
!
interface TenGigE0/0/0/30
 shutdown
!
interface TenGigE0/0/0/31
 shutdown
!
controller Optics0/0/1/0
 breakout 4x10
!
interface HundredGigE0/0/1/1
 shutdown
!
interface FortyGigE0/0/1/2.1 l2transport
 encapsulation dot1q 1
 ethernet cfm
  mep domain UP6 service s6 mep-id 1
  !
 !
!
l2vpn
 xconnect group g1
  p2p p1
   interface TenGigE0/0/0/10.1
   interface FortyGigE0/0/1/2.1
  !
 !
!
end
```

```
RP/0/RP0/CPU0:ios#show ethernet sla statistics on-demand id 1
Mon Sep 11 12:12:00.699 UTC
Source: Interface TenGigE0/0/0/10.1, Domain DOWN0
Destination: Target MEP-ID 6001
================================================================================
```

```
On-demand operation ID #1, packet type 'cfm-delay-measurement'
Started at 12:11:19 UTC Mon 11 September 2017, runs once for 10s

Round Trip Delay
~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 10
    Min: 0.009ms; Max: 0.010ms; Mean: 0.009ms; StdDev: 0.000ms
```

確認

```
One-way Delay (Source->Dest)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 10
    Min: 1912765.961ms; Max: 1912765.961ms; Mean: 1912765.961ms; StdDev: -2147483.648ms


One-way Delay (Dest->Source)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 10
   Min: -1912765.952ms; Max: -1912765.951ms; Mean: -1912765.951ms; StdDev: -2147483.648ms


Round Trip Jitter
~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 9
    Min: 0.000ms; Max: 0.001ms; Mean: 0.000ms; StdDev: 0.000ms


One-way Jitter (Source->Dest)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 9
    Min: 0.000ms; Max: 0.000ms; Mean: 0.000ms; StdDev: 0.000ms


One-way Jitter (Dest->Source)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
```

```
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                  Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 9
    Min: 0.000ms; Max: 0.001ms; Mean: 0.000ms; StdDev: 0.000ms


RP/0/RP0/CPU0:ios#ethernet sla on-demand operation type cfm-syn probe domain DOWN0 source
 interface tenGigE 0/0/0/10.1 target mep-id 6001
Mon Sep 11 12:12:39.259 UTC
Warning: Burst configuration is present and so this profile cannot be represented in the
 MEF-SOAM-PM-MIB configuration tables. However, the statistics are still collected
On-demand operation 2 succesfully created
 / - Completed - statistics will be displayed shortly.
RP/0/RP0/CPU0:ios#
RP/0/RP0/CPU0:ios#
RP/0/RP0/CPU0:ios#show ethernet sla statistics on-demand id 2

Mon Sep 11 12:13:24.825 UTC
Source: Interface TenGigE0/0/0/10.1, Domain DOWN0
Destination: Target MEP-ID 6001
================================================================================
On-demand operation ID #2, packet type 'cfm-synthetic-loss-measurement'
Started at 12:12:41 UTC Mon 11 September 2017, runs once for 10s
Frame Loss Ratio calculated every 10s

One-way Frame Loss (Source->Dest)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:12:41 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 100; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                  Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 1
    Min: 0.000%; Max: 0.000%; Mean; 0.000%; StdDev: 0.000%; Overall: 0.000%


One-way Frame Loss (Dest->Source)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:12:41 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 100; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                  Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 1
    Min: 0.000%; Max: 0.000%; Mean; 0.000%; StdDev: 0.000%; Overall: 0.000%

RP/0/RP0/CPU0:ios#show ethernet cfm local meps verbose
Mon Sep 11 12:13:04.461 UTC
Domain UP6 (level 6), Service s6
Up MEP on FortyGigE0/0/1/2.1 MEP-ID 1
================================================================================
  Interface state: Up      MAC address: 008a.960f.c4a8
  Peer MEPs: 1 up, 0 with errors, 0 timed out (archived)
  Cross-check errors: 0 missing, 0 unexpected

  CCM generation enabled:  Yes, 1s (Remote Defect detected: No)
                           CCM processing offloaded to hardware
  AIS generation enabled:  No
  Sending AIS:             No
  Receiving AIS:           No

  No packets sent/received

Domain DOWN0 (level 0), Service s10
```

```
Down MEP on TenGigE0/0/0/10.1 MEP-ID 2001
================================================================================
  Interface state: Up      MAC address: 008a.960f.c428
  Peer MEPs: 1 up, 0 with errors, 0 timed out (archived)
  Cross-check errors: 0 missing, 0 unexpected

  CCM generation enabled:  Yes, 1s (Remote Defect detected: No)
                           CCM processing offloaded to hardware
  AIS generation enabled:  No
  Sending AIS:             No
  Receiving AIS:           No

  Packet         Sent      Received
  ------  ----------  ----------------------------------------------------------
  DMM             10             0
  DMR              0            10
  SLM            100             0
  SLR              0           100
```

合成損失測定

Y.1731 で定義された損失測定メカニズムを使用できるのはポイントツーポイント ネットワークのみであり、十分なデータ トラフィック フローがある場合にのみ機能します。Y.1731 損失測定メカニズムの難しさは業界全体で認識されており、その結果として、損失を測定するための代替メカニズムが定義および標準化されました。

この代替メカニズムでは、実際のデータ トラフィックの損失は測定せず、代わりに合成 CFM フレームを挿入して、この合成フレームの損失を測定します。データ トラフィック損失の近似値を得るには、統計分析を実行します。この手法を「合成損失測定」（SLM）と呼びます。SLM は Y.1731 標準の最新バージョンに含まれています。SLA を使用して、次の測定を実行します。

- 一方向損失（送信元から宛先）
- 一方向損失（宛先から送信元）

NCS 540 では、SLM は次をサポートしています。

- 物理、バンドル インターフェイス、L2 サブインターフェイス、疑似回線ヘッドエンド インターフェイス、接続回線などのすべての L2 転送インターフェイス。トランスポート ネットワークには EVPN または BGP-MPLS を使用できます。
- アップおよびダウンの MEP。
- パンティングなしに、MIP を通じて SLM パケットを透過的にソフトウェアに渡します。
- 100 の同時 SLM セッション。
- 1000 pps の SLM/SLA トラフィック。

合成損失測定の設定

次の項では、合成損失測定の設定方法について説明します。

Router> enable

RP/0/RP0/CPU0:router # **configure terminal**

(config)RP/0/RP0/CPU0:router # ip sla 1

(config-sla)RP/0/RP0/CPU0:router # profile Prof1 type cfm-loopback

(config-ip-sla)RP/0/RP0/CPU0:router # ethernet y1731 loss SLM domain CISCO evc PROVIDER mpid 5 cos 4 source mpid 6

(config-sla-y1731-loss)RP/0/RP0/CPU0:router # history interval 5

/ * Exit the Y.1731 submode and enters the global configuration mode. * /

(config-sla-y1731-loss)RP/0/RP0/CPU0:router # exit

/ * Schedules the single ended synthetic loss measurement. * /

(config)RP/0/RP0/CPU0:router # ip sla schedule 1 life 100 start-time now

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

CFM 合成損失測定のオンデマンド イーサネット SLA 動作の設定

CFM 合成損失測定のオンデマンド イーサネット SLA 動作を設定するには、特権 EXEC コンフィギュレーション モードで次のコマンドを使用します。

RP/0/RP0/CPU0:router # **ethernet sla on-demand operation type cfm-synthetic-loss-measurement probe domain D1 source interface TenGigE 0/0/0/0 target mac-address 2.3.4**

実行コンフィギュレーション

```
RP/0/RP0/CPU0:ios#show ethernet sla statistics on-demand id 1
Mon Sep 11 12:12:00.699 UTC
Source: Interface TenGigE0/0/0/10.1, Domain DOWN0
Destination: Target MEP-ID 6001
================================================================================
On-demand operation ID #1, packet type 'cfm-delay-measurement'
Started at 12:11:19 UTC Mon 11 September 2017, runs once for 10s
```

確認

```
Round Trip Delay
~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 10
    Min: 0.009ms; Max: 0.010ms; Mean: 0.009ms; StdDev: 0.000ms


One-way Delay (Source->Dest)
~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 10
    Min: 1912765.961ms; Max: 1912765.961ms; Mean: 1912765.961ms; StdDev: -2147483.648ms
```

```
One-way Delay (Dest->Source)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 10
   Min: -1912765.952ms; Max: -1912765.951ms; Mean: -1912765.951ms; StdDev: -2147483.648ms


Round Trip Jitter
~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 9
    Min: 0.000ms; Max: 0.001ms; Mean: 0.000ms; StdDev: 0.000ms


One-way Jitter (Source->Dest)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 9
    Min: 0.000ms; Max: 0.000ms; Mean: 0.000ms; StdDev: 0.000ms


One-way Jitter (Dest->Source)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 9
    Min: 0.000ms; Max: 0.001ms; Mean: 0.000ms; StdDev: 0.000ms
```

イーサネット OAM の設定例

ここでは、次の設定例について説明します。

イーサネット CFM の設定例

ここでは、次の設定例について説明します。

イーサネット CFM ドメインの設定：例

次に、イーサネット CFM の基本的なドメインを設定する例を示します。

```
configure
 ethernet cfm
  traceroute cache hold-time 1 size 3000
  domain Domain_One level 1 id string D1
  commit
```

イーサネット CFM サービスの設定：例

次に、イーサネット CFM ドメインのサービスを作成する例を示します。

```
  service Bridge_Service bridge group BD1 bridge-domain B1
  service Cross_Connect_1 xconnect group XG1 p2p X1
  commit
```

イーサネット CFM サービス設定の柔軟なタギング：例

次に、CFM ドメイン サービスのアップ MEP からの CFM パケット内のタグの数を設定する例を示します。

```
configure
 ethernet cfm
  domain D1 level 1
   service S2 bridge group BG1 bridge-domain BD2
    tags 1
    commit
```

イーサネット CFM サービス設定の連続性チェック：例

次に、イーサネット CFM サービスに対する連続性チェック オプションを設定する例を示します。

```
   continuity-check archive hold-time 100
   continuity-check loss auto-traceroute
   continuity-check interval 100ms loss-threshold 10
   commit
```

イーサネット CFM サービス設定の MIP の作成：例

次に、イーサネット CFM サービスに MIP の自動作成を有効にする例を示します。

```
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# mip auto-create all
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit
```

イーサネット CFM サービス設定のクロスチェック：例

次に、イーサネット CFM サービスの MEP に対してクロスチェックを設定する例を示します。

```
   mep crosscheck
    mep-id 10
    mep-id 20
    commit
```

他のイーサネット CFM サービス パラメータの設定：例

次に、その他のイーサネット CFM サービス オプションを設定する例を示します。

```
   maximum-meps 4000
   log continuity-check errors
   commit
   exit
  exit
 exit
```

MEP の設定：例

次に、インターフェイスでイーサネット CFM に MEP を設定する例を示します。

```
interface TenGigE 0/0/0/1
 ethernet cfm
 mep domain Dm1 service Sv1 mep-id 1
 commit
```

イーサネット CFM の show コマンド：例

次に、イーサネット接続障害管理（CFM）の設定を確認する例を示します。

例 1

次に、インターフェイス上で作成されたすべてのメンテナンス ポイントを表示する例を示します。

```
RP/0/RP0/CPU0:router# show ethernet cfm local maintenance-points

Domain/Level         Service              Interface         Type   ID   MAC
-------------------- -------------------- ----------------- ------ ---- --------
fig/5                bay                  Gi0/10/0/12       Dn MEP    2 44:55:66
fig/5                bay                  Gi0/0/1/0         MIP         55:66:77
fred/3               barney               Gi0/1/0/0         Dn MEP    5 66:77:88!
```

例 2

次に、すべてのドメインのすべての CFM 設定エラーを表示する例を示します。

```
RP/0/RP0/CPU0:router# show ethernet cfm configuration-errors

Domain fig (level 5), Service bay
* MIP creation configured using bridge-domain blort, but bridge-domain blort does not
exist.
 * An Up MEP is configured for this domain on interface TenGigE0/0/0/3 and an Up MEP is
 also configured for domain blort, which is at the same level (5).
 * A MEP is configured on interface TenGigE0/0/0/1 for this domain/service, which has
CC interval 100ms, but the lowest interval supported on that interface is 1s
```

例 3

次に、ローカルのメンテナンス エンドポイント（MEP）の動作状態を表示する例を示します。

```
RP/0/RP0/CPU0:router# show ethernet cfm local meps

 A - AIS received                I - Wrong interval
 R - Remote Defect received      V - Wrong Level
 L - Loop (our MAC received)     T - Timed out (archived)
 C - Config (our ID received)    M - Missing (cross-check)
 X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)
 P - Peer port down

Domain foo (level 6), Service bar
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
  100 Gi1/1/0/1 (Up)       Up     0/0   N  A       L7

Domain fred (level 5), Service barney
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
    2 Gi0/1/0/0 (Up)       Up     3/2   Y  RPC     L6
Domain foo (level 6), Service bar
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
  100 Gi1/1/0/1 (Up)       Up     0/0   N  A

Domain fred (level 5), Service barney
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
    2 Gi0/1/0/0 (Up)       Up     3/2   Y  RPC
```

例 4

次に、ローカル MEP が検出するその他のメンテナンス エンドポイント（MEP）の動作状態を表示する例を示します。

```
RP/0/RP0/CPU0:router# show ethernet cfm peer meps

Flags:
 > - Ok                          I - Wrong interval
 R - Remote Defect received      V - Wrong level
 L - Loop (our MAC received)     T - Timed out
 C - Config (our ID received)    M - Missing (cross-check)
 X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)

Domain fred (level 7), Service barney
Down MEP on TenGigE0/0/0/1, MEP-ID 2
==============================================================================
St     ID MAC address    Port    Up/Downtime   CcmRcvd SeqErr   RDI Error
-- ----- -------------- ------- ------------ --------- ------ ----- -----
 >      1 0011.2233.4455 Up      00:00:01         1234      0     0     0
R>      4 4455.6677.8899 Up      1d 03:04         3456      0   234     0
L       2 1122.3344.5566 Up      3w 1d 6h         3254      0     0  3254
C       2 7788.9900.1122 Test    00:13            2345      6    20  2345
X       3 2233.4455.6677 Up      00:23              30      0     0    30
I       3 3344.5566.7788 Down    00:34           12345      0   300  1234
V       3 8899.0011.2233 Blocked 00:35              45      0     0    45
 T      5 5566.7788.9900         00:56              20      0     0     0
M       6                                           0      0     0     0
U>      7 6677.8899.0011 Up      00:02             456      0     0     0

Domain fred (level 7), Service fig
Down MEP on TenGigE0/0/0/12, MEP-ID 3
==============================================================================
```

```
St    ID MAC address    Port    Up/Downtime  CcmRcvd SeqErr   RDI Error
-- ----- -------------- ------- ----------- -------- ------ ----- -----
 >     1 9900.1122.3344 Up      03:45           4321      0     0     0
```

例 5

次に、ローカルMEPが検出するその他のメンテナンスエンドポイント（MEP）の動作状態を詳細に表示する例を示します。

```
RP/0/RP0/CPU0:router#  show ethernet cfm peer meps detail
Domain dom3 (level 5), Service ser3
Down MEP on TenGigE0/0/0/1 MEP-ID 1
================================================================================
Peer MEP-ID 10, MAC 0001.0203.0403
   CFM state: Wrong level, for 00:01:34
   Port state: Up
   CCM defects detected:    V - Wrong Level
   CCMs received: 5
     Out-of-sequence:             0
     Remote Defect received:      5
     Wrong Level:                 0
     Cross-connect (wrong MAID):  0
     Wrong Interval:              5
     Loop (our MAC received):     0
     Config (our ID received):    0
Last CCM received 00:00:06 ago:
     Level: 4, Version: 0, Interval: 1min
     Sequence number: 5, MEP-ID: 10
     MAID: String: dom3, String: ser3
     Port status: Up, Interface status: Up


Domain dom4 (level 2), Service ser4
Down MEP on TenGigE0/0/0/2 MEP-ID 1
================================================================================
Peer MEP-ID 20, MAC 0001.0203.0402
   CFM state: Ok, for 00:00:04
   Port state: Up
   CCMs received: 7
     Out-of-sequence:             1
     Remote Defect received:      0
     Wrong Level:                 0
     Cross-connect (wrong MAID):  0
     Wrong Interval:              0
     Loop (our MAC received):     0
  Config (our ID received):    0
Last CCM received 00:00:04 ago:
     Level: 2, Version: 0, Interval: 10s
     Sequence number: 1, MEP-ID: 20
     MAID: String: dom4, String: ser4
     Chassis ID: Local: ios; Management address: 'Not specified'
     Port status: Up, Interface status: Up

Peer MEP-ID 21, MAC 0001.0203.0403
   CFM state: Ok, for 00:00:05
   Port state: Up
   CCMs received: 6
     Out-of-sequence:             0
     Remote Defect received:      0
     Wrong Level:                 0
     Cross-connect (wrong MAID):  0
     Wrong Interval:              0
```

```
     Loop (our MAC received):     0
     Config (our ID received):    0
Last CCM received 00:00:05 ago:
     Level: 2, Version: 0, Interval: 10s
     Sequence number: 1, MEP-ID: 21
     MAID: String: dom4, String: ser4
     Port status: Up, Interface status: Up


Peer MEP-ID 601, MAC 0001.0203.0402
   CFM state: Timed Out (Standby), for 00:15:14, RDI received
   Port state: Down
   CCM defects detected:    Defects below ignored on local standby MEP
                            I - Wrong Interval
                            R - Remote Defect received
                            T - Timed Out
                            P - Peer port down
   CCMs received: 2
     Out-of-sequence:             0
     Remote Defect received:      2
     Wrong Level:                 0

     Wrong Interval:              2
     Loop (our MAC received):     0
     Config (our ID received):    0
   Last CCM received 00:15:49 ago:
     Level: 2, Version: 0, Interval: 10s
     Sequence number: 1, MEP-ID: 600
     MAID: DNS-like: dom5, String: ser5
     Chassis ID: Local: ios; Management address: 'Not specified'
     Port status: Up, Interface status: Down
```

CFM 設定の AIS：例

例 1

この例では、CFM ドメイン サービスのアラーム表示信号（AIS）の送信を設定します。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ethernet cfm
RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RP0/CPU0:router(config-cfm-dmn)# service S1 bridge group BG1 bridge-domain BD2
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# ais transmission interval 1m cos 7

RP/0/RP0/CPU0:routerconfigure
RP/0/RP0/CPU0:router(config)# ethernet cfm
RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RP0/CPU0:router(config-cfm-dmn)# service Cross_Connect_1 xconnect group XG1 p2p
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# ais transmission interval 1m cos 7
```

例 2

この例では、AIS パケットまたは LCK パケットをいつ受信したかを表示する接続障害管理（CFM）の AIS ロギングを設定します。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ethernet cfm
RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RP0/CPU0:router(config-cfm-dmn)# service S2 bridge group BG1 bridge-domain BD2
```

```
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log ais

RP/0/RP0/CPU0:routerconfigure
RP/0/RP0/CPU0:router(config)# ethernet cfm
RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RP0/CPU0:router(config-cfm-dmn)# service Cross_Connect_1 xconnect group XG1 p2p
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log ais
```

次に、CFM インターフェイス上で AIS の送信を設定する例を示します。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/2
RP/0/RP0/CPU0:router(config-if)# ethernet cfm
RP/0/RP0/CPU0:router(config-if-cfm)# ais transmission up interval 1m cos 7
```

CFM の show コマンドの AIS：例

ここでは、次の設定例について説明します。

show ethernet cfm interfaces ais コマンド：例

次に、インターフェイス AIS テーブルに公開されている情報を表示する例を示します。

```
RP/0/RP0/CPU0:router# show ethernet cfm interfaces ais

Defects (from at least one peer MEP):
 A - AIS received                I - Wrong interval
 R - Remote Defect received      V - Wrong Level
 L - Loop (our MAC received)     T - Timed out (archived)
 C - Config (our ID received)    M - Missing (cross-check)
 X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)
 P - Peer port down              D - Local port down

                          Trigger                  Transmission
                      AIS ---------    Via    --------------------------
Interface (State)     Dir L Defects  Levels   L Int Last started  Packets
--------------------- --- - -------  -------  - --- ------------ --------
TenGigE0/0/0/0 (Up)    Dn  5 RPC      6        7 1s  01:32:56 ago     5576
TenGigE0/0/0/0 (Up)    Up  0 M        2,3      5 1s  00:16:23 ago      983
TenGigE0/0/0/1 (Dn)    Up    D                 7 60s 01:02:44 ago     3764
TenGigE0/0/0/2 (Up)    Dn  0 RX       1!
```

show ethernet cfm local meps コマンド：例

例 1：デフォルト

次に、ローカルのメンテナンスエンドポイント（MEP）の統計情報を表示する例を示します。

```
RP/0/RP0/CPU0:router# show ethernet cfm local meps

 A - AIS received                I - Wrong interval
 R - Remote Defect received      V - Wrong Level
 L - Loop (our MAC received)     T - Timed out (archived)
 C - Config (our ID received)    M - Missing (cross-check)
 X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)
 P - Peer port down
```

```
Domain foo (level 6), Service bar
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
  100 Gi1/1/0/1 (Up)       Up     0/0   N  A       7

Domain fred (level 5), Service barney
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
    2 Gi0/1/0/0 (Up)       Up     3/2   Y  RPC     6
```

例 2：ドメイン サービス

次に、ドメイン サービスの MEP の統計情報を表示する例を示します。

```
RP/0/RP0/CPU0:router# show ethernet cfm local meps domain foo service bar detail

Domain foo (level 6), Service bar
Down MEP on TenGigE0/0/0/1, MEP-ID 100
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)

  CCM generation enabled:  No
  AIS generation enabled:  Yes (level: 7, interval: 1s)
  Sending AIS:             Yes (started 01:32:56 ago)
  Receiving AIS:           Yes (from lower MEP, started 01:32:56 ago)

Domain fred (level 5), Service barney
Down MEP on TenGigE0/0/0/1, MEP-ID 2
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 3 up, 2 with errors, 0 timed out (archived)
  Cross-check defects: 0 missing, 0 unexpected

  CCM generation enabled:  Yes (Remote Defect detected: Yes)
  CCM defects detected:    R - Remote Defect received
                           P - Peer port down
                           C - Config (our ID received)
  AIS generation enabled:  Yes (level: 6, interval: 1s)
  Sending AIS:             Yes (to higher MEP, started 01:32:56 ago)
  Receiving AIS:           No
```

例 4：詳細

次に、ドメイン サービスの MEP の詳細な統計情報を表示する例を示します。

```
RP/0/RP0/CPU0:router# show ethernet cfm local meps detail

Domain foo (level 6), Service bar
Down MEP on TenGigE0/0/0/1, MEP-ID 100
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)

  CCM generation enabled:  No
  AIS generation enabled:  Yes (level: 7, interval: 1s)
  Sending AIS:             Yes (started 01:32:56 ago)
  Receiving AIS:           Yes (from lower MEP, started 01:32:56 ago)

Domain fred (level 5), Service barney
Down MEP on TenGigE0/0/0/1, MEP-ID 2
```

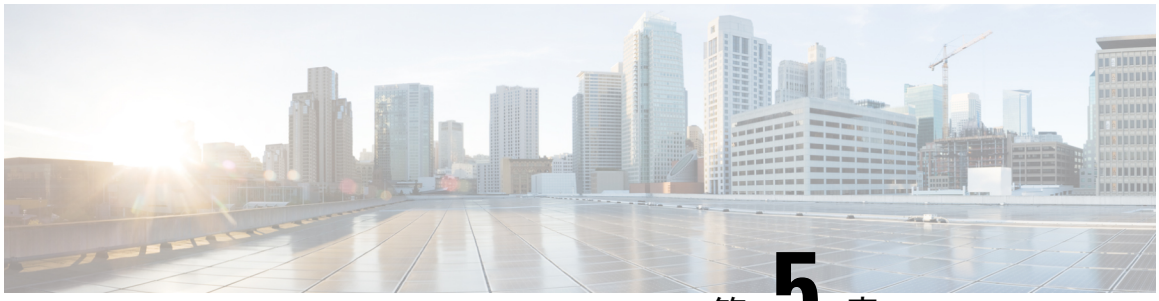
```
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 3 up, 2 with errors, 0 timed out (archived)
  Cross-check defects: 0 missing, 0 unexpected

  CCM generation enabled:  Yes (Remote Defect detected: Yes)
  CCM defects detected:    R - Remote Defect received
                           P - Peer port down
                           C - Config (our ID received)
  AIS generation enabled:  Yes (level: 6, interval: 1s)
  Sending AIS:             Yes (to higher MEP, started 01:32:56 ago)
  Receiving AIS:           No
```

show ethernet cfm local meps detail コマンド：例

show ethernet cfm local meps detail コマンドを使用して MEP 関連の EFD ステータス情報を表示します。次に、EFD が MEP-ID 100 に対してトリガーされる例を示します。

```
RP/0/RP0/CPU0:router# show ethernet cfm local meps detail

Domain foo (level 6), Service bar
Down MEP on TenGigE0/0/0/1, MEP-ID 100
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)
  Cross-check errors: 2 missing, 0 unexpected

  CCM generation enabled:  No
  AIS generation enabled:  Yes (level: 7, interval: 1s)
  Sending AIS:             Yes (started 01:32:56 ago)
  Receiving AIS:           Yes (from lower MEP, started 01:32:56 ago)
  EFD triggered:           Yes

Domain fred (level 5), Service barney
Down MEP on TenGigE0/0/0/1, MEP-ID 2
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 3 up, 0 with errors, 0 timed out (archived)
  Cross-check errors: 0 missing, 0 unexpected

  CCM generation enabled:  Yes (Remote Defect detected: No)
  AIS generation enabled:  Yes (level: 6, interval: 1s)
  Sending AIS:             No
  Receiving AIS:           No
  EFD triggered:           No
```

（注） また、**show interfaces** コマンドと **show interfaces brief** コマンドを使用すると、インターフェイス上で EFD がトリガーされていることを確認できます。EFD トリガーが発生する場合は、これらのコマンドにより、アップとしてインターフェイスのステータスを、ダウンとしてラインプロトコル ステートを表示します。



第 5 章

Integrated Routing and Bridging（IRB）

BVIは、通常のルーテッドインターフェイスのように動作する、ルータ内の仮想インターフェイスです。BVIでブリッジング自体はサポートされませんが、ルータ内の対応するブリッジドメインからルーテッドインターフェイスへのゲートウェイとして機能します。

設定可能なMACアドレスのサポートとは別に、BVIではレイヤ3属性だけがサポートされ、次の特性があります。

- BVIインターフェイスで上書きされていない限り、ローカル シャーシのMACアドレスプールから取得されたMACアドレスを使用します。
- **interface bvi** コマンドを使用してインターフェイス タイプとして設定され、ブリッジドドメインのセグメントのホストと同じサブネット上にあるIPv4アドレスを使用します。
- BVI IDはブリッジ ドメインIDとは無関係です。これらのIDはCisco IOSソフトウェアでの場合のように相関している必要はありません。
- **routed interface bvi** コマンドを使用して、ブリッジ グループに関連付けられます。

ブリッジ グループ仮想インターフェイス

BVIは、通常のルーテッドインターフェイスのように動作する、ルータ内の仮想インターフェイスです。BVIでブリッジング自体はサポートされませんが、ルータ内の対応するブリッジドメインからルーテッドインターフェイスへのゲートウェイとして機能します。

BVI はレイヤ 3 属性のみをサポートしており、次の特性があります。

- BVI インターフェイスで上書きされていない限り、ローカル シャーシの MAC アドレス プールから取得された MAC アドレスを使用します。
- **interface bvi** コマンドを使用してインターフェイス タイプとして設定され、ブリッジド ドメインのセグメントのホストと同じサブネット上にある IPv4 アドレスを使用します。
- BVI ID はブリッジ ドメイン ID とは無関係です。これらの ID は Cisco IOS ソフトウェアでの場合のように相関している必要はありません。
- **routed interface bvi** コマンドを使用して、ブリッジ グループに関連付けられます。
- BVI インターフェイスは、1 ～ 4294967295 の数値範囲をサポートしています。

BVI でサポートされている機能

- 次のインターフェイス コマンドが BVI でサポートされています。
  - **arp purge-delay**
  - arp timeout
  - **bandwidth**（デフォルトは 10 Gbps であり、BVI のルーティング プロトコルのコスト メトリックとして使用されます）
  - **ipv4**
  - **ipv6**
  - **mac-address**
  - **shutdown**

BVI インターフェイスおよびライン プロトコルの状態

ルータの一般的なインターフェイスの状態のように、BVIにはインターフェイスとラインプロトコルの状態の両方があります。

- BVI インターフェイスの状態は次が発生するときに Up です。
  - BVI インターフェイスが作成される。
  - **routed interface bvi** コマンドで設定されているブリッジ ドメインに少なくとも 1 つの使用可能なアクティブ ブリッジ ポートがある（接続回線（AC）または疑似回線（PW））。

（注） BVIは、そのBVIのブリッジドメインに関連付けられたすべてのブリッジポート（イーサネットフローポイント（EFP））がダウンしている場合、ダウン状態に移行します。ただし、すべてのEFPがダウンしていても、少なくとも1つのブリッジポートがアップの場合、BVI はアップのままです。

- 次の特性によって、BVI ライン プロトコルの状態がアップである場合が決定されます。
  - ブリッジ ドメインがアップ状態である。
  - BVI IP アドレスが、ルータの別のアクティブ インターフェイスのその他の IP アドレスと競合していない。

IRB 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。

IRBを設定する前に、次のタスクが実行されており、条件を満たしていることを確認してください。

- ブリッジ仮想インターフェイス（BVI）に設定する IP アドレッシングおよび他のレイヤ 3 情報を理解しています。
- すべての BVI の共通のグローバル MAC アドレスを上書きする場合は、MAC アドレス計画を完了します。
- BVIインターフェイスのスタティックまたはダイナミック ルーティングを実行して、BVI ネットワーク アドレスがアドバタイズされていることを確認します。

IRB の設定の制約事項

IRB を設定する前に、次の制限事項を確認してください。

- 任意のブリッジ ドメインで設定できる BVI は 1 つだけです。
- 同じ BVI を複数のブリッジ ドメインで設定できません。
- 次の領域は、（BVI を使用した）レイヤ 2 ブリッジングでサポートされていません。
  - ブリッジでのスタティック MAC エントリ設定。

- グローバル コンフィギュレーション モードでの MAC エージング設定。
- MAC ラーニングの無効化。
- VLAN 書き換え。

- BVI インターフェイス上の QoS 設定は出力ではサポート対象外。

IRB の設定方法

この項では、次の設定作業について説明します。

ブリッジ グループ仮想インターフェイスの設定

BVI を設定するには、次の手順を実行します。

設定時の注意事項

BVI を設定する場合は、次の注意事項を考慮してください。

- BVI には、ブリッジドセグメントのホストと同じサブネット上にある IPv4 または IPv6 アドレスを割り当てる必要があります。

手順

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 bvi** *identifier*

例：

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

BVI を指定または作成します。ここで、*identifier* は 1 ～ 65535 の数値です。

ステップ 3 **ipv4 address** *ipv4-address mask* [**secondary**] **ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**] [**route-tag** *route-tag value*]

例：

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 10.10.0.4 255.255.255.0
```

インターフェイスのプライマリまたはセカンダリ IPv4 アドレスまたは IPv6 アドレスを指定します。

ステップ 4 **arp purge-delay** *seconds*

例：

```
RP/0/RP0/CPU0:router(config-if)#arp purge-delay 120
```

（任意）インターフェイスがダウンするときの、アドレス解決プロトコル（ARP）テーブルエントリのパージの遅延時間を（秒単位で）指定します。

指定できる範囲は 1 ～ 65535 です。デフォルトでは、パージ遅延は設定されていません。

ステップ 5 **arp timeout** *seconds*

例：

```
RP/0/RP0/CPU0:router(config-if)# arp timeout 12200
```

（任意）インターフェイスで学習されたダイナミック エントリを ARP キャッシュに残す時間を指定します。

値の範囲は 30 ～ 2144448000 秒です。デフォルトは 14,400 秒（4 時間）です。

ステップ 6 **bandwidth** *rate*

例：

```
RP/0/RP0/CPU0:router(config-if)# bandwidth 1000000
```

（任意）インターフェイスに割り当てる帯域幅の量（kbps 単位）を指定します。この数値は、BVI のルーティング プロトコルでコスト メトリックとして使用されます。

指定できる範囲は 0 ～ 4294967295 です。デフォルトは 10000000（10 Gbps）です。

ステップ 7 **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。

レイヤ 2 AC インターフェイスの設定

BVI によるルーティング用のレイヤ 2 AC インターフェイスを設定するには、次の手順を実行します。

手順

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] l2transport**

例：

```
RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/0.1 l2transport
```

ギガビット イーサネットまたは 10 ギガビット イーサネットのインターフェイスまたはサブインターフェイス上でレイヤ 2 転送モードを有効にし、インターフェイスまたはサブインターフェイス コンフィギュレーション モードを開始します。

ステップ 3 **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。

ブリッジ グループの設定およびブリッジ ドメインへのインターフェイスの割り当て

ブリッジ グループを設定し、ブリッジ ドメインにインターフェイスを割り当てるには、次の手順を実行します。

手順

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 **l2vpn**

例：

```
RP/0/RP0/CPU0:router(config)# l2vpn
```

L2VPN 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 **bridge group** *bridge-group-name*

例：

```
RP/0/RP0/CPU0:router(config-l2vpn)# bridge group 10
```

ブリッジ グループを作成し、L2VPN ブリッジ グループ 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 **bridge-domain** *bridge-domain-name*

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-bg)# bridge-domain BD_1
```

ブリッジ ドメインを作成し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

ステップ 5 **interface** [**HundredGigE | TenGigE**

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# interface HundredGigE 0/0/1/0.1
```

100 ギガビット イーサネットまたは 10 ギガビット イーサネットのインターフェイスを指定したブリッジ ドメインに関連付け、L2VPN ブリッジ グループ ブリッジ ドメイン接続回線コンフィギュレーション モードを開始します。

ブリッジ ドメインに関連付けるすべてのインターフェイスに対して必要なだけこの手順を繰り返します。

ステップ 6 **end** または **commit**

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-ac)# end
```

または

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-ac)# 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。

ブリッジ ドメインでのルーテッド インターフェイスとしての BVI の関連付け

ブリッジ ドメインのルーテッド インターフェイスとして BVI を関連付けるには、次の手順を実行します。

手順

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 **l2vpn**

例：

```
RP/0/RP0/CPU0:router(config)# l2vpn
```

L2VPN 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 **bridge group** *bridge-group-name*

例：

```
RP/0/RP0/CPU0:router(config-l2vpn)# bridge group BG_test
```

ブリッジ グループを作成し、L2VPN ブリッジ グループ 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 **bridge-domain** *bridge-domain-name*

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-bg)# bridge-domain 1
```

ブリッジ ドメインを作成し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

ステップ 5 **routed interface bvi** *identifier*

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# routed interface bvi 1
```

指定した BVI をブリッジ ドメインに割り当てられたインターフェイスのルーテッド インターフェイスとして関連付けます。

ステップ 6 **end** または **commit**

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# end
```

または

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# 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。

BVI に関する情報の表示

BVI ステータスおよびパケット カウンタに関する情報を表示するには、次のコマンドを使用します。

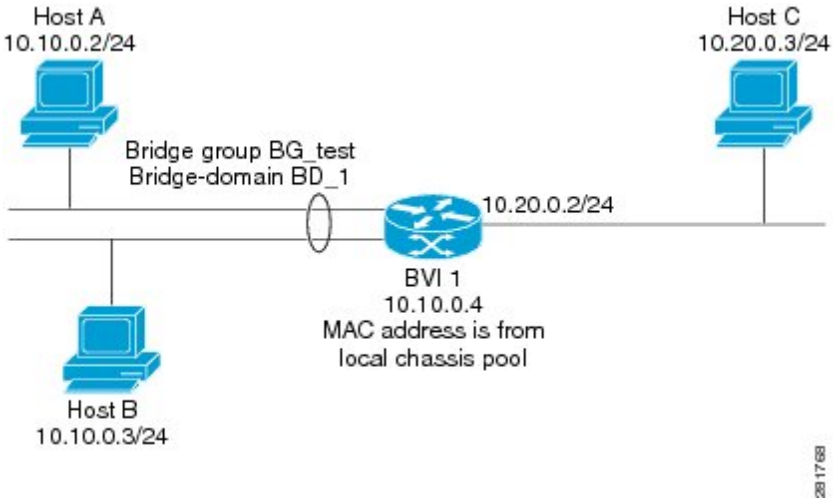
show interfaces bvi *identifier* [**accounting** | **brief** | **description** | **detail**]	指定した BVI のインターフェイス ステータス、ライン プロトコルの状態、およびパケット カウンタを表示します。
show adjacency bvi *identifier* [**detail** | **remote**]	指定した BVI への隣接ごとのパケットおよびバイト送信カウンタを表示します。
show l2vpn bridge-domain detail	BVI がダウンの理由を表示します。

IRB に関する追加情報

IRB を使用したパケット フロー

次の図に、IRB の実装の簡略化された機能図を示し、ホスト A、B、および C 間でのさまざまなパケット フローについて説明します。この例では、ホスト C は同じルータとの接続が確立されているネットワーク上にあります。実際には、別のルータがホスト C と表示されたルータの間に存在可能です。

図 *8 :* ホスト間の *IRB* パケット フロー



IRB をルータで設定すると、次の処理が実行されます。

- ARP 要求は、ブリッジ ドメインの一部であるホストと BVI の間で解決されます。
- 宛先 MAC アドレスが BVI MAC アドレスと一致する場合、ブリッジド インターフェイスのホストからのすべてのパケットが BVI に送信されます。それ以外の場合、パケットはブリッジングされます。
- ルーテッドネットワークのホスト宛てのパケットの場合、BVI はルーテッドインターフェイスに送信する前にルーティング エンジンにパケットを転送します。
- ブリッジド インターフェイスのホストが送信元または宛先であるすべてのパケットは、BVI に最初に送信されます（パケットがブリッジドメイン上のホスト宛ての場合を除く）。
- ルーテッド インターフェイスのルータに入るブリッジ ドメインのセグメント上のホスト宛てパケットの場合、BVI は適切なブリッジインターフェイス経由で転送を行うブリッジング エンジンにパケットを転送します。

ブリッジ ドメインでホスト A がホスト B に送信するときのパケット フロー

10.10.0.0 ネットワークのブリッジ ドメインでホスト A がホスト B にデータを送信すると、ルーティングは実行されません。ホストは同じサブネット上にあり、パケットはルータのセグメント インターフェイス間でブリッジングされます。

ブリッジ ドメインからルーテッド インターフェイスにホスト A がホスト C に送信するときのパケット フロー

IRB ブリッジ ドメインからルーティング ドメインにホスト A がホスト C にデータを送信するとき、この図のホスト情報を使用して、次が実行されます。

- ホスト A は、パケットを BVI に送信します（ARP 要求がホストと BVI の間で解決される限り）。パケットには次の情報があります。
  - ホスト A の送信元 MAC アドレス。
  - BVI の宛先 MAC アドレス。
- ホスト C は別のネットワークにあり、ルーティングされる必要があるため、BVI は次の情報を使用してルーテッド インターフェイスにパケットを転送します。
  - ホスト A の IP 送信元 MAC アドレス（10.10.0.2）は BVI の MAC アドレス（10.10.0.4）に変更されます。
  - IP 宛先アドレスは、ホスト C の IP アドレス（10.20.0.3）です。
- インターフェイス 10.20.0.2 は、ルーテッド BVI 10.10.0.4 からのパケットの受信を認識します。パケットは、次にインターフェイス 10.20.0.2 を通じてホスト C にルーティングされます。

ルーテッド インターフェイスからブリッジ ドメインにホスト C がホスト B に送信するときのパケット フロー

IRB ルーティング ドメインからブリッジ ドメインにホスト C がホスト B にデータを送信するとき、この図のホスト情報を使用して、次が実行されます。

- パケットは、次の情報を使用してルーティング ドメインに入ります。
  - MAC 送信元アドレス：ホスト C の MAC。
  - MAC 宛先アドレス：入力インターフェイス 10.20.0.2 の MAC。
  - IP 送信元アドレス：ホスト C（10.20.0.3）の IP アドレス。
  - IP 宛先アドレス：ホスト B（10.10.0.3）の IP アドレス。

- インターフェイス 10.20.0.2 はパケットを受信すると、ルーティング テーブルを確認し、パケットが 10.10.0.4 の BVI に転送される必要があるかを決定します。
- ルーティング エンジンは BVI 宛てのパケットを取り込み、BVI の対応するブリッジ ドメインに転送します。次にパケットは、ブリッジング テーブルにホスト B の宛先 MAC アドレスがある場合は適切なインターフェイスを通じてブリッジングされます。または、ブリッジングテーブルにそのアドレスがない場合はブリッジグループ内のすべてのインターフェイスにフラッディングされます。

IRB の設定例

ここでは、次の設定例について説明します。

基本的な IRB 設定：例

次に、最も基本的な IRB 設定を行う例を示します。

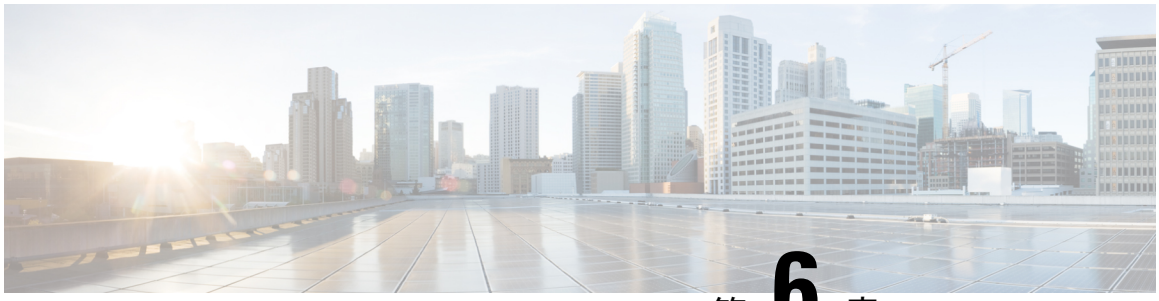
```
! Configure the BVI and its IPv4 address
!
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)#interface bvi 1
RP/0/RP0/CPU0:router(config-if)#ipv4 address 10.10.0.4 255.255.255.0
RP/0/RP0/CPU0:router(config-if))# exit
!
! Configure the Layer 2 AC interface
!
RP/0/RP0/CPU0:router(config)#interface HundredGigE 0/0/1/0 l2transport
RP/0/RP0/CPU0:router(config-if))# exit
!
! Configure the L2VPN bridge group and bridge domain and assign interfaces
!
RP/0/RP0/CPU0:router(config)#l2vpn
RP/0/RP0/CPU0:router(config-l2vpn)#bridge group 10
RP/0/RP0/CPU0:router(config-l2vpn-bg)#bridge-domain 1
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)#interface HundredGigE 0/0/1/0
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-if)# exit
!
! Associate a BVI to the bridge domain
!
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# routed interface bvi 1
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# commit
```

BVI および VRRP を使用した IRB の設定：例

次に、BVI および VRRP の IRB サポートに対する関連設定領域の部分的なルータ設定の例を示します。

（注） VRRPv6 もサポートされます。

```
l2vpn
 bridge group IRB
  bridge-domain IRB-EDGE
   interface TenGigE0/0/0/8
!
   routed interface BVI 100
!
interface TenGigE0/0/0/8
  l2transport
!
interface BVI 100
 ipv4 address 10.21.1.1 255.255.255.0
!
router vrrp
 interface BVI 100
  vrrp 1 ipv4 10.21.1.100
  vrrp 1 priority 100
 !
```



第 6 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、1 つに束ねられたポートのグループであり、1 つのリンクとして振る舞います。リンク バンドルには次のような利点があります。

- 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。

イーサネットリンクバンドルの制限事項と互換性に関する特性

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限を示します。

- 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。
- 単一のルータで最大 256 のバンドル インターフェイスと、バンドルあたりデフォルトで 64 のメンバー リンクをサポートしています。
- 単一のルータで最大 1024 のバンドル サブインターフェイスと、バンドルあたり最大 64 のメンバー リンクをサポートしています。
- HQoS プロファイルが有効な場合、デフォルトでは最大 256 のトランク（物理 + サブインターフェイス）が使用できます。それよりも多くのトランクが必要な場合は、hw-module profile qos max-trunks <256/512/1024> コマンドを設定します。バンドル インターフェイスで HQoS を有効にすると、最大 4 つのプライオリティ レベルがサポートされます。
- デフォルトで使用可能なトランクは 256 です。より多くのトランクが必要な場合は、**hw-module profile bundle-scale <256/512/1024>** コマンドを設定できます。バンドル サブインターフェイスで HQoS をイネーブルにすると、最大 4 つのプライオリティ レベルがサポートされます。
- 次の制限は、HQoS プロファイルを持つサポートされているバンドル メンバーの数に適用されます。
  - 最大 1,024 トランク（128 の物理インターフェイス + 896 のサブインターフェイス）と 16 のバンドル メンバー。
  - 最大 256 トランク（128 の物理インターフェイス + 128 のサブインターフェイス）と 64 のバンドル メンバー。
  - 最大 512 トランク（128 の物理インターフェイス + 384 のサブインターフェイス）と 32 のバンドル メンバー。
- バンドル サブインターフェイスとバンドルごとのメンバー数には、次の制限が適用されます。
  - それぞれに最大 16 のメンバーリンクを含むバンドル サブインターフェイスの最大数は 1024 です。
  - それぞれに最大 64 のメンバーリンクを含むバンドル サブインターフェイスの最大数は 256 です。
  - それぞれに最大 32 メンバーリンクを含むバンドル サブインターフェイスの最大数は 512 です。

- 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。
- 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。
- 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。
- 物理リンクのみがバンドル メンバーになることができます。
- 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、内部プロセスによってメンバー 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ー上で送信されます。
- MC-LAG はサポートされていません。

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。

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され、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。

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る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 32 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、バンドルでプライオリティが最も高いリンクがアクティブになり、プライオリティが 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。

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。

これを行うには、**lacp fast-switchover** コマンドを使用します。

LACP フォールバック

LACP フォールバック機能を使用すると、ポートチャネルがピアから Link Aggregation Control Protocol（LACP）のプロトコルデータ ユニット（PDU）を受信する前に、アクティブなLACP インターフェイスが Link Aggregation Control Protocol（LACP）のポートチャネルを確立することができます。LACP フォールバック機能を設定することで、サーバからLACP PDU を受信する前にサーバが LAG を起動し、1 つのポートをアクティブに保つことができます。これにより、サーバは 1 つのイーサネット ポートを介して PXE サーバへの接続を確立し、そのブートイメージをダウンロードして起動プロセスを続行できます。サーバの起動プロセスが完了すると、サーバは LACP ポートチャネルを完全に形成します。

失敗状況

次の障害が発生した場合、MC-LAG は DHD に対しては変更のないバンドル インターフェイスを表示しながら、影響を受けていない POA にトラフィックをスイッチングすることで、冗長性を提供します。

- リンク障害：POA のいずれかと DHD 間のポートまたはリンクに障害が発生。
- デバイス障害：POA のいずれかにメルトダウンまたはリロードが発生し全体的な接続の喪失が発生（DHD、コアおよび他の POA に対して）。
- コアの分離：POA がコア ネットワークへの接続を失ったために値がなくなり、DHD とのトラフィックの転送が不可能。

POA 間で接続の喪失が発生すると、両方のデバイスは相手側でデバイス障害が発生したと見なし、両方がアクティブ ロールを担うよう試みます。これは、スプリット ブレーンのシナリオと呼ばれ、次のいずれかで発生する可能性があります。

- その他の接続はすべて残り、POA 間リンクだけ失われた場合。
- 1 つの POA がコア ネットワークから切断された場合（つまり 2 つの POA 間の接続がコア ネットワーク経由である場合のコア分離シナリオ）。

MC-LAG 自体はこの状況を回避する方法を提供しません。POA 間の接続の復元力が必須です。バンドル内でアクティブになるリンク数に制限を設定することで、問題を低減する責任は、DHD に与えられます。任意の時点で、POA の 1 つに接続しているリンクのみがアクティブになります。

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

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 リンクとスタンバイ リンクのプライオリティは、**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 8 **interface HundredGigE** interface-path-id

例：

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

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。

HundredGigE キーワードを入力して、インターフェイス タイプを指定します。*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 9 **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は、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 10 **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 リンクのプライオリティを最も高くし（最も小さい値）、スタンバイ リンクのプライオリティを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 11 **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 12 **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 13 **bundle id** *bundle-id* [**mode** {**active** | **passive** | **on**}] **no shutdown exit**

例：

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

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 TenGigE 0/0/1/0

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 14 **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 15 **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 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 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 18 **show bundle Bundle-Ether** *bundle-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 19 **show lacp Bundle-Ether** *bundle-id*

例：

```
RP/0/RP0/CPU0:router# show lacp 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。

LACP フォールバックの設定

この項では、LACP フォールバック機能の設定方法について説明します。

手順

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)# bundle lacp-fallback timeout 4
```

LACP フォールバック機能を有効にします。

ステップ 4 **end** または **commit**

例：

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

設定変更を保存します。

ステップ 5 **show bundle infrastructure database ma bdl-info Bundle-e1010 | inc***text*

例：

```
RP/0/RP0/CPU0:router# show bundle infrastructure database ma bdl-info Bundle-e1010 | inc
 "fallback"
```

（任意）バンドル マネージャの MA 情報を表示します。

ステップ 6 **show bundle infrastructure database ma bdl-info Bundle-e1015 | inc***text*

例：

```
RP/0/RP0/CPU0:router# show bundle infrastructure database ma bdl-info Bundle-e1015 | inc
 "fallback"
```

（任意）バンドル マネージャの MA 情報を表示します。

MC-LAG での VPWS クロスコネクトの設定

MC-LAG で VPWS クロスコネクトを設定するには、次の作業を実行します。

手順

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 **l2vpn**

例：

```
RP/0/RP0/CPU0:router(config)# l2vpn
```

L2VPN 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 **pw-status**

例：

```
RP/0/RP0/CPU0:router(config-l2vpn)# pw-status
```

疑似回線のステータスをイネーブルにします。

（注）
- 接続回線が冗長状態を Active に変更すると、Active pw-status がプライマリおよびバックアップ疑似回線に送信されます。

  接続回線が冗長状態を Standby に変更すると、Standby pw-status がプライマリおよびバックアップ疑似回線に送信されます。

ステップ 4 **xconnect group** *group-name*

例：

```
RP/0/RP0/CPU0:router(config-l2vpn)# xconnect group grp_1
```

クロスコネクト グループの名前を入力します。

ステップ 5 **p2p** *xconnect-name*

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-xc)# p2p p1
```

ポイントツーポイント クロスコネクトの名前を入力します。

ステップ 6 **interface** *type interface-path-id*

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p)# interface Bundle-Ether 1.1
```

インターフェイス タイプ ID を指定します。

ステップ 7 **neighbor** *A.B.C.D* **pw-id** *pseudowire-id*

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p)# neighbor 10.2.2.2 pw-id 2000
```

クロスコネクトの疑似回線セグメントを設定します。

オプションで、コントロール ワードをディセーブルにするか、イーサネットまたは VLAN に transport-type を設定できます。

ステップ 8 **pw-class** {*class-name*}

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw)# pw-class c1
```

疑似回線に使用する疑似回線クラス テンプレート名を設定します。

ステップ 9 **backup neighbor** *A.B.C.D* **pw-id** *pseudowire-id*

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw)# backup neighbor 10.2.2.2 pw-id 2000
```

バックアップ疑似回線を追加します。

ステップ 10 **pw-class** {*class-name*}

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw-backup)# pw-class c2
```

バックアップ疑似回線に使用する疑似回線クラス テンプレート名を設定します。

ステップ 11 **end** または **commit**

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw-backup)# end
```

または

```
RP/0/RP0/CPU0:router(config-l2vpn-xc-p2p-pw-backup)# 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。

MC-LAG での VPLS の設定

MC-LAG で VPLS を設定するには、次の作業を実行します。

（注） デバイスには最大 128K の MAC アドレス エントリを含めることができます。デバイス上のブリッジ ドメインには最大 64K の MAC アドレス エントリを含めることができます。

手順

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 **l2vpn**

例：

```
RP/0/RP0/CPU0:router(config)# l2vpn
```

L2VPN 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 **pw-status**

例：

```
RP/0/RP0/CPU0:router(config-l2vpn)# pw-status
```

（任意）疑似回線のステータスをイネーブルにします。

接続回線の冗長状態に関係なく、VFI のすべての疑似回線は常にアクティブです。

ステップ 4 **bridge group** *bridge-group-name*

例：

```
RP/0/RP0/CPU0:router(config-l2vpn)# bridge group csco
RP/0/RP0/CPU0:router(config-l2vpn-bg)#
```

ブリッジ ドメインを包含できるようにブリッジ グループを作成し、ブリッジ ドメインにネットワーク インターフェイスを割り当てます。

ステップ 5 **bridge-domain** *bridge-domain-name*

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-bg)# bridge-domain abc
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)#
```

ブリッジ ドメインを確立し、L2VPN ブリッジ グループ ブリッジ ドメイン コンフィギュレーション モードを開始します。

ステップ 6 **interface** *type interface-path-id*

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# interface Bundle-Ether 1.1
```

インターフェイス タイプ ID を指定します。

ステップ 7 **vfi** *{vfi-name}*

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-ac)# vfi vfi-east
```

仮想転送インスタンス（VFI）コンフィギュレーション モードを開始します。

ステップ 8 **neighbor** *A.B.C.D* **pw-id** *pseudowire-id*

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-vfi)# neighbor 10.2.2.2 pw-id 2000
```

クロスコネクトの疑似回線セグメントを設定します。

オプションで、コントロール ワードをディセーブルにするか、イーサネットまたは VLAN に transport-type を設定できます。

ステップ 9 **pw-class** {*class-name*}

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-vfi-pw)# pw-class canada
```

疑似回線に使用する疑似回線クラス テンプレート名を設定します。

ステップ 10 **end** または **commit**

例：

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-vfi-pw)# end
```

または

```
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-vfi-pw)# 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。

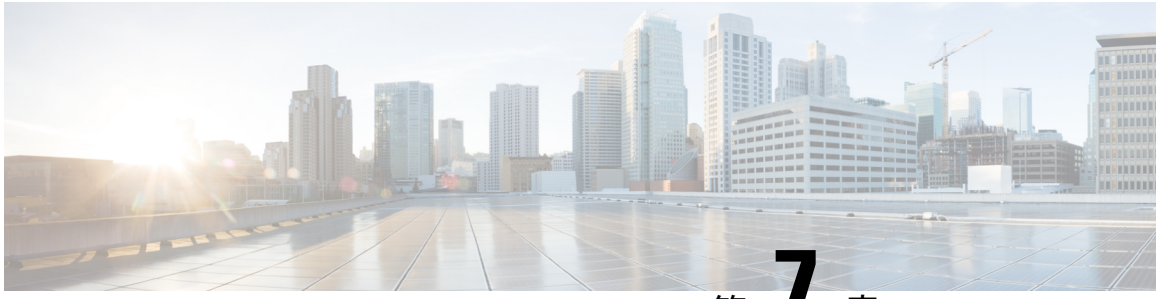
  ```
  Uncommitted changes found, commit them before
  exiting(yes/no/cancel)?
  [cancel]:
  ```

  - **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。



第 7 章

トラフィック ミラーリングの設定

このモジュールでは、トラフィック ミラーリング機能の設定について説明します。トラフィック ミラーリングは、ポート ミラーリング、またはスイッチド ポート アナライザ（SPAN）と呼ばれます。

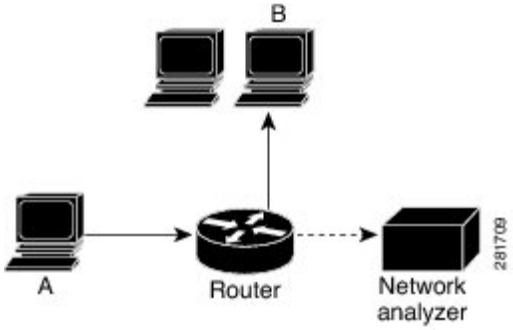
トラフィック ミラーリングの概要

トラフィック ミラーリングは、ポート ミラーリングまたはスイッチド ポート アナライザ（SPAN）と呼ばれることもある、シスコ独自の機能です。この機能を利用すると、一連のポートに入ってくる、または出ていくネットワーク トラフィックをモニタすることができます。このトラフィックを同じルータ上の宛先ポートに渡すことができます。

トラフィック ミラーリングでは、1 つまたは複数の送信元ポートからのトラフィックをコピーし、コピーされたトラフィックを 1 つまたは複数の宛先に送信してネットワーク アナライザまたはその他のモニタリング デバイスに分析させます。トラフィック ミラーリングは、送信元インターフェイスまたはサブインターフェイス上のトラフィックのフローに影響を与えず、ミラーリングされたトラフィックは宛先インターフェイスまたはサブインターフェイスに送信されます。

たとえば、トラフィック アナライザをルータに接続してホスト A によってホスト B に送信されるイーサネット トラフィックをキャプチャできます。

図 *9:* トラフィック ミラーリング動作



ローカル トラフィック ミラーリングが有効になっている場合、ホスト A から送信されるすべてのパケットのコピーを受信するように設定されたポートに、トラフィック アナライザを直接接続します。このポートを「トラフィック ミラーリング ポート」といいます。このマニュアルの他の項で、この機能を調整する方法について説明します。

トラフィック ミラーリングのタイプ

次のタイプのトラフィック ミラーリングがサポートされています。

- **ローカル トラフィック ミラーリング**：最も基本的な形式のトラフィック ミラーリングです。ネットワーク アナライザまたはスニファは宛先インターフェイスに直接接続します。つまり、すべてのモニタ対象ポートが宛先ポートと同じルータ上に存在します。
- **ACL ベースのトラフィック ミラーリング**：トラフィックはインターフェイス ACL の設定に基づいてミラーリングされます。

  インターフェイス アクセス リストの定義に基づいてトラフィックをミラーリングできます。レイヤ 3 トラフィックをミラーリングする際は、**ipv4 access-list** コマンドまたは **ipv6 access-list** コマンド使用し、**capture** オプションを指定して ACL を設定します。**permit** コマンドと **deny** コマンドによって、通常のトラフィックの動作を決定します。**capture** オプションは、パケットが宛先ポートにミラーリングされることを指定します。このオプションは許可タイプのアクセス制御エントリ（ACE）でのみサポートされています。

（注） リリース 6.5.1 より前では、ACL ベースのトラフィック ミラーリングには UDK（ユーザ定義の TCAM キー）と **enable-capture** オプションを使用して **capture** オプションを ACL に設定できるようにする必要がありました。

制約事項

ACL を使用したトラフィックのミラーリングには次の一般的な制約事項が適用されます。

- トラフィック ミラーリング カウンタはサポートされていません。
- ACL ベースのトラフィック ミラーリングはレイヤ 2（イーサネットサービス）ACL ではサポートされていません。
- トラフィックのデフォルトのミラーリングを回避するために、送信元インターフェイス上、または送信元インターフェイスと同じネットワーク処理ユニット上の任意のインターフェイス上に ACL を設定します。バンドル インターフェイスが送信元インターフェイスの場合は、アクティブなすべてのバンドル メンバーと同じネットワーク処理ユニットの任意のインターフェイス上に ACL を設定します。バンドル メンバーは、複数の NPU 上に配置できます。また、設定した ACL が SPAN 設定と同じプロトコル タイプと方向であることを確認します。たとえば、IPv4 または IPv6 の ACL を使用して SPAN を設定する場合は、そのネットワーク処理ユニットに入力 IPv4 ACL または IPv6 ACL をそれぞれ設定します。

次の一般的な制約事項が ERSPAN と SPAN ACL に適用されます。

- ERSPAN トンネルの統計情報はサポートされていません。
- SPAN カウンタはサポートされていせん。
- SPAN 機能と ER-SPAN 機能の両方を同時にルータ上に設定することはできません。SPAN 機能または ERSPAN 機能のいずれかを同じルータ上で設定できます。
- ERSPAN セッション ID の値は常にゼロです。
    - ERPAN を設定するための IOS XR コマンドは使用できません。
- ERSPAN のネクストホップには解決された ARP が必要です。
    - その他のトラフィックまたはプロトコルで ARP をトリガーします。
- ERSPAN は MPLS を介して移動できません。
    - 追加ルータは MPLS でカプセル化される場合があります。
- ERSPAN のカプセル化解除はサポートされていません。
- GRE ネクスト ホップがサブインターフェイスを介して到達可能な場合、ERSPAN は機能しません。ERSPAN が機能するには、メイン インターフェイスを介してネクスト ホップに到達可能である必要があります。
- Rx 方向（入力方向 v4 ACL または v6 ACL）では SPAN-ACL のみがサポートされています。
- SPAN-ACL では、MPLS トラフィックをキャプチャできません。
    - MPLS トラフィックの ACL はサポートされていません。

トラフィック ミラーリングの設定方法

ここでは、トラフィック ミラーリングを設定する方法について説明します。

リモート トラフィック ミラーリングの設定

手順

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 **monitor-session** *session-name*

例：

```
RP/0/RP0/CPU0:router(config)# monitor-session mon1 ethernet
RP/0/RP0/CPU0:router(config-mon)#
```

モニタ セッションを定義し、モニタ セッション 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 **destination interface** *tunnel-ip*

例：

```
RP/0/RP0/CPU0:router(config-mon)# destination interface tunnelip3
```

トラフィックを複製する宛先サブインターフェイスを指定します。

ステップ 4 **exit**

例：

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

モニタ セッション 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 **interface** *type number*

例：

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

指定した送信元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。インターフェイス番号は、*rack*/*slot*/*module*/*port* 表記で入力します。ルータの構文の詳細については、疑問符（?）を使用してオンライン ヘルプを参照してください。

ステップ 6 **monitor-session** *session-name* **ethernet direction rx-onlyport-only**

例：

```
RP/0/RP0/CPU0:router(config-if)# monitor-session mon1 ethernet
direction rx-only port-only
```

このインターフェイスで使用されるモニタ セッションを指定します。**direction** キーワードを使用して、入力または出力のトラフィックのみをミラーリングすることを指定します。

ステップ 7 **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、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。

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 monitor-session [session-name] status [detail] [error]**

例：

```
RP/0/RP0/CPU0:router# show monitor-session
```

トラフィック ミラーリングセッションに関する情報を表示します。

設定可能な送信元インターフェイスの接続

手順

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** *type number*

例：

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

指定した送信元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。インターフェイス番号は、*rack*/*slot*/*module*/*port* 表記で入力します。ルータの構文の詳細については、疑問符（?）を使用してオンライン ヘルプを参照してください。

ステップ 3 **ipv4 access-group** *acl-name* {**ingress** | **egress**}

例：

```
RP/0/RP0/CPU0:router(config-if)# ipv4 access-group acl1 ingress
```

インターフェイスへのアクセスを制御します。

ステップ 4 **monitor-session** *session-name* **ethernet direction rx-onlyport-level acl**

例：

```
RP/0/RP0/CPU0:router(config-if)# monitor-session mon1 ethernet direction rx-only port-level
 acl
RP/0/RP0/CPU0:router(config-if-mon)#
```

送信元インターフェイスにモニタ セッションを付加し、モニタ セッション コンフィギュレーション モードを開始します。

（注） **rx-only** は入力トラフィックのみが複製されることを指定します。

ステップ 5 **acl**

例：

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

定義された ACL に従ってトラフィックをミラーリングすることを指定します。

（注） ACL を名前で設定した場合は、それによってインターフェイス上で設定されている可能性がある ACL がオーバーライドされます。

ステップ 6 **exit**

例：

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

モニタ セッション コンフィギュレーション モードを終了し、インターフェイス コンフィギュレーション モードに戻ります。

ステップ 7 **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、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 monitor-session [session-name] status [detail] [error]**

例：

```
RP/0/RP0/CPU0:router# show monitor-session status
```

モニタ セッションに関する情報を表示します。

トラフィック ミラーリングへの UDF ベースの ACL の設定

手順

	コマンドまたはアクション	目的
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	**udf** *udf-name* **header** {**inner** | **outer**} {**l2** | **l3** | **l4**} **offset** *offset-in-bytes* **length** *length-in-bytes*
	例：
	`RP/0/RP0/CPU0:router(config)# udf udf3 header outer l4 0 length 1 (config-mon)#`
	例：
	`RP/0/RP0/CPU0:router(config)# udf udf3 header inner l4 10 length 2 (config-mon)#`
	例：
	`RP/0/RP0/CPU0:router(config)# udf udf3 header outer l4 50 length 1 (config-mon)#`
		個別の UDF 定義を設定します。UDF の名前、オフセット元のネットワーキング ヘッダー、抽出するデータの長さを指定できます。
		inner キーワードまたは **outer** キーワードは、カプセル化されていないレイヤ 3 またはレイヤ 4 のヘッダーからのオフセットの開始を指定するか、またはカプセル化されたパケットがある場合は内部 L3/L4 からのオフセットの開始を指定します。
		（注） 任意のヘッダーの開始部分から許容される最大オフセットは 63 バイトです。
		length キーワードはオフセットからの長さをバイト単位で指定します。指定できる値の範囲は 1 ～ 4 です。

ステップ 3	**hw-module profile tcam format access-list** {**ipv4** | **ipv6**} [*acl-qualifiers*] [**udf1** *udf-name1 ...* **udf8** *udf-name8*] **enable-capture** 例： `RP/0/RP0/CPU0:router(config)# hw-module profile tcam format access-list ipv4 src-addr dst-addr src-port dst-port proto tcp-flags packet-length frag-bit udf1 udf-test1 udf2 udf-test2 enable-capture`	ハードウェアに送信される ACL キー定義にユーザ定義フィールドを追加します。 （注） 新しい TCAM プロファイルを有効にするには、ライン カードのリロードが必要です。
ステップ 4	**ipv4 access-list** *acl-name* 例：	ACL を作成して、IP ACL 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
	`RP/0/RP0/CPU0:router(config))# ipv4 access-list acl1`	*acl-name* 引数の長さは最大 64 文字です。

ステップ 5	**permit** *regular-ace-match-criteria* **udf** *udf-name1 value1 ... udf-name8 value8*
	例：
	`RP/0/RP0/CPU0:router(config-ipv4-acl)# 10 permit ipv4 any any udf udf1 0x1234 0xffff udf3 0x56 0xff capture`
	`RP/0/RP0/CPU0:router(config-ipv4-acl)# 30 permit ipv4 any any dscp af11 udf udf5 0x22 0x22 capture`
		UDF と一致する ACL を設定します。

ステップ 6	**exit** 例： `RP/0/RP0/CPU0:router(config-ipv4-acl)# exit`	IP ACL 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 7	**interface***type number* 例： `RP/0/RP0/CPU0:router(config)# interface HundredGigE 0/0/1/0`	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 8	**ipv4 access-group** *acl-name* **ingress** 例： `RP/0/RP0/CPU0:router(config-if)# ipv4 access-group acl1 ingress`	アクセス リストをインターフェイスに適用します。
ステップ 9	**commit** 例： `RP/0/RP0/CPU0:router(config-if)# commit`	アクセス リストをインターフェイスに適用します。

トラフィック ミラーリングに関する追加情報

トラフィック ミラーリング用語

- 入力トラフィック：ルータに着信するトラフィック。
- 出力トラフィック：ルータから発信されるトラフィック。

- 送信元（SPAN）インターフェイス：SPAN機能を使用してモニタされているインターフェイス。
- 送信元ポート：トラフィック ミラーリングを使用してモニタされるポート。モニタ対象ポートとも呼ばれます。
- 宛先ポート：送信元ポートをモニタするポート。通常は、このポートにネットワーク アナライザが接続されます。「モニタリング ポート」とも呼ばれます。
- モニタ セッション：SPAN設定の集合に名前を付けたもの。この集合は宛先と送信元のインターフェイスで構成され、宛先は1つ、送信元は1つまたは複数となる可能性があります。

送信元ポートの特性

送信元ポートの特性は、次のとおりです。

-
- 各送信元ポートは、1つのトラフィック ミラーリング セッションでのみモニタできます。
- ポートを送信元ポートとして使用した場合は、同じポートを宛先ポートとしては使用できません。
-

モニタ セッションの特性

モニタ セッションは、1つの宛先インターフェイスと、場合によっては多くの送信元インターフェイスで構成されるトラフィック ミラーリング設定の集まりです。どのモニタ セッションでも、送信元インターフェイス（送信元ポートと呼ばれる）からのトラフィックは、モニタリング ポートまたは宛先ポートに送信されます。1つのモニタリング セッションに複数の送信元ポートがある場合は、多数のミラーリングされたトラフィック ストリームからのトラフィックが宛先ポートにおいて結合されます。その結果、宛先ポートからのトラフィックは、1つまたは複数の送信元ポートからのトラフィックの組み合わせになります。

モニタ セッションには次の特性があります。

- 単一のモニタ セッションの宛先ポートは 1 つだけです。
- 1 つの宛先ポートは 1 つのモニタ セッションだけに属することができます。
-

宛先ポートの特性

各セッションには、送信元ポートからのトラフィックのコピーを受信する宛先ポートが必要です。

宛先ポートの特性は、次のとおりです。

-
-
- いつでも、宛先ポートは 1 つのトラフィック ミラーリング セッションだけに参加できます。1 つのトラフィック ミラーリング セッションの宛先ポートは、別のトラフィック ミラーリングセッションの宛先ポートにできません。つまり、2つのモニタ セッションの宛先ポートが同一であってはなりません。
- 宛先ポートは、送信元ポートにはできません。

上の図のコールアウトは次を示しています。

1. 送信元トラフィック ミラーリング ポート（入力または出力のトラフィック ポート）。
2. 宛先トラフィック ミラーリングポート。

トラフィック ミラーリングの設定例

ここでは、トラフィック ミラーリングを設定する方法の例を示します。

物理インターフェイスを使用したトラフィックミラーリング（ローカル）：例

次に、物理インターフェイスを使用したトラフィック ミラーリングの基本設定の例を示します。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# monitor-session ms1
RP/0/RP0/CPU0:router(config-mon)# destination interface HundredGigE0/0/1/0
RP/0/RP0/CPU0:router(config-mon)# commit

RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface TenGigE0/0/0/0
RP/0/RP0/CPU0:router(config-if)# monitor-session ms1 port-level direction rx-only
RP/0/RP0/CPU0:router(config-if)# commit
```

モニタ セッション ステータスの表示：例

次に、**status** キーワードを指定した **show monitor-session** コマンドの出力例を示します。

```
RP/0/RP0/CPU0:router# show monitor-session status

Monitor-session cisco-rtp1
Destination interface HundredGigE 0/0/1/0
================================================================================
Source Interface   Dir  Status
------------------- ---- --------------------------------------------------
TenGigE0/0/0/4       Both Operational
TenGigE0/0/0/17      Both Operational
```

```
RP/0/RSP0/CPU0:router# show monitor-session status detail

Monitor-session sess1
 Destination interface is not configured
 Source Interfaces
 -----------------
 TenGigE0/0/0/0
  Direction: Both
  ACL match: Disabled
  Portion:  Full packet
  Status:  Not operational (destination interface not known).
 TenGigE0/0/0/1
  Direction: Both
  ACL match: Disabled
  Portion:  First 100 bytes

RP/0/RP0/CPU0:router# show monitor-session status error

Monitor-session ms1
Destination interface TenGigE0/0/0/15 is not configured
================================================================================
Source Interface   Dir  Status
-------------------- ---- ------------------------------------------------------

Monitor-session ms2
Destination interface is not configured
================================================================================
Source Interface   Dir  Status
-------------------- ---- ------------------------------------------------------


RP/0/RP0/CPU0:router# show monitor-session test status

Monitor-session test (ipv4)

Destination Nexthop 255.254.254.4
================================================================================================
Source Interface   Dir         Status
------------------------------------------------------------------------------------------------
Gi0/0/0/2.2     Rx    Not operational (source same as destination)
Gi0/0/0/2.3      Rx    Not operational (Destination not active)
Gi0/0/0/2.4      Rx    Operational
Gi0/0/0/4       Rx    Error: see detailed output for explanation

RP/0/RP0/CPU0:router# show monitor-session test status error

Monitor-session test
Destination Nexthop ipv4 address 255.254.254.4
================================================================
Source Interface     Status
----------------------------------------------------------------
Gi0/0/0/4    < Error: FULL Error Details >
```

トラフィック ミラーリングのトラブルシューティング

トラフィック ミラーリングに問題が発生した場合は、**show monitor-session status** コマンドの出力を確認することからトラブルシューティングを開始します。このコマンドは、すべてのセッションおよび送信元インターフェイスの記録された状態を表示します。

```
# show monitor-session status
```

```
Monitor-session ms1
<session status>
================================================================================


 Interface       Dir   Status
---------------------- ----  ------------------------------------------------------
Gi0/1/0/0.10           Both  <Source interface status>
Gi0/1/0/0.11           Rx    <Source interface status>
Gi0/1/0/0.12           Tx    <Source interface status>
Gi0/2/0/0 (port)       Rx    <Source interface status>
```

上記の例では、<Session status> とマークされた行は、次のいずれかの設定エラーを示している可能性があります。

Session Status	説明
Session is not configured globally	グローバル設定にセッションが存在していません。**show run** コマンドの出力を確認し、セッションが正しい名前で設定されていることを確認します。
Destination interface <intf> (<down-state>)	宛先インターフェイスは、Interface Manager でアップ状態になっていません。**show interfaces** コマンドを使用して状態を確認できます。設定を調べて、インターフェイスがアップ状態にならない原因を特定します（たとえば、サブインターフェイスが適切なカプセル化の設定を必要としています）。

<Source interface status> は次のメッセージを報告できます。

Source Interface Status	説明
Operational	トラフィック ミラーリング PI において、すべてのものが正しく動作しているようです。ミラーリングが期待どおりに動作しない場合は、まずプラットフォーム チームと協力して調査します。
Not operational (Session is not configured globally)	グローバル設定にセッションが存在していません。**show run** コマンドの出力を確認し、セッションが正しい名前で設定されていることを確認します。

Source Interface Status	説明
Not operational (destination not known)	セッションは存在していますが、宛先インターフェイスが設定されていないか、そのセッションに指定されている宛先インターフェイスが存在していません（たとえば、宛先がまだ作成されていないサブインターフェイスであるなど）。
Not operational (source same as destination)	セッションは存在していますが、宛先と送信元が同じインターフェイスであるため、トラフィック ミラーリングは機能しません。
Not operational (destination not active)	宛先インターフェイスまたは疑似配線がアップ状態ではありません。対応する *Session status* のエラーメッセージで、提案されている解決方法を確認します。
Not operational (source state <down-state>)	送信元インターフェイスはアップ状態ではありません。**show interfaces** コマンドを使用して状態を確認できます。設定を調べて、インターフェイスがアップ状態にならない原因を特定します（たとえば、サブインターフェイスが適切なカプセル化の設定を必要としています）。
Error: see detailed output for explanation	トラフィック ミラーリングでエラーが発生しました。**show monitor-session status detail** コマンドを実行して詳細情報を表示します。

show monitor-session status detail コマンドは、設定パラメータの詳細情報と、検出されたエラー（ある場合）を表示します。次に例を示します。

RP/0/RP0/CPU0:router **show monitor-session status detail**

```
Monitor-session sess1
 Destination interface is not configured
 Source Interfaces
 -----------------
 TenGigE0/0/0/1
  Direction: Both
  ACL match: Disabled
  Portion:  Full packet
  Status:  Not operational (destination interface not known)
 TenGigE0/0/0/2
  Direction: Both
  ACL match: Disabled
  Portion:  First 100 bytes
  Status: Not operational (destination interface not known). Error: 'Viking SPAN PD'
detected the 'warning' condition 'PRM connection
          creation failure'.
Monitor-session foo
```

```
 Destination next-hop TenGigE 0/0/0/0
 Source Interfaces
 -----------------
 TenGigE 0/0/0/1.100:
  Direction: Both
  Status:  Operating
 TenGigE 0/0/0/2.200:
  Direction: Tx
  Status:  Error: <blah>

Monitor session bar
 No destination configured
 Source Interfaces
 -----------------
 TenGigE 0/0/0/3.100:
  Direction: Rx
  Status:  Not operational(no destination)
```

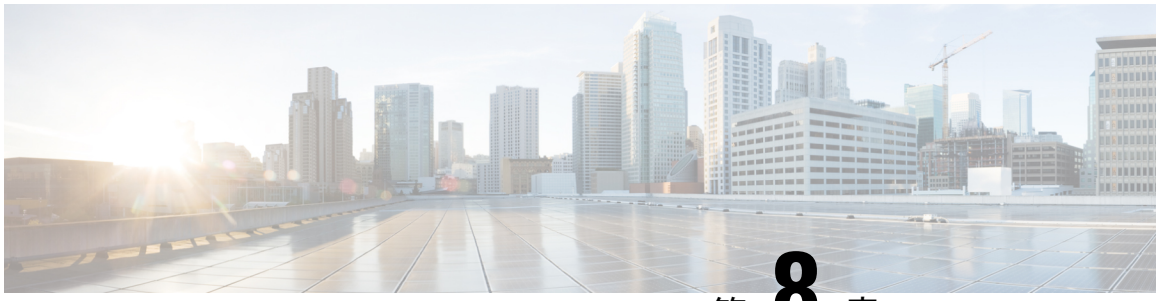
次に追加のトレースとデバッグのコマンドを示します。

```
RP/0/RP0/CPU0:router# show monitor-session platform trace ?

 all   Turn on all the trace
 errors Display errors
 events Display interesting events

RP/0/RP0/CPU0:router# show monitor-session trace ?

 process Filter debug by process

RP/0/RP0/CPU0:router# debug monitor-session platform ?

 all   Turn on all the debugs
 errors VKG SPAN EA errors
 event  VKG SPAN EA event
 info  VKG SPAN EA info

RP/0/RP0/CPU0:router# debug monitor-session process all

RP/0/RP0/CPU0:router# debug monitor-session process ea

RP/0/RP0/CPU0:router# debug monitor-session process ma

RP/0/RP0/CPU0:router# show monitor-session process mgr

 detail  Display detailed output
 errors  Display only attachments which have errors
 internal Display internal monitor-session information
 |     Output Modifiers

RP/0/RP0/CPU0:router# show monitor-session status

RP/0/RP0/CPU0:router# show monitor-session status errors

RP/0/RP0/CPU0:router# show monitor-session status internal
```

UDF ベースの ACL の確認

show monitor-session status detail コマンドを使用して、ACL の UDF の設定を確認します。

```
RP/0/RP0/CPU0:leaf1# show monitor-session 1 status detail

Fri May 12 19:40:39.429 UTC
Monitor-session 1
  Destination interface tunnel-ip3
  Source Interfaces
  -----------------
  TenGigE0/0/0/15
    Direction:  Rx-only
    Port level: True
    ACL match:  Enabled
    Portion:    Full packet
    Interval:   Mirror all packets
    Status:     Not operational (destination not active)
```



第 8 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 （129 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。

制約事項

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

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 loopback** *instance*

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 address** *ip-address*

例：

```
RP/0/RP0/CPU0:router(config-if)# ipv4 address 100.100.100.69 255.255.255.255
```

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 アドレスとサブネット 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 5 **show interfaces***type 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。

例

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 100.100.100.69 255.255.255.255
RP/0/RP0/CPU0:router(config-if)# ipv6 address 100::69/128
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 unreachables** コマンドのみです。 **ipv4 unreachables** 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 unreachables** 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Null0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 unreachables** 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方法について説明します。

手順

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。

- **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 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。

例

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 icmp unreachables disable
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

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 virtual address** *ipv4-*

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。

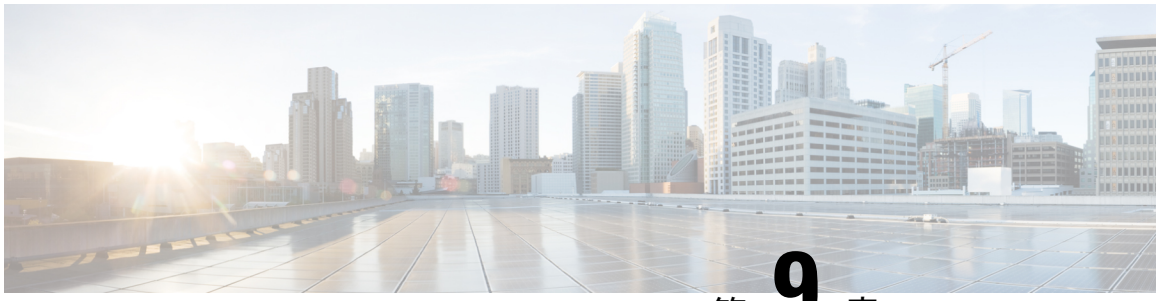
```
Uncommitted changes found, commit them before
exiting(yes/no/cancel)?
[cancel]:
```

- **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。

例

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 9 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。VLAN の範囲は 1 ～ 4094 です。

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 （138 ページ）
- 802.1Q VLAN インターフェイスの設定に関する情報 （143 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

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 {TenGigE | FortyGigE | HundredGigE | Bundle-Ether}** *interface-path-id.subinterface*

例：

```
RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/4.10
```

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 ～ 2147483647 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
```

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-ether** *instance*

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。

確認

次に、イーサネット インターフェイスの設定を確認する例を示します。

```
# show ethernet trunk be 1020 Wed May 17 16:43:32.804 EDT

Trunk                                  Sub types         Sub states
Interface      St Ly    MTU    Subs       L2      L3       Up     Down   Ad-Down
BE1020       Up L3   9100       3        3       0        3       0        0

Summary                           3        3       0        3       0        0
```

VLAN での接続回線の設定

VLAN で接続回線を設定するには、次の手順で操作します。

手順

ステップ 1 **configure**

例：

```
RP/0//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 [GigabitEthernet | TenGigE | Bundle-Ether | FortyGigE]** *interface-path*] *id.subinterface* **l2transport**

例：

```
RP/0//CPU0:router(config)# interface TenGigE 0/0/0/1.1 l2transport
```

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。
- コマンド文字列に **l2transport** キーワードを含める必要があります。そうしないと、AC ではなく、レイヤ 3 サブインターフェイスが作成されます。

ステップ 3 **encapsulation dot1q 100**

例：

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

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、**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 4 **end** または **commit**

例：

```
RP/0//CPU0:router(config-if-l2)# end
```

または

```
RP/0//CPU0:router(config-if-l2)# 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 5 **show interfaces** [**GigabitEthernet** |**FortyGigE**|**Bundle-Ether** | **TenGigE**] *interface-path-id.subinterface*

例：

```
RP/0//CPU0:router# show interfaces TenGigE 0/0/0/3.1
```

（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。

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

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** {**TenGigE** | **FortyGigE** | **HundredGigE** | **Bundle-Ether**] *interface-path-id.subinterface*

例：

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

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* 引数を次のインスタンスのいずれかで置き換えます。

- 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 ～ 2147483647 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、物理またはバンドル 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 0/0/0/0 上のイーサネットサブインターフェイス 23 は、TenGigE 0/0/0/0.23 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に適用可能なスケール値は次のとおりです。

- システムあたりのサブインターフェイス = 1024
- ライン カードあたりのサブインターフェイス = 1024
- NPU あたりのサブインターフェイス = 1024
- インターフェイスあたりのサブインターフェイス = 512
- コアあたりのサブインターフェイス = 512

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。MTU が設定されていない場合、デフォルトのサブインターフェイスは物理インターフェイスの MTU を継承します。サブインターフェイスにはNPU あたり最大 3 つの異なる MTU を使用できます。イーサネットMTUおよびイーサネットインタフェー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**

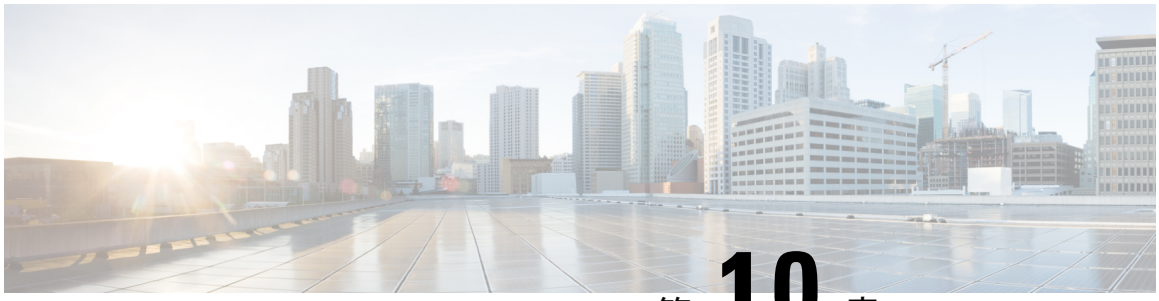
コマンド文字列に**l2transport**キーワードを含めて、そのインターフェイスがレイヤ2インターフェイスであることを指定する必要があります。

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。



第 10 章

GRE トンネルの設定

Generic Routing Encapsulation（GRE）は、カプセル化によってあるプロトコルのパケットを別のプロトコルを介して転送する、簡易な一般的アプローチを提供するトンネリング プロトコルです。このモジュールでは、GRE トンネルの設定方法について説明します。

GRE トンネルの設定

トンネリングでは、1 つのプロトコルのパケットを別のプロトコル内に転送するメカニズムを提供します。Generic Routing Encapsulation（GRE）は、カプセル化によってあるプロトコルのパケットを別のプロトコルを介して転送する、簡易な一般的アプローチを提供するトンネリング プロトコルです。GRE は、ペイロード（外側の IP パケット内部の、宛先ネットワークに渡す必要がある内側のパケット）をカプセル化します。GRE トンネルは、トンネル送信元アドレスとトンネル宛先アドレスによって識別される 2 つのエンドポイントを持つ仮想ポイントツーポイント リンクとして動作します。トンネルのエンドポイントは、介在する IP ネットワークを通じてカプセル化パケットをルーティングすることによって、GRE トンネルを介してペイロードを送信します。途中の IP ルータは、ペイロード（内側のパケット）を解析しません。これらのルータは、GRE トンネル エンドポイントにパケットを転送する際に、外側の IP パケットだけを解析します。トンネル エンドポイントに到達すると、GRE カプセル化が削除され、ペイロードはパケットの最終的な宛先に転送されます。

外部パケットによるカプセル化はトンネルの送信元で行われますが、外部パケットのカプセル化解除はトンネルの宛先で行われます。カプセル化およびカプセル化解除データは定期的に、またはオンデマンドで収集されます。カプセル化統計情報により、トンネルの送信元でカプセル化されたパケットの数が示されます。カプセル化解除統計情報により、トンネルの宛先でカプセル化解除されたパケットの数が示されます。このデータは、ルート プロセッサの統計情報タイプに基づく論理テーブルに統計情報として保存されます。L2 インターフェイス TX 統計情報、L3 インターフェイス TX 統計情報、トラップ統計情報など、さまざまな統計情報タイプがあります。カプセル化の統計情報は、トラフィックの送信元を推測するのに役立ち、カプセル化解除の統計情報にはトラフィックの宛先が示されます。また、カプセル化解除の統計情報はトラフィックのタイプを検出するのにも役立ちます。

GRE トンネル設定の制約事項

GRE トンネル設定時には、次の制約事項が適用されます。

- NCS540 シリーズ ルータは最大 500 の GRE トンネルをサポートしています。
- トンネル送信元でサポートされている一意の送信元 IP アドレスは最大 16 個のみです。

設定例

GRE トンネルを設定するには、トンネル インターフェイスを作成し、トンネルの送信元と宛先を定義します。次に、ルータ 1 とルータ 2 の間に GRE トンネルを設定する例を示します。両方のルータ上でトンネルインターフェイスを設定する必要があります。ルータ 1 のトンネル送信元 IP アドレスは、ルータ 2 のトンネル宛先 IP アドレスとして設定されます。ルータ 1 のトンネル宛先 IP アドレスは、ルータ 2 のトンネル送信元 IP アドレスとして設定されます。次の例では、2 つのルータ間のルーティング プロトコルとして OSPF が使用されています。また、BGP または IS-IS もルーティング プロトコルとして使用できます。

```
RP/0/RP0/CPU0:Router1# configure
RP/0/RP0/CPU0:Router1(config)#  interface tunnel-ip 30
RP/0/RP0/CPU0:Router1(config-if)# tunnel mode gre ipv4
RP/0/RP0/CPU0:Router(config-if)# ipv4 address 10.1.1.1 255.255.255.0
RP/0/RP0/CPU0:Router1(config-if)# tunnel source 192.168.1.1
RP/0/RP0/CPU0:Router1(config-if)# tunnel destination 192.168.2.1
RP/0/RP0/CPU0:Router1(config-if)# exit
RP/0/RP0/CPU0:Router1(config)# interface Loopback 0
RP/0/RP0/CPU0:Router1(config-if)# ipv4 address 1.1.1.1
RP/0/RP0/CPU0:Router1(config-if)# exit
RP/0/RP0/CPU0:Router1(config)# router ospf 1
RP/0/RP0/CPU0:Router1(config-ospf)# router-id 192.168.4.1
RP/0/RP0/CPU0:Router1(config-ospf)# area 0
RP/0/RP0/CPU0:Router1(config-ospf-ar)# interface tunnel-ip 30
RP/0/RP0/CPU0:Router1(config-ospf-ar)#  interface Loopback 0
RP/0/RP0/CPU0:Router1(config-ospf-ar)# commit

RP/0/RP0/CPU0:Router2# configure
RP/0/RP0/CPU0:Router2(config)#  interface tunnel-ip 30
RP/0/RP0/CPU0:Router2(config-if)# tunnel mode gre ipv4
RP/0/RP0/CPU0:Router2(config-if)# ipv4 address 10.1.1.2 255.255.255.0
RP/0/RP0/CPU0:Router2(config-if)# tunnel source 192.168.2.1
RP/0/RP0/CPU0:Router2(config-if)# tunnel destination 192.168.1.1
RP/0/RP0/CPU0:Router2(config-if)# exit
RP/0/RP0/CPU0:Router2(config)# interface Loopback 0
RP/0/RP0/CPU0:Router2(config-if)# ipv4 address 2.2.2.2
RP/0/RP0/CPU0:Router2(config)# router ospf 1
RP/0/RP0/CPU0:Router2(config-ospf)# router-id 192.168.3.1
RP/0/RP0/CPU0:Router2(config-ospf)# area 0
RP/0/RP0/CPU0:Router2(config-ospf-ar)# interface tunnel-ip 30
RP/0/RP0/CPU0:Router2(config-ospf-ar)# interface Loopback 0
RP/0/RP0/CPU0:Router2(config-if)# commit
```

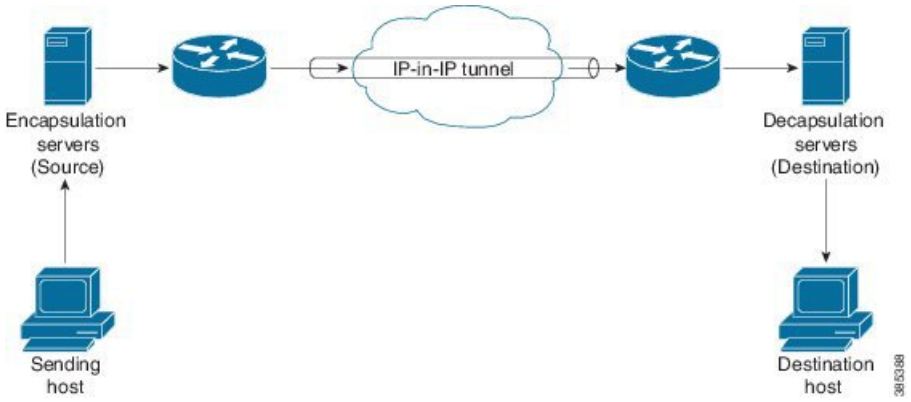
IP-in-IP カプセル化解除

ネットワーク内のデータグラムのカプセル化は、宛先ホストに到達するためにパケットがたどるルートに送信元サーバで影響を与える場合など、いくつかの理由で実行されます。また、送信元サーバはカプセル化サーバとも呼ばれます。

IP-in-IP カプセル化には、既存の IP ヘッダーを介した外部 IP ヘッダーの挿入が含まれています。外部 IP ヘッダー内の送信元と宛先のアドレスは、IP-in-IP トンネルのエンドポイントを指しています。パケットを転送するルータのループバック アドレスをネットワーク管理者が把握している場合は、IPヘッダーのスタックを使用して、パケットを事前に決定さしたパスを介して宛先に送信します。このトンネリング メカニズムは、ほとんどのネットワーク アーキテクチャの可用性と遅延の判断に使用できます。送信元から宛先までのパス全体をヘッダーに含める必要はありませんが、パケットを送信するためのネットワークのセグメントは選択できることに注意してください。

次に、基本的な IP-in-IP カプセル化とカプセル化解除のモデルを説明する図を示します。

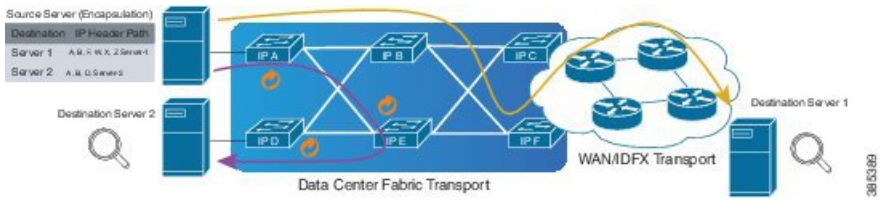
図 *10 : IP-in-IP* トンネルを使用した基本的なカプセル化とカプセル化解除



使用例：IP-in-IP カプセル化解除の設定

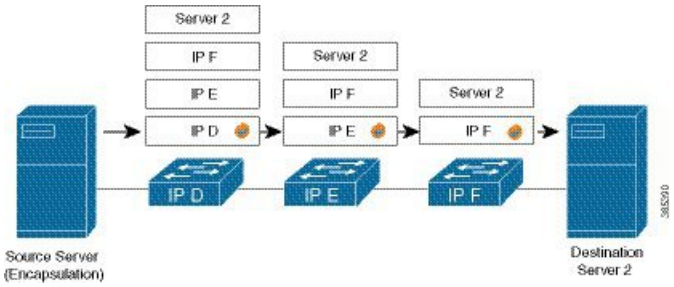
次のトポロジで、送信元から宛先までのネットワークのさまざまなセグメントに IP-in-IP カプセル化とカプセル化解除が使用されている使用例について説明します。IP-in-IP トンネルは、データセンター ファブリック ネットワークを通じてパケットのカプセル化を解除し、送信するために使用する複数のルートで構成されます。

図 *11 :* データセンター ネットワークを通じた *IP-in-IP* カプセル化解除



次の図に、カプセル化解除ルータを通過するときにスタック構成の IPv4 ヘッダーのカプセル化を解除する方法を示します。

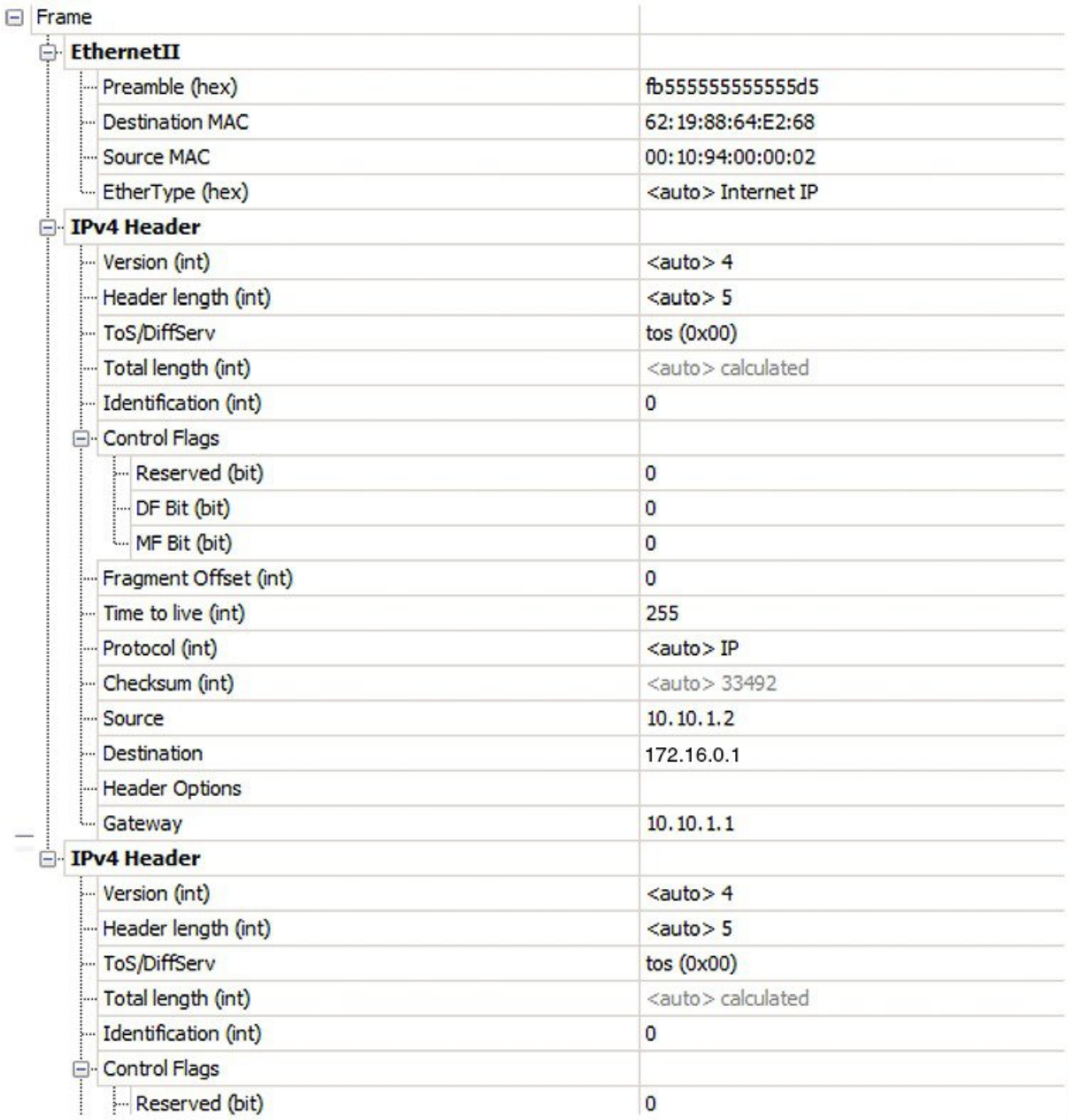
図 *12 : IP* ヘッダーのカプセル化解除



カプセル化されたパケット内のスタック構成の IP ヘッダー

カプセル化されたパケットには、元の IPv4 ヘッダー上に外部 IPv4 ヘッダーが積み重ねられています。

カプセル化されたパケット



Frame	
EthernetII	
Preamble (hex)	fb555555555555d5
Destination MAC	62:19:88:64:E2:68
Source MAC	00:10:94:00:00:02
EtherType (hex)	<auto> Internet IP
IPv4 Header	
Version (int)	<auto> 4
Header length (int)	<auto> 5
ToS/DiffServ	tos (0x00)
Total length (int)	<auto> calculated
Identification (int)	0
Control Flags	
Reserved (bit)	0
DF Bit (bit)	0
MF Bit (bit)	0
Fragment Offset (int)	0
Time to live (int)	255
Protocol (int)	<auto> IP
Checksum (int)	<auto> 33492
Source	10.10.1.2
Destination	172.16.0.1
Header Options	
Gateway	10.10.1.1
IPv4 Header	
Version (int)	<auto> 4
Header length (int)	<auto> 5
ToS/DiffServ	tos (0x00)
Total length (int)	<auto> calculated
Identification (int)	0
Control Flags	
Reserved (bit)	0

設定

IP-in-IP トンネルの通過時にパケットのカプセル化を解除するには、次の設定例をルータに使用します。

```
RP/0/RP0/CPU0:router(config)# interface tunnel-ip 10
RP/0/RP0/CPU0:router(config-if)# tunnel mode ipv4 decap
RP/0/RP0/CPU0:router(config-if)# tunnel source loopback 0
RP/0/RP0/CPU0:router(config-if)# tunnel destination 10.10.1.2/32
```

- **tunnel-ip**：IP-in-IP トンネル インターフェイスを設定します。
- **ipv4 unnumbered loopback address**：ループバック アドレスの場合を除き、明示的なアドレスを使用せずに IPv4 パケット処理を可能にします。
- **tunnel mode ipv4 decap**：IP-in-IP カプセル化解除を有効にします。
- **tunnel source**：ルータ インターフェイスに関して、IP-in-IP カプセル化解除トンネルの送信元インターフェイスを指定します。
- **tunnel destination**：ルータ インターフェイスに関して、IP-in-IP カプセル化解除トンネルの宛先アドレスを指定します。

実行コンフィギュレーション

```
RP/0/RP0/CPU0:router# show running-config interface tunnel-ip 10
...
interface tunnel-ip 10
tunnel mode ipv4 decap
tunnel source Loopback 0
tunnel destination 10.10.1.2/32
```

これにより、IP-in-IP カプセル化解除の設定が完了します。

ライン レートのカプセル化を許可する単一パス GRE のカプセル化

単一パス GRE カプセル化を許可するライン レート カプセル化機能（ロードバランシング機能のプレフィックスベースの GRE トンネルの宛先ともいう）により、ライン レート GRE カプセル化トラフィックを有効にし、フロー エントロピーを有効にします。データプレーン転送パフォーマンスはライン レート全体をサポートし、追加されたカプセル化を考慮するように調整されます。RIB で宛先が使用できない場合は、GRE トンネルがダウンします。リリース 6.3.2 では GRE 単一パス トンネルを介したルーティングがサポートされていません。そのため、GRE カプセル化の対象となるトラフィックは GRE カプセル化に基づく ACL フィルタを使用して識別されます。GRE トンネルの宛先アドレスはエニーキャスト アドレスです。すべての GRE カプセル化を ACL または ポリシーマップのいずれか、あるいはその両方に基づいて割り当てる必要があります。宛先には個別のアドレスか、または /28 プレフィックスも使用できます。

設定

GRE 単一パス エントロピー機能を設定するには、次のタスクを実行します。

- GRE 単一パス
- GRE のエントロピー（ECMP/UCMP）

```
/* GRE Single-Pass */
```

```
Router# configure
Router(config)# interface tunnel-ip30016
Router(config-if)# ipv4 address 216.1.1.1 255.255.255.0
Router(config-if)# ipv6 address 216:1:1::1/64
Router(config-if)# ipv6 enable
Router(config-if)# tunnel mode gre ipv4 encap
Router(config-if)# tunnel source Loopback22
Router(config-if)# tunnel destination 170.170.170.22
Router(config-if)# commit
Router(config-if)# exit

/* GRE Entropy(ECMP/UCMP)*/

ECMP (ISIS)

Router# configure
Router(config)# router isis core
Router(config)# apply-group ISIS-INTERFACE
Router(config-isis)# is-type level-2-only
Router(config-isis)# net 49.1111.0000.0000.002.00
Router(config-isis)# nsr
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide metric 2
Router(config-isis-af)# mpls traffic-eng level-2-only
Router(config-isis-af)# mpls traffic-eng router-id Loopback0
Router(config-isis-af)# maximum-paths 5
Router(config-isis-af)# commit
!

/* UCMP(ISIS) */

Router# configure
Router(config)# router isis core
Router(config)# apply-group ISIS-INTERFACE
Router(config-isis)# is-type level-2-only
Router(config-isis)# net 49.1111.0000.0000.002.00
Router(config-isis)# nsr
Router(config-isis)# log adjacency changes
Router(config-isis)# address-family ipv4 unicast
Router(config-isis-af)# metric-style wide ucmp metric 2
Router(config-isis-af)# mpls traffic-eng level-2-only
Router(config-isis-af)# mpls traffic-eng router-id Loopback0
Router(config-isis-af)# maximum-paths 5
Router(config-isis-af)# redistribute connected
Router(config-isis-af)# commit
Router(config-isis-af)# exit
!

Router# configure
Router(config)# interface Bundle-Ether3
Router(config-if)# apply-group ISIS-INTERFACE
Router(config-if)# address-family ipv4 unicast
Router(config-af)# metric 20
Router(config-af)# commit
Router(config-af)# exit
!

Router# configure
Router(config)# interface Bundle-Ether111
Router(config-if)# apply-group ISIS-INTERFACE
Router(config-if)# address-family ipv4 unicast
Router(config-af)# metric 15
```

```
Router(config-af)# commit
Router(config-af)# exit
!

/* ECMP(OSPF) */

Router# configure
Router(config)# router ospf 3
Router(config-ospf)# nsr
Router(config-ospf)# maximum paths 5
Router(config-ospf)# address-family ipv4 unicast
Router(config-ospf-af)# area 0
Router(config-ospf-af-ar)# interface Bundle-Ether3
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether4
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether111
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether112
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Loopback23
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface HundredGigE 0/9/0/0
Router(config-ospf-af-ar-if)# commit
Router(config-ospf-af-ar-if)# exit


/* UCMP(OSPF) */

Router# configure
Router(config)# router ospf 3
Router(config-ospf)# nsr
Router(config-ospf)# maximum paths 5
Router(config-ospf)# ucmp
Router(config-ospf)# address-family ipv4 unicast
Router(config-ospf-af)# area 0
Router(config-ospf-af-ar)# interface Bundle-Ether3 cost 2
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether4
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether111
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Bundle-Ether112 cost 2
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface Loopback23
Router(config-ospf-af-ar-if)# exit
!
Router(config-ospf-af-ar)# interface HundredGigE 0/9/0/0
Router(config-ospf-af-ar-if)# commit
Router(config-ospf-af-ar-if)# exit


/* ECMP(BGP) */
Router# configure
```

```
Router(config)# router bgp 800
Router(config-bgp)# bgp bestpath as-path multipath-relax
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# network 170.170.170.3/32
Router(config-bgp-af)# network 170.170.170.10/32
Router(config-bgp-af)# network 170.170.170.11/32
Router(config-bgp-af)# network 170.170.172.3/32
Router(config-bgp-af)# network 180.180.180.9/32
Router(config-bgp-af)# network 180.180.180.20/32
Router(config-bgp-af)# network 180.180.180.21/32
Router(config-bgp-af)# network 180.180.180.24/32
Router(config-bgp-af)# network 180.180.180.25/32
Router(config-bgp-af)# commit
!
Router# configure
Router(config)# router bgp 800
Router(config-bgp)# neighbor 4.1.1.2
Router(config-bgp-nbr)# remote-as 300
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy pass-all in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# commit
!

/* UCMP(BGP) */

Router# configure
Router(config)# router bgp 800
Router(config-bgp)# bgp bestpath as-path multipath-relax
Router(config-bgp)# address-family ipv4 unicast
Router(config-bgp-af)# maximum-paths ebgp 5
Router(config-bgp-af)# network 180.180.180.9/32
Router(config-bgp-af)# network 180.180.180.20/32
Router(config-bgp-af)# network 180.180.180.21/32
Router(config-bgp-af)# network 180.180.180.24/32
Router(config-bgp-af)# network 180.180.180.25/32
Router(config-bgp-af)# commit
!
Router# configure
Router(config)# router bgp 800
Router(config-bgp)# neighbor 7.1.5.2
Router(config-bgp-nbr)# remote-as 4000
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy TRANSIT0_IN in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# next-hop-self
Router(config-bgp-nbr-af)# commit
!
Router# configure
Router(config)# router bgp 800
Router(config-bgp)# 4.1.111.2
Router(config-bgp-nbr)# remote-as 4000
Router(config-bgp-nbr)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# address-family ipv4 unicast
Router(config-bgp-nbr-af)# route-policy TRANSIT0_IN in
Router(config-bgp-nbr-af)# route-policy pass-all out
Router(config-bgp-nbr-af)# next-hop-self
Router(config-bgp-nbr-af)# commit
!
```

```
/* Configure roupte policy */

Router# configure
Router(config)# route-policy TRANSIT0_IN
Router(config-rpl)# if destination in (170.170.170.24/32) then
Router(config-rpl-if)# set extcommunity bandwidth (2906:1250000)
Router(config-rpl-if)# else
Router(config-rpl-else)# pass
Router(config-rpl-else)# endif
Router(config-rpl)# end-policy
!

Router# configure
Router(config)# route-policy TRANSIT1_IN
Router(config-rpl)# if destination in (170.170.170.24/32) then
Router(config-rpl-if)# set extcommunity bandwidth (2906:37500000
Router(config-rpl-if)# else
Router(config-rpl-else)# pass
Router(config-rpl-else)# endif
Router(config-rpl)# end-policy
```

実行コンフィギュレーション

```
/* GRE Single-Pass configuration */

interface tunnel-ip30016
ipv4 address 216.1.1.1 255.255.255.0
ipv6 address 216:1:1::1/64
ipv6 enable
tunnel mode gre ipv4 encap
tunnel source Loopback22
tunnel destination 170.170.170.22
!

/* GRE Entropy(ECMP/UCMP) */

ECMP (ISIS)

router isis core
apply-group ISIS-INTERFACE
is-type level-2-only
net 49.1111.0000.0000.002.00
nsr
log adjacency changes
address-family ipv4 unicast
metric-style wide
metric 2
mpls traffic-eng level-2-only
mpls traffic-eng router-id Loopback0
maximum-paths 5
!

/* UCMP(ISIS) */

router isis core
apply-group ISIS-INTERFACE
is-type level-2-only
net 49.1111.0000.0000.002.00
nsr
log adjacency changes
```

```
address-family ipv4 unicast
metric-style wide
ucmp
metric 2
mpls traffic-eng level-2-only
mpls traffic-eng router-id Loopback0
maximum-paths 5
redistribute connected
!
interface Bundle-Ether3
apply-group ISIS-INTERFACE
address-family ipv4 unicast
metric 20
!

interface Bundle-Ether111
apply-group ISIS-INTERFACE
address-family ipv4 unicast
metric 15
!

!

/* ECMP(OSPF) */

router ospf 3
nsr
maximum paths 5
address-family ipv4 unicast
area 0
interface Bundle-Ether3
!
interface Bundle-Ether4
!
interface Bundle-Ether111
!
interface Bundle-Ether112
!
interface Loopback23
!
interface hundredGigE0/9/0/0
!
!
!
/* UCMP (OSPF) */

router ospf 3
nsr
maximum paths 5
ucmp
address-family ipv4 unicast
area 0
interface Bundle-Ether3
cost 2
!
interface Bundle-Ether4
!
interface Bundle-Ether111
!
interface Bundle-Ether112
cost 2
!
interface Loopback23
!
```

```
interface hundredGigE0/9/0/0
!
!
!

/* ECMP(BGP)*/

router bgp 800
bgp bestpath as-path multipath-relax
address-family ipv4 unicast
maximum-paths ebgp 5
network 170.170.170.3/32
network 170.170.170.10/32
network 170.170.170.11/32
network 170.170.172.3/32
network 180.180.180.9/32
network 180.180.180.20/32
network 180.180.180.21/32
network 180.180.180.24/32
network 180.180.180.25/32
!
neighbor 4.1.1.2
remote-as 300
address-family ipv4 unicast
route-policy PASS-ALL in
route-policy PASS-ALL out
next-hop-self
!
!

/* UCMP(BGP) */

router bgp 800
bgp bestpath as-path multipath-relax
address-family ipv4 unicast
maximum-paths ebgp 5
network 180.180.180.9/32
network 180.180.180.20/32
network 180.180.180.21/32
network 180.180.180.24/32
network 180.180.180.25/32
!

neighbor 7.1.5.2
remote-as 4000
address-family ipv4 unicast
route-policy TRANSIT0_IN in
route-policy PASS-ALL out
next-hop-self
!
!
neighbor 4.1.111.2
remote-as 4000
address-family ipv4 unicast
route-policy TRANSIT1_IN in
route-policy PASS-ALL out
next-hop-self
!
!

/* Configure roupte policy */

route-policy TRANSIT0_IN
if destination in (170.170.170.24/32) then
```

```
set extcommunity bandwidth (2906:1250000)
else
pass
endif
end-policy
!
route-policy TRANSIT1_IN
if destination in (170.170.170.24/32) then
set extcommunity bandwidth (2906:37500000)
else
pass
endif
end-policy
!
```

確認

トンネル モードの GRE カプセル化が有効になっていることを確認します。

```
Router# show interfaces tunnel-ip 100

Sun Jul 10 15:49:04.812 VN_TIME

tunnel-ip100 is up, line protocol is up

  Interface state transitions: 2

  Hardware is Tunnel

  Internet address is Unknown

  MTU 1500 bytes, BW 100 Kbit (Max: 100 Kbit)

     reliability 255/255, txload 0/255, rxload 0/255

  Encapsulation TUNNEL_GRE,  loopback not set,

  Tunnel TOS 0

  Tunnel mode GRE IPV4,

  Keepalive is enabled, interval 10 seconds, maximum retry 3

  Tunnel source 172.16.16.1 (GigabitEthernet0_0_0_0), destination 172.16.16.2

  Tunnel TTL 100

  Last input 2d03h, output 2d04h

  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

     689 packets input, 26212 bytes, 0 total input drops

     0 drops for unrecognized upper-level protocol

     Received 0 broadcast packets, 0 multicast packets

     3 packets output, 192 bytes, 0 total output drops
```

```
    Output 0 broadcast packets, 0 multicast packets
```

トンネル モードの GRE カプセル化とカプセル化解除が有効になっていることを確認します。

```
Router# sh interfaces tunnel-ip 5 accounting
Wed May 16 01:50:57.258 UTC
tunnel-ip5
  Protocol              Pkts In         Chars In     Pkts Out        Chars Out
  IPV4_UNICAST              489            55746            0                0
  IPV6_UNICAST              489            55746            0                0
  MPLS                      587            69266            0                0
```

パケットの再循環が Recycle VoQ: 48 で実行されないことを確認します。

```
 Router# show tunnel ip ea summary location 0/RP0/CPU0

Number of tunnel updates to retry: 0
Number of tunnel updates retried: 0
Number of tunnel retries failed: 0
Platform:
Recycle VoQ: 48
          ReceivedBytes   ReceivedPackets    ReceivedKbps
          DroppedBytes    DroppedPackets     DroppedKbps

NPU 0:0      0               0                  0
             0               0                  0
1            0               0                  0
             0               0                  0
2            0               0                  0
             0               0                  0
3            0               0                  0
             0               0                  0
...
NPU 1:0      0               0                  0
             0               0                  0
1            0               0                  0
             0               0                  0
2            0               0                  0
             0               0                  0
3            0               0                  0
             0               0                  0

NPU 2:0      0               0                  0
             0               0                  0
1            0               0                  0
             0               0                  0
2            0               0                  0
             0               0                  0
3            0               0                  0
             0               0                  0
```

トンネル モードの GRE カプセル化が有効になっていることを確認します。

```
Router# show interfaces tunnel-ip * brief

Thu Sep 7 00:04:39.125 PDT
Intf Intf LineP Encap MTU BW
Name     State    State   Type       (byte) (Kbps)
--------------------------------------------------------------------------------
ti30001  down     down    TUNNEL_IP   1500   100
ti30002  up       up      TUNNEL_IP   1500   100
```

RIB のトンネル エンドポイント ルートを確認します。

```
Router# show route 10.1.1.1

Routing entry for 10.0.0.0/8
Known via "static", distance 1, metric 0 (connected)
Installed Oct 2 15:50:56.755 for 00:39:24
Routing Descriptor Blocks
directly connected, via tunnel-ip109
Route metric is 0, Wt is 1
No advertising protos.
```

トンネル モードの GRE カプセル化が有効になっていることを確認します。

```
Router# show tunnel ip ea database tunnel-ip 109 location 0/RP0/CPU0

----- node0_0_CPU0 -----
tunnel ifhandle 0x80022cc
tunnel source 161.115.1.2
tunnel destination 162.1.1.1/32
tunnel transport vrf table id 0xe0000000
tunnel mode gre ipv4, encap
tunnel bandwidth 100 kbps
tunnel platform id 0x0
tunnel flags 0x40003400
IntfStateUp
BcStateUp
Ipv4Caps
Encap
tunnel mtu 1500
tunnel tos 0
tunnel ttl 255
tunnel adjacency flags 0x1
tunnel o/p interface handle 0x0
tunnel key 0x0, entropy length 0 (mask 0xffffffff)
tunnel QT next 0x0
tunnel platform data (nil)
Platform:
Handle: (nil)
Decap ID: 0
Decap RIF: 0
Decap Recycle Encap ID: 0x00000000
Encap RIF: 0
Encap Recycle Encap ID: 0x00000000
Encap IPv4 Encap ID: 0x4001381b
Encap IPv6 Encap ID: 0x00000000
Encap MPLS Encap ID: 0x00000000
DecFEC DecRcyLIF DecStatsId EncRcyLIF
```

QoS テーブルが正しく更新されていることを確認します。

```
Router# show controllers npu stats voq base 48 instance all location
0/RP0/CPU0
Asic Instance = 0
VOQ Base = 48
       ReceivedPkts    ReceivedBytes    DroppedPkts    DroppedBytes
-------------------------------------------------------------------
COS0 =   0              0               0              0
COS1 =   0              0               0              0
COS2 =   0              0               0              0
COS3 =   0              0               0              0

Asic Instance = 1
VOQ Base = 48
       ReceivedPkts    ReceivedBytes    DroppedPkts    DroppedBytes
-------------------------------------------------------------------
COS0 =   0              0               0              0
```

```
COS1 =   0              0               0              0
COS2 =   0              0               0              0
COS3 =   0              0               0              0

Asic Instance = 2
VOQ Base = 48
       ReceivedPkts    ReceivedBytes    DroppedPkts    DroppedBytes
-------------------------------------------------------------------------
COS0 =   0              0               0              0
COS1 =   0              0               0              0
COS2 =   0              0               0              0
COS3 =   0              0               0              0
```