Cisco ASR 9000 アグリゲーション サービス ルータ インターフェイスおよびハードウェア コンポーネント コンフィギュレーション ガイド

Cisco IOS XR ソフトウェア Release 4.3.x

Text Part Number: OL-28377-01-J

【注意】シスコ製品をご使用になる前に、安全上の注意（www.cisco.com/jp/go/safety_warning/）をご確認ください。

本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動 / 変更されている場合がありますことをご了承ください。
あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。

また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

CONTENTS

はじめに

『*Cisco ASR 9000 シリーズ アグリゲーション サービス ルータ 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*』では、ルータ インターフェイスおよびハードウェア設定に関連する情報と手順について説明します。

では、次のトピックについて取り上げます。

- マニュアルの変更履歴
- マニュアルの入手方法およびテクニカル サポート

マニュアルの変更履歴

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

表 1　　マニュアルの変更履歴

リビジョン	日付	変更点
OL-28377-01-J	2012 年 12 月	このマニュアルの初版

マニュアルの入手方法およびテクニカル サポート

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

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

『*What's New in Cisco Product Documentation*』は RSS フィードとして購読できます。また、リーダー アプリケーションを使用してコンテンツがデスクトップに直接配信されるように設定することもできます。RSS フィードは無料のサービスです。シスコは現在、RSS バージョン 2.0 をサポートしています。

リリース 4.3.x の新機能と変更点

次の表は、『Cisco ASR 9000 アグリゲーション サービス ルータ 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』の新機能および変更された機能をまとめたものであり、それぞれの参照先を示しています。

表 2　　新機能および変更された機能

機能	説明	導入/変更されたリリース	参照先
Y.1731 合成損失測定	この機能が導入されました。	リリース 4.3.0	*Cisco ASR 9000 シリーズ ルータでのイーサネット OAM の設定* • 合成損失測定（ITU-T Y.1731） • Y.1731 Performance Monitoring • SLA 動作プロファイルの設定 • オンデマンド SLA 動作の設定 • イーサネット SLA Y.1731 SLM の設定：例 Y.1731 合成損失測定を設定および確認するために使用するコマンドの情報については、『Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Command Reference』の「*Ethernet OAM Commands on the Cisco ASR 9000 Series Router*」の章を参照してください。

機能	説明	導入 / 変更されたリリース	参照先
ASR 9000 の回線エミュレーション SPA のサポート	この機能が導入されました。	リリース 4.3.0	*Cisco ASR 9000 シリーズ ルータでの Circuit Emulation over Packet の設定* • Cisco 24 ポート チャネライズド T1/E1 回線エミュレーションおよびチャネライズド ATM SPA の設定と CEM インターフェイスの作成 • Cisco 2 ポート チャネライズド T3/E3 回線エミュレーションおよびチャネライズド ATM SPA の設定と CEM インターフェイスの作成
リモート サテライト クライアント ノード	この機能が導入されました。	リリース 4.3.0	*Cisco ASR 9000 シリーズ ルータでのサテライト ネットワーク仮想化（nV）システムの設定* • サテライト nV スイッチング システムの概要 • サテライト nV システムの実装 • サテライト nV ソフトウェアのアップグレードおよび管理 サテライトを設定および確認するために使用するコマンドの情報については、『Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Command Reference』の「*Satellite nV System Commands on the Cisco ASR 9000 Series Router*」の章を参照してください。

Cisco ASR 9000 シリーズ ルータでの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、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータでの物理インターフェイスのプリコンフィギュレーションについて説明します。

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

- ギガビット イーサネット
- 10 ギガビット イーサネット
- 管理イーサネット
- Packet-over-SONET/SDH（POS）
- シリアル（Serial）
- SONET コントローラおよびチャネライズド SONET コントローラ

プリコンフィギュレーションによって、モジュラ サービス カードをルータへの装着前に設定できます。カードを装着すると、ただちに設定されます。

プリコンフィギュレーション情報は、通常の方法で設定されたインターフェイスの場合とは異なり、別のシステム データベース ツリー（ルート スイッチ プロセッサ（RSP）上の*プリコンフィギュレーション ディレクトリ*）に作成されます。

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

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

物理インターフェイスのプリコンフィギュレーション機能の履歴

リリース	変更内容
リリース 3.7.2	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が導入されました。
リリース 4.0.0	POS インターフェイスのプリコンフィギュレーションが追加されました。

内容

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ための前提条件

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。

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前に、次の条件が満たされていることを確認してください。

- 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。

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おく必要があります。

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概要

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

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

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

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

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

インターフェイスのプリコンフィギュレーションを行う利点

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

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

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** コマンドやグローバル コンフィギュレーション コマンドを入力すると、プリコンフィギュレーションが完了します。

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

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

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

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

アクティブ/スタンバイ RSP および仮想インターフェイスの設定

スタンバイ RSP は、必要時に使用可能になり、アクティブ RSP から作業を引き継げる状態になります。スタンバイ RSP がアクティブ RSP となり、アクティブ RSP の役割を引き継ぐ必要のある状況を次に示します。

- ウォッチドッグによる障害検出
- スタンバイ RSP に対する管理上の引き継ぎ命令
- シャーシからのアクティブ RSP の取り外し

セカンダリ RSP がシャーシに搭載されていなかった場合、プライマリの稼働中にセカンダリ RSP を搭載すると、自動的にスタンバイ RSP になります。シャーシからスタンバイ RSP を取り外しても、RSP の冗長性が失われるだけで、システムに影響はありません。

フェールオーバー後、すべての仮想インターフェイスはスタンバイ（新たにアクティブになった）RSP に存在します。仮想インターフェイスの状態と設定は変更されず、フェールオーバー中にインターフェイス経由の転送（トンネルの場合）が失われることはありません。Cisco ASR 9000 シリーズ ルータは、ホスト RSP のフェールオーバーを通じて、トンネル上で無停止転送（NSF）を使用します。

（注） スタンバイ インターフェイス設定の維持されることを保証するために、設定は特に必要ありません。

物理インターフェイスのプリコンフィギュレーションを行う方法

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

手順の概要

1. **configure**
2. **interface preconfigure** *type interface-path-id*
3. **ipv4 address** *ip-address subnet-mask*
4. 追加のインターフェイス パラメータを設定します。
5. **end**
   または
   commit
6. **exit**

7. **exit**
8. **show running-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface preconfigure** *type interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface preconfigure GigabitEthernet 0/1/0/0`	インターフェイスのインターフェイス プリコンフィギュレーション モードを開始します。*type* では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* では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	**ipv4 address** *ip-address subnet-mask* または **ipv4 address** *ip-address*/*prefix* **例：** `RP/0/RSP0/CPU0:router(config-if-pre)# ipv4 address 192.168.1.2/32`	IP アドレスとマスクをインターフェイスに割り当てます。
ステップ 4	追加のインターフェイス パラメータを設定します。詳細については、設定するインターフェイスのタイプに対応する、このマニュアルの設定の章を参照してください。	

	コマンドまたはアクション	目的
ステップ 5	**end** または **commit** best-effort **例：** RP/0/RSP0/CPU0:router(config-if-pre)# end または RP/0/RSP0/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、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)? – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、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。 • 設定変更を実行コンフィギュレーション ファイルに保存し、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/RSP0/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。

物理インターフェイスのプリコンフィギュレーション例

ここでは、次の例について説明します。

「インターフェイスのプリコンフィギュレーション：例」（P.6）

インターフェイスのプリコンフィギュレーション：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例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface preconfigure GigabitEthernet 0/1/0/0
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 192.168.1.2/32
RP/0/RSP0/CPU0:router(config-if)# commit
```

その他の関連資料

ここでは、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関連する参考資料について説明します。

関連資料

関連項目	参照先
マスター コマンド リファレンス	*『Cisco ASR 9000 Series Aggregation Services Routers Master Command Listing』*
インターフェイス コンフィギュレーション コマンド	*『Cisco ASR 9000 Series Aggregation Services Routers Interface and Hardware Component Command Reference』*
初期システム起動と設定の情報	*『Cisco ASR 9000 Series Router Getting Started Guide』*
ユーザ グループとタスク ID に関する情報	*『Cisco IOS XR Task ID Reference Guide』*

標準

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

MIB

MIB	MIB のリンク
このモジュールに適用できる MIB はありません。	Cisco IOS XR ソフトウェアを使用して選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	—

シスコのテクニカル サポート

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

Cisco ASR 9000 シリーズ ルータ での管理イーサネット インターフェイスの高度な設定および変更

このモジュールでは、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータでの管理イーサネット 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には、『*Cisco ASR 9000 Series Router Getting Started Guide*』の「*Configuring General Router Features*」モジュールの説明に従って管理イーサネット インターフェイスをセットアップし、Telnet サーバをイネーブルにする必要があります。このモジュールでは、『*Cisco ASR 9000 Series Router Getting Started Guide*』の説明に従って管理イーサネット インターフェイスを設定した後に、そのデフォルト設定を変更する手順について説明します。

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

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

管理イーサネット インターフェイス設定機能の履歴

リリース	変更内容
リリース 3.7.2	この機能は、Cisco ASR 9000 シリーズ ルータで導入されました。

内容

- 「管理イーサネット インターフェイスの設定例」（P.18）
- 「その他の関連資料」（P.20）

管理イーサネット インターフェイス設定の前提条件

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。

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。

- 『*Cisco ASR 9000 Series Router Getting Started Guide*』の「*Configuring General Router Features*」モジュールの説明に従って、管理イーサネット インターフェイスの初期設定を実行しました。
- 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。
- 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。

  インターフェイスの命名規則の詳細については、『*Cisco ASR 9000 Series Router Getting Started Guide*』を参照してください。

（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。

管理イーサネット インターフェイスの設定に関する情報

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

- 「デフォルト インターフェイス設定」（P.10）

デフォルト インターフェイス設定

表 2 に、デフォルトの管理イーサネット インターフェイス設定を示します。これらの設定は、手動設定により変更できます。デフォルト設定は、**show running-config** コマンド出力には表示されません。

表 2　管理イーサネット インターフェイスのデフォルト設定

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

高度な管理イーサネット インターフェイス設定の実行方法

ここでは、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」（P.11）（必須）
- 「管理イーサネット インターフェイスのデュプレックス モードの設定」（P.13）（任意）
- 「管理イーサネット インターフェイスの速度の設定」（P.14）（任意）
- 「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」（P.16）（任意）
- 「管理イーサネット インターフェイス設定の確認」（P.17）（任意）

管理イーサネット インターフェイスの設定

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

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

（注）　『*Cisco ASR 9000 Series Router Getting Started Guide*』の「*Configuring General Router* Features」モジュールの説明に従って、すでに管理イーサネット インターフェイスをセットアップし、Telnet サーバをイネーブルにしている場合は、この作業を行う必要はありません。

手順の概要

1. **configure**
2. **interface MgmtEth** *interface-path-id*
3. **ipv4 address** *ip-address mask*
4. **no shutdown**

5. **end**
   または
   commit
6. **show interfaces MgmtEth** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface MgmtEth** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface MgmtEth 0/RSP0/CPU0/0`	インターフェイス コンフィギュレーション モードを開始し、イーサネット インターフェイス名と *rack*/*slot*/*module*/*port* 表記を指定します。 この例では、スロット 0 に装着された RSP カードのポート 0 を示しています。
ステップ 3	**ipv4 address** *ip-address mask* **例：** `RP/0/RSP0/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。 – スラッシュ（/）と数字による表記。たとえば、/8 は、マスクの最初の 8 ビットが 1 で、対応するアドレスのビットがネットワーク アドレスであることを示します。
ステップ 4	**no shutdown** **例：** `RP/0/RSP0/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。

	コマンドまたはアクション	目的
ステップ 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 6	**show interfaces MgmtEth** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show interfaces MgmtEth 0/RSP0/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。

管理イーサネット インターフェイスのデュプレックス モードの設定

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

手順の概要

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

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface MgmtEth** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface MgmtEth 0/RSP0/CPU0/0`	インターフェイス コンフィギュレーション モードを開始し、管理イーサネット インターフェイスの名前とインスタンスを指定します。
ステップ 3	**duplex** [**full** | **half**] **例：** `RP/0/RSP0/CPU0:router(config-if)# duplex full`	インターフェイスのデュプレックス モードを設定します。有効なオプションは **full** または **half** です。 **（注）** システムをオートネゴシエーションされたデュプレックス操作に戻すには、**no duplex** コマンドを使用します。
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

管理イーサネット インターフェイスの速度の設定

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

手順の概要

1. **configure**
2. **interface MgmtEth** *interface-path-id*

3. **speed** {**10** | **100** | **1000**}
4. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface MgmtEth** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface MgmtEth 0/RSP0/CPU0/0`	インターフェイス コンフィギュレーション モードを開始し、管理イーサネット インターフェイスの名前とインスタンスを指定します。
ステップ 3	**speed** {**10** | **100** | **1000**} **例：** `RP/0/RSP0/CPU0:router(config-if)# speed 100`	インターフェイス速度 speed パラメータを設定します。 Cisco ASR 9000 シリーズ ルータで有効な **speed** オプションは、**10** または **100** Mbps です。 **（注）** デフォルトの管理イーサネット インターフェイス速度はオートネゴシエーションされます。 **（注）** システムをオートネゴシエーションされたデフォルトの速度に戻すには、**no speed** コマンドを使用します。
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

管理イーサネット インターフェイスの MAC アドレスの変更

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

手順の概要

1. **configure**
2. **interface MgmtEth** *interface-path-id*
3. **mac-address** *address*
4. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface MgmtEth** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface MgmtEth 0/RSP0/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。

	コマンドまたはアクション	目的
ステップ 3	**mac-address** *address* **例：** `RP/0/RSP0/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** *address* コマンドを使用します。
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

管理イーサネット インターフェイス設定の確認

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

手順の概要

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

ステップ 1	**show interfaces MgmtEth** *interface-path-id* **例：** RP/0/RSP0/CPU0:router# show interfaces MgmtEth 0/RSP0/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/RSP0/CPU0:router# show running-config interface MgmtEth 0/RSP0/CPU0/0	実行コンフィギュレーションを表示します。

管理イーサネット インターフェイスの設定例

ここでは、次の設定例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：例」（P.18）

管理イーサネット インターフェイスの設定：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/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface MgmtEth 0/RSP0/CPU0/0
RP/0/RSP0/CPU0:router(config)# ipv4 address 172.29.52.70 255.255.255.0
RP/0/RSP0/CPU0:router(config-if)# speed 100
RP/0/RSP0/CPU0:router(config-if)# duplex full
RP/0/RSP0/CPU0:router(config-if)# no shutdown
RP/0/RSP0/CPU0:router(config-if)# commit
RP/0/RSP0/CPU0:Mar 26 01:09:28.685 :ifmgr[190]:%LINK-3-UPDOWN :Interface
MgmtEth0/RSP0/CPU0/0, changed state to Up
RP/0/RSP0/CPU0:router(config-if)# end

RP/0/RSP0/CPU0:router# show interfaces MgmtEth 0/RSP0/CPU0/0

MMgmtEth0/RSP0/CPU0/0 is up, line protocol is up
  Hardware is Management Ethernet, address is 0011.93ef.e8ea (bia 0011.93ef.e8ea
)
  Description: Connected to Lab LAN
  Internet address is 172.29.52.70/24
  MTU 1514 bytes, BW 100000 Kbit
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA,  loopback not set,
  ARP type ARPA, ARP timeout 04:00:00
  Last clearing of "show interface" counters never
  5 minute input rate 3000 bits/sec, 7 packets/sec
  5 minute output rate 0 bits/sec, 1 packets/sec
     30445 packets input, 1839328 bytes, 64 total input drops
     0 drops for unrecognized upper-level protocol
     Received 23564 broadcast packets, 0 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     57 input errors, 40 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     171672 packets output, 8029024 bytes, 0 total output drops
     Output 16 broadcast packets, 0 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
```

```
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions

RP/0/RSP0/CPU0:router# show running-config interface MgmtEth 0/RSP0/CPU0/0

interface MgmtEth0/RSP0/CPU0/0
 description Connected to Lab LAN
 ipv4 address 172.29.52.70 255.255.255.0
!
```

その他の関連資料

管理イーサネット インターフェイスの設定に関連する参考資料を示します。

関連資料

関連項目	参照先
Cisco ASR 9000 シリーズ ルータ マスター コマンド リファレンス	『*Cisco ASR 9000 Series Router Master Commands List*』
Cisco ASR 9000 シリーズ ルータ インターフェイス コンフィギュレーション コマンド	『*Cisco ASR 9000 Series Router Interface and Hardware Component Command Reference*』
Cisco IOS XR ソフトウェアを使用する Cisco ASR 9000 シリーズ ルータの初期システム ブートアップと設定に関する情報。	『*Cisco ASR 9000 Series Router Getting Started Guide*』
ユーザ グループとタスク ID に関する情報	『*Cisco ASR 9000 Series Router Interface and Hardware Component Command Reference*』

標準

標準	タイトル
この機能によりサポートされた新規規格または改訂規格はありません。またこの機能による既存規格のサポートに変更はありません。	—

MIB

MIB	MIB のリンク
このモジュールに適用できる MIB はありません。	Cisco IOS XR ソフトウェアを使用して選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	—

シスコのテクニカル サポート

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

その他の関連資料

Cisco ASR 9000 シリーズ ルータのイーサネット インターフェイスの設定

このモジュールでは、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータでのイーサネット インターフェイスの設定について説明します。

分散ギガビット イーサネットおよび 10 ギガビット イーサネット アーキテクチャと機能により、ネットワーク スケーラビリティとパフォーマンスを提供します。さらに、コアおよびエッジ ルータ、レイヤ 2 およびレイヤ 3 スイッチなど、ルータを POP で他のシステムと相互接続するように設計された高密度、高帯域幅のネットワーキング ソリューションの、サービス プロバイダーによる提供を可能にします。

Cisco ASR 9000 シリーズ ルータでのイーサネット インターフェイス設定の機能履歴

リリース	変更内容
リリース 3.7.2	次のラインカードのサポートが Cisco ASR 9000 シリーズ ルータに追加されました。 • 40 ポート ギガビット イーサネット中キューおよび高キュー ライン カード（A9K-40GE-B および A9K-40GE-E） • 4 ポート 10 ギガビット イーサネット中キューおよび高キュー ライン カード（A9K-4T-B および A9K-4T-E） • 8 ポート 10 ギガビット イーサネット中キューおよび高キュー DX ライン カード（A9K-8T/4-B および A9K-8T/4-E）（2:1 オーバーサブスクライブ型）

リリース 3.9.0	次のラインカードのサポートが Cisco ASR 9000 シリーズ ルータに追加されました。 • 40 ポート ギガビット イーサネット低キュー ラインカード（A9K-40GE-L） • 4 ポート 10 ギガビット イーサネット低キュー ラインカード（A9K-4T-L） • 8 ポート 10 ギガビット イーサネット低キュー DX ラインカード（A9K-8T/4-L）（2:1 オーバーサブスクライブ型） • 8 ポート 10 ギガビット イーサネット低および高キュー ラインカード（A9K-8T-L および A9K-8T-E） • 2 ポート 10 ギガビット イーサネット、20 ポート ギガビット イーサネット中キューおよび高キュー コンビネーション ラインカード（A9K-2T20GE-B および A9K-2T20GE-L） 次の機能のサポートが追加されました。 • 周波数同期化 • SyncE
リリース 3.9.1	次のラインカードのサポートが Cisco ASR 9000 シリーズ ルータに追加されました。 • 8 ポート 10 ギガビット イーサネット中キュー ラインカード（A9K-8T-B） • 16 ポート 10 ギガビット イーサネット SFP+ ラインカード（A9K-16T/8-B および A9K-16T/8-B+AIP）
リリース 4.0.1	レイヤ 2 サブインターフェイス（EFP）でのパフォーマンス モニタリングのためのレイヤ 2 統計情報収集のサポートが追加されました。
リリース 4.1.0	Link Layer Discovery Protocol（LLDP）のサポートが追加されました。
リリース 4.1.1	MAC アドレス アカウンティング機能のサポートが追加されました。
リリース 4.2.2	単方向リンク ルーティング（UDLR）のサポートが追加されました。

内容

イーサネット インターフェイスの前提条件

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。

イーサネット インターフェイスを設定する前に、次のタスクと条件を満たしていることを確認します。

- ルータでサポートされる次のラインカードが少なくとも 1 つインストールされていることを確認してください。
  - 2 ポート 10 ギガビット イーサネット、20 ポート ギガビット イーサネットの組み合わせラインカード（A9K-2T20GE-B および A9K-2T20GE-L）
  - 4 ポート 10 ギガビット イーサネット ラインカード（A9K-4T-L、-B、または -E）
  - 8 ポート 10 ギガビット イーサネット DX ラインカード（A9K-8T/4-L、-B、または -E）
  - 8 ポート 10 ギガビット イーサネット ラインカード（A9K-8T-L、-B、または -E）
  - 16 ポート 10 ギガビット イーサネット SFP+ ラインカード（A9K-16T/8-B および A9K-16T/8-B+AIP）
  - 40 ポート ギガビット イーサネット ラインカード（A9K-40GE-L、-B、または -E）
- インターフェイスの IP アドレスを知っています。
- 汎用インターフェイス名に汎用表記法の *rack/slot/module/port* を適用する方法を理解しています。

イーサネットの設定に関する情報

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

Cisco ASR 9000 シリーズ ルータは、ギガビット イーサネット（1000 Mbps）インターフェイスおよび 10 ギガビット イーサネット（10 Gbps）インターフェイスをサポートしています。

この項では、次の情報について説明します。

16 ポート 10 ギガビット イーサネット SFP+ ラインカード

16 ポート 10 ギガビット イーサネット SFP+ ラインカードは、Small Form Factor（SFP トランシーバ）光ラインカードの 1 つであり、Cisco IOS XR Release 3.9.1 で Cisco ASR 9000 シリーズ ルータに導入されました。16 ポート 10 ギガビット イーサネット SFP+ ラインカードは、ルータで現在サポートされているギガビット イーサネットのコマンドと設定すべてをサポートします。

16 ポート 10 ギガビット イーサネット SFP+ ラインカードは、Cisco ASR 9000 シリーズ ルータの既存のすべてのラインカード、ルート/スイッチ プロセッサ（RSP）、およびシャーシと互換性があります。

機能

16 ポート 10 ギガビット イーサネット SFP+ ラインカードでは次の機能がサポートされます。

- 16 個の 10 ギガビット イーサネット ポート
- システムごとに 128 個の 10 ギガビット イーサネット ポート
- システムごとに 1.28 Tbps
- 160 Gbps の転送
- 120 Gbps 双方向パフォーマンス

- SR/LR/ER SFP+ 光
- 既存のラインカードと同等の機能
- 160 Gbps でのユニキャストおよびマルチキャスト転送（RSP スイッチオーバー中のパケット損失ゼロ）

制約事項

次の機能は 16 ポート 10 ギガビット イーサネット SFP+ ラインカードでサポートされません。

- DWDM（G.709）

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

表 3 は、ギガビット イーサネットまたは 10 ギガビット イーサネットのモジュラ サービス カードおよび PC の脅威対策 PLIM でインターフェイスをイネーブルにしたときに表示される、デフォルトのインターフェイス設定パラメータを示します。

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

表 3　ギガビット イーサネットおよび 10 ギガビット イーサネット モジュラ サービス カードのデフォルト設定値

パラメータ	設定ファイルのエントリ	デフォルト値
MAC アカウンティング	**mac-accounting**	**off**
フロー制御	**flow-control**	出力オン 入力オフ
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）

表 4　ファスト イーサネットのデフォルト設定値

パラメータ	設定ファイルのエントリ	デフォルト値
MAC アカウンティング	**mac-accounting**	**off**
デュプレックス操作	**duplex full** **duplex half**	Auto-negotiates duplex operation
MTU	**mtu**	1500 バイト
インターフェイス速度	**speed**	100 Mbps
オートネゴシエーション	**negotiation auto**	**disable**

イーサネット インターフェイスでのレイヤ 2 VPN

レイヤ 2 バーチャル プライベート ネットワーク（L2VPN）接続は、L2 スイッチド、IP または MPLS 対応の IP ネットワーク上で LAN の動作をエミュレートするものであり、この接続を利用すると、イーサネット デバイス同士が、共通の LAN セグメントに接続されているかのように通信できます。

L2VPN の機能によって、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。通常、SP はアクセス ネットワークを使用して、カスタマーをコア ネットワークに接続します。Cisco ASR 9000 シリーズ ルータでこのアクセス ネットワークは、通常、イーサネットです。

カスタマーからのトラフィックは、このリンク上で SP コア ネットワークのエッジへ伝送されます。このトラフィックは、SP コア ネットワーク上の L2VPN を介して別のエッジ ルータへ伝送されます。エッジ ルータはこのトラフィックを、別の接続回線（AC）を通してカスタマーのリモート サイトまで送信します。

Cisco ASR 9000 シリーズ ルータで AC は、ブリッジ ドメイン、疑似回線またはローカル接続などの L2VPN コンポーネンに接続するインターフェイスです。

L2VPN 機能を利用すると、さまざまなタイプのエンドツーエンド サービスを実装することができます。

Cisco IOS XR ソフトウェアは、ポイントツーポイント エンドツーエンド サービスをサポートしています。つまり、2 つのイーサネット回路が相互に接続されます。L2VPN イーサネット ポートは、次の 2 モードのいずれかで動作します。

- ポート モード : このモードでは、ポートに到達するすべてのパケットは、パケット上に存在する VLAN タグに関係なく、PW（疑似回線）上で送信されます。VLAN モードでは、l2transport コンフィギュレーション モードで設定が実行されます。
- VLAN モード : CE（カスタマー エッジ）の各 VLAN または PE（プロバイダー エッジ）リンクへのアクセス ネットワークは個別の L2VPN 接続として設定できます（VC タイプ 4 または VC タイプ 5 を使用する）。VLAN モードでは、個別のサブインターフェイスで設定を実行します。

切り替えは次の 3 つの方法で実行できます。

- AC-to-PW : PE に到達したトラフィックは PW を介してトンネリングされます（反対に、PW を介して到達したトラフィックは AC を介して送信されます）。これが最も一般的なシナリオです。
- ローカルの切り替え - 1 つの AC 上で到達するトラフィックは、疑似接続を介さずに別の AC へ送出されます。
- PW 切り替え - PW に到達するトラフィックは AC へ送信されませんが、別の PW 上でコアに返信されます。

イーサネット インターフェイスで L2VPN を設定する場合、次の点に気を付けてください。

- L2VPN リンクは QoS（Quality of Service）および MTU（最大伝送単位）の設定をサポートしています。
- ネットワークの要件として、パケットを透過的に伝送することが必須の場合は、サービス プロバイダー（SP）ネットワークのエッジにおいてパケットの宛先 MAC（メディア アクセス コントロール）アドレスを変更することが必要になる可能性があります。こうすることで、SP ネットワークのデバイスによるパケットの消費が回避されます。

AC と PW の情報を表示するには、**show interfaces** コマンドを使用します。

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

- *『Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide』*
- *『Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Command Reference』*

レイヤ 2 サービス ポリシー、たとえば QoS をイーサネット インターフェイスにアタッチするには、該当する Cisco IOS XR ソフトウェアのコンフィギュレーション ガイドを参照してください。

ギガビット イーサネット プロトコル規格の概要

ギガビット イーサネット インターフェイスは次のプロトコル規格をサポートしています。

- 「IEEE 802.3 物理イーサネット インフラストラクチャ」（P.29）
- 「IEEE 802.3ab 1000BASE-T ギガビット イーサネット」（P.29）
- 「IEEE 802.3z 1000 Mbps ギガビット イーサネット」（P.30）
- 「IEEE 802.3ae 10 Gbps イーサネット」（P.30）

各規格の詳細については、このマニュアルで後述します。

IEEE 802.3 物理イーサネット インフラストラクチャ

IEEE 802.3 プロトコル規格では、接続するイーサネットの物理層とデータリンク層の MAC 下位層が定義されています。IEEE 802.3 では、多様な物理メディアで、また多様な速度でキャリア検知多重アクセス / 衝突検出（CSMA/CD）アクセスを使用します。IEEE 802.3 規格は 10 Mbps イーサネットに対応します。IEEE 802.3 規格の拡張では、ギガビット イーサネット、10 ギガビット イーサネット、およびファスト イーサネットの実装を規定しています。

IEEE 802.3ab 1000BASE-T ギガビット イーサネット

IEEE 802.3ab プロトコル規格、つまり銅線上のギガビット イーサネット（別名 1000BaseT）は、既存のファスト イーサネット規格の拡張です。この拡張は、すでに設置されているカテゴリ 5e/6 ケーブル配線システム上のギガビット イーサネットの動作を規定しており、費用有効性の高いソリューションを実現できます。結果として、ファスト イーサネットを実行する銅線ベースの環境では既存のインフラストラクチャ上でギガビット イーサネットも実行できるため、要求の厳しいアプリケーションでもネットワークのパフォーマンスが大幅に向上します。

IEEE 802.3z 1000 Mbps ギガビット イーサネット

ギガビット イーサネットはイーサネット プロトコルの上で構築されますが、速度はファスト イーサネットの 10 倍で、1000 Mbps（1 Gbps）に上がります。ギガビット イーサネットを使用すると、デスクトップで 10 Mbps または 100 Mbps、データセンターで最高 1000 Mbps までイーサネットを拡張できます。ギガビット イーサネットは IEEE 802.3z プロトコル規格に準拠します。

ネットワーク管理者は、現在のイーサネット規格と、すでに設置されているイーサネットおよびファスト イーサネットのスイッチおよびルータのベースを利用することで、ギガビット イーサネットをサポートするために新しいテクノロジーのトレーニングや学習をし直す必要はなくなります。

IEEE 802.3ae 10 Gbps イーサネット

国際標準化組織の開放型システム間相互接続（OSI）モデルでは、イーサネットは基本的にレイヤ 2 プロトコルです。10 ギガビット イーサネットでは、IEEE 802.3 イーサネット MAC プロトコル、IEEE 802.3 イーサネット フレーム形式、および IEEE 802.3 の最小および最大フレーム サイズを使用します。10 Gbps イーサネットは IEEE 802.3ae プロトコル規格に準拠します。

イーサネット モデルに忠実だった 1000BASE-X と 1000BASE-T（ギガビット イーサネット）と同様に、10 ギガビット イーサネットも速度と距離の点でイーサネットが自然に発展した結果です。10 ギガビット イーサネットは全二重方式でファイバのみのテクノロジーなので、低速で半二重方式のイーサネット テクノロジーを定義する CSMA/CD プロトコルを使用した、通信事業者に影響される多重アクセスは必要ありません。他のどの点でも、10 ギガビット イーサネットは元のイーサネット モデルに忠実です。

IEEE 802.3ba 100 Gbps イーサネット

IEEE 802.3ba は、Cisco IOS XR 4.0.1 から、シスコの 1 ポート 100 ギガビット イーサネット PLIM でサポートされます。

MAC Address

MAC アドレスは、レイヤ 2 のインターフェイスを識別する固有の 6 バイト アドレスです。

MAC アカウンティング

MAC アドレス アカウンティング機能を使用すると、LAN インターフェイスの発信元および宛先の MAC アドレスに基づいた IP トラフィックのアカウンティング情報がわかります。この機能では、LAN インターフェイスが固有の MAC アドレスとの間で送受信する IP パケットの合計パケット数および合計バイト数が計算されます。また、最終受信または最終送信のタイム スタンプも記録されます。

これらの統計情報はトラフィック モニタリング、デバッグ、および課金に使用されます。たとえば、この機能を利用すると、NAPS/ ピアリング ポイントにおいてさまざまなピアとの間で送受信されるトラフィックの量を特定できます。この機能が現時点でサポートされているのは、イーサネット、ファスト イーサネット、およびバンドル インターフェイス上です。この機能は、シスコ エクスプレス フォワーディング（CEF）、分散 CEF（dCEF）、フロー、および最適なスイッチングをサポートします。

（注） トランク インターフェイスごとに最大 512 個の MAC アドレスが MAC アドレス アカウンティングに対してサポートされます。

イーサネット MTU

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

Cisco IOS XR ソフトウェアは、2 種類のフレーム転送プロセスをサポートしています。

- IPV4 パケットのフラグメンテーション：このプロセスでは、ネクスト ホップの物理ネットワークの MTU 内に収まるように、必要に応じて IPv4 パケットが分割されます。

  （注） IPv6 はフラグメンテーションをサポートしません。

- MTU の検出プロセスによる最大パケット サイズの決定：このプロセスは、すべての IPV6 デバイスと発信側の IPv4 デバイスに使用できます。このプロセスでは、分割せずに送信できる IPv6 または IPv4 パケットの最大サイズを、発信側の IP デバイスが決定します。最大パケットは、IP 発信元デバイスおよび IP 宛先デバイス間にあるすべてのネットワークの中で、最小 MTU と等値です。このパス内にあるすべてのネットワークの最小 MTU よりもパケットが大きい場合、そのパケットは必要に応じて分割されます。このプロセスによって、発信側のデバイスから大きすぎる IP パケットが送信されなくなります。

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

イーサネット インターフェイスでのフロー制御

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

802.1Q VLAN

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

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

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

VRRP

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

VRRP の詳細については、『*Cisco ASR 9000 Series Router IP Addresses and Services Configuration Guide*』の「*Implementing VRRP*」モジュールを参照してください。

HSRP

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

HSRP は、特定の状況で IP トラフィックを中断しない切り替えをサポートし、ホストからは単一のルータを使用しているように見え、使用している第 1 ホップのルータに障害が発生した場合でも接続を維持できるように設計されています。つまり、HSRP は、発信元のホストが第 1 ホップのルータの IP アドレスを動的に取得できない場合でも、第 1 ホップのルータの障害に対処できます。複数のルータが HSRP に参加し、連携して単一の仮想ルータであるように見せます。HSRP によって、確実に単一のルータが仮想ルータの代わりにパケットを転送します。エンド ホストがそのパケットを仮想ルータに転送します。

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

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

HSRP の詳細については、『*Cisco ASR 9000 Series Router IP Addresses and Services Configuration Guide*』の「*Implementing HSRP*」モジュールを参照してください。

イーサネット インターフェイスのリンクのオートネゴシエーション

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

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

表 5 は、速度モードのさまざまな組み合わせ別のシステム パフォーマンスの説明です。指定された **speed** コマンドによってこのとおりにシステムが動作するには、インターフェイス上で自動ネゴシエーションが設定済みであることが条件となります。

表 5　　duplex コマンドと speed コマンドの関係

duplex コマンド	speed コマンド	システムの動作
no duplex	**no speed**	速度モードとデュプレックス モードの両方がオートネゴシエーションされます。
no duplex	**speed 1000**	デュプレックス モードがオートネゴシエーションされ、強制的に 1000 Mbps が指定されます。
no duplex	**speed 100**	デュプレックス モードがオートネゴシエーションされ、強制的に 100 Mbps が指定されます。
no duplex	**speed 10**	デュプレックス モードがオートネゴシエーションされ、強制的に 10 Mbps が指定されます。
full-duplex	**no speed**	強制的に全二重モードが指定され、速度はオートネゴシエーションされます。
full-duplex	**speed 1000**	強制的に全二重モードと 1000 Mbps が指定されます。
full-duplex	**speed 100**	強制的に全二重モードと 100 Mbps が指定されます。
full-duplex	**speed 10**	強制的に全二重モードと 10 Mbps が指定されます。
half-duplex	**no speed**	強制的に半二重モードが指定され、速度はオートネゴシエーションされます。
half-duplex	**speed 1000**	強制的に半二重モードと 1000 Mbps が指定されます。
half-duplex	**speed 100**	強制的に半二重モードと 100 Mbps が指定されます。
half-duplex	**speed 10**	強制的に半二重モードと 10 Mbps が指定されます。

Cisco ASR 9000 シリーズ ルータのサブインターフェイス

Cisco IOS XR では、インターフェイスは、デフォルトではメイン インターフェイスです。メイン インターフェイスは、VLAN トランキングのコンテキストでのトランクという単語の用法と混同しないように、トランク インターフェイスとも呼ばれます。

3 種類のトランク インターフェイスがあります。

- 物理
- バンドル

Cisco ASR 9000 シリーズ ルータでは、物理インターフェイスはルータがカードとその物理インターフェイスを認識する際、自動的に作成されます。ただし、バンドル インターフェイスは自動作成されません。これらはユーザに設定されたときに作成されます。

次の設定例は、作成されるトランク インターフェイスの例です。

- interface gigabitethernet 0/5/0/0
- interface bundle-ether 1

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

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

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

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

1001、0、97、96、100000

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

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

```
RP/0/RSP0/CPU0:router#  conf
Mon Sep 21 11:12:11.722 EDT
RP/0/RSP0/CPU0:router(config)#  interface GigabitEthernet0/5/0/0.0
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RSP0/CPU0:router(config-subif)# commit

RP/0/RSP0/CPU0:Sep 21 11:12:34.819 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT :
Configuration committed by user 'root'.Use 'show configuration commit changes
1000000152' to view the changes.

RP/0/RSP0/CPU0:router(config-subif)# end

RP/0/RSP0/CPU0:Sep 21 11:12:35.633 : config[65794]: %MGBL-SYS-5-CONFIG_I : Configured
from console by root
RP/0/RSP0/CPU0:router#
```

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

```
RP/0/RSP0/CPU0:router# show run | begin GigabitEthernet0/5/0/0
Mon Sep 21 11:15:42.654 EDT
Building configuration...
interface GigabitEthernet0/5/0/0
 shutdown
!
interface GigabitEthernet0/5/0/0.0
 encapsulation dot1q 100
!
interface GigabitEthernet0/5/0/1
 shutdown
!
```

サブインターフェイスが初めて作成されたときは、Cisco ASR 9000 シリーズ ルータはそのインターフェイスがトランク インターフェイスと交換可能であると認識します（いくつかの例外があります）。新しいサブインターフェイスの設定をさらに行った後で、**show interface** コマンドを実行すると、そのサブインターフェイスが一意のカウンタとともに表示されます。

次に、トランク インターフェイス GigabitEthernet 0/5/0/0 の表示出力を、その後にサブインターフェイス GigabitEthernet 0/5/0/0.0 の表示出力の例を示します。

```
RP/0/RSP0/CPU0:router# show interface gigabitEthernet 0/5/0/0
Mon Sep 21 11:12:51.068 EDT
```

```
GigabitEthernet0/5/0/0 is administratively down, line protocol is administratively
down
  Interface state transitions: 0
  Hardware is GigabitEthernet, address is 0024.f71b.0ca8 (bia 0024.f71b.0ca8)
  Internet address is Unknown
  MTU 1514 bytes, BW 1000000 Kbit
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation 802.1Q Virtual LAN,
  Full-duplex, 1000Mb/s, SXFD, link type is force-up
  output flow control is off, input flow control is off
  loopback not set,
  ARP type ARPA, ARP timeout 04:00:00
  Last input never, output never
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions


RP/0/RSP0/CPU0:router# show interface gigabitEthernet0/5/0/0.0
Mon Sep 21 11:12:55.657 EDT
GigabitEthernet0/5/0/0.0 is administratively down, line protocol is administratively
down
  Interface state transitions: 0
  Hardware is VLAN sub-interface(s), address is 0024.f71b.0ca8
  Internet address is Unknown
  MTU 1518 bytes, BW 1000000 Kbit
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation 802.1Q Virtual LAN, VLAN Id 100,  loopback not set,
  ARP type ARPA, ARP timeout 04:00:00
  Last input never, output never
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
```

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

```
RP/0/RSP0/CPU0:router# conf
Mon Sep 21 10:57:31.736 EDT
RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether1
RP/0/RSP0/CPU0:router(config-if)# no shut
RP/0/RSP0/CPU0:router(config-if)# interface bundle-Ether1.0
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RSP0/CPU0:router(config-subif)# commit
RP/0/RSP0/CPU0:Sep 21 10:58:15.305 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT : C
onfiguration committed by user 'root'.Use 'show configuration commit changes 10
00000149' to view the changes.
RP/0/RSP0/CPU0:router# show run | begin Bundle-Ether1
Mon Sep 21 10:59:31.317 EDT
```

```
Building configuration...
interface Bundle-Ether1
!
interface Bundle-Ether1.0
 encapsulation dot1q 100
!
```

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

```
RP/0/RSP0/CPU0:router#
RP/0/RSP0/CPU0:router# show run | begin GigabitEthernet0/5/0/0
Mon Sep 21 11:42:27.100 EDT
Building configuration...
interface GigabitEthernet0/5/0/0
 negotiation auto
!
interface GigabitEthernet0/5/0/0.0
 encapsulation dot1q 100
!
interface GigabitEthernet0/5/0/1
 shutdown
!
RP/0/RSP0/CPU0:router# conf
Mon Sep 21 11:42:32.374 EDT
RP/0/RSP0/CPU0:router(config)# no interface GigabitEthernet0/5/0/0.0
RP/0/RSP0/CPU0:router(config)# commit
RP/0/RSP0/CPU0:Sep 21 11:42:47.237 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT :
Configuration committed by user 'root'.Use 'show configuration commit changes
1000000159' to view the changes.
RP/0/RSP0/CPU0:router(config)# end
RP/0/RSP0/CPU0:Sep 21 11:42:50.278 : config[65794]: %MGBL-SYS-5-CONFIG_I : Configured
from console by root
RP/0/RSP0/CPU0:router# show run | begin GigabitEthernet0/5/0/0
Mon Sep 21 11:42:57.262 EDT
Building configuration...
interface GigabitEthernet0/5/0/0
 negotiation auto
!
interface GigabitEthernet0/5/0/1
 shutdown
!
```

レイヤ 2、レイヤ 3、および EFP

Cisco ASR 9000 シリーズ ルータでは、トランク インターフェイスはレイヤ 2 またはレイヤ 3 インターフェイスにする必要があります。レイヤ 2 インターフェイスで **l2transport** キーワードを指定した **interface** コマンドを使用して設定します。**l2transport** キーワードを使用しない場合、インターフェイスはレイヤ 3 インターフェイスです。サブインターフェイスは、レイヤ 2 またはレイヤ 3 サブインターフェイスで同じように設定されます。

レイヤ 3 トランク インターフェイスまたはサブインターフェイスは、ルーテッド インターフェイスであり、IP アドレスを割り当てることができます。そのインターフェイスで送信されるトラフィックはルーティングされます。

レイヤ 2 トランク インターフェイスまたはサブインターフェイスはスイッチド インターフェイスであり、IP アドレスを割り当てることができません。レイヤ 2 インターフェイスは、L2VPN コンポーネントに接続する必要があります。これが接続されている場合、アクセス接続と呼ばれます。

サブインターフェイスは、レイヤ 3 トランク インターフェイスの下にのみ作成できます。サブインターフェイスは、レイヤ 2 トランク インターフェイスの下に作成できません。

レイヤ 3 トランク インターフェイスは、レイヤ 2 とレイヤ 3 インターフェイスを組み合わせて使用できます。

次に、レイヤ 2 トランクの下にサブインターフェイスを設定しようとして、コミット エラーが発生する例を示します。レイヤ 2 トランク インターフェイスをレイヤ 3 インターフェイスに変更しようとして、インターフェイスにすでに IP アドレスが関連付けられているためにエラーが発生する例も示します。

```
RP/0/RSP0/CPU0:router# config
Mon Sep 21 12:05:33.142 EDT
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/5/0/0
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 10.0.0.1/24
RP/0/RSP0/CPU0:router(config-if)# commit
RP/0/RSP0/CPU0:Sep 21 12:05:57.824 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT :
Configuration committed by user 'root'.Use 'show configuration commit changes
1000000160' to view the changes.
RP/0/RSP0/CPU0:router(config-if)# end
RP/0/RSP0/CPU0:Sep 21 12:06:01.890 : config[65794]: %MGBL-SYS-5-CONFIG_I : Configured
from console by root
RP/0/RSP0/CPU0:router# show run | begin GigabitEthernet0/5/0/0
Mon Sep 21 12:06:19.535 EDT
Building configuration...
interface GigabitEthernet0/5/0/0
 ipv4 address 10.0.0.1 255.255.255.0
 negotiation auto
!
interface GigabitEthernet0/5/0/1
 shutdown
!
RP/0/RSP0/CPU0:router#
RP/0/RSP0/CPU0:router#
RP/0/RSP0/CPU0:router# conf
Mon Sep 21 12:08:07.426 EDT
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/5/0/0 l2transport
RP/0/RSP0/CPU0:router(config-if-l2)# commit

% Failed to commit one or more configuration items during a pseudo-atomic
operation.All changes made have been reverted.Please issue 'show configuration failed'
from this session to view the errors
RP/0/RSP0/CPU0:router(config-if-l2)# no ipv4 address
RP/0/RSP0/CPU0:router(config-if)# commit
RP/0/RSP0/CPU0:Sep 21 12:08:33.686 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT :
Configuration committed by user 'root'.Use 'show configuration commit changes
1000000161' to view the changes.
RP/0/RSP0/CPU0:router(config-if)# end
RP/0/RSP0/CPU0:Sep 21 12:08:38.726 : config[65794]: %MGBL-SYS-5-CONFIG_I : Configured
from console by root
RP/0/RSP0/CPU0:router#
RP/0/RSP0/CPU0:router# show run interface GigabitEthernet0/5/0/0
Mon Sep 21 12:09:02.471 EDT
interface GigabitEthernet0/5/0/0
 negotiation auto
 l2transport
 !
!
RP/0/RSP0/CPU0:router#
RP/0/RSP0/CPU0:router# conf
Mon Sep 21 12:09:08.658 EDT
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/5/0/0.0
                                                        ^
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/5/0/0.0
RP/0/RSP0/CPU0:router(config-subif)# commit
```

```
% Failed to commit one or more configuration items during a pseudo-atomic
operation.All changes made have been reverted.Please issue 'show configuration failed'
from this session to view the errors
RP/0/RSP0/CPU0:router(config-subif)#
RP/0/RSP0/CPU0:router(config-subif)# interface GigabitEthernet0/5/0/0
RP/0/RSP0/CPU0:router(config-if)# no l2transport
RP/0/RSP0/CPU0:router(config-if)# interface GigabitEthernet0/5/0/0.0
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 99
RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 11.0.0.1/24
RP/0/RSP0/CPU0:router(config-subif)# interface GigabitEthernet0/5/0/0.1 l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 700
RP/0/RSP0/CPU0:router(config-subif)# commit
RP/0/RSP0/CPU0:Sep 21 12:11:45.896 : config[65794]: %MGBL-CONFIG-6-DB_COMMIT :
Configuration committed by user 'root'.Use 'show configuration commit changes
1000000162' to view the changes.
RP/0/RSP0/CPU0:router(config-subif)# end
RP/0/RSP0/CPU0:Sep 21 12:11:50.133 : config[65794]: %MGBL-SYS-5-CONFIG_I : Configured
from console by root
RP/0/RSP0/CPU0:router#
RP/0/RSP0/CPU0:router# show run | b GigabitEthernet0/5/0/0
Mon Sep 21 12:12:00.248 EDT
Building configuration...
interface GigabitEthernet0/5/0/0
 negotiation auto
!
interface GigabitEthernet0/5/0/0.0
 ipv4 address 11.0.0.1 255.255.255.0
 encapsulation dot1q 99
!
interface GigabitEthernet0/5/0/0.1 l2transport
 encapsulation dot1q 700
!
interface GigabitEthernet0/5/0/1
 shutdown
!
```

すべてのサブインターフェイスで、ルータが正しいサブインターフェイスに着信パケットとフレームを送信できるように、一意のカプセル化ステートメントが必要です。サブインターフェイスのカプセル化ステートメントが存在しない場合、ルータはトラフィックを送信しません。

Cisco IOS XR では、イーサネット フロー ポイント（EFP）がレイヤ 2 サブインターフェイスとして実装されるため、レイヤ 2 サブインターフェイスは EFP とよく呼ばれます。EFP の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide*』を参照してください。

レイヤ 2 トランク インターフェイスは、アクセス接続として使用できます。ただし、EFP が、定義上は、トラフィックの全体的なストリームであるサブストリームであるため、レイヤ 2 トランク インターフェイスは EFP ではありません。

Cisco IOS XR には、レイヤ 2 またはレイヤ 3 インターフェイスとして設定可能な内容にその他の制限もあります。特定の設定ブロックは、レイヤ 3 だけを受け入れ、レイヤ 2 は受け入れません。たとえば、OSPF はレイヤ 3 トランクおよびサブインターフェイスだけを受け入れます。その他の制約事項については、適切な Cisco IOS XR のコンフィギュレーション ガイドを参照してください。

レイヤ 2 サブインターフェイス（EFP）の拡張パフォーマンス モニタリング

Cisco IOS XR Release 4.0.1 以降のリリースでは、Cisco ASR 9000 シリーズ ルータはレイヤ 2 サブインターフェイスのパフォーマンス モニタリングの基本的なカウンタのサポートが追加されます。

ここでは、レイヤ 2 インターフェイス カウンタの新しいサポートの概要について説明します。パフォーマンス モニタリングを設定する方法の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router System Monitoring Configuration Guide*』の「Implementing Performance Management」の章を参照してください。

interface basic-counters キーワードは、パフォーマンス統計情報収集の新しいエンティティをサポートし、次のコマンドでレイヤ 2 インターフェイスで表示するために追加されています。

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

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

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

その他のパフォーマンス管理の機能拡張

次の追加のパフォーマンス管理の拡張は、Cisco IOS XR Release 4.0.1 に含まれています。

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

周波数の同期および SyncE

Cisco IOS XR ソフトウェアは、Cisco ASR 9000 シリーズ ルータ上で SyncE 対応イーサネットをサポートします。周波数の同期はネットワーク全体に正確なクロック信号を配信する機能を提供します。非常に正確なタイミング信号が最初に、外部タイミング テクノロジー（セシウム原子時計、GPS など）からネットワーク内の Cisco ASR 9000 シリーズ ルータに送信され、この信号が、ルータの物理インターフェイスのクロッキングに使用されます。ピア ルータは、回線からのこの正確な周波数を正常に戻し、さらにネットワーク全体にこれを転送できます。この機能は、従来は SONET/SDH ネットワークに適用されていましたが、現在ではイーサネット上でも Cisco ASR 9000 シリーズ アグリゲーション サービス ルータに適用されており、そのための機能が「同期イーサネット」です。詳細については、『Cisco ASR 9000 Series Aggregation Services Router System Management Configuration Guide』を参照してください。

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、他のデバイスとの相互運用性を確保するために、Cisco ASR 9000 シリーズ ルータは 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働する 2 つのシステムで互いの情報を学習できます。

LLDP は、ネイバー デバイスに関する情報の学習に使用される属性セットをサポートします。これらの属性には Type-Length-Value（TLV）と呼ばれる定義された形式があります。LLDP をサポートするデバイスは、ネイバーとの情報の送受信に TLV を使用できます。設定情報、デバイスの機能、デバイス ID などの詳細情報は、このプロトコルを使用してアドバタイズできます。

必須 TLV（シャーシ ID、ポート ID、および存続可能時間）に加えて、ルータは、次のオプションの基本管理 TLV もサポートしています。

- Port Description
- System Name
- System Description
- システム機能
- Management Address

これらオプションの TLV は、LLDP がアクティブの場合、自動的に送信されますが、必要に応じて **lldp tlv-select disable** コマンドを使用してディセーブルにできます。

LLDP フレーム形式

LLDP フレームは次のフィールドで構成される IEEE 802.3 形式を使用します。

- 宛先アドレス（6 バイト）: 01-80-C2-00-00-0E のマルチキャスト アドレスを使用します。
- 送信元アドレス（6 バイト）: 送信側デバイスまたはポートの MAC アドレス。
- LLDP Ethertype（2 バイト）: 88-CC を使用します。
- LLDP PDU（1500 バイト）: TLV で構成される LLDP ペイロード。

- FCS（4 バイト）：エラー チェック用の巡回冗長検査（CRC）。

LLDP TLV 形式

LLDP TLV は次の基本形式を使用して LLDP PDU 内のネイバー デバイスに関する情報が含まれます。

- 次のフィールドを含む TLV ヘッダー（16 ビット）
  - TLV タイプ（7 ビット）
  - TLV 情報文字列の長さ（9 ビット）
- TLV 情報文字列（0 ～ 511 バイト）

LLDP 動作

LLDP は一方向のプロトコルです。LLDP の基本動作は、受信デバイスに LLDP フレームの情報の定期的なアドバタイズメントを送信する、LLDP 情報の送信に対応したデバイスで構成されます。

シャーシ ID とポート ID TLV の組み合わせを使用して MSAP（MAC サービス アクセスポイント）を作成し、デバイスが識別されます。受信デバイスは、情報をエージングして削除するまで、TTL TLV で指定された一定時間のネイバーに関する情報を保存します。

LLDP は次の追加の動作特性をサポートします。

- LLDP は送信または受信モードで個別に動作できます。
- LLDP は毎秒 5 フレーム未満の送信速度のタグなしフレームだけを使用して低速プロトコルとして動作します。
- LLDP パケットは次の場合に送信されます。
  - **lldp timer** コマンドで指定したパケット更新頻度に到達した。デフォルトは 30 秒です。
  - ローカル システムの LLDP MIB によって管理対象オブジェクトの値が変わった。
  - LLDP がインターフェイスでアクティブになった（3 フレームが CDP と同様にアクティベーション時に送信されます）。
- LLDP フレームを受信すると、LLDP のリモート サービスおよび PTOPO MIB は、TLV の情報で更新されます。
- LLDP は次の TLV の特性に対して次のアクションをサポートします。
  - TTL 値の 0 を、送信デバイスの情報を自動的に消去する要求として解釈します。これらのシャットダウン LLDPDU はポートが動作不能になる前に通常送信されます。
  - 不正な形式の必須の TLV の LLDP フレームはドロップされます。
  - 無効な値の TLV は無視されます。
  - TTL がゼロでない場合、不明な組織固有の TLV のコピーがネットワーク管理によるもっと遅いアクセス用に維持されます。

サポートされる LLDP 機能

Cisco ASR 9000 シリーズ ルータは次の LLDP 機能をサポートします。

- IPv4 および IPv6 管理アドレス：一般に、IPv4 と IPv6 アドレスの両方が利用できる場合にアドバタイズされ、環境設定は送信インターフェイスに設定されたアドレスに指定されます。

  送信インターフェイスに設定されたアドレスがない場合、TLV は別のインターフェイスのアドレスで読み込まれます。アドバタイズされた LLDP の IP アドレスが、Cisco ASR 9000 シリーズ ルータのインターフェイスの IP アドレスの次の優先順位に従って実装されます。

- ローカルに設定されたアドレス
- MgmtEth0/RSP0/CPU0/0
- MgmtEth0/RSP0/CPU0/1
- MgmtEth0/RSP1/CPU0/0
- MgmtEth0/RSP1/CPU0/1
- ループバック インターフェイス

（注） LLDP の IPv4 と IPv6 アドレス管理にいくつかの違いがあります。

- IPv4 では、IPv4 アドレスがインターフェイスに設定されていれば、LLDP 管理アドレスとして使用できます。
- IPv6 では、IPv6 アドレスがインターフェイスに設定された後、LLDP 管理アドレスとして使用する前に、インターフェイスのステータスがアップになり、DAD（重複アドレス検出）プロセスに合格する必要があります。

- LLDP は、最も近い物理的に接続された非トンネル ネイバーでサポートされます。
- ポート ID TLV は、イーサネット インターフェイス、サブインターフェイス、バンドル インターフェイス、およびバンドル サブインターフェイスでサポートされます。

サポートされない LLDP 機能

次の LLDP 機能は、Cisco ASR 9000 シリーズ ルータではサポートされていません。

- LLDP-MED の組織上一意の拡張：ただし、この拡張をサポートする他のデバイス間の相互運用性はそのままです。
- トンネリングされたネイバー、または 2 ホップ以上離れたネイバー。
- LLDP TLV はインターフェイスごとにディセーブルにできません。ただし、特定のオプションの TLV はグローバルにディセーブルにできます。
- LLDP SNMP トラップ lldpRemTablesChange。

単方向リンク ルーティング

単方向リンク ルーティング（UDLR）とは、1 つのポートで単方向にトラフィックを送信または受信するための機能です。したがって、全二重ギガビット イーサネットまたは 10 ギガビット イーサネット ポート 1 つに 2 本のファイバを使用する代わりに、UDLR ではファイバが 1 本だけ使用され、設定に応じて単方向のトラフィックを送信または受信します。これによって有効性が向上するだけでなく、既存のファイバ インフラストラクチャの帯域幅を倍に増やすことができます。

Cisco IOS XR ソフトウェアは、単方向リンク ルーティング（UDLR）機能を次のラインカードでサポートします。

- A9K- 24T-TR 24 ポート 10 ギガビット イーサネット ラインカード
- A9K- 24T-SE 24 ポート 10 ギガビット イーサネット ラインカード
- A9K- 36T-TR 36 ポート 10 ギガビット イーサネット ラインカード

- A9K- 36T-SE 36 ポート 10 ギガビット イーサネット ラインカード

UDLR の用途の例としては、ビデオ ストリーミングがあります。このアプリケーションでは、トラフィックの大部分が、確認応答なしの単方向ビデオ ブロードキャスト ストリームとして送信されます。

イーサネットの設定方法

ここでは、次の設定手順について説明します。

- 「イーサネット インターフェイスの設定」（P.43）
- 「LLDP の設定」（P.50）

イーサネット インターフェイスの設定

ここでは、次の設定手順について説明します。

- 「ギガビット イーサネット インターフェイスの設定」（P.43）
- 「L2VPN イーサネット ポートの設定」（P.48）
- 「イーサネット インターフェイスでの MAC アカウンティングの設定」（P.46）

ギガビット イーサネット インターフェイスの設定

基本的なギガビット イーサネット インターフェイスまたは 10 ギガビット イーサネット インターフェイスの設定を作成するには、次の手順で操作します。

手順の概要

1. **show version**
2. **show interfaces** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
3. **configure**
4. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
5. **ipv4 address** *ip-address mask*
6. **flow-control** {**bidirectional** | **egress** | **ingress**}
7. **mtu** *bytes*
8. **mac-address** *value1.value2.value3*
9. **negotiation auto** (on Gigabit Ethernet interfaces only)
10. **no shutdown**
11. **end**
    または
    commit
12. **show interfaces** [**GigabitEthernet** | **TenGigE**] *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**show version** **例：** `RP/0/RSP0/CPU0:router# show version`	（任意）現在のソフトウェア バージョンを表示します。また、ルータがモジュラ サービス カードを認識していることを確認する場合にも使用できます。
ステップ 2	**show interfaces** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show interface TenGigE 0/1/0/0`	（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。 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。 • GigabitEthernet • TenGigE
ステップ 3	**configure** **例：** `RP/0/RSP0/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** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/1/0/0`	インターフェイス コンフィギュレーション モードを開始し、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。 • GigabitEthernet • TenGigE **（注）** この例は、モジュラ サービス カード スロット 1 の 8 ポート 10 ギガビット イーサネット インターフェイスです。
ステップ 5	**ipv4 address** *ip-address mask* **例：** `RP/0/RSP0/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。 – スラッシュ（/）と数字による表記。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。

	コマンドまたはアクション	目的
ステップ 6	**flow-control** {**bidirectional**| **egress**	**ingress**} **例：** `RP/0/RSP0/CPU0:router(config-if)# flow control ingress`	（任意）フロー制御のポーズ フレームの送信および処理をイネーブルにします。 • **egress**：出力でフロー制御のポーズ フレームの送信をイネーブルにします。 • **ingress**：入力で受信したポーズ フレームの処理をイネーブルにします。 • **bidirectional**：出力でフロー制御のポーズ フレームの送信をイネーブルにし、入力で受信したポーズ フレームの処理をイネーブルにします。
ステップ 7	**mtu** *bytes* **例：** `RP/0/RSP0/CPU0:router(config-if)# mtu 1448`	（任意）インターフェイスの MTU 値を設定します。 • 通常フレームのデフォルトは 1514 バイト、802.1Q タグ付きフレームのデフォルトは 1518 バイトです。 • ギガビット イーサネットおよび 10 ギガビット イーサネットの mtu 値の範囲は 64 ～ 65535 バイトです。
ステップ 8	**mac-address** *value1.value2.value3* **例：** `RP/0/RSP0/CPU0:router(config-if)# mac address 0001.2468.ABCD`	（任意）[Management Ethernet] インターフェイスの MAC 層アドレスを設定します。 • 値は、それぞれ MAC アドレスの上位、中間、および下位の 2 バイト（16 進）です。各 2 バイト値の範囲は 0 ～ ffff です。
ステップ 9	**negotiation auto** **例：** `RP/0/RSP0/CPU0:router(config-if)# negotiation auto`	（任意）ギガビット イーサネット インターフェイスのオートネゴシエーションをイネーブルにします。 • オートネゴシエーションは接続の両エンドで明示的にイネーブルにするか、接続の両エンドで速度とデュプレックス設定を手動設定する必要があります。 • オートネゴシエーションがイネーブルの場合、手動で設定する速度またはデュプレックス設定が優先されます。 **（注）** **negotiation auto** コマンドは、ギガビット イーサネット インターフェイスだけで使用できます。
ステップ 10	**no shutdown** **例：** `RP/0/RSP0/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。

	コマンドまたはアクション	目的
ステップ 11	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 12	**show interfaces** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show interfaces TenGigE 0/3/0/0`	（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。

次の作業

- マルチプロトコル ラベル スイッチング（MPLS）や Quality of Service（QoS）など、レイヤ 3 サービス ポリシーをイーサネット インターフェイスに付加する方法については、該当する Cisco ASR 9000 シリーズ ルータのコンフィギュレーション ガイドを参照してください。

イーサネット インターフェイスでの MAC アカウンティングの設定

このタスクでは、イーサネット インターフェイスでの MAC アカウンティングの設定方法について説明します。MAC アカウントには、この手順で説明する特殊な **show** コマンドがあります。show コマンド以外は、基本的なイーサネット インターフェイスの設定と同じなので、手順を 1 回のコンフィギュレーション セッションにまとめることができます。イーサネット インターフェイスの他の一般的なパラメータの設定方法については、このモジュールのギガビット イーサネット インターフェイスの設定を参照してください。

手順の概要

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE** | **fastethernet**] *interface-path-id*
3. **ipv4 address** *ip-address mask*

4. **mac-accounting** {**egress** | **ingress**}
5. **end**
   または
   commit
6. **show mac-accounting** *type* **location** *instance*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/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** | **TenGigE** | **fastethernet**] *interface-path-id* **例：** `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0`	物理インターフェイスまたは仮想インターフェイス。 **（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。 ルータ構文の詳細については、疑問符（**?**）オンライン ヘルプ機能を使用します。
ステップ 3	**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。 – スラッシュ（/）と数字による表記。たとえば、/8 は、マスクの最初の 8 ビットが 1 で、対応するアドレスのビットがネットワーク アドレスであることを示します。
ステップ 4	**mac-accounting** {**egress** | **ingress**} **例：** `RP/0/RP0/CPU0:router(config-if)# mac-accounting egress`	LAN インターフェイス上の発信元 MAC アドレスと宛先 MAC アドレスに基づいて、IP トラフィックのアカウンティング情報を生成します。 • MAC アカウンティングをディセーブルにするには、このコマンドの **no** フォームを使用します。

	コマンドまたはアクション	目的
ステップ 5	**end** または **commit** **例：** `RP/0/RP0/CPU0:router(config-if)# end` または `RP/0/RP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 6	**show mac-accounting** *type* **location** *instance* **例：** `RP/0/RP0/CPU0:router# show mac-accounting TenGigE location 0/2/0/4`	インターフェイスの MAC アカウンティングの統計情報を表示します。

L2VPN イーサネット ポートの設定

L2VPN イーサネット ポートを設定するには、次の手順を実行します。

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

手順の概要

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
3. **l2transport**
4. **l2protocol cpsv** {**tunnel** | **reverse-tunnel**}
5. **end**
   または
   commit
6. **show interfaces** [**GigabitEthernet** | **TenGigE**] *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router#`	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**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/1/0/0`	インターフェイス コンフィギュレーション モードを開始し、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。 • GigabitEthernet • TenGigE
ステップ 3	**l2transport** **例：** `RP/0/RSP0/CPU0:router(config-if)# l2transport`	ポートでレイヤ 2 トランスポート モードをイネーブルにし、レイヤ 2 トランスポート コンフィギュレーション モードを開始します。
ステップ 4	**l2protocol cpsv** {**tunnel** | **reverse-tunnel**} **例：** `RP/0/RSP0/CPU0:router(config-if-l2)# l2protocol cpsv tunnel`	プロトコル CDP、PVST+、STP、VTP のイーサネット インターフェイスでのレイヤ 2 プロトコル トンネリングとプロトコル データ ユニット（PDU）フィルタリングを設定します。 • **tunnel**：インターフェイスに入るときのフレームの L2PT カプセル化と、インターフェイスから出るときのフレームのカプセル化解除を指定します。 • **reverse-tunnel**：インターフェイスから出るときのフレームの L2PT カプセル化と、インターフェイスに入るときのフレームのカプセル化解除を指定します。

	コマンドまたはアクション	目的
ステップ 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if-l2)# end` または `RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 6	**show interfaces** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show interfaces TenGigE 0/3/0/0`	（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。

次の作業

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

- *『Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide』*
- *『Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Command Reference』*

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

LLDP の設定

ここでは、LLDP の次の設定のトピックが含まれます。

- 「LLDP のデフォルト設定」（P.51）
- 「LLDP のグローバルなイネーブル化」（P.51）（必須）
- 「グローバルな LLDP の動作特性の設定」（P.52）（任意）

- 「オプションの LLDP TLV の送信のディセーブル化」（P.54）（任意）
- 「インターフェイスの LLDP 送受信動作のディセーブル化」（P.55）（任意）
- 「LLDP コンフィギュレーションの確認」（P.57）

LLDP のデフォルト設定

表 6 に、Cisco ASR 9000 シリーズ ルータでの LLDP のデフォルトの設定値を示します。デフォルト設定を変更するには、LLDP グローバル コンフィギュレーション コマンドおよび LLDP インターフェイス コンフィギュレーション コマンドを使用します。

表 6　　LLDP のデフォルト設定

LLDP の機能	デフォルト
LLDP グローバル ステート	ディセーブル
LLDP ホールドタイム（廃棄までの時間）	120 秒
LLDP タイマー（パケット更新頻度）	30 秒
LLDP 再初期化遅延	2 秒
LLDP TLV の選択	すべての TLV は送受信に対してイネーブルです。
LLDP インターフェイス ステート	LLDP をグローバルにイネーブルにすると送受信の両方の動作に対してイネーブルになります。

LLDP のグローバルなイネーブル化

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。インターフェイスの LLDP 受信または送信動作を選択的にディセーブルにする方法の詳細については、「インターフェイスの LLDP 送受信動作のディセーブル化」（P.55）を参照してください。

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。

手順の概要

1. **configure**
2. **lldp**
3. **end**
   または
   commit

	コマンド	目的
ステップ 1	**configure** **例：** RP/0/RSP0/CPU0:router# **configure**	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**lldp** **例：** RP/0/RSP0/CPU0:router(config) # **lldp**	システム上の送受信の両方の動作に対してグローバルに LLDP をイネーブルにします。
ステップ 3	**end** または **commit** **例：** RP/0/RSP0/CPU0:router(config) # **end** または RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

グローバルな LLDP の動作特性の設定

「LLDP のデフォルト設定」（P.51）では、LLDP のデフォルト動作特性について説明します。**lldp** コマンドを使用してルータ上で LLDP をグローバルにイネーブルにすると、これらのデフォルトがプロトコルに使用されます。

LLDP ネイバー情報のホールドタイム、初期化遅延、パケット レートなどのグローバルな LLDP 動作特性を変更するには、次の手順を実行します。

手順の概要

1. **configure**
2. **lldp holdtime** *seconds*
3. **lldp reinit** *seconds*
4. **lldp timer** *seconds*

5. **end**
   または
   commit

	コマンド	目的
ステップ 1	**configure** **例：** RP/0/RSP0/CPU0:router# **configure**	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**lldp holdtime** *seconds* **例：** RP/0/RSP0/CPU0:router(config) # **lldp holdtime 60**	（任意）LLDP パケットからの情報をエージングし、削除するまで、その情報を受信デバイスで保持する時間を指定します。
ステップ 3	**lldp reinit** *seconds* **例：** RP/0/RSP0/CPU0:router(config) # **lldp reinit 4**	（任意）インターフェイス上で LLDP の初期化を遅らせる時間を指定します。
ステップ 4	**lldp timer** *seconds* **例：** RP/0/RSP0/CPU0:router(config) # **lldp reinit 60**	（任意）LLDP パケット レートを指定します。
ステップ 5	**end** または **commit** **例：** RP/0/RSP0/CPU0:router(config) # **end** または RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

オプションの LLDP TLV の送信のディセーブル化

特定の TLV は、シャーシ ID、ポート ID、および存続可能時間（TTL）TLV などの LLDP パケットで必須に分類されます。これらの TLV は、すべての LLDP パケットに存在しなければなりません。LLDP パケットでの特定のその他のオプションの TLV の送信を抑制できます。

オプションの LLDP TLV の送信をディセーブルにするには、次の手順を実行します。

手順の概要

1. **configure**
2. **lldp tlv-select** *tlv-name* **disable**
3. **end**
   または
   commit

	コマンド	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router#` **`configure`**	グローバル コンフィギュレーション モードを開始します。

	コマンド	目的
ステップ 2	**lldp tlv-select** *tlv-name* **disable** **例：** RP/0/RSP0/CPU0:router(config) # **lldp tlv-select system-capabilities disable**	（任意）LLDP パケットの指定した TLV の送信がディセーブルであることを指定します。*tlv-name* は、次の LLDP TLV タイプのいずれかにすることができます。 • **management-address** • **port-description** • **system-capabilities** • **system-description** • **system-name**
ステップ 3	**end** または **commit** **例：** RP/0/RSP0/CPU0:router(config) # **end** または RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

インターフェイスの LLDP 送受信動作のディセーブル化

ルータ上で LLDP をグローバルにイネーブルにすると、サポートされているすべてのインターフェイスが LLDP 受信および送信の動作に対して自動的にイネーブルにされます。特定のインターフェイスに対してこれらの動作をディセーブルにして、このデフォルトを上書きできます。

インターフェイスの LLDP 送受信動作をディセーブルにするには、次の手順を実行します。

手順の概要

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
3. **lldp**
4. **receive disable**
5. **transmit disable**

6. **end**
   または
   commit

ステップ 1	**configure** **例：** RP/0/RSP0/CPU0:router# **configure**	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface GigabitEthernet 0/2/0/0** **例：** RP/0/RSP0/CPU0:router(config) # **interface GigabitEthernet 0/2/0/0**	インターフェイス コンフィギュレーション モードを開始し、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。 • GigabitEthernet • TenGigE
ステップ 3	**lldp** **例：** RP/0/RSP0/CPU0:router(config-if)# **lldp**	（任意）指定されたインターフェイスの LLDP 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	**receive disable** **例：** RP/0/RSP0/CPU0:router(config-lldp)# **receive disable**	（任意）インターフェイス上での LLDP 受信動作をディセーブルにします。

ステップ 5	**transmit disable** **例：** RP/0/RSP0/CPU0:router(config-lldp)# **transmit disable**	（任意）インターフェイスの LLDP 送信動作をディセーブルにします。
ステップ 6	**end** または **commit** **例：** RP/0/RSP0/CPU0:router(config)# **end** または RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

LLDP コンフィギュレーションの確認

ここでは、グローバルおよび特定のインターフェイスの LLDP 設定を確認する方法について説明します。

LLDP グローバル設定の確認

LLDP グローバル設定のステータスおよび動作特性を確認するには、次の例に示すよう **show lldp** コマンドを使用します。

```
RP/0/RSP0/CPU0:router# show lldp
Wed Apr 13 06:16:45.510 DST
Global LLDP information:
        Status: ACTIVE
        LLDP advertisements are sent every 30 seconds
        LLDP hold time advertised is 120 seconds
        LLDP interface reinitialisation delay is 2 seconds
```

LLDP がグローバルにイネーブルでない場合、**show lldp** コマンドを実行すると、次の出力が表示されます。

```
RP/0/RSP0/CPU0:router# show lldp
Wed Apr 13 06:42:48.221 DST
% LLDP is not enabled
```

LLDP インターフェイス設定の確認

LLDP インターフェイスのステータスおよび設定を確認するには、次の例に示すように、**show lldp interface** コマンドを使用します。

```
RP/0/RSP0/CPU0:router# show lldp interface GigabitEthernet 0/1/0/7
Wed Apr 13 13:22:30.501 DST


GigabitEthernet0/1/0/7:
        Tx: enabled
        Rx: enabled
        Tx state: IDLE
        Rx state: WAIT FOR FRAME
```

次の作業

システム上で LLDP をモニタして維持したり、LLDP ネイバーに関する情報を入手したりするには、次のいずれかのコマンドを使用します。

コマンド	説明
clear lldp	LLDP トラフィック カウンタまたは LLDP ネイバー情報をリセットします。
show lldp entry	LLDP ネイバーの詳細情報を表示します。
show lldp errors	LLDP エラーおよびオーバーフローの統計情報を表示します。
show lldp neighbors	LLDP ネイバーに関する情報を表示します。
show lldp traffic	LLDP トラフィックの統計情報を表示します。

イーサネットの設定例

ここでは、次の設定例について説明します。

- 「イーサネット インターフェイスの設定：例」（P.58）
- 「MAC アカウンティングの設定：例」（P.59）
- 「レイヤ 2 VPN AC：例」（P.59）
- 「LLDP の設定：例」（P.59）

イーサネット インターフェイスの設定：例

次の例は、10 ギガビット イーサネットのモジュラ サービス カードのインターフェイスを設定する方法です。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/0/0/1
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RSP0/CPU0:router(config-if)# flow-control ingress
RP/0/RSP0/CPU0:router(config-if)# mtu 1448
RP/0/RSP0/CPU0:router(config-if)# mac-address 0001.2468.ABCD
RP/0/RSP0/CPU0:router(config-if)# no shutdown
RP/0/RSP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

```
RP/0/RSP0/CPU0:router# show interfaces TenGigE 0/0/0/1

TenGigE0/0/0/1 is down, line protocol is down
  Hardware is TenGigE, address is 0001.2468.abcd (bia 0001.81a1.6b23)
  Internet address is 172.18.189.38/27
  MTU 1448 bytes, BW 10000000 Kbit
     reliability 0/255, txload Unknown, rxload Unknown
  Encapsulation ARPA,
  Full-duplex, 10000Mb/s, LR
  output flow control is on, input flow control is on
  loopback not set
  ARP type ARPA, ARP timeout 01:00:00
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

MAC アカウンティングの設定：例

次の例は、イーサネット インターフェイスで MAC アカウンティングを設定する方法です。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/0/0/2
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RSP0/CPU0:router(config-if)# mac-accounting egress
RP/0/RSP0/CPU0:router(config-if)# commit
RP/0/RSP0/CPU0:router(config-if)# exit
RP/0/RSP0/CPU0:router(config)# exit
```

レイヤ 2 VPN AC：例

次の例は、イーサネット インターフェイスでレイヤ 2 VPN AC を設定する方法です。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/0/0/2
RP/0/RSP0/CPU0:router(config-if)# l2transport
RP/0/RSP0/CPU0:router(config-if-l2)# l2protocol cpsv tunnel
RP/0/RSP0/CPU0:router(config-if-l2)# commit
```

LLDP の設定：例

次に、ルータ上で LLDP をグローバルにイネーブルにし、デフォルト LLDP 動作特性を変更する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# lldp
RP/0/RSP0/CPU0:router(config)# lldp holdtime 60
RP/0/RSP0/CPU0:router(config)# lldp reinit 4
```

```
RP/0/RSP0/CPU0:router(config)# lldp timer 60
RP/0/RSP0/CPU0:router(config)# commit
```

次に、LLDP 送信のために特定のギガビット イーサネット インターフェイスをディセーブルにする例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/2/0/0
RP/0/RSP0/CPU0:router(config-if)# lldp
RP/0/RSP0/CPU0:router(config-lldp)# transmit disable
```

次の作業

イーサネット インターフェイスの設定が完了したら、イーサネット インターフェイスで各 VLAN サブインターフェイスを設定できます。

シェルフ コントローラ（SC）、ルート プロセッサ（RP）、および分散型 RP のイーサネット管理インターフェイスの変更方法については、このマニュアルで後述する「Cisco ASR 9000 シリーズ ルータでの管理イーサネット インターフェイスの高度な設定および変更」モジュールを参照してください。

IPv6 については、『*Cisco* IOS XR IP Addresses and Services Configuration Guide』の「*Implementing Access Lists and Prefix Lists on Cisco IOS XR Software*」モジュールを参照してください。

その他の関連資料

ここでは、ギガビットおよび 10 ギガビット イーサネット インターフェイスの実装に関する参考資料を紹介します。

関連資料

関連項目	参照先
イーサネット L2VPN	*『Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide』* *『Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Command Reference』*

標準

標準	タイトル
IEEE 802.1ag ITU-T Y.1731	—

MIB

MIB	MIB のリンク
IEEE CFM MIB	Cisco IOS XR ソフトウェアを使用して選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	—

シスコのテクニカル サポート

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

Cisco ASR 9000 シリーズ ルータのイーサネット OAM の設定

このモジュールは、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータのイーサネット運用管理および保守（OAM）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
リリース 3.7.2	次の機能のサポートが追加されました。 • イーサネット リンク OAM • イーサネット CFM
リリース 3.7.3	CFM 探索リンクトレース機能のサポートが追加されました。
リリース 3.9.0	イーサネット SLA 機能のサポートが追加されました。
リリース 3.9.1	次の機能のサポートが追加されました。 • リンク集約グループ（LAG）インターフェイス（イーサネット バンドル インターフェイス）、イーサネットおよびバンドル サブインターフェイス、LAG メンバ（バンドル メンバ）インターフェイス上のイーサネット CFM。 • EFD • AIS • 柔軟なタギング • **ethernet cfm mep domain** コマンドは、**ethernet cfm** および **mep domain** コマンドに置き換えられました。

リリース 4.0.0	次の機能のサポートが追加されました。 • **action link-fault** コマンドは、**action uni-directional link fault** コマンドに置き換えられました。 • **efd** キーワードは、次のコマンドのオプションとして、インターフェイスをライン プロトコルのダウン状態にするために追加されました。 – **action capabilities-conflict** – **action discovery-timeout** – **action session-down** – **action uni-directional link-fault** • ローカル リンク障害を特定し、リモート イーサネット OAM ピアに通知を送信するための単方向リンク障害の検出（**uni-directional link-fault detection** コマンドを使用）。 • イーサネット SLA に次の拡張機能のサポートが追加されました。 – **ethernet sla on-demand operation** コマンドを使用したオンデマンド イーサネット SLA 動作のサポート。 – **statistics measure** コマンドの次の新しいキーワード オプションを使用した一方向遅延およびジッター測定：**one-way-delay-ds**。**one-way-delay-sd**。**one-way-jitter-ds**。**one-way-jitter-sd** – 遅延を測定する場合のループバック パケットをパディングするテスト パターンの指定。 – **show ethernet sla statistics detail** コマンドの測定時間内で統計情報の最小値（Min）および最大値（Max）が得られた時間の表示。
リリース 4.0.1	マルチシャーシ リンク集約グループ（MC-LAG）上のイーサネット CFM のサポートが追加されました。
リリース 4.1.0	次の機能のサポートが追加されました。 • E-LMI • 遅延パケットのタイムスタンプは、システム時刻（NTP）クロックによる取得から RSP のクロック インターフェイスでの DTI のタイミング入力に変更されました。 • CFM Y.1731 ITU キャリア コード（ICC）ベースの MEG ID（MAID）形式。
リリース 4.2.0	単方向リンク検出（UDLD）プロトコルのサポートが導入されました。
リリース 4.3.0	ITU-T Y.1731 合成損失測定のサポートが追加されました。

内容

イーサネット OAM を設定するための前提条件

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。

イーサネット OAM を設定する前に、サポートされるギガビット イーサネット ラインカードの少なくとも 1 つがルータに取り付けられていることを確認してください。

- 2 ポート 10 ギガビット イーサネット、20 ポート ギガビット イーサネットの組み合わせラインカード（A9K-2T20GE-B および A9K-2T20GE-L）
- 4 ポート 10 ギガビット イーサネット ラインカード（A9K-4T-L、-B、または -E）
- 8 ポート 10 ギガビット イーサネット DX ラインカード（A9K-8T/4-L、-B、または -E）
- 8 ポート 10 ギガビット イーサネット ラインカード（A9K-8T-L、-B、または -E）
- 16 ポート 10 ギガビット イーサネット SFP+ ラインカード（A9K-16T/8-B および A9K-16T/8-B+AIP）
- 40 ポート ギガビット イーサネット ラインカード（A9K-40GE-L、-B、または -E）

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」（P.66）
- 「イーサネット CFM」（P.68）
- 「イーサネット SLA」（P.88）
- 「イーサネット LMI」（P.94）
- 「単方向リンク検出プロトコル」（P.97）

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、イベントに対してアクションを実行し、必要に応じて、トラブルシューティングのために特定のインターフェイスをループバック モードにできます。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ます。

- 「ネイバー探索」（P.67）
- 「リンク モニタリング」（P.67）
- 「MIB 取得」（P.67）
- 「誤配線検出（シスコ固有）」（P.67）
- 「リモート ループバック」（P.67）
- 「SNMP トラップ」（P.67）
- 「単方向リンク障害検出」（P.67）

ネイバー探索

ネイバー探索では、リンクの両端で、相手側の OAM 機能を学習し、OAM ピア関係を確立できるようにします。両端でセッションを確立する前に、ピアに特定の機能が必要となる場合もあります。**action capabilities-conflict** または **action discovery-timeout** コマンドを使用して、機能の競合がある場合、または検出プロセスがタイムアウトになった場合に実行する特定のアクションを設定できます。

リンク モニタリング

リンク モニタリングでは、OAM ピアで、リンク品質が時間とともに低下する障害をモニタできます。リンク モニタリングをイネーブルにすると、設定したしきい値を超えた場合にアクションを実行するように OAM ピアを設定できます。

MIB 取得

MIB 取得では、インターフェイスの片側の OAM ピアで、リンクのリモート側から MIB 変数を取得できます。リモート OAM ピアから取得された MIB 変数は読み取り専用です。

誤配線検出（シスコ固有）

誤配線検出はシスコ独自の機能で、可能性のある誤配線のケースを特定するために、すべての情報 OAMPDU の 32 ビットのベンダー フィールドを使用します。

リモート ループバック

リモート ループバックでは、テストのために、リンクの片側で、そのリンクのリモート側をループバック モードにできます。リモート ループバックをイネーブルにすると、リンクのマスター側で開始されたすべてのパケットは、マスター側にループバックされ、リモート（スレーブ）側では変更されません。リモート ループバック モードでは、スレーブ側でパケットにデータを挿入できません。

SNMP トラップ

SNMP トラップは、イーサネット OAM インターフェイスでイネーブルまたはディセーブルにできます。

単方向リンク障害検出

単方向リンク障害検出はイーサネット リンク OAM 機能の 1 つで、リモート ホストにリンク障害をシグナリングするために定義されたリンク障害メッセージを使用する、物理イーサネット インターフェイス（VLAN サブインターフェイスまたはバンドル以外）で直接実行します。単方向リンク障害検出は、ギガビット イーサネットと 10 ギガビット イーサネット ハードウェア レベルのリンク障害のシグナリングと同様の機能ですが、イーサネット リンク OAM の一部として、上位プロトコル レイヤで実行されます。ハードウェア機能は、アウトオブバンドがシグナリングされる、フレームに設定されたリモート障害表示ビットを使用します。この場合、単方向リンク障害検出が、OAMPDU を使用してエラーをシグナリングします。

単方向リンク障害検出は単一の物理リンクだけに適用されます。リモート ホストがリンク障害メッセージを受信すると、そのインターフェイスをすべての上位レイヤ プロトコルでシャットダウンできます。具体的には、レイヤ 2 のスイッチングとレイヤ 3 のルーティング プロトコルです。障害が検出

されている間、リンク障害メッセージがリモート ホストに定期的に送信されます。障害が検出されなくなると、リンク障害メッセージは送信されなくなり、リモート ホストはインターフェイスを元に戻すことができます。

単方向リンク障害検出は、**uni-directional link-fault detection** コマンドを使用して設定します。ルータによるリンク障害メッセージの受信の処理方法に影響することはありません。リンク障害メッセージの受信で実行されるアクションは、**action uni-directional link-fault** コマンドを使用して設定します。

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。

Cisco ASR 9000 シリーズ ルータのイーサネット CFM は、ITU-T Y.1731 の次の機能をサポートします。

- 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。
- ETH-DM、ETH-SLM：これは、イーサネット SLA 機能とともにサポートされます。イーサネット SLA の詳細については、「イーサネット SLA」（P.88）を参照してください。

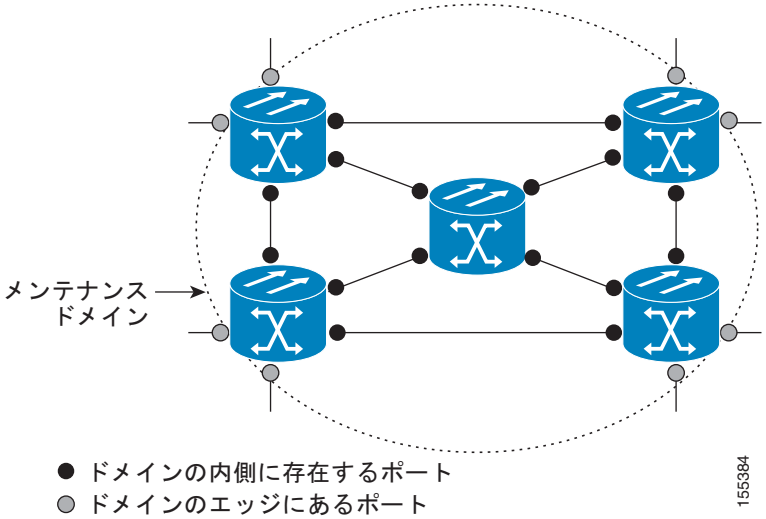
CFM メンテナンス モデルの仕組みを理解するには、次の概念および機能を理解する必要があります。

- 「メンテナンス ドメイン」（P.69）
- 「サービス」（P.71）
- 「メンテナンス ポイント」（P.71）
- 「CFM プロトコル メッセージ」（P.74）
- 「MEP クロスチェック」（P.82）
- 「設定可能なロギング」（P.83）
- 「EFD」（P.83）
- 「CFM の柔軟な VLAN タギング」（P.84）
- 「MC-LAG の CFM」（P.85）

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、図 1 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。

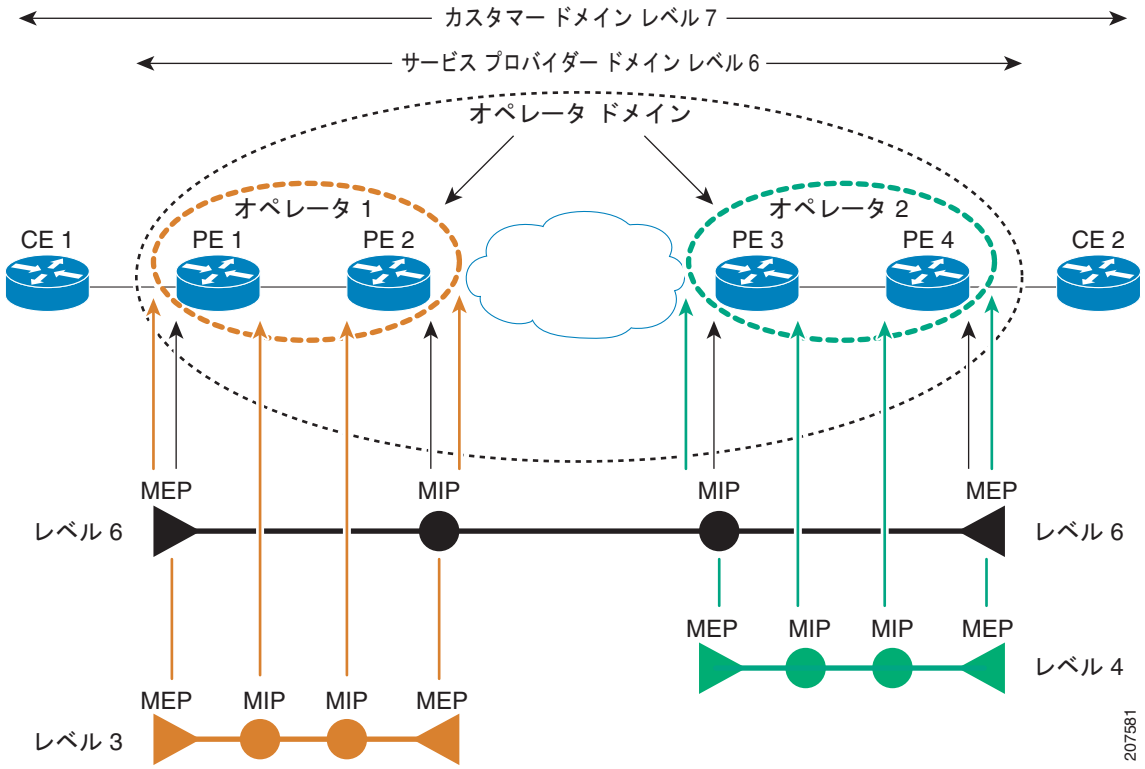
- カスタマーは、ネットワーク全体の接続の確認と管理に CE デバイス間の CFM を使用できます。
- サービス プロバイダーは、提供するサービスの確認と管理に PE デバイス間の CFM を使用できます。
- 各オペレータは、ネットワーク内の接続の確認と管理にオペレータ ネットワーク内の CFM を使用できます。

各組織は別の CFM メンテナンス ドメインを使用します。

図 2 に、ネットワーク内のメンテナンス ドメインの異なるレベルの例を示します。

（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、「メンテナンス ポイント」（P.71）を参照してください。

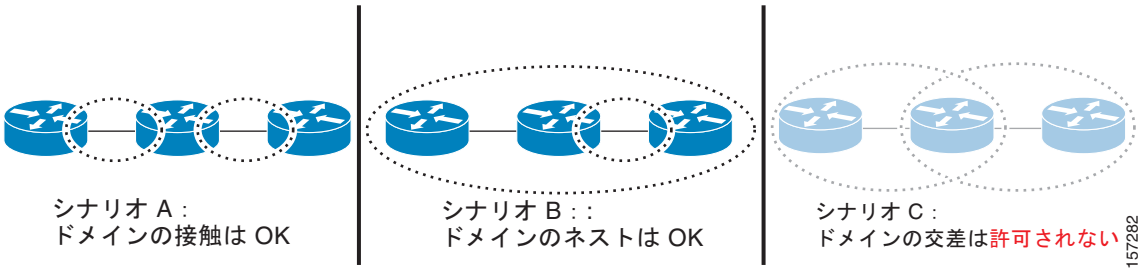
図 2　ネットワーク上の異なる CFM メンテナンス ドメイン



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。図 3 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、およびサポートされていないドメインの交差を示します。

図 3　　サポートされる CFM メンテナンス ドメイン構造



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、「メンテナンス ドメイン」（P.69）で説明するメンテナンス ドメイン階層の実施に役立ち、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（ブリッジ ドメインまたは相互接続の方向）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。

図 4 に、ダウン MEP とアップ MEP のモニタ対象領域について示します。

図 4　ダウン MEP とアップ MEP のモニタ対象領域

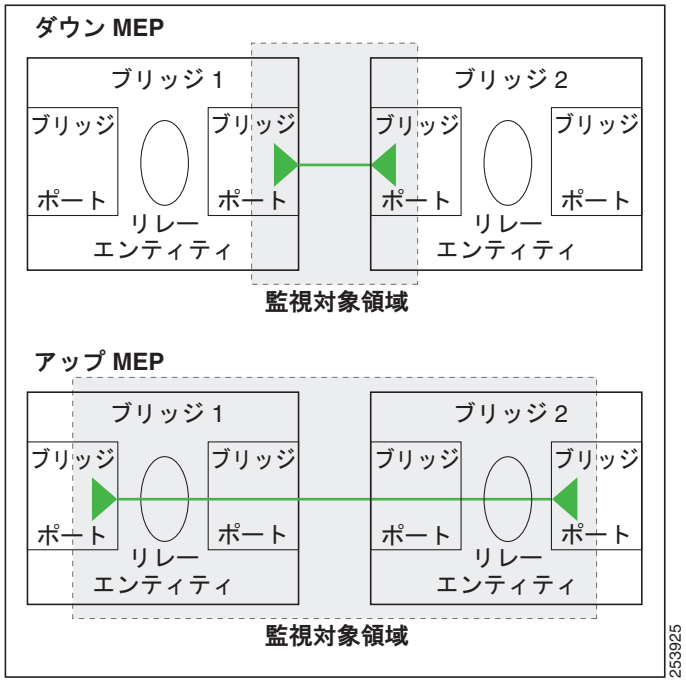
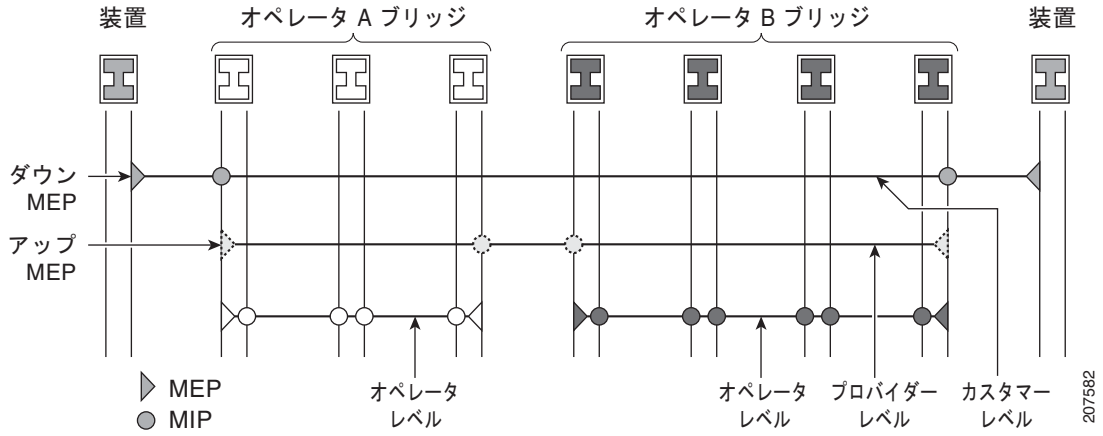


図 5 に、さまざまなレベルのメンテナンス ポイントを示します。ドメインはネストできますが交差できないため（図 3 を参照）、低いレベルの MEP は、より高いレベルの MEP または MIP と常に対応します。また、どのインターフェイスにも MIP を 1 つだけ使用できます。これは通常、MEP がないインターフェイスに存在する最下位ドメインに作成されます。

図 5　さまざまなレベルの CFM メンテナンス ポイント



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。

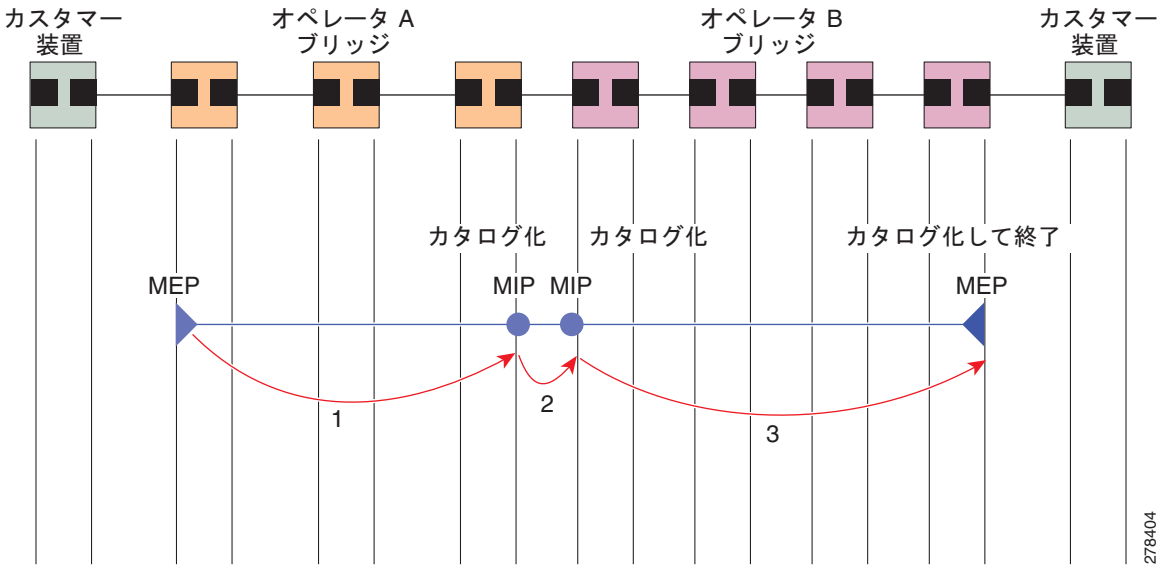
- 「連続性チェック（IEEE 802.1ag と ITU-T Y.1731）」（P.74）
- 「ループバック（IEEE 802.1ag と ITU-T Y.1731）」（P.76）
- 「リンクトレース（IEEE 802.1ag と ITU-T Y.1731）」（P.77）
- 「探索リンクトレース（シスコ）」（P.79）
- 「アラーム表示信号（ITU-T Y.1731）」（P.80）
- 「遅延およびジッター測定（ITU-T Y.1731）」（P.81）
- 「合成損失測定（ITU-T Y.1731）」（P.81）

連続性チェック（IEEE 802.1ag と 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）」（P.77）を参照してください。

図 6　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。

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。
- 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。
- シーケンス番号。
- 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。

- 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。

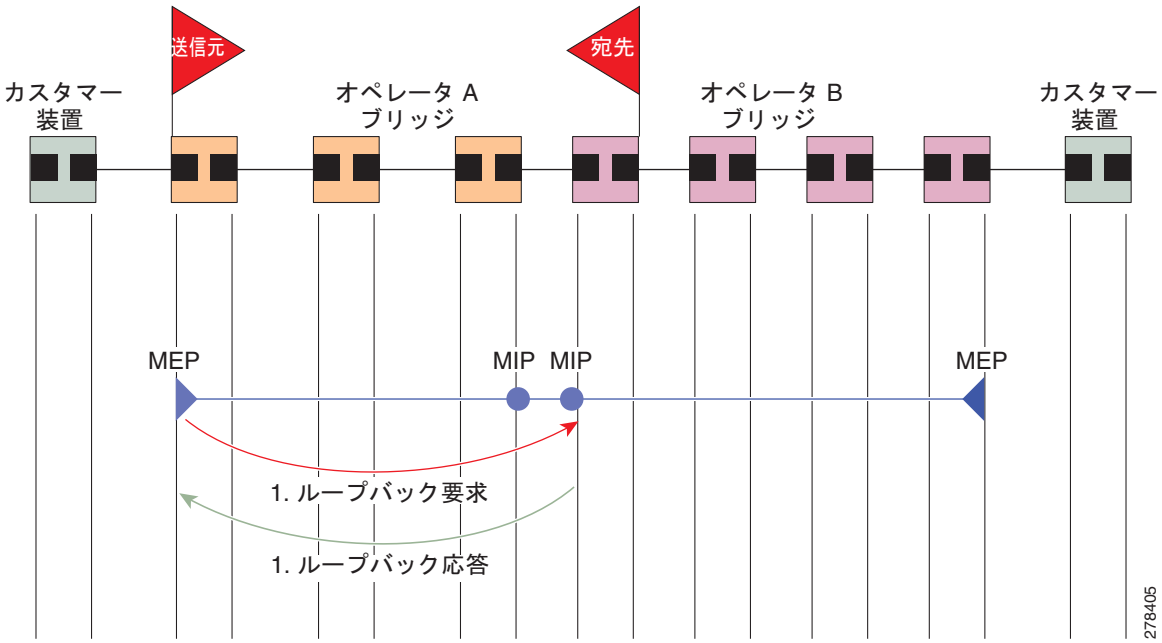
シーケンス外の CCM は、各ピア MEP から受信した CCM のシーケンス番号のモニタリングによっても検出できます。ただし、これは CCM 障害とは見なされません。

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。

図 7 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 7　　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。

一方向遅延およびジッター測定を除き、ループバック メッセージは、ピアが遅延測定をサポートしていない場合イーサネット SLA に使用できます。

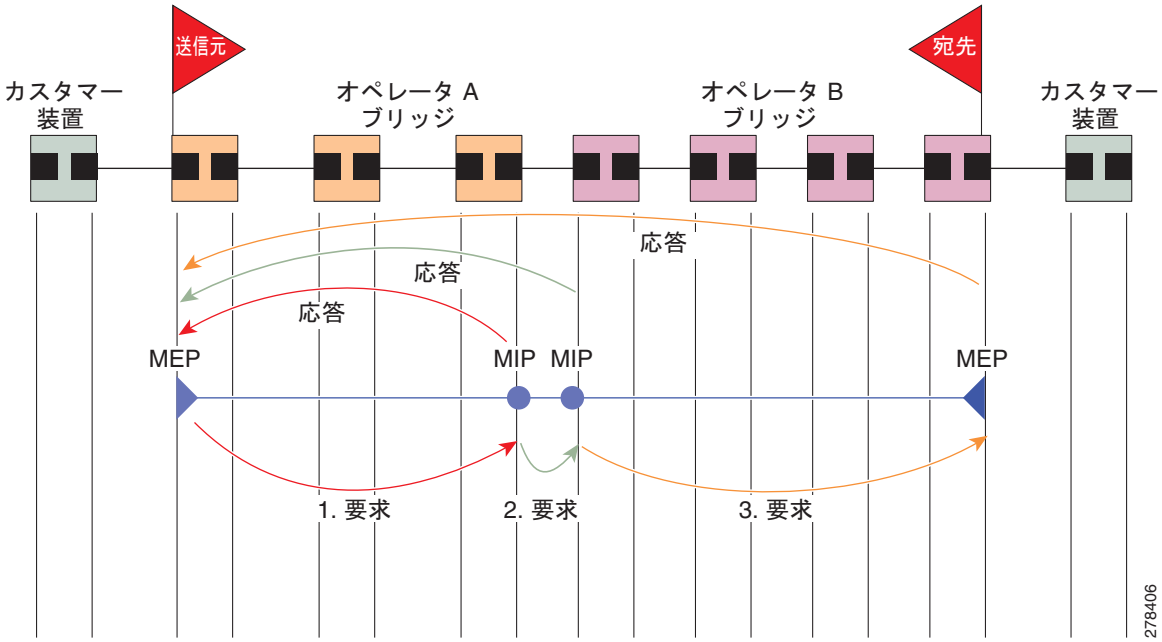
（注） イーサネット CFM ループバック機能は、イーサネット リンク OAM のリモート ループバック機能と混同しないでください（「リモート ループバック」（P.67）を参照）。CFM ループバックは、リモート MP との接続テストに使用され、CFM LBM パケットだけが戻ってきますが、イーサネット リンク OAM リモート ループバックは、通常のサービスから取り出し、すべてのパケットを返すモードに移行することによって、リンクをテストするために使用されます。

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。

図 8 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 8　　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。

探索リンクトレース（シスコ）

探索リンクトレースは前述の標準リンクトレース メカニズムに対するシスコの拡張です。次の 2 つの主な目的があります。

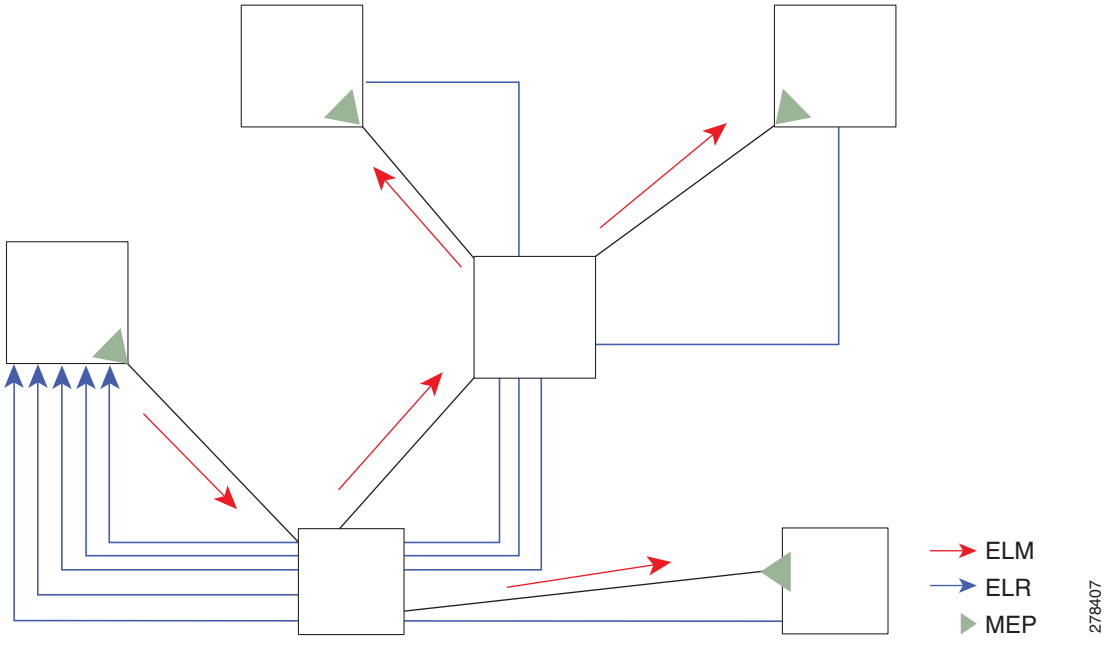
- ネットワーク内に MAC アドレスが以前から存在しないなど、標準リンクトレースが動作しない場合に障害を検出するメカニズムを提供します。たとえば、新しい MEP がプロビジョニングされたが、稼働していない場合、標準のリンクトレースは、新しい MEP からフレームを受信したことがないため、問題の切り分けに役に立ちません。探索リンクトレースでこの問題を解決します。
- 1 つのノードから完全なアクティブ ネットワーク トポロジをマッピングするメカニズムを提供します。これは現在、個別にネットワークの各ノードでトポロジ（たとえば、STP ブロッキング ステート）を検査し、全体のアクティブ トポロジ マップを作成するために手動でこの情報を組み合わせることで実行できるだけです。探索リンクトレースは、これを 1 つのノードから自動的に実行できます。

探索リンクトレースは、ITU-T Y.1731 で定義されたベンダー固有メッセージ（VSM）およびベンダー固有応答（VSR）フレームを使用して実装されます。これらは、ベンダー固有の拡張を相互運用性を低下させずに実装できます。探索リンクトレースは、それらの実装では探索リンクトレース メッセージを無視するだけであるため、他の CFM の実装を含むネットワークで安全に配置できます。

探索リンクトレースは管理者の要求に応じて開始され、ローカル MEP がマルチキャスト探索リンクトレース メッセージを送信することになります。メッセージを受信するネットワークの各 MP は、探索リンクトレース応答を送信します。MIP は受信するメッセージを転送します。開始側 MEP はネットワーク トポロジ全体のツリーを作成するためにすべての応答を使用します。

図 9 に、MEP 間の探索リンクトレース メッセージ フローの例を示します。

図 9　　探索リンクトレース メッセージおよび応答



大規模ネットワークでの応答による発信元 MEP の過負荷を防ぐため、応答側 MP は、応答の送信をランダムな時間遅延させます。その時間はネットワークのサイズが大きいほど長くなります。

大規模なネットワークでは、応答が相当して大量になり、その結果のトポロジ マップも同様に大きくなります。ネットワークの一部だけを対象にする場合、たとえば、問題が小さい領域にすでに狭められているなどの場合、探索リンクトレースを特定の MP で開始するように「指示」できます。応答は、ネットワーク内のそのポイントを越える MP からしか受信されません。それでも応答は発信元 MEP に返されます。

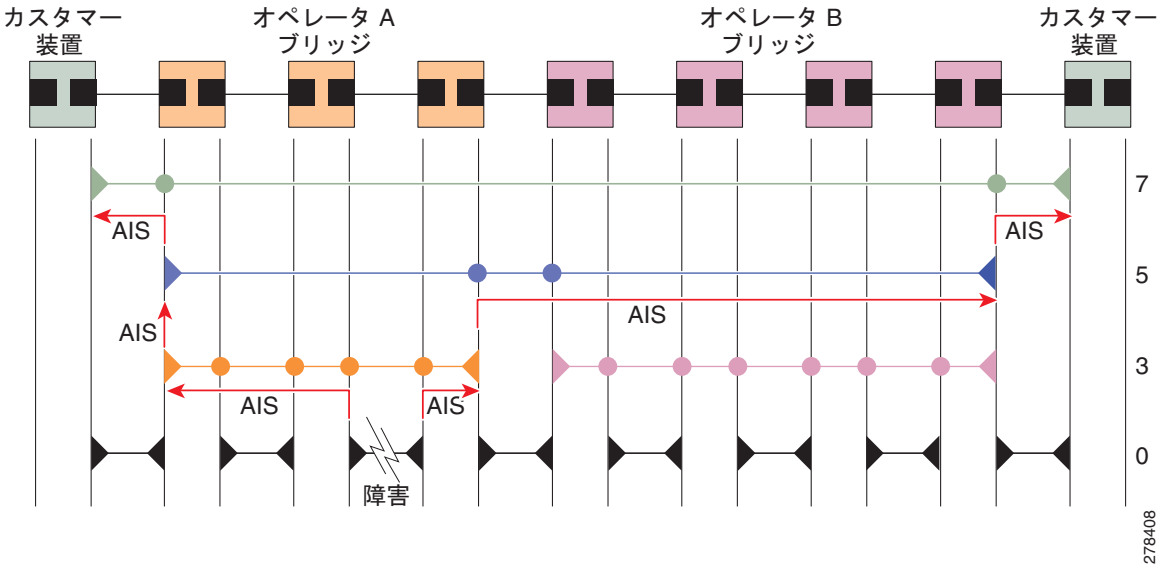
アラーム表示信号（ITU-T Y.1731）

アラーム表示信号（AIS）メッセージは、障害がドメインの途中で検出されると、イベント駆動の方法で迅速に MEP に通知するために使用します。MEP はそれによって、連続するいくつかの CCM を受信できなかったなど、連続性の喪失の検出に頼る場合より、非常に早く障害について学習します。

他のすべての CFM メッセージとは異なり、AIS メッセージはドメインの中間に挿入され、ドメインのエッジの MEP 方向に外に向かって送信されます。通常、AIS メッセージは下位レベルのドメインの MEP によって挿入されます。別の言い方をすれば、MEP による AIS メッセージの送信時に、MEP が送信する他の CFM メッセージとは逆の方向に、MEP 独自のレベルより上のレベルで送信されます。AIS メッセージは、AIS を送信する MEP と同じドメインのピア MEP ではなく、上位レベルのドメインの MEP によって受信されます。MEP は、AIS メッセージを受信すると、自身で別の AIS メッセージをさらに上位レベルで送信できます。

図 10 に、AIS メッセージ フローの例を示します。メンテナンス ドメイン レベルは、図の右側に番号を付けています。

図 10　　AIS メッセージ フロー



AIS はポイントツーポイント ネットワークだけに適用されます。冗長パスがあるマルチポイント ネットワークでは、ネットワークが障害リンクを迂回してルーティングするため、再コンバージェンスすることがあるように、下位レベルの障害が必ずしも上位レベルで障害になるとは限りません。

AIS メッセージは、MEP によって通常送信されます。ただし、インターフェイスがダウンするなど、障害が基本的な転送で検出された場合は、MEP が存在しないときにも AIS メッセージを送信できます。ITU-T Y.1731 でこれらは *サーバ MEP* と呼ばれます。

AIS メッセージは、複数の障害状況に応じて送信されます。

- CCM 障害の検出（「連続性チェック（IEEE 802.1ag と ITU-T Y.1731）」（P.74）で説明）。
- 連続性の喪失。

- AIS メッセージの受信。
- インターフェイスがダウンしている場合など、基本的な転送の障害。

受信した AIS メッセージは、連続性の喪失を待機するよりも速く、障害を検出して対処するために使用できます。障害が下位レベルですでに検出され、そこで処理されるという前提で、障害アクションを抑制するためにも使用できます。これは、ITU-T Y.1731 で説明されています。ただし、多くの場合、前者の方が有用です。

遅延およびジッター測定（ITU-T Y.1731）

ルータは、次の 2 つのパケット タイプを使用した一方向および双方向の遅延測定をサポートします。

- 遅延測定メッセージ（DMM）
- 遅延測定応答（DMR）

これらのパケットはループバック メッセージと同じようなユニキャストです。パケットは、より正確な遅延測定をサポートするため、システムの時刻クロックによって生成されたタイムスタンプを伝送し、SLA の管理性のフロントエンドもサポートします。Cisco IOS XR Release 4.1 からは、DDM および DDR パケットが RSP のクロック インターフェイス ポートの DTI タイミング入力から取得したタイムスタンプを伝送します。

ただし、ループバック メッセージとは異なり、これらのメッセージ タイプは、宛先から送信元に、または送信元から宛先に一方向の遅延とジッターを測定することもできます。

SLA の詳細については、「イーサネット SLA」（P.88）を参照してください。

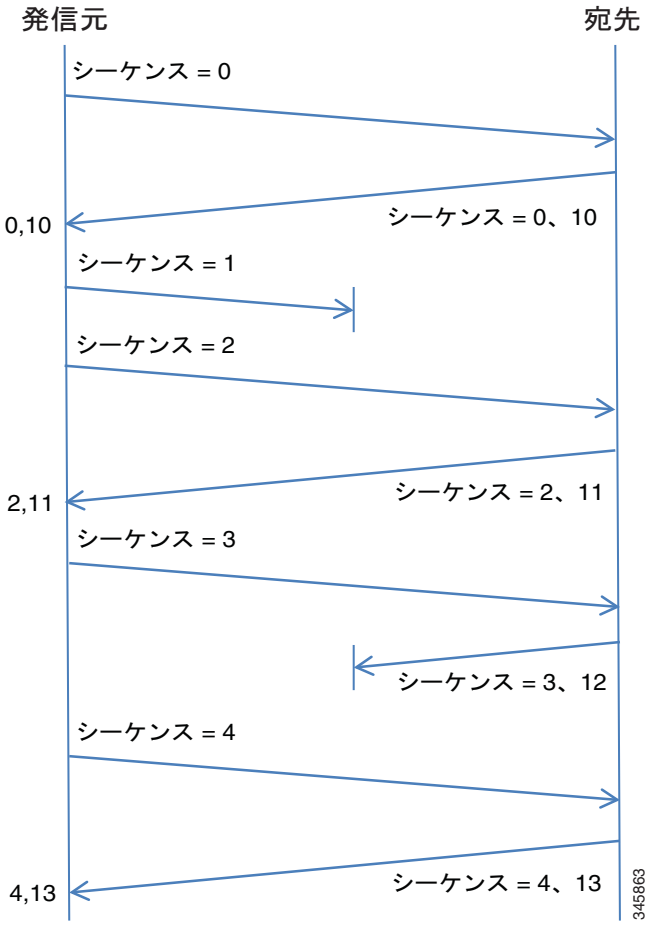
合成損失測定（ITU-T Y.1731）

合成損失測定（SLM）とは、合成測定プローブを挿入してそのプローブの損失を測定するメカニズムです。この目的は、実際のデータ トラフィックの損失を測定することです。各プローブ パケットは 1 つのシーケンス番号を伝送します。送信側はパケットを 1 つ送信するたびにこのシーケンス番号に 1 を加算するので、受信側は、シーケンス番号が欠落しているかどうかを調べるとパケットの損失を検出できます。

SLM パケットには 2 つのシーケンス番号が格納されています。1 つは発信側が SLM に書き込んで応答側が SLR にコピーするものであり、もう 1 つは応答側によって割り当てられて SLR に書き込まれます。前者を「送信元から宛先（sd）シーケンス番号」と呼び、後者を「宛先から送信元（ds）シーケンス番号」と呼びます。

図 11 の例は、各方向のフレーム損失率（FLR）の計算にシーケンス番号がどのように使用されるかを示しています。

図 11　合成損失測定



MEP クロスチェック

MEP クロスチェックでは、認識されていた MEP のいずれかが失われた場合、または予定したグループに存在しない追加のピア MEP が検出された場合にエラーを検出できるように、一連の予想されるピア MEP の設定がサポートされます。

サービス内の予想される MEP ID のセットは、ユーザが定義します。オプションで、対応する MAC アドレスも指定できます。CFM は、CCM の受信元になっている一連のピア MEP をモニタします。予想される指定のピア MEP のいずれからも CCM を受信していない、または連続性の喪失が検出された場合に、クロスチェックの「欠落」の障害が検出されます。同様に、CCM を一致した MEP ID から受信したが、間違った送信元 MAC アドレスの場合、クロスチェックの「欠落」の障害が検出されます。予想される MEP ID と一致する（さらに、指定した場合は予期される MAC アドレスとも一致する）CCM をそれ以降受信すると、障害がクリアされます。

（注）　連続性の喪失はどのピア MEP でも検出できますが、クロスチェックが設定されている場合にのみ、障害状態として扱われます。

クロスチェックが設定され、予期しない MEP ID を持つピア MEP から CCM を受信した場合、これは、クロスチェックの「予定外」の状態として検出されます。ただし、これは、障害状態として扱われません。

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「欠落」または「予定外」の状態が検出される。
- 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。

EFD

イーサネット障害検出（EFD）は、CFM などのイーサネット OAM プロトコルが、インターフェイスの「ライン プロトコル」ステートの制御を可能にするためのメカニズムです。

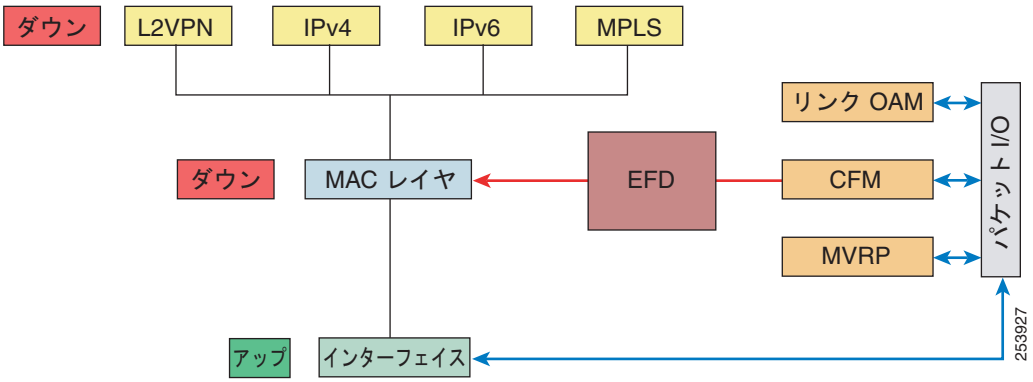
他の多くのインターフェイス タイプとは異なり、イーサネット インターフェイスにライン プロトコルはありません。ライン プロトコルのステートはインターフェイスのステートから独立しています。イーサネット インターフェイスの場合、このロールは、物理層のイーサネット プロトコル自体で処理されるため、インターフェイスが物理的にアップしている場合に使用可能であり、トラフィックが通過できます。

EFD は、CFM がイーサネット インターフェイスのライン プロトコルとして機能できるように、これを変更します。これで、CFM 障害（AIS や連続性の喪失など）が予期されたピア MEP により検出された場合、インターフェイスをシャットダウンできるように CFM でインターフェイス ステートを制御できます。これにより、トラフィック フローを停止するだけでなく、問題を避けてルーティングするために、上位レベルのプロトコルのアクションをトリガーします。たとえば、レイヤ 2 インターフェイスの場合は、MAC テーブルがクリアされ、MSTP は再コンバージェンスされます。レイヤ 3 インターフェイスの場合は、ARP キャッシュがクリアされ、IGP が再コンバージェンスされます。

（注） EFD はダウン MEP にしか使用できません。EFD を使用してインターフェイスをシャット ダウンした場合、CFM フレームはフローを続けます。これにより、CFM で問題が解決されたタイミングを検出できるため、インターフェイスを自動的に元に戻します。

図 12 に、インターフェイスに対応する MAC レイヤにエラーを EFD シグナリングするセッションの 1 つでの CFM のエラー検出を示します。これにより、MAC はダウン状態になり、さらにすべての上位レベルのプロトコル（レイヤ 2 疑似回線、IP プロトコルなど）のダウンと、可能な場所での再コンバージェンスも引き起こします。CFM がエラーがなくなったことを検出するとすぐに、EFD へのシグナリングが可能になり、すべてのプロトコルが再びアクティブになります。

図 12　　CFM エラー検出および EFD トリガー



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。

図 13 に、CFM を使用した複数の VLAN を使用するネットワークの例を示します。

図 13　複数の VLAN と CFM のサービス プロバイダー ネットワーク

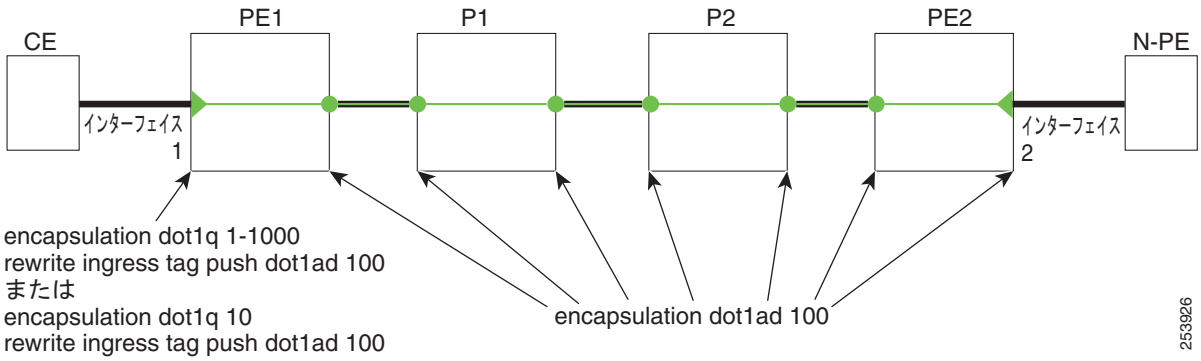


図 13 に、S-VLAN タグがサービス デリミタとして使用される、プロバイダーのアクセス ネットワークを示します。PE1 はカスタマーと対し、PE2 はコア方向のアクセス ネットワークのエッジにあります。N:1 バンドルを使用するので、C-VLAN タグの範囲にインターフェイスのカプセル化が一致します。これは潜在的に全範囲であり、総数:1 バンドルになります。単一 C-VLAN のみを一致させる使用例もありますが、それでも S-VLAN はサービス デリミタとして使用されます。これは、IEEE モデルにより沿ったものですが、プロバイダーは 4094 個のサービスに制限されます。

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、ドメイン サービスの特定の番号にタグを送信できます。現在、タグは 1 に設定できるだけです。

MC-LAG の CFM

マルチシャーシ リンク集約グループの CFM は次の一般的なネットワーク環境の Cisco ASR 9000 シリーズ ルータでサポートされます。

- カスタマー エッジ（CE）デバイスは、2 台のプロバイダー エッジ（PE）接続ポイント（POA）デバイスに接続されたデュアルホーム接続デバイスです。ただし、デュアルホーム接続デバイスは複数の PE への接続を認識せずに動作します。
- PE の 2 つの接続ポイントは冗長グループ（RG）を形成し、1 つの POA をアクティブ POA として機能させ、その他はデュアルホーム接続リンクのスタンバイ POA として機能させます。
- 一般的なフェールオーバーのシナリオと同様に、障害がアクティブ POA で発生した場合、ネットワークへのデュアルホーム接続デバイスの接続を保持するため、スタンバイ POA が引き継ぎます。

MC-LAG の CFM のサポートは 2 つのレベルで条件付けられます。

- RG レベルの CFM : CFM コンテキストは、冗長グループ単位であり、RG 全体の接続を確認します。
- POA レベルの CFM : CFM コンテキストは接続ポイント単位であり、単一 POA への接続を確認します。

CFM サポートの両方のレベルでは、正しく実装するために考慮する必要がある特定の制限と設定時の注意事項があります。

この項では、次のトピックについて取り上げます。

- 「RG レベル CFM」（P.86）
- 「POA レベル CFM」（P.87）
- 「MC-LAG の CFM でサポートされる機能」（P.87）
- 「MC-LAG の CFM の制限」（P.88）

Cisco ASR 9000 シリーズ ルータの LAG と MC-LAG の詳細については、このマニュアルの「Cisco ASR 9000 シリーズ ルータ でのリンク バンドルの設定」の章を参照してください。

RG レベル CFM

RG レベル CFM は、次のモニタリングの 3 つのエリアで構成されています。

- 「RG ダウンリンク モニタリング」（P.86）
- 「RG アップリンク モニタリング」（P.86）
- 「エンドツーエンド サービス モニタリング」（P.87）

RG ダウンリンク モニタリング

RG ダウンリンク モニタリングはデュアルホーム接続デバイスと RG 間の接続を確認するために CFM を使用します。

RG ダウンリンク モニタリングを設定するには、次の要件を満たしていることを確認します。

- ダウン MEP がバンドルで設定されている。
- 各 POA のダウン MEP が同じ MEP ID および送信元 MAC アドレスを使用して、同じように設定されている。

この設定には次の制限があります。

- 現在サポートされている最短 CCM 間隔が 100 ms であるため、CCM 損失時間がフェールオーバー時間（通常は 50 ms）より大きくなり、最短 CCM 損失時間は 350 ms になります。

RG アップリンク モニタリング

RG アップリンク モニタリングがアクティブ POA からコアへの接続を確認するために CFM を使用します。

RG アップリンク モニタリングを設定するには、次の要件を満たしていることを確認します。

- アップ MEP が各 POA のバンドル インターフェイスまたはバンドル サブインターフェイスで設定されている。
- 各 POA のアップ MEP は同じ MEP ID および送信元 MAC アドレスを使用して、同じように設定されている。

エンドツーエンド サービス モニタリング

エンドツーエンド サービス モニタリングは、デュアルホーム接続デバイス間のエンドツーエンド サービスを確認するために CFM を使用します。

エンドツーエンド サービス モニタリングを設定するには、次の要件を満たしていることを確認します。

- ダウン MEP がデュアルホーム接続デバイスのバンドル インターフェイスまたはバンドル サブインターフェイスで設定されている。
- 任意の MIP が設定されている場合、各 POA がバンドルの MIP で設定されている。
- 各 POA にアップリンク インターフェイスで MIP を設定できる（ネイティブ イーサネットが使用されている場合）。
- アクティブおよびスタンバイ POA が同じように設定されている。

この設定には次の制限があります。

- スタンバイ POA の MIP はループバックやリンクトレース要求に応答しません。

POA レベル CFM

POA レベル モニタリングはデュアルホーム接続デバイスと単一 POA 間の接続を確認するために CFM を使用します。

POA レベルの CFM を設定するには、次の要件を満たしていることを確認します。

- ダウン MEP がバンドルのメンバだけで設定されている。

この設定には次の制限があります。

- POA レベル モニタリングは、単一 POA とコア間のアップリンクではサポートされません。

MC-LAG の CFM でサポートされる機能

MC-LAG の CFM は次の CFM 機能をサポートします。

- Cisco ASR 9000 シリーズ ルータの既存のすべての IEEE 802.1ag および Y.1731 機能が MC-LAG RG でサポートされます。
- CFM メンテナンス ポイントは MC-LAG インターフェイスでサポートされます。スタンバイ リンクのメンテナンス ポイントがスタンバイ状態になります。
- スタンバイ状態のメンテナンス ポイントは、CFM メッセージを受信しますが、どの CFM メッセージに対しても送信または応答しません。
- MEP がアクティブからスタンバイに移行すると、CCM 障害およびアラームはすべてクリアされます。
- スタンバイ MEP は、リモート MEP エラーとタイムアウトを記録しますが、障害を報告しません。これは、**show** コマンドでリモート MEP およびそのエラーが表示されますが、ログ、アラーム、MIB トラップ、または EFD はトリガーされず、AIS メッセージは送信されないことを意味します。
- MEP がスタンバイからアクティブに移行すると、MEP がスタンバイであった間にすでに検出された CCM 障害があれば再適用され、ただちに処理が実行されます（ログ、アラーム、MIB トラップ、EFD など）。
- MC-LAG の CFM では、Cisco ASR 9000 シリーズ ルータでサポートされるバンドル インターフェイスに対して同じスケールをサポートします。

MC-LAG の CFM の制限

MC-LAG の CFM をサポートするには、次の制限および要件を考慮する必要があります。

- CFM 設定は、アクティブおよびスタンバイ POA の両方で同じでなければなりません。
- CFM 状態は 2 つの POA 間で同期されません。これは、EFD が設定された場合、POA フェールオーバーでインターフェイス ライン プロトコル ステートのフラッピングの原因になる可能性があります。障害アラームは、障害が検出された直後にフェールオーバーが発生すると遅延する場合もあります。
- POA レベルの CFM モニタリングは、ネイティブ イーサネット アップリンク インターフェイスではサポートされません。
- レベル 0 のバンドル インターフェイスの MEP はサポートされません。
- ループバック、リンクトレースおよび Y.1731 SLA 動作はスタンバイ状態の MEP から開始できません。
- POA が同じ設定になるようにするための、MEP ID 設定の一貫性のチェックはサポートされません。
- Y.1731 SLA 統計情報は、2 つの POA 間でフェールオーバーが発生すると分割できます。外部ネットワーク管理システムでは、2 つの POA からこれらの統計情報を収集し、成形する必要があります。

イーサネット SLA

カスタマーはサービス プロバイダーがサービス レベル契約（SLA）に従うよう求めています。このため、サービス プロバイダーは、そのネットワークのパフォーマンス特性をモニタできる必要があります。同様に、カスタマーがそのネットワークのパフォーマンス特性をモニタすることも必要です。シスコは、シスコのイーサネット SLA 機能を使用して Y.1731 パフォーマンス モニタリングを提供します。

SLA はサービス プロバイダー ネットワークを使用するカスタマーに対するサービスの最低レベルを保証する一連の基準を定義します。基準では、遅延、ジッター、フレーム損失とアベイラビリティなど、多くのさまざまな領域をカバーできます。

シスコのイーサネット SLA 機能は次の規格に準拠しています。

- IEEE 802.1ag
- ITU-T Y.1731

シスコのイーサネット SLA 機能はレイヤ 2 でネットワークをモニタするアーキテクチャを提供します。このアーキテクチャは、SLA 統計情報の収集、保存、表示、および分析などの機能を提供します。これらの SLA 統計情報はさまざまな方法で保存および表示でき、統計情報の分析が実行できるようになります。

イーサネット SLA はパフォーマンス モニタリングの次の主要な機能を実行するためのフレームワークを提供します。

- 1 つまたは複数のパケットで構成されるプローブをパフォーマンスの測定のために送信する

  イーサネット SLA は、パフォーマンスの測定用に SLA プローブを送信するための柔軟なメカニズムを提供します。プローブは CFM ループバックまたは CFM 遅延測定パケットのいずれかで構成できます。パケットの送信頻度の変更、およびサイズ、プライオリティなどのプローブ パケットの属性を指定するためのオプションが使用できます。

- 定期的なプローブで構成される動作のスケジューリング。

各プローブを実行すべき頻度、維持時間、最初のプローブを開始すべきタイミングを指定するための柔軟なメカニズムはイーサネット SLA によって提供されます。プローブは、バックツーバックを実行して連続的な測定を提供するようにスケジュールしたり、1 分に 1 回から週 1 回までの範囲で定義された間隔でスケジュールしたりできます。

- 結果の収集と保存。

  イーサネット SLA は、測定プローブごとに収集、保存する必要のあるパフォーマンス パラメータを指定する柔軟性を提供します。パフォーマンス パラメータは、フレームの遅延およびジッター（フレーム間の遅延変動）が含まれます。各パフォーマンス パラメータについて、個々の結果をそれぞれ保存するか、または特定の範囲内に分類された結果数のカウンタの保存によって結果を集約できます。設定可能な量の履歴データを、最新結果の他に保存できます。

- 結果の分析と表示。

  イーサネット SLA は、最小偏差、最大偏差、平均偏差および標準偏差の計算などの収集結果の基本統計分析を実行します。また、プローブ パケットのいずれかが失われたか、順序に誤りがあるか、結果にパフォーマンスが正しく反映されていない原因があるかを記録します（たとえば、測定が行われている間にローカル時刻クロックの大幅なずれが検出された場合など）。

Y.1731 パフォーマンス モニタリング

ITU-T Y.1731 標準では、キャリア イーサネット ネットワークのパフォーマンス モニタリングに使用できるさまざまなメカニズムが定義されています。この標準で定義された測定メカニズムは次のとおりです。

遅延測定：タイムスタンプが格納された CFM フレームを交換することによって、フレーム遅延を正確に測定できます。連続する遅延測定値を比較すると、フレーム間遅延変動（ジッター）を測定できます。遅延測定メッセージを使用すると、次の測定を実行できます。

- ラウンドトリップ時間
- ラウンドトリップ ジッター
- 一方向遅延（SD と DS の両方）
- 一方向ジッター（SD と DS の両方）
- SLA プローブ パケット破損数
- 順序不正 SLA プローブ パケット数
- SLA プローブ パケット損失

損失測定：送信 / 受信フレーム カウンタが格納された CFM フレームを交換することによって、データ トラフィックの損失を正確に測定できます。また、高損失の期間をトラッキングすることによって、アベイラビリティを測定できます。損失測定メッセージを使用すると、次の測定を実行できます。

- SLA プローブ パケット破損数
- 順序不正 SLA プローブ パケット数
- SLA プローブ パケット損失
- データ パケット損失

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。その後で、統計分析を使用してデータ トラフィック損失の概算を求めます。この手法を「合成損失測定」と呼びます。これは、Y.1731 標準の最新バージョンに組み込まれています。合成損失測定メッセージを使用すると、次の測定を実行できます。

- 一方向損失（送信元から宛先）
- 一方向損失（宛先から送信元）

ループバック：これは、パフォーマンス モニタリングを主な目的とするものではありませんが、ラウンドトリップの遅延およびジッターの概算値を求めることができます。たとえば、ピア デバイスが遅延測定をサポートしていない場合に使用できます。ループバック メッセージを使用すると、次の測定を実行できます。

- ラウンドトリップ時間
- ラウンドトリップ ジッター
- SLA プローブ パケット破損数
- 順序不正 SLA プローブ パケット数
- SLA プローブ パケット損失

イーサネット SLA の概念

正常にシスコのイーサネット SLA 機能を設定するには、次の概念を理解する必要があります。

- 「イーサネット SLA 統計情報」（P.90）
- 「イーサネット SLA 測定パケット」（P.91）
- 「イーサネット SLA のサンプル」（P.91）
- 「イーサネット SLA プローブ」（P.92）
- 「イーサネット SLA バースト」（P.92）
- 「イーサネット SLA スケジュール」（P.92）
- 「イーサネット SLA バケット」（P.92）
- 「イーサネット SLA 集約ビン」（P.92）
- 「イーサネット SLA 動作プロファイル」（P.92）
- 「イーサネット SLA 動作」（P.93）
- 「イーサネット SLA オンデマンド動作」（P.93）

イーサネット SLA 統計情報

イーサネット SLA の*統計情報*は、単一のパフォーマンス パラメータです。次の統計情報をイーサネット SLA で測定できます。

- ラウンドトリップ遅延
- ラウンドトリップ ジッター
- 送信元から宛先への一方向遅延
- 送信元から宛先への一方向ジッター
- 送信元から宛先への一方向フレーム損失

- 宛先から送信元への一方向遅延
- 宛先から送信元への一方向ジッター
- 宛先から送信元への一方向フレーム損失

（注） すべてのタイプのパケットで、すべての統計情報が測定できるわけではありません。たとえば、一方向の統計情報は CFM ループバック パケットを使用する場合は測定できません。

イーサネット SLA 測定パケット

イーサネット SLA *測定パケット*は、SLA 測定を行うためにネットワークで送信される単一のプロトコル メッセージと対応する応答です。次のタイプの測定パケットがサポートされます。

- CFM 遅延測定（Y.1731 DMM/DMR パケット）: CFM 遅延測定パケットには、フレームの遅延およびジッターを正確に測定するために使用できるパケット データ内のタイムスタンプが含まれます。これらのパケットはラウンドトリップまたは一方向の統計情報の測定に使用できます。ただし、DMM/DMR パケット サイズは変更できません。

（注） Cisco IOS XR Release 4.3.0 以降では、Y.1731 DMM v1 フレームを使用するようにイーサネット SLA プロファイルを設定できます。CFM MEP ごとに設定できるイーサネット SLA 動作は 150 個までという制約は解除されました。これは、DMM フレームを使用するプロファイルだけでなく、サポートされるその他の Y.1731 フレーム タイプを使用するプロファイル（たとえばループバック測定や合成損失測定）も該当します。相互運用性を目的として、DMM v0 フレームを使用するように動作を設定することは引き続き可能です。このようにするには、**ethernet SLA profile** コマンドでタイプとして **cfm-delay-measurement-v0** を指定します。この場合は、CFM MEP ごとに設定できる動作は 150 個までという制約が引き続き適用されます。

- CFM ループバック（LBM/LBR）: CFM ループバック パケットは正確さには欠けますが、ピア デバイスが DMM/DMR パケットをサポートしない場合に使用できます。これらのパケットはタイムスタンプが含まれていないため、ラウンドトリップ統計情報だけを測定できます。ただし、ループバック パケットはパディングできるため、測定は特定のサイズのフレームを使用して行うことができます。
- CFM 合成損失測定（Y.1731 SLM/SLR パケット）: SLM パケットには 2 つのシーケンス番号が格納されています。1 つは発信側が SLM に書き込んで応答側が SLR にコピーするものであり、もう 1 つは応答側によって割り当てられて SLR に書き込まれます。前者を「送信元から宛先（sd）シーケンス番号」と呼び、後者を「宛先から送信元（ds）シーケンス番号」と呼びます。

（注） SLM は統計的なサンプリング手法であるため、測定値と実際の損失値との間に多少の相違が存在する可能性があります。測定の精度を高めるには、各 FLR 計算あたりの SLM パケット数を増やします。

イーサネット SLA のサンプル

*サンプル*は、特定の統計情報に関する単一の結果（数値）です。ラウンドトリップ遅延などの一部の統計情報の場合、サンプルは、単一の測定パケットを使用して測定できます。ジッターなど他の統計情報の場合、サンプルを取得するには、2 つの測定パケットが必要です。

イーサネット SLA プローブ

*プローブ*は、一連の特定の統計情報の SLA サンプルの収集に使用される測定パケットのシーケンスです。プローブの測定パケットは特定のタイプで（たとえば、CFM 遅延測定または CFM ループバック）、フレーム サイズ、プライオリティなどの特定の属性を持ちます。

（注） 1 つのプローブは同じ測定パケットを使用して異なる統計情報のデータを、同時に収集できます（たとえば、一方向遅延およびラウンドトリップ ジッター）。

イーサネット SLA バースト

プローブでは、測定パケットはバーストで、または個別に送信できます。*バースト*は短時間内で別々に送信された 2 つ以上のパケットが含まれます。バーストはそれぞれ 1 分まで継続でき、プローブ内の継続的な測定を行うため、バースト同士が相互に間を空けず流れます。

サンプルごとに 2 個の測定パケットが必要な統計情報（ジッターなど）の場合、サンプルは同じバーストの測定パケットだけに基づいて計算されます。すべての統計情報については、バーストを使用すると、個別のパケットを送信するよりも効率的です。

イーサネット SLA スケジュール

イーサネット SLA *スケジュール*は、プローブの送信頻度、各プローブの存続期間、および最初のプローブの開始時刻を示します。

イーサネット SLA バケット

特定の統計情報の場合、*バケット*は、一定の時間中に収集される結果の集合です。バケットによって示された時間内に開始された測定のサンプルはすべてそのバケットに格納されます。バケットでは、さまざまな期間の結果が比較できます（ピーク トラフィックとオフピーク トラフィックなど）。

デフォルトでは、個別のバケットは各プローブに対して作成されます。つまり、バケットはプローブの開始と同じ開始時刻と、プローブの継続時間を表します。したがってバケットはそのプローブによって行われた測定に関するすべての結果が含まれます。

イーサネット SLA 集約ビン

各サンプルをバケット内に個別に保存するのではなく、その代わりに、ビンにサンプルを集約します。*集約ビン*は、サンプル値の範囲で、その範囲内に分布する受信サンプル数のカウンタが含まれています。ビンのセットは、ヒストグラムを形成します。集約がイネーブルの場合、各バケットには個別のビンのセットが含まれます。図 15（P.180）を参照してください。

イーサネット SLA 動作プロファイル

*動作プロファイル*は、次の動作の側面を定義する設定エンティティです。

- 送信するパケット タイプおよびその量（プローブとバースト設定）
- 測定する統計情報およびそれらを集約する方法
- プローブをスケジュールするタイミング

動作プロファイルはそれ自体でパケットを送信したり、収集された統計情報を生成することはなく、動作インスタンスを作成するために使用されます。

イーサネット SLA 動作

*動作*は、アクティブにパフォーマンス データを収集している特定の動作プロファイルのインスタンスです。動作インスタンスは、特定の送信元（インターフェイスおよび MEP）と特定の宛先（MEP ID または MAC アドレス）と動作プロファイルを関連付けることで作成されます。動作インスタンスは設定が適用されている限り存在し、無限に継続して実行されます。

イーサネット SLA オンデマンド動作

*オンデマンド動作*は、必要に応じて特定の時間に限り、実行できるイーサネット SLA 動作の方式です。新しいサービスを開始している場合、変更の影響を確認するためにサービスのパラメータを変更している場合、または進行中のスケジュールされた動作によって問題が検出されたときにさらに詳細なプローブを実行する場合などの状況で役立ちます。

オンデマンド動作はプロファイルを使用せず、期間が限定されます。収集される統計情報は、動作完了後から限定された時間が経過するか（2 週間）、または手動でクリアされた場合に廃棄されます。

オンデマンド動作は永続的でないので、カードのリロードまたは Minimal Disruptive Restart（MDR）などの特定のイベント中に失われます。

統計情報測定およびイーサネット SLA 動作の概要

ネットワーク パフォーマンスのイーサネット SLA 統計情報測定は、パケットを送信し、次のようなデータのメトリックを保存することで行われます。

- ラウンドトリップ遅延時間：パケットが送信元から宛先へ運ばれ、また送信元に戻ってくる時間。
- ラウンドトリップ ジッター：ラウンドトリップ遅延時間（遅延）の分散です。
- 一方向遅延およびジッター：ルータは、送信元から宛先または宛先から送信元の一方向の遅延またはジッター測定をサポートします。
- 一方向フレーム損失：ルータは、送信元から宛先への、または宛先から送信元への一方向フレーム損失測定もサポートします。

これらのメトリックに加えて、SLA プローブ パケットに対する次の統計情報も保存されます。

- パケット損失カウント
- パケット破損イベント
- 順序不正イベント
- フレーム損失率（FLR）

パケット損失、破損、および順序不正パケットのカウンタはバケットごとに保存され、いずれの場合でも、バケットの合計サンプル数に対する割合が報告されます（たとえば、パケット破損 4 % など）。遅延、ジッター、および損失の統計情報については、バケット全体の最小値、最大値、平均値、標準偏差が報告されるほか、個々のサンプルまたは集約ビンも報告されます。また、合成損失測定統計情報の場合は、バケットの全体的な FLR、および個々の FLR 測定値または集約ビンも報告されます。パケット損失数は、両方向の全体的なパケット損失測定数であり、一方向 FLR は、各方向の損失を個別に測定した値です。

aggregate コマンドを使用して集約をイネーブルにすると、**width** キーワードで設定された、特定の値の範囲内に含まれるサンプル数を保存するためのビンが作成されます。各ビンの範囲内にある結果数のカウンタだけが保持されます。これで使用するメモリは、個々の結果を保存する場合より少なくなります。集約を使用しない場合は、各サンプルが別々に格納されるため、動作のより正確な統計情報の分析を実行できますが、各サンプルでストレージが独立していることでメモリの負荷が上がります。

バケットは統計情報を収集する期間を表します。その期間中に受信したすべての結果が対応するバケットに記録されます。集約がイネーブルの場合、各バケットはビンとカウンタの独自のセットを持ち、これらのカウンタには、バケットで示される期間に開始された測定に関する結果だけが含まれます。

デフォルトでは、プローブごとに個別のバケットがあります。期間は、プローブの存続期間で決まります（**probe**、**send**（**SLA**）、および **schedule**（**SLA**）コマンドで設定）。プローブごとのバケットを増やしたり、プローブごとのバケットを減らしたりできるようにバケット サイズを変更できます（バケットの数を少なくすると、複数のプローブの結果を同じバケットに含めることができます）。特定のメトリックのバケット サイズを変更すると、そのメトリックのすべてのストレージ データをクリアします。すべての既存バケットは削除され、新しいバケットが作成されます。

設定されたスケジュールに基づいて、スケジュールされた SLA 動作プロファイルは無期限に実行され、収集された統計情報は、新しいバケットの記録が必要になると、最も古いバケットのデータが廃棄されるローリング バッファに保存されます。

フレーム損失率（FLR）は、損失測定値に基づいて計算できるプライマリ属性です。FLR とは、送信パケットに対する損失パケットの比率であり、パーセント単位で表します。FLR は方向別（送信元から宛先へ、および宛先から送信元へ）に測定されます。アベイラビリティは属性の 1 つであり、通常は、週、月などの長時間にわたって測定されます。目的は、全体の時間のうち、高損失が続いた時間の割合を求めることです。

スケジュールされたイーサネット SLA 動作の設定の概要

スケジュールされたイーサネット SLA 動作を設定するには、次の基本的な作業を実行します。

1. 各プローブのパケットの送信方法、プローブのスケジュール方法および結果を保存する方法を定義するグローバル プロファイルを設定します。
2. これらのプロファイルを使用して、特定のローカル MEP から特定のピア MEP への動作を設定します。

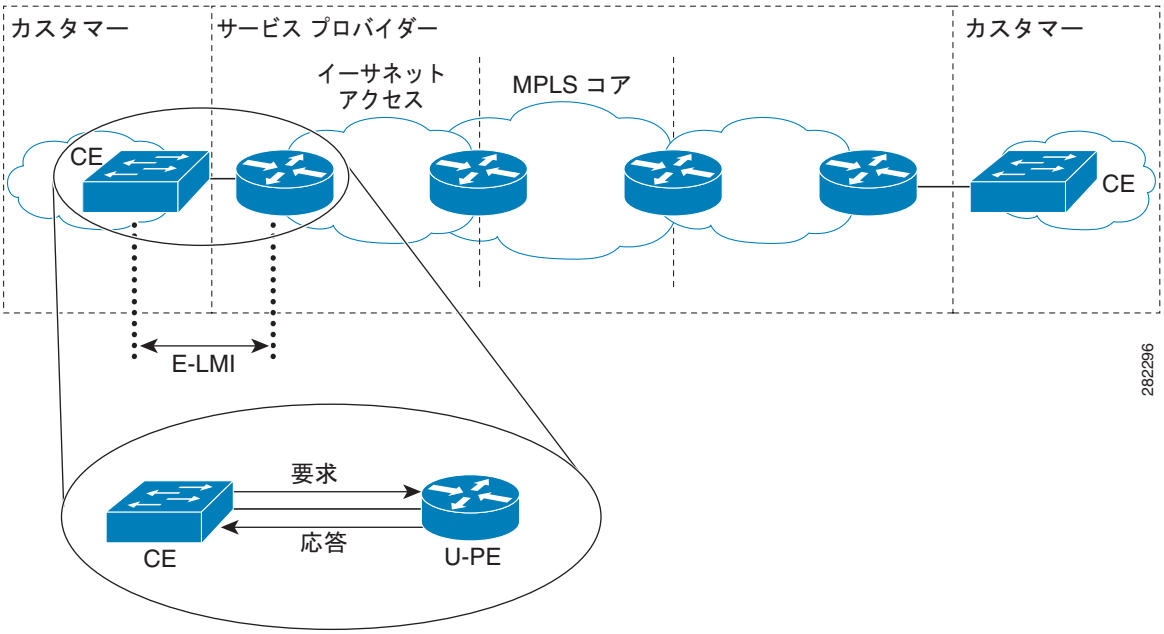
（注） 特定のイーサネット SLA 設定は大量のメモリを使用し、システムの他の機能のパフォーマンスに影響を与える可能性があります。詳細については、「イーサネット SLA の設定」（P.132）を参照してください。

イーサネット LMI

Cisco ASR 9000 シリーズ ルータは、「*Metro Ethernet Forum, Technical Specification MEF 16, Ethernet Local Management Interface (E-LMI), January 2006*」規格によって定義されているイーサネット ローカル管理インターフェイス（E-LMI）プロトコルをサポートします。

E-LMI はカスタマー エッジ（CE）デバイスとプロバイダー エッジ（PE）デバイス間のリンク、またはユーザ ネットワーク インターフェイス（UNI）で動作し、PE デバイスによって提供されるサービスを、CE デバイスで自動設定またはモニタする方法を提供します（図 14 を参照）。

図 14　CE-to-PE リンクでの E-LMI 通信



E-LMI は、CE からユーザ側 PE（uPE）に送信されたステータス問い合わせメッセージの応答にステータス メッセージを使用して、CE への接続ステータスおよび設定パラメータを提供する uPE デバイスを必要とする基本動作を行う非対称プロトコルです。

E-LMI メッセージング

MEF 16 規格で定義されているように E-LMI プロトコルは、2 つのメッセージ タイプ（ステータス問い合わせとステータス）だけの使用を定義します。

これらの E-LMI メッセージは情報要素という必須およびオプションのフィールドで構成され、すべての情報要素が、割り当て済み識別子に関連付けられます。すべてのメッセージには、プロトコル バージョン、メッセージ タイプ、およびレポート情報要素が含まれ、その後に情報要素とサブ情報要素が続きます。

E-LMI メッセージは、IEEE 802.3 タグなし MAC フレーム形式に基づく 46 ～ 1500 バイトのイーサネット フレームにカプセル化されます。E-LMI フレームは次のフィールドがあります。

- 宛先アドレス（6 バイト）：標準の MAC アドレスである 01:80:C2:00:00:07 を使用します。
- 送信元アドレス（6 バイト）：送信側デバイスまたはポートの MAC アドレス。
- E-LMI Ethertype（2 バイト）：88-EE を使用します。
- E-LMI PDU（46 ～ 1500 バイト）：最小 46 バイト長を満たす必要があれば、データに 0x00 のパディングを足します。
- CRC（4 バイト）：エラー検出用の巡回冗長検査。

E-LMI メッセージおよびサポートされる情報要素の詳細については、「*Ethernet Forum, Technical Specification MEF 16, Ethernet Local Management Interface (E-LMI), January 2006*」を参照してください。

シスコ独自のリモート UNI 詳細の情報要素

E-LMI MEF 16 仕様では、独自の情報を送信する方法を定義していません。

E-LMI プロトコル内で追加情報を指定するには、Cisco IOS XR ソフトウェアで、リモート UNI 名および状態に関する情報を CE に送信する、リモート UNI 詳細と呼ばれるシスコ独自の情報要素を実装します。この情報要素により、E-LMI MEF 16 仕様では現在未使用の ID が組み込まれます。

この ID を標準プロトコルで実装する必要が生じたか、または別の理由で E-LMI の将来の実装に対して互換性を確保するには、**extension remote-uni disable** コマンドを使用してリモート UNI 情報要素の伝送をディセーブルにできます。

E-LMI 動作

E-LMI の基本動作は、定期的にステータス問い合わせメッセージを PE デバイスに送信する CE デバイスで構成されます。このメッセージに続いて、PE デバイスによって、要求された情報を含むステータス メッセージ応答が行われます。CE と PE 間のステータス問い合わせおよびステータス メッセージを関連付けるためにシーケンス番号が使用されます。

CE は、レポート タイプと呼ばれる、ステータス問い合わせメッセージの次の 2 つのフォームを送信します。

- E-LMI チェック：PE を使用してデータ インスタンス（DI）番号を検証し、CE に最新の E-LMI 情報があることを確認します。
- フル ステータス：UNI とすべての EVC に関する PE からの情報を要求します。

CE デバイスはステータス問い合わせメッセージの送信を追跡するためにポーリング タイマーを使用しますが、PE デバイスはポーリング検証タイマー（PVT）を使用することもできます。これは、PE のステータス メッセージが送信されてから CE デバイスからのステータス問い合わせが受信されるまでの許容時間を指定するものであり、この時間を過ぎるとエラーが記録されます。

E-LMI 情報を交換するための定期的なステータス問い合わせ / ステータス メッセージ シーケンスに加え、PE デバイスは、EVC ステータスに変更が発生するとすぐに、その情報の送信を CE デバイスが指示しなくても、情報を伝達するために CE デバイスに非同期ステータス メッセージも送信できます。

CE と PE デバイスは両方、ステータス カウンタ（N393）を使用して、E-LMI プロトコル ステータスの変更を宣言するまで、受信した連続するエラーを追跡することで E-LMI のローカル動作ステータスを決定します。

Cisco ASR 9000 シリーズ ルータでサポートされる E-LMI PE 機能

Cisco ASR 9000 シリーズ ルータは MEN で E-LMI の PE デバイスとして機能し、次の PE 機能がサポートされます。

- 物理インターフェイスが CE にステータスを報告する EVC として機能する、イーサネット フローポイント（EFP）としてのレイヤ 2 サブインターフェイスで設定されたイーサネット物理インターフェイスで E-LMI プロトコルをサポートします。Cisco IOS XR ソフトウェアは、イーサネット仮想接続（EVC）の特定の管理コンテキストをサポートしません。

（注） Cisco ASR 9000 シリーズ ルータの E-LMI では、このマニュアルの用語 EVC はレイヤ 2 サブインターフェイス/EFP を指します。

- MEF 16 仕様で定義されている次の E-LMI オプションを設定する機能を提供します。
  - T392 ポーリング検証タイマー（PVT）
  - N393 ステータス カウンタ
- EVC の追加と削除の通知を送信します。
- 設定された EVC が使用可能か（アクティブ）、使用不可か（非アクティブ、一部アクティブ）のステータス通知を送信します。
- ローカル UNI 名の通知を送信します。
- シスコ独自のリモート UNI 詳細情報要素を使用してリモート UNI 名およびステータス、およびシスコ独自のリモート UNI 情報要素をディセーブルにする機能についての通知を送信します。
- 次のような、CE に UNI および EVC 属性に関する情報を送信します（CE がこれらの属性を自動的に設定できるようにするため）。
  - CE-VLAN の EVC へのマップ
  - CE-VLAN マップ タイプ（バンドリング、All-to-one バンドリング、サービス多重化）
  - サービス タイプ（ポイントツーポイントまたはマルチポイント）
- CFM のアップ MEP を使用して EVC の状態、EVC サービス タイプ、およびリモート UNI 詳細を入手します。
- コマンドライン インターフェイス（CLI）または Extensible Markup Language（XML）インターフェイスを使用して、プロトコルのインターフェイス単位の動作状態（現在プロトコルを使用して CE に通信しているすべての情報を含む）を取得する機能があります。
- ラインカード（物理インターフェイスごとに 1 つ）あたり最大 80 個の E-LMI セッションをサポートします。
- E-LMI がイネーブルになっているすべての物理インターフェイスのラインカードごとに EVC を合計最大 32000 個サポートします。

サポートされていない E-LMI 機能

E-LMI の次の領域は Cisco ASR 9000 シリーズ ルータでサポートされていません。

- CE 機能

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。

UDLD 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 が正しく動作するには、リンク上の両方のデバイスで UDLD がサポートされており、それぞれのポートで有効になっている必要があります。

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」（P.98）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。

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 ポートによって規定され、リモート側のメッセージ間隔に依存します。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。

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 リンク OAM の設定」（P.100）
- 「イーサネット CFM の設定」（P.109）
- 「イーサネット SLA の設定」（P.132）
- 「イーサネット LMI の設定」（P.144）
- 「UDLD の設定」（P.162）

イーサネット リンク OAM の設定

カスタム EOAM の設定は、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、個々のインターフェイスにプロファイルをアタッチすることによって、複数のインターフェイスで設定および共有できます。プロファイルの設定は、プロファイルがインターフェイスにアタッチされるまで有効になりません。EOAM プロファイルがインターフェイスにアタッチされた後に、必要なときにプロファイル設定を上書きするように、それぞれの EOAM 機能をインターフェイスで個別に設定できます。

ここでは、次の手順で EOAM プロファイルを設定してインターフェイスにアタッチする方法について説明します。

- 「イーサネット OAM プロファイルの設定」（P.100）
- 「インターフェイスへのイーサネット OAM プロファイルのアタッチ」（P.106）
- 「イーサネット OAM のインターフェイスでの設定およびプロファイル設定の上書き」（P.107）
- 「イーサネット OAM の設定の確認」（P.108）

イーサネット OAM プロファイルの設定

イーサネット OAM プロファイルを設定するには、次の手順を実行します。

手順の概要

1. **configure**
2. **ethernet oam profile** *profile-name*
3. **link-monitor**
4. **symbol-period window** *window*
5. **symbol-period threshold low** *threshold*
6. **frame window** *window*
7. **frame threshold low** ***threshold***
8. **frame-period window** ***window***

9. **frame-period threshold low** *threshold*
10. **frame-seconds window** *window*
11. **frame-seconds threshold low** *threshold*
12. **exit**
13. **mib-retrieval**
14. **connection timeout** *seconds*
15. **hello-interval** {**100ms** | **1s**}
16. **mode** {**active** | **passive**}
17. **require-remote mode** {**active** | **passive**}
18. **require-remote link-monitoring**
19. **require-remote mib-retrieval**
20. **action capabilities-conflict** {**disable** | **efd** | **error-disable-interface**}
21. **action critical-event** {**disable** | **error-disable-interface**}
22. **action discovery-timeout** {**disable** | **efd** | **error-disable-interface** }
23. **action dying-gasp** {**disable** | **error-disable-interface**}
24. **action high-threshold** {**error-disable-interface** | **log**}
25. **action remote-loopback disable**
26. **action session-down** {**disable** | **efd** | **error-disable-interface**}
27. **action session-up disable**
28. **action uni-directional link-fault** {**disable** | **efd** | **error-disable-interface**}
29. **action wiring-conflict** {**disable** | **efd** | **log**}
30. **uni-directional link-fault detection**
31. **commit**
32. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/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 2	**ethernet oam profile** *profile-name* **例：** `RP/0/RSP0/CPU0:router(config)# ethernet oam profile Profile_1`	新しいイーサネット運用管理および保守（OAM）プロファイルを作成し、イーサネット OAM 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	**link-monitor** **例：** `RP/0/RSP0/CPU0:router(config-eoam)# link-monitor`	イーサネット OAM リンク モニタ コンフィギュレーション モードを開始します。
ステップ 4	**symbol-period window** *window* **例：** `RP/0/RSP0/CPU0:router(config-eoam-lm)# symbol-period window 60000`	（任意）イーサネット OAM シンボル期間エラー イベントのウィンドウ サイズをミリ秒で設定します。 範囲は 1000 ～ 60000 です。 デフォルト値は 1000 です。
ステップ 5	**symbol-period threshold low** *threshold* **high** *threshold* **例：** `RP/0/RSP0/CPU0:router(config-eoam-lm)# symbol-period threshold low 10000000 high 60000000`	（任意）イーサネット OAM シンボル期間エラー イベントをトリガーするしきい値を（シンボル単位）設定します。上限しきい値はオプションです。下限しきい値とともにのみ設定できます。 範囲は 0 ～ 60000000 です。 デフォルトの下限しきい値は 1 です。
ステップ 6	**frame window** *window* **例：** `RP/0/RSP0/CPU0:router(config-eoam-lm)# frame window 60`	（任意）OAM フレーム エラー イベントのフレームのウィンドウ サイズをミリ秒で設定します。 範囲は 1000 ～ 60000 です。 デフォルト値は 1000 です。
ステップ 7	**frame threshold low** *threshold* **high** *threshold* **例：** `RP/0/RSP0/CPU0:router(config-eoam-lm)# frame threshold low 10000000 high 60000000`	（任意）イーサネット OAM フレーム エラー イベントをトリガーするしきい値を（シンボル単位）設定します。上限しきい値はオプションです。下限しきい値とともにのみ設定できます。 範囲は 0 ～ 60000000 です。 デフォルトの下限しきい値は 1 です。
ステップ 8	**frame-period window** *window* **例：** `RP/0/RSP0/CPU0:router(config-eoam-lm)# frame-period window 60000`	（任意）イーサネット OAM フレーム期間エラー イベントのウィンドウ サイズをミリ秒で設定します。 範囲は 100 ～ 60000 です。 デフォルト値は 1000 です。
ステップ 9	**frame-period threshold low** *threshold* **high** *threshold* `RP/0/RSP0/CPU0:router(config-eoam-lm)# frame-period threshold low 100 high 1000000`	（任意）イーサネット OAM フレーム期間エラー イベントをトリガーするしきい値を（フレーム単位）設定します。上限しきい値はオプションです。下限しきい値とともにのみ設定できます。 範囲は 0 ～ 1000000 です。 デフォルトの下限しきい値は 60000 です。
ステップ 10	**frame-seconds window** *window* **例：** `RP/0/RSP0/CPU0:router(config-eoam-lm)# frame-seconds window 900000`	（任意）OAM フレーム秒数エラー イベントのウィンドウ サイズをミリ秒で設定します。 範囲は 10000 ～ 900000 です。 デフォルト値は 6000 です。

	コマンドまたはアクション	目的
ステップ 11	**frame-seconds threshold low** *threshold* **high** *threshold* **例：** `RP/0/RSP0/CPU0:router(config-eoam-lm)# frame-seconds threshold 3 threshold 900`	（任意）フレーム秒数エラー イベントをトリガーするしきい値を（秒単位）設定します。上限しきい値は下限しきい値とともにのみ設定できます。 範囲は 1 ～ 900 です デフォルト値は、1 です
ステップ 12	**exit** **例：** `RP/0/RSP0/CPU0:router(config-eoam-lm)# exit`	イーサネット OAM モードに戻ります。
ステップ 13	**mib-retrieval** **例：** `RP/0/RSP0/CPU0:router(config-eoam)# mib-retrieval`	イーサネット OAM プロファイルまたはイーサネット OAM インターフェイスで MIB 取得をイネーブルにします。
ステップ 14	**connection timeout** *seconds* **例：** `RP/0/RSP0/CPU0:router(config-eoam)# connection timeout 30`	イーサネット OAM セッションのタイムアウト値を（秒単位）設定します。 範囲は 2 ～ 30 です。 デフォルト値は 5 です。
ステップ 15	**hello-interval** {**100ms**|**1s**} **例：** `RP/0/RSP0/CPU0:router(config-eoam)# hello-interval 100ms`	イーサネット OAM セッションの hello パケット間の間隔を設定します。デフォルトは 1 秒（**1s**）です。
ステップ 16	**mode** {**active**|**passive**} **例：** `RP/0/RSP0/CPU0:router(config-eoam)# mode passive`	イーサネット OAM モードを設定します。デフォルトは active です。
ステップ 17	**require-remote mode** {**active**|**passive**} **例：** `RP/0/RSP0/CPU0:router(config-eoam)# require-remote mode active`	OAM セッションがアクティブになる前にアクティブ モードまたはパッシブ モードをリモート エンドで設定する必要があります。
ステップ 18	**require-remote link-monitoring** **例：** `RP/0/RSP0/CPU0:router(config-eoam)# require-remote link-monitoring`	OAM セッションがアクティブになる前に、リンク モニタリングをリモート エンドで設定する必要があります。
ステップ 19	**require-remote mib-retrieval** **例：** `RP/0/RSP0/CPU0:router(config-eoam)# require-remote mib-retrieval`	OAM セッションがアクティブになる前に MIB 取得をリモート エンドで設定する必要があります。

	コマンドまたはアクション	目的
ステップ 20	**action capabilities-conflict** {**disable** | **efd** | **error-disable-interface**} **例：** `RP/0/RSP0/CPU0:router(config-eoam)# action capabilities-conflict efd`	機能の矛盾のイベントが発生したときにインターフェイスで実行するアクションを指定します。デフォルト アクションは、syslog エントリの作成です。 **（注）** デフォルトを変更する場合、プロファイルの設定を上書きし、インターフェイスのイベントが発生した場合にそれを記録するには、**log** キーワード オプションをインターフェイス イーサネット OAM コンフィギュレーション モードで使用できます。
ステップ 21	**action critical-event** {**disable** | **error-disable-interface**} **例：** `RP/0/RSP0/CPU0:router(config-eoam)# action critical-event error-disable-interface`	重大イベント通知をリモート イーサネット OAM ピアから受信したときにインターフェイスで実行するアクションを指定します。デフォルト アクションは、syslog エントリの作成です。 **（注）** デフォルトを変更する場合、プロファイルの設定を上書きし、インターフェイスのイベントが発生した場合にそれを記録するには、**log** キーワード オプションをインターフェイス イーサネット OAM コンフィギュレーション モードで使用できます。
ステップ 22	**action discovery-timeout** {**disable** | **efd** | **error-disable-interface**} **例：** `RP/0/RSP0/CPU0:router(config-eoam)# action discovery-timeout efd`	接続タイムアウトが発生したときにインターフェイスで実行するアクションを指定します。デフォルト アクションは、syslog エントリの作成です。 **（注）** デフォルトを変更する場合、プロファイルの設定を上書きし、インターフェイスのイベントが発生した場合にそれを記録するには、**log** キーワード オプションをインターフェイス イーサネット OAM コンフィギュレーション モードで使用できます。
ステップ 23	**action dying-gasp** {**disable** | **error-disable-interface**} **例：** `RP/0/RSP0/CPU0:router(config-eoam)# action dying-gasp error-disable-interface`	dying-gasp 通知をリモート イーサネット OAM ピアから受信したときにインターフェイスで実行するアクションを指定します。デフォルト アクションは、syslog エントリの作成です。 **（注）** デフォルトを変更する場合、プロファイルの設定を上書きし、インターフェイスのイベントが発生した場合にそれを記録するには、**log** キーワード オプションをインターフェイス イーサネット OAM コンフィギュレーション モードで使用できます。
ステップ 24	**action high-threshold** {**error-disable-interface** | **log**} **例：** `RP/0/RSP0/CPU0:router(config-eoam)# action high-threshold error-disable-interface`	上限しきい値を超過した場合にインターフェイスで実行するアクションを指定します。デフォルトは上限しきい値を超過した場合、何のアクションも実行しません。 **（注）** デフォルトを変更する場合、プロファイルの設定を上書きし、イベントが発生した場合にインターフェイスでアクションしないようにするには、**disable** キーワード オプションをインターネット イーサネット OAM コンフィギュレーション モードで使用できます。

	コマンドまたはアクション	目的
ステップ 25	**action remote-loopback disable** **例：** `RP/0/RSP0/CPU0:router(config-eoam)# action remote-loopback disable`	リモート ループバックのイベント発生時に処理がインターフェイスで実行されないことを指定します。デフォルト アクションは、syslog エントリの作成です。 **（注）** デフォルトを変更する場合、プロファイルの設定を上書きし、インターフェイスのイベントが発生した場合にそれを記録するには、**log** キーワード オプションをインターフェイス イーサネット OAM コンフィギュレーション モードで使用できます。
ステップ 26	**action session-down** {**disable** | **efd** | **error-disable-interface**} **例：** `RP/0/RSP0/CPU0:router(config-eoam)# action session-down efd`	イーサネット OAM セッションがダウンした場合にインターフェイスで実行するアクションを指定します。 **（注）** デフォルトを変更する場合、プロファイルの設定を上書きし、インターフェイスのイベントが発生した場合にそれを記録するには、**log** キーワード オプションをインターフェイス イーサネット OAM コンフィギュレーション モードで使用できます。
ステップ 27	**action session-up disable** **例：** `RP/0/RSP0/CPU0:router(config-eoam)# action session-up disable`	イーサネット OAM セッションが設定された場合にアクションがインターフェイスで実行されないことを指定します。デフォルト アクションは、syslog エントリの作成です。 **（注）** デフォルトを変更する場合、プロファイルの設定を上書きし、インターフェイスのイベントが発生した場合にそれを記録するには、**log** キーワード オプションをインターフェイス イーサネット OAM コンフィギュレーション モードで使用できます。
ステップ 28	**action uni-directional link-fault** {**disable** | **efd** | **error-disable-interface**}	リンク障害通知をリモート イーサネット OAM ピアから受信したときにインターフェイスで実行するアクションを指定します。デフォルト アクションは、syslog エントリの作成です。 **（注）** デフォルトを変更する場合、プロファイルの設定を上書きし、インターフェイスのイベントが発生した場合にそれを記録するには、**log** キーワード オプションをインターフェイス イーサネット OAM コンフィギュレーション モードで使用できます。 **（注）** Cisco IOS XR Release 4.0 では、このコマンドは **action link-fault** コマンドを置き換えます。
ステップ 29	**action wiring-conflict** {**disable** | **efd** | **log**} **例：** `RP/0/RSP0/CPU0:router(config-eoam)# action session-down efd`	配線競合イベントが発生したときにインターフェイスで実行するアクションを指定します。デフォルトはインターフェイスを errdisable ステートにします。 **（注）** デフォルトを変更する場合、プロファイルの設定を上書きし、イベントが発生した場合にインターフェイスを errdisable ステートにするには、**error-disable-interface** キーワード オプションをインターフェイス イーサネット OAM コンフィギュレーション モードで使用できます。

	コマンドまたはアクション	目的
ステップ 30	**uni-directional link-fault detection** **例：** `RP/0/RSP0/CPU0:router(config-eoam)# uni-directional link-fault detection`	ローカルの単方向リンク障害の検出をイネーブルにし、イーサネット OAM ピアにその障害の通知を送信します。
ステップ 31	**commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# commit`	実行中のコンフィギュレーション ファイルに設定の変更を保存し、引き続きコンフィギュレーション セッションを実行します。
ステップ 32	**end** **例：** `RP/0/RSP0/CPU0:router(config-if)# end`	コンフィギュレーション セッションを終了し、EXEC モードに戻ります。

インターフェイスへのイーサネット OAM プロファイルのアタッチ

インターフェイスにイーサネット OAM プロファイルをアタッチするには、次の手順を実行します。

手順の概要

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
3. **ethernet oam**
4. **profile** *profile-name*
5. **commit**
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/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 2	**interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/1/0/0`	インターフェイス コンフィギュレーション モードを開始し、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。 **（注）** この例は、モジュラ サービス カード スロット 1 の 8 ポート 10 ギガビット イーサネット インターフェイスです。
ステップ 3	**ethernet oam** **例：** `RP/0/RSP0/CPU0:router(config-if)# ethernet oam`	イーサネット OAM をイネーブルにし、インターフェイス イーサネット OAM コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 4	**profile** *profile-name* **例：** `RP/0/RSP0/CPU0:router(config-if-eoam)# profile Profile_1`	インターフェイスに指定されたイーサネット OAM プロファイル（*profile-name*）、および設定すべてをアタッチします。
ステップ 5	**commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# commit`	実行中のコンフィギュレーション ファイルに設定の変更を保存し、引き続きコンフィギュレーション セッションを実行します。
ステップ 6	**end** **例：** `RP/0/RSP0/CPU0:router(config-if)# end`	コンフィギュレーション セッションを終了し、EXEC モードに戻ります。

イーサネット OAM のインターフェイスでの設定およびプロファイル設定の上書き

EOAM プロファイルの使用は、共通の EOAM の設定でいくつかのインターフェイスを設定する効率的な方法です。ただし、プロファイルを使用して特定のインターフェイスの特定の機能の動作を変更する場合、プロファイル設定を上書きできます。インターフェイスに適用される特定のプロファイル設定を上書きするには、そのインターフェイスの動作を変更するようにインターフェイス イーサネット OAM コンフィギュレーション モードでこのコマンドを設定できます。

場合によっては、コマンドのデフォルト設定により、特定のキーワード オプションだけをインターフェイス イーサネット OAM コンフィギュレーション モードで使用できます。たとえば、**action** コマンドを設定しなければ、このコマンドの複数の形式のデフォルト動作では、プロファイルが作成されインターフェイスに適用されるときに syslog エントリを作成します。したがって、**log** キーワードは、デフォルトの動作であるため、プロファイルのこれらのコマンドについてはイーサネット OAM 設定で使用できなくなります。ただし、プロファイルの設定でデフォルトが変更された場合、インターフェイス イーサネット OAM 設定で **log** キーワードを使用でき、特定のインターフェイスの syslog エントリの作成のアクションを保持できるようになります。

デフォルトのイーサネット OAM コンフィギュレーション設定すべてを表示するには、「イーサネット OAM の設定の確認」（P.108）を参照してください。

イーサネット OAM 設定をインターフェイスで設定し、プロファイルの設定を上書きするには、次の手順を実行します。

手順の概要

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
3. **ethernet oam**
4. *interface-Ethernet-OAM-command*
5. **commit**
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/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 2	**interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/1/0/0`	インターフェイス コンフィギュレーション モードを開始し、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。 **（注）** この例は、モジュラ サービス カード スロット 1 の 8 ポート 10 ギガビット イーサネット インターフェイスです。
ステップ 3	**ethernet oam** **例：** `RP/0/RSP0/CPU0:router(config-if)# ethernet oam`	イーサネット OAM をイネーブルにし、インターフェイス イーサネット OAM 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-Ethernet-OAM-command* **例：** `RP/0/RSP0/CPU0:router(config-if-eoam)# action capabilities-conflict error-disable-interface`	イーサネット OAM コンフィギュレーション コマンドを設定し、プロファイル設定の設定を上書きします。ここで、*interface-Ethernet-OAM-command* は、インターフェイス イーサネット OAM コンフィギュレーション モードのプラットフォームでサポートされるいずれかのコマンドです。
ステップ 5	**commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# commit`	実行中のコンフィギュレーション ファイルに設定の変更を保存し、引き続きコンフィギュレーション セッションを実行します。
ステップ 6	**end** **例：** `RP/0/RSP0/CPU0:router(config-if)# end`	コンフィギュレーション セッションを終了し、EXEC モードに戻ります。

イーサネット OAM の設定の確認

show ethernet oam configuration コマンドを使用して、特定のインターフェイスまたはすべてのインターフェイスのイーサネット OAM 設定の値を表示します。次の例は、イーサネット OAM の設定のデフォルト値を示します。

```
RP/0/RSP0/CPU0:router# show ethernet oam configuration
Thu Aug  5 22:07:06.870 DST
GigabitEthernet0/4/0/0:
  Hello interval:                                      1s
  Link monitoring enabled:                              Y
  Remote loopback enabled:                              N
  Mib retrieval enabled:                                N
  Uni-directional link-fault detection enabled:         N
  Configured mode:                                 Active
  Connection timeout:                                   5
  Symbol period window:                                 0
  Symbol period low threshold:                          1
```

```
Symbol period high threshold:                          None
Frame window:                                          1000
Frame low threshold:                                      1
Frame high threshold:                                  None
Frame period window:                                   1000
Frame period low threshold:                               1
Frame period high threshold:                           None
Frame seconds window:                                 60000
Frame seconds low threshold:                              1
Frame seconds high threshold:                          None
High threshold action:                                 None
Link fault action:                                      Log
Dying gasp action:                                      Log
Critical event action:                                  Log
Discovery timeout action:                               Log
Capabilities conflict action:                           Log
Wiring conflict action:                          Error-Disable
Session up action:                                      Log
Session down action:                                    Log
Remote loopback action:                                 Log
Require remote mode:                                 Ignore
Require remote MIB retrieval:                             N
Require remote loopback support:                          N
Require remote link monitoring:                           N
```

イーサネット CFM の設定

イーサネット CFM を設定するには、次の作業を実行します。

- 「CFM メンテナンス ドメインの設定」（P.109）（必須）
- 「CFM メンテナンス ドメインのサービスの設定」（P.111）（必須）
- 「CFM サービスの連続性チェックのイネーブル化および設定」（P.113）（任意）
- 「CFM サービスの自動 MIP 作成の設定」（P.115）（任意）
- 「CFM サービスの MEP でのクロスチェックの設定」（P.117）（任意）
- 「CFM サービスのその他のオプションの設定」（P.119）（任意）
- 「CFM MEP の設定」（P.121）（必須）
- 「Y.1731 AIS の設定」（P.123）（任意）
- 「CFM サービスの EFD の設定」（P.127）（任意）
- 「CFM の柔軟な VLAN タギングの設定」（P.128）（任意）
- 「CFM 設定の確認」（P.130）
- 「トラブルシューティングのヒント」（P.130）

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。

手順の概要

1. **configure**
2. **ethernet cfm**

3. **domain** *domain-name* **level** ***level-value*** [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
4. **traceroute cache hold-time** ***minutes*** **size** ***entries***
5. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ethernet cfm** **例：** `RP/0/RSP0/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/RSP0/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 ID（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。

	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/RSP0/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/RSP0/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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

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 32000 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。

手順の概要

1. **configure**
2. **ethernet cfm**
3. **domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
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*]]
5. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ethernet cfm** **例：** `RP/0/RSP0/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/RSP0/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 ID（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。

	コマンドまたはアクション	目的
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/RSP0/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 xconnect 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/RSP0/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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

CFM サービスの連続性チェックのイネーブル化および設定

Cisco ASR 9000 シリーズ ルータでは、IEEE 802.1ag 仕様で定義されている連続性チェックをサポートし、100 ms 以上の CCM 間隔をサポートします。CCM メッセージの全体的なパケット レートは、カードごとに送信が最大 16000 CCM/秒、受信が最大 16000 CCM/秒です。

（注） イーサネット SLA が設定されている場合、CCM および SLA フレーム全体を合わせたパケット レートは、カードごとの各方向で 16000 フレーム/秒です。

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。

手順の概要

1. **configure**
2. **ethernet cfm**

3. **domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
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*]]
5. **continuity-check interval** ***time*** [**loss-threshold** ***threshold***]
6. **continuity-check archive hold-time** ***minutes***
7. **continuity-check loss auto-traceroute**
8. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ethernet cfm** **例：** `RP/0/RSP0/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/RSP0/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 ID（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/RSP0/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 xconnect 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。

	コマンドまたはアクション	目的
ステップ 5	**continuity-check interval** *time* [**loss-threshold** *threshold*] **例：** `RP/0/RSP0/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/RSP0/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/RSP0/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/RSP0/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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

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」（P.72）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。

手順の概要

1. **configure**

2. **ethernet cfm**
3. **domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
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*]]
5. **mip auto-create {all | lower-mep-only}**
6. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ethernet cfm** **例：** `RP/0/RSP0/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：** `RP/0/RSP0/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 ID（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/RSP0/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 xconnect 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。

	コマンドまたはアクション	目的
ステップ 5	**mip auto-create** {**all** | **lower-mep-only**} **例：** `RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# mip auto-create all`	（任意）ブリッジ ドメインまたは xconnect の MIP の自動作成をイネーブルにします。
ステップ 6	**end** または **commit** **例：** `RP/0/RSP0/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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

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。

手順の概要

1. **configure**
2. **ethernet cfm**
3. **domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
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*]]
5. **mep crosscheck**
6. **mep-id** *mep-id-number* [**mac-address** *mac-address*]
7. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ethernet cfm** **例：** `RP/0/RSP0/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：** `RP/0/RSP0/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 ID（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/RSP0/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 xconnect 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/RSP0/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。

	コマンドまたはアクション	目的
ステップ 6	**mep-id** *mep-id-number* [**mac-address** *mac-address*] **例：** `RP/0/RSP0/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/RSP0/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]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

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。

手順の概要

1. **configure**
2. **ethernet cfm**
3. **domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
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*]]
5. **maximum meps** *number*
6. **log** {**ais** | **continuity-check errors** | **continuity-check mep changes** | **crosscheck errors** | **efd**}
7. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ethernet cfm** **例：** `RP/0/RSP0/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：** `RP/0/RSP0/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 ID（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/RSP0/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 xconnect 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/RSP0/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。

	コマンドまたはアクション	目的
ステップ 6	**log** {**ais**|**continuity-check errors**|**continuity-check mep changes**|**crosscheck errors**|**efd**} **例：** `RP/0/RSP0/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/RSP0/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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

CFM MEP の設定

CFM MEP を設定する場合、次の注意事項に従ってください。

- 最大 32000 個のローカル MEP がカードごとにサポートされます。
- CFM メンテナンス ポイントは、次のインターフェイス タイプで作成できます。
  - すべての物理イーサネット インターフェイス（RSP 管理インターフェイスを除く）。
  - イーサネット バンドル インターフェイス。
  - すべての物理およびバンドル イーサネット サブインターフェイス（次のガイドラインに従ってカプセル化が設定されている場合）。

    フレームは、VLAN ID および CoS ビットに基づいて一致するだけです。

    フレームは VLAN の「any」を使用して一致することはありません
  - イーサネット バンドル メンバ インターフェイス：レベル 0 だけのダウン MEP を作成できます。
- CFM メンテナンス ポイントは、レイヤ 2 およびレイヤ 3 の両方のインターフェイスで作成できます。L3 インターフェイスではダウン MEP だけを作成できます。

制約事項

MEP を設定する場合、次の制約事項を考慮してください。

- レベル 0 のメンテナンス ポイントは、バンドル インターフェイスではサポートされません。
- タグなしイーサネット フレームを照合するサブインターフェイスを設定する場合（**encapsulation default** コマンドを設定するなど）、基礎となる物理またはバンドル インターフェイスのダウン MEP を作成できません。
- アップ MEP はレイヤ 3 インターフェイスではサポートされません。

手順の概要

1. **configure**
2. **interface** {**GigabitEthernet** | **TenGigE** | **Bundle-Ether**} *interface-path-id.subinterface*
3. **ethernet cfm**
4. **mep domain** *domain-name* **service** *service-name* **mep-id** *id-number*
5. **cos** *cos*
6. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** {**GigabitEthernet** | **TenGigE** | **Bundle-Ether**} *interface-path-id.subinterface* **例：** `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/1`	MEP を作成するイーサネット インターフェイスのタイプ。**GigabitEthernet**、**TenGigE**、または **Bundle-Ether** および物理インターフェイスまたは仮想インターフェイスの後にサブインターフェイス パス ID を続けて入力します。 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。 ルータ構文の詳細については、疑問符（**?**）オンライン ヘルプ機能を使用します。
ステップ 3	**ethernet cfm** **例：** `RP/0/RSP0/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 4	**mep domain** *domain-name* **service** *service-name* **mep-id** *id-number* **例：** `RP/0/RSP0/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。

	コマンドまたはアクション	目的
ステップ 5	**cos** *cos* **例：** `RP/0/RSP0/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。
ステップ 6	**end** または **commit** **例：** `RP/0/RSP0/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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

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 インターフェイスの AIS の設定

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。

手順の概要

1. **configure**
2. **ethernet cfm**
3. **domain** *name* **level** *level*
4. **service** *name* **bridge group** *name* **bridge-domain** *name*
5. **ais transmission** [**interval** {**1s** | **1m**}][**cos** *cos*]
6. **log ais**

7. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ethernet cfm** **例：** `RP/0/RSP0/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/RSP0/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/RSP0/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	**ais transmission** [**interval** {**1s**|**1m**}][**cos** *cos*] **例：** `RP/0/RSP0/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。

	コマンドまたはアクション	目的
ステップ 6	**log ais** **例：** `RP/0/RSP0/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 7	**end** または **commit** **例：** `RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

CFM インターフェイスの 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。

手順の概要

1. **configure**
2. **interface gigabitethernet** *interface-path-id*
3. **ethernet cfm**
4. **ais transmission up interval 1m cos** *cos*
5. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface gigabitethernet** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# interface gigabitethernet 0/1/0/2`	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/RSP0/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/RSP0/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/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

CFM サービスの EFD の設定

CFM サービスの EFD を設定するには、次の手順を実行します。

制約事項

EFD はアップ MEP ではサポートされません。これは、特定のサービス内のダウン MEP でしか設定できません。

手順の概要

1. **configure**
2. **ethernet cfm**
3. **domain** *domain-name* **level** *level-value*
4. **service** *service-name* **down-meps**
5. **efd**
6. **log efd**
7. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ethernet cfm** **例：** `RP/0/RSP0/CPU0:router(config)# ethernet cfm`	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* **例：** `RP/0/RSP0/CPU0:router(config-cfm-dmn)# domain D1 level 1`	CFM ドメインを指定または作成し、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。
ステップ 4	**service** *service-name* **down-meps** **例：** `RP/0/RSP0/CPU0:router(config-cfm-dmn)# service S1 down-meps`	ダウン MEP の CFM サービスを指定または作成し、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。
ステップ 5	**efd** **例：** `RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# efd`	すべてのダウン MEP の EFD をダウン MEP サービスでイネーブルにします。

	コマンドまたはアクション	目的
ステップ 6	**log efd** **例：** `RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# log efd`	（任意）インターフェイスでの EFD 状態変更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/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

EFD 設定の確認

次に、イーサネット障害検出（EFD）のためにシャット ダウンされたすべてのインターフェイスを表示する例を示します。

```
RP/0/RSP0/CPU0:router# show efd interfaces

Server VLAN MA
==============
Interface         Clients
--------------------------
GigE0/0/0/0.0     CFM
```

CFM の柔軟な VLAN タギングの設定

アップ MEP からの CFM パケット内のタグの数を、CFM ドメイン サービスで 1 に設定するには、次の手順を使用します。

手順の概要

1. **configure**
2. **ethernet cfm**
3. **domain** *name* **level** *level*

4. **service** *name* **bridge group** *name* **bridge-domain** *name*
5. **tags** *number*
6. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ethernet cfm** **例：** `RP/0/RSP0/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/RSP0/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/RSP0/CPU0:router(config-cfm-dmn)# service S2 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。

	コマンドまたはアクション	目的
ステップ 5	**tags** *number* **例：** `RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# tags 1`	アップ MEP からの 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/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

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。

コマンド	目的
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。

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解決するには、次の手順を実行します。

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/RSP0/CPU0:router# ping ethernet cfm domain D1 service S1 mep-id 16 source
```

```
interface GigabitEthernet 0/0/0/0

Type escape sequence to abort.
Sending 5 CFM Loopbacks, timeout is 2 seconds -
Domain foo (level 2), Service foo
Source: MEP ID 1, interface GigabitEthernet0/0/0/0
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** コマンドの結果がピア MEP への接続の問題を示している場合、次の例に示すように、問題の場所をさらに切り分けるのに役立つ **traceroute ethernet cfm** コマンドを使用します。

```
RP/0/RSP0/CPU0:router# traceroute ethernet cfm domain D1 service S1 mep-id 16 source
interface gigabitethernet 0/0/0/0

Traceroutes in domain D1 (level 4), service S1
Source: MEP-ID 1, interface GigabitEthernet0/0/0/0
================================================================================
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    Gi0/0/0/0
  2 abc                                             0001.0203.0401 [Ok]    FDB
      ios                                           Not present
  3 bcd                      0001.0203.0402 [Ok]                           Hit
      abc                    GigE0/0
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** コマンドを使用した場合、MAC アドレスは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場所での問題を示しています。

イーサネット SLA の設定

ここでは、イーサネット SLA を設定する方法について説明します。

イーサネット SLA の設定時の注意事項

注意 特定の SLA 設定は大量のメモリを使用し、ルータの他の機能のパフォーマンスに影響を与える可能性があります。

イーサネット SLA を設定する前に、次の注意事項に従ってください。

- 集約：**aggregate none** コマンドを使用すると、個々の測定がそれぞれ記録されるため、各集約ビンのカウンタが単に増えるだけでなく、必要なメモリの量が大幅に増加します。集約を設定する場合、ビンが増えることで必要なメモリも多くなることを考慮してください。
- バケットのアーカイブ：**buckets archive** コマンドを設定する場合、さらに多くの履歴が保存され、より多くのメモリが使用されることを考慮してください。
- 2 つの統計情報（遅延およびジッターの両方など）を測定することは、1 つの統計情報の測定の約 2 倍のメモリを使用します。
- 一方向の送信元から宛先および宛先から送信元の測定の統計情報は別々に保存され、ラウンドトリップの統計情報の単一セットを保存するときの 2 倍のメモリを消費します。
- Cisco ASR 9000 シリーズ ルータは、100 ms 以上の SLA パケット間隔をサポートします。イーサネット SLA が設定されている場合、CCM および SLA フレーム全体を合わせたパケット レートは、カードごとの各方向で 16000 フレーム / 秒です。

次の手順は、レイヤ 2 でのイーサネット サービス レベル契約（SLA）のモニタリングを設定する手順について説明します。

SLA を設定するには、次の作業を実行します。

- 「SLA 動作プロファイルの設定」（P.132）
- 「プロファイルの SLA プローブ パラメータの設定」（P.133）
- 「プロファイルの SLA 統計情報測定の設定」（P.135）
- 「プロファイルの SLA 動作プローブのスケジュールの設定」（P.137）
- 「SLA 動作の設定」（P.139）
- 「オンデマンド SLA 動作の設定」（P.140）
- 「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」（P.143）

SLA 動作プロファイルの設定

プロファイルを設定するには、次の手順を実行します。

手順の概要

1. **configure**
2. **ethernet sla**
3. **profile** *profile-name* **type** {**cfm-delay-measurement** | **cfm-loopback** | **cfm-synthetic-loss-measurement**}

4. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ethernet sla** **例：** `RP/0/RSP0/CPU0:router# ethernet sla`	SLA 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	**profile** *profile-name* **type** {**cfm-delay-measurement** | **cfm-loopback** | **cfm-synthetic-loss-measurement**} **例：** `RP/0/RSP0/CPU0:router(config-sla)# profile Prof1 type cfm-loopback`	SLA 動作プロファイルを作成して、SLA プロファイル 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	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-sla)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

プロファイルの SLA プローブ パラメータの設定

プロファイルの SLA プローブ パラメータを設定するには、SLA プロファイル コンフィギュレーション モードから次の手順を実行します。

手順の概要

1. **probe**
2. **send burst** {**every** *number* {**seconds** | **minutes** | **hours**}| **once**} **packet count** *packets* **interval** *number* {**seconds** | **milliseconds**}
   または
   send packet {**every** *number* {**milliseconds** | **seconds** | **minutes** | **hours**} | **once**}
3. **packet size** *bytes* [**test pattern** {**hex 0x***HHHHHHHH* | **pseudo-random**}]
4. **priority** *priority*
5. **synthetic loss calculation packets** *number*
6. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**probe** **例：** `RP/0/RSP0/CPU0:router(config-sla-prof)# probe`	SLA プロファイル プローブ 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	**send burst** {**every** *number* {**seconds** | **minutes** | **hours**} | **once**} **packet count** *packets* **interval** *number* {**seconds** | **milliseconds**} または **send packet** {**every** *number* {**milliseconds** | **seconds** | **minutes** | **hours**} | **once**} **例：** `RP/0/RSP0/CPU0:router(config-sla-prof-pb)# send burst every 60 seconds packet count 100 interval 100 milliseconds` または `RP/0/RSP0/CPU0:router(config-sla-prof-pb)# send burst once packet count 2 interval 1 second` または `RP/0/RSP0/CPU0:router(config-sla-prof-pb)# send packet every 100 milliseconds`	動作プロファイルのプローブが送信するパケットの数とタイミングを設定します。
ステップ 3	**packet size** *bytes* [**test pattern** {**hex 0x***HHHHHHHH* | **pseudo-random**}] **例：** `RP/0/RSP0/CPU0:router(config-sla-prof-pb)# packet size 9000`	発信プローブ パケットの最小サイズ（バイト単位）を設定します。これには、必要なパディングも含まれます。パディング文字として使用する 16 進文字列、または疑似乱数ビット シーケンスを指定するテスト パターンのキーワードを使用します。デフォルトのパディングは 0 の連続です。パケット サイズは、SLM、ループバック、および DMM/R プローブに対して設定できます。
ステップ 4	**priority** *priority* **例：** `RP/0/RSP0/CPU0:router(config-sla-prof-pb)# priority 7`	発信 SLA プローブ パケットのプライオリティを設定します。

	コマンドまたはアクション	目的
ステップ 5	**synthetic loss calculation packets** *number* **例：** `RP/0/RSP0/CPU0:router(config-sla-prof-pb)# synthetic loss calculation packets 25`	合成損失測定の場合に、1 回の FLR 計算に使用する必要のあるパケット数を設定します。この項目を設定できるのは、合成損失測定をサポートするパケット タイプに対してだけです。 FLR 値は、パケットのブロックごとに計算されます。たとえば、値が 10 と設定されている場合は、最初の FLR 値はパケット 0 ～ 9 に基づいて計算され、2 番目の FLR 値はパケット 10 ～ 19 に基づいて計算され、以下も同様です。
ステップ 6	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-sla-prof-pb)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

プロファイルの SLA 統計情報測定の設定

イーサネット SLA 機能は、一方向および双方向の遅延およびジッター統計情報と、一方向 FLR 統計情報の測定をサポートします。

前提条件

一方向の遅延またはジッター測定を設定するには、最初に **profile（SLA）**コマンドを設定する必要があります。設定するには、このコマンドの **type cfm-delay-measurement** 形式を使用します。

一方向の遅延測定を設定するには、次のクロッキングの前提条件を満たしていることを確認します。

- 周波数の同期化がデフォルトの回線タイミング モードでグローバルに設定されている（**clock-interface timing mode** コマンドは設定されていない）。
- **port-parameters dti** コマンドを使用して、RSP のクロック インターフェイス（Sync 0/Sync 1）が DTI ポートとして設定されている。
- 有効な DTI 入力信号が RSP のクロック インターフェイスのポートで使用できる。
- ローカルおよびリモートの両方のルータが、DTI 入力信号を使用している。

周波数の同期化の設定の詳細については、「Cisco ASR 9000 シリーズ ルータのイーサネット インターフェイスの設定」を参照してください。

制約事項

一方向の遅延およびジッター測定は CFM ループバック プロファイル タイプでサポートされません。

プロファイルの中で SLA 統計情報測定を設定するには、次の手順を SLA プロファイル コンフィギュレーション モードで実行します。

手順の概要

1. **statistics measure** {**one-way-delay-ds** | **one-way-delay-sd** | **one-way-jitter-ds** | **one-way-jitter-sd** | **round-trip-delay** | **round-trip-jitter** | **one-way-loss-ds** | **one-way-loss-sd**}
2. **aggregate** {**bins** *count* **width** *width* | **none**}
3. **buckets size** *number* {**per-probe** | **probes**}
4. **buckets archive** *number*
5. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**statistics measure** {**one-way-delay-ds** | **one-way-delay-sd** | **one-way-jitter-ds** | **one-way-jitter-sd** | **round-trip-delay** | **round-trip-jitter** | **one-way-loss-ds** | **one-way-loss-sd**} **例：** `RP/0/RSP0/CPU0:router(config-sla-prof)# statistics measure round-trip-delay`	SLA 統計情報の収集をイネーブルにして、SLA プロファイル統計情報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	**aggregate** {**bins** *count* **width** *width* | **none**} **例：** `RP/0/RSP0/CPU0:router(config-sla-prof-stat-cfg)# aggregate bins 100 width 10000`	統計情報の収集結果を集約するビンのサイズと数を設定します。 • 遅延測定の場合に、ビン数が 2 以上のときは、幅を 1 ～ 10000（ミリ秒単位）の範囲内で指定する必要があります。 • ジッター測定の場合に、ビン数が 3 以上のときは、幅を 1 ～ 10000（ミリ秒単位）の範囲内で指定する必要があります。 • 損失測定の場合に、ビン数が 2 以上のときは、幅を 1 ～ 100（% 単位）の範囲内で指定する必要があります。
ステップ 3	**buckets size** *number* {**per-probe** | **probes**} **例：** `RP/0/RSP0/CPU0:router(config-sla-prof-stat-cfg)# buckets size 100 per-probe`	統計情報を収集するバケット サイズを設定します。

	コマンドまたはアクション	目的
ステップ 4	**buckets archive** *number* **例：** `RP/0/RSP0/CPU0:router(config-sla-prof-stat-cfg)# buckets archive 50`	メモリに保存するバケット数を設定します。
ステップ 5	**end** または **commit** **例：** `RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

プロファイルの SLA 動作プローブのスケジュールの設定

ここでは、SLA プロファイル内で継続的に SLA 動作プローブのスケジュールを設定する方法について説明します。限定されたオンデマンド SLA 動作のスケジュールを設定する方法の詳細については、「オンデマンド SLA 動作の設定」（P.140）を参照してください。

SLA 動作プローブのスケジュールを設定するには、SLA プロファイル コンフィギュレーション モードから次の手順を実行します。

手順の概要

1. **schedule every week on** *day* [**at** *hh*:*mm*] [**for** *duration* {**seconds** | **minutes** | **hours** | **days** | **week**}]
   または
   schedule every day [**at** *hh*:*mm*] [**for** *duration* {**seconds** | **minutes** | **hours** | **days** | **week**}]
   または
   schedule every *number* {**hours** | **minutes**}[**first at** *hh*:*mm*[.*ss*]] [**for** *duration* {**seconds** | **minutes** | **hours** | **days** | **week**}]
2. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**schedule every week on** *day* [**at** *hh*:*mm*] [**for** *duration* {**seconds** | **minutes** | **hours** | **days** | **week**}] または **schedule every day** [**at** *hh*:*mm*] [**for** *duration* {**seconds** | **minutes** | **hours** | **days** | **week**}] または **schedule every** *number* {**hours** | **minutes**}[**first at** *hh*:*mm*[.*ss*]] [**for** *duration* {**seconds** | **minutes** | **hours** | **days** | **week**}] **例：** `RP/0/RSP0/CPU0:router(config-sla-prof)# schedule every week on Monday at 23:30 for 1 hour` または `RP/0/RSP0/CPU0:router(config-sla-prof)# schedule every day at 11:30 for 5 minutes` または `RP/0/RSP0/CPU0:router(config-sla-prof)# schedule every 2 hours first at 13:45:01` または `RP/0/RSP0/CPU0:router(config-sla-prof)# schedule every 6 hours for 2 hours`	プロファイルの動作プローブをスケジューリングします。1 つのプロファイルには、1 つのスケジュールだけを含めることができます。
ステップ 2	**end** または **commit** **例：** `RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

SLA 動作の設定

ここでは、SLA プロファイルを使用して、MEP で継続中の SLA 動作を設定する方法について説明します。

手順の概要

1. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
2. **ethernet cfm**
3. **mep domain** *domain-name* **service** *service-name* **mep-id** *id-number*
4. **sla operation profile** *profile-name* **target** {**mep-id** *id* | **mac-address** *mac-address*}
5. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config-if)# interface gigabitethernet 0/1/0/1`	物理インターフェイスまたは仮想インターフェイス。 **（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。 ルータ構文の詳細については、疑問符（**?**）オンライン ヘルプ機能を使用します。
ステップ 2	**ethernet cfm** **例：** `RP/0/RSP0/CPU0:router(config-if)# ethernet cfm`	インターフェイス 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	**mep domain** *domain-name* **service** *service-name* **mep-id** *id-number* **例：** `RP/0/RSP0/CPU0:router(config-if-cfm)# mep domain Dm1 service Sv1 mep-id 1`	インターフェイス上で MEP を作成し、インターフェイス 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 4	**sla operation profile** *profile-name* **target** {**mep-id** *id* | **mac-address** *mac-address*} **例：** `RP/0/RSP0/CPU0:router(config-if-cfm-mep)# sla operation profile Profile_1 target mac-address 01:23:45:67:89:ab`	MEP から特定の宛先への動作インスタンスを作成します。
ステップ 5	**end** または **commit** **例：** `RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

オンデマンド SLA 動作の設定

Cisco ASR 9000 シリーズ ルータ では、オンデマンド SLA 動作の設定がサポートされます。これは、必要に応じて、期間限定で実行するためのものです。

この項では、次のトピックについて取り上げます。

- 「設定時の注意事項」（P.140）
- 「CFM の遅延測定のオンデマンド イーサネット SLA 動作の設定」（P.141）
- 「CFM ループバックのオンデマンド イーサネット SLA 動作の設定」（P.142）
- 「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」（P.143）

設定時の注意事項

オンデマンド SLA 動作を設定する場合、次の注意事項に従ってください。

- 各 MEP は最大 50 のオンデマンド動作をサポートします。
- 各カードでは最大 250 のオンデマンド動作をサポートします。

- オンデマンド イーサネット SLA 動作は、設定済みのスケジュールされた他の継続中の SLA 動作に加えて実行でき、同じ量の CPU とルータのメモリを使用します。オンデマンド イーサネット SLA 動作を設定する場合、既存の SLA 動作設定と、通常の動作に対する追加のパケット処理の影響の可能性を考慮する必要があります。
- オンデマンド動作のスケジュールを指定しない場合、プローブのデフォルトは、コマンドの実行から開始 2 秒で 1 回実行され、10 秒間実行されます。
- プローブで測定する統計情報を指定しない場合のデフォルトは、すべての統計情報の測定です。これには、プローブのタイプ別に、次の統計情報が含まれます。
  - CFM ループバック：双方向の遅延およびジッターがデフォルトで測定されます。
  - CFM 遅延測定：双方向の遅延およびジッターの他に、一方向の遅延およびジッターが両方向でデフォルトで測定されます。
  - CFM 合成損失測定：両方の方向の一方向 FLR がデフォルトで測定されます。
- デフォルトの動作モードは同期です。動作の進行状況がコンソールにレポートされ、統計収集の出力が表示されます。

CFM の遅延測定のオンデマンド イーサネット SLA 動作の設定

CFM の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。

コマンド	目的
ethernet sla on-demand operation type cfm-delay-measurement probe [**priority** *number*] [**send** {**packet** {**once** | **every** *number* {**milliseconds** | **seconds** | **minutes** | **hours**}} | **burst** {**once** | **every** *number* {**seconds** | **minutes** | **hours**}} **packet count** *number* **interval** *number* {**milliseconds** | **seconds**}] **domain** *domain-name* **source interface** *type interface-path-id* **target** {**mac-address** *H.H.H.H* | **mep-id** *id-number*} [**statistics measure** {**one-way-delay-ds** | **one-way-delay-sd** | **one-way-jitter-ds** | **one-way-jitter-sd** | **round-trip-delay** | **round-trip-jitter**}][**aggregate** {**none** | **bins** *number* **width** *milliseconds*}] [**buckets** {**archive** *number* | **size** *number* {**per-probe** | **probes**}}] [**schedule** {**now** | **at** *hh*:*mm*[.*ss*] [*day* [*month* [*year*]]] | **in** *number* {**seconds** | **minutes** | **hours**}}[**for** *duration* {**seconds** | **minutes** | **hours**}][**repeat every** *number* {**seconds** | **minutes** | **hours**} **count** *probes*]] [**asynchronous**] **例：** `RP/0/RSP0/CPU0:router# ethernet sla on-demand operation type cfm-delay-measurement probe domain D1 source interface TenGigE 0/6/1/0 target mep-id 100`	CFM の遅延測定のオンデマンド イーサネット SLA 動作を設定します。 次の例は、次のデフォルトを使用して、ローカル ドメイン、および送信元インターフェイスおよびターゲット MEP を指定する最小設定を示します。 • パケット数が 10、間隔が 1 秒のバーストを一度送信します（10 秒プローブ）。 • 出力インターフェイスのデフォルトのサービス クラス（CoS）を使用します。 • 一方向およびラウンドトリップの遅延およびジッター統計情報を含むすべての統計情報を測定します。 • 統計情報を 1 つのビンに集約します。 • すぐにスケジュールします。 • コンソールに結果を表示します。

CFM ループバックのオンデマンド イーサネット SLA 動作の設定

CFM ループバックのオンデマンド 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。

コマンド	目的
ethernet sla on-demand operation type cfm-loopback probe [**packet size** *bytes* [**test pattern** {**hex 0x***HHHHHHHH* | **pseudo-random**}]] [**priority** *number*] [**send** {**packet** {**once** | **every** *number* {**milliseconds** | **seconds** | **minutes** | **hours**}} | **burst** {**once** | **every** *number* {**seconds** | **minutes** | **hours**}} **packet count** *number* **interval** *number* {**milliseconds** | **seconds**}] **domain** *domain-name* **source interface** *type interface-path-id* **target** {**mac-address** *H.H.H.H* | **mep-id** *id-number*} [**statistics measure** {**round-trip-delay** | **round-trip-jitter**}][**aggregate** {**none** | **bins** *number* **width** *milliseconds*}][**buckets** {**archive** *number* | **size** *number* {**per-probe** | **probes**}}] [**schedule** {**now** | **at** *hh*:*mm*[.*ss*] [*day* [*month* [*year*]]] | **in** *number* {**seconds** | **minutes** | **hours**}}[**for** *duration* {**seconds** | **minutes** | **hours**}][**repeat every** *number* {**seconds** | **minutes** | **hours**} **count** *probes*]] [**asynchronous**] **例：** `RP/0/RSP0/CPU0:router# ethernet sla on-demand operation type cfm-loopback probe packet size 1500 domain D1 source interface TenGigE 0/6/1/0 target mep-id 100`	CFM ループバックのオンデマンド イーサネット SLA 動作を設定します。 例では、最小設定を示しますが、最小パケット サイズのオプションを指定し、次のデフォルトを使用してローカル ドメイン、および送信元インターフェイスおよびターゲット MEP を指定します。 • パケット数が 10、間隔が 1 秒のバーストを一度送信します（10 秒プローブ）。 • パディングは、デフォルト テスト パターンである 0 の連続を使用します。 • 出力インターフェイスのデフォルトのサービス クラス（CoS）を使用します。 • すべての統計情報を測定します。 • 統計情報を 1 つのビンに集約します。 • すぐにスケジュールします。 • コンソールに結果を表示します。

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。

コマンド	目的
ethernet sla on-demand operation type cfm-synthetic-loss-measurement probe [**priority** *number*] [**send** {**packet** {**once** | **every** *number* {**milliseconds** | **seconds** | **minutes** | **hours**}} | **burst** {**once** | **every** *number* {**seconds** | **minutes** | **hours**}} **packet count** *number* **interval** *number* {**milliseconds** | **seconds**}] **domain** *domain-name* **source interface** *type interface-path-id* **target** {**mac-address** *H.H.H.H* | **mep-id** *id-number*} [**synthetic loss calculation packets** *number*] [**statistics measure** { **one-way-loss-ds** | **one-way-loss-sd**}][**aggregate** {**none** | **bins** *number* **width** *milliseconds*}] [**buckets** {**archive** *number* | **size** *number* {**per-probe** | **probes**}}] [**schedule** {**now** | **at** *hh*:*mm*[.*ss*] [*day* [*month* [*year*]]] | **in** *number* {**seconds** | **minutes** | **hours**}}[**for** *duration* {**seconds** | **minutes** | **hours**}][**repeat every** *number* {**seconds** | **minutes** | **hours**} **count** *probes*]] [**asynchronous**] **例：** `RP/0/RSP0/CPU0:router# ethernet sla on-demand operation type cfm-synthetic-loss-measurement probe domain D1 source interface TenGigE 0/6/1/0 target mac-address 2.3.4`	CFM 合成損失測定のオンデマンド イーサネット SLA 動作を設定します。 この例が示すのは、最小限の設定です。ローカル ドメインと送信元インターフェイスおよびターゲット MEP を指定しています。

SLA 設定の確認

SLA の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。

コマンド	目的
show ethernet sla configuration-errors [**domain** *domain-name*] [**interface** *interface-path-id*] [**profile** *profile-name*]	設定済み SLA 動作のアクティブ化を妨げているエラーと、発生した警告に関する情報を表示します。
show ethernet sla operations [**detail**] [**domain** *domain-name*] [**interface** *interface-path-id*] [**profile** *profile-name*]	設定済み SLA 動作に関する情報を表示します。

イーサネット LMI の設定

イーサネット LMI を設定するには、次の作業を実行します。

E-LMI の設定の前提条件

Cisco ASR 9000 シリーズ ルータで E-LMI を設定する前に、次の要件を実行してください。

- E-LMI を実行するネットワークのローカルおよびリモート UNI を特定し、その命名規則を定義します。
- E-LMI CE 動作をサポートする Cisco Catalyst 3750 Metro シリーズ スイッチなどのデバイス上で対応する CE インターフェイス リンクの E-LMI をイネーブルにします。

E-LMI の設定に関する制約事項

E-LMI を設定する場合、次の制約事項を考慮してください。

- E-LMI はサブインターフェイスまたはバンドル インターフェイスでサポートされません。E-LMI は、イーサネットの物理インターフェイスでのみ設定できます。

E-LMI の EVC の作成

Cisco ASR 9000 シリーズ ルータでの E-LMI の EVC は、E-LMI が実行される CE へのローカル UNI の物理イーサネット インターフェイス リンク、およびリモート UNI リンクの EFP（レイヤ 2 サブインターフェイス）を最初に設定して確立されます。次に、EFP を EVC を作成する L2VPN ブリッジ ドメインに割り当てる必要があります。

EVC を作成するには、次の作業を実行します。

- 「EFP の設定」（P.145）（必須）
- 「ブリッジ グループの設定およびブリッジ ドメインへの EFP の割り当て」（P.147）（必須）

EFP の設定

ここでは、EFP の基本設定について説明します。サポートされているその他のレイヤ 2 サービスの設定の詳細については、『*Cisco ASR 9000 Series Aggregation Services Routers L2VPN and Ethernet Services Configuration Guide*』を参照してください。

EFP を設定するには、次のタスクを実行します。

手順の概要

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id.subinterface* **l2transport**
3. **encapsulation dot1q** *vlan-id* [**, untagged** | **,** *vlan-id* | *–vlan-id*] [**exact** | **ingress source-mac** *mac-address* | **second-dot1q** *vlan-id*]
4. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id.subinterface* **l2transport** **例：** `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/0/0/0.0 l2transport`	レイヤ 2 転送モードの VLAN サブインターフェイスを作成し、レイヤ 2 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	**encapsulation dot1q** *vlan-id* [**, untagged** | **,** *vlan-id* | **-***vlan-id*] [**exact** | **ingress source-mac** *mac-address* | **second-dot1q** *vlan-id*] **例：** `RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 1-20`	インターフェイスの 802.1Q フレーム入力を適切なサービス インスタンスにマップするための一致基準を定義します。
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-subif)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ブリッジ グループの設定およびブリッジ ドメインへの EFP の割り当て

ブリッジ グループを設定し、EVC を作成するためにブリッジ ドメインに EFP を割り当てるには、次の手順を実行します。

手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *name*
4. **bridge-domain** *name*
5. **interface** {**GigabitEthernet** | **TenGigE**} *interface-path-id.subinterface*
6. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**l2vpn** **例：** `RP/0/RSP0/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/RSP0/CPU0:router(config-l2vpn)# bridge group BG1`	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/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain BD1`	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。

	コマンドまたはアクション	目的
ステップ 5	**interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id.subinterface* **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface GigabitEthernet 0/0/0/0.0`	EFP（EVC）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。ここで、*interface-path-id* はインターフェイスの *rack/slot/module/port* ロケーションとして指定し、*.subinterface* はサブインターフェイス番号です。 ブリッジ ドメインに関連付けようとする EFP（EVC）の数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/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# end` または `RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

E-LMI のイーサネット CFM の設定

Cisco ASR 9000 シリーズ ルータは、E-LMI の EVC のステータスをモニタするためにイーサネット CFM を使用します。E-LMI に CFM を使用するには、CFM メンテナンス ドメインとサービスをルータで設定し、EFP を CFM のアップ MEP として設定する必要があります。

E-LMI のイーサネット CFM を設定するには、次の作業を実行します。

- 「イーサネット CFM の設定」（P.109）（必須）
- 「EFP を CFM のアップ MEP として設定する」（P.149）（必須）

イーサネット CFM の設定

イーサネット CFM を使用する E-LMI をサポートするための最小設定は、ルータの CFM メンテナンス ドメインおよびサービスの設定です。その他の CFM オプションも設定できます。

イーサネット CFM を設定するタスクの詳細については、「イーサネット CFM の設定」（P.109）を参照してください。

EFP を CFM のアップ MEP として設定する

ここでは、CFM MEP として EFP を設定するために必要な最小限の作業について説明します。CFM MEP の設定の詳細については、「CFM MEP の設定」（P.121）を参照してください。

CFM MEP として EFP を設定するには、各 E-LMI EFP に対して次の作業を実行します。

手順の概要

1. **configure**
2. **interface** {**GigabitEthernet** | **TenGigE**} *interface-path-id*.*subinterface*
3. **ethernet cfm**
4. **mep domain** *domain-name* **service** *service-name* **mep-id** *id-number*
5. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface gigabitethernet** *interface-path-id*.*subinterface* **l2transport** **例：** `RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/0/0/0.0 l2transport`	EFP のレイヤ 2 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	**ethernet cfm** **例：** `RP/0/RSP0/CPU0:router(config-subif)# 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。

	コマンドまたはアクション	目的
ステップ 4	**mep domain** *domain-name* **service** *service-name* **mep-id** *id-number* **例：** `RP/0/RSP0/CPU0:router(config-if-cfm)# mep domain GLOBAL service CustomerA mep-id 22`	インターフェイス上で MEP を作成し、インターフェイス 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 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if-cfm-mep)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

物理インターフェイスの UNI 名の設定

E-LMI プロトコルの管理に役立つように、ローカルおよびリモート UNI への物理インターフェイス リンクの UNI 名を設定することを推奨します。UNI 名を設定するには、ローカルおよびリモート UNI への物理インターフェイス リンクで次の作業を実行します。

手順の概要

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
3. **ethernet uni id** *name*
4. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/0/0/0`	物理インターフェイスのインターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ethernet uni id** *name* **例：** `RP/0/RSP0/CPU0:router(config-if)# ethernet uni id PE1-CustA-Slot0-Port0`	イーサネット UNI インターフェイス リンクの名前（最大 64 文字）を指定します。
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

物理インターフェイスで E-LMI のイネーブル化

Cisco ASR 9000 シリーズ ルータは、物理イーサネット インターフェイス上だけで E-LMI プロトコルをサポートします。E-LMI をイネーブルにするには、ローカル UNI の物理イーサネット インターフェイス リンクで次の作業を実行します。

手順の概要

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
3. **ethernet lmi**
4. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# interface gigabitethernet 0/0/0/0`	物理インターフェイスのインターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ethernet lmi** **例：** `RP/0/RSP0/CPU0:router(config-if)# ethernet lmi`	インターフェイスでイーサネット ローカル管理インターフェイス動作をイネーブルにして、インターフェイス イーサネット LMI 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	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if-lmi)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ポーリング検証タイマーの設定

MEF T392 ポーリング検証タイマー（PVT）は、エラーを記録する前に、ステータス メッセージが送信されてから UNI-C のステータス問い合わせが受信されるまでの許容時間を指定します。デフォルト値は 15 秒です。

デフォルト値を変更またはすべての PVT をディセーブルにするには、次の作業を実行します。

手順の概要

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
3. **ethernet lmi**
4. **polling-verification-timer** {*interval* | **disable**}
5. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** [**GigabitEthernet**	**TenGigE**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# interface gigabitethernet 0/0/0/0`	物理インターフェイスのインターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ethernet lmi** **例：** `RP/0/RSP0/CPU0:router(config-if)# ethernet lmi`	インターフェイスでイーサネット ローカル管理インターフェイス動作をイネーブルにして、インターフェイス イーサネット LMI コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 4	**polling-verification-timer** {*interval* | **disable**} **例：** `RP/0/RSP0/CPU0:router(config-if-lmi)# polling-verification-timer 30`	エラーを記録する前に、ステータス メッセージが送信されてから UNI-C のステータス問い合わせが受信されるまでの許容時間（秒単位）を指定する、E-LMI 動作の MEF T392 ポーリング検証タイマーを設定またはディセーブルにします。デフォルト値は 15 です。
ステップ 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if-lmi)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ステータス カウンタの設定

連続する有効なパケットの受信またはパケットの PVT の連続的な期限切れを追跡して、E-LMI 動作ステータスを決定するために MEF N393 ステータス カウンタ値が使用されます。デフォルト カウンタは 4 です。これは、E-LMI プロトコルがダウン状態の間、プロトコルがアップ状態に変わるには、4 つの有効なパケットを連続して受信するか、E-LMI プロトコルがアップ状態の間にインターフェイスで E-LMI プロトコルがダウンに変わる前に連続して 4 回 PVT の期限切れが発生する必要があることを示します。

ステータス カウンタのデフォルト値を変更するには、次の作業を実行します。

手順の概要

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
3. **ethernet lmi**
4. **status-counter** *threshold*
5. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# interface gigabitethernet 0/0/0/0`	物理インターフェイスのインターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ethernet lmi** **例：** `RP/0/RSP0/CPU0:router(config-if)# ethernet lmi`	インターフェイスでイーサネット ローカル管理インターフェイス動作をイネーブルにして、インターフェイス イーサネット LMI 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	**status-counter** *threshold* **例：** `RP/0/RSP0/CPU0:router(config-if-lmi)# status-counter 5`	ピアからの連続した有効および無効パケットの受信を追跡して、E-LMI 動作ステータスの判別に使用する MEF N393 ステータス カウンタ値を設定します。デフォルトは 4 です。
ステップ 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if-lmi)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

E-LMI エラーまたはイベントの syslog メッセージのディセーブル化

E-LMI プロトコルは、特定のエラーおよびイベントを追跡し、カウントは **show ethernet lmi interfaces** コマンドを使用して表示できます。

E-LMI エラーまたはイベントの syslog メッセージをディセーブルにするには、次の作業を実行します。

手順の概要

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
3. **ethernet lmi**
4. **log** {**errors** | **events**} **disable**
5. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# interface gigabitethernet 0/0/0/0`	物理インターフェイスのインターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ethernet lmi** **例：** `RP/0/RSP0/CPU0:router(config-if)# ethernet lmi`	インターフェイスでイーサネット ローカル管理インターフェイス動作をイネーブルにして、インターフェイス イーサネット LMI コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 4	**log** {**errors** | **events**} **disable** **例：** `RP/0/RSP0/CPU0:router(config-if-lmi)# log events disable`	E-LMI エラーまたはイベントの syslog メッセージをオフにします。
ステップ 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if-lmi)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

シスコ独自のリモート UNI 詳細情報要素の使用のディセーブル化

E-LMI プロトコル内で追加情報を指定するには、Cisco IOS XR ソフトウェアで、リモート UNI 名および状態に関する情報を CE に送信する、リモート UNI 詳細と呼ばれるシスコ独自の情報要素を実装します。この情報要素により、E-LMI MEF 16 仕様では現在未使用の ID が組み込まれます。

リモート UNI 詳細情報要素の使用をディセーブルにするには、次の作業を実行します。

手順の概要

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
3. **ethernet lmi**
4. **extension remote-uni disable**
5. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# interface gigabitethernet 0/0/0/0`	物理インターフェイスのインターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ethernet lmi** **例：** `RP/0/RSP0/CPU0:router(config-if)# ethernet lmi`	インターフェイスでイーサネット ローカル管理インターフェイス動作をイネーブルにして、インターフェイス イーサネット LMI 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	**extension remote-uni disable** **例：** `RP/0/RSP0/CPU0:router(config-if-lmi)# extension remote-uni disable`	E-LMI ステータス メッセージでのシスコ独自のリモート UNI 詳細情報要素の送信をディセーブルにします。
ステップ 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if-lmi)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

イーサネット LMI の設定の確認

show ethernet lmi interfaces detail コマンドを使用して、特定のインターフェイスまたはすべてのインターフェイスのイーサネット LMI の設定の値を表示できます。次の例は、コマンドのサンプル出力を示しています。

```
RP/0/RSP0/CPU0:router# show ethernet lmi interfaces detail
Interface: GigabitEthernet0/0/0/0
  Ether LMI Link Status: Up
  UNI Id: PE1-CustA-Slot0-Port0
  Line Protocol State: Up
  MTU: 1514 (1 PDU reqd.for full report)
  CE-VLAN/EVC Map Type: Bundling (1 EVC)
  Configuration: Status counter 4, Polling Verification Timer 15 seconds
  Last Data Instance Sent: 0
  Last Sequence Numbers: Sent 0, Received 0

  Reliability Errors:
    Status Enq Timeouts                 0 Invalid Sequence Number             0
    Invalid Report Type                 0

  Protocol Errors:
    Malformed PDUs                      0 Invalid Procotol Version            0
    Invalid Message Type                0 Out of Sequence IE                  0
    Duplicated IE                       0 Mandatory IE Missing                0
    Invalid Mandatory IE                0 Invalid non-Mandatory IE            0
    Unrecognized IE                     0 Unexpected IE                       0

  Full Status Enq Received     never       Full Status Sent         never
  PDU Received                 never       PDU Sent                 never
  LMI Link Status Changed   00:00:03 ago   Last Protocol Error      never
  Counters cleared             never

  Sub-interface: GigabitEthernet0/0/0/0.0
    VLANs: 1-20
    EVC Status: Active
    EVC Type: Point-to-Point
    OAM Protocol: CFM
      CFM Domain: Global (level 5)
      CFM Service: CustomerA
    Remote UNI Count: Configured = 1, Active = 1

    Remote UNI Id                                                 Status
    -------------                                                 ------
    PE1-CustA-Slot0-Port1                                         Up
```

E-LMI 設定のトラブルシューティングのヒント

ここでは、次の内容で E-LMI 設定のトラブルシューティングに関する基本的な情報について説明します。

- 「イーサネット LMI のリンク ステータスのトラブルシューティング」（P.161）
- 「イーサネット LMI ライン プロトコル ステートのトラブルシューティング」（P.161）
- 「イーサネット LMI エラー カウンタのトラブルシューティング」（P.162）
- 「イーサネット LMI リモート UNI のトラブルシューティング」（P.162）

イーサネット LMI のリンク ステータスのトラブルシューティング

E-LMI プロトコルの動作ステータスは、**show ethernet lmi interfaces** コマンドの形式の出力の「Ether LMI Link Status」フィールドおよび「ELMI state」フィールドで報告されます。「Up」以外のリンク ステータスを調査するには、次のガイドラインを考慮してください。

- Unknown (PVT disabled)：ポーリング検証タイマーが無効として設定されているため、ステータス情報が提供できないことを示します。ステータスが「Up」か「Down」かを知るには、PVT をイネーブルにする必要があります。詳細については、「ポーリング検証タイマーの設定」（P.153）を参照してください。
- Down：E-LMI のリンク ステータスは、次の理由でダウンしている可能性があります。
  - PVT が **status-counter** コマンドで指定された回数、タイム アウトになっています。これは、ステータス問い合わせメッセージを CE デバイスから受信していないことを示します。これには次の原因が考えられます。

    — CE デバイスが PE デバイスに接続されていない。PE デバイスで E-LMI が有効なインターフェイスに CE デバイスが接続されていることを確認します。

    — CE デバイスがステータス問い合わせを送信していない。PE デバイスに接続している CE インターフェイスで E-LMI がイネーブルになっていることを確認します。

    — プロトコル エラーが PVT の期限切れの原因になっている。PVT は、有効な（エラーになっていない）ステータス問い合わせメッセージを受信するとリセットされるだけです。
  - ライン プロトコル ステートが「Down」または「Admin Down」になっています。
  - EVC に関する UNI Id または詳細など、配信する有用な情報がないため、インターフェイスでプロトコルがまだ開始されていません。これはプロビジョニング設定ミスの現象です。

（注） プロトコルが開始すると、E-LMI は「Down」状態であれば、引き続きステータス問い合わせメッセージに応答します。

イーサネット LMI ライン プロトコル ステートのトラブルシューティング

E-LMI ライン プロトコル ステートは、**show ethernet lmi interfaces** コマンドの形式の出力の「Line Protocol State」フィールドまたは「LineP State」フィールドで報告されます。ライン プロトコル ステートは、物理インターフェイスの E-LMI プロトコルのステートです。

アップ以外のライン プロトコル ステートを確認するには、次の注意事項に従ってください。

- Admin-Down：インターフェイスは **shutdown** コマンドを使用して設定します。インターフェイスをアップにするには、**no shutdown** コマンドを使用します。
- Down：インターフェイスの障害を示します。詳細については、インターフェイスの状態およびインターフェイス ライン プロトコル ステートの両方の詳細を表示するには、**show interfaces** コマンドを実行し、さらに調査するために次の操作を行います。
  - 状態が両方ともダウンしている場合、リンクの物理的な問題を提示しています（たとえば、ケーブルが PE または CE デバイスに接続されていないなど）。
  - インターフェイスの状態はアップで、ライン プロトコル ステートがダウンの場合、OAM プロトコルが障害によってライン プロトコル ステートをダウンにしていることを提示します。詳細を参照するには、**show efd interface** コマンドを使用します。

イーサネット LMI エラー カウンタのトラブルシューティング

show ethernet lmi interfaces コマンドは次の 2 つの項目のエラー カウンタを表示します。

- 信頼性エラー：メッセージが PE および CE デバイス間で失われていることを示します。出力の最後のブロックのタイマーは、メッセージが PE デバイスによって送受信中であることを示す必要があります。
- プロトコル エラー：CE デバイスが PE デバイスにパケットを送信しているが、PE は、これらのパケットを認識しないことを示します。これは、CE 側の E-LMI プロトコルが正しく設定されていない、または CE と PE 間のパス上のパケットの破損を提示しています。E-LMI パケットに MEF 16 標準で正確に定義された構造があり、そこからの逸脱がプロトコル エラーを発生させます。PE は、形式が誤っていて、プロトコル エラーを引き起こすパケットには応答しません。

E-LMI を設定した直後に、すべてのエラー カウンタはゼロになりますが、ステータス問い合わせのタイムアウト カウンタは例外になる可能性があります。ステータス問い合わせのタイムアウト カウンタは、対応する CE インターフェイスで開始する前に、PE インターフェイスで E-LMI プロトコルが開始された場合、ゼロになります。ただし、プロトコルが両方のデバイスで開始されると、このカウンタの増加を止める必要があります。

ステータス問い合わせのタイムアウト カウンタがゼロ以外の場合、問い合わせを CE デバイスから受信していないことを示します。これは、次のような状態が原因の可能性があります。

- CE デバイスが接続されていないか、ステータス問い合わせメッセージを送信していない。詳細については、「イーサネット LMI のリンク ステータスのトラブルシューティング」（P.161）も参照してください。
- CE デバイスのポーリング タイマーは、PE デバイスの PVT より大きい値に設定されている。PE デバイスで **polling-verification-timer** コマンドの値が CE のポーリング タイマーの値よりも大きいことを確認します。

詳細については、『*Cisco ASR 9000 Aggregation Services Router Interfaces and Hardware Component Command Reference*』の show ethernet lmi interfaces コマンドのマニュアルも参照してください。

イーサネット LMI リモート UNI のトラブルシューティング

リモート UNI に関する情報は、**show ethernet lmi interfaces detail** コマンドの出力で報告されます。リモート UNI ID フィールドは、**ethernet uni id** コマンドによって設定される UNI 名を表示するか、UNI 名が設定されていない場合は UNI の CFM MEP ID を表示します。

リモート UNI がテーブルまったくない場合、これは、次のような状態が原因の可能性があります。

- リモート UNI の EFP が L2VPN 設定のブリッジ ドメインから失われている。**show ethernet cfm configuration-errors** コマンドを使用して設定を確認します。
- CFM MEP がリモート UNI の EFP に設定されていない。

UDLD の設定

UDLD は、インターフェイスごとに設定されます。インターフェイスは、物理イーサネット インターフェイスでなければなりません。

インターフェイスに対して UDLD プロトコルを設定するには、次の手順を実行します。

手順の概要

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*

3. **ethernet udld**
4. **mode {normal | aggressive}**
5. **message-time**
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/1/0/0`	インターフェイス コンフィギュレーション モードを開始し、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。 **（注）** この例は、モジュラ サービス カード スロット 1 の 8 ポート 10 ギガビット イーサネット インターフェイスです。
ステップ 3	**ethernet udld** **例：** `RP/0/RSP0/CPU0:router(config-if)# ethernet udld`	イーサネット UDLD 機能をイネーブルにし、インターフェイス イーサネット UDLD 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	**mode {normal |aggressive}** **例：** `RP/0/RSP0/CPU0:router(config-if-udld)# mode normal`	（任意）UDLD の動作モードを指定します。オプションは normal と aggressive です。
ステップ 5	**message-time** *[7-90]* **例：** `RP/0/RSP0/CPU0:router(config-if-udld)# message-time 70`	（任意）UDLD プロトコルに使用するメッセージ時間を秒単位で指定します。値の範囲は、7 ～ 90 秒です。
ステップ 6	**logging disable** **例：** `RP/0/RSP0/CPU0:router(config-if-udld)# loggig disable`	（任意）このコマンドは動作中の UDLD syslog メッセージを抑制します。
ステップ 7	**end** **例：** `RP/0/RSP0/CPU0:router(config-if-udld)# end`	コンフィギュレーション セッションを終了し、EXEC モードに戻ります。

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。

EOAM インターフェイスの設定例

ここでは、次の設定例について説明します。

イーサネット OAM プロファイルのグローバルな設定：例

次に、イーサネット OAM プロファイルをグローバルに設定する例を示します。

```
configure terminal
 ethernet oam profile Profile_1
  link-monitor
   symbol-period window 60000
   symbol-period threshold low 10000000 high 60000000
   frame window 60
   frame threshold low 10000000 high 60000000
   frame-period window 60000
   frame-period threshold low 100 high 12000000
   frame-seconds window 900000
   frame-seconds threshold 3 threshold 900
   exit
  mib-retrieval
  connection timeout 30
  require-remote mode active
  require-remote link-monitoring
  require-remote mib-retrieval
  action dying-gasp error-disable-interface
  action critical-event error-disable-interface
  action discovery-timeout error-disable-interface
  action session-down error-disable-interface
  action capabilities-conflict error-disable-interface
  action wiring-conflict error-disable-interface
  action remote-loopback error-disable-interface
  commit
```

個々のインターフェイスでのイーサネット OAM 機能の設定：例

次に、個々のインターフェイス上でイーサネット OAM 機能を設定する例を示します。

```
configure terminal
 interface TenGigE 0/1/0/0
  ethernet oam
   link-monitor
    symbol-period window 60000
    symbol-period threshold low 10000000 high 60000000
    frame window 60
    frame threshold low 10000000 high 60000000
    frame-period window 60000
    frame-period threshold low 100 high 12000000
    frame-seconds window 900000
    frame-seconds threshold 3 threshold 900
    exit
   mib-retrieval
   connection timeout 30
   require-remote mode active
   require-remote link-monitoring
   require-remote mib-retrieval
   action link-fault  error-disable-interface
   action dying-gasp error-disable-interface
   action critical-event error-disable-interface
   action discovery-timeout error-disable-interface
   action session-down error-disable-interface
   action capabilities-conflict error-disable-interface
   action wiring-conflict error-disable-interface
   action remote-loopback error-disable-interface
   commit
```

個々のインターフェイスでプロファイルを上書きするためのイーサネット OAM 機能の設定：例

次に、イーサネット OAM 機能を設定し、次にインターフェイスでその設定を上書きする例を示します。

```
configure terminal
 ethernet oam profile Profile_1
  mode passive
  action dying-gasp disable
  action critical-event disable
  action discovery-timeout disable
  action session-up disable
  action session-down disable
  action capabilities-conflict disable
  action wiring-conflict disable
  action remote-loopback disable
  action uni-directional link-fault error-disable-interface
  commit

configure terminal
 interface TenGigE 0/1/0/0
  ethernet oam
   profile Profile_1
    mode active
    action dying-gasp log
    action critical-event log
    action discovery-timeout log
    action session-up log
    action session-down log
```

```
    action capabilities-conflict log
    action wiring-conflict log
    action remote-loopback log
    action uni-directional link-fault log
    uni-directional link-fault detection
    commit
```

イーサネット OAM ピアのリモート ループバックの設定：例

次に、イーサネット OAM ピアのリモート ループバックを設定する例を示します。

```
configure terminal
 interface gigabitethernet 0/1/5/6
  ethernet oam
   profile Profile_1
   remote-loopback
```

次に、設定済みのイーサネット OAM インターフェイスでリモート ループバックを開始する例を示します。

```
ethernet oam loopback enable TenGigE 0/6/1/0
```

インターフェイスのイーサネット OAM 統計情報のクリア：例

次に、インターフェイスのイーサネット OAM 統計情報をクリアする例を示します。

```
RP/0/RP0/CPU0:router# clear ethernet oam statistics interface gigabitethernet 0/1/5/1
```

ルータの SNMP サーバ トラップのイネーブル化：例

次に、ルータの SNMP サーバ トラップをイネーブルにする例を示します。

```
configure terminal
  ethernet oam profile Profile_1
  snmp-server traps ethernet oam events
```

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」（P.167）
- 「イーサネット CFM サービスの設定：例」（P.167）
- 「イーサネット CFM サービス設定の柔軟なタギング：例」（P.167）
- 「イーサネット CFM サービス設定の連続性チェック：例」（P.167）
- 「イーサネット CFM サービス設定の MIP の作成：例」（P.167）
- 「イーサネット CFM サービス設定のクロスチェック：例」（P.167）
- 「他のイーサネット CFM サービス パラメータの設定：例」（P.168）
- 「MEP の設定：例」（P.168）
- 「イーサネット CFM の show コマンド：例」（P.168）
- 「CFM 設定の AIS：例」（P.171）
- 「CFM の show コマンドの AIS：例」（P.172）

- 「EFD 設定：例」（P.175）
- 「EFD 情報の表示：例」（P.176）

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 サービスの MIP の自動作成をイネーブルにする例を示します。

```
   mip auto-create all
   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 の MEP を設定する例を示します。

```
interface gigabitethernet 0/1/0/1
 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/RSP0/CPU0:router# show ethernet cfm local maintenance-points

Domain/Level         Service             Interface         Type   ID   MAC
-------------------- ------------------- ----------------- ------ ---- --------
fig/5                bay                 Gi0/10/0/12.23456 Dn MEP    2 44:55:66
fig/5                bay                 Gi0/0/1/0.1       MIP         55:66:77
fred/3               barney              Gi0/1/0/0.1       Up 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/RSP0/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 GigabitEthernet0/1/2/3.234 and an
Up MEP is also configured for domain blort, which is at the same level (5).
 * A MEP is configured on interface GigabitEthernet0/3/2/1.1 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）の動作ステートを表示する例を示します。

```
RP/0/RSP0/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 Gi1/1/0/1.234 (Up)       Up     0/0   N  A       L7

Domain fred (level 5), Service barney
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
    2 Gi0/1/0/0.234 (Up)       Up     3/2   Y  RPC     L6
```

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）の動作ステートを表示する例を示します。

```
RP/0/RSP0/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
Up MEP on GigabitEthernet0/1/0/0.234, 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 GigabitEthernet0/10/0/12.123, 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）の動作ステートを詳細に表示する例を示します。

```
RP/0/RSP0/CPU0:router# show ethernet cfm peer meps detail
Domain dom3 (level 5), Service ser3
Down MEP on GigabitEthernet0/0/0/0 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 GigabitEthernet0/0/0/0 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


Domain dom5 (level 2), Service ser5
Up MEP on Standby Bundle-Ether 1 MEP-ID 1
================================================================================
Peer MEP-ID 600, MAC 0001.0203.0401
   CFM state: Ok (Standby), for 00:00:08, RDI received
   Port state: Down
   CCM defects detected:    Defects below ignored on local standby MEP
                            I - Wrong Interval
                            R - Remote Defect received
   CCMs received: 5
     Out-of-sequence:             0
     Remote Defect received:   5
     Wrong Level:                0
     Cross-connect W(wrong MAID): 0
```

```
      Wrong Interval:              5
      Loop (our MAC received):     0
      Config (our ID received):    0
    Last CCM received 00:00:08 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

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
     Cross-connect (wrong MAID):  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

次に、CFM ドメイン サービスのアラーム表示信号（AIS）の送信を設定する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# ethernet cfm
RP/0/RSP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RSP0/CPU0:router(config-cfm-dmn)# service S1 bridge group BG1 bridge-domain BD2
RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# ais transmission interval 1m cos 7
```

例 2

次に、AIS または LCK パケットを受信したときに示すように、接続障害管理（CFM）の AIS ロギングを設定する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# ethernet cfm
RP/0/RSP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RSP0/CPU0:router(config-cfm-dmn)# service S2 bridge group BG1 bridge-domain BD2
RP/0/RSP0/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/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/2
RP/0/RSP0/CPU0:router(config-if)# ethernet cfm
RP/0/0RP0RSP0/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」（P.173）
- 「show ethernet cfm local meps コマンド：例」（P.173）

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/RSP0/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
--------------------- ---  - -------  -------  - --- ------------ ---------
Gi0/1/0/0.234 (Up)    Dn   5 RPC      6        7 1s  01:32:56 ago     5576
Gi0/1/0/0.567 (Up)    Up   0 M        2,3      5 1s  00:16:23 ago      983
Gi0/1/0/1.1 (Dn)      Up     D                7 60s 01:02:44 ago     3764
Gi0/1/0/2 (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/RSP0/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 Gi1/1/0/1.234 (Up)      Up     0/0   N  A       7

Domain fred (level 5), Service barney
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ----------------------- --- -------- -- ------- ---
    2 Gi0/1/0/0.234 (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/RSP0RP0/CPU0:router# show ethernet cfm local meps domain foo service bar detail

Domain foo (level 6), Service bar
Up MEP on GigabitEthernet0/1/0/0.234, 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
Up MEP on GigabitEthernet0/1/0/0.234, 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 3：Verbose

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

（注） 廃棄された CCM フィールドは、数値がゼロ（0）の場合は表示されません。これは、ピア MEP 数の制限に達したときだけ CCM が廃棄されるため、廃棄された CCM のカウントがゼロ以外のものになることは一般的ではありません。

```
RP/0/RSP0RP0/CPU0:router# show ethernet cfm local meps domain foo service bar verbose

Domain foo (level 6), Service bar
Up MEP on GigabitEthernet0/1/0/0.234, 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)

  Packet        Sent      Received
  ------  ----------  ----------------------------------------------------
  CCM              0             0  (out of seq: 0)
  LBM              0             0
  LBR              0             0  (out of seq: 0, with bad data: 0)
  AIS           5576             0
  LCK              -             0

Domain fred (level 5), Service barney
Up MEP on GigabitEthernet0/1/0/0.234, 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
```

```
  Packet        Sent      Received
  ------  ----------  ----------------------------------------------------------
  CCM          12345         67890  (out of seq: 6, discarded: 10)
  LBM              5             0
  LBR              0             5  (out of seq: 0, with bad data: 0)
  AIS              0         46910
  LCK              -             0
```

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/RSP0/CPU0:router# show ethernet cfm local meps detail

Domain foo (level 6), Service bar
Up MEP on GigabitEthernet0/1/0/0.234, 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
Up MEP on GigabitEthernet0/1/0/0.234, 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
```

EFD 設定：例

次に、EFD をイネーブルにする例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# ethernet cfm
RP/0/RSP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RSP0/CPU0:router(config-cfm-dmn)# service S1 down-meps
RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# efd
```

次に、EFD ロギングをイネーブルにする例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# ethernet cfm
RP/0/RSP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RSP0/CPU0:router(config-cfm-dmn)# service S1 down-meps
RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# log efd
```

EFD 情報の表示：例

次に、EFD に関する情報を表示する例を示します。

- 「show efd interfaces コマンド：例」（P.176）
- 「show ethernet cfm local meps detail コマンド：例」（P.176）

show efd interfaces コマンド：例

次に、EFD アクションに応じてシャットダウンされたインターフェイスをすべて表示する例を示します。

```
RP/0/RSP0/CPU0:router# show efd interfaces

Server VLAN MA
==============
Interface          Clients
-------------------------
GigE0/0/0/0.0      CFM
```

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/RSP0/CPU0:router# show ethernet cfm local meps detail

Domain foo (level 6), Service bar
Up MEP on GigabitEthernet0/1/0/0.234, 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
Up MEP on GigabitEthernet0/1/0/0.234, 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） **show interfaces** および **show interfaces brief** コマンドを使用しても、EFD がインターフェイスでトリガーされた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。

イーサネット SLA 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 SLA プロファイル タイプの設定：例

次の例では、イーサネット SLA でサポートされるさまざまなプロファイル タイプの設定方法を示します。

例 1

この例では、「Prof1」という名前のプロファイルを CFM ループバック測定用に設定する方法を示します。

```
configure
 ethernet sla
  profile  Prof1 type cfm-loopback
   commit
```

例 2

この例では、「Prof1」という名前のプロファイルを CFM 遅延測定用に設定します。このタイプの設定は、追加の一方向の遅延およびジッターの統計情報を測定するようにプローブを設定できます。

```
configure
 ethernet sla
  profile  Prof1 type cfm-delay-measurement
   commit
```

イーサネット SLA プローブの設定：例

次の例では、イーサネット CFM ループバック プローブのパケット オプションを設定する方法を示します。

例 1

次に、100 ミリ秒間隔で 100 個のパケット グループの送信を設定し、そのバーストを 60 秒ごとに繰り返す例を示します。パケットは、サイズが 9000 バイトになるように、必要に応じてパディングされます。パディングには、16 進数テスト パターン「abcdabcd」を使用します。サービス クラス値は 7 です。

（注） バーストの全体の長さ（パケット カウントに間隔値を乗じる）が 1 分を超えてはなりません。

```
configure
 ethernet sla
  profile  Prof1 type cfm-loopback
   probe
    send burst every 60 seconds packet count 100 interval 100 milliseconds
    packet size 9000 test pattern hex 0xabcdabcd
    priority 7
    commit
```

例 2

次の例は、例 1 の設定と同じ特性がありますが、単一バーストで 50 パケットを 1 秒ごとに送信します。

```
configure
 ethernet sla
  profile  Prof1 type cfm-loopback
   probe
    send burst once packet count 50 interval 1 second
    packet size 9000 test pattern hex 0xabcdabcd
    priority 7
    commit
```

例 3

次に、プローブ中に 100 ミリ秒間隔でパケットの連続ストリームを設定する例を示します。パケットは疑似乱数テスト パターンを使用して必要に応じて 9000 バイトのサイズにパディングされます。サービス クラス値は 7 です。

```
configure
 ethernet sla
  profile Prof1 type cfm-loopback
   probe
    send burst every 60 seconds packet count 600 interval 100 milliseconds
    packet size 9000 test pattern pseudo-random
    priority 7
    commit
```

プロファイル統計情報測定の設定：例

次の例では、さまざまなタイプの統計情報測定を設定する方法を示します。

例 1

次に、CFM ループバック SLA プロファイル タイプによって測定できる 2 種類の使用可能な統計情報の例を示します。

```
configure
 ethernet sla
  profile  Prof1 type cfm-loopback
  statistics measure round-trip-delay
  statistics measure round-trip-jitter
   commit
```

例 2

次に、CFM 遅延測定の SLA プロファイル タイプのラウンドトリップ遅延、一方向ジッター（宛先から送信元の方向）の測定を設定する例を示します。

（注） CFM 遅延測定プロファイル タイプはすべてのラウンドトリップおよび一方向の遅延およびジッター統計情報の測定をサポートします。

```
configure
 ethernet sla
  profile  Prof1 type cfm-delay-measurement
  statistics measure round-trip-delay
  statistics measure one-way-jitter-ds
   commit
```

スケジュールされた SLA 動作プローブ設定：例

次の例では、SLA 動作プローブに対してさまざまなスケジュールを設定する方法を示します。

例 1

次に、指定された期間、時間単位で実行するようにプローブを設定する例を示します。

```
configure
 ethernet sla
  profile  Prof1 type cfm-delay-measurement
  schedule every 1 hours for 15 minutes
  commit
```

例 2

次に、指定した期間中、毎日実行するようにプローブを設定する例を示します。

```
configure
 ethernet sla
  profile  Prof1 type cfm-delay-measurement
  schedule every day at 11:30 for 5 minutes
  commit
```

例 3

次に、指定期間中、週単位で実行し、指定された時刻に開始するようにプローブを設定する例を示します。

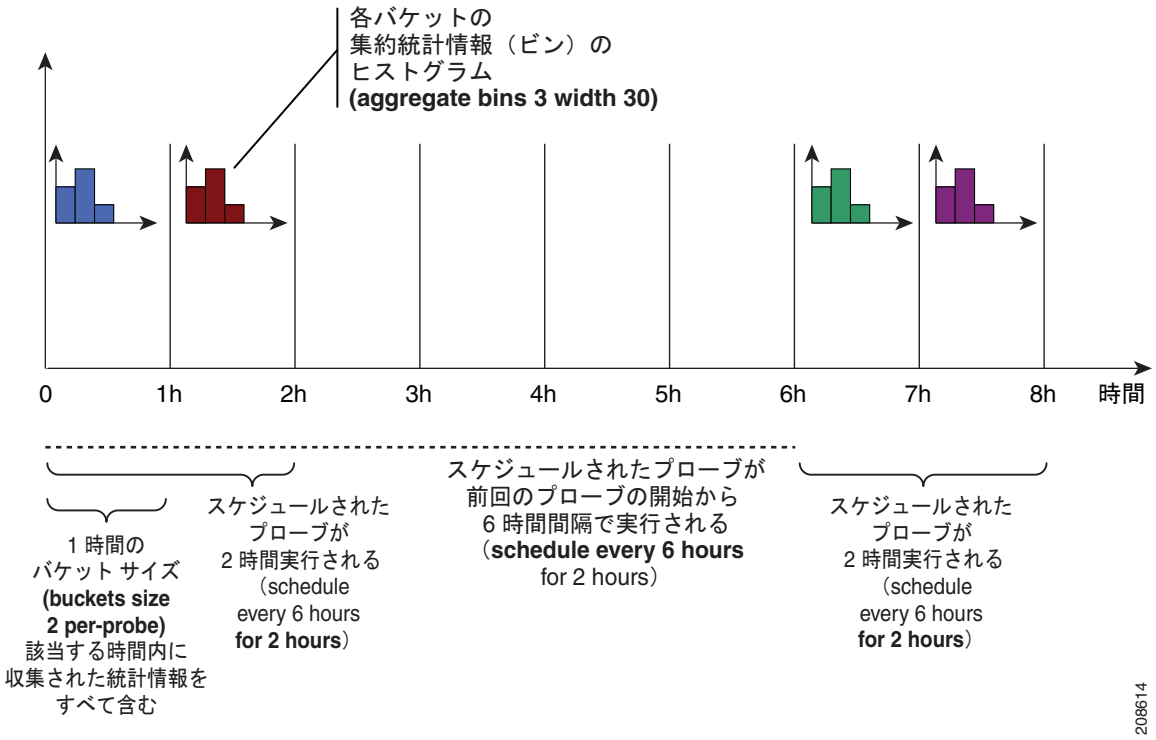
```
configure
 ethernet sla
  profile  Prof1 type cfm-delay-measurement
  schedule every week on Monday at 23:30 for 1 hour
  commit
```

イーサネット SLA 動作プローブのスケジューリングおよび集約の設定：例

図 15 に、プローブのスケジューリングおよび測定の設定の一部の仕組みに集約を使用する包括的な例を示します。次の設定は、図に示された概念の一部をサポートします。

```
configure
 ethernet sla profile Prof1 type cfm-loopback
  probe
   send packet every 60 seconds
   schedule every 6 hours for 2 hours
   statistics measure round-trip-delay
    aggregate bins 3 width 30
    buckets size 2 per-probe
    buckets archive 4
    commit
```

図 15　ビン集約による SLA プローブのスケジュール動作



次の例は、次の特徴を持つプローブをスケジュールしています。

- パケットを 60 秒ごとに送信します（2 時間プローブの場合は、120 個の個別のパケットの送信が行われます）。
- 6 時間ごとにプローブが 2 時間実行されます。
- 各プローブに 2 つのバケットにデータが収集され、各バケットが 2 時間のプローブ期間のうちの 1 時間に対応します。
- バケット内の統計情報をそれぞれ次の範囲で 3 つのビンに集約します。
  - ビン 1 には 0 ms 以上 30 ms 未満の範囲のサンプルを含めます。
  - ビン 2 には 30 ms 以上 60 ms 未満の範囲のサンプルを含めます。
  - ビン 3 には 60 ms 以上の範囲（制限なし）のサンプルを含めます。
- 最後の 4 つのバケットがメモリに保存されます。

進行中のイーサネット SLA 動作の設定：例

次の例では、MEP に対して進行中イーサネット SLA 動作を設定する例を示します。

```
interface gigabitethernet 0/1/0/1
 ethernet cfm
 mep domain Dm1 service Sv1 mep-id 1
 sla operation profile Profile_1 target mac-address 01:23:45:67:89:ab s
 commit
 end
```

オンデマンド イーサネット SLA 動作の基本設定：例

次の例では、オンデマンド イーサネット SLA 動作を設定する方法を示します。

例 1

次の例では、CFM ループバック プローブに対して基本的なオンデマンド イーサネット SLA 動作を設定する方法を示します。このプローブでは、デフォルトでは 1 回限りの 10 秒間の動作のラウンドトリップ遅延とラウンドトリップ ジッターをターゲット MEP に対して測定します。

```
RP/0/RSP0/CPU0:router# ethernet sla on-demand operation type cfm-loopback probe domain D1
source interface TenGigE 0/6/1/0 target mep-id 1
```

例 2

次の例では、CFM 遅延測定プローブに対して基本的なオンデマンド イーサネット SLA 動作を設定する方法を示します。このプローブでは、デフォルトでは一方向の遅延およびジッターを両方向で測定するほか、1 回限りの 10 秒間の動作のラウンドトリップ遅延とラウンドトリップ ジッターをターゲット MEP に対して測定します。

```
RP/0/RSP0/CPU0:router# ethernet sla on-demand operation type cfm-delay-measurement probe
domain D1 source interface TenGigE 0/6/1/0 target mep-id 1
```

イーサネット SLA Y.1731 SLM の設定：例

次の例では、合成損失測定統計情報を設定する方法を示します。

例 1

この例では、Y.1731 SLM のデフォルト設定が表示されます。

```
ethernet sla
  profile sl1 type cfm-synthetic-loss-measurement
    statistic measure one-way-loss-sd
    statistic measure one-way-loss-ds
```

例 2

この例では、「Sl2」という名前のプロファイルを合成損失測定用に設定します。プローブおよび SLM 統計情報を設定するパラメータを指定します。

```
ethernet sla
  profile sl2 type cfm-synthetic-loss-measurement
    probe
      send burst every 5 seconds packet count
                    100 interval 50 milliseconds
      packet size 400 test pattern hex 0xABDC1234
      synthetic loss calculation packets 200
    schedule every 1 hours for 1 minute
    statistic measure one-way-loss-sd
    statistic measure one-way-loss-ds
      aggregate bins 3 width 30
      bucket size 24 probes
```

イーサネット SLA の show コマンド：例

次の例では、設定済みの SLA 動作に関する情報を表示する方法を示します。

show ethernet sla operations コマンド：例 1

```
RP/0/RSP0/CPU0:router# show ethernet sla operations interface gigabitethernet 0/1/0/1.1

Interface GigabitEthernet0/1/0/1.1
Domain mydom Service myser to 00AB.CDEF.1234
------------------------------------------------------------------------------
Profile 'business-gold'
Probe type CFM-delay-measurement:
    bursts sent every 1min, each of 20 packets sent every 100ms
    packets padded to 1500 bytes with zeroes
    packets use priority value of 7
Measures RTT: 5 bins 20ms wide; 2 buckets/ probe; 75/100 archived
Measures Jitter (interval 1): 3 bins 40ms wide; 2 buckets/probe; 50 archived
Scheduled to run every Sunday at 4am for 2 hours:
    last run at 04:00 25/05/2008
```

show ethernet sla configuration-errors コマンド：例 2

```
RP/0/RSP0/CPU0:router# show ethernet sla configuration-errors

Errors:
-------
  Profile 'gold' is not defined but is used on Gi0/0/0/0.0
  Profile 'red' defines a test-pattern, which is not supported by the type
```

次の例では、プローブによって収集された SLA メトリックが格納されているバケットの内容を表示する方法を示します。

show ethernet sla statistics current コマンド：例 3

```
RP/0/RSP0/CPU0:router# show ethernet sla statistics current interface GigabitEthernet
0/0/0/0.0

Interface GigabitEthernet 0/0/0/0.0
Domain mydom Service myser to 00AB.CDEF.1234
==============================================================================
Profile 'business-gold', packet type 'cfm-loopback'
Scheduled to run every Sunday at 4am for 2 hours

Round Trip Delay
~~~~~~~~~~~~~~~~
2 buckets per probe

Bucket started at 04:00 Sun 17 Feb 2008 lasting 1 hour:
    Pkts sent: 2342; Lost 2 (0%); Corrupt: 0 (0%); Misordered: 0 (0%)
    Min: 13ms; Max: 154ms; Mean: 28ms; StdDev: 11ms

Round Trip Jitter
~~~~~~~~~~~~~~~~~
2 buckets per probe

Bucket started at 04:00 Sun 17 Feb 2008 lasting 1 hour:
    Pkts sent: 2342; Lost: 2 (0%); Corrupt: 0 (0%); Misordered: 0 (0%)
    Min: -5ms; Max: 8ms; Mean: 0ms; StdDev: 3.6ms

Bucket started at 05:00 Sun 17 Feb 2008 lasting 1 hour:
    Pkts sent: 2342; Lost: 2 (0%); Corrupt: 0 (0%); Misordered: 0 (0%)
    Min: 0; Max: 4; Mean: 1.4; StdDev: 1
```

show ethernet sla statistics history detail コマンド：例 4

```
RP/0/RSP0/CPU0:router# show ethernet sla history detail GigabitEthernet 0/0/0/0.0

Interface GigabitEthernet 0/0/0/0.0
Domain mydom Service myser to 00AB.CDEF.1234
================================================================================
Profile 'business-gold', packet type 'cfm-loopback'
Scheduled to run every Sunday at 4am for 2 hours

Round Trip Delay
~~~~~~~~~~~~~~~~
2 buckets per probe

Bucket started at 04:00 Sun 17 Feb 2008 lasting 1 hour:
    Pkts sent: 2342; Lost: 2 (0%); Corrupt: 0 (0%); Misordered: 0 (0%)
    Min: 13ms, occurred at 04:43:29 on Sun 22 Aug 2010 UTC
    Max: 154ms, occurred at 05:10:32 on Sun 22 Aug 2010 UTC
    Mean: 28ms; StdDev: 11ms

    Results suspect as more than 10 seconds time drift detected
    Results suspect as scheduling latency prevented some packets being sent

    Samples:
    Time sent       Result  Notes
    ------------  --------  ----------
    04:00:01.324      23ms
    04:00:01.425      36ms
    04:00:01.525         -  Timed Out
    ...

Round Trip Jitter
~~~~~~~~~~~~~~~~~
2 buckets per probe

Bucket started at 04:00 Sun 17 Feb 2008, lasting 1 hour:
    Pkts sent: 2342; Lost: 2 (0%); Corrupt: 0 (0%); Misordered: 0 (0%)
    Min: -5ms; Max: 10ms; Mean: 0ms; StdDev: 3.6ms

    Samples:
    Time sent       Result  Notes
    ------------  --------  ----------
    04:00:01.324         -
    04:00:01.425      13ms
    04:00:01.525         -  Timed out
    ...
```

show ethernet sla statistics history detail on-demand：例 5

次の例では、オンデマンド動作のすべての完全なバケットの統計情報の詳細を表示する方法を示します。

```
RP/0/RSP0/CPU0/router #show ethernet sla statistics history detail on-demand

Interface GigabitEthernet0/0/0/0.1
Domain mydom Service myser to 0123.4567.890A
================================================================================
On-demand operation ID #1, packet type 'cfm-delay-measurement'
Started at 15:38 on 06 July 2010 UTC, runs every 1 hour for 1 hour

Round Trip Delay
~~~~~~~~~~~~~~~~
1 bucket per probe

Bucket started at 15:38 on Tue 06 Jul 2010 UTC, lasting 1 hour:
```

```
    Pkts sent: 1200; Lost: 4 (0%); Corrupt: 600 (50%); Misordered: 0 (0%)
    Min: 13ms, occurred at 15:43:29 on Tue 06 Jul 2010 UTC
    Max: 154ms, occurred at 16:15:34 on Tue 06 Jul 2010 UTC
    Mean: 28ms; StdDev: 11ms

    Bins:
    Range              Samples    Cum. Count      Mean
    ------------  ------------  ------------  --------
     0 - 20 ms       194 (16%)     194 (16%)      17ms
    20 - 40 ms       735 (61%)     929 (77%)      27ms
    40 - 60 ms       212 (18%)    1141 (95%)      45ms
    > 60    ms        55  (5%)    1196            70ms

Bucket started at 16:38 on Tue 01 Jul 2008 UTC, lasting 1 hour:
    Pkts sent: 3600; Lost: 12 (0%); Corrupt: 1800 (50%); Misordered: 0 (0%)
    Min: 19ms, occurred at 17:04:08 on Tue 06 Jul 2010 UTC
    Max: 70ms, occurred at 16:38:00 on Tue 06 Jul 2010 UTC
    Mean: 28ms; StdDev: 11ms

    Bins:
    Range              Samples    Cum. Count      Mean
    ------------  ------------  ------------  --------
     0 - 20 ms       194 (16%)     194 (16%)      19ms
    20 - 40 ms       735 (61%)     929 (77%)      27ms
    40 - 60 ms       212 (18%)    1141 (95%)      45ms
    > 60    ms        55  (5%)    1196            64ms
```

show ethernet sla statistics profile コマンド：例 6

次の例では、合成損失測定の統計情報の詳細を表示する方法を示します。

```
RP/0/RSP0/CPU0:router#show ethernet sla statistics profile sl2 statistic one-way-loss-sd
detail

Source: Interface GigabitEthernet0/0/0/0, Domain dom1
Destination: Target MAC Address 0002.0003.0005
================================================================================
Profile 'sl1', packet type 'cfm-synthetic-loss-measurement'
Scheduled to run every 1hr first at 00:50:00 UTC for 1min
Frame Loss Ratio calculated every 10s

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

Bucket started at 04:50:00 PDT Thu 15 September 2012 lasting 1hr
    Pkts sent: 1200; Lost: 27 (2.25%); Corrupt: 0 (0.0%);
                  Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Min: 0.00%, occurred at 04:50:50 PDT Thu 15 September 2011
    Max: 5.50%, occurred at 04:50:20 PDT Thu 15 September 2011
    Mean: 2.08%; StdDev: 1.99%; Overall: 2.08%

    Measurements:
    Time                       Result  Notes
    ----------  --------------------  ----------
    04:50:00.0    1.50% (3 of 200)
    04:50:10.0    2.00% (4 of 200)
    04:50:20.0    5.50% (11 of 200)
    04:50:30.0    3.00% (6 of 200)
    04:50:40.0    0.50% (1 of 200)
    04:50:50.0    0.00% (0 of 200)
```

例 6 では、統計情報の説明部分に、損失数と全体的な FLR が「Lost: 27 (2.25%)」および「Overall: 2.08%」と表示されています。この損失数の意味は、1200 個の SLM のうち 27 が損失したということですが、どの方向で損失したかは特定できない可能性があります。全体的な FLR として報告されるのは、「発信元から宛先」の方向の全体的な損失です。

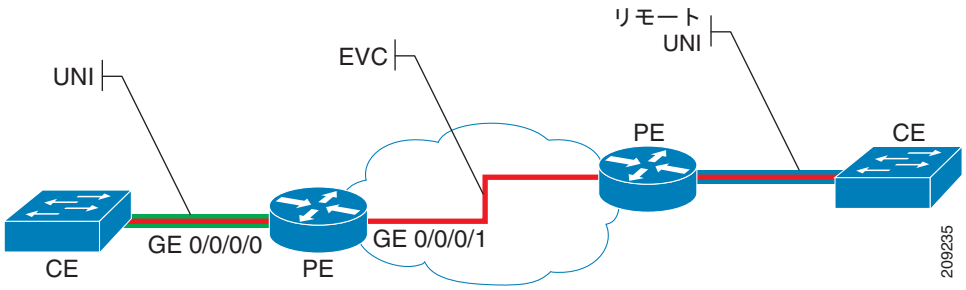
show ethernet sla statistics profile コマンド：例 7

```
RP/0/RSP0/CPU0:ios#show ethernet sla statistics profile sl2 statistic one-way-loss-ds
detail
Source: Interface GigabitEthernet0/0/0/0, Domain dom1
Destination: Target MAC Address 0002.0003.0005
================================================================================
Profile 'sl2', packet type 'cfm-synthetic-loss-measurement'
Scheduled to run every 1hr first at 00:55:00 UTC for 1min
Frame Loss Ratio calculated every 10s

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

Bucket started at 04:55:00 PDT Thu 15 September 2012 lasting 1 day
    Pkts sent: 28800; Lost: 14691 (51.01%); Corrupt: 0 (0.0%);
                      Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Min: 10.00%, occurred at 04:55:00 PDT Thu 15 September 2011
    Max: 68.80%, occurred at 06:55:00 PDT Thu 15 September 2011
    Mean: 52.5%; StdDev: 0.00%; Overall: 51.00%

    Bins:
    Range             Count  Cum. Count    Mean
    -----------  ----------  ----------  ------
      0 to  30%  20 (13.9%)  20 (13.9%)  21.00%
     30 to  60%  71 (49.3%)  91 (63.2%)  57.90%
     60 to 100%  49 (34.0%) 144 (100.0%) 62.00%
```

イーサネット LMI の設定例

図 16 に、ギガビット イーサネット インターフェイス 0/0/0/0 を使用して PE として機能する Cisco ASR 9000 シリーズ ルータで定義されているローカル UNI の基本 E-LMI ネットワーク環境と、ギガビット イーサネット インターフェイス 0/0/0/1 上のリモート UNI への接続を示します。

図 16　基本 E-LMI UNI およびリモート UNI の図



次の設定は、物理ギガビット イーサネット インターフェイス 0/0/0/0 および 0/0/0/1 を使用したローカル UNI で Cisco ASR 9000 シリーズ ルータを PE デバイスとした場合に、図 16 で示された環境の基本 E-LMI 設定を提供します。

```
RP/0/RSP0/CPU0:router# configure
!
!Configure the Local UNI EFPs
!
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/0/0/0.0 l2transport
RP/0/RSP0/CPU0:router(config-subif)# #encapsulation dot1q 1-20
RP/0/RSP0/CPU0:router(config-subif)# exit
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/0/0/1.1 l2transport
RP/0/RSP0/CPU0:router(config-subif)# #encapsulation dot1q 1-20
RP/0/RSP0/CPU0:router(config-subif)# exit
!
!Create the EVC
!
RP/0/RSP0/CPU0:router(config)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group BG1
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain BD1
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface GigabitEthernet0/0/0/0.0
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface GigabitEthernet0/0/0/1.1
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# exit
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# exit
RP/0/RSP0/CPU0:router(config-l2vpn)# exit
!
!Configure Ethernet CFM
!
RP/0/RSP0/CPU0:router(config)# ethernet cfm
RP/0/RSP0/CPU0:router(config-cfm)# domain GLOBAL level 5
RP/0/RSP0/CPU0:router(config-cfm-dmn)# service CustomerA bridge group BG1 bridge-domain
BD1
RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# continuity-check interval 100ms
RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# mep crosscheck mep-id 22
RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# mep crosscheck mep-id 11
RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# exit
RP/0/RSP0/CPU0:router(config-cfm-dmn)# exit
RP/0/RSP0/CPU0:router(config-cfm)# exit
!
!Configure EFPs as CFM MEPs
!
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/0/0/0.0 l2transport
RP/0/RSP0/CPU0:router(config-subif)# ethernet cfm
RP/0/RSP0/CPU0:router(config-if-cfm)# mep domain GLOBAL service CustomerA mep-id 22
RP/0/RSP0/CPU0:router(config-if-cfm)# exit
RP/0/RSP0/CPU0:router(config-subif)# exit
!
!Configure the Local UNI Name
!
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/0
RP/0/RSP0/CPU0:router(config-if)# ethernet uni id PE1-CustA-Slot0-Port0
RP/0/RSP0/CPU0:router(config-if)# exit
!
!Enable E-LMI on the Local UNI Physical Interface
!
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/0
RP/0/RSP0/CPU0:router(config-if)# ethernet lmi
RP/0/RSP0/CPU0:router(config-if)# exit
RP/0/RSP0/CPU0:router(config)# commit
```

次の作業

イーサネット インターフェイスの設定が完了したら、イーサネット インターフェイスで各 VLAN サブインターフェイスを設定できます。

シェルフ コントローラ（SC）、ルート プロセッサ（RP）、および分散型 RP のイーサネット管理インターフェイスの変更方法については、このマニュアルで後述する「Cisco ASR 9000 シリーズ ルータでの管理イーサネット インターフェイスの高度な設定および変更」モジュールを参照してください。

IPv6 については、『*Cisco* IOS XR IP Addresses and Services Configuration Guide』の「*Implementing Access Lists and Prefix Lists on Cisco IOS XR Software*」モジュールを参照してください。

その他の関連資料

ここでは、ギガビットおよび 10 ギガビット イーサネット インターフェイスの実装に関する参考資料を紹介します。

関連資料

関連項目	参照先
イーサネット L2VPN	*『Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide』* *『Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Command Reference』*
Cisco IOS XR マスター コマンド リファレンス	*『Cisco IOS XR Master Commands List』*
Cisco IOS XR インターフェイス コンフィギュレーション コマンド	*『Cisco IOS XR Interface and Hardware Component Command Reference』*
ユーザ グループとタスク ID に関する情報	*『Cisco IOS XR Interface and Hardware Component Command Reference』*

標準

標準	タイトル
IEEE 802.1ag	*『Connectivity Fault Management』*
ITU-T Y.1731（2011 年 7 月）	*OAM Functions and Mechansims for Ethernet Based Networks*
MEF 16	*Metro Ethernet Forum, Technical Specification MEF 16, Ethernet Local Management Interface (E-LMI), January 2006*

MIB

MIB	MIB のリンク
IEEE8021-CFM-MIB	Cisco IOS XR ソフトウェアを使用して選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	—

シスコのテクニカル サポート

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

Cisco ASR 9000 シリーズ ルータ での Integrated Routing and Bridging の設定

このモジュールでは、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータでの Integrated Routing and Bridging（IRB）の設定について説明します。IRB では、Cisco ASR 9000 シリーズ ルータ上のブリッジ サービスとブリッジ グループ仮想インターフェイス（BVI）を使用するルーテッド インターフェイスとの間でトラフィックを交換できます。

IRB 機能の履歴

リリース	変更内容
リリース 4.0.1	この機能は、Cisco ASR 9000 シリーズ ルータで次のラインカードに対して導入されました。 • 2 ポート 10 ギガビット イーサネット、20 ポート ギガビット イーサネット コンビネーション ラインカード（A9K-2T20GE-B および A9K-2T20GE-L） • 4 ポート 10 ギガビット イーサネット ラインカード（A9K-4T-B、-E、-L） • 8 ポート 10 ギガビット イーサネット DX ラインカード（A9K-8T/4-B、-E、-L） • 8 ポート 10 ギガビット イーサネット ラインカード（A9K-8T-B、-E、-L） • 16 ポート 10 ギガビット イーサネット ラインカード（9K-16T/8-B、-E、-L） • 40 ポート ギガビット イーサネット ラインカード（A9K-40GE-B、-E、-L）
リリース 4.1.0	• コア向きのインターフェイスとしてサポートされる SPA とともに Cisco ASR 9000 SIP-700 を使用する、次の IRB 環境のサポートが追加されました。 – Cisco ASR 9000 SIP-700 から IRB をサポートするギガビット イーサネット ラインカード上のレイヤ 2 ブリッジド インターフェイスへのレイヤ 3 ルーテッド トラフィック。 – IPv4 ユニキャスト トラフィックのみ。 • IRB の IPv6 ユニキャスト アドレッシングのサポートおよび BVI インターフェイスを使用する 6PE/6VPE のサポートが、次のラインカードに対して追加されました。 – 2 ポート 10 ギガビット イーサネット、20 ポート ギガビット イーサネット コンビネーション ラインカード（A9K-2T20GE-B および A9K-2T20GE-L） – 4 ポート 10 ギガビット イーサネット ラインカード（A9K-4T-B、-E、-L） – 8 ポート 10 ギガビット イーサネット DX ラインカード（A9K-8T/4-B、-E、-L） – 8 ポート 10 ギガビット イーサネット ラインカード（A9K-8T-B、-E、-L） – 16 ポート 10 ギガビット イーサネット ラインカード（9K-16T/8-B、-E、-L） – 40 ポート ギガビット イーサネット ラインカード（A9K-40GE-B、-E、-L）

内容

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。

- ルータのコア向き側に設置された Cisco ASR 9000 SIP-700 がある場合、IPv4 ユニキャスト トラフィックのレイヤ 2 ブリッジド トラフィック フローにルーティングされているレイヤ 3 の IRB をサポートできます。ここで、レイヤ 2 の宛先は、IRB のサポートされるギガビット イーサネット ラインカードの 1 つです。
- レイヤ 3 からレイヤ 2 へのトラフィック フローとレイヤ 2 からレイヤ 3 へのトラフィック フローの両方をサポートする IRB のサポートを計画するギガビット イーサネット ラインカードとして、次のタイプだけが設定されていることを確認します。
  - 2 ポート 10 ギガビット イーサネット、20 ポート ギガビット イーサネット コンビネーション ラインカード（A9K-2T20GE-B および A9K-2T20GE-L）
  - 4 ポート 10 ギガビット イーサネット ラインカード（A9K-4T-B、-E、-L）
  - 8 ポート 10 ギガビット イーサネット DX ラインカード（A9K-8T/4-B、-E、-L）
  - 8 ポート 10 ギガビット イーサネット ラインカード（A9K-8T-B、-E、-L）
  - 16 ポート 10 ギガビット イーサネット ラインカード（9K-16T/8-B、-E、-L）
  - 40 ポート ギガビット イーサネット ラインカード（A9K-40GE-B、-E、-L）
- 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。詳細については、「IRB の設定に関する制約事項」（P.192）を参照してください。
- 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 の設定に関する制約事項

IRB を設定する前に、次の制約事項を考慮してください。

- 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。

注意 Cisco ASR 9000 SIP-700 も設置されている Cisco ASR 9000 シリーズ ルータで IRB をサポートする場合は、SIP-700 と BVI インターフェイス間のトラフィックの損失を防ぐため、ルーティング設定を必ず実行する必要があります。詳細については、後述の制約事項を参照してください。

- Cisco IOS XR Release 4.1 以降、IRB は、Cisco ASR 9000 SIP-700 も設置されているシステムのサポートされるギガビット イーサネット ラインカードに次の制約事項に従って実装できます。
  - Cisco ASR 9000 SIP-700 は、IPv4 アドレッシングで設定された BVI インターフェイスがあるルータのコア向き側に設置されている必要があります。
  - Cisco ASR 9000 SIP-700 は、次のギガビット イーサネット ラインカードのいずれかがレイヤ 2 ブリッジ ドメインにある、IRB を使用するレイヤ 3 からブリッジド レイヤ 2 インターフェイスへの IPv4 ユニキャスト トラフィックのルーティングをサポートできます。

    - 2 ポート 10 ギガビット イーサネット､20 ポート ギガビット イーサネット コンビネーション ラインカード（A9K-2T20GE-B および A9K-2T20GE-L）

    - 4 ポート 10 ギガビット イーサネット ラインカード（A9K-4T-B、-E、-L）

    - 8 ポート 10 ギガビット イーサネット DX ラインカード（A9K-8T/4-B、-E、-L）

    - 8 ポート 10 ギガビット イーサネット ラインカード（A9K-8T-B、-E、-L）

    - 16 ポート 10 ギガビット イーサネット ラインカード（9K-16T/8-B、-E、-L）

    - 40 ポート ギガビット イーサネット ラインカード（A9K-40GE-B、-E、-L）

（注） これらのラインカードからレイヤ 3 の Cisco ASR 9000 SIP-700 へのレイヤ 2 ブリッジド トラフィックの逆方向もサポートされます。

- 次の領域は、BVI でサポートされて*いません*。
  - アクセス コントロール リスト（ACL）。ただし、レイヤ 2 ACL はブリッジ ドメインの各レイヤ 2 ポートで設定できます。
  - IP 高速再ルーティング（FRR）
  - NetFlow
  - MoFRR
  - MPLS ラベル スイッチング
  - mVPNv4
  - Quality of Service（QoS）
  - トラフィック ミラーリング
  - BVI のアンナンバード インターフェイス
  - ビデオ モニタリング（Vidmon）
- 802.1ah を使用する IRB（BVI とプロバイダー バックボーン ブリッジ（PBB）は、同じブリッジ ドメイン内に設定しないでください)。

- PIM スヌーピング （選択的フラッディングを使用する必要があります）。
- VRF-aware DHCP リレーはサポートされません。
- BVI は次の特性を持つブリッジ ドメインでのみサポートされます。
  - ブリッジ ドメインは、曖昧ではない「完全一致」EFP カプセル化による、一重および二重タグ付き dot1q および dot1ad カプセル化を使用する EFP をサポートします。一重および二重タグ付きカプセル化は、**rewrite ingress tag pop symmetric** コマンドが設定されている限り指定できます。
  - すべてのレイヤ 2 タグを削除する必要があります。VLAN 範囲はサポートされません。
  - タグなし EFP はサポートされます。
- 次の追加機能は、コア向き側に Cisco ASR 9000 SIP-700 がある環境の BVI インターフェイスでサポートされて*いません*。
  - 『ARP』
  - フレーム リレー
  - IPv4 マルチキャスト トラフィック
  - IPv6 ユニキャストおよびマルチキャスト トラフィック
  - SIP-700 から任意のレイヤ 3 インターフェイスへのレイヤ 2 トラフィック フロー
  - BVI インターフェイスのレイヤ 2/ レイヤ 3 機能
  - 負荷統計間隔
  - MIB
  - **show adjacency details** コマンドはサポートされません。

IRB の設定に関する情報

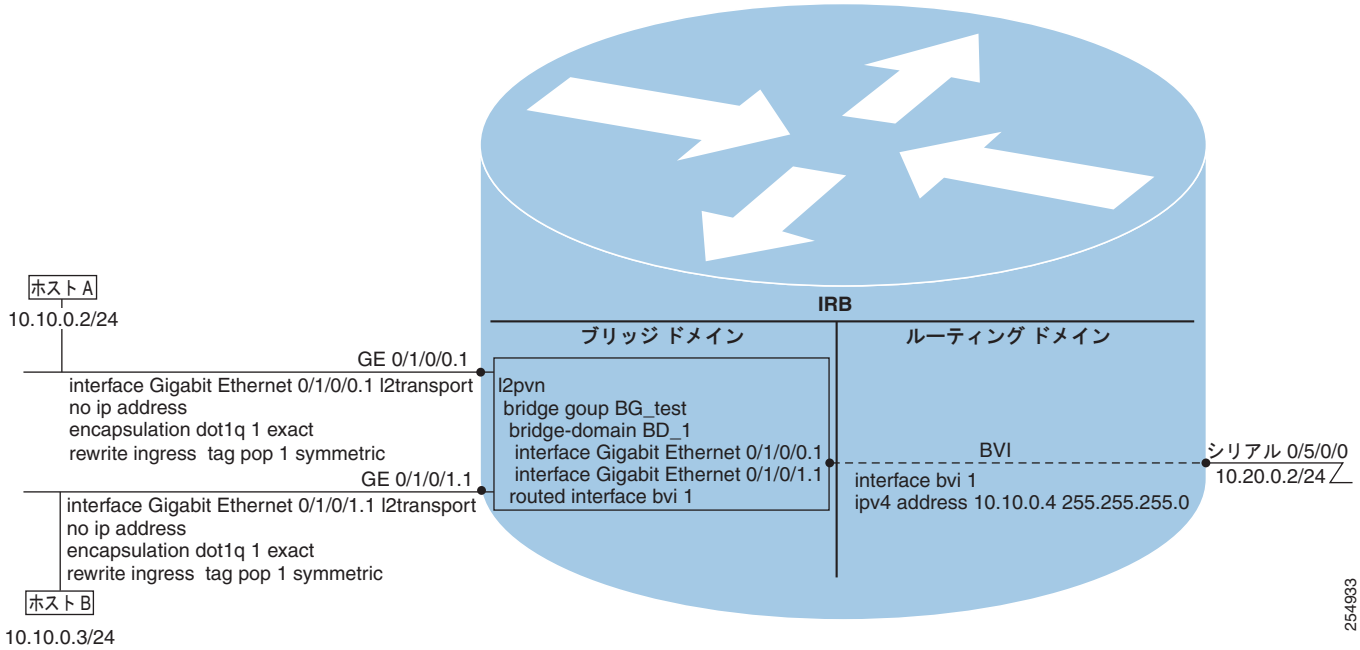
この項では、次のトピックについて取り上げます。

IRB の概要

IRB では、BVI を使用してブリッジ グループとルーテッド インターフェイス間でルーティングできます。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 は単一ブリッジ ドメインに関連付けられ、ルータのブリッジおよびルーティング ドメイン間のリンクを表します。ルーテッド インターフェイス宛てのブリッジド インターフェイスからのパケットの受信をサポートするには、BVI に適切な IP アドレスと関連するレイヤ 3 属性を設定する必要があります。

IRB がサポートされていない Cisco IOS XR 4.0.1 よりも前のソフトウェア リリースでは、同じ Cisco ASR 9000 シリーズ ルータのレイヤ 3 ルーティング ドメイン インターフェイスに出力レイヤ 2 ブリッジ ドメイン インターフェイスを接続するための物理ケーブル接続ソリューションを実装する必要があります。Cisco IOS XR Release 4.0.1 では、IRB が BVI およびそれがサポートするインターフェイスとブリッジ グループによる図 1 に示す設定を使用して、同じ機能を実現します。

図 1　IRB の機能の概観と設定要素



ブリッジ グループ仮想インターフェイス

ここでは次の内容について説明します。

BVI の概要

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 タイプが設定され、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 は、セカンダリ アドレスもサポートします。
- 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 でサポートされる機能

- 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**（Cisco ASR 9000 SIP-700 のある IRB 環境ではサポートされません）
  - **mac-address**
  - **mtu**（デフォルトは 1500 バイトです）
  - **shutdown**
- BVI は、IP ヘルパー アドレッシングおよびセカンダリ IP アドレッシングをサポートします。

BVI MAC アドレス

デフォルトで、Cisco ASR 9000 シリーズ ルータはルータのすべての BVI インターフェイスに対して 1 つの MAC アドレスを使用します。一方で、これは MAC アドレスがグローバルに一意でないことを意味します。デフォルトを上書きして、一意な MAC アドレスを BVI に指定する場合は、BVI インターフェイスで設定できます。

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））。

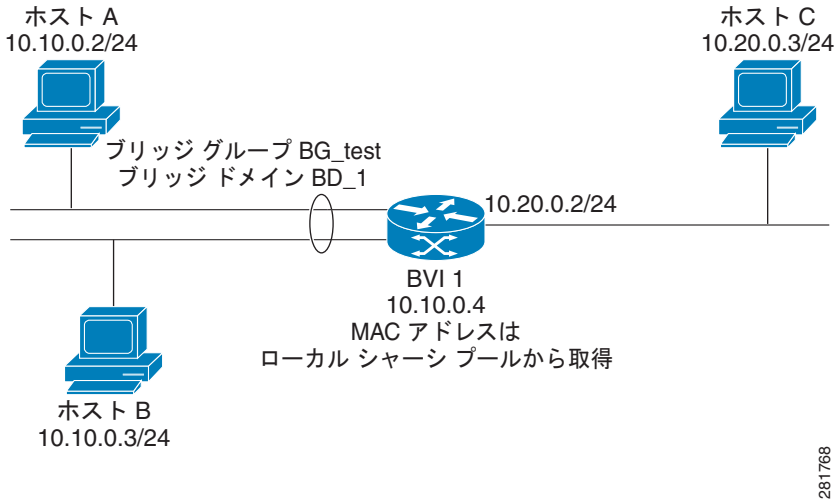
（注） BVI は、その BVI のブリッジ ドメインに関連付けられたすべてのブリッジ ポート（イーサネット フロー ポイント（EFP））がダウンしている場合、Down 状態に移行します。ただし、すべての EFP がダウンしていても、少なくとも 1 つの疑似回線がアップの場合、BVI はアップのままです。

- 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 Up 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 を使用するパケット フロー

図 2 に、ホスト A、B、および C の間のさまざまなパケット フローを説明する、IRB 実装の単純化した機能図を示します。この例では、ホスト C は同じルータへの接続を持つネットワークです。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 2　　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、図 2 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、図 2 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 でサポートされる環境

次の環境および設定要素が Cisco ASR 9000 シリーズ ルータの IRB でサポートされます。

- ブリッジ ドメインごとに 1 つの BVI の設定。
- BVI で設定されたブリッジ ドメインに関連付けられた仮想プライベート LAN サービス（VPLS）仮想転送インスタンス（VFI）設定。
- BVI ベースのプレフィックスの BGP PIC エッジ。
- Open Shortest Path First（OSPF）、Intermediate System-to-Intermediate System（IS-IS）、Routing Information Protocol Version 2（RIPv2）、およびボーダー ゲートウェイ プロトコル（BGP）を使用した BVI のトラフィック転送。
- インターネット グループ管理プロトコル（IGMP）スタティック グループ。

- ダイナミック ホスト コンフィギュレーション プロトコル（DHCP）リレー エージェント。IP アドレスを取得するために DHCP リレーが集約ノードから使用される場合、デフォルト ゲートウェイは BVI で設定された IP アドレスになります。BVI IP アドレスは、IP アドレスを割り当てるために集約ノードで使用している DHCP プールと共通のサブネットにある必要があります。
- 仮想ルータ冗長プロトコル（VRRP）設定およびプライオリティ。
- ホットスタンバイ ルータ プロトコル（HSRP）。
- BVI インターフェイスあたり最大 255 の VRRF/HSRP VMAC。
- BVI で設定されているブリッジ ドメインの非 IP パケットのブリッジング。
- Cisco ASR 9000 シリーズ ルータのレイヤ 3 サブインターフェイスで現在サポートされているものと同様のステートフル プロトコルを使用するパリティのサポート。
- Cisco ASR 9000 シリーズ ルータのレイヤ 3 サブインターフェイスで現在サポートされているものと同様の IP SLA サポート。
- ECMP パス（最大 32 のパス）としての BVI のロード バランシング。
- インターフェイス MIB。
- BVI インターフェイスのパケット カウンタ。
- BVI を使用するブリッジ ドメインのメンバであるリンク バンドルのマルチシャーシ リンク集約（LAG）。

次の各項では、IRB でサポートされる追加の IPv4 および IPv6 固有の環境について説明します。

- 「IRB でサポートされる追加の IPv4 固有の環境」（P.198）
- 「IRB でサポートされる追加の IPv6 固有の環境」（P.198）

IRB でサポートされる追加の IPv4 固有の環境

- 最大 2000 の BVI の設定。
- 最大 128k の IPv4 隣接。
- 入力 IP マルチキャスト トラフィックを取り込み、マルチキャスト グループの一部であるブリッジ ドメインの複数のレイヤ 2 サブインターフェイス（イーサネット フロー ポイント）にブリッジングする機能があるレイヤ 3 IP マルチキャスト。

（注） コア向き側の Cisco ASR 9000 SIP-700 で使用する場合はサポートされません。

- IPv4 用 VRF（プレフィックス単位ではなく、VPN ラベル単位の VRF のみ）。

IRB でサポートされる追加の IPv6 固有の環境

- 最大 2000 の BVI の設定。その内、最大 512 の BVI が IPv6 アドレッシングをサポートできます。
- 最大 5k の IPv6 隣接。
- PE デバイスとして Cisco ASR 9000 シリーズ ルータのカスタマー エッジ（CE）向き側で BVI インターフェイスを使用する、Cisco IPv6 プロバイダー エッジ ルータ over MPLS（6PE）および IPv6 VPN プロバイダー エッジ（6VPE）の次の制約事項をともなうサポート。
  - PE デバイス上の次のラインカードでサポートされます。

    - 2 ポート 10 ギガビット イーサネット、20 ポート ギガビット イーサネット コンビネーション ラインカード（A9K-2T20GE-B および A9K-2T20GE-L）

- 4 ポート 10 ギガビット イーサネット ラインカード（A9K-4T-B、-E、-L）
- 8 ポート 10 ギガビット イーサネット DX ラインカード（A9K-8T/4-B、-E、-L）
- 8 ポート 10 ギガビット イーサネット ラインカード（A9K-8T-B、-E、-L）
- 16 ポート 10 ギガビット イーサネット ラインカード（9K-16T/8-B、-E、-L）
- 40 ポート ギガビット イーサネット ラインカード（A9K-40GE-B、-E、-L）

– IPv6 アドレッシングを使用する最大 512 の BVI がサポートできます。

– VRF 単位のラベル割り当てだけがサポートされます（**label-allocation-mode per-vrf** コマンドを使用）。

設定例は、「BVI を使用する 6PE/6VPE の設定：例」（P.210）を参照してください。

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」（P.199）（必須）
- 「レイヤ 2 AC インターフェイスの設定」（P.201）（必須）
- 「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」（P.203）（必須）
- 「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の BVI の関連付け」（P.205）（必須）
- 「BVI に関する情報の表示」（P.207）（任意）

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。
- ブリッジ型ネットワークに複数の IP ネットワークがある場合、BVI には各ネットワークのセカンダリ IP アドレスを割り当てる必要があります。

手順の概要

1. **configure**
2. **interface bvi** *identifier*
3. **ipv4 address** *ipv4-address mask* [**secondary**]

   または

   ipv6 address *ipv6-prefix/prefix-length* [**eui-64**] [**route-tag** *route-tag value*]
4. **arp purge-delay** *seconds*
5. **arp timeout** *seconds*

6. **bandwidth** *rate*
7. **mac-address** *value1.value2.value3*
8. **mtu** *bytes*
9. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface bvi** *identifier* **例：** `RP/0/RSP0/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/RSP0/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/RSP0/CPU0:router(config-if)#arp purge-delay 120`	（任意）インターフェイスがダウンになるときの、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（*seconds* に）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/RSP0/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。 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/RSP0/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。 範囲は 0 ～ 4294967295 です。デフォルトは 10000000（10 Gbps）です。
ステップ 7	**mac-address** *value1.value2.value3* **例：** `RP/0/RSP0/CPU0:router(config-if)# mac-address 1111.2222.3333`	（任意）BVI の 48 ビット MAC アドレスを 3 つのドット付き 16 進値で指定し、デフォルト MAC アドレスの使用を上書きします。各値の範囲は 0000 ～ ffff です。すべてが 0 の MAC アドレスはサポートされません。

	コマンドまたはアクション	目的
ステップ 8	**mtu** *bytes* **例：** `RP/0/RSP0/CPU0:router(config-if)# mtu 2000`	（任意）インターフェイスのパケットの最大伝送単位（MTU）サイズを指定します。範囲は 64 ～ 65535 です。デフォルトは 1514 です。
ステップ 9	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

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。

前提条件

ブリッジ ドメインのレイヤ 2 AC として設定され、BVI によりルーティングされるインターフェイスは、Cisco ASR 9000 シリーズ ルータの IRB をサポートする次のタイプのカードにある必要があります。

- 2 ポート 10 ギガビット イーサネット、20 ポート ギガビット イーサネット コンビネーション ラインカード（A9K-2T20GE-B および A9K-2T20GE-L）
- 4 ポート 10 ギガビット イーサネット ラインカード（A9K-4T-B、-E、-L）
- 8 ポート 10 ギガビット イーサネット DX ラインカード（A9K-8T/4-B、-E、-L）
- 8 ポート 10 ギガビット イーサネット ラインカード（A9K-8T-B、-E、-L）
- 40 ポート ギガビット イーサネット ラインカード（A9K-40GE-B、-E、-L）

手順の概要

1. **configure**
2. **interface** {**GigabitEthernet** | **TenGigE**} *interface-path-id*[.*subinterface*] **l2transport**
3. **no ip address**
4. **encapsulation dot1q** *vlan-id* **exact**

   または

   encapsulation dot1ad *vlan-id* **dot1q** *vlan-id*
5. **rewrite ingress tag pop** {**1** | **2**} **symmetric**
6. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** [**GigabitEthernet**	**TenGigE**] *interface-path-id*[.*subinterface*] **l2transport** **例：** `RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/1/0/0.1 l2transport`	ギガビット イーサネットもしくは 10 ギガビット イーサネット インターフェイスまたはサブインターフェイスのレイヤ 2 転送モードをイネーブルにし、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。ここで、*interface-path-id* は *rack*/*slot*/*module*/*port* としてインターフェイスの場所を指定し、*.subinterface* はオプションのサブインターフェイス番号です。
ステップ 3	**encapsulation dot1q** *vlan-id* [**exact**] または **encapsulation dot1ad** *vlan-id* **dot1q** *vlan-id* **例：** `RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1q 1 exact`	（任意）指定された VLAN だけで IEEE 802.1q カプセル化を指定します。

	コマンドまたはアクション	目的
ステップ 4	**rewrite ingress tag pop** {**1** | **2**} **symmetric** **例：** `RP/0/RSP0/CPU0:router(config-if)# rewrite ingress tag pop 1 symmetric`	（VLAN タギングが設定されている場合は必須）入力インターフェイスに到達するブリッジ ドメインへのフレームから 1 つまたは 2 つのタグ（ネットワーク設定による）を削除する必要があることを指定します。 **（注）** dot1ad および dot1q カプセル化を使用する二重タグを設定している場合、**rewrite ingress tag pop 2 symmetric** コマンドを使用する必要があります。
ステップ 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

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。

手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *name*
4. **bridge-domain** *name*
5. **interface** {**GigabitEthernet** | **TenGigE**} *interface-path-id*[**.***subinterface*]
6. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**l2vpn** **例：** `RP/0/RSP0/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/RSP0/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/RSP0/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。

	コマンドまたはアクション	目的
ステップ 5	**interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*[*.subinterface*] **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface GigabitEthernet 0/1/0/0.1`	ギガビット イーサネットおよび 10 ギガビット イーサネット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。ここで、*interface-path-id* は *rack*/*slot*/*module*/*port* としてインターフェイスの場所を指定し、*.subinterface* はオプションのサブインターフェイス番号です。 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/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# end` または `RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

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。

手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group-name*
4. **bridge-domain** *bridge-domain-name*
5. **routed interface bvi** *identifier*
6. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**l2vpn** **例：** `RP/0/RSP0/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/RSP0/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/RSP0/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/RSP0/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/RSP0/CPU0:router(config-l2vpn-bg-bd)# end` または `RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

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。

コマンド	目的
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 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」（P.207）
- 「VLAN のある AC を使用する IRB：例」（P.208）
- 「複数の IP ネットワークをサポートする BVI の IPv4 アドレッシング：例」（P.208）
- 「BVI バンドル インターフェイスおよびマルチキャスト設定を含む包括的 IRB 設定：例」（P.208）
- 「BVI および VRRP を使用する IRB の設定：例」（P.210）

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/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface bvi 1
RP/0/RSP0/CPU0:router(config-if))# ipv4 address 10.10.0.4 255.255.255.0
RP/0/RSP0/CPU0:router(config-if))# exit
!
! Configure the Layer 2 AC interface
!
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/1/0/0 l2transport
RP/0/RSP0/CPU0:router(config-if))# exit
!
! Configure the L2VPN bridge group and bridge domain and assign interfaces
!
RP/0/RSP0/CPU0:router(config)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group 10
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain 1
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface GigabitEthernet 0/1/0/0
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-if)# exit
!
! Associate a BVI to the bridge domain
!
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# routed interface bvi 1
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# commit
```

VLAN のある AC を使用する IRB：例

次に、802.1q カプセル化 VLAN を使用するレイヤ 2 AC でブリッジ ドメイン上の IRB を設定する例を示します。

```
! Configure the BVI and its IPv4 address
!
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface bvi 1
RP/0/RSP0/CPU0:router(config-if))# ipv4 address 10.10.0.4 255.255.255.0
RP/0/RSP0/CPU0:router(config-if))# exit
!
! Configure the Layer 2 AC interfaces using dot1q encapsulation on a VLAN
!
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/1/0/0.1 l2transport
RP/0/RSP0/CPU0:router(config-if))# no ip address
RP/0/RSP0/CPU0:router(config-if))# encapsulation dot1q 1 exact
RP/0/RSP0/CPU0:router(config-if))# rewrite ingress tag pop 1 symmetric
RP/0/RSP0/CPU0:router(config-if))# exit
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/1/0/1.1 l2transport
RP/0/RSP0/CPU0:router(config-if))# no ip address
RP/0/RSP0/CPU0:router(config-if))# encapsulation dot1q 1 exact
RP/0/RSP0/CPU0:router(config-if))# rewrite ingress tag pop 1 symmetric
RP/0/RSP0/CPU0:router(config-if))# exit
!
! Configure the L2VPN bridge group and bridge domain and assign interfaces
!
RP/0/RSP0/CPU0:router(config)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group 10
RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain 1
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface GigabitEthernet 0/1/0/0.1
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface GigabitEthernet 0/1/0/1.1
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-if)# exit
!
! Associate a BVI to the bridge domain
!
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# routed interface bvi 1
RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# commit
```

複数の IP ネットワークをサポートする BVI の IPv4 アドレッシング：例

次に、10.10.10.0/24、10.20.20.0/24、および 10.30.30.0/24 ネットワークのブリッジ ドメインをサポートする BVI のセカンダリ IPv4 アドレスを設定する例を示します。この例では、BVI がそれぞれのブリッジ ドメイン ネットワーク上のアドレスを持っている必要があります。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface bvi 1
RP/0/RSP0/CPU0:router(config-if))# ipv4 address 10.10.10.4 255.255.255.0
RP/0/RSP0/CPU0:router(config-if))# ipv4 address 10.20.20.4 255.255.255.0 secondary
RP/0/RSP0/CPU0:router(config-if))# ipv4 address 10.30.30.4 255.255.255.0 secondary
RP/0/RSP0/CPU0:router(config-if))# commit
```

BVI バンドル インターフェイスおよびマルチキャスト設定を含む包括的 IRB 設定：例

次に、IRB および BVI のマルチキャストをサポートするより包括的なルータ設定の例を示します。

```
interface Bundle-Ether25
```

```
 ipv4 address 10.21.0.2 255.255.255.0
!
interface Loopback0
 ipv4 address 10.5.5.5 255.255.255.255
!
interface GigabitEthernet0/0/0/1
 negotiation auto
!
interface GigabitEthernet0/0/0/1.1 l2transport
 encapsulation dot1q 1
 rewrite ingress tag pop 1 symmetric
!
interface GigabitEthernet0/0/0/1.2 l2transport
 encapsulation dot1q 2
 rewrite ingress tag pop 1 symmetric
!

interface GigabitEthernet0/0/0/9
 bundle id 25 mode active
!
interface GigabitEthernet0/0/0/19
 bundle id 25 mode active
!
interface GigabitEthernet0/0/0/29
 bundle id 25 mode active
!

interface GigabitEthernet0/0/0/39
 bundle id 25 mode active

interface BVI1
 ipv4 address 10.1.1.1 255.255.255.0
!
interface BVI2
 ipv4 address 10.1.2.1 255.255.255.0

router ospf 100
 router-id 10.5.5.5
 area 0
  interface Bundle-Ether25
    interface Loopback0
    interface BVI1
   interface BVI2
  !
l2vpn
 bridge group IRB
  bridge-domain IRB1
   igmp snooping profile IRB_SNOOP
   interface GigabitEthernet0/0/0/1.1
   !
   routed interface BVI1
  !
  bridge-domain IRB2
   igmp snooping profile IRB_SNOOP
   interface GigabitEthernet0/0/0/1.2
   !
   routed interface BVI2

multicast-routing
 address-family ipv4
  interface all enable
igmp snooping profile IRB_SNOOP
 report-suppression disable
!
```

```
router pim
 address-family ipv4
  rp-address 10.10.10.10
```

BVI および VRRP を使用する 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 GigabitEthernet0/0/0/8
!
   routed interface BVI 100
!
interface GigabitEthernet0/0/0/8
  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
 !
```

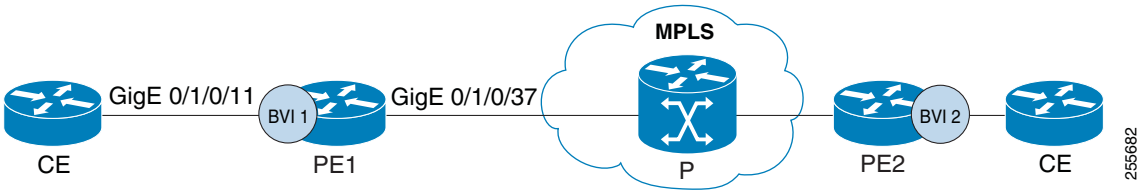
BVI を使用する 6PE/6VPE の設定：例

次に、PE デバイスとして Cisco ASR 9000 シリーズ ルータの CE 向き側に BVI を使用して、MPLS 6PE/6VPE 環境を設定する例を示します。Cisco 6PE/6VPE およびその設定の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router MPLS Layer 3 VPN Configuration Guide*』の「Implementing IPv6 VPN Provider Edge Transport Over MPLS」の章を参照してください。

（注） この環境は、Cisco ASR 9000 シリーズ ルータでサポートされるギガビット イーサネット ラインカードで IRB を使用する場合にだけサポートされます。Cisco ASR 9000 SIP-700 SPA ではサポートされません。

図 3 に、PE1 および PE2 デバイスとして使用する Cisco ASR 9000 シリーズ ルータの BVI インターフェイス（緑色のアイコン）の位置を示します。

図 3　　MPLS 6PE/6VPE ネットワークの CE 向き側の BVI インターフェイス



次の例は、Cisco ASR 9000 シリーズ ルータ（PE1）デバイスのみのサンプル設定を示しています。番号 1 の BVI インターフェイスが CE 側にあり、非 BVI インターフェイス（ギガビット イーサネット 0/1/0/37）がコア側にあります。同様の設定が、PE2 デバイスに適用されます。

```
! Be sure to configure IPv6 unicast address families
!
vrf 1
address-family ipv6 unicast
  import route-target
   100:2
  export route-target
   100:2

interface Loopback0
 ipv4 address 10.11.11.11/32
!
! Configure the BVI interface to participate in the VRF
! and with an IPv6 address.
!
interface BVI1
 vrf 1
 ipv6 address 2001:DB8:1/32
!
! Assign the Gigabit Ethernet CE-facing interface to the
! L2VPn bridge domain where the routed BVI interface is also associated.
!
l2vpn
 bridge group 1
  bridge-domain 1
   interface Gigabit Ethernet 0/1/0/11
 routed interface BVI1
!
! Configure OSPF routing for the BVI interface for
! advertisement of its IPv6 address.
!
router ospfv3 1
 graceful-restart
 redistribute bgp 1
 area 1
  interface BVI1
  interface Loopback0
!
! Configure BGP routing and be sure to specify the
! IPv6 unicast address family.
! Note that the per-VRF label allocation mode is required
! and is the only supported label allocation mode.
!
router bgp 1
```

```
 bgp router-id 10.11.11.11
 bgp redistribute-internal
 bgp graceful-restart

 address-family ipv6 unicast
  redistribute ospfv3 1 match internal external
  label-allocation-mode per-vrf
  allocate-label all
 !
 address-family vpnv6 unicast
 !
 neighbor 10.11.12.12
  remote-as 1
  update-source Loopback0
  address-family ipv6 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
  address-family ipv6 labeled-unicast
  !
  address-family vpnv6 unicast
   route-policy pass-all in
   route-policy pass-all out
  !
 vrf 1
  rd 100:2
 label-allocation-mode per-vrf
  address-family ipv6 unicast
   redistribute connected

mpls ldp
 router-id 10.11.11.11
 graceful-restart
 interface Gigabit Ethernet 0/1/0/37
```

その他の関連資料

次の各項では、Cisco ASR 9000 シリーズ ルータでの IRB の設定に関する参考資料について説明します。

関連資料

関連項目	参照先
イーサネット L2VPN	*『Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide』* *『Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Command Reference』*
Cisco IOS XR マスター コマンド リファレンス	*『Cisco ASR 9000 Series Aggregation Services Router Master Command Listing, Release 4.0』*
Cisco IOS XR インターフェイス コンフィギュレーション コマンド	*『Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Command Reference』*

関連項目	参照先
Cisco IOS XR マルチキャスト設定	*『Cisco ASR 9000 Series Aggregation Services Router Multicast Configuration Guide』*
MPLS レイヤ 3 VPN の設定	*『Cisco ASR 9000 Series Aggregation Services Router MPLS Layer 3 VPN Configuration Guide』*

標準

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

MIB

MIB	MIB のリンク
IF-MIB	Cisco IOS XR ソフトウェアを使用して選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	—

シスコのテクニカル サポート

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

その他の関連資料

Cisco ASR 9000 シリーズ ルータ でのリンク バンドルの設定

このモジュールでは、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータのリンク バンドル インターフェイスの設定について説明します。

リンク バンドルは、1 つ以上のポートを集約したグループで、1 つのリンクとして扱われます。

各バンドルには、1 つの MAC、1 つの IP アドレス、1 つの設定セット（ACL など）があります。POS リンク バンドルに MAC アドレスはありません。イーサネット リンク バンドルだけに MAC アドレスがあります。

（注） Cisco ASR 9000 シリーズ ルータは、レイヤ 2 およびレイヤ 3 リンク バンドルの両方をサポートします。リンク バンドルがレイヤ 3 インターフェイスである場合、IP アドレスが必要です。リンク バンドルがレイヤ 2 インターフェイスの場合、IP アドレスは要求されません。Cisco ASR 9000 シリーズ ルータのリンク バンドル内には、レイヤ 2 およびレイヤ 3 サブインターフェイスが含まれている場合があります。その場合、レイヤ 3 サブインターフェイスには IP アドレスが必要ですが、リンク バンドル インターフェイスには IP アドレスは不要です。POS リンク バンドル化は、レイヤ 3 リンク バンドル上でのみサポートされます。

Cisco ASR 9000 シリーズ ルータは、次のタイプのインターフェイスへのバンドルをサポートします。

- イーサネット インターフェイス
- ASR 9000 SIP-700 ラインカード上の POS インターフェイス。

リンク バンドルの設定の機能履歴

リリース	変更内容
リリース 3.7.2	この機能は、Cisco ASR 9000 シリーズ ルータで導入されました。
リリース 3.9.0	ロード バランシングのサポートが追加されました。 バンドル メンバ リングは、バンドル インターフェイスがシャット ダウンされたときに、新しい err-disable リンク インターフェイス状態および admin-down protocol 状態に追加されます。
リリース 3.9.1	レイヤ 2 リンク バンドルでのレイヤ 3 ロード バランシングのサポートが追加されました。
リリース 4.0.0	次のサポートが追加されました。 • バンドルごとに最大 64 のメンバ リンク。 • IPv6 アドレッシング。 • マルチシャーシ リンク集約。
リリース 4.0.1	リンク集約（LAG）メンバのダイナミック ロード バランシングのサポートが追加されました。 L2VPN コンフィギュレーション モードで **hw-module load-balance bundle l2-service l3-params** コマンドが **load-balancing flow** コマンドに置き換わりました。詳細については、『*Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide*』および『*Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Command Reference*』を参照してください。
リリース 4.1.0	マルチギガビット サービス コントロール ポイントのサポートが追加されました。
リリース 4.2.0	POS インターフェイスのリンク バンドルのサポートが追加されました。

内容

この章で説明する内容は、次のとおりです。

リンク バンドルを設定するための前提条件

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。

リンク バンドルの前提条件は、この機能を設定しようとしているプラットフォームに依存します。ここでは次の内容について説明します。

- 「Cisco ASR 9000 シリーズ ルータでリンク バンドルを設定するための前提条件」（P.217）

Cisco ASR 9000 シリーズ ルータでリンク バンドルを設定するための前提条件

リンク バンドルを設定する前に、次の作業が終了し条件が満たされていることを確認してください。

- インターフェイスの IP アドレス（レイヤ 3 のみ）がわかっていること。
- 設定するバンドルに含めるリンクがわかっていること。
- イーサネット リンク バンドルを設定する場合、ルータに少なくとも次のイーサネット ラインカードのいずれかが搭載されていること。
  - 4 ポート 10 ギガビット イーサネット ラインカード
  - 8 ポート 10 ギガビット イーサネット ラインカード
  - 40 ポート ギガビット イーサネット ラインカード
- POS リンク バンドルを設定する場合は、次のラインカードがルータに搭載されている必要があります。
  - ASR 9K SIP700 ラインカード
- POS リンク バンドル機能は、次の共有ポート アダプタ（SPA）でサポートされます。
  - 2 ポート OC-48 POS/SDH SPA
  - 4 ポート OC-48 POS/SDH SPA
  - 1 ポート OC-192 POS/XFP SPA
  - 4 ポート OC-3 POS-V2 SPA
  - 8 ポート OC-3 POS/SDH SPA
  - 8 ポート OC-12 POS/SDH SPA

（注） 物理インターフェイス、PLIM、およびモジュラ サービス カードの詳細については、『*Cisco ASR 9000 Series Router Hardware Installation Guide*』を参照してください。

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。

- 「リンク バンドルの概要」（P.218）
- 「イーサネット リンク バンドルの機能および互換性のある特性」（P.218）
- 「LACP を通じたリンク集約」（P.220）
- 「マルチシャーシ リンク集約」（P.221）
- 「Load Balancing」（P.228）
- 「QoS およびリンク バンドル」（P.231）
- 「イーサネット リンク バンドル上の VLAN」（P.231）

リンク バンドルの概要

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形成します。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 つで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。帯域幅はパケット フローを中断することなく追加できます。

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。

たとえば、1 つのバンドルに含まれるインターフェイスは、すべてイーサネット インターフェイスであるか、すべて POS インターフェイスになります。イーサネット インターフェイスと POS インターフェイスを同時に含めることはできません。

Cisco IOS XR ソフトウェアでは、次の方法でイーサネット インターフェイスのバンドルを形成できます。

- 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。
- EtherChannel または POS チャネル：リンクをバンドルに参加させるようにユーザが設定するためのシスコ独自のテクノロジー。ただし、バンドル内のリンクに互換性があるかどうかを確認するメカニズムはありません。（EtherChannel はイーサネット インターフェイスに適用され、POS チャネルは POS インターフェイスに適用されます）。

イーサネット リンク バンドルの機能および互換性のある特性

次のリストは Cisco ASR 9000 シリーズ ルータでのイーサネット リンク バンドルの特性と制限の説明です。

- 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。

- バンドル メンバーシップは、単一ルータにインストールされている複数のラインカードにまたがることができます。1 つのイーサネット リンク バンドルで最大 64 本の物理リンクをサポートできます。64 本を超えるリンクをバンドルに追加した場合は、そのリンクのうち 64 本だけが distributing 状態になり、残りのリンクは待機状態になります。
- 1 台の Cisco ASR 9000 シリーズ ルータで最大 256 個のバンドルがサポートされます。
- 1 つのイーサネット リンク バンドル内のリンクは、すべて同じ速度でなければなりません。
- 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。Cisco IOS XR Release 3.9.0 から、バンドル インターフェイスシャット ダウンすると、メンバ リンクは err-disable link interface 状態および admin-down line protocol 状態になります。**show interfaces** コマンドを使用して、バンドル インターフェイスの状態およびそのメンバを表示できます。
- 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。
- LACP が設定されている場合、バンドル内の各リンクでは、異なるメンバに対して異なるキープアライブ周期を設定できます。
- ロード バランシング（メンバー リンク間のデータの分散）は、パケットではなくフロー単位で実行されます。データはバンドル対するそのリンクの帯域幅に比例して、リンクに配信されます。
- QoS がサポートされており、各バンドル メンバーに均等に適用されます。
- CDP キープアライブや HDLC キープアライブなどのリンク層プロトコルは、バンドル内の各リンク上で独立して動作します。
- 上位層プロトコル（ルーティング アップデートや hello など）は、イーサネット インターフェイス バンドルの任意のメンバー リンク上で送信されます。
- 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。 どちらのシステムも直接接続されている必要があります。
- 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。
- 1 つのバンドル内のすべてのリンクは、802.3ad（LACP）または EtherChannel（非 LACP）のいずれかを実行するように設定する必要があります。1 つのバンドル内の混合リンクはサポートされません。
- バンドル インターフェイスには、物理リンクと VLAN サブインターフェイスのみを含めることができます。トンネルは、バンドルのメンバにはできません。
- リンク バンドルでのアクセス コントロール リスト（ACL）の設定は、通常のインターフェイスでの ACL の設定と同じです。
- 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、内部処理によってメンバ リンクが選択され、そのフローのすべてのトラフィックがそのメンバ上で送信されます。

Cisco ASR 9000 シリーズ ルータの POS リンク バンドルの特性

ここでは、Cisco ASR 9000 シリーズ ルータに固有の POS リンク バンドルの特性を示します。

- 各バンドルは、直接接続されたシステムのペア間を結ぶように設定する必要があります。
- 同じバンドルのすべてのメンバーが POS である必要があります。
- Cisco ASR 9000 SIP-700 ラインカードは、最大 32 個の POS リンク バンドルに物理的に対応できます。
- POS リンク バンドルでは、最大 32 本の物理リンクがサポートされますが、これはすべてのリンクの速度が同じである場合です。リンクの速度がそれぞれ異なる場合は、物理リンク数は 32 に達しません。
- 物理インターフェイスだけがバンドル メンバーになることができます。
- すべてのバンドルは静的に設定する必要があります。
- cHDLC カプセル化タイプだけが現時点では POS リンク バンドルに対してサポートされます。
- POS SPA だけが POS リンク バンドルでサポートされます。チャネライズド SPA はサポートされません。
- 上位層プロトコル（ルーティング アップデートや hello など）は、バンドル インターフェイスを通して送信されます。
- ポリサーとキューの帯域幅は、絶対値ではなくパーセンテージで設定する必要があります。
- キュー制限は、バイト単位ではなく時間単位で設定する必要があります。
- POS リンク バンドルの場合、1 つのバンドル内でリンク速度が異なっていてもよく、バンドルのメンバー間で許容される速度の差は、最大 4 倍です。つまり、サポートされる帯域幅比は 4 倍までとなります。

Cisco ASR 9000 シリーズ ルータの POS リンク バンドルの制限事項

ここでは、Cisco ASR 9000 シリーズ ルータに固有の POS リンク バンドルの制限事項を示します。

- Cisco IOS XR Release 4.2.0 では、LACP は POS リンク バンドルに対してサポートされません。
- Cisco IOS XR Release 4.2.0 では、IPv6 および ACL は POS リンク バンドルに対してサポートされません。
- Cisco IOS XR Release 4.2.0 では、マルチキャスト ルーティングは POS リンク バンドルに対してサポートされません。

LACP を通じたリンク集約

オプションの Link Aggregation Control Protocol（LACP）は IEEE 802 規格で定義されています。LACP では、2 台の直接接続されたシステム（ピア）間で通信し、バンドル メンバーの互換性が確認されます。Cisco ASR 9000 シリーズ ルータの場合、ピアは、別のルータまたはスイッチにすることができます。LACP は、リンク バンドルの動作状態を監視し、次のことを確認します。

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

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

IEEE 802.3ad 規格

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

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

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

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

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

ピア システムからの情報はローカル システムの情報と組み合わされ、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。

Cisco ASR 9000 シリーズ ルータのバンドル MAC アドレスは、バックプレーンの一連の予約済み MAC アドレスに由来します。この MAC アドレスは、バンドル インターフェイスが存在する限り、このバンドルに付いたままになります。バンドルは、ユーザが別の MAC アドレスを設定するまで、この MAC アドレスを使用します。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。

マルチシャーシ リンク集約

マルチシャーシ リンク集約（MC-LAG）機能は、キャリア イーサネット ネットワークでのエンドツーエンドのシャーシ間冗長ソリューションを提供します。MC-LAG に、（第 3 の）接続デバイスから見たときに、単一の LAG として共同で動作する 2 台のデバイスが含まれているため、デバイス レベルとリンク レベルの冗長性が提供されます。

そのためには、2 台のデバイスは相互に強調して、相手側のデバイスに対して単一の（2 台のデバイスにスパニングする）LACP バンドルとして表示されるようにします。転送ループのリスクを排除するため、任意の時点でのトラフィックの転送は 1 台のデバイスのみで行います。障害が発生すると、これらのデバイスは協調してスイッチオーバーを実行し、リンク LACP 状態を操作してトラフィックの転送左記デバイスを変更します。

コア ネットワーク内の既存の疑似回線冗長性は、次の内容に基づいてアクセス ネットワークの冗長性と協調します。

- マルチシャーシ Link Aggregation Control Protocol（mLACP）
- シャーシ間通信プロトコル（ICCP）

mLACP プロトコルは、2 台のデバイス間の予想される動作を定義し、シャーシ間制御プロトコル（ICCP）を使用して TLV を交換して、動作で使用するピア デバイスを識別します。プロバイダー ネットワークのエッジでは、標準 LACP だけをサポートする単純なカスタマー エッジ（CE）デバイスが、2 台のプロバイダー エッジ（PE）デバイスに接続されます。したがって CE デバイスはデュアルホーム接続となり、プロバイダー側からより適切な L2 冗長性が提供されます。mLACP 用語では、CE デバイスにはデュアルホーム接続デバイス（DHD）と呼ばれ、各 PE デバイスは接続ポイント（POA）と呼ばれます。バンドルに対する POA 転送トラフィックは、そのバンドルのアクティブ デバイスであり、その他の POA はスタンバイ デバイスです。

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。

シャーシ間通信プロトコル

図 4 に、シャーシ間通信プロトコル（ICCP）をグラフィカルに表示されます。

図 4　　ICCP プロトコル

2 つの POA がシャーシ間通信プロトコル（ICCP）を使用して LDP リンクを介して相互に通信します。ICCP は、冗長グループの POA 間で LDP セッションが作成される LDP ベースのプロトコルであり、ICCP メッセージは LDP セッションを介して伝送されます。冗長グループの PE ルータは、シングル ホップ（直接接続）、または相互にマルチホップである場合があります。ICCP プロトコルは設定を管理し、冗長グループを制御します。また、ICCP 接続を確立、維持、解除します。ICCP プロトコルは route-watch を使用して、特定の冗長グループの PE への接続をモニタリングします。これはコア分離の障害をトラッキングする役割もあります。この場合、すべてのクライアント アプリケーションに対して、障害（コアの分離およびアクティブ PE 障害）が通知されます。

ICCP を動作させるには、デバイスは冗長グループ（RG）のメンバとして設定します。

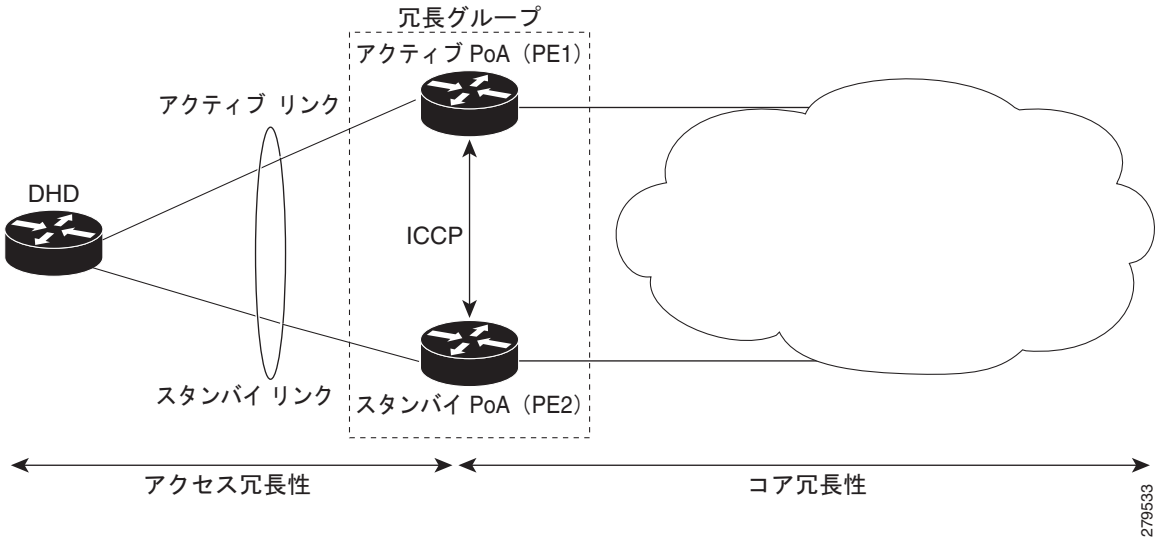
（注） mLACP の設定では、2 台のデバイスは、各 RG のメンバとして設定されます（1 つのメンバだけを残してデバイスレベルのエラーが発生するまで）。ただし、各デバイスは複数の RG のメンバにすることができます。

各冗長グループでは、POA の mLACP ピアは、ICCP を介した mLACP を使用して通信している相手側である、そのグループ内の別の POA になります。各バンドルについて、両端の POA および DHD は標準 LACP プロトコルを使用して通信する LACP パートナーです。

アクセス ネットワーク冗長モデル

マルチシャーシ Link Aggregation Control Protocol（mLACP）をベースとした、カスタマー エッジ（CE）デバイスまたはアクセス ネットワークとプロバイダー エッジ（PE）デバイス間の冗長性は、CE が 2 台の PE ルータに接続できるようにすることによって実現されます。2 台の PE ルータは、ICCP を介してデータを同期します。そのため、これらは CE に対して 1 つのデバイスとして表示されます。

図 5　　mLACP/ICCP 冗長モデル



CE は、デュアルホーム接続デバイス（DHD）とも呼ばれ、PE は接続ポイント（POA）とも呼ばれます。単一 DHD に接続された POA のペアは、冗長グループ（RG）を形成します。

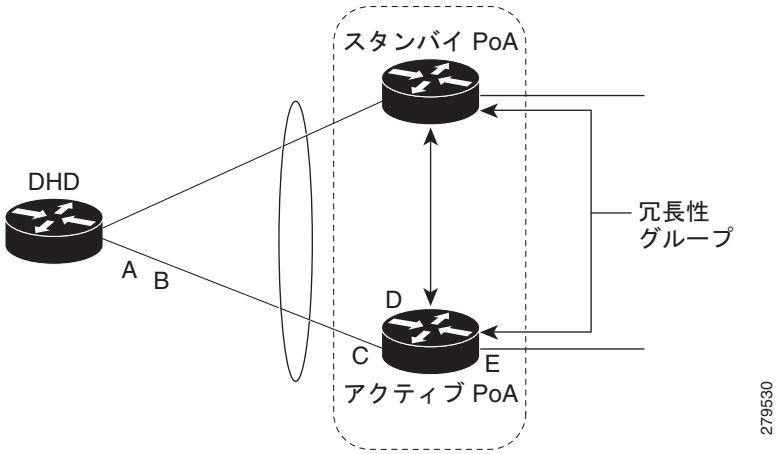
常に、1 つの POA だけがバンドルに対してアクティブです。DHD とアクティブ POA 間のリンクのセットだけが、アクティブにトラフィックを送信します。DHD とスタンバイ POA 間のリンクのセットはトラフィックを転送しません。マルチシャーシ リンク バンドルのソフトウェアは、アクティブ POA への接続が失敗したことを検出すると、スタンバイ POA がアクティブ POA になり、トラフィックが DHD と新しくアクティブになった POA 間のリンクを使用してフローするようにトリガーします。

ICCP プロトコルは、アクティブ POA およびスタンバイ POA 間で動作し、POA がまたは設定を調整し、いずれをアクティブ POA にするかを決定し、POA がアクティブになるようにトリガーします。2 つの POA で動作するアプリケーション（mLACP、IGMP スヌーピング、DHCP スヌーピングまたは ANCP）は、ICCP を使用して状態を同期させます。

障害モード

mLACP 機能には、ポート障害、リンク障害、およびノード障害からの保護によるネットワーク復元力が備わっています。図 6 に、さまざまな障害モードを示します。

図 6　　障害モード



障害のカテゴリは次のとおりです。

- A：DHD アップリンク ポート障害。POA に接続する DHD 上のポートの障害です。
- B：DHD アップリンク障害。DHD と POA 間の接続の障害です。
- C：アクティブ POA のダウンリンク ポートの障害。
- D：アクティブ POA のノード障害。
- E：アクティブ POA アップリンク障害（ネットワークの分離）。アクティブ POA とコア ネットワーク間のリンクの障害です

コア ネットワーク冗長モデル

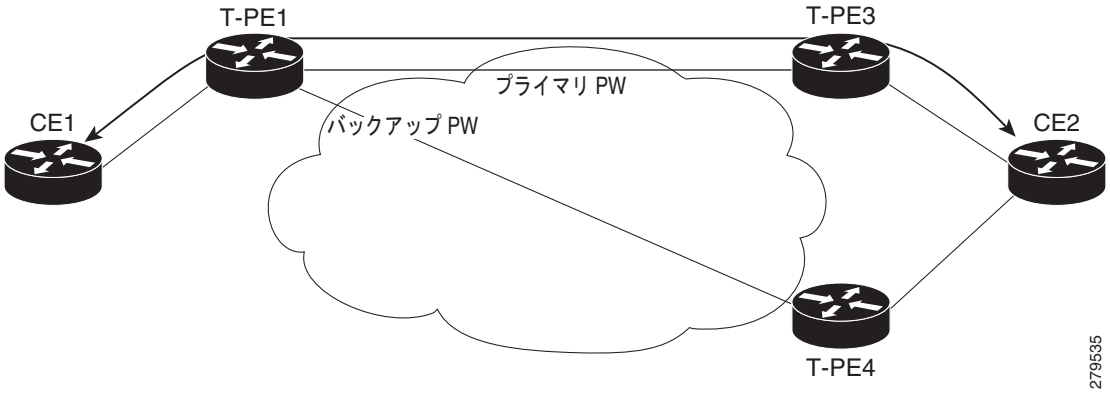
この項では次の内容について説明します。

- 一方向疑似回線冗長性
- 双方向疑似回線冗長性

一方向疑似回線冗長性

図 7 に、VPWS 一方向疑似回線冗長性モデルを示します。疑似回線の片端だけがバックアップ疑似回線によって保護されます。

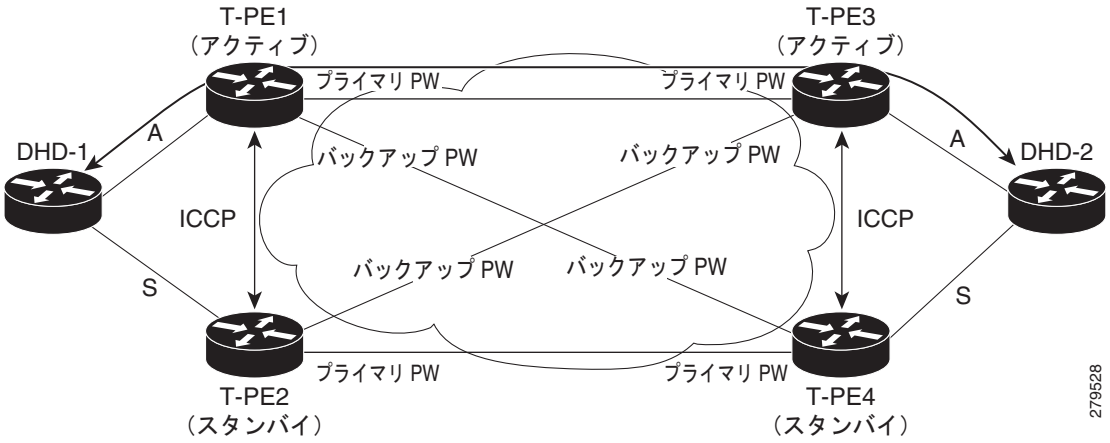
図 7　VPWS 一方向疑似回線冗長性



双方向疑似回線冗長性

図 8 に、VPWS 双方向疑似回線冗長性モデルを示します。このトポロジでは、PW の端にある各 T-PE は各プライマリおよびバックアップ PW があります。PW の状態は、DHD と PE 間の mLACP リンクの状態と調整されます。

図 8　VPWS 双方向疑似回線冗長性



スイッチオーバー

POA のアクティブ/スタンバイ ロールを変更するスイッチオーバーは、動的優先権管理またはブルート フォース動作を使用して実行されます。

動的優先権管理

動的優先権管理には、メンバ リンクの LACP ポート プライオリティを処理する POA 間の調整が含まれます。2 つのプライオリティ値が各リンクについて追跡されます。

- 明示的に設定するか、デフォルトの 32768 で設定する、設定されたプライオリティ
- LACP ネゴシエーションで使用される運用上のプライオリティ。スイッチオーバーが発生している場合、設定されたプライオリティと異なる場合があります。

常に、ハイ プライオリティ LACP リンクはロー プライオリティ LACP リンクより先に選択されます。これは、運用上のプライオリティを操作して、（POA および DHD の）標準 LACP 選択ロジックで、両端の目的のリンクが強制的に選択されるようにできることを意味します。

たとえば、DHD が各 POA に対して 2 個のリンクを持ち、各 POA の最小アクティブ リンクが 2 に設定されている場合を検討します。（これはアクティブ リンク数が 2 を下回るとバンドルが POA でダウンすることを意味します）。メンバ リンクの運用上のプライオリティは、POA-1 で 1、POA-2 で 2 です。つまり、POA-1 はアクティブ（ハイ プライオリティ）であり POA-2 のリンクはスタンバイ状態のままになっています。スイッチオーバーのイベント シーケンスは次のとおりです。

1. リンクの障害が POA-1 で発生し、アクティブ リンクの数が最小の 2 未満になります。
2. POA-1 は、両リンクの運用上のプライオリティを 3 に変更し、これにより POA 2 のリンクがハイ プライオリティになります。
3. POA-1 は DHD に LACP メッセージ、POA-2 に mLACP にメッセージを送信し、両方のデバイスに変更を通知します。
4. 現在、POA-2 の方がハイ プライオリティになるため、DHD は POA-2 に接続されたリンクをアクティブ化しようとします。
5. また、POA-2 はそのリンクが最も高いプライオリティであることも確認し、DHD へのリンクをアクティブにします。

この時点でスイッチオーバーが完了しました。

ブルート フォースの動作

ブルート フォースのスイッチオーバーでは、ポート プライオリティは変更されません。その代わり、障害の発生した POA は LACP 経由で DHD に *Dying Gasp* を送信し、強制的にリンクが選択解除されるようにします。次に、そのリンクでの LACP の通信を終了します。これにより、選択できるリンクとして DHD と POA-2 間のリンクだけが残ります。したがって、両端でこれらのリンクを選択します。

MC-LAG のトポロジ

ここでは、サポートされている MC-LAG トポロジについて説明します。

図 9　　冗長グループの VPWS 一方向疑似回線冗長性

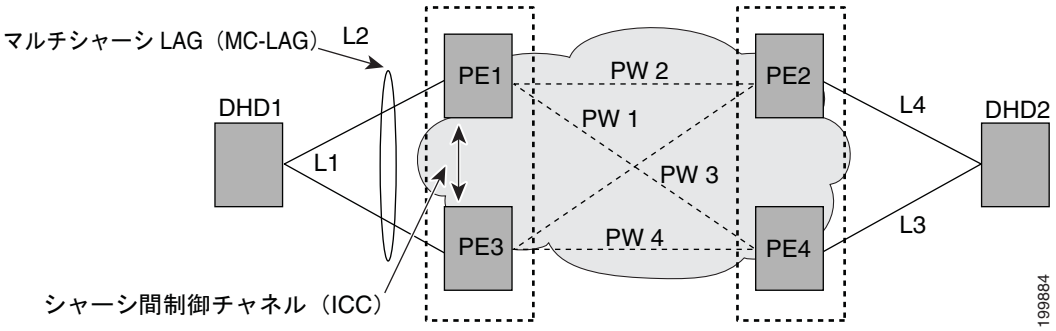


図 10　　VPWS 双方向疑似回線冗長性

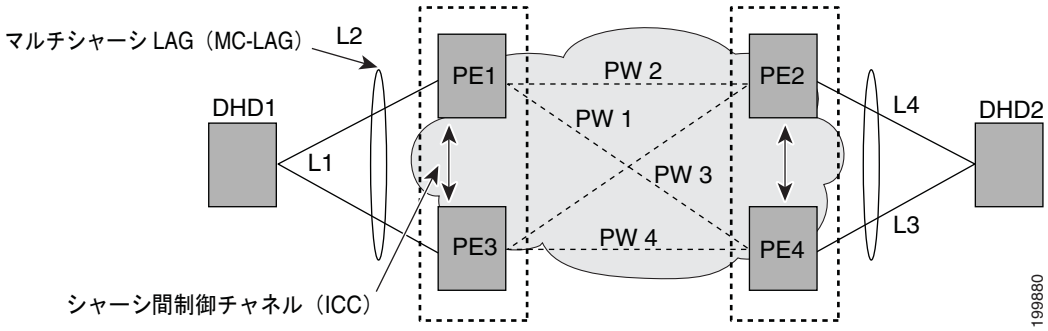


図 11　　1 つの冗長性グループの VPLS 疑似回線

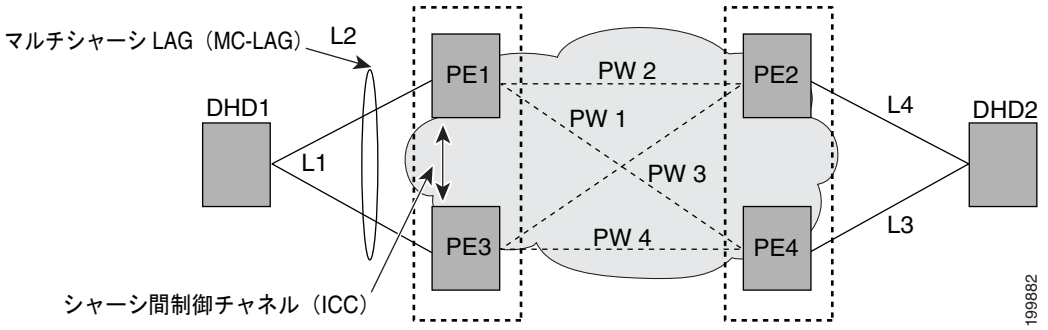


図 12　　2 つの冗長性グループの VPLS 疑似回線

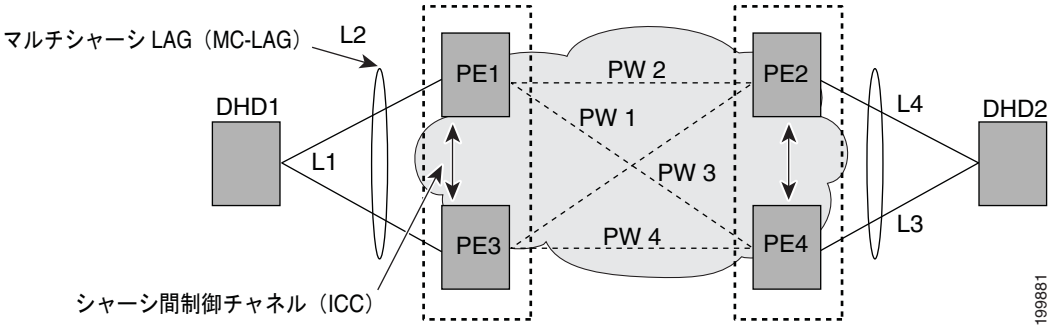


図 13　H-VPLS：アクセス疑似回線上の EoMPLS

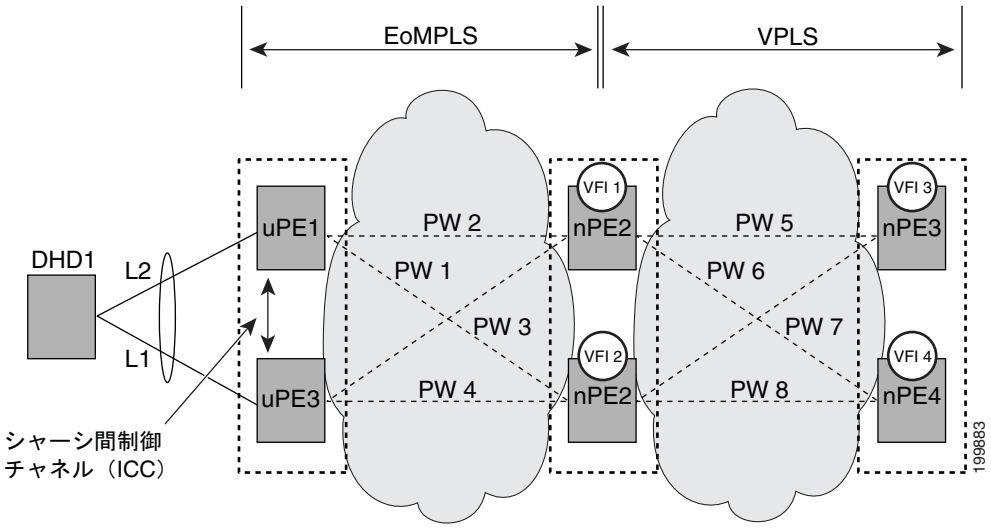
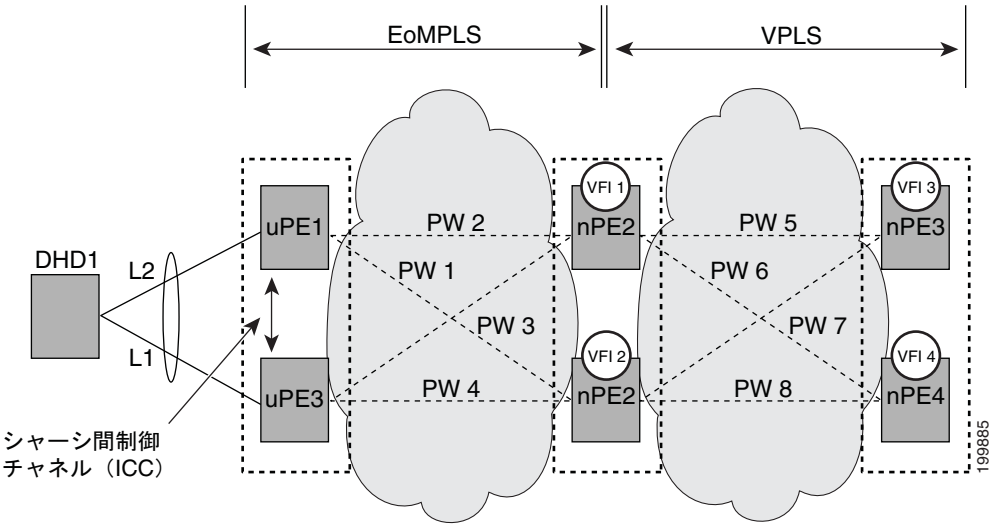


図 14　H-VPLS：uPE 上の VPWS 疑似回線との nPE 上のアクセス疑似回線



Load Balancing

ロード バランシングは、特定のパラメータに基づいて複数のリンクのトラフィックを配信するトランスポート メカニズムです。Cisco ASR 9000 シリーズ ルータは、レイヤ 2、レイヤ 3、およびレイヤ 4 ルーティング情報を使用して、バンドル内のすべてのリンクのロード バランシングをサポートします。

ここでは、リンク バンドルのロード バランシング サポートについて説明します。

Cisco ASR 9000 シリーズ ルータでのその他のロード バランシング形式の詳細については、次の資料を参照してください。

- レイヤ 3 およびレイヤ 4 ルーティング情報を使用した非バンドル インターフェイス上のフローごとのロード バランシング：『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Configuration Guide*』を参照してください。

- Cisco IOS XR 4.0.1 以降の疑似回線（PW）ロード バランシング：『*Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide*』を参照してください。

リンク バンドルのレイヤ 2 入力ロード バランシング

デフォルトで、レイヤ 2 リンク バンドルのロード バランシングは、着信パケット ヘッダーの送信元および宛先 MAC アドレス（SA/DA）フィールドに基づいて行われます。表 1 に、デフォルト モード、EFP ベース、フローベースのいずれのロード バランシングが使用中であるかに応じて、レイヤ 2 での着信トラフィックのロード バランシングに使用されるパラメータのサマリーを表示します。

フローごとのロード バランシングは、バンドルのすべてのリンクでサポートされます。この方法では、ルータが、ハッシュ計算で決定されたバンドル内のリンクの 1 つを経由してパケットを配信することによって、ロード シェアリングが実行されます。ハッシュ計算は特定のパラメータに基づいたリンク選択のアルゴリズムです。

標準のハッシュ計算は、次のパラメータを使用する 5 タプル ハッシングです。

- IP 送信元アドレス
- IP 宛先アドレス
- ルータ ID
- レイヤ 4 送信元ポート
- レイヤ 4 宛先ポート

フローごとのロード バランシングをイネーブルにすると、特定の送信元と宛先間のペア間のすべてのパケットは、使用可能なリンクが複数あっても、同じリンクを通過します。フローごとのロード バランシングは、特定の送信元と宛先ペアのパケットが順序どおりに到達できるようにします。

（注） マルチキャスト トラフィックに対するロード バランシングは、発信インターフェイスがリンク バンドル インターフェイスまたはサブインターフェイスの場合だけ適用されます。

表 1　着信トラフィックのバンドルロード バランシング

入力ユニキャスト、フラッド、またはマルチキャスト トラフィック	パラメータ	設定
デフォルト	• 送信元 MAC アドレス • 宛先 MAC アドレス	n/a
EFP ベース自動モード	xconnect の XID	自動モードは、**bundle load-balancing hash auto** コマンドを使用してイネーブルにします。
ユーザ ハッシュを使用する EFP ベース	ユーザ ハッシュ	ユーザ ハッシュがバンドルは **bundle load-balancing** *hash-value* コマンドで設定します。
IP 送信元と宛先と使用するフローベース	• 送信元 IP アドレス • 宛先 IP アドレス	L2VPN **load-balancing flow src-dst-ip** コマンドを使用してイネーブルにします。
MAC 送信元と宛先と使用するフローベース	• 送信元 MAC アドレス • 宛先 MAC アドレス	L2VPN **load-balancing flow src-dst-mac** コマンドを使用してイネーブルにします。

リンク バンドルのレイヤ 3 出力ロード バランシング

レイヤ 3 ロード バランシングのサポートは、Cisco ASR 9000 シリーズ ルータの Cisco IOS XR 3.9.1 から開始され、Cisco IOS XR Release 4.0.1 で変更が導入されました。

Cisco IOS XR Release 4.0.1 よりもの前のレイヤ 3 ロード バランシング

Cisco IOS XR 3.9.1 から Cisco IOS XR 4.0 では、リンク バンドルのレイヤ 3 ロード バランシングは、パケットの IPv4 送信元および宛先アドレスに基づいて、イーサネット フロー ポイント（EFP）で実行されます。レイヤ 3 サービス固有のロード バランシングが設定されている場合、すべての出力バンドルは IPv4 送信元および宛先アドレスにロード バランシングされます。パケットに IPv4 アドレスがない場合、デフォルトのロード バランシングが使用されます。

リンク バンドルのレイヤ 3 ロード バランシングは、次のコマンドを使用して、グローバルにイネーブルになります。

hw-module load-balance bundle l2-service l3-params

Cisco IOS XR リリース 4.0.1 以降のレイヤ 3 ロード バランシング

リンク バンドルのレイヤ 3 ロード バランシングは、発信インターフェイスがバンドルまたはバンドル サブインターフェイスのときに実行されます。5 タプル ハッシングは、次のパラメータを使用して、バンドルのメンバ リンク間のロード バランシングに使用されます。

- IP 送信元アドレス
- IP 宛先アドレス
- ルータ ID
- レイヤ 4 送信元ポート
- レイヤ 4 宛先ポート

入力ラインカードはバンドル メンバを選択し、選択したバンドル メンバに対応するラインカードおよびネットワーク プロセッサ（NP）パケットを転送します。入力と出力の両方のラインカードに同じハッシュ値が使用されます。したがって、出力ラインカードでもメンバ選択を行う場合でも、入力ラインカードによって選択された同じバンドル メンバが選択されます。

マルチキャスト IPv4 および IPv6 トラフィック

発信マルチキャスト IPv4 または IPv6 トラフィックの場合は、出力ラインカードのセットがシステムによって事前に決定されます。バンドルまたはバンドル インターフェイスのサブインターフェイスが発信インターフェイスの場合、システムはマルチキャスト グループ アドレスに基づいてルートの各発信インターフェイスのバンドル メンバを選択します。これは、特定のルートで特定のトラフィック シーケンスを維持しながら、異なるバンドル メンバに対するマルチキャスト ルーテッド トラフィックの負荷分散を実行する場合に役立ちます。

バンドル メンバが出力ラインカード内の複数の NP に分散した場合、出力ラインカードは同じアプローチを使用して NP を選択します。

パケットが出力 NP に到着すると、5 タプル ハッシュを使用して、各パケットの NP 内のバンドル メンバを選択します。これにより、NP 内のバンドル メンバの状態変更の復元性が向上します。

LAG のダイナミック ロード バランシング

Cisco IOS XR Release 4.0.1 以降の Cisco ASR 9000 シリーズ ルータでは、リンク集約（LAG）メンバ間のダイナミック ロード バランシング方式がサポートされています。ダイナミック ロード バランシングによって、バンドル内の現在のアクティブ メンバの数に基づいて、リンク選択のハッシュ アルゴリズムに最大 64 のリンクが含まれます。

QoS およびリンク バンドル

Cisco ASR 9000 シリーズ ルータでは、QoS が入力または出力方向のバンドルに適用される場合、各メンバ インターフェイスに QoS が適用されます。Cisco ASR 9000 シリーズ ルータでのリンク バンドルの QoS の設定方法の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Configuration Guide*』および『*Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Command Reference*』を参照してください。

イーサネット リンク バンドル上の VLAN

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

- 各バンドルに許可される VLAN の最大数は、4096 です。
- 各ルータに許可されるバンドル VLAN の最大数は、16384 です。

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

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

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

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

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

レイヤ 2 EFP は次のように設定します。

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

レイヤ 3 VLAN サブインターフェイスは次のように設定します。

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

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

リンク バンドルの設定の概要

リンク バンドルの設定プロセスの一般的な概要を次の手順に示します。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 64 個のリンクを追加できます。

（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。

カードのフェールオーバー時のノンストップ フォワーディング

Cisco IOS XR ソフトウェアは、アクティブおよびスタンバイ RSP カード間でのフェールオーバー時のノンストップ フォワーディングをサポートしています。ノンストップ フォワーディングを使用すると、フェールオーバーが発生したときにリンク バンドルの状態が変化しません。

たとえば、アクティブな RSP が障害になった場合、スタンバイ RSP が動作可能になります。障害になった RSP のコンフィギュレーション、ノードの状態、チェックポイント データは、スタンバイ RSP に複製されます。スタンバイ RSP がアクティブ RSP になったとき、バンドルされたインターフェイスはすべて存在します。

（注） フェールオーバー先は常にスタンバイ RSP です。

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

リンクのフェールオーバー

バンドルのメンバ リンクの 1 つに障害が発生すると、トラフィックは動作可能な残りのメンバ リンクにリダイレクトされ、トラフィック フローは中断されません。

マルチギガビット サービス コントロール ポイント

マルチギガビット サービス コントロール ポイント（MGSCP）は、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータの特定のリンク バンドルおよび転送機能を使用する導入モデルで、Cisco Service Control Engine（SCE）デバイスのブロードバンド加入者トラフィックのロード バランシング、クラスタリング、および冗長性をサポートします。

Cisco SCE プラットフォームは、ブロードバンド加入者にユーザ認可、レポート、およびアプリケーション帯域幅測定などのさまざまなサービスを提供するために使用されます。これは、アプリケーションおよび加入者の認識に基づいてステートフル処理メカニズムを使用して IP トラフィックを管理します。このステートフルネスを維持するには、SCE プラットフォームがセッションのアップストリームおよびダウンストリームの両方のフローをキャプチャして、それを分類し、アプリケーション レベルでレイヤ 7 プロセスを提供する必要があります。

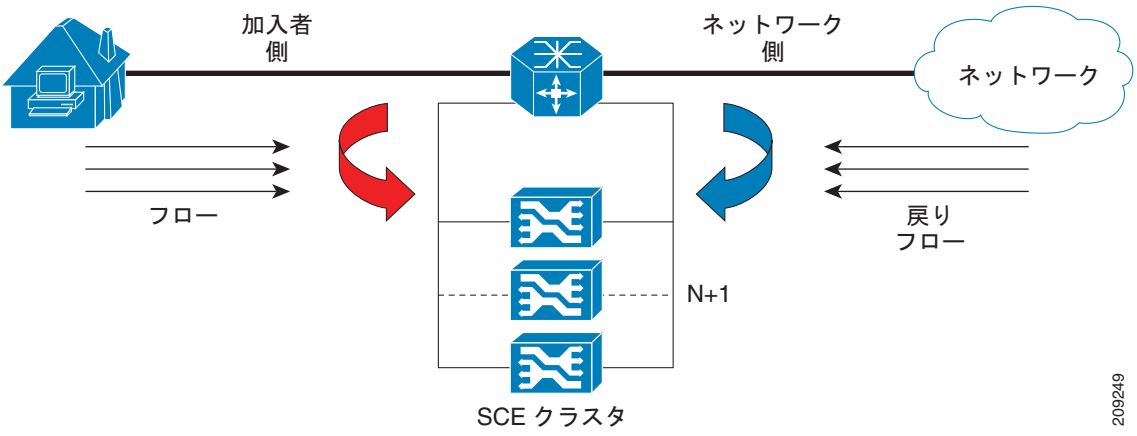
FTP または Session Initiation Protocol（SIP）などの、フローのバンドルとあわせて実装されているアプリケーションを処理するには、SCE プラットフォームは、このアプリケーションでセッションを構成するすべてのフローを処理する必要があります。また、SCE プラットフォームが加入者ごとのレポートまたは制御（*加入者認識*とも呼ばれる）を実装するように設定されている場合、特定の加入者が生成するすべてのトラフィック フローを処理する必要があります。

この加入者レベルへのステートフル処理が理由で、SCE プラットフォームは「bump-in-the-wire」トポロジのネットワーク内で、レイヤ 2 およびレイヤ 3 透過性を実現するために実装されます。ただし、SCE プラットフォームがサポートする必要のある帯域幅にあわせてブロードバンド加入者数が増加すると、このソリューションが、非対称ルーティングが実装されていることが多く、1 つのセッションの 2 方向（または特定の加入者の多数のフロー）が異なるリンク間で分割される一般的なネットワーク環境に挿入された場合、ソリューションのスケーリングにおいて特定の問題が生じます。

Cisco ASR 9000 シリーズ ルータの MGSCP ソリューションは、すべての加入者トラフィックが同じバンドルのメンバ リンクを介して送信されるリンク バンドルを使用してルータに接続しているクラスタの、複数の SCE デバイスを拡張するためのトポロジを提供することで、これらの要件を満たしています。また、MGSCP は、ロード バランシングと冗長性の利点があります。

図 15 に、Cisco ASR 9000 シリーズ ルータが加入者とコア ネットワーク間に接続され、接続された SCE クラスタのディスパッチャとして動作する、MGSCP の基本的なネットワーク トポロジを示しています。N+1 表記は、SCE の両側にある他のアクティブ リンクに対するバックアップ（または保護）リンクを示します。

図 15　基本的な MGSCP ネットワーク トポロジ



リンク バンドルの設定方法

ここでは、次の手順について説明します。

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 リンク バンドルでは MAC アカウンティングはサポートされていません。

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

手順の概要

イーサネット リンク バンドルを作成するには、次の手順のように、バンドルを作成し、そのバンドルにメンバー インターフェイスを追加します。

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **ipv4 address** *ipv4-address mask*
4. **bundle minimum-active bandwidth** *kbps* （任意）
5. **bundle minimum-active links** *links* （任意）
6. **bundle maximum-active links** *links* （任意）
7. **exit**
8. **interface** {**GigabitEthernet** | **TenGigE**}
9. **bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}
10. **no shutdown**
11. **exit**
12. ステップ 2 で作成した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。
13. **end**
    または
    commit
14. **exit**
15. **exit**

16. 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。
17. **show bundle Bundle-Ether** *bundle-id* [**reasons**]
18. **show lacp Bundle-Ether** *bundle-id*

イーサネット リンク バンドルでの EFP ロード バランシングの設定

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

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

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

手順の概要

イーサネット リンク バンドルの EFP ロード バランシングを設定するには、次の手順を実行します。

1. **configure**
2. **hw-module load-balance bundle l2-service l3-params**
3. **interface Bundle-Ether** *bundle-id* **l2transport**
4. **bundle load-balance hash** *hash-value* [**auto**]
5. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**hw-module load-balance bundle l2-service l3-params** **例：** `RP/0/RSP0/CPU0:router(config)# hw-module load-balance bundle l2-service l3-params`	（任意）レイヤ 2 リンク バンドルでのレイヤ 3 ロード バランシングをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 3	**interface Bundle-Ether** *bundle-id* **l2transport** **例：** `RP/0/RSP0/CPU0:router#(config)# interface Bundle-Ether 3 l2transport`	指定した *bundle-id* を使用し、レイヤ 2 転送をイネーブルにして、新しいイーサネット リンク バンドルを作成します。 指定できる範囲は 1 ～ 65535 です。
ステップ 4	**bundle load-balance hash** *hash-value* [**auto**] **例：** `RP/0/RSP0/CPU0:router(config-subif)# bundle load-balancing hash 1` または `RP/0/RSP0/CPU0:router(config-subif)# bundle load-balancing hash auto`	バンドルの固定メンバのすべての出力トラフィックを、同じ物理メンバ リンクを通過するように設定します。 • *hash-value*：このバンドルのすべての出力トラフィックが通過する物理メンバ リンクを指定する数値。値は 1 ～ 8 です。 • **auto**：このバンドルのすべての出力トラフィックが通過する物理メンバ リンクが自動的に選択されます。
ステップ 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

VLAN バンドルの設定

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

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

これらの作業について、以降の手順で詳しく説明します。

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

手順の概要

VLAN リンク バンドルの作成について、次の手順で説明します。

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

POS リンク バンドルの設定

ここでは、POS リンク バンドルの設定方法について説明します。

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

手順の概要

バンドルされた POS インターフェイスの作成では、次のステップに示すように、バンドルとメンバー インターフェイスの両方を設定します。

1. **configure**
2. **interface Bundle-POS** *bundle-id*
3. **ipv4 address** *ipv4-address mask*
4. **bundle minimum-active bandwidth** *kbps*
5. **bundle minimum-active links** *links*
6. **bundle maximum-active links** *links* [**hot-standby**]
7. **exit**
8. **interface POS** *interface-path-id*
9. **bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}]
10. **bundle port-priority** *priority*
11. **no shutdown**
12. **exit**
13. ステップ 2 で作成したバンドルに さらにイーサネット インターフェイスを追加するには、ステップ 19 から 21 を繰り返します。
14. **end**
    または
    commit
15. **exit**
16. **exit**
17. 接続のリモート エンドでステップ 1 から 23 を実行します。
18. **show bundle Bundle-POS** *bundle-id* [**reasons**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**interface Bundle-POS** *bundle-id* **例：** RP/0/RSP0/CPU0:router#(config)#interface Bundle-POS 2	名前と新たにバンドルされた POS インターフェイスを設定します。 インターフェイス コンフィギュレーション サブモードを開始します。ここから、インターフェイス固有のコンフィギュレーション コマンドを実行します。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/RSP0/CPU0:router(config-if)# ipv4 address 10.1.2.3 255.0.0.0	ip address コンフィギュレーション サブコマンドを使用して、IP アドレスとサブネット マスクを仮想インターフェイスに割り当てます。
ステップ 4	**bundle minimum-active bandwidth** *kbps* **例：** RP/0/RSP0/CPU0:router(config-if)# bundle minimum-active bandwidth 620000	（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/RSP0/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/RSP0/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 対応リンク間でのスイッチオーバーが、専用の最適化に従って実装されることを指定します。 **（注）** アクティブおよびスタンバイ リンクの優先順位は、**bundle port-priority** コマンドの値で決まります。
ステップ 7	**exit**	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 POS** *interface-path-id* **例：** RP/0/RSP0/CPU0:router(config)# interface POS 0/1/0/0	POS 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、POS インターフェイス名を指定します。interface-path-id は、*rack/slot/module/port* の形式で指定します。

	コマンドまたはアクション	目的
ステップ 9	**bundle id** *bundle-id* [**mode** {**active** | **on** | **passive**}] **例：** `RP/0/RSP0/CPU0:router(config-if)# bundle-id 3`	指定したバンドルにリンクを追加します。 バンドル上でアクティブ LACP またはパッシブ LACP をイネーブルにするには、オプションの **mode active** キーワードまたは **mode passive** キーワードをコマンド文字列に追加します。 LACP をサポートせずにバンドルにリンクを追加するには、オプションの **mode on** キーワードをコマンド文字列に追加します。 **（注）** **mode** キーワードを指定しない場合、デフォルトのモードは **on** になります（LACP はポート上で動作しません）。
ステップ 10	**bundle port-priority** *priority* **例：** `RP/0/RSP0/CPU0:router(config-if)# bundle port-priority 1`	（任意）**bundle maximum-active links** コマンドに 1 を設定する場合、アクティブ リンクの優先順位を最も高くし（最も小さい値）、スタンバイ リンクの優先順位を 2 番目に高く（次に小さい値）する必要があります。たとえば、アクティブ リンクの優先順位を 1 に設定し、スタンバイ リンクの優先順位を 2 に設定します。
ステップ 11	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-if)# no shutdown`	シャットダウン コンフィギュレーションを削除します。これにより、インターフェイスが管理上ダウンになります。**no shutdown** コマンドは、コンフィギュレーション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/RSP0/CPU0:router# exit`	POS 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	バンドルにさらにリンクを追加するには、ステップ 19 ～ 21 を繰り返します。	（任意）ステップ 2 で作成したバンドルにさらにリンクを追加します。

	コマンドまたはアクション	目的
ステップ 14	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、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。 • 設定変更を実行コンフィギュレーション ファイルに保存し、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/RSP0/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/RSP0/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 23 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-POS** *number* **例：** `RP/0/RSP0/CPU0:router# show bundle Bundle-POS 1`	（任意）指定した POS リンク バンドルに関する情報を表示します。

マルチシャーシ リンク集約の設定

マルチシャーシ リンク集約（MC-LAG）を設定するには、次の作業を行います。

シャーシ間通信プロトコルの設定

シャーシ間通信プロトコル（ICCP）を設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **redundancy iccp group** *group-id*
3. **member neighbor** *neighbor-ip-address*
4. **backbone interface** *interface-type-id*
5. **isolation recovery-delay** *delay*
6. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**redundancy iccp group** *group-id* **例：** `RP/0/RSP0/CPU0:router#(config-redundancy-iccp-group) # redundancy iccp group 100`	ICCP 冗長性グループを追加します。

	コマンドまたはアクション	目的
ステップ 3	**member neighbor** *neighbor-ip-address* **例：** `RP/0/RSP0/CPU0:router#(config-redundancy-iccp-group) # member neighbor 10.1.1.1`	ICCP メンバを設定します。 この冗長グループの ICCP ピアです。冗長性グループごとに 1 つのネイバーだけを設定できます。IP アドレスは、ネイバーの LDP router-ID です。この設定は ICCP が機能するためには必須です。
ステップ 4	**backbone interface** *interface-type-id* **例：** `RP/0/RSP0/CPU0:router#(config-redundancy-iccp-group) # backbone interface GigabitEthernet0/1/0/2`	ICCP バックボーン インターフェイスを設定します。 これはネットワーク コアからの分離を検出するオプションの設定で、問題が発生している POA がアクティブな場合はピア POA へのスイッチオーバーをトリガーします。複数のバックボーン インターフェイスは、各冗長グループ用に設定できます。すべてのバックボーン インターフェイスがアップでない場合、これはコア分離の表示です。1 つ以上のバックボーン インターフェイスがアップの場合、POA はネットワークのコアから分離されていません。バックボーン インターフェイスは、通常は L2VPN 疑似回線が使用できるインターフェイスです。

	コマンドまたはアクション	目的
ステップ 5	**isolation recovery-delay** *delay* **例：** `RP/0/RSP0/CPU0:router#(config-redundancy-iccp-group) # isolation recovery-delay 30`	分離パラメータを設定し、障害からの復旧後に分離状態をクリアするまでの遅延を指定します。 分離リカバリ遅延タイマーはコア分離状態がクリアされたときに開始します。タイマーの期限が切れると、POA は（バンドルのリカバリ遅延タイマーなどの他の条件に応じて）アクティブ POA として引き継ぐことができます。これにより、次が可能になります。 • バックボーン インターフェイスがアップした後のネットワーク コアの再コンバージェンス • MCLAG バンドルが過度にフラップしないように、POA が入るべき状態を把握できるようにするための ICCP 状態の交換。 この設定は、オプションです。設定しない場合、遅延はデフォルトで 180 秒に設定されます。
ステップ 6	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-redundancy-iccp-group)# end` または `RP/0/RSP0/CPU0:router(config-redundancy-iccp-group)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

マルチシャーシ Link Aggregation Control Protocol セッションの設定

マルチシャーシ Link Aggregation Control Protocol セッションをイネーブルにするには、次の作業を実行します。

手順の概要

1. **configure**
2. **redundancy iccp group** *group-id*
3. **mlacp system mac** *mac-id*
4. **mlacp system priority** *priority*
5. **mlacp node** *node-id*
6. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**redundancy iccp group** *group-id* **例：** `RP/0/RSP0/CPU0:router#(config-redundancy-iccp-group)` `# redundancy iccp group 100`	ICCP 冗長性グループを追加します。
ステップ 3	**mlacp system mac** *mac-id* **例：** `RP/0/RSP0/CPU0:router#(config-redundancy-iccp-group)` `# mlacp system mac 1.1.1`	LACP システム ID がこの ICCP グループで使用されるように設定します。 **（注）** *mac-id* は、POA で使用される LACP システム LAG-ID のユーザ設定値です。*mac-ids* は、両方の POA で同じ値にすることを強く推奨します。異なるグループごとに異なる LAG ID を持つことができます。
ステップ 4	**mlacp system priority** *priority* **例：** `RP/0/RSP0/CPU0:router#(config-redundancy-iccp-group)` `# mlacp system priority 10`	LACP システム プライオリティがこの ICCP グループで使用されるように設定します。 **（注）** POA のシステム プライオリティは、DHD の LACP LAG ID よりも低い数値（ハイ プライオリティ）に設定することを推奨します。DHD の方がシステム プライオリティが高い場合、および、動的優先権管理が機能せず、ブルート フォース スイッチオーバーが自動的に使用されます。

	コマンドまたはアクション	目的
ステップ 5	**mlacp node** *node-id* **例：** `RP/0/RSP0/CPU0:router#(config-redundancy-iccp-group) # mlacp node 1`	LACP システム プライオリティがこの ICCP グループで使用されるように設定します。 **（注）** *node-id* は、各 POA に固有である必要があります。
ステップ 6	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

マルチシャーシ Link Aggregation Control Protocol バンドルの設定

マルチシャーシ Link Aggregation Control Protocol（mLACP）バンドルを設定するには、次の作業を実行します。

手順の概要

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **mac-address** *mac-id*
4. **bundle wait-while** *milliseconds*
5. **lacp switchover suppress-flaps** *milliseconds*
6. **mlacp iccp-group** *group-id*
7. **mlacp port-priority** *priority*
8. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface Bundle-Ether** *bundle-id* **例：** `RP/0/RSP0/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。
ステップ 3	**mac-address** *mac-id* **例：** `RP/0/RSP0/CPU0:router#(config-if)# mac-address 1.1.1`	インターフェイスに MAC アドレスを設定します。 **（注）** 両方の POA に同じ MAC アドレスを設定することを強く推奨します。
ステップ 4	**bundle wait-while** *milliseconds* **例：** `RP/0/RSP0/CPU0:router#(config-if)# bundle wait-while 100`	このバンドル メンバに wait-while タイムアウトを設定します。
ステップ 5	**lacp switchover suppress-flaps** *milliseconds* **例：** `RP/0/RSP0/CPU0:router#(config-if)# lacp switchover suppress-flaps 300`	LACP のスイッチオーバー中のフラップを抑制する時間を設定します。 **（注）** *milliseconds* 引数に使用する値は、ローカル デバイス（および DHD）の wait-while タイマーよりも大きくすることを推奨します。

	コマンドまたはアクション	目的
ステップ 6	**mlacp iccp-group** *group-id* **例：** `RP/0/RSP0/CPU0:router#(config-if)# mlacp iccp-group 10`	このバンドルが動作する ICCP 冗長性グループを設定します。
ステップ 7	**mlacp port-priority** *priority* **例：** `RP/0/RSP0/CPU0:router#(config-if)# mlacp port-priority 10`	mLACP を実行するときの、このデバイスのすべてのメンバ リンクの開始プライオリティを設定します。 **（注）** 値が小さいほど、プライオリティが高くなります。動的優先権管理を使用している場合、スイッチオーバーが発生したときに、リンクのプライオリティが変わります。
ステップ 8	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

デュアルホーム接続デバイスの設定

デュアルホーム接続デバイス（DHD）を設定するには、次の作業を実行します。

（注） ASR 9000 シリーズ ルータを DHD として使用する場合は、**bundle maximum-active links** *links* コマンド（*links* は DHD を POA の 1 つに接続するリンクの数）を設定することを推奨します。

手順の概要

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **bundle wait-while** *milliseconds*
4. **lacp switchover suppress-flaps** *milliseconds*
5. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface Bundle-Ether** *bundle-id* **例：** `RP/0/RSP0/CPU0:router#(config-if)# 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。
ステップ 3	**bundle wait-while** *milliseconds* **例：** `RP/0/RSP0/CPU0:router#(config-if)# bundle wait-while 100`	このバンドル メンバに wait-while タイムアウトを設定します。

	コマンドまたはアクション	目的
ステップ 4	**lacp switchover suppress-flaps** *milliseconds* **例：** `RP/0/RSP0/CPU0:router#(config-if)# lacp switchover suppress-flaps 300`	LACP のスイッチオーバー中のフラップを抑制する時間を設定します。
ステップ 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

POA の 1 つのバンドルに追加されたメンバは *Active*、別の POA のメンバは *Standby* 状態になります。これは、いずれかの POA で **show bundle** コマンドを使用し、両方の POA で正しく設定されたメンバのメンバーシップ情報を表示することで確認できます。

```
RP/0/RSP0/CPU0:router# show bundle

Bundle-Ether1
  Status:                                  Up
  Local links <active/standby/configured>:   1 / 0 / 1
  Local bandwidth <effective/available>:     1000000 (1000000) kbps
  MAC address (source):                    0000.deaf.0000 (Configured)
  Minimum active links / bandwidth:        1 / 1 kbps
  Maximum active links:                    64
  Wait while timer:                        100 ms
  LACP:                                    Operational
    Flap suppression timer:                300 ms
  mLACP:                                   Operational
    ICCP Group:                            1
    Role:                                  Active
    Foreign links <active/configured>:     0 / 1
    Switchover type:                       Non-revertive
    Recovery delay:                        300 s
    Maximize threshold:                    Not configured
  IPv4 BFD:                                Not configured
```

```
Port                  Device           State        Port ID         B/W, kbps
--------------------  ---------------  -----------  --------------  ----------
Gi0/0/0/0             Local            Active       0x8001, 0x9001     1000000
    Link is Active
Gi0/0/0/0             5.4.3.2          Standby      0x8002, 0xa001     1000000
    Link is marked as Standby by mLACP peer
```

（注） アクティブ POA に切り替えるには、現在アクティブなルータで **mlacp switchover Bundle-Ether** コマンドを使用します。

アクセス バックアップ疑似回線の設定

VPLS アクセス疑似回線にバックアップ疑似回線を追加するには、次の作業を実行します。

手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *bridge-group name*
4. **bridge-domain** *bridge-domain name*
5. **neighbor** ***A.B.C.D ip-address*** **pw-id** *pseudowire-id*
6. **pw-class** {***class-class name***}
7. **backup neighbor** *A.B.C.D ip-address* **pw-id** *pseudowire-id*
8. **pw-class** {***class-class name***}
9. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**l2vpn** **例：** `RP/0/RSP0/CPU0:router(config)# l2vpn` `RP/0/RSP0/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/RSP0/CPU0:router(config-l2vpn)# bridge group csco` `RP/0/RSP0/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。

	コマンドまたはアクション	目的
ステップ 4	**bridge-domain** *bridge-domain-name* **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain abc` `RP/0/RSP0/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 5	**neighbor** *A.B.C.D* **pw-id** *pseudowire-id* **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# neighbor 10.2.2.2 pw-id 2000`	疑似回線セグメントを設定します。
ステップ 6	**pw-class** {*class-name*} **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pw)# pw-class class1`	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 7	**backup neighbor** *A.B.C.D* **pw-id** *pseudowire-id* **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pw)# backup neighbor 10.2.2.2 pw-id 2000`	VPLS アクセス疑似回線（PW）にバックアップ疑似回線を追加します。

	コマンドまたはアクション	目的
ステップ 8	**pw-class** {*class-name*} **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-pw)# pw-class class2`	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 9	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-mac)# end` または `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-mac)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

MC-LAG での一方向疑似回線冗長性の設定

冗長グループが設定されている場合に、一方向疑似回線冗長性の動作を許可するには、次の作業を実行します。

手順の概要

1. **configure**
2. **l2vpn**
3. **pw-class** {*class-name*}
4. **encapsulation mpls**
5. **redundancy one-way**
6. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**l2vpn** **例：** `RP/0/RSP0/CPU0:router(config)# l2vpn` `RP/0/RSP0/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-class** {*class-name*} **例：** `RP/0/RSP0/CPU0:router(config-l2vpn)# pw-class class1`	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 4	**encapsulation mpls** **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-pwc)#` `encapsulation mpls`	MPLS に疑似回線カプセル化を設定します。

	コマンドまたはアクション	目的
ステップ 5	**redundancy one-way** **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-pwc-mpls)# redundancy one-way`	一方向 PW 冗長性の動作を設定します。 **（注）** **redundancy one-way** コマンドは、冗長グループが設定されている場合にだけ有効です。
ステップ 6	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-mac)# end` または `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-mac)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

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。

手順の概要

1. **configure**
2. **l2vpn**
3. **pw-status**
4. **xconnect group** *group-name*
5. **p2p** *xconnect-name*
6. **interface** *type interface-path-id*
7. **neighbor** ***A.B.C.D ip-address*** **pw-id** *pseudowire-id*
8. **pw-class** {***class-class name***}
9. **backup neighbor** *A.B.C.D ip-address* **pw-id** *pseudowire-id*
10. **pw-class** {***class-class name***}
11. **end**
    または
    commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**l2vpn** **例：** `RP/0/RSP0/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/RSP0/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/RSP0/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。

	コマンドまたはアクション	目的
ステップ 5	**p2p** *xconnect-name* **例：** `RP/0/RSP0/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/RSP0/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/RSP0/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/RSP0/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/RSP0/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。

	コマンドまたはアクション	目的
ステップ 10	**pw-class** {*class-name*} **例：** `RP/0/RSP0/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/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw-backup)` `# end` または `RP/0/RSP0/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、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

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。

手順の概要

1. **configure**
2. **l2vpn**
3. **pw-status**
4. **bridge group** *bridge-group-name*
5. **bridge-domain** *bridge-domain-name*
6. **interface** ***type interface-path-id***
7. **vfi** *vfi-name*
8. **neighbor** ***A.B.C.D ip-address*** **pw-id** *pseudowire-id*
9. **pw-class** {***class-class name***}
10. **end**
    または
    commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**l2vpn** **例：** `RP/0/RSP0/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/RSP0/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/RSP0/CPU0:router(config-l2vpn)# bridge group csco` `RP/0/RSP0/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。

	コマンドまたはアクション	目的
ステップ 5	**bridge-domain** *bridge-domain-name* **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain abc` `RP/0/RSP0/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/RSP0/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/RSP0/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/RSP0/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。

	コマンドまたはアクション	目的
ステップ 9	**pw-class** {*class-name*} **例：** RP/0/RSP0/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/RSP0/CPU0:router(config-l2vpn-bg-bd-vfi-pw)# end または RP/0/RSP0/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、変更をコミットするように要求されます。 Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]: – **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

MGSCP の設定方法

MGSCP の設定の前提条件

MGSCP を設定する前に、次の前提条件を満たしていることを確認してください。

- Cisco ASR 9000 シリーズ ルータにインストールされたギガビット イーサネットまたは 10 ギガビット イーサネット ラインカードがあります。
- Service Control Engine（SCE）デバイスのクラスタの設定方法を理解し、ネットワークの目的の要件（MGSCP サポートの次の要件を含む）に応じてこれを設定すること。

- Cisco ASR 9000 シリーズ ルータに SCE デバイスを接続する場合は、各 SCE デバイスに、次のようなCisco ASR 9000 シリーズ ルータの 2 つのバンドル インターフェイスに接続する 2 つの個別の物理リンクがあることを確認してください。

  - 各 SCE デバイスからの 1 つのリンクはネットワークのアクセス（または加入者）側にルーティングされているバンドル インターフェイス上のリンクに接続されます。

  - 各 SCE デバイスからのもう 1 つのリンクはネットワークのコア側にルーティングされている別のバンドル インターフェイス上のリンクに接続されます。

- SCE デバイスに、リンク障害リフレクション用の SCE ポートを設定（**link failure-reflection** コマンドを使用）し、SCE の片方のリンクがダウンした場合に、他方のリンクが自動的にシャット ダウンされるようにしていること。詳細については、ご使用のデバイスの『Cisco SCE software configuration guide』の「Configuring the Connection」の章および次の URL のリリースを参照してください。

  http://www.cisco.com/en/US/products/ps6134/products_installation_and_configuration_guides_list.html

- Cisco ASR 9000 シリーズ ルータのバンドルの設定について、次の情報を特定します。
  - サポートするアクティブ リンクの最大数。
  - 保護（バックアップ）リンクとするバンドル リンク。最大 4 つの保護リンクを設定できます。
- 接続された SCE のステートフルネスを維持するため、すべての加入者フローは同じ SCE を通過します。したがって、MGSCP を設定する前に、加入者トラフィックをルータがどのようにリダイレクトするかを決定し、その SCE に接続されている適切なバンドル インターフェイスを加入者トラフィックが通過するようにしておく必要があります。

  次のいずれかの方法を使用できます。

  - ACL ベースの転送（ABF）：ネクスト ホップの IP アドレスだけをサポートします。設定は複雑です。ABF の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Configuration Guide*』の「Implementing Access Lists and Prefix Lists」の章を参照してください。
  - 仮想ルーティングおよびフォワーディング（VRF）：推奨。OSPF および BGP によるスタティックまたはダイナミック ルーティングを使用してルーティングされる、アクセス バンドルおよびネットワーク バンドルの VRF インスタンスを使用します。

MGSCP の設定に関する制約事項

MGSCP を設定する前に、次の制限事項を確認してください。

- バンドルには、最大 4 個の保護リンクを設定できます。
- IPv6 アドレス指定はサポートされていません。IPv4 アドレッシングを使用する必要があります。
- MPLS はサポートされていません。
- 1 つのバンドルには、最大 8 本のメンバー リンクを設定できます。

加入者側のアクセス バンドルの設定

ネットワークの加入者側に面するアクセス バンドルの設定は、コア バンドルの設定に似ています。次の注意事項を参照してください。

- VRF を使用して加入者トラフィックを同じ SCE にルーティングする場合（推奨）、加入者側には別の VRF が使用されます。

- リンク順序シグナリングによりロード バランシング テーブルのリンク順序番号（LON）の LACP プロセスをイネーブルにする必要があります。
- バンドルのロード バランシングは、送信元 IP アドレスに基づいて設定されます。
- アクティブ リンクの最大数はコア バンドルのアクティブ リンクの最大数と一致するように設定する必要があります。

手順の概要

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **vrf** *vrf-name*
4. **ipv4 address** *ipv4-address mask*
5. **lacp cisco enable link-order signaled**
6. **bundle load-balancing hash src-ip**
7. **bundle maximum-active links** *links* [**hot-standby**]
8. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface Bundle-Ether** *bundle-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether 100`	ネットワークの加入者側のイーサネット バンドル インターフェイスを指定または作成（ここで、*bundle-id* は 1 ～ 65535 の数値）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 3	**vrf** *vrf-name* **例：** `RP/0/RSP0/CPU0:router(config-if)# vrf access`	（任意：推奨）このイーサネット バンドルが参加しているネットワークの加入者側の VRF インスタンスを指定します。
ステップ 4	**ipv4 address** *ipv4-address mask* **例：** `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 10.1.1.1 255.255.255.0`	このインターフェイスの指定した VRF の一部である IPv4 アドレスおよびマスクを指定します。ここで、*ipv4-address* は 32 ビット IP アドレスであり、対応するマスクをドット付き 10 進表記形式（A.B.C.D）で持ちます。 **（注）** このコマンドは、IP アドレスが VRF インスタンスの一部であることを確認するために、**vrf** コマンドの後ろに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
ステップ 5	**lacp cisco enable link-order signaled** **例：** `RP/0/RSP0/CPU0:router(config-if)# lacp cisco enable link-order signaled`	このバンドルで処理する LACP プロセスの一部としてリンクの順序番号を含む Cisco TLV の使用をイネーブルにします。
ステップ 6	**bundle load-balancing hash src-ip** **例：** `RP/0/RSP0/CPU0:router(config-if)# bundle load-balancing hash src-ip`	加入者のバンドル インターフェイスのロード バランシングに使用されるハッシュは、送信元 IP アドレスに基づいていることを指定します。
ステップ 7	**bundle maximum-active links** *links* [**hot-standby**] **例：** `RP/0/RSP0/CPU0:router(config-if)# bundle maximum-active links 2`	バンドルで許可されるアクティブ リンクの最大数を指定し、をロード バランシング テーブルで使用されるリンク順序番号の上限を設定します。 **（注）** MGSCP をサポートするには、このコマンドもコア バンドルと同じ値に設定する必要があります。
ステップ 8	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router (config-bfd-if)# end` または `RP/0/RSP0/CPU0:router(config-bfd-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

コア側のネットワーク バンドルの設定

ネットワークのコア側に面するバンドルの設定は、アクセス バンドルの設定に似ています。次の注意事項を参照してください。

- VRF を使用して加入者トラフィックを同じ SCE にルーティングする場合（推奨）、コア側には別の VRF が使用されます。
- リンク順序シグナリングによりロード バランシング テーブルの LON の LACP プロセスをイネーブルにする必要があります。
- バンドルのロード バランシングは、宛先 IP アドレスに基づいて設定されます。

- アクティブ リンクの最大数はアクセス バンドルのアクティブ リンクの最大数と一致するように設定する必要があります。

手順の概要

1. **configure**
2. **interface Bundle-Ether** *bundle-id*
3. **vrf** *vrf-name*
4. **ipv4 address** *ipv4-address mask*
5. **lacp cisco enable link-order signaled**
6. **bundle load-balancing hash dst-ip**
7. **bundle maximum-active links** *links* [**hot-standby**]
8. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface Bundle-Ether** *bundle-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface Bundle-Ether 100`	ネットワークの加入者側のイーサネット バンドル インターフェイスを指定または作成（ここで、*bundle-id* は 1 ～ 65535 の数値）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 3	**vrf** *vrf-name* **例：** `RP/0/RSP0/CPU0:router(config-if)# vrf access`	（任意：推奨）このイーサネット バンドルが参加しているネットワークのコア側の VRF インスタンスを指定します。
ステップ 4	**ipv4 address** *ipv4-address mask* **例：** `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 10.1.1.1 255.255.255.0`	このインターフェイスの指定した VRF の一部である IPv4 アドレスおよびマスクを指定します。ここで、*ipv4-address* は 32 ビット IP アドレスであり、対応するマスクをドット付き 10 進表記形式（A.B.C.D）で持ちます。 **（注）** このコマンドは、IP アドレスが VRF インスタンスの一部であることを確認するために、**vrf** コマンドの後ろに指定する必要があります。
ステップ 5	**lacp cisco enable link-order signaled** **例：** `RP/0/RSP0/CPU0:router(config-if)# lacp cisco enable link-order signaled`	このバンドルで処理する LACP プロセスの一部としてリンクの順序番号を含む Cisco TLV の使用をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 6	**bundle load-balancing hash dst-ip** **例：** `RP/0/RSP0/CPU0:router(config-if)# bundle load-balancing hash dst-ip`	加入者のバンドル インターフェイスのロード バランシングに使用されるハッシュは、宛先 IP アドレスに基づいていることを指定します。
ステップ 7	**bundle maximum-active links** *links* [**hot-standby**] **例：** `RP/0/RSP0/CPU0:router(config-if)# bundle maximum-active links 2`	バンドルで許可されるアクティブ リンクの最大数を指定し、をロード バランシング テーブルで使用されるリンク順序番号の上限を設定します。 **（注）** MGSCP をサポートするには、このコマンドもアクセス バンドルと同じ値に設定する必要があります。
ステップ 8	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router (config-bfd-if)# end` または `RP/0/RSP0/CPU0:router(config-bfd-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

バンドル メンバ インターフェイスの設定

アクセス バンドルおよびコア バンドルが設定されている場合、バンドル インターフェイスはこれらのバンドル上のアクティブ リンクおよび保護リンクとして設定する必要があります。次の注意事項を参照してください。

- リンクは **bundle id** コマンドを使用し、対応するバンドル インターフェイスの ID を指定してバンドル メンバとなります。MGSCP では、2 種類の異なるバンドルが存在し、1 つはアクセス側トラフィック用、もう 1 つはコア側トラフィック用です。これらの各バンドルには SCE の両側に接続するリンクがあります。慎重に適切なバンドルにインターフェイスをマッピングしてください。
- リンクは **mode active** を使用してバンドル上に設定する必要があるため、MGSCP には LACP が必要です。
- アクティブ リンクおよびバックアップ（保護）リンクは **bundle port-priority** コマンドで設定します。

- 機能（アクティブ）リンクを設定するには、プライオリティ 1 を使用します。設定できるアクティブ リンクの最大数はバンドルの **bundle maximum-active links** コマンドの値によって決まります。
- 1 以外の任意のプライオリティで、リンクを保護リンクに指定します。最大 4 つの保護リンクを設定できます。

手順の概要

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
3. **bundle id** *bundle-id* **mode active**
4. **bundle port-priority** *priority*
5. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/0`	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 インターフェイスを指定または作成し、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。ここで、*interface-path-id* は *rack*/*slot*/*module*/*port* 表記を使用したインターフェイスの物理的な場所です。
ステップ 3	**bundle id** *bundle-id* **mode active** **例：** `RP/0/RSP0/CPU0:router(config-if)# bundle id 100 mode active`	インターフェイスを指定したバンドルのメンバとして追加し、インターフェイス上で LACP をアクティブ モードで実行して、MGSCP の LACP パケットを交換します。

	コマンドまたはアクション	目的
ステップ 4	**bundle port-priority** *priority* **例：** `RP/0/RSP0/CPU0:router(config-if)# bundle port-priority 1`	LACP プライオリティをインターフェイスに指定して、バンドル インターフェイスがアクティブであるか、または MGSCP の保護であるかを特定します。 • 値 1：リンクがアクティブ インターフェイスであることを指定します。 • 1 以外の値：リンクが保護インターフェイスであることを指定します。 デフォルトは 32768 です。
ステップ 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router (config-bfd-if)# end` または `RP/0/RSP0/CPU0:router(config-bfd-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

トラフィックをバンドルにルーティングする VRF の設定

ステートフルネスのために、すべての加入者トラフィックが同じ SCE に残るようにするためには、VRF は、すべての加入者トラフィックをバンドルにルーティングする方法として推奨されます。MGSCP に VRF を設定するには、次のいずれかの作業を実行します。

- 「スタティック ルーティングを使用した VRF の設定」（P.268）
- 「ダイナミック ルーティングを使用した VRF の設定」（P.269）

スタティック ルーティングを使用した VRF の設定

次のステップは、スタティック ルーティングを使用して VRF を設定するために必要なタスクの要約です。

1. 2 つの VRF をグローバル コンフィギュレーションで設定します（ネットワークのアクセス側に 1 つ、コア側に 1 つ）。IPv4 ユニキャスト アドレス ファミリを指定してください。

2. IPv4 アドレスをバンドルの各インターフェイスとして設定し、ネットワークのアクセス側およびコア側にグローバル コンフィギュレーションで設定した、対応する VRF にこれらのアドレスを関連付けます。
3. IPv4 アドレスをギガビット イーサネット物理インターフェイスとして設定し、ネットワークのアクセス側およびコア側にグローバル コンフィギュレーションで設定した、対応する VRF にこれらのアドレスを関連付けます。
4. **router static** コマンドを使用してスタティック ルーティングを設定し、アクセス VRF およびコア VRF を対応するバンドル インターフェイスにマッピングします。

設定例については、「例：スタティック ルーティングを使用した VRF の設定」（P.277）を参照してください。

ダイナミック ルーティングを使用した VRF の設定

MGSCP の VRF では、OSPF と BGP の両方のルーティング プロトコルがサポートされます。グローバル コンフィギュレーションと、バンドルおよび物理インターフェイスの VRF の全般設定は、スタティック ルーティングと同じです。

次のステップは、OSPF ルーティングを使用して VRF を設定するために必要なタスクの要約です。

1. 2 つの VRF をグローバル コンフィギュレーションで設定します（ネットワークのアクセス側に 1 つ、コア側に 1 つ）。IPv4 ユニキャスト アドレス ファミリを指定してください。
2. IPv4 アドレスをバンドルの各インターフェイスとして設定し、ネットワークのアクセス側およびコア側にグローバル コンフィギュレーションで設定した、対応する VRF にこれらのアドレスを関連付けます。
3. IPv4 アドレスをギガビット イーサネット物理インターフェイスとして設定し、ネットワークのアクセス側およびコア側にグローバル コンフィギュレーションで設定した、対応する VRF にこれらのアドレスを関連付けます。
4. **router ospf** コマンドを使用して OSPF などのダイナミック ルーティング プロトコルを設定することで VRF を定義し、バンドルおよび物理インターフェイスを OSPF 領域に関連付けます。

設定例については、「例：OSPF ルーティングを使用した VRF の設定」（P.278）を参照してください

リンク バンドルの設定例

ここでは、次の例を示します。

例：イーサネット リンク バンドルの設定

次に、2 つのポートを結合して、LACP が動作する EtherChannel バンドルを構成する例を示します。

```
RP/0/RSP0/CPU0:Router# config
RP/0/RSP0/CPU0:Router(config)# interface Bundle-Ether 3
RP/0/RSP0/CPU0:Router(config-if)# ipv4 address 1.2.3.4/24
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active links 1
RP/0/RSP0/CPU0:Router(config-if)# exit
RP/0/RSP0/CPU0:Router(config)# interface TenGigE 0/3/0/0
RP/0/RSP0/CPU0:Router(config-if)# bundle id 3 mode active
RP/0/RSP0/CPU0:Router(config-if)# no shutdown
RP/0/RSP0/CPU0:Router(config)# exit
RP/0/RSP0/CPU0:Router(config)# interface TenGigE 0/3/0/1
RP/0/RSP0/CPU0:Router(config-if)# bundle id 3 mode active
RP/0/RSP0/CPU0:Router(config-if)# no shutdown
RP/0/RSP0/CPU0:Router(config-if)# exit
```

例：VLAN リンク バンドルの設定

次に、イーサネット バンドル上で 2 つの VLAN を作成しアップ状態にする例を示します。

```
RP/0/RSP0/CPU0:Router# config
RP/0/RSP0/CPU0:Router(config)# interface Bundle-Ether 1
RP/0/RSP0/CPU0:Router(config-if)# ipv4 address 1.2.3.4/24
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active links 1
RP/0/RSP0/CPU0:Router(config-if)# exit
RP/0/RSP0/CPU0:Router(config)# interface Bundle-Ether 1.1
RP/0/RSP0/CPU0:Router(config-subif)# dot1q vlan 10
RP/0/RSP0/CPU0:Router(config-subif)# ip addr 10.2.3.4/24
RP/0/RSP0/CPU0:Router(config-subif)# no shutdown
RP/0/RSP0/CPU0:Router(config-subif)# exit
RP/0/RSP0/CPU0:Router(config)# interface Bundle-Ether 1.2
RP/0/RSP0/CPU0:Router(config-subif)# dot1q vlan 20
RP/0/RSP0/CPU0:Router(config-subif)# ip addr20.2.3.4/24
RP/0/RSP0/CPU0:Router(config-subifif)# no shutdown
RP/0/RSP0/CPU0:Router(config-subifif)# exit
RP/0/RSP0/CPU0:Router(config)# interface gig 0/1/5/7
RP/0/RSP0/CPU0:Router(config-if)# bundle-id 1 mode act
RP/0/RSP0/CPU0:Router(config-if)# commit
RP/0/RSP0/CPU0:Router(config-if)# exit
```

例：POS リンク バンドルの設定

次の例では、2 つのポートを結合して Packet-over-SONET（POS）リンク バンドルを形成する方法を示します。

```
RP/0/RSP0/CPU0:Router# config
RP/0/RSP0/CPU0:Router(config)# interface Bundle-POS 5
RP/0/RSP0/CPU0:Router(config-if)# ipv4 address 1.2.3.4/24
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RSP0/CPU0:Router(config-if)# bundle minimum-active bandwidth 620000
RP/0/RSP0/CPU0:Router(config-if)# exit
RP/0/RSP0/CPU0:Router(config)# interface POS 0/0/1/1
RP/0/RSP0/CPU0:Router(config-if)# bundle id 5
RP/0/RSP0/CPU0:Router(config-if)# no shutdown
RP/0/RSP0/CPU0:Router(config-if)# exit
```

例：イーサネット リンク バンドルでの EFP ロード バランシングの設定

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

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

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

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

例：マルチシャーシ リンク集約の設定

次に、POA を設定する例を示します。

アクティブ POA

```
interface Bundle-Ether10
   mlacp iccp-group 1
   mlacp port-priority 10
```

スタンバイ POA

```
interface Bundle-Ether10
   mlacp iccp-group 1
   mlacp port-priority 20
```

次に、ICCP を設定する例を示します。

```
redundancy iccp group
   member neighbor 1.2.3.4
   backbone interface GigabitEthernet 0/0/0/0
   isolation recovery-delay 30
```

次に、mLACP を設定する例を示します。

```
configure
 redundancy iccp group 100
   mlacp system mac 1.1.1
   mlacp system priority 10
   mlacp node 1
     interface Bundle-Ether 3
        mac-address 1.1.1
        bundle wait-while 100
        lacp switchover suppress-flaps 300
        mlacp iccp-group 100
```

次に、スイッチオーバーの例を示します。

```
RP/0/0/CPU0:router# show bundle

Bundle-Ether1
  Status:                                  Up
  Local links <active/standby/configured>:   1 / 0 / 1
  Local bandwidth <effective/available>:     1000000 (1000000) kbps
```

```
  MAC address (source):                       0000.deaf.0000 (Configured)
  Minimum active links / bandwidth:           1 / 1 kbps
  Maximum active links:                       64
  Wait while timer:                           100 ms
  LACP:                                       Operational
    Flap suppression timer:                   300 ms
  mLACP:                                      Operational
    ICCP Group:                               1
    Role:                                     Active
    Foreign links <active/configured>:        0 / 1
    Switchover type:                          Non-revertive
    Recovery delay:                           300 s
    Maximize threshold:                       Not configured
  IPv4 BFD:                                   Not configured

  Port                  Device           State        Port ID         B/W, kbps
  --------------------  ---------------  -----------  --------------  ----------
  Gi0/0/0/0             Local            Active       0x8001, 0x9001     1000000
      Link is Active
  Gi0/0/0/0             5.4.3.2          Standby      0x8002, 0xa001     1000000
      Link is marked as Standby by mLACP peer


RP/0/0/CPU0:router#mlacp switchover Bundle-Ether 1

This will trigger the peer device (Node 5.4.3.2 in IG 1) to become active for
Bundle-Ether1.This may result in packet loss on the specified bundle.

Proceed with switch over?[confirm]

RP/0/0/CPU0:Jan 31 23:46:44.666 : BM-DISTRIB[282]: %L2-BM-5-MLACP_BUNDLE_ACTIVE : This
device is no longer the active device for Bundle-Ether1
RP/0/0/CPU0:Jan 31 23:46:44.668 : BM-DISTRIB[282]: %L2-BM-6-ACTIVE :
GigabitEthernet0/0/0/0 is no longer Active as part of Bundle-Ether1 (Not enough links
available to meet minimum-active threshold)

RP/0/0/CPU0:router#show bundle
Mon Jun  7 06:04:17.778 PDT

Bundle-Ether1
  Status:                                     mLACP hot standby
  Local links <active/standby/configured>:    0 / 1 / 1
  Local bandwidth <effective/available>:      0 (0) kbps
  MAC address (source):                       0000.deaf.0000 (Configured)
  Minimum active links / bandwidth:           1 / 1 kbps
  Maximum active links:                       64
  Wait while timer:                           100 ms
  LACP:                                       Operational
    Flap suppression timer:                   300 ms
  mLACP:                                      Operational
    ICCP Group:                               1
    Role:                                     Standby
    Foreign links <active/configured>:        1 / 1
    Switchover type:                          Non-revertive
    Recovery delay:                           300 s
    Maximize threshold:                       Not configured
  IPv4 BFD:                                   Not configured

  Port                  Device           State        Port ID         B/W, kbps
  --------------------  ---------------  -----------  --------------  ----------
  Gi0/0/0/0             Local            Standby      0x8003, 0x9001     1000000
      mLACP peer is active
  Gi0/0/0/0             5.4.3.2          Active       0x8002, 0xa001     1000000
      Link is Active
```

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

次に、VPLS アクセス疑似回線にバックアップ疑似回線を追加する例を示します。

```
l2vpn bridge group bg1
  bridge-domain bd1
   neighbor 101.101.101.101 pw-id 5000
       pw-class class1
           backup neighbor 102.102.102.102 pw-id 3000
               pw-class class1
           !
        !
    !
!
```

次に、冗長グループが設定されている場合に、一方向疑似回線冗長性の動作を設定する例を示します。

```
l2vpn pw-class class_mpls
   encapsulation mpls
       redundancy one-way
    !
!
```

次に、全体的な MC-LAG 設定の例を示します。

トロポジの場合：

```
DHD                      POA 1                      POA 2

Gi0/0/0/0 --------------- Gi0/0/0/0
Gi0/0/0/1 --------------- Gi0/0/0/1
Gi0/0/0/2
Gi0/0/0/3 ------------------------------------------ Gi0/0/0/0
Gi0/0/0/4 ------------------------------------------ Gi0/0/0/1
                          Gi0/0/0/2                  Gi0/0/0/2
                          Gi0/0/0/3 --------------- Gi0/0/0/3
                          Gi0/0/0/4 --------------- Gi0/0/0/4
```

POA 1 の場合：

```
redundancy
 iccp
  group 1
   mlacp node 1
   mlacp system mac 000d.000e.000f
   mlacp system priority 1
   member
    neighbor 5.4.3.2
   !
!
!
!
interface Bundle-Ether1
 lacp switchover suppress-flaps 300
 mlacp iccp-group 1
 mac-address 0.deaf.0
 bundle wait-while 100
!
interface Loopback0
 ipv4 address 5.4.3.1 255.255.255.255
!
interface GigabitEthernet0/0/0/0
 description Connected to DHD Gi0/0/0/0
```

```
 bundle id 1 mode active
 lacp period short
 no shutdown
!
interface GigabitEthernet0/0/0/3
 description Connected to POA2 Gi0/0/0/3
 ipv4 address 1.2.3.1 255.255.255.0
 proxy-arp
 no shutdown
!
router static
 address-family ipv4 unicast
  5.4.3.2/32 1.2.3.2
 !
!
mpls ldp
 router-id 5.4.3.1
 discovery targeted-hello accept
 log
  neighbor
 !
interface GigabitEthernet0/0/0/3
 !
!
```

POA 2 の場合：

```
redundancy
 iccp
  group 1
   mlacp node 2
   mlacp system mac 000d.000e.000f
   mlacp system priority 1
   member
    neighbor 5.4.3.1
   !
!
!
!
interface Bundle-Ether1
 lacp switchover suppress-flaps 300
 mlacp iccp-group 1
 mac-address 0.deaf.0
 bundle wait-while 100
!
interface Loopback0
 ipv4 address 5.4.3.2 255.255.255.255
!
interface GigabitEthernet0/0/0/0
 description Connected to DHD Gi0/0/0/3
 bundle id 1 mode active
 lacp period short
 no shutdown
!
interface GigabitEthernet0/0/0/3
 description Connected to POA1 Gi0/0/0/3
 ipv4 address 1.2.3.2 255.255.255.0
 proxy-arp
 no shutdown
!
router static
 address-family ipv4 unicast
  5.4.3.1/32 1.2.3.1
 !
```

```
!
mpls ldp
 router-id 5.4.3.2
 discovery targeted-hello accept
 log
  neighbor
 !
interface GigabitEthernet0/0/0/3
 !
!
```

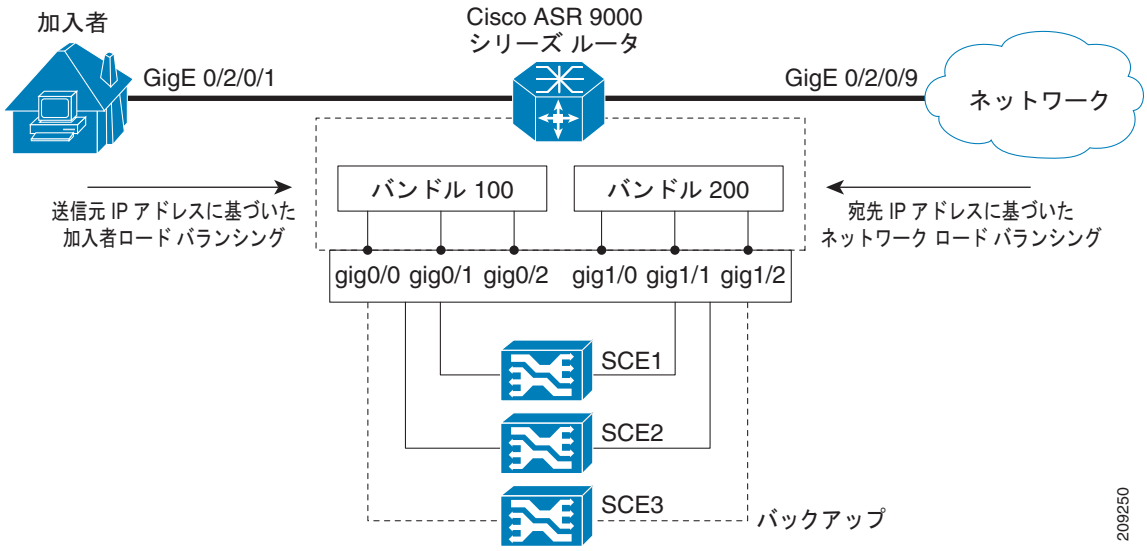
DHD の場合：

```
interface Bundle-Ether1
 lacp switchover suppress-flaps 300
 bundle wait-while 100
!
interface GigabitEthernet0/0/0/0
 description Connected to POA1 Gi0/0/0/0
 bundle id 1 mode active
 lacp period short
 no shutdown
!
interface GigabitEthernet0/0/0/3
 description Connected to POA2 Gi0/0/0/0
 bundle id 1 mode active
 lacp period short
 no shutdown
!
```

MGSCP の設定例

図 16 に、設定例として使用する SCE デバイスのクラスタのディスパッチャとして、1 台の Cisco ASR 9000 シリーズ ルータを使用したネットワーク例について説明します。

図 16　SCE クラスタのディスパッチャとしての Cisco ASR 9000 シリーズ ルータ



ここでは、次の設定例について説明します。

例：バンドル インターフェイスおよびメンバ リンクの設定

次に、図 16 に示した Cisco ASR 9000 シリーズ ルータ上に 2 つのバンドルを設定する例を示します。各バンドルは、最大 2 個のアクティブ リンクをサポートし（両方のバンドルの設定は一致させる必要があります）、1 つはバックアップ保護リンクとします。

イーサネット バンドル 100 のバンドル インターフェイス メンバは、送信元 IP アドレスに基づいたロード バランシングを使用して、ネットワークの加入者側への SCE デバイス リンクに接続します。イーサネット バンドル 200 のバンドル インターフェイス メンバは、宛先 IP アドレスに基づいたロード バランシングを使用して、ネットワークのコア側への SCE デバイス リンクに接続します。

加入者側アクセス バンドルの設定

```
interface Bundle-Ether 100
 description Subscriber-facing end
 vrf access
 ipv4 address 10.10.1.2 255.255.255.0
 lacp cisco enable link-order signaled
 bundle load-balancing hash src-ip
 bundle maximum-active links 2
!
interface GigabitEthernet 0/0/0/0
 description to SCE1
 bundle id 100 mode active
 bundle port-priority 1
!
interface GigabitEthernet 0/0/0/1
 description to SCE2
 bundle id 100 mode active
 bundle port-priority 1
!
interface GigabitEthernet 0/0/0/3
 description to SCE3 (backup)
 bundle id 100 mode active
```

コア側バンドル設定

```
interface Bundle-Ether 200
 description Core-facing end
 vrf core
 ipv4 address 10.20.1.2 255.255.255.0
 lacp cisco enable link-order signaled
 bundle load-balancing hash dst-ip
 bundle maximum active links 2
!
interface GigabitEthernet 0/0/1/0
 description from SCE1
 bundle id 200 mode active
 bundle port-priority 1
!
interface GigabitEthernet 0/0/1/1
 description from SCE2
 bundle id 200 mode active
 bundle port-priority 1
!
interface GigabitEthernet 0/0/1/2
```

```
 description from SCE3 (standby)
 bundle id 200 mode active
```

例：トラフィックをバンドルにルーティングする VRF の設定

同じ加入者との間のトラフィックが確実に SCE の同じポートを通過するようにするには、VRF が推奨されます。MGSCP には、2 つの VRF を設定する必要があります（1 つはアクセス トラフィック用、もう 1 つはコア トラフィック用）。

このセクションの例では、VRF のバンドル インターフェイスで、スタティックまたはダイナミック（OSPF）ルーティングのいずれかの VRF を使用してルーティングできる、2 つの方法を示します。

- 「例：スタティック ルーティングを使用した VRF の設定」（P.277）
- 「例：OSPF ルーティングを使用した VRF の設定」（P.278）

例：スタティック ルーティングを使用した VRF の設定

次の設定例では、VRF は IPv4 を使用してネットワークのコア側とアクセス側に確立されます。そこから、各側のバンドル インターフェイス アドレスは、それぞれ VRF の一部として、また 2 つの物理インターフェイスとして設定されます。設定の最後では、バンドル インターフェイスを使用して各 VRF にスタティック ルートを設定しています。

VRF グローバル コンフィギュレーション

```
vrf core
 address-family ipv4 unicast
  import route-target
   1:1
  !
export route-target
   1:1
  !
vrf access
 address-family ipv4 unicast
  import route-target
   1:1
  !
export route-target
   1:1
  !
```

バンドル インターフェイスの VRF 設定

```
interface Bundle-Ether100
 vrf access
 ipv4 address 10.10.1.2 255.255.255.0
!
interface Bundle-Ether200
 vrf core
 ipv4 address 10.20.1.2 255.255.255.0
```

物理インターフェイスの VRF 設定

```
interface GigabitEthernet0/2/0/1
 vrf access
 ipv4 address 10.10.1.4 255.255.255.0

interface GigabitEthernet0/2/0/9
 vrf core
```

```
 ipv4 address 10.20.1.4 255.255.255.0
 negotiation auto
```

VRF のバンドル インターフェイスへのスタティック ルーティング設定

```
router static
 vrf core
  address-family ipv4 unicast
   0.0.0.0/0 Bundle-Ether200
  !
!
vrf access
  address-family ipv4 unicast
   0.0.0.0/0 Bundle-Ether100
  !
!
```

例：OSPF ルーティングを使用した VRF の設定

次の設定例では、VRF は IPv4 を使用してネットワークのコア側とアクセス側に確立されます。そこから、OSPF ルーティング インスタンスおよび VRF を含める領域を設定し、バンドル インターフェイスと物理インターフェイスを関連付けます。

VRF グローバル コンフィギュレーション

```
vrf core
 address-family ipv4 unicast
  import route-target
   1:1
  export route-target
   1:1

vrf access
 address-family ipv4 unicast
  import route-target
   1:1
  export route-target
   1:1
```

物理インターフェイスの VRF 設定

```
interface GigabitEthernet0/2/0/1
 vrf access
 ipv4 address 10.10.1.4 255.255.255.0

_
interface GigabitEthernet0/2/0/9
 vrf core
 ipv4 address 10.20.1.4 255.255.255.0
```

VRF の OSPF ルーティング設定とバンドルと物理インターフェイス

```
router ospf 100
 vrf core
  router-id 10.20.1.2
  area 0
   interface Bundle-Ether200
   interface GigabitEthernet0/2/0/9

 vrf access
  router-id 10.10.1.2
  area 0
   interface Bundle-Ether100
   interface GigabitEthernet0/2/0/1
```

例 : ABF を使用しバンドルにトラフィックをルーティングする MGSCP の設定

次に、アクセス リストを使用してバンドルにトラフィックを転送するトラフィックのルーティングの例を示します。

```
ipv4 access-list inbound
!
!Set the nexthop address to be a virtual IP address on the same network
!as the access bundle.
!
10 permit ipv4 any any nexthop 10.10.1.5
!
ipv4 access-list outbound
!
!Set the nexthop address to be a virtual IP address on the same network
!as the core bundle.
!
10 permit ipv4 any any nexthop 10.20.1.5
!
!Configure static ARP for the virtual IP addresses
!
arp vrf default 10.10.1.5 0024.98eb.bf8a ARPA
arp vrf default 10.20.1.5 0024.98eb.bf8b ARPA

interface Bundle-Ether100
 ipv4 address 10.10.1.2 255.255.255.0
!
interface Bundle-Ether200
 ipv4 address 10.20.1.2 255.255.255.0
!
interface GigabitEthernet0/2/0/1
 ipv4 address 10.10.1.3 255.255.255.0
 ipv4 access-group inbound
!
interface GigabitEthernet0/2/0/9
ipv4 address 10.20.1.3 255.255.255.0
 ipv4 access-group outbound
!
```

その他の関連資料

ここでは、リンク バンドルの設定に関連する参考資料を示します。

関連資料

関連項目	参照先
Cisco ASR 9000 シリーズ ルータ マスター コマンド リファレンス	*『Cisco ASR 9000 Series Router Master Commands List』*
Cisco ASR 9000 シリーズ ルータ インターフェイス コンフィギュレーション コマンド	*『Cisco ASR 9000 Series Router Interface and Hardware Component Command Reference』*
Cisco IOS XR ソフトウェアを使用する Cisco ASR 9000 シリーズ ルータの初期システム ブートアップと設定に関する情報。	*『Cisco ASR 9000 Series Router Getting Started Guide』*
ユーザ グループとタスク ID に関する情報	*『Cisco ASR 9000 Series Router Interface and Hardware Component Command Reference』*

標準

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

MIB

MIB	MIB のリンク
このモジュールに適用できる MIB はありません。	Cisco IOS XR ソフトウェアを使用して選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	—

シスコのテクニカル サポート

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

その他の関連資料

Cisco ASR 9000 シリーズ ルータでのトラフィック ミラーリングの実装

このモジュールでは、Cisco ASR 9000 シリーズ ルータのトラフィック ミラーリング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。

Cisco ASR 9000 シリーズ ルータのトラフィック ミラーリング設定の機能履歴

リリース	変更内容
リリース 3.9.1	この機能は、Cisco ASR 9000 シリーズ ルータで導入されました。
リリース 4.0.1	次のトラフィック ミラーリング機能が追加されました。 • 疑似回線上のトラフィック ミラーリング • フローまたは ACL ベースのトラフィック ミラーリング • レイヤ 3 インターフェイスのサポート • 部分的パケット ミラーリング

内容

トラフィック ミラーリングに関する制約事項

最大 8 個のモニタリング セッション、および 800 個の送信元ポートがサポートされます。

単一のモニタリング セッションで 800 個の送信元ポートを設定するか、または最大 8 個のモニタリング セッションで合計 800 個の送信元ポートを設定できます。

次の形式のトラフィック ミラーリングはサポートされていません。

- GRE トンネル（Cisco IOS ソフトウェアではカプセル化リモート スイッチド ポート アナライザ（ER-SPAN）とも呼ばれる）へのミラーリング トラフィック。
- フルブリッジ ドメインからのトラフィックのミラーリング（Cisco IOS ソフトウェアでは VLAN ベース SPAN とも呼ばれる）。

トラフィック ミラーリングのパフォーマンスへの影響

ミラーリングするのは通過トラフィック全体の 15% 以下とすることを推奨します。Cisco ASR 9000 イーサネット ラインカードで、10 ギガビット イーサネット インターフェイスまたはバンドル インターフェイスが使用される場合は、入力と出力それぞれのトラフィックのうち、ミラーリングできるデータは 1.5G までという制限があります。この制限は、Cisco ASR 9000 Enhanced イーサネット ラインカードでは適用されません。

トラフィック ミラーリングに関する情報

次の各項で、トラフィック ミラーリングに関する情報を示します。

- 「トラフィック ミラーリングとは」（P.284）
- 「トラフィック ミラーリング用語」（P.286）
- 「サポートされるトラフィック ミラーリングのタイプ」（P.288）

トラフィック ミラーリングとは

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、イーサネット インターフェイスのセットに入ってくる、または出ていくレイヤ 2 またはレイヤ 3 のネットワーク 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ネットワーク アナライザに渡して、分析することができます。

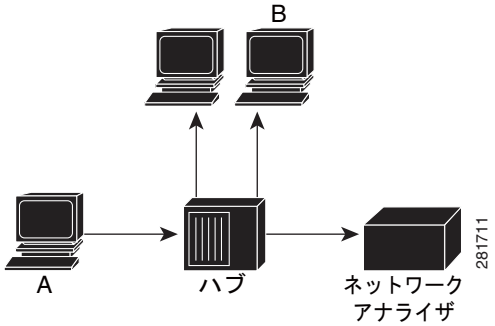
トラフィック ミラーリングによって、1 つ以上のレイヤ 3 インターフェイスまたはレイヤ 2 インターフェイスまたはサブインターフェイス（レイヤ 2 リンク バンドル インターフェイスまたはサブインターフェイスを含む）からトラフィックがコピーされ、このコピーされたトラフィックが 1 つ以上の宛先に送信されます。これで、ネットワーク アナライザなどのモニタリング デバイスで分析できるようになります。トラフィック ミラーリングは、送信元インターフェイスやサブインターフェイス上のトラフィックのスイッチングに影響を与えることはありません。ミラーリングされたトラフィックは、宛先であるインターフェイスまたはサブインターフェイスに送信できます。

トラフィック ミラーリングがスイッチにおいて導入されたのは、スイッチとハブの間には根本的な相違があるからです。ハブが 1 つのポートでパケットを受信すると、そのハブはパケットを受信したポート以外のすべてのポートからそのパケットのコピーを送信します。スイッチの場合は、スイッチ起動後

にレイヤ 2 フォワーディング テーブルの作成が開始します。この基になるのは、スイッチが受信するさまざまなパケットの送信元 MAC アドレスです。このフォワーディング テーブルが作成された後は、スイッチは MAC アドレス宛てのトラフィックを、対応するポートに直接転送します。

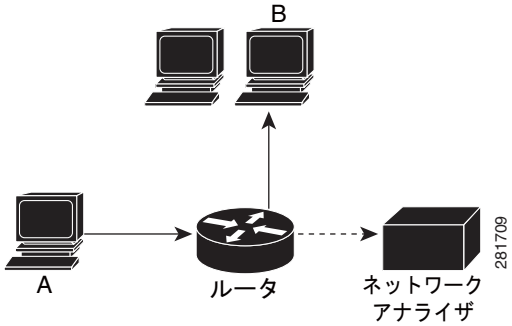
たとえば、ホスト A からホスト B に送信されるイーサネット トラフィックをキャプチャしようとするときに、双方が同じハブに接続されていれば、トラフィック アナライザをこのハブに接続するだけで済みます。他のすべてのポートは、ホスト A と B の間のトラフィックを認識します（図 17）。

図 17　　ハブのトラフィック ミラーリング動作



スイッチまたはルータにおいて、ホスト B の MAC アドレスが学習された後は、ホスト A から B へのユニキャスト トラフィックは B のポートだけに転送されます。したがって、トラフィック アナライザは、このトラフィックを認識しません（図 18）。

図 18　　トラフィック ミラーリングなしではルータ上ではネットワーク分析は動作しない



この設定では、トラフィック アナライザがキャプチャするのは、次のような、すべてのポートにフラッディングされるトラフィックだけです。

- ブロードキャスト トラフィック
- CGMP またはインターネット グループ管理プロトコル（IGMP）スヌーピングがディセーブル状態のマルチキャスト トラフィック
- スイッチ上の未知のユニキャスト トラフィック

ホスト A から送信されるユニキャスト パケットを人工的にコピーする、追加機能が必要です。この追加機能は、トラフィック ミラーリングです。トラフィック ミラーリングがイネーブルのときは、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。

Cisco ASR 9000 シリーズ ルータでのトラフィック ミラーリングの実装

トラフィック ミラーリング用語

- 入力トラフィック：スイッチに入るトラフィックです。
- 出トラフィック：スイッチから出るトラフィックです。
- 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：トラフィック ミラーリング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、宛先は単一、送信元は多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（モニタ対象ポートとも呼ばれます）は、ネットワーク トラフィック分析のためのモニタ対象であるスイッチドまたはルーテッド ポートです。1 つのローカルまたはリモート トラフィック ミラーリング セッションにおいて、送信元ポート トラフィック、たとえば、入力トラフィック（Rx）、出力トラフィック（Tx）、または双方向（入力および出力トラフィックの両方）をモニタできます。ルータは、任意の数の送信元ポートをサポートします（最大数は 800）。

送信元ポートの特性は、次のとおりです。

- ポート タイプは、バンドル インターフェイス、ギガビット イーサネット、10 ギガビット イーサネット、EFP など、どれでもかまいません。

（注） ブリッジ グループ仮想インターフェイス（BVI）はサポートされません。

- 1 つの送信元ポートは最大 1 つのトラフィック ミラーリング セッションでモニタできます。
- 宛先ポートにすることはできません。
- 部分的パケット ミラーリング。最初の 64 ～ 256 バイトのパケットをミラーリングできます。
- モニタする方向（入力、出力、または両方）を指定して、各送信元ポートを設定できます。バンドルの場合は、モニタ方向はグループ内のすべての物理ポートに適用されます。

図 19　Cisco ASR 9000 ルータでのトラフィック ミラーリングを使用したネットワーク分析

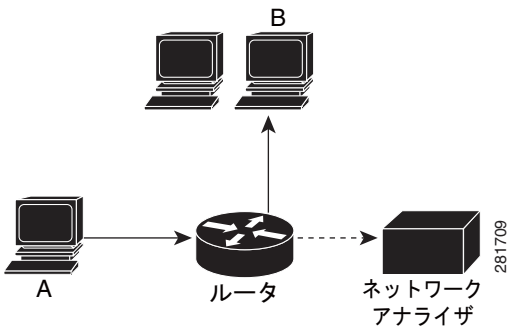


図 19 では、ネットワーク アナライザが接続されるポートは、ホスト A から送信されるすべてのパケットのコピーを受信するように設定されます。このポートを「トラフィック ミラーリング ポート」といいます。

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（*宛先ポート*と呼ばれる）に送信されます。VLAN タグ インポジションや ACL フィルタリングなどの操作を、ミラーリングされたトラフィック ストリームに対して実行することもできます。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、宛先ポートから送出されるトラフィックは、1 つまたは複数の送信元ポートからのトラフィックを結合したものとなり、各送信元ポートからのトラフィックは、VLAN プッシュ操作や ACL が適用されていることも、されていないこともあります。

モニタ セッションの特性を次に示します。

- 1 台の Cisco ASR 9000 ルータで最大 8 個のモニタ セッションを実行できます。
- 1 つのモニタ 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。
- Cisco ASR 9000 ルータ 1 台あたりの送信元ポート最大数は 800 です。
- 1 つのモニタ セッションあたりの送信元ポートの最大数は 800 です。ただし、すべてのモニタリング セッションの送信元ポート数合計が 800 を超えないものとします。

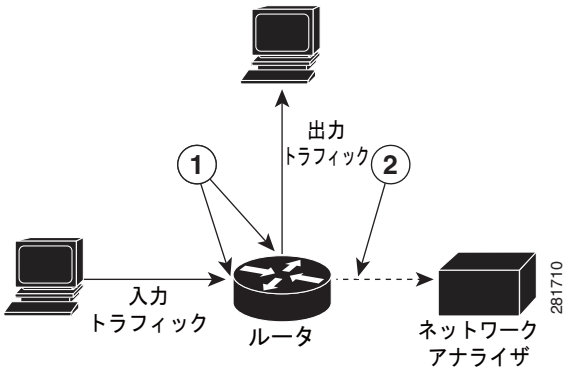
宛先ポートの特性

ローカルまたはリモート宛先セッションのそれぞれに 1 つの宛先ポート（モニタリング ポートとも呼ばれる）が必要です。このポートは、送信元ポートからトラフィックのコピーを受信します。

宛先ポートの特性は、次のとおりです。

- 宛先ポートは送信元ポートと同じルータ上に存在する必要があります。
- 任意のイーサネット物理ポート、EFP、疑似回線が宛先ポートなりますが、バンドル インターフェイスは宛先ポートにできません。
- 宛先ポートとなることができるのは、レイヤ 2 トランスポート インターフェイスだけです。L3 インターフェイスを SPAN の宛先として Cisco ASR 9000 シリーズ ルータ上で設定することはできません。
- 宛先ポートは、トランク（メイン）インターフェイスでもサブインターフェイスでもかまいません。
- 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。

図 20　Cisco ASR 9000 ルータでのトラフィック ミラーリングを使用したネットワーク分析



1	送信元トラフィック ミラーリング ポート（入力または出力トラフィック ポート）	2	トラフィック ミラーリング宛先ポート

サポートされるトラフィック ミラーリングのタイプ

これらのタイプのトラフィック ミラーリングがサポートされます。

- ローカル トラフィック ミラーリング。これは、最も基本的なトラフィック ミラーリングの形式です。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スイッチ上に存在します。
- リモート トラフィック ミラーリング（R-SPAN と呼ばれます）。この場合、ネットワーク アナライザは、宛先インターフェイスに直接接続するのではなく、スイッチがアクセス可能な VLAN に接続します。たとえば、宛先インターフェイスは、VLAN カプセル化を使用するサブ インターフェイスです。

  リモート トラフィック ミラーリングの制限形式は、ブリッジ ドメイン経由でスイッチングするのではなく、VLAN タグをプッシュする単一の宛先ポートにトラフィックを送信することで、実装できます。

  - ネットワーク アナライザと宛先は分離できますが、オンボックス冗長性はありません。
  - トラフィック ミラーリング VLAN に接続できるものである限り、複数のリモート ネットワーク アナライザを使用できます。

  これは、宛先ポートが VLAN タグをプッシュできる EFP であるため、Cisco IOS XR ソフトウェアでサポートされています。
- 疑似配線トラフィック ミラーリング（Cisco IOS ソフトウェアでは PW-SPAN と呼ばれます）。標準宛先インターフェイスを使用せずに、疑似回線経由でトラフィックが MPLS リモート サイトにミラーリングされます。
- ACL ベース トラフィック ミラーリング。トラフィックはグローバル 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。
- 部分的パケット ミラーリング。最初の 64 ～ 256 バイトのパケットをミラーリングできます。
- レイヤ 2 またはレイヤ 3 トラフィックのミラーリングがサポートされます。レイヤ 2 およびレイヤ 3 の両方の送信元ポートをミラーリングできます。

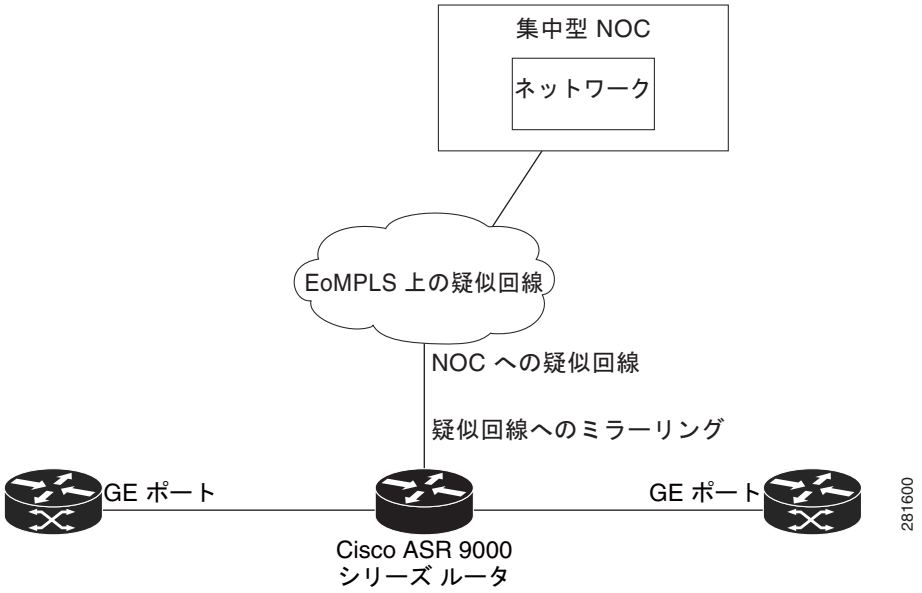
疑似配線トラフィック ミラーリング

トラフィック ミラーリングの宛先ポートとして、物理ポートではなく疑似配線を設定できます。この場合、送信元ポートで指定したトラフィックは、中央への疑似回線にミラーリングされます。これにより、高価なネットワーク トラフィック分析ツールを集中化することができます。

疑似配線が伝送するのはミラーリングされたトラフィックだけであるため、このトラフィックは基本的に単方向です。リモート プロバイダー エッジからのトラフィックはないはずです。

疑似配線トラフィック ミラーリング パスをネットワーク障害から保護するために、トラフィック エンジニアリング トンネルを優先パスとして設定し、高速再ルーティング保護を疑似配線に対してイネーブルにすることができます。

図 21　　疑似配線トラフィック ミラーリング



ACL ベースのトラフィック ミラーリング

グローバル インターフェイス アクセス リスト（ACL）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 2 トラフィックをミラーリングする場合、**ethernet-services access-list** コマンドを使用して **capture** キーワードを指定して ACL を設定します。レイヤ 3 トラフィックをミラーリングするときは、ACL の設定には
ipv4 access-list または **ipv6 access-list** コマンドと **capture** キーワードを使用します。**permit** コマンドおよび **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** キーワードは、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。

トラフィック ミラーリングの設定

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。

- 「ローカル トラフィック ミラーリングの設定方法」（P.290）
- 「リモート トラフィック ミラーリングの設定方法」（P.292）
- 「疑似回線でのミラーリング トラフィックの設定方法」（P.294）

- 「ACL ベース トラフィック ミラーリングの設定方法」（P.298）
- 「部分的パケット ミラーリングの設定方法」（P.301）

ローカル トラフィック ミラーリングの設定方法

手順の概要

1. **configure**
2. **monitor-session** *session-name*
3. **destination interface** *dest-interface*
4. **exit**
5. **interface** *source-interface*
6. **l2transport**
7. **monitor-session** *session-name* [**direction** {**rx-only** | **tx-only**]
8. **end**
   または
   commit
9. **show monitor-session** [*session-name*] **status** [**detail**] [**error**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**monitor-session** *session-name* **例：** `RP/0/RSP0/CPU0:router(config)# monitor-session mon1` `RP/0/RSP0/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** *dest-interface* **例：** `RP/0/RSP0/CPU0:router(config-mon)# destination interface gigabitethernet0/0/0/15`	トラフィックを複製する宛先インターフェイスを指定します。
ステップ 4	**exit** **例：** `RP/0/RSP0/CPU0:router(config-mon)# exit` `RP/0/RSP0/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。

	コマンドまたはアクション	目的
ステップ 5	**interface** *source-interface* **例：** RP/0/RSP0/CPU0:router(config)# interface gigabitethernet0/0/0/11	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。インターフェイス番号は、*rack*/*slot*/*module*/*port* 表記で入力します。ルータ構文の詳細については、疑問符（?）オンライン ヘルプ機能を使用します。
ステップ 6	**l2transport** **例：** RP/0/RSP0/CPU0:router(config-if)# l2transport	（任意）インターフェイスに対してレイヤ 2 トランスポート モードをイネーブルにし、レイヤ 2 トランスポート コンフィギュレーション モードを開始します。 **（注）** **l2transport** コマンドを使用してすべてのトラフィック タイプをミラーリングします。
ステップ 7	**monitor-session** *session-name* [**direction** {**rx-only** | **tx-only**] **例：** RP/0/RSP0/CPU0:router(config-if-l2)# monitor-session mon1	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トラフィックまたは出トラフィックだけがミラーリングされるように指定します。
ステップ 8	**end** または **commit** **例：** RP/0/RSP0/CPU0:router(config-if-l2)# end または RP/0/RSP0/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、変更をコミットするように要求されます。 Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]: – **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 9	**show monitor-session [session-name] status** [**detail**] [**error**] **例：** RP/0/RSP0/CPU0:router# show monitor-session	モニタ セッションに関する情報を表示します。

リモート トラフィック ミラーリングの設定方法

手順の概要

1. **configure**
2. **monitor-session** *session-name*
3. **destination interface** *dest-subinterface*
4. **exit**
5. **interface** *dest-subinterface* **l2transport**
6. **encapsulation dot1q** *vlan*
7. **rewrite ingress tag pop** *tag-to-remove*
8. **interface** *source-interface* [**l2transport**]
9. **monitor-session** *session-name* [**direction** {**rx-only** | **tx-only**]
10. **end**
    または
    commit
11. **show monitor-session** [*session-name*] **status** [**detail**] [**error**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**monitor-session** *session-name* **例：** `RP/0/RSP0/CPU0:router(config)# monitor-session mon1` `RP/0/RSP0/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** *dest-subinterface* **例：** `RP/0/RSP0/CPU0:router(config-mon)# destination interface gigabitethernet0/0/0/15`	どの宛先サブインターフェイスにトラフィックを複製するかを指定します。
ステップ 4	**exit** **例：** `RP/0/RSP0/CPU0:router(config-mon)# exit` `RP/0/RSP0/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。

	コマンドまたはアクション	目的
ステップ 5	**interface** *dest-subinterface* **l2transport** **例：** `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet0/0/0/11.10 l2transport`	指定したサブインターフェイスのインターフェイス コンフィギュレーション モードを開始します。インターフェイス番号は、*rack*/*slot*/*module*/*port* 表記で入力します。ルータ構文の詳細については、疑問符（?）オンライン ヘルプ機能を使用します。 **l2transport** キーワードを使用して、宛先サブインターフェイスに対してレイヤ 2 トランスポート モードをイネーブルにします。
ステップ 6	**encapsulation dot1q** *vlan* **例：** `RP/0/RSP0/CPU0:router(config-if)# encapsulation dot1q 1`	802.1Q カプセル化と、使用する VLAN 番号を指定します。
ステップ 7	**rewrite ingress tag pop** *tag-to-remove* **例：** `RP/0/RSP0/CPU0:router(config-if)# rewrite ingress tag pop 1`	EFP の外部タグだけを削除するように指定します。
ステップ 8	**interface** *source-subinterface* [**l2transport**] **例：** `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet0/0/0/11.10 l2transport`	特定のサブインターフェイスのインターフェイス コンフィギュレーション モードを開始します。インターフェイス番号は、*rack*/*slot*/*module*/*port* 表記で入力します。ルータ構文の詳細については、疑問符（?）オンライン ヘルプ機能を使用します。 レイヤ 2 サブインターフェイスを送信元インターフェイスとして設定するために、**l2transport** キーワードを使用してレイヤ 2 トランスポート モードをサブインターフェイスに対してイネーブルにします。
ステップ 9	**monitor-session** *session-name* [**direction** {**rx-only** | **tx-only**] **例：** `RP/0/RSP0/CPU0:router(config-if-l2)# monitor-session mon1`	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だけをミラーリングすることを指定します。

	コマンドまたはアクション	目的
ステップ 10	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 11	**show monitor-session [session-name] status [detail] [error]** **例：** `RP/0/RSP0/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。

疑似回線でのミラーリング トラフィックの設定方法

手順の概要

1. **configure**
2. **monitor-session** *session-name*
3. **destination pseudowire**
4. **exit**
5. **interface** *source-interface*
6. **l2transport**
7. **monitor-session** *session-name*
8. **exit**
9. **exit**
10. **exit**
11. **l2vpn**
12. **pw-class** *class-name*

13. **encapsulation mpls**
14. **exit**
15. **exit**
16. **xconnect group** *group-name*
17. **p2p** *xconnect-name*
18. **monitor-session** *session-name*
19. **neighbor** *peer-ip* **pw-id** *pseudowire-id*
20. **pw-class** *class-name*
21. **end**
    または
    commit
22. **show monitor-session** [*session-name*] **status** [**detail**] [**error**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**monitor-session** *session-name* **例：** `RP/0/RSP0/CPU0:router(config)# monitor-session mon1` `RP/0/RSP0/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 psuedowire** **例：** `RP/0/RSP0/CPU0:router(config-mon)# destination pseudowire`	トラフィックが疑似回線に複製されるように指定します。
ステップ 4	**exit** **例：** `RP/0/RSP0/CPU0:router(config-mon)# exit` `RP/0/RSP0/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** *source-interface* **例：** `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet0/0/0/11.10`	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。インターフェイス番号は、*rack*/*slot*/*module*/*port* 表記で入力します。ルータ構文の詳細については、疑問符（?）オンライン ヘルプ機能を使用します。

	コマンドまたはアクション	目的
ステップ 6	**l2transport** **例：** `RP/0/RSP0/CPU0:router(config-if)# l2transport`	（任意）サブインターフェイスでレイヤ 2 転送モードをイネーブルにし、レイヤ 2 転送コンフィギュレーション モードを開始します。 **（注）** **l2transport** コマンドを使用してすべてのトラフィック タイプをミラーリングします。
ステップ 7	**monitor-session** *session-name* [**direction** {**rx-only** | **tx-only**] **例：** `RP/0/RSP0/CPU0:router(config-if-l2)# monitor-session mon1`	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だけをミラーリングすることを指定します。
ステップ 8	**exit** **例：** `RP/0/RSP0/CPU0:router(config-if-mon)# exit` `RP/0/RSP0/CPU0:router(config-if-l2)#`	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、l2transport コンフィギュレーション モードに戻ります。
ステップ 9	**exit** **例：** `RP/0/RSP0/CPU0:router(config-if-l2)# exit` `RP/0/RSP0/CPU0:router(config-if)#`	l2transport 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 10	**exit** **例：** `RP/0/RSP0/CPU0:router(config-if)# exit` `RP/0/RSP0/CPU0:router(config)#`	インターフェイス コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。
ステップ 11	**l2vpn** **例：** `RP/0/RSP0/CPU0:router(config)# l2vpn` `RP/0/RSP0/CPU0:router(config-l2vpn)#`	レイヤ 2 VPN コンフィギュレーション モードを開始します。
ステップ 12	**pw-class** *class-name* **例：** `RP/0/RSP0/CPU0:router(config-l2vpn)# pw-class pw1`	疑似回線クラス テンプレートを設定して、疑似回線クラス テンプレート コンフィギュレーション モードを開始します。
ステップ 13	**encapsulation mpls** **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-pwc)# encapsulation mpls`	MPLS に疑似回線カプセル化を設定します。
ステップ 14	**exit** **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-pwc-mpls)# exit` `RP/0/RSP0/CPU0:router(config-l2vpn-pwc)`	疑似配線カプセル化コンフィギュレーション モードを終了します。

	コマンドまたはアクション	目的
ステップ 15	**exit** **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-pwc)# exit` `RP/0/RSP0/CPU0:router(config-l2vpn)`	疑似配線クラス テンプレート 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	**xconnect group** *group-name* **例：** `RP/0/RSP0/CPU0:router(config-l2vpn)# xconnect group g1`	グループ相互接続を設定します。
ステップ 17	**p2p** *xconnect-name* **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-xc)# p2p xc1`	ポイントツーポイント相互接続を設定します。
ステップ 18	**monitor-session** *session-name* **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# monitor-session mon1`	トラフィック ミラーリング セッションをポイントツーポイント相互接続に接続します。
ステップ 19	**neighbor** *peer-ip* **pw-id** *pseudowire-id* **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# neighbor 192.168.2.2 pw-id 3`	ポイントツーポイント相互接続を設定します。
ステップ 20	**pw-class** *class-name* **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# pw-class pw1`	疑似配線クラス テンプレートをこの相互接続に使用することを指定します。

	コマンドまたはアクション	目的
ステップ 21	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw)# end` または `RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 22	**show monitor-session [session-name] status** [**detail**] [**error**] **例：** `RP/0/RSP0/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。

ACL ベース トラフィック ミラーリングの設定方法

前提条件

グローバル インターフェイス ACL が、次のコマンドの 1 つと **capture** キーワードを使用して設定されていること。

- **ipv4 access-list**
- **ipv6 access-list**
- **ethernet-services access-list**

詳細については、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』または『*ASR 9000 Series Aggregation Services Router L2 VPN and Ethernet Services Command Reference*』を参照してください。

手順の概要

1. **configure**
2. **monitor-session** *session-name*
3. **destination interface** *dest-interface*
4. **exit**
5. **interface** *source-interface*
6. **l2transport**
7. **exit**
8. **ethernet-services access-group** *access-list-name* **ingress**
9. **monitor-session** *session-name*
10. **acl**
11. **end**
    または
    commit
12. **show monitor-session** [*session-name*] **status** [**detail**] [**error**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**monitor-session** *session-name* **例：** `RP/0/RSP0/CPU0:router(config)# monitor-session mon1` `RP/0/RSP0/CPU0:router(config-mon)#`	**（注）** 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** *dest-interface* **例：** `RP/0/RSP0/CPU0:router(config-mon)# destination interface gigabitethernet0/0/0/15`	トラフィックを複製する宛先インターフェイスを指定します。 **（注）**
ステップ 4	**exit** **例：** `RP/0/RSP0/CPU0:router(config-mon)# exit` `RP/0/RSP0/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** *source-interface* **例：** `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet0/0/0/11`	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。インターフェイス番号は、*rack*/*slot*/*module*/*port* 表記で入力します。ルータ構文の詳細については、疑問符（?）オンライン ヘルプ機能を使用します。

	コマンドまたはアクション	目的
ステップ 6	**l2transport** **例：** `RP/0/RSP0/CPU0:router(config-if)# l2transport`	（任意）サブインターフェイスでレイヤ 2 転送モードをイネーブルにし、レイヤ 2 転送コンフィギュレーション モードを開始します。 **（注）** **l2transport** コマンドを使用してすべてのトラフィック タイプをミラーリングします。
ステップ 7	**exit** **例：** `RP/0/RSP0/CPU0:router(config-if-l2)# exit` `RP/0/RSP0/CPU0:router(config-if)#`	レイヤ 2 トランスポート コンフィギュレーション モードを終了してインターフェイス コンフィギュレーション モードに戻ります。
ステップ 8	**ethernet-services access-group** *access-list-name* [**ingress** | **egress**] **例：** `RP/0/RSP0/CPU0:router(config-if)#` `ethernet-services access-group acl1 ingress`	アクセス リスト定義を、ミラーリング対象のインターフェイスに関連付けます。
ステップ 9	**monitor-session** *session-name* [ipv4|ipv6] [direction {rx-only|tx-only} **例：** `RP/0/RSP0/CPU0:router(config-if)#` `monitor-session mon1 direction rx-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。
ステップ 10	**acl** **例：** `RP/0/RSP0/CPU0:router(config-if-mon)# acl`	定義されたグローバル インターフェイス 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。
ステップ 11	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

	コマンドまたはアクション	目的
ステップ 12	**show monitor-session [session-name] status** [**detail**] [**error**] **例：** RP/0/RSP0/CPU0:router# show monitor-session	モニタ セッションに関する情報を表示します。

ACL ベース トラフィック ミラーリングのトラブルシューティング

設定に関する次の問題に注意してください。

- ミラーリング送信元ポートに対して **acl** コマンドが設定されていても、ACL コンフィギュレーション コマンドで **capture** キーワードが使用されていない場合は、トラフィックはミラーリングされません。
- ACL 設定で **capture** キーワードが使用されていても、**acl** コマンドが送信元ポートに対して設定されていない場合は、トラフィックはミラーリングされますが、アクセス リスト コンフィギュレーションは適用されません。
- 入力トラフィックはすべて、ACL 定義に関係なくミラーリングされます。出力トラフィックは、ACL 定義で許可されるものだけがミラーリングされます。

次の例では、ACL 定義の **capture** キーワードおよびインターフェイスに対する **acl** コマンドの両方が正しく設定されています。

```
monitor-session tm_example
!
ethernet-services access-list tm_filter
 10 deny 0000.1234.5678 0000.abcd.abcd any capture
!
interface GigabitEthernet0/2/0/0
 monitor-session tm_example direction rx-only
  acl
!
l2transport
 !
 ethernet-services access-group tm_filter ingress
!
end
```

部分的パケット ミラーリングの設定方法

手順の概要

1. **configure**
2. **monitor-session** *session-name*
3. **destination interface** *dest-interface*
4. **exit**
5. **interface** *source-interface*
6. **monitor-session** *session-name*
7. **mirror first** *bytes*

8. **end**
   または
   commit
9. **show monitor-session** [*session-name*] **status**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**monitor-session** *session-name* **例：** `RP/0/RSP0/CPU0:router(config)# monitor-session mon1` `RP/0/RSP0/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** *dest-interface* **例：** `RP/0/RSP0/CPU0:router(config-mon)# destination interface gigabitethernet0/0/0/15`	トラフィックを複製する宛先インターフェイスを指定します。
ステップ 4	**exit** **例：** `RP/0/RSP0/CPU0:router(config-mon)# exit` `RP/0/RSP0/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** *source-interface* **例：** `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet0/0/0/11.10`	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。インターフェイス番号は、*rack*/*slot*/*module*/*port* 表記で入力します。ルータ構文の詳細については、疑問符（?）オンライン ヘルプ機能を使用します。
ステップ 6	**monitor-session** *session-name*[**direction** {**rx-only** | **tx-only**] **例：** `RP/0/RSP0/CPU0:router(config-if-l2)# monitor-session mon1`	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だけをミラーリングすることを指定します。
ステップ 7	**mirror first** *bytes* **例：** `RP/0/RSP0/CPU0:router(config-if-mon)# mirror first bytes`	ミラーリングするパケットのバイト数を指定します。値の範囲は 64 ～ 256 バイトです。

	コマンドまたはアクション	目的
ステップ 8	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 9	**show monitor-session [session-name] status** **例：** `RP/0/RSP0/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。

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するトラフィック ミラーリング（ローカル）：例

次の例では、物理インターフェイスを使用するトラフィック ミラーリングの基本設定を示します。トラフィックがポイントツーポイント相互接続を介して gig0/2/0/19 と gig0/2/0/11 の間を流れるときに、gig0/2/0/19 で受信および送信されるパケットは gig0/2/0/15 にもミラーリングされます。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# monitor-session ms1
```

```
RP/0/RSP0/CPU0:router(config-mon)# destination interface gig0/2/0/15
RP/0/RSP0/CPU0:router(config-mon)# commit

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface gig0/2/0/11
RP/0/RSP0/CPU0:router(config-subif)# l2transport
RP/0/RSP0/CPU0:router(config-if-l2)# commit

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface gig0/2/0/15
RP/0/RSP0/CPU0:router(config-subif)# l2transport
RP/0/RSP0/CPU0:router(config-if-l2)# commit

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface gig0/2/0/19
RP/0/RSP0/CPU0:router(config-subif)# l2transport
RP/0/RSP0/CPU0:router(config-subif-l2)# monitor-session ms1
RP/0/RSP0/CPU0:router(config-if-l2)# commit

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# xconnect group xg1
RP/0/RSP0/CPU0:router(config-l2vpn-xc)# p2p xg1_p1
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# interface gig0/2/0/11
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# interface gig0/2/0/19
RP/0/RSP0/CPU0:router(config-if-l2)# commit
```

EFP を使用するトラフィック ミラーリング（リモート）：例

次の例では、EFP インターフェイスを使用するリモート トラフィック ミラーリングの基本設定を示します。トラフィックがポイントツーポイント相互接続を介して gig0/2/0/19.10 と gig0/2/0/11.10 の間を流れるときに、gig0/2/0/19.10 で受信および送信されるパケットは gig0/2/0/10.1 にもミラーリングされます。

```
RP/0/RSP0/CPU0:router#monitor-session ms1
RP/0/RSP0/CPU0:router(config)# destination interface gig0/2/0/10.1

RP/0/RSP0/CPU0:router(config)# interface gig0/2/0/10.1 l2transport
RP/0/RSP0/CPU0:router(config-if-l2)# encapsulation dot1q 1
RP/0/RSP0/CPU0:router(config-if-l2)# rewrite ingress tag pop 1

RP/0/RSP0/CPU0:router(config)# interface gig0/2/0/11.10 l2transport
RP/0/RSP0/CPU0:router(config-if-l2)# encapsulation dot1q 10

RP/0/RSP0/CPU0:router(config)# interface gig0/2/0/19.10 l2transport
RP/0/RSP0/CPU0:router(config-if-l2)# encapsulation dot1q 10
RP/0/RSP0/CPU0:router(config-if-l2)# monitor-session ms1

RP/0/RSP0/CPU0:router(config)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# xconnect group xg1
RP/0/RSP0/CPU0:router(config-l2vpn-xc)# p2p xg1_p1
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# interface gig0/2/0/11.10
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# interface gig0/2/0/19.10
```

モニタ セッションのステータスの表示：例

次の例は、**show monitor-session** コマンドに **status** キーワードを指定したときの出力例です。

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

```
Monitor-session cisco-rtp1
Destination interface GigabitEthernet0/5/0/38
================================================================================
Source Interface      Dir   Status
--------------------- ----  ------------------------------------------------
Gi0/5/0/4             Both  Operational
Gi0/5/0/17            Both  Operational

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

Monitor-session sess1
  Destination interface is not configured
  Source Interfaces
  -----------------
  GigabitEthernet0/0/0/0
    [Direction:] Both
    ACL match: Enabled
    Portion:   Full packet
    Status:    Not operational (destination interface not known).
GigabitEthernet0/0/0/2
    [Direction:] Both
    ACL match: Disabled
    Portion:   First 100 bytes

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

Monitor-session ms1
Destination interface GigabitEthernet0/2/0/15 is not configured
================================================================================
Source Interface      Dir   Status
--------------------- ----  ------------------------------------------------

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

モニタ セッション統計情報：例

show monitor-session コマンドと **counters** キーワードを使用して、さまざまな送信元ポートの統計情報/カウンタ（受信/送信/ドロップ）を表示します。このコマンドでは、モニタ セッションごとに、すべての送信元インターフェイスのリストと、各インターフェイスの複製パケット統計情報が表示されます。

各インターフェイスに関して表示される統計情報のすべてのセットは次のとおりです。

- 複製された RX パケットおよびオクテット
- 複製された TX パケットおよびオクテット
- 複製されないパケットとオクテット

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

Monitor-session ms1
  GigabitEthernet0/2/0/19.10
    Rx replicated: 1000 packets, 68000 octets
    Tx replicated: 1000 packets, 68000 octets
```

```
    Non-replicated: 0 packets, 0 octets
```

clear monitor-session counters コマンドを使用して、収集済みの統計情報をクリアします。デフォルトでは、このコマンドはすべての保存された統計情報をクリアします。ただし、任意のインターフェイス フィルタを使用できます。

```
RP/0/RSP0/CPU0:router# clear monitor-session counters
```

疑似回線上のトラフィック ミラーリング：例

次に、疑似回線にトラフィック ミラーリングを設定する方法の例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/11/0/1
RP/0/RSP0/CPU0:router(config-if)# l2transport
RP/0/RSP0/CPU0:router(config-if-l2)# monitor-session pw-span-test

RP/0/RSP0/CPU0:router(config)# monitor-session pw-span-test
RP/0/RSP0/CPU0:router(config-mon)# destination pseudowire

RP/0/RSP0/CPU0:router(config)# l2vpn
RP/0/RSP0/CPU0:router(config-l2vpn)# pw-class class1
RP/0/RSP0/CPU0:router(config-l2vpn-pwc)# encapsulation mpls

RP/0/RSP0/CPU0:router(config-l2vpn)# xconnect group g1
RP/0/RSP0/CPU0:router(config-l2vpn-xc)# p2p x1
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# monitor-session pw-span-test
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p)# neighbor 2.2.2.2 pw-id 1
RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw)# pw-class class1

RP/0/RSP0/CPU0:router(config-l2vpn-xc-p2p-pw)# commit
```

レイヤ 3 ACL ベース トラフィック ミラーリング：例

次の例では、レイヤ 3 ACL ベース トラフィック ミラーリングを設定する方法を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# monitor-session ms1
RP/0/RSP0/CPU0:router(config-mon)# destination interface gig0/2/0/15
RP/0/RSP0/CPU0:router(config-mon)# commit

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface gig0/2/0/11
RP/0/RSP0/CPU0:router(config-if)# ipv4 access-group span ingress
RP/0/RSP0/CPU0:router(config-if)# monitor-session ms1
RP/0/RSP0/CPU0:router(config-if-mon)# commit

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# ipv4 access-list span
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 5 permit ipv4 any any dscp 5 capture
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit ipv4 any any
RP/0/RSP0/CPU0:router(config-ipv4-acl)# commit
```

レイヤ 2 ACL ベースのトラフィック ミラーリング：例

次に、レイヤ 2 ACL ベースのトラフィック ミラーリングを設定する方法の例を示します。

```
RP/0/RSP0/CPU0:router# configure
```

```
RP/0/RSP0/CPU0:router(config)# monitor-session ms1
RP/0/RSP0/CPU0:router(config-mon)# destination interface gig0/2/0/15
RP/0/RSP0/CPU0:router(config-mon)# commit

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface gig0/2/0/11
RP/0/RSP0/CPU0:router(config-if)# l2transport
RP/0/RSP0/CPU0:router(config-if-l2)# exit
RP/0/RSP0/CPU0:router(config-if)# ethernet-services access-group acl_mirror ingress
RP/0/RSP0/CPU0:router(config-if)# acl
RP/0/RSP0/CPU0:router(config-if)# monitor-session ms1
RP/0/RSP0/CPU0:router(config-if-mon)# commit

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# ipv4 access-list acl_mirror
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 5 permit ipv4 any any dscp 5 capture
RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit ipv4 any any
RP/0/RSP0/CPU0:router(config-ipv4-acl)# commit
```

部分的パケット ミラーリング：例

次の例では、パケットの最初の 100 バイトをミラーリングするように設定する方法を示します。

```
RP/0/RP0/CPU0:router(config)# interface gigabitethernet0/0/0/11
RP/0/RP0/CPU0:router(config-if-l2)# monitor-session mon1
RP/0/RSP0/CPU0:router(config-if-mon)# mirror first 100
```

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ある場合は、まず、**show monitor-session status** コマンドの出力を確認して、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。

```
Monitor-session sess1
<Session status>
================================================================================
Source Interface      Dir   Status
--------------------- ----  ------------------------------------------------
Gi0/0/0/0             Both  <Source interface status>
Gi0/0/0/2             Both  <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** コマンドの出力を調べて、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> is not configured	宛先として設定されたインターフェイスは存在しません。たとえば、VLAN サブインターフェイスを宛先インターフェイスとして設定したが、その VLAN サブインターフェイスがまだ作成されていない場合です。

Session Status	説明
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。状態を確認するには、**show interfaces** コマンドを使用します。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）。
Destination pseudowire is not configured	疑似配線をセットアップする L2VPN 設定が見つかりません。トラフィック ミラーリング セッション名を、xconnect p2p の 1 セグメントとして設定します。
Destination pseudowire <name> (down)	疑似配線は設定されていますが、ダウンしています。L2VPN 設定を調べて、疑似配線がアップ状態にならない原因を特定します。

<Source interface status> で報告される可能性のあるメッセージは次のとおりです。

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** コマンドの出力を調べて、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。
Not operational (destination interface 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。状態を確認するには、**show interfaces** コマンドを使用します。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** コマンドを実行すると、追加情報が表示されます。

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

Monitor-session sess1
  Destination interface is not configured
  Source Interfaces
  -----------------
  GigabitEthernet0/0/0/0
    [Direction:] Both
    ACL match: Enabled
    Portion:   Full packet
    Status:    Not operational (destination interface not known).
GigabitEthernet0/0/0/2
    [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'.
```

この詳細な出力を見ると、問題を特定できることがあります。

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/RSP0/CPU0:router# show monitor-session platform trace ?

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

RP/0/RSP0/CPU0:router# show monitor-session trace ?

  process  Filter debug by process

RP/0/RSP0/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/RSP0/CPU0:router# debug monitor-session platform all

RP/0/RSP0/CPU0:router# debug monitor-session platform event

RP/0/RSP0/CPU0:router# debug monitor-session platform info

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

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

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

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

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

次の作業

イーサネット インターフェイスの設定が完了したら、イーサネット インターフェイスで各 VLAN サブインターフェイスを設定できます。

シェルフ コントローラ（SC）、ルート プロセッサ（RP）、および分散型 RP のイーサネット管理インターフェイスの変更方法については、このマニュアルで後述する「Cisco ASR 9000 シリーズ ルータでの管理イーサネット インターフェイスの高度な設定および変更」モジュールを参照してください。

IPv6 については、『*Cisco* IOS XR IP Addresses and Services Configuration Guide』の「*Implementing Access Lists and Prefix Lists on Cisco IOS XR Software*」モジュールを参照してください。

その他の関連資料

ここでは、トラフィック ミラーリングの実装に関連する参考資料を示します。

関連資料

関連項目	参照先
イーサネット L2VPN	*『Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide』* *『Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Command Reference』*
Cisco IOS XR マスター コマンド リファレンス	*『Cisco ASR 9000 Series Aggregation Services Router Master Commands List』*
Cisco IOS XR インターフェイス コンフィギュレーション コマンド	*『Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Command Reference』*
ユーザ グループとタスク ID に関する情報	*『Cisco IOS XR Interface and Hardware Component Command Reference』*

標準

標準	タイトル
なし	—

MIB

MIB	MIB のリンク
なし	Cisco IOS XR ソフトウェアを使用して選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	—

シスコのテクニカル サポート

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

その他の関連資料

Cisco ASR 9000 シリーズ ルータでの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、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータでの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インターフェイスは、グローバル スコープを持ちますが、関連付けられた位置は持ちません。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 1Loopback1、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（RSPRP）上に存在します。設定とコントロール プレーンは、スタンバイ RSP RP 上にミラーリングされ、フェールオーバースイッチ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 RP となります。

Cisco IOS XR ソフトウェアでのループバック インターフェイスおよびヌル インターフェイス設定機能の履歴

リリース	変更内容
リリース 3.7.2	この機能は、Cisco ASR 9000 シリーズ ルータで導入されました。

内容

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」（P.314）
- 「ヌル インターフェイスの概要」（P.314）
- 「仮想管理インターフェイスの概要」（P.315）
- 「アクティブ/スタンバイ RP および仮想インターフェイスの設定」（P.315）

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、ただちに同一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、では、仮想ループバック インターフェイスは次の機能を実行します。

- ループバック 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がループバック インターフェイス宛てで送信されない IP パケットは、ドロップされます。これらの 2 つの状況では、ループバック インターフェイスはヌル インターフェイスのように動作できます。

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ります。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。

Null 0 Null0 インターフェイスは、起動時にデフォルトで RSP RP 上に作成され、削除はできません。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 Null0 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。

仮想管理インターフェイスの概要

IPv4 仮想アドレスを設定することにより、どの RSP RP がアクティブであるかを事前に把握していなくても、管理ネットワークでの単一の仮想アドレスからルータにアクセスすることができます。IPv4 仮想アドレスは、ルート スイッチ プロセッサ（RSPRP）フェールオーバースイッチオーバーが発生しても存続します。そのためには、IPv4 仮想アドレスは両方の RP の管理イーサネット インターフェイスと共通の IPv4 サブネットを共有する必要があります。

Cisco ASR 9000 シリーズ ルータ Cisco XR 12000 シリーズ ルータ Cisco CRS-1 ルータで各 RSP RP が複数の管理イーサネット インターフェイスを持つ場合は、IPv4 仮想アドレスのマッピング先は同じ IP サブネットを共有するアクティブ RSP RP の管理イーサネット インターフェイスとなります。

アクティブ/スタンバイ RP および仮想インターフェイスの設定

スタンバイ RSP RP は、使用可能であり、アクティブ RSP RP からの作業引き継ぎが必要であればいつでも引き継げる状態になっています。スタンバイ RSP RP がアクティブ RSP RP となってアクティブ RSPRP の役割を引き継ぐことが必要になる状況としては、次のものがあります。

- ウォッチドッグによる障害検出
- 管理コマンドの引き継ぎ
- シャーシからのアクティブ RSP RP の取り外し

セカンダリ RSP RP がシャーシに存在していない状態でプライマリが動作中のときに、セカンダリ RSP RP を挿入すると、自動的にスタンバイ RSP RP になります。シャーシからスタンバイ RSP RP を取り外しても、RSP RP の冗長性が失われるだけで、システムに影響はありません。

フェールオーバースイッチオーバー後は、すべての仮想インターフェイスがスタンバイ（新たにアクティブになった）RSP RP 上に存在します。仮想インターフェイスのステートと設定は変更されず、フェールオーバースイッチオーバー時にインターフェイス経由の転送（トンネルの場合）が失われることはありません。ルータは、ホスト RSP RP のフェールオーバースイッチオーバーを通じて、バンドルおよびトンネルで上で無停止転送（NSF）を使用します。

（注） スタンバイ インターフェイスの設定維持を保証するために、ユーザ側で何かを設定する必要はありません。

（注） tacacs source-interface、snmp-server trap-source、ntp source、logging source-interface などのプロトコル コンフィギュレーションでは、送信元として仮想管理 IP アドレスをデフォルトでは使用しません。**ipv4 virtual address use-as-src-addr** コマンドを使用して、プロトコルが仮想 IPv4 アドレスを送

信元アドレスとして使用するようにします。また、指定した、または目的の IPv4 アドレスを使用してループバック アドレスを設定し、それを TACACS+ などのプロトコルの送信元として **tacacs source-interface** コマンドにより設定することもできます。

仮想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。

- 「仮想ループバック インターフェイスの設定」（P.316）（必須）
- 「ヌル インターフェイスの設定」（P.317）（必須）
- 「仮想 IPv4 IPV4 インターフェイスの設定」（P.319）（必須）

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。

手順の概要

1. **configure**
2. **interface loopback** *instance*
3. **interface loopback** *interface-path-id*
4. **ipv4 address** *ip-address*
5. **end**
   または
   commit
6. **show interfaces** *type instance*
7. **show interfaces** *type interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0RP00/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* **interface loopback** *interface-path-id* **例：** `RP/0/RSP0RP00/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/RSP0RP000/CPU0:router(config-if)# ipv4 address 172.18.189.38/32`	**ipv4 address** コンフィギュレーション コマンドを使用して、仮想ループバック インターフェイスに IP アドレスおよびサブネット マスクを割り当てます。
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0RP00/CPU0:router(config-if)# end` または `RP/0/RSP0RP00/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	**show interfaces** *type instance* **show interfaces** *type interface-path-id* **例：** `RP/0/RSP0RP00/CPU0:router# show interfaces Loopback 3`	（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。

ヌル インターフェイスの設定

ここでは、基本的なヌル ヌル インターフェイスの設定手順について説明します。

手順の概要

1. **configure**
2. **interface null 0**
3. **end**
   または
   commit
4. **show interface null 0**
5. **show interfaces** *type interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0RP00/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/RSP0RP00/CPU0:router#(config)# interface null 0`	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/RSP0RP00/CPU0:router(config-null0)# end` または `RP/0/RSP0RP00/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 4	**show interfaces null 0** **例：** `RP/0/RSP0RP00/CPU0:router# show interfaces null 0null0`	ヌル インターフェイスの設定を確認します。

仮想 IPv4 IPV4 インターフェイスの設定

ここでは、IPv4 仮想インターフェイスの設定手順について説明します。

手順の概要

1. **configure**
2. **ipv4 address virtual address** *ipv4ip-address subnet mask*

3. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0RP00/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ipv4 address virtual address** *ipv4-address subnet address/mask* **例：** `RP/0/RSP0RP00/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/RSP0RP00/CPU0:router(config-null0)# end` または `RP/0/RSP0RP00/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

仮想インターフェイスの設定例

ここでは、次の設定例について説明します。

- 「ループバック インターフェイスの設定例」（P.320）
- 「ヌル インターフェイスの設定例」（P.321）

ループバック インターフェイスの設定例

次に、ループバック インターフェイスの設定例を示します。

```
RP/0/RSP0RP00/CPU0:router# configure
RP/0/RSP0RP00/CPU0:router(config)# interface Loopback 3
RP/0/RSP0RP00/CPU0:router(config-if)# ipv4 address 172.18.189.38/32
RP/0/RSP0RP00/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
RP/0/RSP0RP00/CPU0:router# show interfaces Loopback 3

Loopback3 is up, line protocol is up
  Hardware is Loopback interface(s)
  Internet address is 172.18.189.38/32
  MTU 1514 bytes, BW Unknown
     reliability 0/255, txload Unknown, rxload Unknown
  Encapsulation Loopback,  loopback not set
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
     0 packets output, 0 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
```

ヌル インターフェイスの設定例

次に、ヌル インターフェイスの設定例を示します。

```
RP/0/RSP0RP00/CPU0:router# configure
RP/0/RSP0RP00/CPU0:router(config)# interface Null 0
RP/0/RSP0RP00/CPU0:router(config-null0)# ipv4 unreachables
RP/0/RSP0RP00/CPU0:router(config-null0)# end
Uncommitted changes found, commit them? [yes]: yes
RP/0/RSP0RP00/CPU0:router# show interfaces Null 0

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

仮想 IPv4 インターフェイスの設定：例

```
RP/0/RSP0RP00/CPU0:router# configure
RP/0/RSP0RP00/CPU0:router(config)# ipv4 virtual address 10.3.32.154/8
RP/0/RSP0RP00/CPU0:router(config-null0)# commit
```

その他の関連資料

ここでは、ループバック インターフェイスおよびヌル インターフェイスの設定に関連する参考資料について説明します。

関連資料

関連項目	参照先
Cisco ASR 9000 シリーズ ルータ マスター コマンド リファレンス	*『Cisco ASR 9000 Series Router Master Commands List』*
Cisco ASR 9000 シリーズ ルータ インターフェイス コンフィギュレーション コマンド	*『Cisco ASR 9000 Series Router Interface and Hardware Component Command Reference』*
Cisco IOS XR ソフトウェアを使用する Cisco ASR 9000 シリーズ ルータの初期システム ブートアップと設定に関する情報。	*『Cisco ASR 9000 Series Router Getting Started Guide』*
ユーザ グループとタスク ID に関する情報	*『Cisco ASR 9000 Series Router Interface and Hardware Component Command Reference』*

関連項目	参照先
Cisco IOS XR マスター コマンド リファレンス	*『Cisco IOS XR Master Commands List』*
Cisco IOS XR インターフェイス コンフィギュレーション コマンド	*『Cisco IOS XR Interface and Hardware Component Command Reference』*
Cisco IOS XR ソフトウェアを使用するルータを初回に起動し設定するための情報	*『Cisco IOS XR Getting Started Guide』*
ユーザ グループとタスク ID に関する情報	*『Cisco IOS XR Interface and Hardware Component Command Reference』*
リモートの Craft Works Interface（CWI）クライアント管理アプリケーションからのインターフェイスとその他のコンポーネントの設定に関する情報	*『Cisco Craft Works Interface Configuration Guide』*

標準

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

MIB

MIB	MIB のリンク
このモジュールに適用できる MIB はありません。	Cisco IOS XR ソフトウェアを使用して選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	—

シスコのテクニカル サポート

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

その他の関連資料

Cisco ASR 9000 シリーズ ルータ でのチャネライズド SONET/SDH の設定

このモジュールでは、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータでのチャネライズド SONET/SDH の設定について説明します。

Cisco IOS XR ソフトウェアでのチャネライズド SONET/SDH 設定機能の履歴

リリース	変更内容
リリース 3.9.0	Cisco ASR 9000 シリーズ ルータ に関して、次の SPA のサポートが追加されました。 • Cisco 2 ポート チャネライズド OC-12/DS0 SPA
リリース 4.0.0	Cisco ASR 9000 シリーズ ルータ に関して、次の SPA のサポートが追加されました。 • Cisco 1 ポート チャネライズド OC-48/STM-16 SPA SDH、E3、E1 および POS チャネル化のサポートが Cisco 2 ポート チャネライズド OC-12/DS0 および Cisco 1 ポート チャネライズド OC-48/STM-16 SPA に対して追加されました。
リリース 4.0.1	Cisco ASR 9000 シリーズ ルータ に関して、次の SPA のサポートが追加されました。 • Cisco 1 ポート チャネライズド OC-3/STM-1 SPA

内容

チャネライズド SONET/SDH 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。

チャネライズド SONET/SDH を設定する前に、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。

- シャーシに、次の SPA のうち少なくとも 1 つが設置されている必要があります。
  - Cisco 1 ポート チャネライズド OC-3/STM-1 SPA
  - Cisco 2 ポート チャネライズド OC-12c/DS0 SPA
  - Cisco 1 ポート チャネライズド OC-48/STM-16 SPA
- 汎用表記 *rack/slot/module/port* を使用して SONET コントローラ名と *interface-path-id* を適用 / 指定する方法を理解している必要があります。SONET コントローラ名と *interface-path-id* は、**controller sonet** コマンドで必要となります。

チャネライズド SONET/SDH の設定に関する情報

チャネライズド SONET/SDH 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。

- 「チャネライズド SONET の概要」（P.326）
- 「チャネライズド SDH の概要」（P.331）
- 「チャネライズド SONET/SDH のデフォルト設定値」（P.334）
- 「チャネライズド SONET/SDH の設定方法」（P.335）

チャネライズド SONET の概要

同期光ファイバ ネットワーク（SONET）は、光ファイバでのデジタル テレコミュニケーション サービス伝送において使用される米国規格協会（ANSI）の規格形式です。

同期デジタル ハイアラーキ（SDH）は、SONET の国際版に相当します。

チャネライズド SONET では、多重化 T3/E3 および仮想トリビュタリ グループ（VTG）チャネルで SONET フレームを転送することができます。

チャネライズド SONET は、次の SPA でサポートされます。

- Cisco 1 ポート チャネライズド OC-48/STM-16 SPA
- Cisco 1 ポート チャネライズド OC-3/STM-1 SPA
- Cisco 2 ポート チャネライズド OC-12c/DS0 SPA

チャネライズド SDH は、次の SPA でサポートされます。

- Cisco 1 ポート チャネライズド OC-48/STM-16 SPA
- Cisco 1 ポート チャネライズド OC-3/STM-1 SPA
- Cisco 2 ポート チャネライズド OC-12c/DS0 SPA

SONET は、同期転送信号（STS）フレーム構成を使用します。STS は、オプティカル キャリア 1（OC-1）の電気版に相当します。

SDH は、同期転送モード（STM）フレーム構成を使用します。1 つの STM-1 は、3 つのオプティカル キャリア 1（OC-1）の電気版に相当します。

チャネライズド SONET インターフェイスは、複数の STS ストリームを複合したものであり、固有のペイロード ポインタを持つ独立したフレームとして維持されます。フレームは、転送される前に多重化されます。

回線がチャネル化されると、*パス*と呼ばれるより小さい帯域幅のチャネルに論理的に分割されます。これらのパスが SONET ペイロードを伝送します。全パスの帯域幅の合計は回線の帯域幅を超過できません。

回線がチャネル化されない場合、この回線は*クリア チャネル*と呼ばれ、回線の全帯域幅がブロードバンド サービスを伝送する単一のチャネル専用となります。

STS ストリームは、次のタイプのチャネルにチャネル化することができます。

- T3/E3
- VT1.5 がマッピングされた T1
- Packet over SONET/SDH（POS）（OC12 および OC48 のみ）

T3/E3 チャネルは、さらに T1 にチャネル化でき、T1 はタイムスロット（DS0）にチャネル化できます。ただし、T1 または DS0 をサポートしない 1 ポート チャネライズド OC-48/STM-16 SPA の場合は除きます。

SONET 回線のチャネル化は、次の 2 つの主要なプロセスで構成されます。

- コントローラの設定
- インターフェイスのチャネライズド パスへの設定

最初に、STS パスのモードを設定することによりコントローラを設定します。モードは、使用するハードウェアのサポートに応じて、T3、または VT1.5 マッピング T1、または POS に設定できます。

（注） POS は、Cisco 1 ポート チャネライズド OC-12/DS0 SPA の STS-3c パスと STS-12c パス、および Cisco 1 ポート チャネライズド OC-48/STM-16 SPA の STS-3c、STS-12c、STS-48c の各パスでのみサポートされます。

モードが指定されると、各コントローラが作成され、残りの設定がそのコントローラに適用されます。たとえば、T3 モードでは T3 コントローラが作成されます。T3 コントローラは、シリアル チャネルに対して設定するか、または T1 を伝送するためにさらにチャネル化できます。これらの T1 は、シリアル インターフェイスに対して設定できます。

設置されている SPA のサポートに応じて、各 STS パスを個別に T3、E3、VTG などに設定できます。

図 22 に、1 台の SONET コントローラに対する 3 つの STS パスの例を示します。ただし、2 ポート チャネライズド OC-12/DS0 SPA は最大 12 個の STS パスをサポートし、1 ポート チャネライズド OC-48/STM-16 SPA は最大 48 個の STS パスをサポートしますが、1 ポート チャネライズド OC-48/STM-16 SPA は VTG をサポートしません。

図 22　　SONET コントローラ STS パス

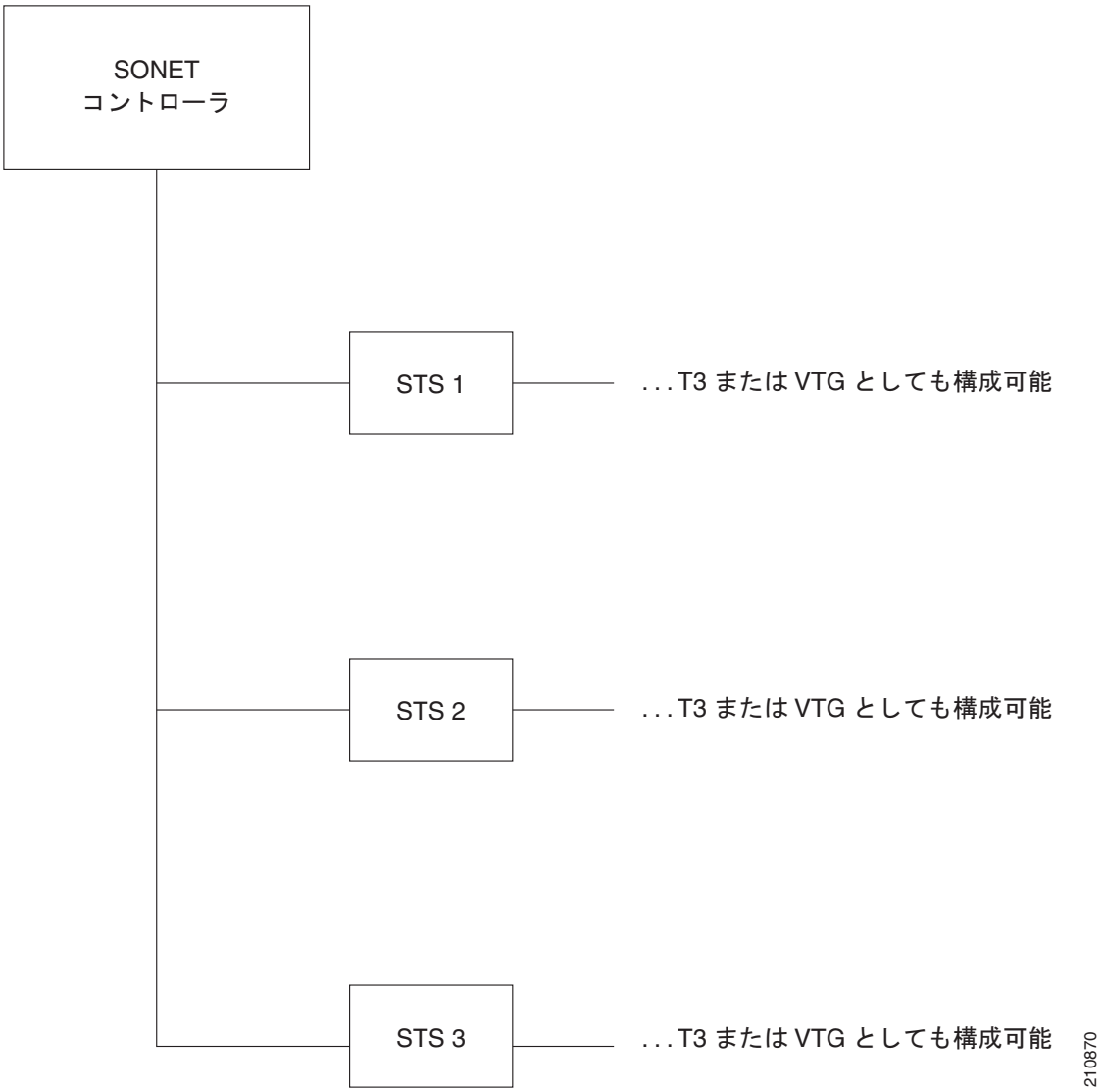


図 23 に、SONET コントローラ設定の組み合わせの例を示します。

（注） Cisco ASR 9000 シリーズ ルータの 1 ポート チャネライズド OC-48/STM-16 SPA は VTG をサポートしません。

図 23　　SONET コントローラの設定の組み合わせ

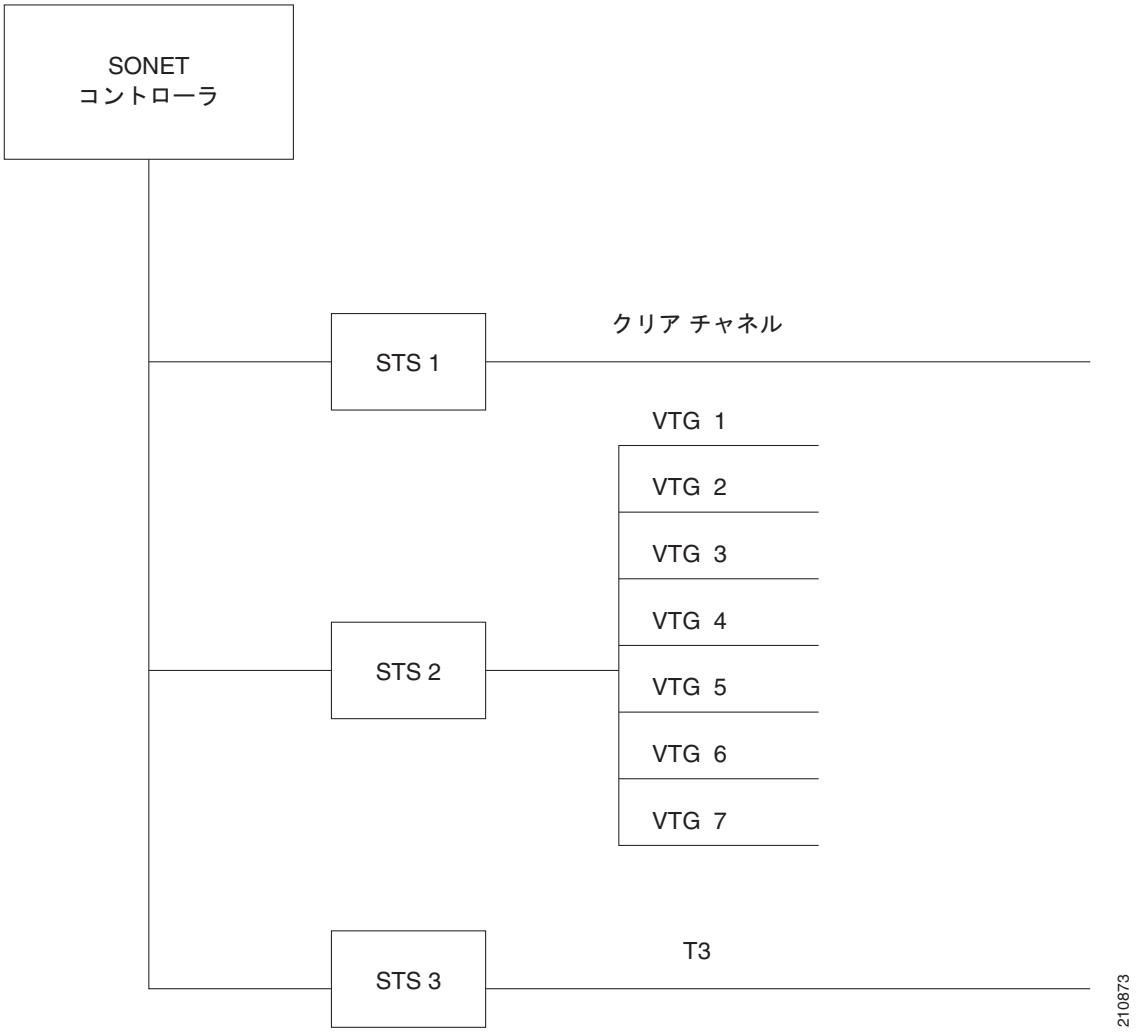


図 24 は、設定可能な T3 パスを示しています。

（注） チャネライズド T3 パスは、1 ポート チャネライズド OC-3/STM-1 SPA と 2 ポート チャネライズド OC-12c/DS0 SPA でのみサポートされます。

図 24　　SONET T3 チャネライズド パス

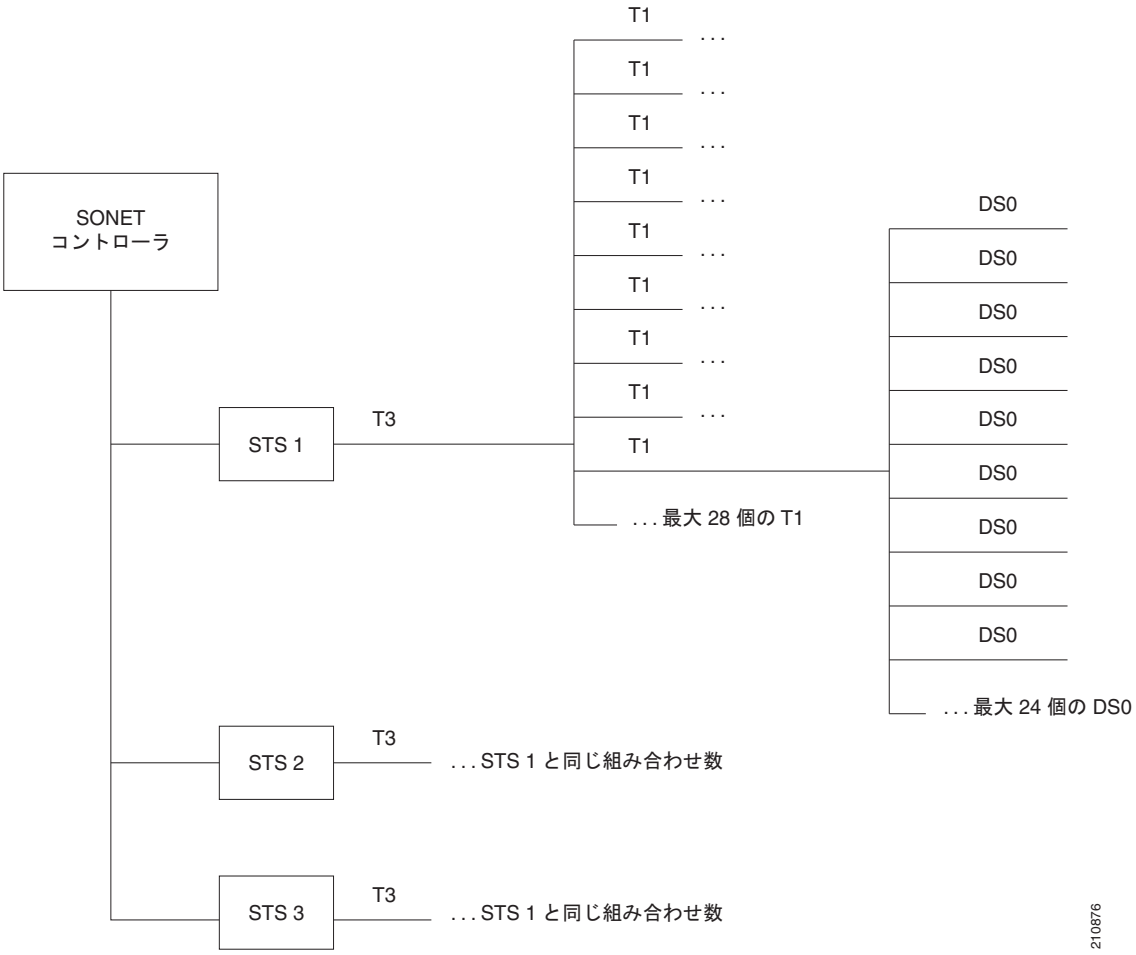
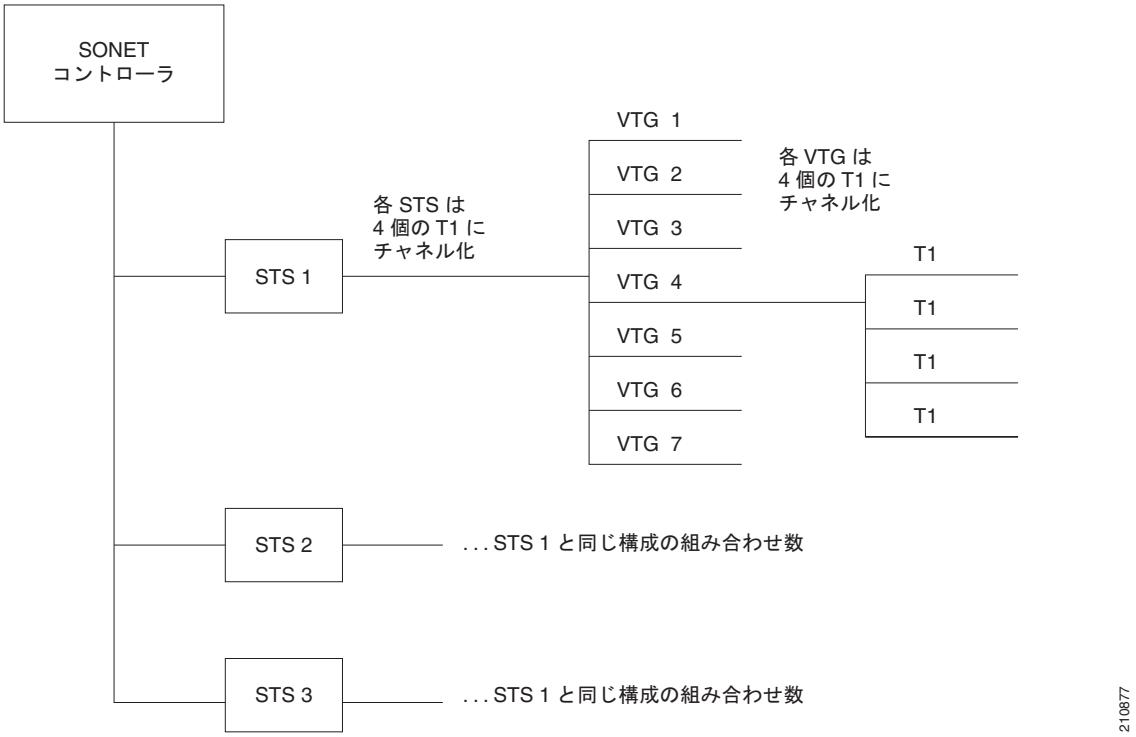


図 25 は、設定可能な VTG パスを示しています。

（注） VTG パスがサポートされるのは、Cisco ASR 9000 シリーズ ルータ上の Cisco 1 ポート チャネライズド OC-3/STM-1 SPA および Cisco 2 ポート チャネライズド OC-12c/DS0 SPA のみです。

図 25　　SONET VTG チャネライズド パス



チャネライズド SDH の概要

同期デジタル ハイアラーキ（SDH）は、SONET の国際版に相当します。

チャネライズド SDH は、次の SPA でサポートされます。

- Cisco 1 ポート チャネライズド OC-48/STM-16 SPA
- Cisco 1 ポート チャネライズド OC-3/STM-1 SPA
- Cisco 2 ポート チャネライズド OC-12/DS0 SPA

同期転送モジュール（STM）信号は、SONET の STS の同期デジタル ハイアラーキ（SDH）版に相当しますが、各帯域幅で番号は異なります。ここでは、STM という用語はパス幅と光回線レートの両方を表します。STM 信号内のパスは、管理ユニット（AU）と呼ばれます。

SONET と SDH 間での基本的な用語の違いの概要を次に示します。

- SONET の STS は、SDH の管理ユニット（AU）に相当
- SONET の仮想トリビュタリ（VT）は、SDH のトリビュタリ ユニット（TU）に相当
- SDH の基本ビルディング ブロックは STM-1（STS-3 に相当）および STM-0（STS-1 に相当）

管理ユニット（AU）は、より上位のパス レイヤと多重化セクション レイヤ間の適合を可能にする情報構造です。AU は、情報ペイロード（より上位の仮想コンテナ）と管理ユニット ポインタで構成されます。管理ユニット ポインタは、ペイロード フレーム開始のオフセットを多重化セクション フレーム開始と相対的に示します。

AU は、トリビュタリ ユニット（TU）およびトリビュタリ ユニット グループ（TUG）にチャネル化することができます。

管理ユニット 4（AU-4）は、3 つの STM-1 または 1 つの STM-3 で構成されます。

管理ユニット 3（AU-3）は、1 つの STM-1 で構成されます。

管理ユニット グループ（AUG）は、STM ペイロードにおいて固定の定義された位置を占める 1 つまたは複数の管理ユニットで構成されます。

表 2　　SONET/SDH 用語対照表

SONET 用語	SDH 用語
SONET	SDH
STS-3c	AU-4
STS-1	AU-3
VT	TU
SPE	VC
セクション	リジェネレータ セクション
回線	多重化セクション
パス	パス

Cisco ASR 9000 シリーズ ルータでは、次のレベルの SDH チャネル化がサポートされます。

- 1 ポート チャネライズド OC-3/STM-1 SPA
  - AU4 から TUG-3 から TUG-2 から VC-12 から E1 から NxDS0
  - AU4 から TUG-3 から VC-3 から DS3（クリア チャネル）
  - AU4 から TUG-3 から VC-3 から E3（クリア チャネル）
  - AU3 から TUG-2 から VC-11 から DS1 から NxDS0
- 2 ポート チャネライズド OC-12/DS0 SPA
  - AU-4-4c（VC-4-4c）
  - AU-4（VC-4）
  - AU-4 から TUG-3 から VC-3 から DS3
  - AU-4 から TUG-3 から VC-3 から E3
  - AU-4 から TUG-3 から TUG-2 から VC-11 から T1 から NxDS0
  - AU-4 から TUG-3 から TUG-2 から VC-12 から E1 から NxDS0
  - AU-3 から VC-3 から DS3
  - AU-3 から TUG-2 から VC-11 から T1 から NxDS0
  - AU-3 から TUG-2 から VC-12 から E1 から NxDS0
  - AU-3 から VC-3 から E3
  - AU-3 から VC-3 から DS3 から T1 から NxDS0

  - AU-3 から VC-3 から DS3 から E1 から NxDS0
- 1 ポート チャネライズド OC-48/STM-16 SPA
  - DS3
  - E3
  - AU-3（VC-3）
  - AU-4（VC-4）
  - AU-4-4c（VC-4-4c）
  - AU-4-16c（VC-4-16c）

図 26 に、サポートされている SPA で設定可能な SDH AU-3 パスの例を示します。

（注） 1 ポート チャネライズド OC-48/STM-16 SPA では、AU-3 パスをさらに T1 にチャネル化することはサポートされません。

図 26　　SDH AU3 パス

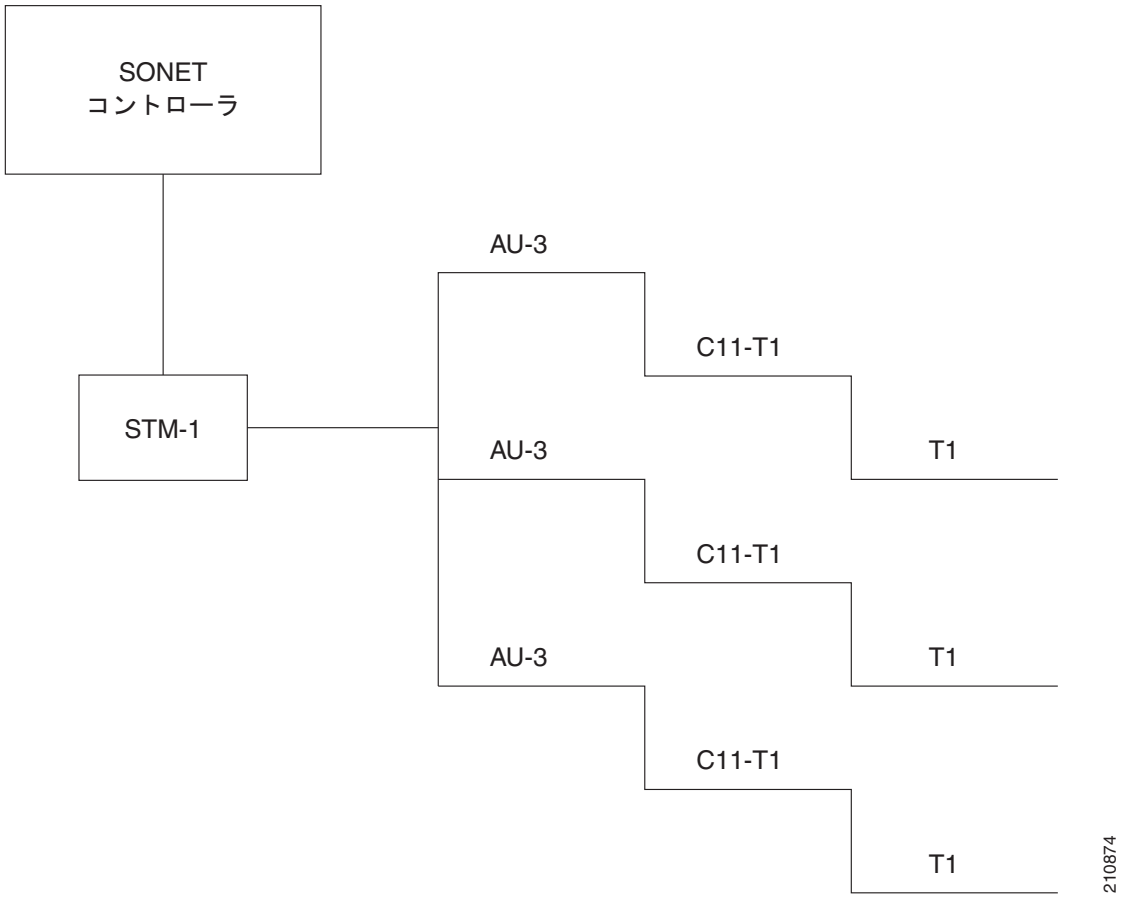
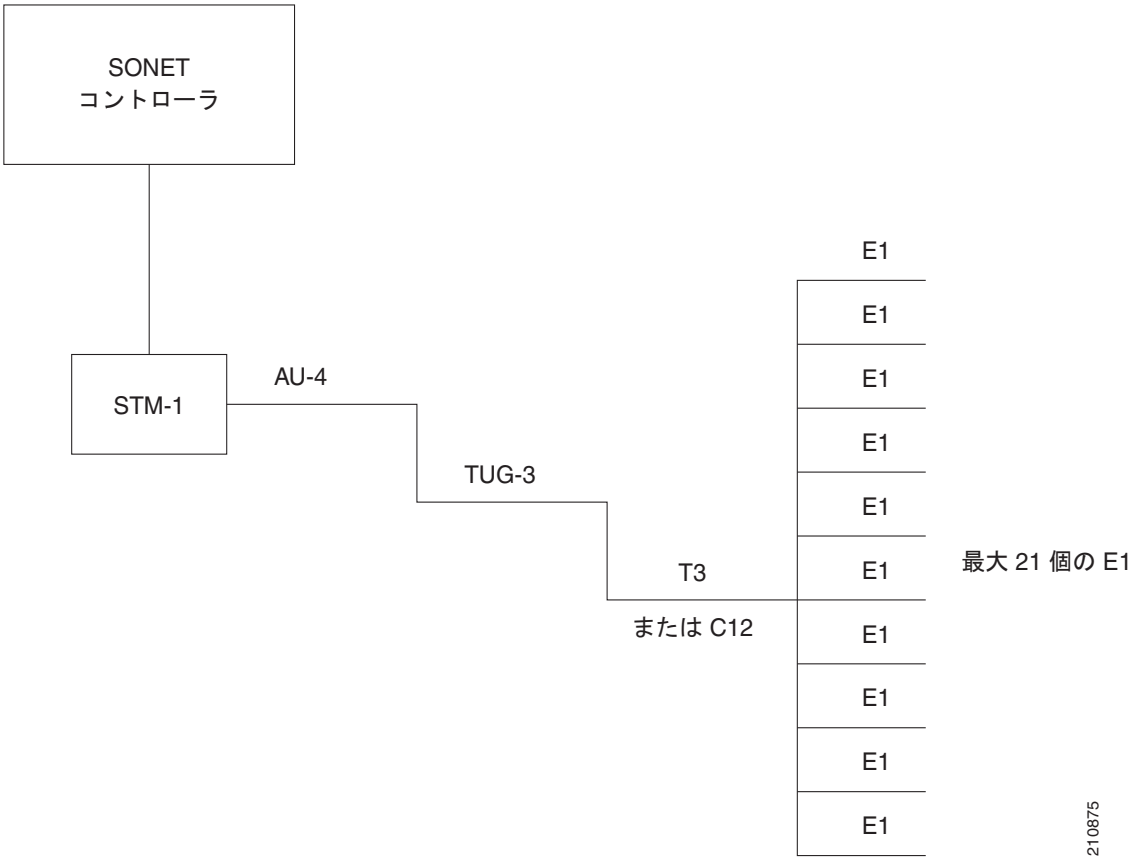


図 27 に、サポートされる SPA で設定できる SDH AU4 パスを表示します。

（注） 1 ポート チャネライズド OC-48/STM-16 SPA は、T3 または E3 レベルへのチャネル化だけをサポートします。AU-4 パスのさらなるチャネル化はサポートされません。

図 27　　SDH AU4 パス



チャネライズド SONET/SDH のデフォルト設定値

表 3 に、チャネライズド SONET/SDH に存在するデフォルト設定パラメータを示します。

表 3　　SONET/SDH コントローラのデフォルト設定値

パラメータ	デフォルト値	設定ファイルのエントリ
クロック ソース	line	**clock source** {**internal** | **line**}
SONET フレーミング	sonet	**framing** {**sdh** | **sonet**}

チャネライズド SONET/SDH の設定方法

ここでは、次の手順について説明します。

SONET T3 チャネルおよび VT1.5 がマッピングされた T1 チャネルの設定

ここでは、SONET 回線を T3 チャネルおよび VT がマッピングされた T1 チャネルに設定する手順について説明します。

前提条件

- *「Cisco ASR 9000 シリーズ ルータ でのクリア チャネル SONET コントローラの設定」*モジュールの「クリア チャネル SONET コントローラの設定方法」に示す SONET コントローラの設定方法を理解している必要があります。
- 次の SPA では、STS パスを T3 にチャネル化することができます。
  - Cisco 1 ポート チャネライズド OC-48/STM-16 SPA
  - Cisco 1 ポート チャネライズド OC-3/STM-1 SPA
  - Cisco 2 ポート チャネライズド OC-12/DS0 SPA
- STS パスの VTG がマッピングされた T1 へのチャネル化は、次の SPA で行えます。
  - Cisco 1 ポート チャネライズド OC-3/STM-1 SPA
  - Cisco 2 ポート チャネライズド OC-12/DS0 SPA
- 次の SPA では、T3 パスを T1 または E1 にチャネル化することができます。
  - Cisco 1 ポート チャネライズド OC-3/STM-1 SPA
  - Cisco 2 ポート チャネライズド OC-12/DS0 SPA
- Cisco 2 ポート チャネライズド OC-12/DS0 SPA では、T1 パスを NxDS0 にチャネル化することができます。

制約事項

T1 および E1 は、Cisco 1 ポート チャネライズド OC-48/STM-16 SPA でサポートされません。

手順の概要

1. **configure**
2. **controller sonet** *interface-path-id*
3. **clock source** {**internal** | **line**}

4. **framing sonet**
5. **sts** *number*
6. **mode** *mode*
7. **width** *number*
8. **root**
9. **controller** *controllerName instance*
10. **mode** *mode*
11. **root**
12. **controller t1** *interface-path-id*
13. **channel-group** *number*
14. **timeslots** *num1:num2:num3:num4* または
    timeslots *range1-range2*
15. **show configuration**
16. **root**
17. **interface serial** *interface-path-id*
18. **encapsulation** {**frame-relay** | **hdlc** | **ppp**}
19. **ipv4** *ip-address mask*
20. **no shutdown**
21. **end**
    または
    commit
22. **show**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller sonet** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller sonet 0/1/1/0`	SONET コントローラ コンフィギュレーション サブモードを開始して、SONET コントローラ名と *interface-path-id* を *rack/slot/module/port* 表記で指定します。

	コマンドまたはアクション	目的
ステップ 3	**clock source** {**internal** | **line**} **例：** `RP/0/RSP0/CPU0:router(config-sonet)# clock source internal`	SONET ポート転送クロック ソースを設定します。ここで、**internal** キーワードは内部クロック、**line** キーワードは回線から回収されたクロックを設定します。 • ネットワークからクロッキングを得られる場合は、必ず **line** キーワードを使用します。2 つのルータがバックツーバックまたは光ファイバで接続されており、クロッキングが得られない場合は、**internal** キーワードを使用します。 • デフォルト キーワードは **line** です。 **（注）** スペース再利用プロトコル（SRP）インターフェイスでは、内部クロッキングが必要です。
ステップ 4	**framing sonet** **例：** `RP/0/RSP0/CPU0:router(config-sonet)# framing sonet`	SONET フレーム構成のコントローラを設定します。 SONET フレーム構成（**sonet**）がデフォルトです。
ステップ 5	**sts** *number* **例：** `RP/0/RSP0/CPU0:router(config-sonet)# sts 1`	*number* により指定された STS ストリームを設定します。有効値の範囲を次に示します。 • 1 ～ 48：1 ポート チャネライズド OC-48/STM-16 SPA • 1 ～ 3：1 ポート チャネライズド OC-3/STM-1 SPA • 1 ～ 12：2 ポート チャネライズド OC-12/DS0 SPA
ステップ 6	**mode** *mode* **例：** `RP/0/RSP0/CPU0:router(config-stsPath)# mode t3`	STS レベルでのインターフェイスのモードを設定します。設定可能なモードは、次のとおりです。 • t3：T3 を伝送する SONET パス • vt15-t1：仮想トリビュタリ 1.5 T1 を伝送する SONET パス（VT15 T1）（1 ポート チャネライズド OC-3/STM-1 SPA および 2 ポート チャネライズド OC-12c/DS0 SPA のみ） • pos：Packet over SONET
ステップ 7	**width** *number* **例：** `RP/0/RSP0/CPU0:router(config-stsPath)# width 3`	連結される STS ストリーム数を設定します。 *number* に設定可能な値を次に示します。 • 1：STS ストリーム数 1 を示します。 • 3：STS ストリーム数 3 を示します（STS-3c）。 • 12：12 個の STS ストリームが連結することを示します（STS-12c）。 • 48：48 個の STS ストリームが連結することを示します（STS-48c）。これは、1 ポート チャネライズド OC-48/STM-16 SPA のデフォルトです。 自然境界の STS パスには、幅 3、12、48 が設定されます。これは、次のパス番号と適合します。 • STS-3c では 1、4、7、10 など • STS-12c では 1、13、25、37 • STS-48c では 1

	コマンドまたはアクション	目的
ステップ 8	**root** **例：** `RP/0/RSP0/CPU0:router(config-stsPath)# root`	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 9	**controller** *controllerName instance* **例：** `RP/0/RSP0/CPU0:router(config)# controller t3 0/1/1/0/0`	コントローラ コンフィギュレーション サブモードを開始して、コントローラ名とインスタンス ID を *rack/slot/module/port/controllerName* 表記で指定します。コントローラ名を次に示します。 • t3：T3 を伝送する SONET パス • vt15-t1：仮想トリビュタリ 1.5 T1 を伝送する SONET パス（VT15 T1）（1 ポート チャネライズド OC-3/STM-1 SPA および 2 ポート チャネライズド OC-12c/DS0 SPA のみ）
ステップ 10	**mode** *mode* **例：** `RP/0/RSP0/CPU0:router(config-t3)# mode t1`	このレベルでのインターフェイスのモードを設定します。設定可能なモードは、次のとおりです。 • t1：28 の T1 にチャネル化（1 ポート チャネライズド OC-3/STM-1 SPA および 2 ポート チャネライズド OC-12c/DS0 SPA のみ） • e1：21 の E1 にチャネル化（1 ポート チャネライズド OC-3/STM-1 SPA および 2 ポート チャネライズド OC-12c/DS0 SPA のみ） • serial：HDLC に類似するペイロードを伝送するクリア チャネル
ステップ 11	**root** **例：** `RP/0/RSP0/CPU0:router(config-t3)# root`	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 12	**controller t1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t1 0/1/1/0/0/0`	T1 コントローラ コンフィギュレーション サブモードを開始して、T1 コントローラ名と *interface-path-id* を *rack/slot/module/port/T3Num/T1num* 表記で指定します。 （1 ポート チャネライズド OC-3/STM-1 SPA および 2 ポート チャネライズド OC-12c/DS0 SPA のみ）
ステップ 13	**channel-group** *number* **例：** `RP/0/RSP0/CPU0:router(config-t1)# channel-group 1`	タイム スロットの割り当て先となるチャネル グループ番号を設定します。範囲は 1 ～ 24 です。
ステップ 14	**timeslots** *num1:num2:num3:num4* または **timeslots** *range1-range2* **例：** `RP/0/0/CPU0:router(config-t1-channel_group)# timeslots 1:3:7:9` `RP/0/0/CPU0:router(config-t1-channel_group)# timeslots 1-24`	インターフェイスのタイム スロットを *num1:num2:num3:num4* 表記で数字で指定するか、*range1-range2* 表記で範囲として指定します。

	コマンドまたはアクション	目的
ステップ 15	**show configuration** **例：** `RP/0/RSP0/CPU0:router(config-t1-channel_group)# show configuration`	コミットされていない設定の内容を表示します。
ステップ 16	**root** **例：** `RP/0/RSP0/CPU0:router(config-t3)# root`	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 17	**interface serial** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface serial 0/1/1/0/0/0:0`	完全なインターフェイス番号を *rack/slot/module/port/T3Num/T1num:instance* 表記で指定します。
ステップ 18	**encapsulation** {**frame-relay** | **hdlc** | **ppp**} **例：** `RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp`	カプセル化のタイプを、次のいずれかのキーワードを使用して指定します。 • **frame-relay**：フレームリレー ネットワーク プロトコル • **hdlc**：ハイレベル データリンク コントロール（HDLC）同期プロトコル • **ppp**：ポイントツーポイント プロトコル
ステップ 19	**ipv4** *ip-address mask* **例：** `RP/0/RSP0/CPU0:router(config-if)# ip address 10.10.10.10 255.255.255.255`	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。
ステップ 20	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-if)# no shutdown`	shutdown 設定を削除します。 **（注）** shutdown 設定を削除することにより、インターフェイスでの強制的な管理上の停止が排除されるため、インターフェイスはアップ状態またはダウン状態に移行することができます（親 SONET レイヤが管理上の停止状態に設定されていないことを前提とします）。

	コマンドまたはアクション	目的
ステップ 21	**end** または **commit** **例：** `RP/0/0RSP0/CPU0:router(config-sonet)# end` または `RP/0/RSP0/CPU0:router(config-sonet)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 22	**show controllers sonet** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show controllers sonet 0/1/1/0`	SONET コントローラの設定を確認します。

Packet over SONET チャネルの設定

ここでは、チャネライズド SONET をサポートする SPA の Packet over SONET（POS）チャネルを設定する方法について説明します。

前提条件

次のいずれかの SPA がインストールされていること。

- Cisco 1 ポート チャネライズド OC-48/STM-16 SPA
- Cisco 2 ポート チャネライズド OC-12/DS0 SPA

手順の概要

1. **configure**
2. **controller sonet** *interface-path-id*
3. **clock source** {**internal** | **line**}
4. **framing** {**sdh** | **sonet**}
5. **sts** *number*
6. **width** *number*

7. **mode** *mode* **scramble**
8. **root**
9. **interface pos** *interface-path-id*
10. **encapsulation** [**hdlc** | **ppp** | **frame-relay** [**IETF**]]
11. **pos crc** {**16** | **32**}
12. **mtu** *value*
13. **no shutdown**
14. **end**
    または
    commit
15. **show interfaces pos** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller sonet** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller sonet 0/1/1/0`	SONET コントローラ コンフィギュレーション サブモードを開始して、SONET コントローラ名と *interface-path-id* を *rack/slot/module/port* 表記で指定します。
ステップ 3	**clock source** {**internal** | **line**} **例：** `RP/0/RSP0/CPU0:router(config-sonet)# clock source internal`	SONET ポート転送クロック ソースを設定します。ここで、**internal** キーワードは内部クロック、**line** キーワードは回線から回収されたクロックを設定します。 • ネットワークからクロッキングを得られる場合は、必ず **line** キーワードを使用します。2 つのルータがバックツーバックまたは光ファイバで接続されており、クロッキングが得られない場合は、**internal** キーワードを使用します。 • デフォルト キーワードは **line** です。 **（注）** スペース再利用プロトコル（SRP）インターフェイスでは、内部クロッキングが必要です。
ステップ 4	**framing** {**sdh** | **sonet**} **例：** `RP/0/RSP0/CPU0:router(config-sonet)# framing sonet`	（任意）同期デジタル ハイアラーキ（SDH）フレーム構成の場合は **sdh** キーワード、SONET フレーム構成の場合は **sonet** キーワードを使用して、コントローラのフレーム構成を設定します。 SONET フレーム構成（**sonet**）がデフォルトです。
ステップ 5	**sts** *number* **例：** `RP/0/RSP0/CPU0:router(config-sonet)# sts 1`	*number* により指定された STS ストリームを設定します。有効値の範囲を次に示します。 • 2 ポート チャネライズド OC12c/DS0 SPA では 1 ～ 12 • 1 ポート チャネライズド OC48/DS3 SPA では 1 ～ 48

	コマンドまたはアクション	目的
ステップ 6	**width** *number* **例：** `RP/0/RSP0/CPU0:router(config-stsPath)# width 3`	連結される STS ストリーム数を設定します。 *number* に設定可能な値を次に示します。 • 3：STS ストリーム数 3 を示します（STS-3c）。 • 12：12 個の STS ストリームが連結することを示します（STS-12c）。 • 48：48 個の STS ストリームが連結することを示します（STS-48c）。 自然境界の STS パスには、幅 3、12、48 が設定されます。これは、次のパス番号と適合します。 • STS-3c では 1、4、7、10 など • STS-12c では 1、13、25、37 • STS-48c では 1 **（注）** 幅が 1 の場合、POS インターフェイスはサポートされません。
ステップ 7	**mode** *mode* **scramble** **例：** `RP/0/RSP0/CPU0:router(config-stsPath)# mode pos scramble`	STS レベルでのインターフェイスのモードを設定します。POS インターフェイスを作成するために、モードを pos に設定します（OC12 および OC48 のみ）。
ステップ 8	**root** **例：** `RP/0/RSP0/CPU0:router(config-stsPath)# root`	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 9	**interface pos** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface POS 0/1/1/0`	POS インターフェイス名と *rack/slot/module/port* 表記を指定して、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。
ステップ 10	**encapsulation** [**hdlc** | **ppp** | **frame-relay** [**IETF**]] **例：** `RP/0/RSP0/CPU0:router(config-if)# encapsulation hdlc`	（任意）インターフェイス カプセル化パラメータおよび HDLC やポイントツーポイント プロトコル（PPP）などの詳細を設定します。デフォルトは HDLC です。
ステップ 11	**pos crc** {**16** | **32**} **例：** `RP/0/RSP0/CPU0:router(config-if)# pos crc 32`	（任意）インターフェイスの CRC 値を設定します。16 ビットの CRC モードを指定するには **16** キーワード、**32** ビットの CRC モードを指定するには 32 キーワードを入力します。 デフォルト CRC は **32** です。
ステップ 12	**mtu** *value* **例：** `RP/0/RSP0/CPU0:router(config-if)# mtu 4474`	（任意）POS MTU 値を設定します。 有効値の範囲は 64 ～ 65,535 です。

	コマンドまたはアクション	目的
ステップ 13	**no shutdown** **例：** `RP/0/RSP0/CPU0:router (config-if)# no shutdown`	shutdown 設定を削除します。 **（注）** shutdown 設定を削除することにより、インターフェイスでの強制的な管理上の停止が排除されるため、インターフェイスはアップ状態またはダウン状態に移行することができます（親 SONET レイヤが管理上の停止状態に設定されていないことを前提とします）。
ステップ 14	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-sonet)# end` または `RP/0/RSP0/CPU0:router(config-sonet)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 15	**show interfaces pos** *interface-path-id* **例：** `RP/0/0/CPU0:router# show interfaces pos 0/1/1/0`	（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。

T3 のためのクリア チャネル SONET コントローラの設定

このタスクでは、SONET 回線を、*クリア チャネル*と呼ばれる単一の T3 シリアル チャネルとするように設定する方法について説明します。クリア チャネルは、T3 コントローラ モードを serial に設定することにより確立されます。

前提条件

- *「Cisco ASR 9000 シリーズ ルータ でのクリア チャネル SONET コントローラの設定」*モジュールの「クリア チャネル SONET コントローラの設定方法」に示す SONET コントローラの設定方法を理解している必要があります。

手順の概要

1. **configure**
2. **controller sonet** *interface-path-id*

3. **clock source** {**internal** | **line**}
4. **framing sonet**
5. **sts** *number*
6. **mode t3**
7. **root**
8. **controller t3** *interface-path-id*
9. **mode serial**
10. **root**
11. **interface serial** *interface-path-id*
12. **encapsulation** {**frame-relay** | **hdlc** | **ppp**}
13. **ipv4** *ip-address mask*
14. **no shutdown**
15. **end**
    または
    commit
16. **show controllers sonet** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller sonet** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller sonet 0/1/1/0`	SONET コントローラ コンフィギュレーション サブモードを開始して、SONET コントローラ名と *interface-path-id* を *rack/slot/module/port* 表記で指定します。
ステップ 3	**clock source** {**internal**	**line**} **例：** `RP/0/RSP0/CPU0:router(config-sonet)# clock source internal`	SONET ポート転送クロック ソースを設定します。ここで、**internal** キーワードは内部クロック、**line** キーワードは回線から回収されたクロックを設定します。 • ネットワークからクロッキングを得られる場合は、必ず **line** キーワードを使用します。2 つのルータがバックツーバックまたは光ファイバで接続されており、クロッキングが得られない場合は、**internal** キーワードを使用します。 • デフォルト キーワードは **line** です。 **（注）** スペース再利用プロトコル（SRP）インターフェイスでは、内部クロッキングが必要です。

	コマンドまたはアクション	目的
ステップ 4	**framing sonet** **例：** RP/0/RSP0/CPU0:router(config-sonet)# framing sonet	SONET フレーム構成のコントローラを設定します。SONET フレーム構成（**sonet**）がデフォルトです。
ステップ 5	**sts** *number* **例：** RP/0/RSP0/CPU0:router(config-sonet)# sts 1	*number* により指定された STS ストリームを設定します。有効値の範囲を次に示します。 • 1 ～ 48：1 ポート チャネライズド OC-48/DS3 SPA • 1 ～ 3：1 ポート チャネライズド OC-3/STM-1 SPA • 1 ～ 12：2 ポート チャネライズド OC-12/DS0 SPA
ステップ 6	**mode t3** **例：** RP/0/RSP0/CPU0:router(config-stsPath)# mode t3	STS レベルでのインターフェイスのモードを T3 用に設定します。
ステップ 7	**root** **例：** RP/0/RSP0/CPU0:router(config-stsPath)# root	グローバル コンフィギュレーション モードに戻ります。
ステップ 8	**controller t3** *interface-path-id* **例：** RP/0/RSP0/CPU0:router(config)# controller t3 0/1/1/0/0	T3 コントローラ コンフィギュレーション サブモードを開始して、T3 コントローラ名と interface-path-id の ID を *rack/slot/module/port/T3Num* 表記で指定します。
ステップ 9	**mode serial** **例：** RP/0/RSP0/CPU0:router(config-t3)# mode serial	クリア チャネルを確立するためにインターフェイスのモードをシリアルに設定します。
ステップ 10	**root** **例：** RP/0/RSP0/CPU0:router(config-t3)# root	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 11	**interface serial** *interface-path-id* **例：** RP/0/RSP0/CPU0:router(config)# interface serial 0/1/1/0/0/0:0	完全なインターフェイス番号を *rack/slot/module/port/T3Num/T1num:instance* 表記で指定します。
ステップ 12	**encapsulation** {**frame-relay** | **hdlc** | **ppp**} **例：** RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp	カプセル化のタイプを、次のいずれかのキーワードを使用して指定します。 • **frame-relay**：フレームリレー ネットワーク プロトコル • **hdlc**：ハイレベル データリンク コントロール（HDLC）同期プロトコル • **ppp**：ポイントツーポイント プロトコル

	コマンドまたはアクション	目的
ステップ 13	**ipv4** *ip-address mask* **例：** `RP/0/RSP0/CPU0:router(config-if)# ip address 10.10.10.10 255.255.255.255`	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。
ステップ 14	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-if)# no shutdown`	shutdown 設定を削除します。 **（注）** shutdown 設定を削除することにより、インターフェイスでの強制的な管理上の停止が排除されるため、インターフェイスはアップ状態またはダウン状態に移行することができます（親 SONET レイヤが管理上の停止状態に設定されていないことを前提とします）。
ステップ 15	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-sonet)# end` または `RP/0/RSP0/CPU0:router(config-sonet)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 16	**show controllers sonet** *interface-path-id* **例：** `RP/0//RSP0/CPU0:router# show controllers sonet 0/1/1/0`	SONET コントローラの設定を確認します。

チャネライズド SONET 自動保護スイッチング（APS）の設定

ここでは、チャネライズド SONET 回線で自動保護スイッチング（APS）を設定する手順について説明します。

前提条件

- *「Cisco ASR 9000 シリーズ ルータ でのクリア チャネル SONET コントローラの設定」*モジュールの「クリア チャネル SONET コントローラの設定方法」に示す SONET コントローラの設定方法を理解している必要があります。

- *「Cisco ASR 9000 シリーズ ルータ でのクリア チャネル SONET コントローラの設定」*モジュールの「SONET APS の設定」に示す SONET APS の設定方法を理解している必要があります。

制約事項

- 1 ポート チャネライズド OC-48/STM-16 SPA では、SONET APS はサポートされません。
- Cisco ASR 9000 シリーズ ルータでマルチルータ APS がサポートされるのは、次の SPA のみです。
  - 1 ポート チャネライズド OC-3/STM-1 SPA
  - 2 ポート チャネライズド OC-12c/DS0 SPA

手順の概要

1. **aps group** *number*
2. **channel 0 local sonet** *interface*
   または
   channel 0 remote *ip-address*
3. **channel 1 local sonet** *interface*
   または
   channel 1 remote *ip-address*
4. **signalling** {**sonet** | **sdh**}
5. **end**
   または
   commit
6. **show aps**
7. **show aps group** [*number*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**aps group** *number* **例：** `RP/0/RSP0/CPU0:router(config)# aps group 1`	指定した番号を持つ APS グループを追加して、APS グループ コンフィギュレーション モードを開始します。 • **aps group** コマンドは、グローバル コンフィギュレーション モードで使用します。 • グループを削除するには、**no aps group** *number* のように、このコマンドの **no** 形式を使用します。有効値の範囲は 1 ～ 255 です。 **（注）** **aps group** コマンドを使用するには、aps コマンドの適切なタスク ID に関連付けられたユーザ グループのメンバーでなければなりません。 **（注）** **aps group** コマンドは、設定する保護グループが 1 つだけの場合でも使用します。
ステップ 2	**channel 0 local sonet** *interface* または **channel 0 remote** *ip-address* **例：** `RP/0/RSP0/CPU0:router(config-aps)# channel 0 local SONET 0/0/0/1` または `RP/0/RSP0/CPU0:router(config-aps)# channel 0 remote 172.18.69.123`	APS グループの保護チャネルを作成します。**0** は保護チャネルを表します。 **（注）** アクティブ チャネルを割り当てる前に、保護チャネルを割り当てる必要があります。 **（注）** チャネルが両方とも 1 つのルータにある APS を設定するには、保護チャネルとアクティブ チャネルの両方に **channel local** コマンドを使用します。 アクティブ チャネルが 1 つのルータにあり、保護チャネルが別のルータにある異なる 2 つのルータを使用する APS を設定するには、**channel local** コマンドを保護またはアクティブ チャネルのいずれかに使用し、もう一方のチャネルに **channel remote** コマンドを使用します。
ステップ 3	**channel 1 local sonet** *interface* または **channel 1 remote** *ip-address* **例：** `RP/0/RSP0/CPU0:router(config-aps)# channel 1 local SONET 0/0/0/2` または `RP/0/0/CPU0:router(config-aps)# channel 1 remote 172.18.69.123`	APS グループのアクティブ チャネルを作成します。**1** はアクティブ チャネルを表します。 **（注）** アクティブ チャネルの割り当ては、保護チャネルが割り当てられてから行う必要があります。 **（注）** チャネルが両方とも 1 つのルータにある APS を設定するには、保護チャネルとアクティブ チャネルの両方に **channel local** コマンドを使用します。 アクティブ チャネルが 1 つのルータにあり、保護チャネルが別のルータにある異なる 2 つのルータを使用する APS を設定するには、**channel local** コマンドを保護またはアクティブ チャネルのいずれかに使用し、もう一方のチャネルに **channel remote** コマンドを使用します。
ステップ 4	**signalling** {**sonet** | **sdh**} **例：** `RP/0/RSP0/CPU0:router(config-aps)# signalling sonet`	自動保護スイッチング（APS）で使用される K1K2 オーバーヘッド バイトを設定します。使用可能なキーワードを次に示します。 • **sonet**：シグナリングを SONET に設定します。 • **sdh**：シグナリングを同期デジタル ハイアラーキ（SDH）に設定します。

	コマンドまたはアクション	目的
ステップ 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-sonet)# end` または `RP/0/RSP0/CPU0:router(config-sonet)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 6	**show aps** **例：** `RP/0/RSP0/CPU0:router# show aps`	（任意）設定済みのすべての SONET APS グループの動作ステータスを表示します。
ステップ 7	**show aps group** [*number*] **例：** `RP/0/RSP0/CPU0:router# show aps group 3`	（任意）設定済みの SONET APS グループの動作ステータスを表示します。 **（注）** 複数のグループを定義する場合は、**show aps group** コマンドのほうが **show aps** コマンドよりも有用です。

SDH AU-3 の設定

ここでは、次の作業について説明します。

- 「C11-T1 または C12-E1 にマッピングされる SDH AU-3 の設定」（P.349）
- 「T3 または E3 にマッピングされる SDH AU-3 の設定」（P.353）

C11-T1 または C12-E1 にマッピングされる SDH AU-3 の設定

ここでは、c11-t1 または c12-e1 にマッピングされる SDH AU-3 を設定する方法について説明します。

前提条件

- *「Cisco ASR 9000 シリーズ ルータ でのクリア チャネル SONET コントローラの設定」*モジュールの「クリア チャネル SONET コントローラの設定方法」に示す SONET コントローラの設定方法を理解している必要があります。

制約事項

c11-t1 または c12-e1 にマッピングされるチャネライズド SDH AU-3 は、次の SPA でサポートされます。

- Cisco 1 ポート チャネライズド OC-3/STM-1 SPA
- Cisco 2 ポート チャネライズド OC-12c/DS0 SPA

手順の概要

1. **configure**
2. **controller sonet** *interface-path-id*
3. **clock source** {**internal** | **line**}
4. **framing sdh**
5. **au** *number*
6. **mode** *mode*
7. **root**
8. **controller t1** *interface-path-id*
9. **channel-group** *number*
10. **timeslots** *num1:num2:num3:num4* または
    timeslots *range1-range2*
11. **show configuration**
12. **root**
13. **interface serial** *interface-path-id*
14. **encapsulation** {**frame-relay** | **hdlc** | **ppp**}
15. **ipv4** *ip-address mask*
16. **no shutdown**
17. **end**
    または
    commit
18. **show controllers sonet** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller sonet** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller sonet 0/1/1/0`	SONET コントローラ コンフィギュレーション サブモードを開始して、SONET コントローラ名と interface-path-id の ID を *rack/slot/module/port* 表記で指定します。

	コマンドまたはアクション	目的
ステップ 3	**clock source** {**internal** | **line**} **例：** `RP/0/RSP0/CPU0:router(config-sonet)# clock source internal`	SONET ポート転送クロック ソースを設定します。ここで、**internal** キーワードは内部クロック、**line** キーワードは回線から回収されたクロックを設定します。 • ネットワークからクロッキングを得られる場合は、必ず **line** キーワードを使用します。2 つのルータがバックツーバックまたは光ファイバで接続されており、クロッキングが得られない場合は、**internal** キーワードを使用します。 • デフォルト キーワードは **line** です。 **（注）** スペース再利用プロトコル（SRP）インターフェイスでは、内部クロッキングが必要です。
ステップ 4	**framing sdh** **例：** `RP/0/RSP0/CPU0:router(config-sonet)# framing sdh`	同期デジタル階層（SDH）フレーミングのコントローラ フレーミングを設定します。 SONET フレーム構成（**sonet**）がデフォルトです。
ステップ 5	**au** *number* **例：** `RP/0/RSP0/CPU0:router(config-sonet)# au 1`	管理ユニット（AU）グループを指定し、AU パス コンフィギュレーション モードを開始します。AU-3 の有効範囲は、次のとおりです。 • 1 ～ 3 : 1 ポート チャネライズド OC-3/STM-1 SPA • 1 ～ 12 : 2 ポート チャネライズド OC-12c/DS0 SPA **（注）** **au** コマンドは AU タイプを指定しません。これは、設定する AU タイプの AU グループの番号を指定するものです。AU コマンドの範囲は、AU-3 と AU-4 のどちらを設定するかによって異なります。
ステップ 6	**mode** *mode* **例：** `RP/0/RSP0/CPU0:router(config-auPath)# mode c11-t1`	AU レベルでのインターフェイスのモードを設定します。AU-3 パスは、サポートされている SPA で c11-t1 または c12-e1 にマッピングできます。
ステップ 7	**root** **例：** `RP/0/RSP0/CPU0:router(config-auPath)# root`	グローバル コンフィギュレーション モードに戻ります。
ステップ 8	**controller t1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller T1 0/1/1/0/0/0/0`	T1 コントローラ コンフィギュレーション サブモードを開始して、T1 コントローラ名と *interface-path-id* を *rack/slot/module/port/auNum/t1Num* 表記で指定します。
ステップ 9	**channel-group** *number* **例：** `RP/0/RSP0/CPU0:router(config-t1)# channel-group 0`	タイム スロットの割り当て先となるチャネル グループ番号を設定します。範囲は 1 ～ 28 です。

	コマンドまたはアクション	目的
ステップ 10	**timeslots** *num1:num2:num3:num4* または **timeslots** *range1-range2* **例：** `RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 1:3:7:9` `RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 1-12`	インターフェイスのタイム スロットを *num1:num2:num3:num4* 表記で数字で指定するか、*range1-range2* 表記で範囲として指定します。
ステップ 11	**show configuration** **例：** `RP/0/RSP0/CPU0:router(config-t1-channel_group)# show configuration`	コミットされていない設定の内容を表示します。
ステップ 12	**root** **例：** `RP/0/RSP0/CPU0:router(config-t3)# root`	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 13	**interface serial** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface serial 0/1/1/0/0/0:0`	完全なインターフェイス番号を *rack/slot/module/port/T3Num/T1num:instance* 表記で指定します。
ステップ 14	**encapsulation** {**frame-relay** | **hdlc** | **ppp**} **例：** `RP/0/RSP0/CPU0:router(config-if)# encapsulation frame-relay`	カプセル化のタイプを、次のいずれかのキーワードを使用して指定します。 • **frame-relay**：フレームリレー ネットワーク プロトコル • **hdlc**：ハイレベル データリンク コントロール（HDLC）同期プロトコル • **ppp**：ポイントツーポイント プロトコル
ステップ 15	**ipv4** *ip-address mask* **例：** `RP/0/RSP0/CPU0:router(config-if)# ip address 10.10.10.10 255.255.255.255`	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。
ステップ 16	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-if)# no shutdown`	shutdown 設定を削除します。 **（注）** shutdown 設定を削除することにより、インターフェイスでの強制的な管理上の停止が排除されるため、インターフェイスはアップ状態またはダウン状態に移行することができます（親 SONET レイヤが管理上の停止状態に設定されていないことを前提とします）。

	コマンドまたはアクション	目的
ステップ 17	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-sonet)# end` または `RP/0/RSP0/CPU0:router(config-sonet)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 18	**show controllers sonet** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show controllers sonet 0/1/1/0`	SONET コントローラの設定を確認します。

T3 または E3 にマッピングされる SDH AU-3 の設定

このタスクでは、T3 または E3 にマッピングされる SDH AU-3 の設定方法について説明します。

前提条件

- *「Cisco ASR 9000 シリーズ ルータ でのクリア チャネル SONET コントローラの設定」*モジュールの「クリア チャネル SONET コントローラの設定方法」に示す SONET コントローラの設定方法を理解している必要があります。

制約事項

T3 または E3 にマッピングされるチャネライズド SDH AU-3 は、次の SPA でサポートされます。

- Cisco 1 ポート チャネライズド OC-48/STM-16 SPA
- Cisco 1 ポート チャネライズド OC-3/STM-1 SPA
- Cisco 2 ポート チャネライズド OC-12c/DS0 SPA

手順の概要

1. **configure**
2. **controller sonet** *interface-path-id*
3. **clock source** {**internal** | **line**}

4. **framing sdh**
5. **au** *number*
6. **mode t3**
   または
   mode e3
7. **root**
8. **controller** {**t3** | **e3**} *interface-path-id*
9. **mode serial**
10. **show configuration**
11. **root**
12. **interface serial** *interface-path-id*
13. **encapsulation** {**frame-relay** | **hdlc** | **ppp**}
14. **ipv4** *ip-address mask*
15. **no shutdown**
16. **end**
    または
    commit
17. **show controllers sonet** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller sonet** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller sonet 0/1/1/0`	SONET コントローラ コンフィギュレーション サブモードを開始して、SONET コントローラ名と interface-path-id の ID を *rack/slot/module/port* 表記で指定します。
ステップ 3	**clock source** {**internal** | **line**} **例：** `RP/0/RSP0/CPU0:router(config-sonet)# clock source internal`	SONET ポート転送クロック ソースを設定します。ここで、**internal** キーワードは内部クロック、**line** キーワードは回線から回収されたクロックを設定します。 • ネットワークからクロッキングを得られる場合は、必ず **line** キーワードを使用します。2 つのルータがバックツーバックまたは光ファイバで接続されており、クロッキングが得られない場合は、**internal** キーワードを使用します。 • デフォルト キーワードは **line** です。 **（注）** スペース再利用プロトコル（SRP）インターフェイスでは、内部クロッキングが必要です。

	コマンドまたはアクション	目的
ステップ 4	**framing sdh** **例：** `RP/0/RSP0/CPU0:router(config-sonet)# framing sdh`	同期デジタル階層（SDH）フレーミングのコントローラ フレーミングを設定します。 SONET フレーム構成（**sonet**）がデフォルトです。
ステップ 5	**au** *number* **例：** `RP/0/RSP0/CPU0:router(config-sonet)# au 1`	管理ユニット（AU）グループを指定し、AU パス コンフィギュレーション モードを開始します。AU-3 の有効範囲は、次のとおりです。 • 1 ～ 48 : 1 ポート チャネライズド OC-48/DS3 SPA • 1 ～ 3 : 1 ポート チャネライズド OC-3/STM-1 SPA • 1 ～ 12 : 2 ポート チャネライズド OC-12c/DS0 SPA **（注）** **au** コマンドは AU タイプを指定しません。これは、設定する AU タイプの AU グループの番号を指定するものです。AU コマンドの範囲は、AU-3 と AU-4 のどちらを設定するかによって異なります。
ステップ 6	**mode t3** または **mode e3** **例：** `RP/0/RSP0/CPU0:router(config-auPath)# mode t3`	AU レベルでのインターフェイスのモードを T3 または E3 に設定します。
ステップ 7	**root** **例：** `RP/0/RSP0/CPU0:router(config-auPath)# root`	グローバル コンフィギュレーション モードに戻ります。
ステップ 8	**controller** {**t3** | **e3**} *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller T3 0/1/1/0/0`	T3 または E3 コントローラ コンフィギュレーション サブモードを開始して、T3 または E3 コントローラ名と *interface-path-id* を *rack/slot/module/port/auNum* 表記で指定します。
ステップ 9	**mode serial** **例：** `RP/0/RSP0/CPU0:router(config-t3)# mode serial`	ポートのモードをクリア チャネル シリアルに設定します。
ステップ 10	**show configuration** **例：** `RP/0/RSP0/CPU0:router(config-t3)# show configuration`	コミットされていない設定の内容を表示します。
ステップ 11	**root** **例：** `RP/0/RSP0/CPU0:router(config-t3)# root`	グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
ステップ 12	**interface serial** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface serial 0/1/1/0/0/0:0`	完全なインターフェイス番号を *rack/slot/module/port/T3Num/T1num:instance* 表記で指定します。
ステップ 13	**encapsulation frame-relay** | **hdlc** | **ppp** **例：** `RP/0/RSP0/CPU0:router(config-if)# encapsulation frame-relay | hdlc | ppp`	カプセル化のタイプを、次のいずれかのキーワードを使用して指定します。 • **frame-relay**：フレームリレー ネットワーク プロトコル • **hdlc**：ハイレベル データリンク コントロール（HDLC）同期プロトコル • **ppp**：ポイントツーポイント プロトコル
ステップ 14	**ipv4** *ip-address mask* **例：** `RP/0/RSP0/CPU0:router(config-if)# ip address 10.10.10.10 255.255.255.255`	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。
ステップ 15	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-if)# no shutdown`	shutdown 設定を削除します。 **（注）** shutdown 設定を削除することにより、インターフェイスでの強制的な管理上の停止が排除されるため、インターフェイスはアップ状態またはダウン状態に移行することができます（親 SONET レイヤが管理上の停止状態に設定されていないことを前提とします）。
ステップ 16	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-sonet)# end` または `RP/0/RSP0/CPU0:router(config-sonet)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 17	**show controllers sonet** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show controllers sonet 0/1/1/0`	SONET コントローラの設定を確認します。

SDH AU-4 の設定

ここでは、SDH AU-4 ストリームを E3 にマッピングされた TUG-3 チャネルに設定する手順について説明します。

前提条件

- *「Cisco ASR 9000 シリーズ ルータ でのクリア チャネル SONET コントローラの設定」*モジュールの「クリア チャネル SONET コントローラの設定方法」に示す SONET コントローラの設定方法を理解している必要があります。

制約事項

- チャネライズド SDH は、次の SPA でサポートされます。
  - Cisco 1 ポート チャネライズド OC-48/STM-16 SPA
  - Cisco 1 ポート チャネライズド OC-3/STM-1 SPA
  - Cisco 2 ポート チャネライズド OC-12/DS0 SPA
- このリリースでは、AU-4 パスのチャネル化は TUG-3 のみが可能です。
- 1 ポート チャネライズド OC-48/STM-16 SPA は T1 または E1 のチャネル化をサポートしません。

手順の概要

1. **configure**
2. **controller sonet** *interface-path-id*
3. **clock source** {**internal** | **line**}
4. **framing sdh**
5. **au** *number*
6. **mode tug3**
7. **width** *number*
8. **tug3** *number*
9. **mode** *mode*
10. **root**
11. **controller** *name interface-path-id*
12. **mode** *mode*
13. **root**
14. **controller** *name instance*
15. **channel-group** *number*
16. **timeslots** *num1:num2:num3:num4* または
    timeslots *range1-range2*
17. **show configuration**
18. **root**
19. **interface serial** *interface-path-id*

20. **encapsulation** {**frame-relay** | **hdlc** | **ppp**}
21. **ipv4** *ip-address mask*
22. **no shutdown**
23. **end**
    または
    commit
24. **show controllers sonet** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/0/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	**controller sonet** *interface-path-id* **例：** `RP/0/0/CPU0:router(config)# controller sonet 0/1/1/0`	SONET コントローラ コンフィギュレーション サブモードを開始して、SONET コントローラ名と *interface-path-id* を *rack/slot/module/port* 表記で指定します。
ステップ 3	**clock source** {**internal** | **line**} **例：** `RP/0/0/CPU0:router(config-sonet)# clock source internal`	SONET ポート転送クロック ソースを設定します。ここで、**internal** キーワードは内部クロック、**line** キーワードは回線から回収されたクロックを設定します。 • ネットワークからクロッキングを得られる場合は、必ず **line** キーワードを使用します。2 つのルータがバックツーバックまたは光ファイバで接続されており、クロッキングが得られない場合は、**internal** キーワードを使用します。 • デフォルト キーワードは **line** です。 **（注）** スペース再利用プロトコル（SRP）インターフェイスでは、内部クロッキングが必要です。
ステップ 4	**framing sdh** **例：** `RP/0/0/CPU0:router(config-sonet)# framing sdh`	同期デジタル階層（SDH）のコントローラを設定します。 SONET フレーム構成（**sonet**）がデフォルトです。
ステップ 5	**au** *number* **例：** `RP/0/RSP0/CPU0:router(config-sonet)# au 1`	管理ユニット（AU）グループを指定し、AU パス コンフィギュレーション モードを開始します。AU-4 の場合の有効範囲は、次のとおりです。 • 1 ～ 16 : 1 ポート チャネライズド OC-48/DS3 SPA • 1 ～ 3 : 1 ポート チャネライズド OC-3/STM-1 SPA • 1 ～ 4 : 2 ポート チャネライズド OC-12c/DS0 SPA **（注）** **au** コマンドは AU タイプを指定しません。これは、設定する AU タイプの AU グループの番号を指定するものです。AU コマンドの範囲は、AU-3 と AU-4 のどちらを設定するかによって異なります。

	コマンドまたはアクション	目的
ステップ 6	**mode tug3** **例：** `RP/0/0/CPU0:router(config-auPath)# mode tug3`	AU レベルでのインターフェイスのモードを設定します。現在サポートされているのは TUG3 のみです。
ステップ 7	**width** *number* **例：** `RP/0/0/CPU0:router(config-auPath)# width 3`	AU ストリーム数を設定します。
ステップ 8	**tug3** *number* **例：** `RP/0/0/CPU0:router(config-auPath)#tug3 1`	トリビュタリ ユニット グループ（TUG）の *number* を指定して、config-tug3Path モードを開始します。範囲は 1 ～ 3 です。
ステップ 9	**mode** *mode* **例：** `RP/0/0/CPU0:router(config-tug3Path)# mode e3`	tug3 レベルでのインターフェイスのモードを設定します。使用可能なモードを次に示します。 • **c11**：TU-11 を伝送する TUG-3 パス • **c11-t1**：TU-11 から T1 を伝送する TUG-3 パス • **c12**：TU-12 を伝送する TUG-3 パス • **c12-e1**：TU-12 から E1 を伝送する TUG-3 パス • **e3**：E3 を伝送する TUG-3 パス • **t3**：T3 を伝送する TUG-3 パス **（注）** 1 ポート チャネライズド OC-48/STM-16 SPA は **e3** および **t3** オプションだけをサポートします。
ステップ 10	**root** **例：** `RP/0/0/CPU0:router(config-tug3Path)# root`	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 11	**controller** *name instance* **例：** `RP/0/0/CPU0:router(config)# controller e3 0/1/1/0/0/0`	コントローラ コンフィギュレーション サブモードを開始して、コントローラ名とインスタンス ID を *rack/slot/module/port/name/instance* 表記で指定します。コントローラ名を次に示します。 • **e3**：E3 を伝送する TUG3 パス • **t3**：T3 を伝送する TUG3 パス • **e1**：チャネライズド E1 ポート **（注）** この手順では、E3 または T3 コントローラを作成して T3 コントローラの下に手順 14 に示すように T1 チャネルを追加するか、またはこの時点でチャネライズド E1 ポートを作成することができます。 **（注）** 1 ポート チャネライズド OC-48/STM-16 SPA では、E1 はサポートされません。

	コマンドまたはアクション	目的
ステップ 12	**mode** *mode* **例：** `RP/0/0/CPU0:router(config-e3)#mode e1`	インターフェイスのモードを設定します。使用可能なモードを次に示します。 • **e1**：21 個の E1 にチャネル化 • **serial**：HDLC に類似するペイロードを伝送するクリア チャネル • **t1**：28 個の T1 にチャネル化 **（注）** T1 および E1 は、1 ポート チャネライズド OC-48/STM-16 SPA ではサポートされません。
ステップ 13	**root** **例：** `RP/0/0/CPU0:router(config-e3)# root`	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 14	**controller** *name instance* **例：** `RP/0/0/CPU0:router(config)# controller E1 0/1/1/0/0/0/0/0`	コントローラ コンフィギュレーション サブモードを開始して、コントローラ名とインスタンス ID を *rack/slot/module/port/name/instance1/instance2* 表記で指定します。コントローラ名を次に示します。 • serial：HDLC に類似するペイロードを伝送するクリア チャネル。 • t1：24 個の T1 にチャネル化。
ステップ 15	**channel-group** *number* **例：** `RP/0/0/CPU0:router(config-e1)# channel-group 0`	タイム スロットの割り当て先となるチャネル グループ番号を設定します。 • t1 の場合、有効値の範囲は 1 ～ 24 です。 • e1 の場合、有効値の範囲は 1 ～ 32 です。
ステップ 16	**timeslots** *num1:num2:num3:num4* または **timeslots** *range1-range2* **例：** `RP/0/0/CPU0:router(config-e1-channel_group)# timeslots 1:3:7:9` `RP/0/0/CPU0:router(config-e1-channel_group)# timeslots 1-12`	インターフェイスのタイム スロットを *num1:num2:num3:num4* 表記で数字で指定するか、*range1-range2* 表記で範囲として指定します。
ステップ 17	**show configuration** **例：** `RP/0/0/CPU0:router(config-e1-channel_group)# show configuration`	コミットされていない設定の内容を表示します。
ステップ 18	**root** **例：** `RP/0/0/CPU0:router(config-e1-channel_group)# root`	グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
ステップ 19	**interface serial** *interface-path-id* **例：** `RP/0/0/CPU0:router(config)# interface serial 0/1/1/0/0/0:0`	完全なインターフェイス番号を *rack/slot/module/port/T3Num/T1num:instance* 表記で指定します。
ステップ 20	**encapsulation** {**frame-relay** | **hdlc** | **ppp**} **例：** `Router(config-if)# encapsulation frame-relay | hdlc | ppp`	カプセル化のタイプを、次のいずれかのキーワードを使用して指定します。 • **frame-relay**：フレームリレー ネットワーク プロトコル • **hdlc**：ハイレベル データリンク コントロール（HDLC）同期プロトコル • **ppp**：ポイントツーポイント プロトコル
ステップ 21	**ipv4** *ip-address mask* **例：** `Router(config-if)# ip address 10.10.10.10 255.255.255.255`	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。
ステップ 22	**no shutdown** **例：** `RP/0/0/CPU0:router (config-if)# no shutdown`	shutdown 設定を削除します。 **（注）** shutdown 設定を削除することにより、インターフェイスでの強制的な管理上の停止が排除されるため、インターフェイスはアップ状態またはダウン状態に移行することができます（親 SONET レイヤが管理上の停止状態に設定されていないことを前提とします）。
ステップ 23	**end** または **commit** **例：** `RP/0/0/CPU0:router(config-sonet)# end` または `RP/0/0/CPU0:router(config-sonet)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 24	**show controllers sonet** *interface-path-id* **例：** `RP/0/0/CPU0:router# show controllers sonet 0/1/1/0`	SONET コントローラの設定を確認します。

チャネライズド SONET の設定例

ここでは、次の例を示します。

チャネライズド SONET の例

チャネライズド SONET T3 から T1 への設定：例

次に、SONET T3 から T1 への設定例を示します。

```
configure
 controller sonet 0/1/1/0
  clock source internal
  framing sonet
  sts 1
  mode t3
  width 3
  root
 controller t3 0/1/1/0/0
  mode t1
  root
 controller t1 0/1/1/0/0/0
  framing esf
  channel-group 0
   timeslots 1:3:7:9
   show configuration
  root
 interface serial 0/1/1/0/0/0:0
 encapsulation hdlc
 ip address 10.10.10.10 255.255.255.255
 no shutdown
  commit
show controllers sonet 0/1/1/0
```

VT1.5 モードでのチャネライズド SONET と T1 の NxDS0 へのチャネル化

（注） この例は、1 ポート チャネライズド OC-48/STM-16 SPA ではサポートされません。

次の例では、NxDS0 への SONET チャネル化を、SONET VT1.5 モードで行う方法を説明します。

```
configure
 controller sonet 0/1/1/0
  clock source internal
  framing sonet
  sts 1
```

```
  mode vt15-t1
  root
controller t1 0/1/1/0/0/0
  channel-group 0 timeslots 1
  channel-group 1 timeslots 2-3
  commit
```

チャネライズド Packet over SONET の設定：例

次に、チャネライズド Packet over SONET の設定例を示します。

```
configure
 controller sonet 0/1/1/0
  clock source internal
  framing sonet
  sts 1
  mode pos scramble
  width 3
  root
 interface POS 0/1/1/0
  encapsulation hdlc
  pos crc 32
  mtu 4474
  no shutdown
  commit
show interfaces pos 0/1/1/0
```

SONET クリア チャネル T3 の設定：例

次に、SONET クリア チャネルを T3 に設定する例を示します。

```
configure
 controller sonet 0/1/1/0
  clock source internal
  framing sonet
  sts 1
  mode t3
  root
 controller t3 0/1/1/0/0
  mode serial
  root
 interface serial 0/1/1/0/0/0:0
 encapsulation ppp
 ip address 10.10.10.10 255.255.255.255
 no shutdown
  commit
show controllers sonet 0/1/1/0
```

チャネライズド SONET APS マルチルータの設定：例

次に、SONET APS マルチルータの設定例を示します。

```
 aps group 1
  channel 0 local SONET 0/0/0/1
  channel 1 remote 172.18.69.123
  signalling sonet
  commit
show aps
show aps group 3
```

チャネライズド SDH の例

チャネライズド SDH AU-3 の設定：例

ここでは、次の設定例を示します。

チャネライズド SDH AU-3 から VC-3 およびクリア チャネル T3/E3：例

次に、SDH AU-3 から VC-3 およびクリア チャネル T3 を設定する例を示します。

```
configure
 controller sonet 0/1/1/0
  clock source internal
  framing sdh
  au 1
  width 1
  mode t3
  root
 controller t3 0/1/1/0/1
  mode serial
  commit
```

次に、SDH AU-3 から VC-3 およびクリア チャネル E3 を設定する例を示します。

```
configure
 controller sonet 0/1/1/0
  clock source internal
  framing sdh
  au 1
  width 1
  mode e3
  root
 controller e3 0/1/1/0/1
  mode serial
  commit
```

チャネライズド SDH AU-3 から TUG-2、VC-11、T1 および NxDS0：例

（注） この例は、1 ポート チャネライズド OC-48/STM-16 SPA ではサポートされません。

次に、SDH AU-3 から TUG-2、VC-11 およびチャネライズド T1 から NxDS0 を設定する例を示します。

```
configure
 controller sonet 0/1/1/0
  clock source internal
  framing sdh
  au 1
```

```
  mode c11-t1
  width 1
  root
 controller T1 0/1/1/0/0/1/1
  channel-group 0
   timeslots 1-12
   show configuration
  root
 interface serial 0/1/1/0/1/1:0
 encapsulation ppp
 ip address 10.10.10.10 255.255.255.255
 no shutdown
  commit
show controllers sonet 0/1/1/0
```

チャネライズド SDH AU-3 から TUG-2、VC-12、E1 および NxDS0：例

（注） この例は、1 ポート チャネライズド OC-48/STM-16 SPA ではサポートされません。

次に、SDH AU-3 から TUG-2、VC-12 およびチャネライズド E1 から NxDS0 を設定する例を示します。

```
configure
 controller sonet 0/1/1/0
  clock source internal
  framing sdh
  au 1
  mode c12-e1
  width 1
  root
 controller e1 0/1/1/0/0/1/1
  channel-group 0
   timeslots 1-12
   show configuration
  root
 interface serial 0/1/1/0/1/1:0
 encapsulation ppp
 ip address 10.10.10.10 255.255.255.255
 no shutdown
  commit
show controllers sonet 0/1/1/0
```

チャネライズド SDH AU-4 の設定：例

ここでは、次の設定例を示します。

- 「チャネライズド SDH AU-4 から TUG-3 およびクリア チャネル T3/E3：例」（P.365）
- 「チャネライズド SDH AU-4 から TUG-3、TUG-2、ならびに T1/E1 および NxDS0：例」（P.366）

チャネライズド SDH AU-4 から TUG-3 およびクリア チャネル T3/E3：例

次の例は、SDH AU-4 から TUG-3 へのチャネル化およびクリア チャネル T3 を示します。

```
configure
 controller sonet 0/4/0/0
  framing sdh
  au 1
  width 3
```

```
  mode tug3
  tug3 1
   mode t3
   root
 controller t3 0/4/0/0/1/1
  mode serial
  commit
```

次の例は、SDH AU-4 から TUG-3 へのチャネル化およびクリア チャネル E3 を示します。

```
configure
 controller sonet 0/4/0/0
  framing sdh
  au 1
  width 3
  mode tug3
  tug3 1
   mode e3
   root
 controller e3 0/4/0/0/1/1
  mode serial
  commit
```

チャネライズド SDH AU-4 から TUG-3、TUG-2、ならびに T1/E1 および NxDS0：例

（注） T1/E1 および NxDS0 へのチャネル化は、1 ポート チャネライズド OC-48/STM-16 SPA ではサポートされません。

次の例に示す SDH AU-4 の設定では、非フレーム化 E1 コントローラとシリアル インターフェイスが指定されています。

```
configure
 controller sonet 0/1/2/0
  framing sdh
  au 1
  width 3
  mode tug3
  tug3 1
   mode c12-e1
!
  tug3 2
   mode c12-e1
!
  tug3 3
   mode c12-e1
!
controller E1 0/1/2/0/1/1/1/1
framing unframed
!
controller E1 0/1/2/0/1/1/1/2
framing unframed
!
controller E1 0/1/2/0/1/1/1/3
framing unframed
!
interface Serial0/1/2/0/1/1/1/1:0
encapsulation ppp
multilink
  group 1
!
interface Serial0/1/2/0/1/1/1/2:0
```

```
encapsulation ppp
multilink
  group 1
!
!
interface Serial0/1/2/0/1/1/1/3:0
encapsulation ppp
multilink
  group 1
!
```

次に、E1 コントローラ チャネル グループおよびシリアル インターフェイスを使用する SDH AU-4 の設定の例を示します。

```
configure
 controller SONET0/3/2/0
   framing sdh
   au 1
   width 3
   mode tug3
   tug3 1
   mode c12-e1
!
   tug3 2
   mode c12-e1
!
   tug3 3
   mode c12-e1
!
controller E1 0/3/2/0/1/1/1/1
 framing crc4
 channel-group 0
  timeslots 1-4
!
controller E1 0/3/2/0/1/1/3/1
 framing crc4
 channel-group 0
  timeslots 1-31
!
controller E1 0/3/2/0/1/1/1/2
 framing crc4
 channel-group 0
  timeslots 1-31
!
controller E1 0/3/2/0/1/2/7/3
 framing crc4
 channel-group 0
  timeslots 1-5
!
 channel-group 1
  timeslots 6-31
!
interface Serial0/3/2/0/1/1/1/1:0
 encapsulation frame-relay IETF
 frame-relay lmi-type ansi
 frame-relay intf-type dce
!
interface Serial0/3/2/0/1/1/1/1:0.1 point-to-point
 ipv4 address 192.168.200.2 255.255.255.252
 ipv4 verify unicast source reachable-via rx
 pvc 100
  encap ietf
!
interface Serial0/3/2/0/1/1/3/1:0
 encapsulation ppp
```

```
 multilink
  group 1
!
interface Serial0/3/2/0/1/1/1/2:0
 encapsulation ppp
 multilink
  group 1
```

その他の関連資料

ここでは、チャネライズド SONET の設定に関する参考資料について説明します。

関連資料

関連項目	参照先
Cisco IOS XR マスター コマンド リファレンス	*『Cisco IOS XR Master Commands List』*
Cisco IOS XR インターフェイス コンフィギュレーション コマンド	*『Cisco IOS XR Interface and Hardware Component Command Reference』*
Cisco IOS XR ソフトウェアを使用するルータを初回に起動し設定するための情報	*『Cisco IOS XR Getting Started Guide』*
ユーザ グループとタスク ID に関する情報	『*Cisco IOS XR System Security Configuration Guide*』の「*Configuring AAA Services on Cisco IOS XR* Software」モジュール

標準

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

MIB

MIB	MIB のリンク
• CISCO-SONET-MIB • ENTITY-MIB • SONET-MIB（RFC 3592） 次の追加の MIB は、Cisco ASR 9000 シリーズ ルータの Cisco 1 ポート チャネライズド OC-3/STM-1 SPA および Cisco 2 ポート チャネライズド OC-12c/DS0 SPA でサポートされます。 • CISCO-IF-EXTENSION-MIB • DS1-MIB • DS3-MIB • IF-MIB	Cisco IOS XR ソフトウェアを使用して選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	—

シスコのテクニカル サポート

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

その他の関連資料

Cisco ASR 9000 シリーズ ルータでの Circuit Emulation over Packet の設定

このモジュールでは、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータでの Circuit Emulation over Packet（CEoP）共有ポート アダプタ（SPA）の設定について説明します。

Cisco ASR 9000 シリーズ ルータでの CEoP 設定の機能履歴

リリース	変更内容
リリース 4.2.0	• Circuit Emulation Service over Packet Switched Network のサポートが次の SPA で追加されました。 – Cisco 1 ポート チャネライズド OC3/STM-1 SPA（SPA-1CHOC3-CE-ATM）
リリース 4.3.0	• Circuit Emulation Service over Packet Switched Network のサポートが次の SPA で追加されました。 – Cisco 24 ポート チャネライズド T1/E1 回線エミュレーションおよびチャネライズド ATM SPA（SPA-24CHT1-CE-ATM） – Cisco 2 ポート チャネライズド T3/E3 回線エミュレーションおよびチャネライズド ATM SPA（SPA-2CHT3-CE-ATM）

内容

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。

Circuit Emulation over Packet（CEoP）サービスをルータ上で設定する前に、次の条件を満たしていることを確認してください。

- 次の SPA の 1 つがシャーシに設置されている必要があります。
  - Cisco 2 ポート チャネライズド T3/E3 回線エミュレーションおよびチャネライズド ATM SPA
  - Cisco 24 ポート チャネライズド T1/E1 回線エミュレーションおよびチャネライズド ATM SPA
  - Cisco 1 ポート チャネライズド OC3/STM-1 回線エミュレーションおよび ATM SPA
- 汎用表記 *rack/slot/module/port* を使用して SONET コントローラ名と *interface-path-id* を適用 / 指定する方法を理解している必要があります。SONET コントローラ名と *interface-path-id* は、**controller sonet** コマンドで必要となります。
- T3/E3 および T1/E1 コントローラ名および *interface-path-id* を、汎用表記 *rack/slot/module/port* を使用して適用および指定する方法を理解している必要があります。T3/E3、T1/E1 コントローラ名と *interface-path-id* は、**controller {T3|E3|T1|E1}** コマンドに必要です。

Circuit Emulation over Packet サービスの概要

Circuit Emulation over Packet（CEoP）は、TDM 回線をパケット スイッチド ネットワーク上で伝送する方法の 1 つです。Circuit Emulation over Packet は、物理接続の模倣です。CEoP の目標は、専用回線およびレガシー TDM ネットワークを置き換えることです。この機能によって、ネットワーク管理者は既存の IP または MPLS ネットワークを使用して専用回線エミュレーション サービスを提供できるようになります。また、他のマルチサービス プラットフォーム インターフェイスの形式の要件を満たさないデータ ストリームやプロトコルを伝送できるようになります。CEoP によって、TDM ビットがパケットに変換され、適切なヘッダーにカプセル化されてから、PSN を通して送信されます。CEoP の受信側では、TDM ビット ストリームがパケットから復元されます。

CEoP SPA はハーフハイト（HH）共有ポート アダプタ（SPA）であり、CEoP SPA ファミリを構成するものとしては、24xT1/E1/J1、2xT3/E3、および 1xOC3/STM1 非構造化 / 構造化（NxDS0）1/4 レート、ハーフハイト SPA があります。CEoP SPA が実行するビット透過データ転送は、完全にプロトコル非依存です。

CEoP には、次の 2 つのモードがあります。

- 非構造化モード : SAToP（Structure Agnostic TDM over Packet）と呼ばれます。SAToP では、着信データはその内容にかかわらず、純粋なビット ストリームと見なされます。
- 構造化モード : CESoPSN（Circuit Emulation Service over Packet Switched Network）と呼ばれます。CESoPSN では、着信 TDM ビット ストリームの構造が DS0 レベルで認識されます。

CESoPSN および SAToP では、MPLS、UDP/IP および L2TPv3 を転送メカニズムとして使用できます。

（注） Cisco IOS XR Release 4.2.x では、転送メカニズムとして MPLS だけがサポートされます。

これらの SPA は、Circuit Emulation Services over Packet Switched Network（CESoPSN）および Structure-Agnostic Transport over Packet（SAToP）の転送に関する新しい標準に適合するように設計された、シスコの最初のルータ インターフェイスです。SAToP モードでは、これらの SPA はデータの

形式や構造が事前定義されていないものとして動作します。データは単に、任意の内容のビット ストリームと見なされます。すべてのデータ ビットはそのまま、定義済みの宛先まで、IP/MPLS パケットとしてカプセル化された状態で転送されます。CESoPSN モードでは、キャリアが形式を定義します。SPA は、E1 および T1 フレーミングをすべてサポートします。CESoPSN アプリケーションに使用される帯域幅を節約するには、有効なタイム スロットのみを送信用に選択します。主な用途は、次のとおりです。

- 2G と 3G のネットワーク トラフィックを、パケット ネットワークを介して転送します（モバイル事業者向け）。モバイル サービス プロバイダーは、新たな収益創出サービスをサポートするために、HSDPA を使用する高速データ ネットワークを実装しています。SPA の独自の特色として、モバイル ネットワーク（2G および 3G）の多世代移行に対応していることや、TDM と ATM のトラフィックを同時に IP/MPLS ネットワーク上で伝送できることが挙げられます。このテクノロジーによるメカニズムを利用すると、セル サイトへの IP/MPLS が可能になります。これによって最終的に、モバイル トラフィックをエンドツーエンドで IP を介して転送できるようになる可能性があります。
- 専用回線を置き換えるための T3/E3 回線エミュレーション。
- 専用回線を置き換えるための T1/E1 回線エミュレーション。
- PBX 間を PSN 経由で接続する。
- IP/MPLS を経由する高密度 SS7 バックホール。
- Inter-MSC 接続。
- 政府や防衛などの高セキュリティ アプリケーションのための事前暗号化データ。
- 運輸や公共サービス（電気など）の産業で使用されている、独自の同期または非同期データ プロトコル。
- 都市圏（メトロ）イーサネットまたは WAN サービス プロバイダー環境における、専用回線エミュレーション サービス。

回線エミュレーション サービスの概念、設定および例については、『*Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide*』の「*Implementing Point to Point Layer 2 Services*」モジュールを参照してください。

CEoP チャネライズド SONET/SDH の設定に関する情報

Circuit Emulation over Packet チャネライズド SONET/SDH 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。

- 「チャネライズド SONET および SDH の概要」（P.373）
- 「チャネライズド SONET/SDH のデフォルト設定値」（P.378）

チャネライズド SONET および SDH の概要

同期光ファイバ ネットワーク（SONET）は、光ファイバでのデジタル テレコミュニケーション サービス伝送において使用される米国規格協会（ANSI）の規格形式です。

チャネライズド SONET では、多重化 T3/E3 および仮想トリビュタリ グループ（VTG）チャネルで SONET フレームを転送することができます。

SONET は、同期転送信号（STS）フレーム構成を使用します。STS は、オプティカル キャリア 1（OC-1）の電気版に相当します。

チャネライズド SONET インターフェイスは、複数の STS ストリームを複合したものであり、固有のペイロード ポインタを持つ独立したフレームとして維持されます。フレームは、転送される前に多重化されます。

回線がチャネル化されると、*パス*と呼ばれるより小さい帯域幅のチャネルに論理的に分割されます。これらのパスが SONET ペイロードを伝送します。全パスの帯域幅の合計は回線の帯域幅を超過できません。

回線がチャネル化されない場合、この回線は*クリア チャネル*と呼ばれ、回線の全帯域幅がブロードバンド サービスを伝送する単一のチャネル専用となります。

T3/E3 チャネルを T1 にチャネル化でき、T1 をさらに DS0 タイムスロットにチャネル化することができます。

SONET 回線のチャネル化は、次の 2 つの主要なプロセスで構成されます。

- コントローラの設定
- インターフェイスのチャネライズド パスへの設定

最初に、STS パスのモードを設定することによりコントローラを設定します。

モードが指定されると、各コントローラが作成され、残りの設定がそのコントローラに適用されます。たとえば、T3 モードでは T3 コントローラが作成されます。T3 コントローラは、シリアル チャネルに対して設定するか、または T1 を伝送するためにさらにチャネル化できます。これらの T1 は、シリアル インターフェイスに対して設定できます。

設置されている SPA のサポートに応じて、各 STS パスを個別に T3、E3、VTG などに設定できます。

次のレベルの SONET チャネライゼーション モードが CEoP SPA でサポートされます。

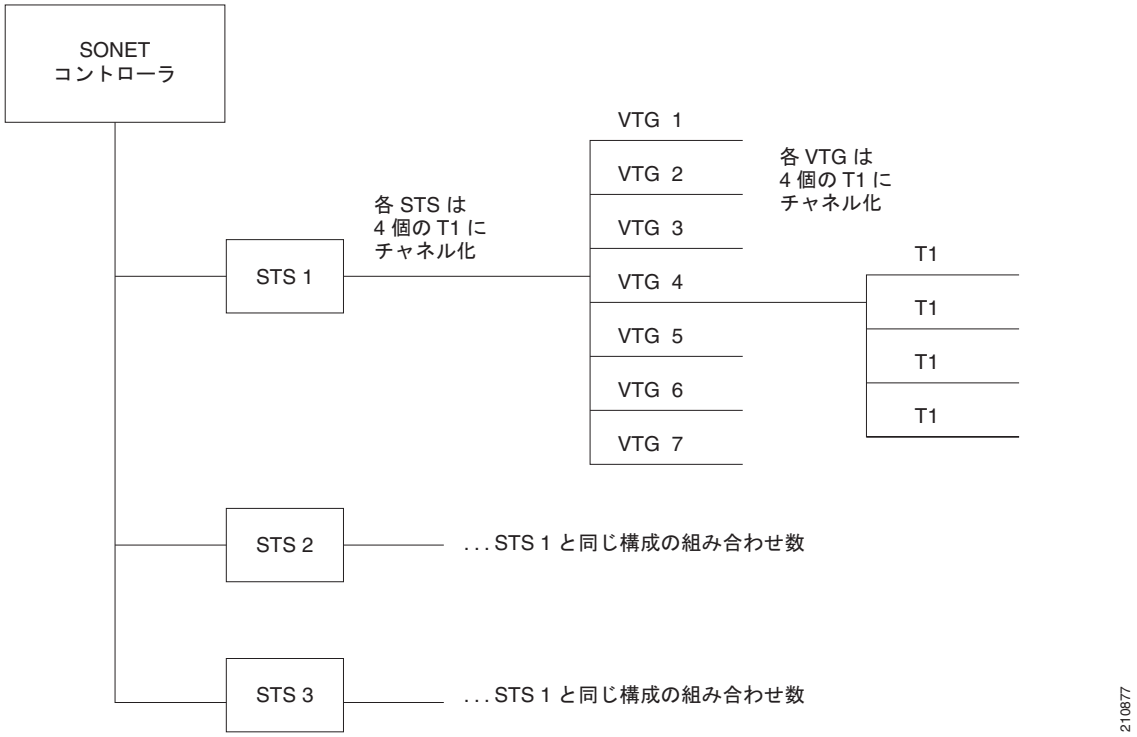
OC3->STS-1->VTG-> VT1.5 -> 非フレーム化 T1

OC3->STS-1->VTG-> VT1.5 -> T1 -> DS0

図 28 は、設定可能な VTG パスを示しています。

（注） Cisco ASR 9000 シリーズ ルータの Cisco 1 ポート チャネライズド OC-3/STM-1 SPA では、VTG パスのみがサポートされます。

図 28　　SONET VTG チャネライズド パス



同期デジタル ハイアラーキ（SDH）は、SONET の国際版に相当します。

SDH は、同期転送モード（STM）フレーム構成を使用します。1 つの STM-1 は、3 つのオプティカル キャリア 1（OC-1）の電気版に相当します。同期転送モジュール（STM）信号は、SONET の STS の同期デジタル ハイアラーキ（SDH）版に相当しますが、各帯域幅で番号は異なります。ここでは、STM という用語はパス幅と光回線レートの両方を表します。STM 信号内のパスは、管理ユニット（AU）と呼ばれます。

SONET と SDH 間での基本的な用語の違いの概要を次に示します。

- SONET の STS は、SDH の管理ユニット（AU）に相当
- SONET の仮想トリビュタリ（VT）は、SDH のトリビュタリ ユニット（TU）に相当
- SDH の基本ビルディング ブロックは STM-1（STS-3 に相当）および STM-0（STS-1 に相当）

管理ユニット（AU）は、より上位のパス レイヤと多重化セクション レイヤ間の適合を可能にする情報構造です。AU は、情報ペイロード（より上位の仮想コンテナ）と管理ユニット ポインタで構成されます。管理ユニット ポインタは、ペイロード フレーム開始のオフセットを多重化セクション フレーム開始と相対的に示します。

AU は、トリビュタリ ユニット（TU）およびトリビュタリ ユニット グループ（TUG）にチャネル化することができます。

管理ユニット 3（AU-3）は、1 つの STM-1 で構成されます。

管理ユニット グループ（AUG）は、STM ペイロードにおいて固定の定義された位置を占める 1 つまたは複数の管理ユニットで構成されます。

表 4 に、SONET 規格で一般的に使用される表記および用語と、SDH での対応する用語を示します。

表 4　SONET/SDH 用語対照表

SONET 用語	SDH 用語
SONET	SDH
STS-3c	AU-4
STS-1	AU-3
VT	TU
SPE	VC
セクション	リジェネレータ セクション
回線	多重化セクション
パス	パス

次のレベルの SDH チャネル化が CEoP SPA でサポートされます。

- E1 の場合：
  - STM1-> AU-4 -> TUG-3 -> TUG-2 ->VC12-> 非フレーム化 E1
  - STM1-> AU-4 -> TUG-3 -> TUG-2 ->VC12-> E1 -> DS0

- T1 の場合：
  - STM1-> AU-3-> TUG-2 -> VC11-> 非フレーム化 T1
  - STM1-> AU-3-> TUG-2 -> VC11->T1 -> DS0

図 29 に、CEoP SPA で設定可能な SDH AU-3 パスの例を示します。

図 29　　SDH AU3 パス

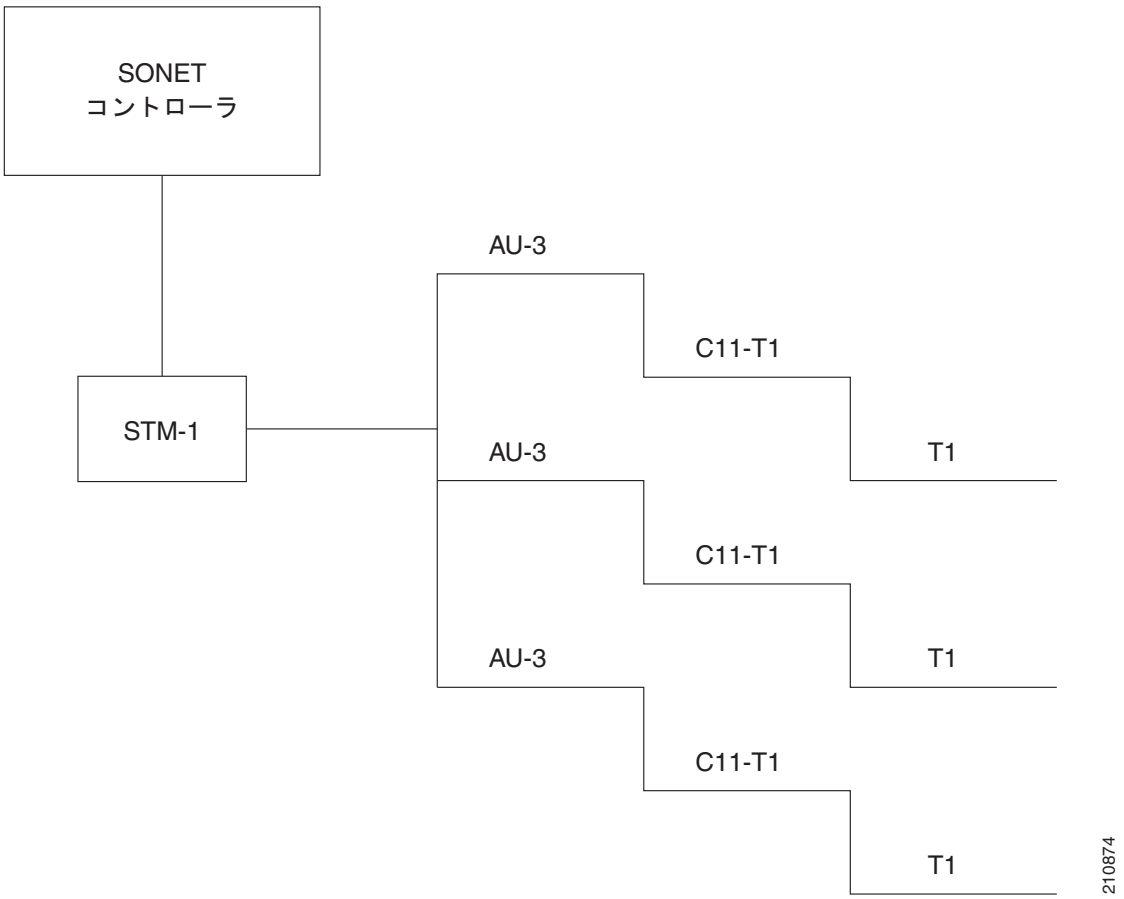
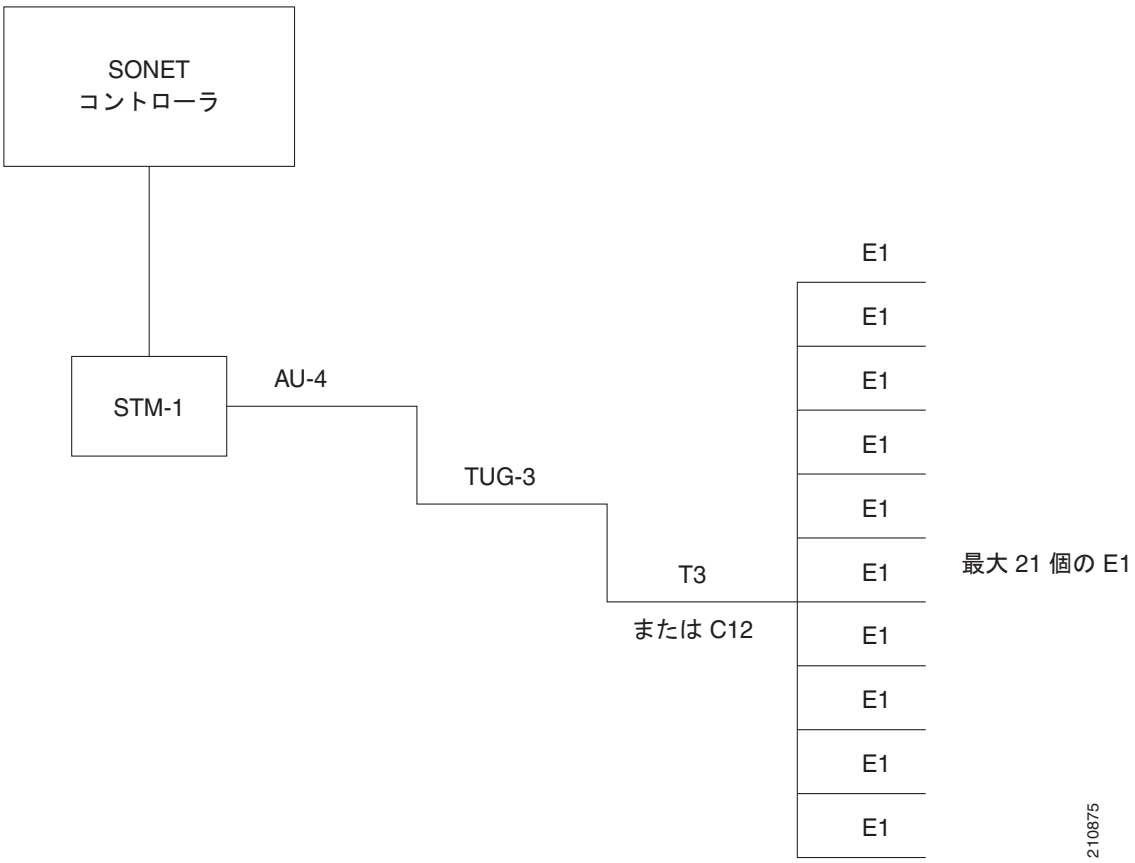


図 30 に、CEoP SPA で設定できる SDH AU4 パスを示します。

図 30　　SDH AU4 パス



チャネライズド SONET/SDH のデフォルト設定値

表 5 に、チャネライズド SONET/SDH に存在するデフォルト設定パラメータを示します。

表 5　　SONET/SDH コントローラのデフォルト設定値

パラメータ	デフォルト値	設定ファイルのエントリ
クロック ソース	line	**clock source** {**internal** | **line**}
SONET フレーミング	sonet	**hw-module sub-slot** *node-id* **cardtype** {**sonet** | **sdh**}

クロック配信

CEoP SPA でのクロック配信には、次の方法があります。

- 同期クロッキング：同期クロッキングによって、送信元と宛先の TDM 回線が同じクロックに合わせて同期化されます。このクロックは、何らかの物理的クロック配信手段で配信されたものです（SONET/SDH、BITS、GPS など）。特定の TDM 回線へのクロックは、次のものから渡すことができます。
  - 回線：送信クロックは、同じ物理回線の受信者からのものです。
  - 内部：送信クロックはラインカードから取得されます。内部の自励発振器から、または別の物理回線から導出できます。
  - 回復：送信クロックの導出に使用される、CEM インターフェイスでのインバンド疑似配線ベースのアクティブ クロック回復。

CEoP SPA に対して設定可能な再生クロックの数は次のとおりです。

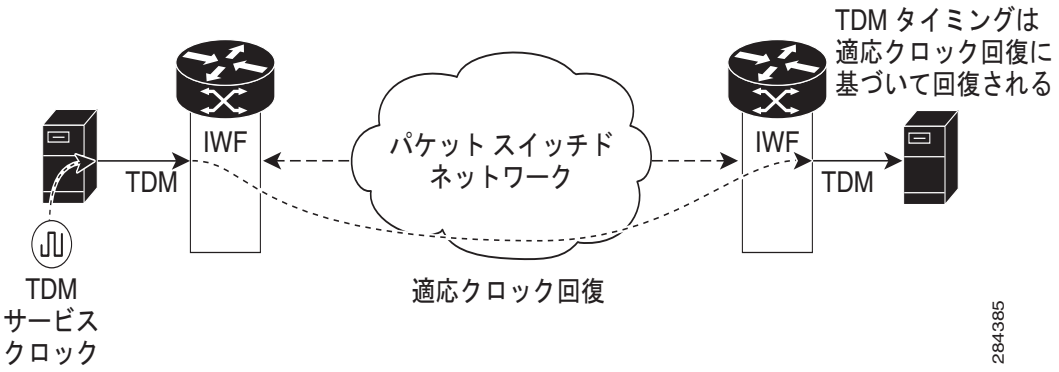
  - Cisco 24 ポート チャネライズド T1/E1 回線エミュレーションおよびチャネライズド ATM SPA：SPA ごとに 24 クロック。
  - Cisco 2 ポート チャネライズド T3/E3 回線エミュレーションおよびチャネライズド ATM SPA：SPA ごとに 10 クロック（T1/E1 モード）、SPA ごとに 2 クロック（T3/E3 モード）。
  - Cisco 1 ポート チャネライズド OC3/STM-1 回線エミュレーションおよびチャネライズド ATM SPA：SPA ごとに 10 クロック（T1/E1 モード）。

- 適応クロッキング：適応クロッキングは、ルータに共通クロック ソースがない場合に使用されます。図 31 を参照してください。クロックは、パケット到着率に基づいて導出されます。適応クロック回復アルゴリズムには、次の 2 つのタイプがあります。
  - デジッタ バッファ占有レベルに基づく
  - パケット到着率に基づく

クロック品質はパケット サイズによって異なり、パケット損失/破損への耐性は高くありません。また、不要な遅延の原因となります。クロック回復用バッファに十分な数のパケットを集める必要があるからです。デジッタ バッファ サイズでは、ネットワーク ジッタを容認するエミュレート サーキットの機能が決定されます。CEoP ソフトウェアのデジッタ バッファは設定可能であり、最大値は 500 ミリ秒です。

（注） CEoP SPA ハードウェアがサポートするのは、パケット到着率アルゴリズムだけです。

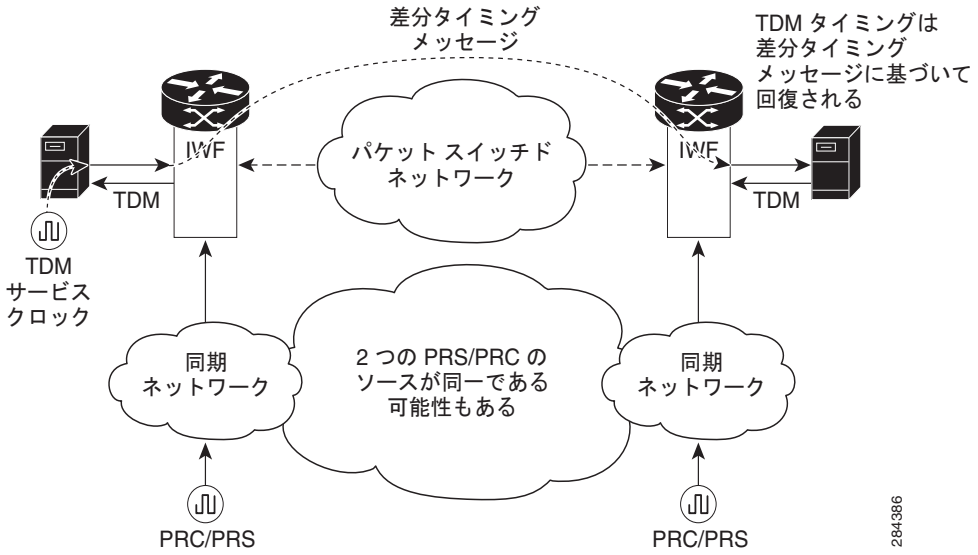
図 31　適応クロック回復



- 差分クロッキング：差分クロッキングが使用されるのは、セル サイトと集約ルータに共通のクロック ソースがあるが、TDM 回線のクロッキングに別のソースが使用されている場合です。TDM のクロックは、共通のクロックを基準とする、パケットの RTP ヘッダーの差分情報から導出されます。差分クロック回復の基になるのは、RTP ヘッダーで受信したタイムスタンプです。マスター側では、TDM クロックとネットワーク クロックの差が RTP ヘッダーに記録されます。スレーブ側では、このタイムスタンプが RTP ヘッダーから読み込まれ、クロック回復が行われて、このクロックが同期に使用されます。図 32 を参照してください。

（注） Cisco 1 ポート チャネライズド OC3/STM-1 CEoP SPA ハードウェアで回復できるのは、最大 10 個の CEM インターフェイスの一意のクロック 10 個までです。クロック回復が設定される CEM インターフェイスは、それぞれ異なる T1 にあることが必要です。

図 32　差分クロック回復



CEM の設定とコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide*』の「*Implementing Point to Point Layer 2 Services*」モジュールおよび『*Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Command Reference*』を参照してください。

サンプル CEM インターフェイス設定については、「回線エミュレーション インターフェイス設定：例」（P.409）を参照してください。

CEM の実装方法

ここでは、次の手順について説明します。

SONET VT1.5 マッピング T1 チャネルの設定と CEM インターフェイスの作成

Cisco 1 ポート チャネライズド OC3/STM-1 CEoP SPA の場合は、STS ストリームをチャネル化して、VT1.5 マッピングされた T1 チャネルとすることができます。

このタスクでは、VT マッピングされた T1 チャネルとなるように SONET 回線を設定する方法について説明します。

前提条件

なし。

制約事項

チャネライズド SONET STS ストリームと VT1.5-T1 マッピングの組み合わせは、次の SPA でサポートされます。

- Cisco 1 ポート チャネライズド OC-3/STM-1 SPA

手順の概要

1. **configure**
2. **hw-module subslot** *node-id* **cardtype** *type*
3. **commit**
4. **controller sonet** *interface-path-id*
5. **sts** *number*
6. **mode** *mode*
7. **root**
8. **controller t1** *interface-path-id*
9. **cem-group unframed**
10. **controller t1** ***interface-path-id***
11. **cem-group framed** *group-number* **timeslots** *range1-range2*

12. **no shutdown**
13. **end**
    または
    commit
14. **show runn interface cem** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**hw-module subslot** *node-id* **cardtype {sonet| sdh}** **例：** `RP/0/RSP0/CPU0:router(config-sonet)# hw-module subslot 0/3/0 sonet`	SONET フレーム構成のコントローラを設定します。 SONET フレーム構成（**sonet**）がデフォルトです。フレーミング モード（SONET/SDH）の変更があるたびに、SPA は自動的にリロードされます。リロードが発生するのは、すべての CEM インターフェイス、T1 コントローラおよび SONET コントローラの設定が完全に削除された場合だけです。これは、初めて設定するときには当てはまりません。T1 コントローラとインターフェイスの設定は存在していないからです。 この設定は、CEoP SPA がフレーミング モードの 1 つで正常に動作するために必須です。初めて設定するときは、カード タイプが SONET に設定されていれば、SPA のリロードは発生しません。
ステップ 3	**commit**	実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 4	**controller sonet** *interface-path-id*	コントローラ コンフィギュレーション サブモードを開始して、SONET コントローラ名とインスタンス ID を *rack/slot/module/port/controllerName* 表記で指定します。
ステップ 5	**sts** *number* **例：** `RP/0/RSP0/CPU0:router(config-sonet)# sts 1`	*number* により指定された STS ストリームを設定します。範囲は 1 ～ 3 です。
ステップ 6	**mode** *mode* **例：** `RP/0/RSP0/CPU0:router(config-stsPath)# mode t1`	STS レベルでのインターフェイスのモードを設定します。設定可能なモードは、次のとおりです。 • vt15-t1：仮想トリビュタリ 1.5 T1（VT15 T1）を伝送する SONET パス

	コマンドまたはアクション	目的
ステップ 7	**root** **例：** `RP/0/RSP0/CPU0:router(config-stsPath)# root`	グローバル コンフィギュレーション モードに戻ります。構造を認識しない CEM インターフェイスを作成する場合は、ステップ 7 に進みます。構造を認識する CEM インターフェイスを作成する場合は、ステップ 9 に進みます。
ステップ 8	**controller t1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t1 0/0/1/0/1/4/1`	T1 コントローラ コンフィギュレーション サブモードを開始して、T1 コントローラ名と *interface-path-id* を *rack/slot/module/port/sts-num/vtg-num/T1-num* 表記で指定します。
ステップ 9	**cem-group unframed** **例：** `RP/0/RSP0/CPU0:router(config)# cem-group unframed`	構造を認識しない CEM インターフェイスを作成します。
ステップ 10	**controller t1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t1 0/0/1/0/1/5/1`	T1 コントローラ コンフィギュレーション サブモードを開始して、T1 コントローラ名と *interface-path-id* を *rack/slot/module/port/sts-num/vtg-num/T1-num* 表記で指定します。
ステップ 11	**cem-group framed** *group-number* **timeslots** *range1-range2* **例：** `RP/0/RSP0/CPU0:router(config)# cem-group framed 0 timeslots 1`	構造を認識する CEM インターフェイスを作成します。**timeslots** キーワードでは、インターフェイスのタイム スロットを範囲として指定します。この指定には、*range1-range2* という表記を使用します。
ステップ 12	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-if)# no shutdown`	shutdown 設定を削除します。 **（注）** shutdown 設定を削除することにより、インターフェイスでの強制的な管理上の停止が排除されるため、インターフェイスはアップ状態またはダウン状態に移行することができます（親 SONET レイヤが管理上の停止状態に設定されていないことを前提とします）。

	コマンドまたはアクション	目的
ステップ 13	**end** または **commit** **例：** `RP/0/0RSP0/CPU0:router(config-sonet)# end` または `RP/0/RSP0/CPU0:router(config-sonet)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 14	**show runn interface cem** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show runn interface cem 0/0/2/0/1/1/1/1:1`	CEM インターフェイスの設定を確認します。

C11-T1 または C12-E1 にマッピングされる SDH AU-3 の設定

ここでは、次の作業について説明します。

- 「SDH AU-3 の C11-T1 へのマッピングの設定と CEM インターフェイスの作成」（P.384）
- 「SDH AU-3 の C12-E1 へのマッピングの設定と CEM インターフェイスの作成」（P.387）

SDH AU-3 の C11-T1 へのマッピングの設定と CEM インターフェイスの作成

このタスクでは、c11-t1 マッピングを行うように SDH AU-3 を設定する方法について説明します。

前提条件

- SONET/SDH コントローラの設定方法を理解している必要があります。

制約事項

チャネライズド SDH AU-3 と c11-t1 マッピングの組み合わせがサポートされるのは、次の SPA です。

- Cisco 1 ポート チャネライズド OC-3/STM-1 SPA

手順の概要

1. **configure**

2. **hw-module subslot** *node-id* **cardtype** *type*
3. **commit**
4. **controller sonet** ***interface-path-id***
5. **au** *number*
6. **mode** *mode*
7. **root**
8. **controller t1** *interface-path-id*
9. **cem-group unframed**
10. **controller t1** ***interface-path-id***
11. **cem-group framed** *group-number* **timeslots** *range1-range2*
12. **no shutdown**
13. **end**
    または
    commit
14. **show runn interface cem** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**hw-module sub-slot** *node-id* **cardtype** *type* **例：** `RP/0/RSP0/CPU0:router(config-sonet)# hw-module sub-slot <> cardtype sdh`	同期デジタル階層（SDH）のコントローラを設定します。**hw-module subslot** *node-id* **cardtype** *type* コマンドは、SPA を SONET/SDH モードで動作するように設定します。 このコマンドを実行すると、コミットのときに SPA の自動リロードが発生します。リロードが発生するのは、すべての CEM インターフェイス、T1 コントローラおよび SONET コントローラの設定が完全に削除された場合だけです。これは、初めて設定するときには当てはまりません。T1 コントローラとインターフェイスの設定は存在していないからです。 この設定は、CEoP SPA がフレーミング モードの 1 つで正常に動作するために必須です。SONET フレーム構成（**sonet**）がデフォルトです。
ステップ 3	**commit**	実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 4	**controller sonet** *interface-path-id*	コントローラ コンフィギュレーション サブモードを開始して、SDH コントローラ名とインスタンス ID を *rack/slot/module/port/controllerName* 表記で指定します。

	コマンドまたはアクション	目的
ステップ 5	**au** *number* **例：** `RP/0/RSP0/CPU0:router(config-sonet)# au 1`	管理ユニット（AU）グループを指定し、AU パス コンフィギュレーション モードを開始します。AU-3 の有効範囲は、次のとおりです。 • 1 ～ 3：1 ポート チャネライズド OC-3/STM-1 SPA **（注）** **au** コマンドは AU タイプを指定しません。これは、設定する AU タイプの AU グループの番号を指定するものです。AU コマンドの範囲は、AU-3 と AU-4 のどちらを設定するかによって異なります。
ステップ 6	**mode** *mode* **例：** `RP/0/RSP0/CPU0:router(config-auPath)# mode c11-t1`	AU レベルでのインターフェイスのモードを設定します。AU-3 パスは、サポートされている SPA で c11-t1 にマッピングできます。
ステップ 7	**root** **例：** `RP/0/RSP0/CPU0:router(config-auPath)# root`	グローバル コンフィギュレーション モードに戻ります。
ステップ 8	**controller t1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller T1 0/0/2/0/1/1/4`	T1 コントローラ コンフィギュレーション サブモードを開始して、T1 コントローラ名と *interface-path-id* を *rack/slot/module/port/auNum/t1Num* 表記で指定します。
ステップ 9	**cem-group unframed** **例：** `RP/0/RSP0/CPU0:router(config)# cem-group unframed`	構造を認識しない CEM インターフェイスを作成します。
ステップ 10	**controller t1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t1 0/0/2/0/1/1/7`	T1 コントローラ コンフィギュレーション サブモードを開始して、T1 コントローラ名と *interface-path-id* を *rack/slot/module/port/auNum/t1Num* 表記で指定します。
ステップ 11	**cem-group framed** *group-number* **timeslots** *range1-range2* **例：** `RP/0/RSP0/CPU0:router(config)# cem-group framed 1 timeslots 2-3`	構造を認識する CEM インターフェイスを作成します。**timeslots** キーワードでは、インターフェイスのタイム スロットを範囲として指定します。この指定には、*range1-range2* という表記を使用します。

	コマンドまたはアクション	目的
ステップ 12	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-if)# no shutdown`	shutdown 設定を削除します。 **（注）** shutdown 設定を削除することにより、インターフェイスでの強制的な管理上の停止が排除されるため、インターフェイスはアップ状態またはダウン状態に移行することができます（親 SONET レイヤが管理上の停止状態に設定されていないことを前提とします）。
ステップ 13	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-sonet)# end` または `RP/0/RSP0/CPU0:router(config-sonet)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 14	**show runn interface cem** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show runn interface cem 0/0/2/0/1/1/1/1:1`	CEM インターフェイスの設定を確認します。

SDH AU-3 の C12-E1 へのマッピングの設定と CEM インターフェイスの作成

このタスクでは、c12-e1 マッピングを行うように SDH AU-3 を設定する方法について説明します。

前提条件

- SONET/SDH コントローラの設定方法を理解している必要があります。

制約事項

チャネライズド SDH AU-3 と c12-e1 マッピングの組み合わせがサポートされるのは、次の SPA です。

- Cisco 1 ポート チャネライズド OC-3/STM-1 SPA

手順の概要

1. **configure**

2. **hw-module subslot** *node-id* **cardtype** *type*
3. **commit**
4. **controller sonet** ***interface-path-id***
5. **au** *number*
6. **mode tug3**
7. **width** *number*
8. **tug3** *number*
9. **mode** *mode*
10. **root**
11. **controller e1** *interface-path-id*
12. **cem-group unframed**
13. **controller e1** ***interface-path-id***
14. **cem-group framed** *group-number* **timeslots** *range1-range2*
15. **no shutdown**
16. **end**
    または
    commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**hw-module sub-slot** *node-id* **cardtype** *type* **例：** `RP/0/RSP0/CPU0:router(config-sonet)# hw-module sub-slot <> cardtype sdh`	同期デジタル階層（SDH）フレーミングのコントローラ フレーミングを設定します。**hw-module subslot** *node-id* **cardtype** *type* コマンドは、SPA を SONET/SDH モードで動作するように設定します。このコマンドを実行すると、コミットのときに SPA の自動リロードが発生します。リロードが発生するのは、すべての CEM インターフェイス、E1 コントローラおよび SONET コントローラの設定が完全に削除された場合だけです。
ステップ 3	**commit**	実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 4	**controller sonet** *interface-path-id*	コントローラ コンフィギュレーション サブモードを開始して、SDH コントローラ名とインスタンス ID を *rack/slot/module/port/controllerName* 表記で指定します。

	コマンドまたはアクション	目的
ステップ 5	**au** *number* **例：** `RP/0/RSP0/CPU0:router(config-sonet)# au 1`	管理ユニット（AU）グループを指定し、AU パス コンフィギュレーション モードを開始します。AU-3 の有効範囲は、次のとおりです。 • 1 ～ 3：1 ポート チャネライズド OC-3/STM-1 SPA **（注）** **au** コマンドは AU タイプを指定しません。これは、設定する AU タイプの AU グループの番号を指定するものです。AU コマンドの範囲は、AU-3 と AU-4 のどちらを設定するかによって異なります。
ステップ 6	**mode tug3** **例：** `RP/0/RSP0/CPU0:router(config-auPath)# mode tug3`	AU レベルでのインターフェイスのモードを設定します。現在サポートされているのは TUG3 のみです。
ステップ 7	**width** *number* **例：** `RP/0/RSP0/CPU0:router(config-auPath)# width 3`	AU ストリーム数を設定します。
ステップ 8	**tug3** *number* **例：** `RP/0/RSP0/CPU0:router(config-auPath)#tug3 1`	トリビュタリ ユニット グループ（TUG）の *number* を指定して、config-tug3Path モードを開始します。範囲は 1 ～ 3 です。
ステップ 9	**mode** *mode* **例：** `RP/0/RSP0/CPU0:router(config-tug3Path)# mode c12-e1`	tug3 レベルでのインターフェイスのモードを設定します。使用可能なモードを次に示します。 • **c12-e1**：TU-12 から E1 を伝送する TUG-3 パス
ステップ 10	**root** **例：** `RP/0/RSP0/CPU0:router(config-auPath)# root`	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 11	**controller e1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller E1 0/0/2/0/1/1/1`	E1 コントローラ コンフィギュレーション サブモードを開始して、E1 コントローラ名と *interface-path-id* を *rack/slot/module/port/auNum/tugNum/t1Num* 表記で指定します。
ステップ 12	**cem-group unframed** **例：** `RP/0/RSP0/CPU0:router(config)# cem-group unframed`	構造を認識しない CEM インターフェイスを作成します。

	コマンドまたはアクション	目的
ステップ 13	**controller e1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller E1 0/0/2/0/1/1/7`	E1 コントローラ コンフィギュレーション サブモードを開始して、E1 コントローラ名と *interface-path-id* を *rack/slot/module/port/auNum/tugNum/t1Num* 表記で指定します。
ステップ 14	**cem-group framed** *group-number* **timeslots** *range1-range2* **例：** `RP/0/RSP0/CPU0:router(config)# cem-group framed 0 timeslots 1`	構造を認識する CEM インターフェイスを作成します。
ステップ 15	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-if)# no shutdown`	shutdown 設定を削除します。 **（注）** shutdown 設定を削除することにより、インターフェイスでの強制的な管理上の停止が排除されるため、インターフェイスはアップ状態またはダウン状態に移行することができます（親 SONET レイヤが管理上の停止状態に設定されていないことを前提とします）。
ステップ 16	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-sonet)# end` または `RP/0/RSP0/CPU0:router(config-sonet)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

Cisco 24 ポート チャネライズド T1/E1 回線エミュレーションおよびチャネライズド ATM SPA の設定と CEM インターフェイスの作成

このタスクでは、Cisco 24 ポート チャネライズド T1/E1 回線エミュレーションおよびチャネライズド ATM SPA を設定する方法を説明します。

手順の概要

1. **configure**
2. **hw-module subslot** *node-id* **cardtype** *type*
3. **hw-module subslot** *node-id* **mode** *CEM*
4. **commit**
5. **controller t1/e1** *interface-path-id*
6. **cem-group unframed**
7. **controller t1/e1** ***interface-path-id***
8. **cem-group framed** *group-number* **timeslots** *range1-range2*
9. **no shutdown**
10. **end**
    または
    commit
11. **show runn interface cem** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**hw-module subslot** *node-id* **cardtype {t1| e1}** **例：** `RP/0/RSP0/CPU0:router(config-t1)# hw-module subslot 0/3/0 cardtype t1`	**hw-module subslot** *node-id* **cardtype** *type* コマンドは、SPA を t1/e1 モードで動作するように設定します。 リロードが発生するのは、すべての CEM インターフェイスおよび T1 コントローラの設定が完全に削除された場合だけです。これは、初めて設定するときには当てはまりません。T1 コントローラとインターフェイスの設定は存在していないからです。

	コマンドまたはアクション	目的
ステップ 3	**hw-module subslot** *node-id* **mode CEM** **例：** `RP/0/RSP0/CPU0:router(config-t1)# hw-module subslot 0/3/0 mode CEM`	**hw-module subslot** *node-id* **mode** *CEM* コマンドは、SPA を CEM モードで動作するように設定します。
ステップ 4	**commit**	実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	**controller t1|e1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t1 0/0/1/0/1`	T1 コントローラ コンフィギュレーション サブモードを開始して、T1 コントローラ名と *interface-path-id* を *rack/slot/module/port/T1num* 表記で指定します。
ステップ 6	**cem-group unframed** **例：** `RP/0/RSP0/CPU0:router(config-t1)# cem-group unframed`	構造を認識しない CEM インターフェイスを作成します。
ステップ 7	**controller t1|e1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t1 0/0/1/0/1`	T1 コントローラ コンフィギュレーション サブモードを開始して、T1 コントローラ名と *interface-path-id* を *rack/slot/module/port/T1num* 表記で指定します。
ステップ 8	**cem-group framed** *group-number* **timeslots** *range1-range2* **例：** `RP/0/RSP0/CPU0:router(config-t1)# cem-group framed 0 timeslots 1`	構造を認識する CEM インターフェイスを作成します。**timeslots** キーワードでは、インターフェイスのタイム スロットを範囲として指定します。この指定には、*range1-range2* という表記を使用します。
ステップ 9	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-if)# no shutdown`	shutdown 設定を削除します。 **（注）** shutdown 設定を削除することにより、インターフェイスでの強制的な管理上の停止が排除されるため、インターフェイスはアップ状態またはダウン状態に移行することができます。

	コマンドまたはアクション	目的
ステップ 10	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-t1)# end` または `RP/0/RSP0/CPU0:router(config-t1)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 11	**show runn interface cem** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show runn interface cem 0/0/2/0/1:1`	CEM インターフェイスの設定を確認します。

Cisco 2 ポート チャネライズド T3/E3 回線エミュレーションおよびチャネライズド ATM SPA の設定と CEM インターフェイスの作成

T3/E3 チャネライゼーション モード

ここでは、Cisco 2 ポート チャネライズド T3/E3 回線エミュレーションおよびチャネライズド ATM SPA を、T3 チャネル化を使用して設定する方法について説明します。

（注） Cisco 2 ポート チャネライズド T3/E3 回線エミュレーションおよびチャネライズド ATM SPA では、T3 チャネルを T1 または E1 にチャネル化でき、T1 や E1 をさらに DS0 タイムスロットにチャネル化できます。

手順の概要

1. **configure**
2. **hw-module subslot** *node-id* **cardtype** *type*

3. **hw-module subslot** *node-id* **mode** *CEM*
4. **commit**
5. **controller {t3 | e3}** *interface-path-id*
6. **cem-group unframed**
7. **no shutdown**
8. **end**
   または
   commit
9. **show runn interface cem** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**hw-module subslot** *node-id* **cardtype {t3| e3}** **例：** `RP/0/RSP0/CPU0:router(config-t3)# hw-module subslot 0/3/0 cardtype t3`	**hw-module subslot** *node-id* **cardtype** *type* コマンドは、SPA を t3/e3 モードで動作するように設定します。 フレーミング モード（t3/e3）の変更があるたびに、SPA は自動的にリロードされます。リロードが発生するのは、すべての CEM インターフェイスおよび T3 コントローラの設定が完全に削除された場合だけです。これは、初めて設定するときには当てはまりません。T3 コントローラとインターフェイスの設定は存在していないからです。
ステップ 3	**hw-module subslot** *node-id* **mode CEM** **例：** `RP/0/RSP0/CPU0:router(config-t3)# hw-module subslot 0/3/0 mode CEM`	**hw-module subslot** *node-id* **mode** *CEM* コマンドは、SPA を CEM モードで動作するように設定します。
ステップ 4	**commit**	実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	**controller {t3|e3}** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t3 0/0/1/0/4`	T3/E3 コントローラ コンフィギュレーション サブモードを開始して、T3/E3 コントローラ名と *interface-path-id* を *rack/slot/module/port/T3Num* 表記で指定します。

	コマンドまたはアクション	目的
ステップ 6	**cem-group unframed** **例：** `RP/0/RSP0/CPU0:router(config-t3)# cem-group unframed`	構造を認識しない CEM インターフェイスを作成します。非フレーム化 CEM インターフェイスだけがこのモードでサポートされます。
ステップ 7	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-if)# no shutdown`	shutdown 設定を削除します。 **（注）** shutdown 設定を削除することにより、インターフェイスでの強制的な管理上の停止が排除されるため、インターフェイスはアップ状態またはダウン状態に移行することができます。
ステップ 8	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-t3)# end` または `RP/0/RSP0/CPU0:router(config-t3)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 9	**show runn interface cem** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show runn interface cem 0/0/1/0/4:1`	CEM インターフェイスの設定を確認します。

T1/E1 チャネライゼーション モード

ここでは、Cisco 2 ポート チャネライズド T3/E3 回線エミュレーションおよびチャネライズド ATM SPA を、T1/E1 チャネル化を使用して設定する方法について説明します。

手順の概要

1. **configure**
2. **hw-module subslot** *node-id* **cardtype** *type*
3. **hw-module subslot** *node-id* **mode** *CEM*

4. **commit**
5. **controller t3** *interface-path-id*
6. **mode {t1|e1}**
7. **controller t1** *interface-path-id*
8. **cem-group unframed**
9. **controller t1** ***interface-path-id***
10. **cem-group framed** *group-number* **timeslots** *range1-range2*
11. **no shutdown**
12. **end**
    または
    commit
13. **show runn interface cem** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**hw-module subslot** *node-id* **cardtype {t3|e3}** **例：** `RP/0/RSP0/CPU0:router(config-sonet)# hw-module subslot 0/3/0 t3`	**hw-module subslot** *node-id* **cardtype** *type* コマンドは、SPA を t3/e3 モードで動作するように設定します。 フレーミング モード（t3/e3）の変更があるたびに、SPA は自動的にリロードされます。リロードが発生するのは、すべての CEM インターフェイスおよび T3 コントローラの設定が完全に削除された場合だけです。これは、初めて設定するときには当てはまりません。T3 コントローラとインターフェイスの設定は存在していないからです。
ステップ 3	**hw-module subslot** *node-id* **mode CEM** **例：** `RP/0/RSP0/CPU0:router(config-t3)# hw-module subslot 0/3/0 mode CEM`	**hw-module subslot** *node-id* **mode** *CEM* コマンドは、SPA を CEM モードで動作するように設定します。
ステップ 4	**commit**	実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	**controller t3** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t3 0/0/1/0/4`	T3 コントローラ コンフィギュレーション サブモードを開始して、T3 コントローラ名と *interface-path-id* を *rack/slot/module/port/T3Num* 表記で指定します。

	コマンドまたはアクション	目的
ステップ 6	**mode {t1|e1}**	インターフェイスのモードを設定します。設定可能なモードは、T1 および E1 のチャネライゼーション モードです。
ステップ 7	**controller t1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t1 0/0/1/0/4/1`	T1 コントローラ コンフィギュレーション サブモードを開始して、T1 コントローラ名と *interface-path-id* を *rack/slot/module/port/T3Num/T1num* 表記で指定します。
ステップ 8	**cem-group unframed** **例：** `RP/0/RSP0/CPU0:router(config-t1)# cem-group unframed`	構造を認識しない CEM インターフェイスを作成します。
ステップ 9	**controller t1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t1 0/0/1/0/4/1`	T1 コントローラ コンフィギュレーション サブモードを開始して、T1 コントローラ名と *interface-path-id* を *rack/slot/module/port/T3Num/T1num* 表記で指定します。
ステップ 10	**cem-group framed** *group-number* **timeslots** *range1-range2* **例：** `RP/0/RSP0/CPU0:router(config-t1)# cem-group framed 0 timeslots 1`	構造を認識する CEM インターフェイスを作成します。**timeslots** キーワードでは、インターフェイスのタイム スロットを範囲として指定します。この指定には、*range1-range2* という表記を使用します。
ステップ 11	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-if)# no shutdown`	shutdown 設定を削除します。 **（注）** shutdown 設定を削除することにより、インターフェイスでの強制的な管理上の停止が排除されるため、インターフェイスはアップ状態またはダウン状態に移行することができます。

	コマンドまたはアクション	目的
ステップ 12	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-t1)# end` または `RP/0/RSP0/CPU0:router(config-t1)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 13	**show runn interface cem** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show runn interface cem 0/0/2/0/1/1/1/1:1`	CEM インターフェイスの設定を確認します。

CEM インターフェイスの設定

ここでは、CEM の設定方法について説明します。CEM は、時分割多重（TDM）ネットワークと、マルチ プロトコル ラベル スイッチング（MPLS）を使用するパケット ネットワークとの間のブリッジとなります。ルータは、TDM データを MPLS パケットにカプセル化し、CEM 疑似回線を介して、そのデータをリモートのプロバイダー エッジ（PE）ルータに送信します。

次の各項で、CEM を設定する方法について説明します。

- 設定時の注意事項および制約事項
- グローバル CEM クラスの設定
- CEM クラスのアタッチ
- ペイロード サイズの設定
- デジッタ バッファ サイズの設定
- アイドル パターンの設定
- ダミー モードのイネーブル化
- ダミー パターンの設定

設定時の注意事項および制約事項

ペイロード サイズおよびデジッタ バッファ サイズのすべての組み合わせがサポートされるとは限りません。適合しないペイロード サイズまたはデジッタ バッファの設定を適用すると、ルータはその設定を拒否して以前の設定に戻ります。

グローバル CEM クラスの設定

ここでは、グローバル CEM クラスを設定する方法について説明します。

（注） どのインターフェイス設定も、優先度は、CEM クラスのアタッチによって適用された設定よりも高くなります。また、インターフェイスにアタッチされた CEM クラスの優先度は、親コントローラにアタッチされた CEM クラスよりも高くなります。たとえば、ダミー パターン値 *0xcf* がインターフェイスに直接適用されているときに、ダミー パターン値 *0xaa* を持つ CEM クラスが同じインターフェイスにアタッチされた場合は、ダミー パターン値は *0xcf* となります。新しい設定が適用されるのは、インターフェイスに直接適用されたダミー パターン値が削除された後となります。

手順の概要

1. **configure**
2. **cem class** *class-name*
3. **payload** *value*
4. **dejitter** *value*
5. **idle pattern** *value*
6. **dummy mode {last-frame|user-defined}**
7. **dummy pattern** *value*
8. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**cem class** *class-name* **例：** `RP/0/RSP0/CPU0:router(config)# cem class Default`	新しい CEM クラスを作成します。

	コマンドまたはアクション	目的
ステップ 3	**payload** *value* **例：** `RP/0/RSP0/CPU0:router(config-cem-class)# payload 512`	CEM クラスのペイロード サイズを入力します。
ステップ 4	**dejitter** *value* **例：** `RP/0/RSP0/CPU0:router(config-cem-class)# dejitter 10`	CEM クラスのデジッタ バッファ サイズを入力します。
ステップ 5	**idle pattern** *value* **例：** `RP/0/RSP0/CPU0:router(config-cem-class)# idle pattern 0x55`	CEM クラスのアイドル パターン値を入力します。
ステップ 6	**dummy mode** **例：** `RP/0/RSP0/CPU0:router(config-cem-class)# dummy mode last-frame`	CEM クラスのダミー モードを開始します。選択肢は last-frame と user-defined です。
ステップ 7	**dummy pattern** *value* **例：** `RP/0/RSP0/CPU0:router(config-cem-class)# dummy pattern`	CEM クラスのダミー パターン値を入力します。この値が適用されるのは、ダミー モードが user-defined の場合のみです。
ステップ 8	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-cem-class)# end` または `RP/0/RSP0/CPU0:router(config-cem-class)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

CEM クラスのアタッチ

ここでは、グローバル CEM クラスをアタッチする手順について説明します。

（注） CEM クラスは、CEM インターフェイスにアタッチすることも、T1/E1 コントローラにアタッチすることもできます。

手順の概要

1. **configure**
2. **interface cem** *interface-path-id*（または）**controller {t1|e1}** ***rack/slot/subslot/port***
3. **cem class-attach** *class-name*
4. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface cem** *interface-path-id* （または） **controller {t1|e1}** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t1 0/0/2/0/1/1`	CEM インターフェイスまたは T1/E1 コントローラを指定します。

	コマンドまたはアクション	目的
ステップ 3	**cem class-attach** *class-name* **例：** `RP/0/RSP0/CPU0:router(config)# cem class-attach Default`	CEM クラスをインターフェイスまたはコントローラにアタッチします。
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-cem-class)# end` または `RP/0/RSP0/CPU0:router(config-cem-class)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ペイロード サイズの設定

1 つの IP パケットとしてカプセル化されるバイト数を指定するには **cem payload** コマンドを使用します。size 引数は、各パケットのペイロードでのバイト数を指定します。範囲は 32 ～ 1312 バイトです。

非構造化 CEM チャネルのデフォルトのペイロード サイズは、次のとおりです。

- E1 = 256 バイト
- T1 = 192 バイト
- E3 = 1024 バイト
- T3 = 1024 バイト

構造化 CEM チャネルのデフォルトのペイロード サイズは、チャネルを構成するタイム スロットの数によって異なります。ペイロード（L、バイト単位）、タイム スロットの数（N）、およびパケット化遅延（D、ミリ秒単位）の間には、L = 8 * N * D という関係があります。

デフォルトのペイロード サイズは、パケット化遅延を使用して計算されます。この数値は、CEM インターフェイスが表すタイムスロットの数に依存します。タイムスロットの数とパケット化遅延との関係を次に示します。

- N = 1 の場合は、D は 8 ミリ秒（対応するパケット ペイロード サイズは 64 バイト）
- 2 <= N <= 4 の場合は、D は 4 ミリ秒（対応するパケット ペイロード サイズは 32 * N バイト）
- N >= 5 の場合は、D は 1 ミリ秒（対応するパケット ペイロード サイズは 8 * N オクテット）

N = 1 の場合にパケット化遅延 5 ミリ秒のサポートを推奨します。

デジッタ バッファ サイズの設定

ネットワーク フィルタを補うために使用されるデジッタ バッファのサイズを指定するには、**cem dejitter** コマンドを使用します。設定されたデジッタ バッファ サイズはミリ秒単位からパケット単位に変換され、次のパケット数（整数）に切り上げられます。バッファのサイズをミリ秒単位で指定するには、size 引数を使用します。範囲は 1 ～ 500 ミリ秒です。次に、例を示します。

```
Router(config-cem)# cem dejitter 5
```

CEM チャネルのデフォルトのデジッタ バッファは、CESoPSN か SAToP かにかかわらず、次のとおりです。

- E1 = 16 ミリ秒
- T1 = 16 ミリ秒
- E3 = 5 ミリ秒
- T3 = 5 ミリ秒

（注） SAToP T1/E1、T3/E3、および CESoPSN の回線でのペイロードとデジッタ バッファとの関係については、表 6、表 7、および表 8 を参照してください。ペイロードとデジッタの設定は、表に示した最小値と最大値に従って行う必要があります。

（注） 最大および最小デジッタ バッファ値の範囲は、ペイロード値に対して固定です。

アイドル パターンの設定

アイドル パターンを指定するには、[no] **cem idle pattern pattern** コマンドを使用します。それぞれの損失 CESoPSN データ パケットのペイロードは、代わりになる同等量のデータに置き換える必要があります。pattern の範囲は 0x0 ～ 0xff です。デフォルトのアイドル パターンは 0xff です。次に例を示します。

```
Router(config-cem)# cem idle pattern 0xff
```

ある CEM インターフェイスで受信するはずの CEM パケットが受信されず、損失したと考えられる場合は、CEoP SPA によってアイドル パターンが TDM 接続回線に向けて出力されます。これは、CEM グループ内で設定された、それぞれのタイムスロットで行われます。

ダミー モードのイネーブル化

ダミー モードは、損失フレームまたは破損フレームの穴埋め用ビット パターンをイネーブルにします。ダミー モードをイネーブルにするには、**cem dummy mode** [**last-frame** | **user-defined**] コマンドを使用します。デフォルトは **last-frame** です。次に例を示します。

```
Router(config-cem)# cem dummy mode last-frame
```

パケット損失の原因が順序不正の場合や、パケットの順序変更に失敗した場合は、CEoP SPA はダミー パターンを TDM 接続回線に向けて出力します。これは、CEM グループ内で設定された、それぞれのタイムスロットで行われます。

ダミー パターンの設定

ダミー モードが user-defined に設定されている場合は、**cem dummy-pattern** コマンドを使用してダミー パターンを設定できます。pattern の範囲は、0x0 ～ 0xff です。デフォルトのダミー パターンは 0xff です。次に例を示します。

```
Router(config-cem)# cem dummy-pattern 0xff
```

表 6 に、T1/E1 SAToP 回線のペイロードとデジッタとの関係を示します。

表 6　T1/E1 SAToP 回線：ペイロードおよびジッターの制限

T1/E1	最大ペイロード	最大ジッター	最小ジッター	最小ペイロード	Maximum Jitter	最小ジッター
T1	960	320	10	192	64	2
E1	1280	320	10	256	64	2

表 7 に、T3/E3 SAToP 回線のペイロードとデジッタとの関係を示します。

表 7　T3/E3 SAToP 回線：ペイロードおよびジッターの制限

T3/E3	最大ペイロード	最大ジッター	最小ジッター	最小ペイロード	Maximum Jitter	最小ジッター
T3	1312	8	2	672	8	2
E3	1312	16	2	512	8	2

表 8 に、DS0 回線のペイロードとデジッタとの関係を示します。

表 8 CESoPSN DS0 回線：ペイロードおよびジッターの制限

DS0	最大ペイロード	最大ジッター	最小ジッター	最小ペイロード	Maximum Jitter	最小ジッター
1	40	320	10	32	256	8
2	80	320	10	32	128	4
3	120	320	10	33	128	4
4	160	320	10	32	64	2
5	200	320	10	40	64	2
6	240	320	10	48	64	2
7	280	320	10	56	64	2
8	320	320	10	64	64	2
9	360	320	10	72	64	2
10	400	320	10	80	64	2
11	440	320	10	88	64	2
12	480	320	10	96	64	2
13	520	320	10	104	64	2
14	560	320	10	112	64	2
15	600	320	10	120	64	2
16	640	320	10	128	64	2
17	680	320	10	136	64	2
18	720	320	10	144	64	2
19	760	320	10	152	64	2
20	800	320	10	160	64	2
21	840	320	10	168	64	2
22	880	320	10	176	64	2
23	920	320	10	184	64	2
24	960	320	10	192	64	2
25	1000	320	10	200	64	2
26	1040	320	10	208	64	2
27	1080	320	10	216	64	2
28	1120	320	10	224	64	2
29	1160	320	10	232	64	2
30	1200	320	10	240	64	2
31	1240	320	10	248	64	2
32	1280	320	10	256	64	2

クロッキングの設定

SPA ポートごとに、ホスト カードからのシステム クロックを使用するか、ループ タイムとするかを個別に設定する必要があります。また、各 SPA は、ホストに参照クロックを渡します。これは、受信したポート クロックから選択できます。ここでは、1xOC3 SPA でクロッキングを設定する方法について説明します。

ここでは、次の内容について説明します。

クロック回復の設定

クロック回復を設定する場合は、次の注意事項に従ってください。

適応クロック回復

- クロック ソース：
  - Cisco IOS XR Release 4.2.x 以降では、1 ポート チャネライズド OC-3/STM1 CEoP SPA の CEM インターフェイスからの再生クロックを、その SPA 自身のクロック ソースとして使用できます。
- 許容クロック ソース数：
  - 詳細については、「クロック配信」（P.379）を参照してください。
- ルータに使用するクロックは、ネットワーク クロックと同じものにする必要があります。この場合、疑似回線がクロックを伝送できます。
- 安定したクロック回復を提供するネットワーク上での CEM 疑似回線の最少バンドル サイズは 4 DS0 です。
- 安定したクロック回復を提供するネットワーク上での CEM 疑似回線の最少パケット サイズは 64 バイトです。

クロッキング差

- 1 ポート チャネライズド OC-3/STM1 CEoP SPA をソースとする差分クロックの最大数は 10 です。
- 1 ポート チャネライズド OC-3/STM1 CEoP SPA は、最大 10 個の T1/E1 クロックを回復できます。
- 同一ポートから送信されるバンドルにはいくつかあります。ポートのクロック伝送に使用されるバンドルは、ポートの最初に作成されたバンドルです。ポートの最初の DS0 を含む疑似回線だけが、クロック差を伝送できます。
- Stratum-1 クロックが存在している必要があります。これは、両方の PE ルータに送信される共通クロックです。ない場合は、回復が期待どおりに動作しません。

CEoP SPA でクロック回復を設定し、再生クロックをコントローラに適用するには、次の手順を使用します。

手順の概要

1. **configure**
2. **interface cem** *rack/slot/subslot/port:cem-group*
3. **transmit-clock** differential

4. **recover-clock** *clock-id* **{adaptive |differential}**
5. **controller {t1|e1|t3|e3}** ***rack/slot/subslot/port***
6. **clock source recovered** *clock-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface cem** `rack/slot/subslot/port:cem-group` **例：** `RP/0/RSP0/CPU0:router(config)# interface cem 0/1/0/0:2`	完全な CEM インターフェイス インスタンスを指定します。
ステップ 3	**transmit-clock** {**differential**} **例：** `RP/0/RSP0/CPU0:router(config-if)# transmit-clock source internal`	CEM ポート送信クロック ソースを設定します。これは一般的には、マスターとしてクロックを送信するノードにおいて設定されます。このコマンドは、適応クロック回復に必須ではありません。
ステップ 4	**recover-clock** *clock-id* **{adaptive| differential}** **例：** `RP/0/RSP0/CPU0:router(config-if)# recover-clock clock-id <> adaptive`	再生クロック番号およびクロック回復タイプを指定します。これは一般的には、スレーブとしての役割を持つノード（コアからの着信 CEM パケットからクロックを回復する）において設定されます。
ステップ 5	**controller** *name instance* **例：** `RP/0/RSP0/CPU0:router(config)# controller t1 0/1/1/0/0/0`	コントローラ コンフィギュレーション サブモードを開始して、コントローラ名とインスタンス ID を *rack/slot/module/port/name/instance1/instance2* 表記で指定します。
ステップ 6	**clock source recovered** *clock-id* **例：** `RP/0/RSP0/CPU0:router(config-t1)# clock source recovered 3`	再生クロック番号を指定します。これは、T1/E1 コントローラの CEM インターフェイスからの再生クロックに適用されます。

クロック回復の確認

クロック回復を確認するには、**show recovered-clock** コマンドを使用します。

```
Router# show recovered-clock sublsot 0/3/0
Recovered clock status for subslot 0/3/0
----------------------------------------
Clock    Mode         Port CEM  Status     Frequency Offset(ppb)
1        ADAPTIVE     0    1    HOLDOVER   0
Router# show recovered-clock
Recovered clock status for subslot 3/0
----------------------------------------
Clock    Mode         Port CEM  Status     Frequency Offset(ppb)
1        ADAPTIVE     0    1    ACQUIRING  -694
```

CEM 用の show コマンド

コマンド **show controller cem** <*forward interface instance*> を使用すると、CEM パラメータ情報を確認できます。次の例は、コマンドのサンプル出力を示しています。

show controller cem forward interface instance コマンドの出力

```
RP/0/RSP0/CPU0:Router# show controllers cem 0/4/1/0:0

Interface               : CEM0/4/1/0:0
Admin state             : Up
Driver link state       : Up
Port bandwidth(kbps)    : 1984
Dejitter buffer         : 16
Payload size            : 248
Dummy mode              : last-frame
Dummy pattern           : 0xff
Idle pattern            : 0xff
Signalling              : No CAS
RTP                     : Enabled
Ingress packets         : 1638960097, Ingress packets drop     : 0
Egress packets          : 1207954294, Egress packets drop      : 287140468
Missing packets         : 160475876, Reordered packets        : 50092
Malformed packets       : 73, Misorder drops           : 7
Jitter buffer underrun  : 28, Error seconds            : 79673
Severely error seconds  : 25721, Unavailable seconds      : 160361
Failure counts          : 2
```

CEM の設定例

ここでは、次の例を示します。

回線エミュレーション インターフェイス設定：例

次の例では、Cisco 1 ポート チャネライズド OC3/STM-1 SPA でのサンプル CEM インターフェイス設定を示します。

チャネライズド SONET/SDH 設定と CEM インターフェイスの作成

SONET - T1 チャネル化と CEM インターフェイス作成

```
hw-module subslot  <loc> cardtype sonet
controller SONET 0/0/1/0
 sts 1
  mode vt15-t1
sts 2
  mode vt15-t1
sts 3
  mode vt15-t1
commit
```

構造を認識しない CEM インターフェイスの場合：

```
controller T1 0/0/1/0/1/4/1
 cem-group unframed
```

構造を認識する CEM インターフェイスの場合：

```
controller T1 0/0/1/0/1/5/1
 cem-group framed 0 timeslots 1
 cem-group framed 1 timeslots 2-3
 cem-group framed 2 timeslots 4-6
 cem-group framed 3 timeslots 7-10
 cem-group framed 4 timeslots 11-15
```

```
 cem-group framed 5 timeslots 16-21
 cem-group framed 6 timeslots 22-24
```

SDH - T1 チャネル化と CEM インターフェイス作成

```
hw-module subslot  <loc> cardtype sdh
controller SONET0/0/2/0
 au 1
  mode c11-t1
 au 2
  mode c11-t1
 au 3
  mode c11-t1
commit
```

構造を認識しない CEM インターフェイスの場合：

```
controller T1 0/0/2/0/1/1/4
 cem-group unframed
```

構造を認識する CEM インターフェイスの場合：

```
controller T1 0/0/2/0/1/7/1
 cem-group framed 0 timeslots 1
 cem-group framed 1 timeslots 2-3
 cem-group framed 2 timeslots 4-6
 cem-group framed 3 timeslots 7-10
 cem-group framed 4 timeslots 11-15
 cem-group framed 5 timeslots 16-21
 cem-group framed 6 timeslots 22-24
```

SDH - E1 チャネル化と CEM インターフェイス作成

```
hw-module subslot  <loc> cardtype sdh
controller SONET 0/0/2/0
 au 1
  mode tug3
  width 3
  tug3 1
   mode c12-e1
tug3 2
   mode c12-e1
 tug3 3
   mode c12-e1
commit
```

構造を認識しない CEM インターフェイスの場合：

```
controller E1 0/0/2/0/1/1/1/1
 cem-group unframed
```

構造を認識する CEM インターフェイスの場合：

```
controller E1 0/0/2/0/1/1/7/1
 cem-group framed 0 timeslots 1
 cem-group framed 1 timeslots 2-3
 cem-group framed 2 timeslots 4-6
 cem-group framed 3 timeslots 7-10
 cem-group framed 4 timeslots 11-15
 cem-group framed 5 timeslots 16-21
 cem-group framed 6 timeslots 22-31
```

CEM インターフェイス コンフィギュレーション

```
RP/0/RSP0/CPU0:CEOP-01#show runn interface cem 0/0/2/0/1/1/1/1:1

interface CEM0/0/2/0/1/1/1/1:1
 l2transport
 !
CEM Interface Config Options :

RP/0/RSP0/CPU0:CEOP-01(config)#interface cem 0/0/2/0/1/1/1/1:1
RP/0/RSP0/CPU0:CEOP-01(config-if)#cem ?
  class-attach  Attach a CEM class to this interface
  clock         Configure clocks on this CEM interface
  dejitter      Configure dejitter buffer
  dummy         Configure dummy frame parameters
  idle          Configure idle frame parameters
  payload       Configure payload size of CEM frames
```

Cisco 2 ポート チャネライズド T3/E3 回線エミュレーションおよびチャネライズド ATM SPA での T3/E3 の SAToP CEM インターフェイス作成

```
RP/0/0/CPU0:router(config)#controller t3 0/4/2/0
RP/0/0/CPU0:router(config-t3)#cem-group ?
  unframed  clear channel carrying CEM
RP/0/0/CPU0:router(config-t3)#cem-group unframed
RP/0/0/CPU0:router(config-t3)#commit
RP/0/0/CPU0:router(config-t3)#
```

Cisco 2 ポート チャネライズド T3/E3 回線エミュレーションおよびチャネライズド ATM SPA での T1/E1 の SAToP CEM インターフェイス作成

```
RP/0/0/CPU0:router(config)#controller t3 0/4/2/0
RP/0/0/CPU0:router(config-t3)#mode ?
  atm     clear channel carrying atm
  e1      channelize into 21 E1s
  serial  clear channel carrying hdlc like payload
  t1      channelized into 28 T1s
RP/0/0/CPU0:router(config-t3)#mode e1
RP/0/0/CPU0:router(config-t3)#commit

RP/0/0/CPU0:router(config)#controller e1 0/4/2/0/1
RP/0/0/CPU0:router(config-e1)#cem-group ?
  framed    Configure a framed CEM interface on T1/E1
  unframed  Configure a unframed CEM interface on T1/E1
RP/0/0/CPU0:router(config-e1)#cem-group unframed ?
  <cr>
RP/0/0/CPU0:router(config-e1)#cem-group unframed
RP/0/0/CPU0:router(config-e1)#commit
```

Cisco 2 ポート チャネライズド T3/E3 回線エミュレーションおよびチャネライズド ATM SPA での T1/E1 の CESoPSN CEM インターフェイス作成

```
RP/0/0/CPU0:router(config)#controller t3 0/4/2/1
RP/0/0/CPU0:router(config-t3)#mode ?
  atm     clear channel carrying atm
  e1      channelize into 21 E1s
```

```
  serial  clear channel carrying hdlc like payload
  t1      channelized into 28 T1s
RP/0/0/CPU0:router(config-t3)#mode t1
RP/0/0/CPU0:router(config-t3)#commit

RP/0/0/CPU0:router(config)#controller t1 0/4/2/1/1
RP/0/0/CPU0:router(config-t1)#cem-group ?
  framed    Configure a framed CEM interface on T1/E1
  unframed  Configure a unframed CEM interface on T1/E1
RP/0/0/CPU0:router(config-t1)#cem-group framed ?
  <0-23>  CEM group number
RP/0/0/CPU0:router(config-t1)#cem-group framed 0 ?
  timeslots  List of timeslots in the CEM group
RP/0/0/CPU0:router(config-t1)#cem-group framed 0 timeslots ?
  WORD  timeslot string seprated by (:) or (-) from 1 to 24. (:) indicates individual
timeslot and (-) represent range
RP/0/0/CPU0:router(config-t1)#cem-group framed 0 timeslots 1:23
RP/0/0/CPU0:router(config-t1)#commit
```

Cisco 24 ポート チャネライズド T1/E1 回線エミュレーションおよびチャネライズド ATM SPA での T1/E1 の SAToP CEM インターフェイス作成

```
RP/0/0/CPU0:router(config)#controller e1 0/4/1/2
RP/0/0/CPU0:router(config-e1)#cem-group ?
  framed    Configure a framed CEM interface on T1/E1
  unframed  Configure a unframed CEM interface on T1/E1
RP/0/0/CPU0:router(config-e1)#cem-group unframed ?
  <cr>
RP/0/0/CPU0:router(config-e1)#cem-group unframed
RP/0/0/CPU0:router(config-e1)#commit
```

Cisco 24 ポート チャネライズド T1/E1 回線エミュレーションおよびチャネライズド ATM SPA での T1/E1 の CESoPSN CEM インターフェイス作成

```
RP/0/0/CPU0:router(config)#controller e1 0/4/1/1
RP/0/0/CPU0:router(config-e1)#cem-group framed ?
  <0-30>  CEM group number
RP/0/0/CPU0:router(config-e1)#cem-group framed 1 ?
  timeslots  List of timeslots in the CEM group
RP/0/0/CPU0:router(config-e1)#cem-group framed 1 timeslots ?
  WORD  timeslot string seprated by (:) or (-) from 1 to 31. (:) indicates individual
timeslot and (-) represent range
RP/0/0/CPU0:router(config-e1)#cem-group framed 1 timeslots 1:20
RP/0/0/CPU0:router(config-e1)#commit
```

クロック回復：例

適応クロック回復の設定：

（E1 の設定は、次に示す T1 と同様です）

```
CE1
----
Router (config)#controller t1 0/0/2/0/1/1/4
Router (config-t1)#clock source internal
```

PE1 (Acts as source of clock, but no specific configuration under CEM Interface is needed here)

```
Router (config)#controller t1 0/0/2/0/1/1/4
Router (config-t1)#clock source line
```

PE2 (On PE node where clock recovery is done):

適応クロックを回復するには、次のようにします。

```
Router(config)# interface cem 0/0/2/0/1/1/4:0
Router(config-if)#cem clock recover <clock-id> adaptive
```

再生クロックを適用するには、次のようにします。

```
Router (config)#controller t1 0/0/2/0/1/1/4
Router (config-t1)#clock source recovered <clock-id>
```

CE2

```
Router (config)#controller t1 0/0/2/0/1/1/4
Router (config-t1)#clock source line
```

差分クロック回復の設定：

CE1

```
Router (config)#controller t1 0/0/2/0/1/1/4
Router (config-t1)#clock source internal
```

PE1（クロックのソースとしての役割を持つ）

```
Router (config)#controller t1 0/0/2/0/1/1/4
Router (config-t1)#clock source line
Router(config)# interface cem 0/0/2/0/1/1/4:0
Router(config-if)#cem clock transmit differential
```

PE2（差分クロックを削除するため）：

```
Router (config)#interface cem 0/0/2/0/1/1/4:0
Router (config-t1)#cem clock recover <clock-id> differential
```

再生クロックを適用するには、次のようにします。

```
Router (config)#controller t1 0/0/2/0/1/1/4
Router (config-t1)#cem clock recovered <clock-id>
```

CE2

```
Router (config)#controller t1 0/0/2/0/1/1/4
Router (config-t1)#clock source line
```

その他の関連資料

次の項では、関連マニュアルの参照先を示します。

関連資料

関連項目	参照先
Cisco IOS XR マスター コマンド リファレンス	*『Cisco IOS XR Master Commands List』*
Cisco IOS XR インターフェイス コンフィギュレーション コマンド	*『Cisco ASR 9000 Series Aggregation Services RouterInterface and Hardware Component Command Reference』*
Cisco IOS XR ソフトウェアを使用するルータを初回に起動し設定するための情報	*『Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide』*
ユーザ グループとタスク ID に関する情報	『*Cisco IOS XR System Security Configuration Guide*』の「*Configuring AAA Services on Cisco IOS XR* Software」モジュール

標準

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

MIB

MIB	MIB のリンク
• CISCO-SONET-MIB • ENTITY-MIB • SONET-MIB（RFC 3592） これらの追加 MIB がサポートされるのは、Cisco ASR 9000 シリーズ ルータの Cisco 1 ポート チャネライズド OC-3/STM-1 SPA のみです。 • CISCO-IF-EXTENSION-MIB • DS1-MIB • DS3-MIB • IF-MIB	Cisco IOS XR ソフトウェアを使用して選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
RFC 5086、RFC 4553、RFC 4197、RFC 5287	• *『Structure-Aware Time Division Multiplexed (TDM) Circuit Emulation Service over Packet Switched Network (CESoPSN)』* • *『Structure-Agnostic Time Division Multiplexing (TDM) over Packet (SAToP)』* • *『Requirements for Edge-to-Edge Emulation of Time Division Multiplexed (TDM) Circuits over Packet Switching Networks』* • *『Control Protocol Extensions for the Setup of Time-Division Multiplexing (TDM) Pseudowires in MPLS Networks』*

シスコのテクニカル サポート

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

その他の関連資料

Cisco ASR 9000 シリーズ ルータ でのクリア チャネル SONET コントローラの設定

このモジュールでは、Cisco ASR 9000 シリーズ ルータのクリア チャネル SONET コントローラの設定について説明します。

SONET コントローラの設定は、ポイントツーポイント プロトコル（PPP）にシャーシ間ステートフル スイッチオーバー（ICSSO）を設定するため、およびCisco ASR 9000 シリーズ ルータでマルチリンク PPP（MLPPP）、チャネライズド SONET、またはシリアル インターフェイスを設定するための前提条件です。

SONET では、多重化されたデジタル トラフィックのため光信号と同期フレーム構造を定義できます。これは、米国規格協会（ANSI）T1.105、ANSI T1.106、および ANSI T1.117 で規定されている、光ネットワークのための速度と形式を定義した規格セットです。

チャネライズド SONET コントローラの設定の詳細については、「Cisco ASR 9000 シリーズ ルータ でのチャネライズド SONET/SDH の設定」モジュールを参照してください。

レイヤ 1 SONET コントローラを設定するためのコマンドは、『*Cisco IOS XR Interface and Hardware Component Command Reference*』を参照してください。

Cisco IOS XR ソフトウェアでの SONET コントローラの設定機能の履歴

リリース	変更内容
リリース 3.9.0	Cisco ASR 9000 シリーズ ルータ に関して、次の SPA のサポートが追加されました。 • Cisco 2 ポート チャネライズド OC-12c/DS0 SPA
リリース 4.0.0	次の SPA のサポートが Cisco ASR 9000 シリーズ ルータで導入されました。 • Cisco 1 ポート チャネライズド OC-48/STM-16 SPA • Cisco 1 ポート OC-192c/STM-64 POS/RPR XFP SPA • Cisco 2 ポート OC-48c/STM-16 POS/RPR SPA • Cisco 8 ポート OC-12c/STM-4 POS SPA
リリース 4.0.1	次の SPA のサポートが Cisco ASR 9000 シリーズ ルータで導入されました。 • Cisco 4 ポート OC-3c/STM-1 POS SPA • Cisco 8 ポート OC-3c/STM-1 POS SPA

内容

クリア チャネル SONET コントローラを設定するための前提条件

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。

SONET コントローラを設定する前に、次の作業が終了し条件が満たされていることを確認してください。

- 次のいずれかの SPA がインストールされていること。
  - Cisco 2 ポート チャネライズド OC-12c/DS0 SPA
  - Cisco 1 ポート チャネライズド OC-48/STM-16 SPA
  - Cisco 4 ポート OC-3c/STM-1 POS SPA
  - Cisco 8 ポート OC-3c/STM-1 POS SPA
  - Cisco 1 ポート OC-192c/STM-64 POS/RPR XFP SPA
  - Cisco 2 ポート OC-48c/STM-16 POS/RPR SPA
  - Cisco 8 ポート OC-12c/STM-4 POS SPA
- SONET コントローラ名とインスタンス ID を、汎用表記 *rack/slot/module/port* で指定する方法を知っていること。SONET コントローラ名とインスタンス ID は、**controller sonet** コマンドで必要です。

SONET コントローラの設定に関する情報

SONET コントローラを設定するには、次の概念について理解する必要があります。

SONET コントローラの概要

Cisco IOS XR ソフトウェアをサポートするルータでは、特定のラインカード上の物理ポートが「コントローラ」と呼ばれます。チャネライズド SONET またはシリアル インターフェイスを設定するには、その前に SONET コントローラを設定する必要があります。

物理 SONET ポートを設定するために使用するコマンドは、SONET コントローラ コンフィギュレーション モードにグループ化されています。SONET コントローラ コンフィギュレーション モードを開始するには、グローバル コンフィギュレーション モードで **controller sonet** コマンドを入力します。**controller preconfigure sonet** グローバル コンフィギュレーション コマンドを使用して SONET コントローラを事前設定することもできます。

ルータは SONET コントローラをレイヤ 1 およびレイヤ 2 の処理で使用します。

SONET コントローラのデフォルト設定値

表 9 は、SONET コントローラに存在するデフォルト コンフィギュレーション パラメータの一部を説明したものです。

表 9　SONET コントローラのデフォルト設定値

パラメータ	デフォルト値	設定ファイルのエントリ
SONET コントローラの次のアラームのレポート • ビット 1（B1）ビット エラー レート（BER）しきい値超過アラート（TCA）エラー • ビット 2（B2）BER TCA エラー • 信号障害 BER エラー • セクション フレーム損失（SLOF）エラー • セクション信号消失（SLOS）エラー	**enabled**	デフォルトでイネーブルになっているアラームのレポートをディセーブルにするには、SONET/SDH コンフィギュレーション モードで **no report** [**b1-tca** | **b2-tca** | **sf-ber** | **slof** | **slos**] コマンドを使用します。 ラインのアラーム表示信号（LAIS）、ラインのリモート障害表示（LRDI）、信号劣化 BER エラーのレポートをイネーブルにするには、SONET/SDH コンフィギュレーション モードで **report** [**lais** | **lrdi** | **sd-ber**] コマンドを使用します。
SONET パス コントローラの次のアラームのレポート • ビット 3（B3）BER TCA エラー • パス ポインタ損失（PLOP）エラー	**enabled**	SONET パス コントローラ上で B3 BER TCA または PLOP レポートをディセーブルにするには、SONET/SDH パス コンフィギュレーション サブモードで **no report b3-tca** コマンドまたは **no report plop** コマンドを入力します。 パス アラーム検出信号（PAIS）、パス ペイロード ミスマッチ（PPLM）、パス リモート障害検出（PRDI）、Path Trace Identity Mismatch（PTIM）エラーのレポートをイネーブルにするには、SONET/SDH パス コンフィギュレーション サブモードで **report** [**pais** | **pplm** | **prdi** | **ptim**] コマンドを使用します。

表 9　SONET コントローラのデフォルト設定値 （続き）

パラメータ	デフォルト値	設定ファイルのエントリ
同期ペイロード エンベロープ（SPE）スクランブリング	**enabled**	SONET コントローラで SPE スクランブリングをディセーブルにするには、SONET コントローラ コンフィギュレーション サブモードで **path scrambling disable** コマンドを入力します。
キープアライブ タイマー	**enabled**	キープアライブ タイマーをオフにするには、インターフェイス コンフィギュレーション モードで **keepalive disable** コマンドを入力します。

SONET APS

自動保護スイッチング（APS）機能を使用すると、障害時にインターフェイスのスイッチオーバーが可能になります。この機能は、SONET 機器を交換機器に接続する際に多くの場合必要になります。APS は、SONET ネットワーク内の*保護*インターフェイスを*現用*インターフェイスのバックアップとして使用するメカニズムです。現用インターフェイスに障害が発生した場合、保護インターフェイスが即座にそのトラフィック負荷を引き継ぎます。現用インターフェイスとその保護インターフェイスが、*APS グループ*を構成します。

Cisco IOS XR ソフトウェアで、SONET APS コンフィギュレーションにより、各冗長回線ペアの現用回線と保護回線が定義されます。現用回線は、プライマリ（優先される）回線であり、回線が動作可能であるかぎり、通信はその回線上で行われます。現用回線が障害になった場合、APS は保護回線へのスイッチオーバーを開始します。2 つのルータ間で APS が適切に動作するためには、1 つのルータの現用回線が、他のルータでも現用回線になっている必要があります。同じことは、保護回線にも当てはまります。

SONET APS グループで、各接続は双方向または単方向、リバーティブまたは非リバーティブです。同じ信号ペイロードが現用インターフェイスと保護インターフェイスに送信されます。現用インターフェイスと保護インターフェイスの終端は、2 台の異なるルータです。

保護インターフェイスは、劣化、チャネル信号消失、手動介入の際に、アクティブ化するか非アクティブ化するかを現用インターフェイスに対して指示します。現用インターフェイスと保護インターフェイスの間の通信が失われた場合、現用ルータは、保護回線が存在しないかのように、現用インターフェイスの完全な制御を引き受けます。

APS グループでは、各回線は*チャネル*と呼ばれます。双方向モードでは、受信および送信チャネルはペアとして切り替わります。単一方向モードでは、送信および受信チャネルは個別に切り替わります。たとえば、双方向モードで、現用インターフェイスの受信チャネルでチャネル信号消失が発生した場合、受信チャネルと送信チャネルの両方が切り替えられます。

クリア チャネル SONET コントローラの設定方法

ここでは、次の手順について説明します。

- 「クリア チャネル SONET コントローラの設定」（P.422）
- 「SONET APS の設定」（P.426）
- 「Fast Reroute がトリガーされないように hold-off タイマーを設定する」（P.431）

クリア チャネル SONET コントローラの設定

ここでは、POS またはシリアル インターフェイスを設定するための前提条件として SONET コントローラを設定する方法について説明します。

前提条件

- サポートされる POS SPA またはチャネライズド SPA がルータに搭載されており、そのルータでは、対応するサポート対象の Cisco IOS XR ソフトウェア リリースが実行されている必要があります。
- ファイバ障害または機器障害から回復できるようにするには、「SONET APS の設定」（P.-426）の説明に従って、ルータ上で SONET APS を設定します。

手順の概要

1. **configure**
2. **controller sonet** *interface-path-id*
3. **clock source** {**internal** | **line**}
4. **line delay trigger** *value*
5. **line delay clear** *value*
6. **framing** {**sdh** | **sonet**}
7. **loopback** {**internal** | **line**}
8. **overhead** {**j0** | **s1s0**} *byte-value*
9. **path** *keyword* [*values*]
10. **end**
    または
    commit
11. **show controllers sonet** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller sonet** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller sonet 0/1/0/0`	SONET コントローラ コンフィギュレーション サブモードを開始し、SONET コントローラ名とインスタンス ID を、*rack/slot/module/port* 表記で指定します。
ステップ 3	**clock source** {**internal** | **line**} **例：** `RP/0/RSP0/CPU0:router(config-sonet)# clock source internal`	SONET ポート送信クロック ソースを設定します。**internal** キーワードを指定すると内部クロックが設定され、**line** キーワードを指定すると回線から再生したクロックが設定されます。 • ネットワークからクロッキングを得られる場合は、必ず **line** キーワードを使用します。**internal** キーワードは、2 台のルータがバックツーバックで接続されているか、クロッキングが利用できないファイバで接続されている場合に使用します。 **（注）** デフォルトは **line** クロックです。
ステップ 4	**line delay trigger** *value* **例：** `RP/0/RSP0/CPU0:router(config-sonet)# line delay trigger 3000`	（任意）SONET 回線遅延トリガー値を設定します。トリガー値の範囲は 0 ～ 60000 ミリ秒で、デフォルトの遅延トリガー値は 0 ミリ秒です。
ステップ 5	**line delay clear** *value* **例：** `RP/0/RSP0/CPU0:router(config-sonet)# line delay clear 4000`	（任意）SONET 回線遅延トリガー アラームがクリアされるまでの時間を設定します。範囲は 1000 ～ 180000 ミリ秒で、デフォルトは 10 秒です。

	コマンドまたはアクション	目的
ステップ 6	**framing** {**sdh** | **sonet**} **例：** RP/0/RSP0/CPU0:router(config-sonet)# framing sonet	（任意）同期デジタル ハイアラーキ（SDH）フレーム構成の場合は **sdh** キーワード、SONET フレーム構成の場合は **sonet** キーワードを使用して、コントローラのフレーム構成を設定します。 SONET フレーム構成（**sonet**）がデフォルトです。
ステップ 7	**loopback** {**internal** | **line**} **例：** RP/0/RSP0/CPU0:router(config-sonet)# loopback internal	（任意）ループバック用の SONET コントローラを設定します。**internal** キーワードは、内部（ターミナル）ループバックを選択し、**line** キーワードは回線（ファシリティ）ループバックを選択します。
ステップ 8	**overhead** {**j0** | **s1s0**} *byte-value* **例：** RP/0/RSP0/CPU0:router(config-sonet)# overhead s1s0	（任意）コントローラのオーバーヘッドを設定します。**j0** キーワードは STS ID（J0/C1）バイトを指定し、**s1s0** キーワードは H1 バイトのビット s1 および s0 を指定します。 • **j0** キーワードのデフォルト バイト値は 0xcc、**s1s0** キーワードのデフォルト バイト値は 0 です。 • **j0** と **s1s0** の有効な値の範囲は 0 ～ 255 です。

	コマンドまたはアクション	目的
ステップ 9	**path** *keyword* [*values*] **例：** RP/0/RSP0/CPU0:router(config-sonet)# path delay trigger 25	（任意）SONET コントローラ パス値を設定します。 キーワードの定義は次のとおりです。 • **ais-shut**：シャットダウン時のパス アラーム検出信号（PAIS）の送信を設定します。 • **b3-ber-prdi**：ビット エラー レート（BER）ビット インターリーブ パリティ（BIP）しきい値を超えたときの、パス レベル リモート故障表示（PRDI）送信をイネーブルにします。 • **delay clear** *value*：同期転送信号（STS）パス遅延トリガー アラームがクリアされるまでの時間の長さを設定します。*value* 引数には、0 ～ 180000 ミリ秒の範囲の数値を指定します。デフォルト値は 10 秒です。 • **delay trigger** *value*：SONET パス遅延または遅延トリガー値を設定します。*value* 引数には、0 ～ 60000 ミリ秒の範囲の数値を指定します。デフォルト値は 0 ミリ秒です。 • **overhead** [**c2** *byte-value* | **j1** *line*]：SONET POH バイト値またはビット値を設定します。STS SPE コンテント（C2）バイトを指定するには **c2** キーワードを入力し、*byte-value* 引数に 0 ～ 255 の範囲の数値を指定します。SONET パス トレース（J1）バッファを設定するには、**j1** キーワードを入力し、*line* 引数にパス トレース バッファ ID を（ASCII テキストで）指定します。 • **report [b3-tca** | **pais** | **plop** | **pplm** | **prdi** | **ptim]**：SONET パス アラーム レポートを設定します。レポートするアラームと、アラームを発生させるビット エラー レート（BER）しきい値を指定します。デフォルトでは、B3 BER しきい値超過アラート（TCA）とパス ポインタ損失（PLOP）レポートがイネーブルになっています **pais** キーワードを指定すると、PAIS レポート ステータスが設定されます。**pplm** は、パス ペイロード ミスマッチ（PPLM）障害レポート ステータスを設定します。**prdi** は、パス リモート障害検出（PRDI）レポート ステータスを設定します。**ptim** は、Path Trace Identity Mismatch（PTIM）障害レポート ステータスを設定します。 SONET/SDH パス コンフィギュレーション サブモードで **no report b3-tca** コマンドおよび **no report plop** コマンドを入力すると、それぞれ B3 BER TCA および PLOP レポート ステータスがディセーブルになります。 • **scrambling disable**：SPE スクランブリングをディセーブルにします。SPE スクランブリングはデフォルトでイネーブルになっています。 • **threshold b3-tca** *BER*：SONET パス BER しきい値を設定します。*BER* 引数には、3 ～ 9 の範囲の数値を指定します。しきい値は、ビット誤り率を決定するときに、10 の負の指数として解釈されます。たとえば、値 5 は、ビット誤り率が 10 のマイナス 5 乗であることを意味します。デフォルトの BER しきい値は 6 です。 • **uneq-shut**：シャットダウン時の未実装（UNEQ）の送信を設定します。

	コマンドまたはアクション	目的
ステップ 10	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-sonet)# end` または `RP/0/RSP0/CPU0:router(config-sonet)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 11	**show controllers sonet** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show controllers sonet 0/1/0/0`	SONET コントローラの設定を確認します。

SONET APS の設定

SONET APS は、SONET 回線層でファイバ（外部）障害または機器（インターフェイスおよび内部）障害からの回復機能を提供します。ここでは、ルータで基本的な自動保護スイッチング（APS）を設定する方法と、**aps group** コマンドを使用して、ルータ上で複数の保護インターフェイスまたは現用インターフェイスを設定する方法について説明します。

コンフィギュレーションを確認する場合や、スイッチオーバーが発生したかどうかを確認するには、**show aps** コマンドを使用します。

前提条件

SONET APS を設定する前に、サポートされるチャネライズド SPA がルータに搭載されており、そのルータで Cisco IOS XR ソフトウェアが実行されていることを確認してください。

Cisco ASR 9000 シリーズ ルータでは、2 ポート チャネライズド OC-12c/DS0 SPA が搭載されている必要があります。

制約事項

SONET APS を設定する前に、次の制約事項を考慮します。

- Cisco ASR 9000 シリーズ ルータ上の POS SPA は、シングル ルータとマルチルータのどちらの APS もサポートしません。

- Cisco ASR 9000 シリーズ ルータは、マルチルータ APS を 2 ポート チャネライズド OC-12/DS0 SPA 上でサポートします。
- 2 つのルータ間で APS が適切に動作するためには、1 つのルータの現用回線が、他のルータでも現用回線になっている必要があります。同じことは、保護回線にも当てはまります。

手順の概要

1. **configure**
2. **aps group** *number*
3. **channel** {**0** | **1**} **local sonet** *interface*
4. APS グループ内の各チャネルに対してステップ 3 を繰り返します。
5. **exit**
6. **interface loopback** *number*
7. **ipv4 address** *ip-address mask*
8. **exit**
9. **interface pos** *interface-path-id* または
   interface serial *interface-path-id*
10. **ipv4 address** *ip-address mask*
11. **pos crc** {**16** | **32**}
    または
    crc {**16** | **32**}
12. **encapsulation** {**frame-relay** | **hdlc** | **ppp**}（シリアル インターフェイスのみ）
13. **keepalive** {*interval* | **disable**}[*retry*]
14. **no shutdown**
15. グループ内の各チャネルに対してステップ 9 ～ 13 を繰り返します。
16. **exit**
17. **controller sonet** *interface-path-id*
18. **ais-shut**
19. **path scrambling disable**
20. **clock source** {**internal** | **line**}
21. グループの各チャネルに対してステップ 16 ～ 19 を繰り返します。
22. **end**
    または
    commit
23. **exit**
24. **exit**
25. **show aps**
26. **show aps group** [*number*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**aps group** *number* **例：** `RP/0/RSP0/CPU0:router(config)# aps group 1`	指定した番号を持つ APS グループを追加して、APS グループ コンフィギュレーション モードを開始します。 • **aps group** コマンドは、グローバル コンフィギュレーション モードで使用します。 • グループを削除するには、このコマンドの **no** 形式を使用します。たとえば、**no aps group** *number* のように入力します。ここで、値の範囲は 1 ～ 255 です。 **（注）** **aps group** コマンドを使用するには、aps コマンドの適切なタスク ID に関連付けられたユーザ グループのメンバーでなければなりません。 **（注）** **aps group** コマンドは、設定する保護グループが 1 つだけの場合でも使用します。
ステップ 3	**channel** {**0** | **1**} **local sonet** *interface* **例：** `RP/0/RSP0/CPU0:router(config-aps)# channel 0 local SONET 0/0/0/1`	APS グループのチャネルを作成します。**0** は保護チャネルを指定し、**1** は現用チャネルを指定します。 **（注）** 保護チャネルがローカルな場合、現用チャネルを割り当てる*前に*、**channel** コマンドを使用して割り当てる必要があります。
ステップ 4	グループ内の各チャネルに対してステップ 3 を繰り返します	—
ステップ 5	**exit**	APS グループ コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードを開始します。
ステップ 6	**interface loopback** *number* **例：** `RP/0/RSP0/CPU0:router(config)# interface loopback 1`	（任意）2 台のルータによる APS が望ましい場合にループバック インターフェイスを設定し、ループバック インターフェイスのインターフェイス コンフィギュレーション モードを開始します。 **（注）** この例で、ループバック インターフェイスは相互接続として使用されています。
ステップ 7	**ipv4 address** *ip-address mask* **例：** `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 172.18.0.1 255.255.255.224`	ループバック インターフェイスに IPV4 アドレスおよびサブネット マスクを割り当てます。
ステップ 8	**exit**	ループバック インターフェイスのインターフェイス コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 9	**interface pos** *interface-path-id* または **interface serial** *interface-path-id* **例：** RP/0/RSP0/CPU0:router(config)# interface POS 0/2/0/0 または RP/0/RSP0/CPU0:router(config)# interface serial 0/1/1/0/0/0:0	ステップ 3 で選択したチャネルのインターフェイスを接続し、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。 シリアル インターフェイスの場合は、完全なインターフェイス番号を *rack/slot/module/port/T3Num/T1num:instance* 表記で指定します。
ステップ 10	**ipv4 address** *ip-address mask* **例：** RP/0//CPU0:router(config-if)# ipv4 address 172.18.0.1 255.255.255.224	IPv4 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。
ステップ 11	**pos crc** {**16** | **32**} または **crc** {**16** | **32**} **例：** RP/0/RSP0/CPU0:router(config-if)# pos crc 32 または RP/0/RSP0/CPU0:router(config-if)# crc 32	チャネルの CRC 値を選択します。16 ビットの CRC モードを指定するには **16** キーワード、**32** ビットの CRC モードを指定するには 32 キーワードを入力します。POS インターフェイスの場合は、デフォルト CRC は 32 です。シリアル インターフェイスの場合は、デフォルトは 16 です。
ステップ 12	**encapsulation** {**frame-relay** | **hdlc** | **ppp**} **例：** RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp	（シリアル インターフェイスのみ）インターフェイスでのレイヤ 2 カプセル化を設定します。
ステップ 13	**keepalive** {*interval* | **disable**}[*retry*] **例：** RP/0/RSP0/CPU0:router(config-if)# keepalive disable	チャネルのキープアライブ タイマーを設定します。ここで、 • *interval*：キープアライブ メッセージ間の秒数（1 ～ 30）。デフォルトは 10 です。 • **disable**：キープアライブ タイマーをオフにします。 • *retry*：（任意）リンクがダウン状態に遷移する前に、応答なしでピアに送信できるキープアライブ メッセージの数（1 ～ 255）。デフォルトは、PPP カプセル化を使用するインターフェイスの場合は 5、HDLC カプセル化を使用するインターフェイスの場合は 3 です。 **keepalive** コマンドは、フレーム リレー カプセル化を使用するインターフェイスには適用されません。
ステップ 14	**no shutdown** **例：** RP/0/RSP0/CPU0:router(config-if)# no shutdown	shutdown 設定を削除します。 • shutdown 設定を削除すると、インターフェイスで強制管理ダウンが削除され、インターフェイスがアップまたはダウン状態に移行できるようになります（親 SONET レイヤが管理ダウンに設定されていないことが前提です）。
ステップ 15	グループ内の各チャネルに対してステップ 9 ～ 13 を繰り返します。	—

	コマンドまたはアクション	目的
ステップ 16	**exit**	インターフェイス コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードを開始します。
ステップ 17	**controller sonet** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller sonet 0/1/0/0`	SONET コントローラ コンフィギュレーション モードを開始し、SONET コントローラ名とインスタンス ID を *rack/slot/module/port* 表記で指定します。
ステップ 18	**ais-shut** **例：** `RP/0/RSP0/CPU0:router(config-sonet)# ais-shut`	シャットダウン時のアラーム表示信号（AIS）など、SONET パス値を設定します。
ステップ 19	**path scrambling disable** **例：** `RP/0/RSP0/CPU0:router(config-sonet)# path scrambling disable`	（任意）同期ペイロード エンベロープ（SPE）のスクランブリングをディセーブルにします。 **（注）** SPE スクランブリングはデフォルトでイネーブルです。
ステップ 20	**clock source** {**internal** | **line**} **例：** `RP/0/RSP0/CPU0:router(config-sonet)# clock source internal`	SONET ポートの TX クロック ソースを設定します。**internal** キーワードを指定すると内部クロックが設定され、**line** キーワードを指定すると回線から再生されたクロックが設定されます。 • クロッキングをネットワークから得る場合には、必ず **line** キーワードを使用します。**internal** キーワードは、2 台のルータがバックツーバックで接続されているか、クロッキングが利用できないファイバで接続されている場合に使用します。 • デフォルトは回線クロック（**line**）です。
ステップ 21	グループ内の各チャネルに対してステップ 16 ～ 19 を繰り返します。	—

	コマンドまたはアクション	目的
ステップ 22	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-sonet)# end` または `RP/0/RSP0/CPU0:router(config-sonet)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 23	**exit**	SONET コントローラ コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードを開始します。
ステップ 24	**exit**	グローバル コンフィギュレーション モードを終了し、EXEC モードを開始します。
ステップ 25	**show aps** **例：** `RP/0/RSP0/CPU0:router# show aps`	（任意）設定済みのすべての SONET APS グループの動作ステータスを表示します。
ステップ 26	**show aps group** [*number*] **例：** `RP/0/RSP0/CPU0:router# show aps group 3`	（任意）設定済みの SONET APS グループの動作ステータスを表示します。 **（注）** 複数のグループを定義する場合は、**show aps group** コマンドのほうが **show aps** コマンドよりも有用です。

Fast Reroute がトリガーされないように hold-off タイマーを設定する

APS がルータ上で設定されている場合、トンネルに対する保護が提供されません。この制限のため、Fast Reroute（FRR）がいまだにマルチプロトコル ラベル スイッチング（MPLS）トラフィック エンジニアリングの保護メカニズムになっています。

APS が SONET コア ネットワーク中で設定されている場合、ルータのダウンストリームに向けてアラームが生成されることがあります。ルータのダウンストリームで FRR が設定されている場合は、SONET レベルで hold-off タイマーを設定し、CORE ネットワークの復元中に FRR がトリガーされないようにしたいことがあります。遅延を設定するにはこの作業を実施します。

前提条件

「SONET APS の設定」（P.426）に従って SONET APS を設定します。

手順の概要

1. **configure**
2. **controller sonet** *interface-path-id*
3. **line delay trigger** *value*
   または
   path delay trigger *value*
4. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller sonet** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller sonet 0/6/0/0`	SONET 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	**line delay trigger** *value* または **path delay trigger** *value* **例：** `RP/0/RSP0/CPU0:router(config-sonet)# line delay trigger 250` または `RP/0/RSP0/CPU0:router(config-sonet)# path delay trigger 300`	SONET ポート遅延トリガー値をミリ秒単位で設定します。 **ヒント** ステップ 2 とステップ 3 のコマンドは、1 つのコマンド文字列に結合して、グローバル コンフィギュレーション モードから **controller sonet** *r/s/m/p* **line delay trigger** または **controller sonet** *r/s/m/p* **path delay trigger** のように入力できます。
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-sonet)# end` または `RP/0/RSP0/CPU0:router(config-sonet)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

SONET コントローラの設定例

ここでは、次の例を示します。

- 「SONET コントローラの設定：例」（P.433）
- 「SONET APS グループの設定：例」（P.434）

SONET コントローラの設定：例

次に、「クリア チャネル SONET コントローラの設定」（P.422）で概要を示した手順の後で SONET コントローラの設定を行った場合のコマンドと出力の例を示します。この例は、すべてのオプション コマンドの使用方法と、コマンド内でのオプションの一覧を示しています。実際の設定では、これらのコマンドの一部を使用しない場合があります。

```
configure
controller sonet 0/1/0/0
```

```
    ais-shut
    clock source internal
    framing sonet
    loopback internal
Loopback is a traffic-effecting operation
    overhead s1s0 1
    path ais-shut
    path delay trigger 0
    path overhead j1 line 11
    path report pais
    path scrambling disable
    path threshold b3-tca 6
    path uneq-shut
    report pais
    threshold b2-tca 4
    commit
```

SONET APS グループの設定：例

次に、SONET リモート（2 台のルータ）APS の設定例を示します。

```
RP/0/0/CPU0:router(config)# aps group 1
    channel 0 local SONET 0/0/0/1
    channel 1 remote 172.18.69.123
    signalling sonet
    commit
show aps
show aps group 3
```

その他の関連資料

ここでは、SONET コントローラの設定に関する関連資料について説明します。

関連資料

関連項目	参照先
Cisco IOS XR マスター コマンド リファレンス	*『Cisco IOS XR Master Commands List』*
Cisco IOS XR インターフェイス コンフィギュレーション コマンド	*『Cisco IOS XR Interface and Hardware Component Command Reference』*
Cisco IOS XR ソフトウェアを使用するルータを初回に起動し設定するための情報	*『Cisco IOS XR Getting Started Guide』*
ユーザ グループとタスク ID に関する情報	『*Cisco IOS XR System Security Configuration Guide*』の「*Configuring AAA Services on Cisco IOS XR* Software」モジュール

標準

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

MIB

MIB	MIB のリンク
このモジュールに適用できる MIB はありません。	Cisco IOS XR ソフトウェアを使用して選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	—

シスコのテクニカル サポート

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

その他の関連資料

Cisco ASR 9000 シリーズ ルータでのクリア チャネル T3/E3 およびチャネライズド T3 および T1/E1 コントローラの設定

このモジュールでは、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータでのクリア チャネル T3/E3 コントローラおよびチャネライズド T3 および T1/E1 コントローラの設定について説明します。

関連付けられたシリアル インターフェイスを設定する前に、T3/E3 コントローラを設定する必要があります。

T3/E3 コントローラ インターフェイス設定の機能履歴

リリース	変更内容
リリース 3.9.0	この機能が Cisco ASR 9000 シリーズ ルータで Cisco 2 ポート チャネライズド OC-12c/DS0 SPA に対して追加されました。
リリース 4.0.0	次の機能のサポートが、Cisco 2 ポート チャネライズド OC-12c/DS0 SPA に対して追加されました。 • NxDS0 チャネライゼーション • リンク ノイズ モニタ 1 ポート チャネライズド OC-48/STM-16 SPA でクリア チャネル T3 コントローラのサポートが導入されました。
リリース 4.0.1	次の SPA のサポートが追加されました。 • Cisco 1 ポート チャネライズド OC-3/STM-1 SPA • Cisco 2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA
リリース 4.1.0	• 次の SPA のサポートが追加されました。 – Cisco 4 ポート チャネライズド T3/DS0 SPA – Cisco 8 ポート チャネライズド T1/E1 SPA • Cisco 2 ポート チャネライズド OC-12c/DS0 SPA での T1/E1 リンクでのノイズ エラーのしきい値設定のためのリンク ノイズ モニタ拡張機能のサポート。これは、PPP にノイズ属性を通知して MLPPP バンドル リンクを削除させるために使用されます。

内容

T3/E3 コントローラ設定の前提条件

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。

T3/E3 コントローラを設定する前に、次のサポートされている SPA の 1 つがルータにインストールされていることを確認してください。

- Cisco 2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA
- Cisco 4 ポート チャネライズド T3/DS0 SPA

（注） 4 ポート チャネライズド T3/DS0 SPA は、クリア チャネル モードで動作することも、チャネル化して 28 個の T1 コントローラまたは 21 個の E1 コントローラとすることもできます。

- Cisco 1 ポート チャネライズド OC-3/STM-1 SPA
- Cisco 2 ポート チャネライズド OC-12c/DS0 SPA
- Cisco 1 ポート チャネライズド OC-48/STM-16 SPA
- Cisco 8 ポート チャネライズド T1/E1 SPA
- チャネライズド SONET SPA にクリア チャネル T3 コントローラを設定する前に、T3 にチャネル化された STS ストリーム用に SPA を設定する必要があります。詳細については、「Cisco ASR 9000 シリーズ ルータ でのチャネライズド SONET/SDH の設定」モジュールを参照してください。

T3/E3 コントローラおよびシリアル インターフェイスに関する情報

2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA は、シリアル ライン上でのみ、クリア チャネル サービスをサポートします。4 ポート チャネライズド T3/DS0 SPA は、クリア チャネル サービスおよびチャネライズド シリアル ラインをサポートします。コントローラがチャネル化されない場合, このコントローラはクリア チャネル コントローラとなり、関連付けられたシリアル ラインの全帯域幅がシリアル サービスを伝送する単一のチャネル専用となります。

T3 コントローラがチャネル化されると、より小さい帯域幅の T1 または E1 コントローラに論理的に分割されます。どちらのコントローラに分割されるかは、選択したチャネル化のモードによって決まります。T1 または E1 コントローラのシリアル インターフェイスの帯域幅の合計は、チャネル化された T1 または E1 コントローラを含む T3 コントローラの帯域幅を超過できません。

T3 コントローラをチャネル化すると、T1 または E1 の各コントローラは自動的にさらに DS0 タイムスロットにチャネル化されます。単一の T1 コントローラは 24 DS0 タイムスロットを伝送し、単一の E1 コントローラは 31 DS0 タイムスロットを伝送します。ユーザは、これらの DS0 タイムスロットを個々のチャネル グループに分割できます。各チャネル グループはそれぞれ、単一のシリアル インターフェイスをサポートします。

コントローラがチャネル化され、チャネル グループが作成されると、サービスは関連付けられたシリアル インターフェイスでプロビジョニングされます。

このリリースのチャネル化機能では、次のタイプのチャネルにチャネル化することができます。

- 単一の T3 コントローラを 28 個の T1 コントローラにチャネル化（コントローラ サイズ合計は 44210 kbps）。
- 単一の T3 コントローラを 21 E1 コントローラにチャネル化（コントローラ サイズ合計は 43008 kbps）。
- 単一の T1 コントローラで最大 1.536 MB がサポートされます。
- 単一の E1 コントローラで最大 2.048 MB がサポートされます。

（注） 単一の共有ポート アダプタ（SPA）は、最大 448 チャネル グループをサポートできます。

ここでは、次の内容について説明します。

サポートされる機能

表 10 に、サポートされる主な機能の要約を SPA のタイプ別に示します。

表 10　チャネライズド T3/E3、T1/E1、およびクリア チャネル SPA でサポートされる機能

	1 ポート チャネライズド OC-3/STM-1 SPA	2 ポート チャネライズド OC-12c/DS0 SPA	1 ポート チャネライズド OC-48/STM-16 SPA	4 ポート チャネライズド T3/DS0 SPA	8 ポート チャネライズド T1/E1 SPA	2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA
ビット誤り率テスト（BERT）	T3、T1、E3、E1、および DS0 チャネル 最大 12 セッション[1] T1 の場合は最大 1 セッション	T3 チャネル	T3 および E3 STS-12 ごとに、最大 2 つの同時 BERT テストが可能。	T3、T1、E1、および DS0 チャネル	T1、E1、DS0 チャネル	T3 および E3 ポートごとに 1 セッション
チャネライゼーションとクリア チャネル モード	チャネライズド SONET/SDH DS0 へのチャネライズド T1/E1 クリア チャネル SONET シリアル インターフェイスに対しては SDH モードのクリア チャネル T3 および E3	チャネライズド SONET/SDH チャネライズド T3/E3 DS0 へのチャネライズド T1/E1 クリア チャネル SONET	チャネライズド SONET/SDH チャネライズド T3/E3 クリア チャネル SONET	チャネライズド T3 チャネライズド T1/E1 T3 クリア チャネル	DS0 へのチャネライズド T1/E1。 クリア チャネル T1 および E1	クリアチャネル T3 または E3 のみ
DSU モード	Adtran Digital-link Cisco Kentrox Larscom Verilink **E3**： Cisco（デフォルト） Digital Link Kentrox	Adtran Digital-link Cisco Kentrox Larscom Verilink	Adtran Digital-link シスコ Kentrox Larscom Verilink **（注）** E3 のサブレートはサポートされません。	Adtran Digital-link Cisco Kentrox Larscom Verilink	Adtran Digital-link Cisco Kentrox Larscom Verilink	Adtran Digital-link Cisco Kentrox Larscom Verilink
カプセル化	フレーム リレー HDLC PPP	HDLC PPP	フレーム リレー HDLC PPP	フレーム リレー HDLC PPP	フレーム リレー HDLC PPP	フレームリレー HDLC PPP

表 10　チャネライズド T3/E3、T1/E1、およびクリア チャネル SPA でサポートされる機能 （続き）

	1 ポート チャネライズド OC-3/STM-1 SPA	2 ポート チャネライズド OC-12c/DS0 SPA	1 ポート チャネライズド OC-48/STM-16 SPA	4 ポート チャネライズド T3/DS0 SPA	8 ポート チャネライズド T1/E1 SPA	2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA
等コスト マルチパス（ECMP）	Yes	PPP または HDLC カプセルを使用した T3 または T1 スピード チャネルを介した出力パスに対する ECMP サポート 複数のコントローラ、SPA、および SIP 上のパスに対する ECMP サポート	Yes	Yes	Yes	Yes
ファシリティ データ リンク（FDL）	Yes	Yes	Yes	Yes	Yes	No
Far End Alarm Control（FEAC）	T3 C ビット フレーム構成用	T3 C ビット フレーム構成用	T3 C ビット フレーム構成用	T3 C ビット フレーム構成用		T3 C ビット フレーム構成用
シャーシ間ステートフル スイッチオーバー（ICSSO）[2]	PPP の場合は T3、T1 および E1 チャネルのみ（DS0 は対象外） MLPPP の場合は T1 および E1 セッションが対象	PPP の場合は T3 チャネルが対象 T1 の場合は T3 チャネルが同じシステム、SIP、SPA またはポート上で設定されているとき	No	T3、T1 および E1 チャネルのみ（DS0 は対象外）	T1 および E1 チャネルのみ（DS0 なし）	No
IP の高速再ルーティング（IP-FRR）	No	PPP 用のみ	No	T3、T1、および E1 チャネル	T1 および E1 チャネル	No
リンク ノイズ モニタ	No	Yes	No	No	No	No
ループバック[3]	Yes	Yes	Yes	Yes：DS0 以外	Yes：DS0 以外	Yes

表 10　チャネライズド T3/E3、T1/E1、およびクリア チャネル SPA でサポートされる機能 （続き）

	1 ポート チャネライズド OC-3/STM-1 SPA	2 ポート チャネライズド OC-12c/DS0 SPA	1 ポート チャネライズド OC-48/STM-16 SPA	4 ポート チャネライズド T3/DS0 SPA	8 ポート チャネライズド T1/E1 SPA	2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA
メンテナンス データ リンク（MDL）メッセージ サポート	Yes	Yes	Yes	Yes	該当なし	Yes
Circuit Emulation Service Over Packet Switched Network のサポート	Yes	Yes	No	No	No	No
混合チャネル サポート	No：T3 と E3 の混合はできません。 T1 と E1 は単一 STS-1 上で共存できません。	Yes：T3 と T1 が同じ SIP、SPA、またはポート上でサポートされます	Yes	Yes	No：すべてのチャネルは、T1 または E1 モードである必要があります。	No：すべてのポートが T3 であるか、すべて E3 であることが必要です。
拡張性	SPA あたり 1000 チャネル	SIP あたり T3 チャネル 48 個 SPA あたり T3 チャネル 24 個 インターフェイスあたり T3 チャネル 12 個	T3/E3 チャネル 48 個	SPA あたり 1000 チャネル	T1 または E1 ポート 8 個 全二重 HDLC チャネル最大 256 個 チャネル スピード *N*x64K または *N*x56K。*N* は、T1 の場合は 24 以下、E1 の場合は 32 以下	2 ～ 4 の T3 または E3 ポート

1. 最初の 3 つの物理ポート間での 6 つの同時 BERT セッションおよび第 4 ポートでの 6 つの同時 BERT セッション。
2. 1 ポート チャネライズド OC-3/STM-1 SPA の SONET/SDH コントローラに設定されたすべてのインターフェイスが IC-SSO で保護されているか、またはすべてが IC-SSO で保護されていない必要があります。
3. ループバック サポートの詳細については、「ループバック サポート」（P.443）を参照してください。

ループバック サポート

Cisco 1 ポート チャネライズド OC-3/STM-1 SPA

このセクションでは、1 ポート チャネライズド OC-3/STM-1 SPA でサポートされるループバックのタイプについて説明します。

- SONET コントローラの場合：
  - ローカル ループバック
  - ネットワーク ライン ループバック
- T3 の場合：
  - ローカル ループバック
  - ネットワーク ループバック
  - リモート ループバック ライン（FEAC を C ビット モードで T3 に使用）
  - リモート ループバック ペイロード（FEAC を C ビット モードで T3 に使用）
- E3 の場合：
  - ローカル ループバック
  - ネットワーク ループバック
- T1 の場合：
  - ローカル ループバック
  - ネットワーク ライン ループバック
  - リモート ライン FDL ANSI ループバック（別名リモート CSU ループバック - ESF モード）
  - リモート ライン FDL ベルコア ループバック（別名リモート SmartJack ループバック - ESF モード）
  - リモート ライン インバンド ループバック（SF インバンド ループバック）
  - リモート ペイロード FDL ANSI ループバック（ESF リモート ペイロード ループバック）
- E1 の場合：
  - ローカル ループバック
  - ネットワーク ライン ループバック

Cisco 2 ポート チャネライズド OC-12c/DS0 SPA

このセクションでは、2 ポート チャネライズド OC-12c/DS0 SPA でサポートされるループバックのタイプについて説明します。

- T3：
  - ローカル ループバック
  - ネットワーク ライン ループバック
- ポートの場合
  - ローカル ライン ループバック
  - ネットワーク ライン ループバック

Cisco 1 ポート チャネライズド OC-48/STM-16 SPA

このセクションでは、1 ポート チャネライズド OC-48/STM-16 SPA でサポートされるループバックのタイプについて説明します。

- SONET の場合：
  - ローカル ライン ループバック
  - ネットワーク ライン ループバック
- T3 の場合：
  - ローカル ループバック
  - ネットワーク ライン ループバック
  - ネットワーク ペイロード ループバック
- E3 の場合：
  - ローカル ループバック
  - ネットワーク ループバック

Cisco 4 ポート チャネライズド T3/DS0 SPA

このセクションでは、4 ポート チャネライズド T3/DS0 SPA でサポートされるループバックのタイプについて説明します。

- T3 の場合：
  - ローカル ループバック
  - ネットワーク ループバック
  - リモート ループバック ライン
- T1 の場合：
  - ローカル ループバック
  - ネットワーク ライン ループバック
  - リモート ライン FDL ANSI ループバック（別名リモート CSU ループバック - ESF モード）
  - リモート ライン FDL ベルコア ループバック（別名リモート SmartJack ループバック - ESF モード）
- E1 の場合：
  - ローカル ループバック
  - ネットワーク ライン ループバック

Cisco 8 ポート チャネライズド T1/E1 SPA

このセクションでは、8 ポート チャネライズド T1/E1 SPA でサポートされるループバック タイプについて説明します。

- T1 の場合：
  - ローカル ループバック
  - ネットワーク ライン ループバック
  - リモート ライン FDL ANSI ループバック（別名リモート CSU ループバック - ESF モード）

  - リモート ライン FDL ベルコア ループバック（別名リモート SmartJack ループバック - ESF モード）
- E1 の場合：
  - ローカル ループバック

Cisco 2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA

ここでは、2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA でサポートされるループバックのタイプについて説明します。

- ローカル ループバック
- ネットワーク ペイロード ループバック（リモート側から受信したすべてのデータをリモート側に返すようにローカル フレーマを設定します）。
- ネットワーク ライン ループバック（リモート側から受信したすべてのデータをリモート側に返すようにローカル LIU を設定します）。
- リモート回線ループバック（FEAC を使用して、SPA にループバックするようにリモート インターフェイスに要求：T3 のみ）

コンフィギュレーションの概要

チャネル化された T3 コントローラおよびその関連付けられたシリアル インターフェイスと設定は、4 段階の手順で行います。

ステップ 1 T3 コントローラを設定し、コントローラのモードを T1 または E1 に設定します。

ステップ 2 T1 または E1 コントローラを設定します。

ステップ 3 チャネル グループを作成し、目的に合わせて DS0 タイムスロットをこれらのチャネル グループに割り当てます。

ステップ 4 このマニュアルで後述する「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」モジュールの説明に従って、各チャネル グループに関連付けられたシリアル インターフェイスを設定します。

T3 および E3 コントローラのデフォルト設定値

表 11 に、T3 および E3 コントローラのデフォルト設定パラメータを示します。

（注）
- 2 ポート チャネライズド OC-12c/DS0 SPA では、自動検出フレーミングはサポートされません。
- E3 は、4 ポート チャネライズド T3/DS0 SPA ではサポートされません。

表 11　T3 および E3 コントローラのデフォルト設定値

パラメータ	デフォルト値	設定ファイルのエントリ
データ ラインのフレーム タイプ	T3 の場合：C ビット フレーム構成 E3 の場合：G.751	**framing** {**auto-detect** | **c-bit** | **m23**}
各 T3/E3 リンクのクロッキング	**internal**	**clock source** {**internal** | **line**}
ケーブル長	224 フィート	**cablelength** *feet*
メンテナンス データ リンク（MDL）メッセージ （T3 のみ）	**disable**	**mdl transmit** {**idle-signal** | **path** | **test-signal**} {**disable** | **enable**}
E3 ポートの各国用予約ビット （E3 のみ）	**enable**、ビット パターン値は 1	**national bits** {**disable** | **enable**}

（注）　シリアル リンクでクロッキングを設定する場合、一方のエンドを **internal** にし、もう一方を **line** にする必要があります。接続の両エンドに **internal** クロッキングを設定すると、フレーム同期のずれが生じます。接続の両エンドに **line** クロッキングを設定すると、ラインはアップ状態になりません。

T1 および E1 コントローラのデフォルト設定値

表 12 に、T1 および E1 コントローラのデフォルト設定パラメータを示します。

表 12　T1 および E1 コントローラのデフォルト設定値

パラメータ	デフォルト値	設定ファイルのエントリ
データ ラインのフレーム タイプ	T1 の場合：拡張スーパーフレーム（**esf**）E1 の場合：CRC-4 エラー モニタリング機能付きフレーミング（**crc4**）。	T1 の場合：**framing** {**sf** | **esf**}E1 の場合：**framing** {**crc4** | **no-crc4** | **unframed**
検出および T1 イエロー アラームの生成 （T1 のみ）	T1 チャネルでイエロー アラームが検出され、生成されます。	**yellow** {**detection** | **generation**} {**disable** | **enable**}
各 T1 および E1 リンクのクロッキング	**internal**	**clock source** {**internal** | **line**}
ケーブル長 （T1 のみ）	**cablelength long** コマンドの場合：*db-gain-value*: gain26; *db-loss-value*: 0db **cablelength short** コマンドの場合：533 feet	ケーブル長を 655 フィートよりも長く設定する場合：**cablelength long** *db-gain-value db-loss-value* ケーブル長を 655 フィート以下に設定する場合：**cablelength short** *length*

表 12　T1 および E1 コントローラのデフォルト設定値 （続き）

パラメータ	デフォルト値	設定ファイルのエントリ
ANSI T1.403 または AT&T TR54016 についての秒単位のパフォーマンス レポートの、T1 チャネルのファシリティ データ リンク（FDL）を通じた伝送 （T1 のみ）	**disable**	**fdl** {**ansi** | **att**} {**enable** | **disable**}
E1 ポートの各国用予約ビット （E1 のみ）	0（16 進表記の *0x1f* に一致します）	**national bits** *bits*

（注）　シリアル リンクでクロッキングを設定する場合、一方のエンドを **internal** にし、もう一方を **line** にする必要があります。接続の両エンドに **internal** クロッキングを設定すると、フレーム同期のずれが生じます。接続の両エンドに **line** クロッキングを設定すると、ラインはアップ状態になりません。

T1 または E1 リンクでのリンク ノイズ モニタ

リンク ノイズ モニタ（LNM）とは、Cisco ASR 9000 シリーズ ルータの 2 ポート チャネライズド OC-12c/DS0 SPA 上の T1 および E1 リンクにおけるパス コード違反（PCV）エラーをモニタリングする機能です。この目的は、これらのリンクにおけるノイズが、設定済みのしきい値（**set** しきい値）に達するか超える状態が継続したときに、イベントとアラームでこのエラーを通知することです。また、ノイズが設定された改善しきい値（**clear** しきい値）以下に下がった場合も通知されます。

Cisco IOS XR リリース 4.1 から、PPP にノイズ属性を通知して、指定したしきい値を超過した場合に MLPPP バンドル メンバ リンクを削除できるように、LNM 機能で **lnm remove** コマンドをサポートしています。

（注）　LCV は、極性違反（BPV）または過剰ゼロ（EXZ）エラーの発生であり、PCV はタイムスロットの CRC エラーの発生です。ただし、LNM 機能でモニタリングされるのは現時点では PCV エラーだけです。PCV 値が指定されない場合は、LCV 値は予期される PCV の計算だけに使用されます。PCV 値が指定されている場合は、LCV 値は無視されます。

LNM イベント

LNM によって生成されるイベントには、2 つの基本的なタイプがあります。

- 超過イベント：*超過*イベントが生成されるのは、PCV しきい値（**set** でメジャーおよびマイナーの警告に対して指定された値）に達するか超えた状態が、指定された時間（**duration**）続いたときです。超過イベントが発生すると、コントローラのメジャーまたはマイナー モニタリング タイプが *alarm* 状態としてレポートされます。超過イベントが存在しなくなったときは、モニタリング タイプが *stable* 状態に戻ります。

  次に、超過イベントの例を示します。

```
RP/0/RSP0/CPU0:Router#0/1/CPU0:May 13 9:54:10.980 : g_spa_1[181]:
%L2-T1E1_LNM-3-MINWARNNOISE :
Interface T10/1/1/0/1/1/1, noise crossed minor warning threshold

RP/0/RSP0/CPU0:Router#0/1/CPU0:May 13 9:54:11.980 : g_spa_1[181]:
%L2-T1E1_LNM-3-MAJWARNNOISE :
```

```
Interface T10/1/1/0/1/1/1, noise crossed major warning threshold
```

- クリア イベント：超過したしきい値が、メジャーおよびマイナー警告の指定した **clear** 値未満に低下した場合に送信される *cleared* イベント信号。

  次に、クリア イベントの例を示します。

```
RP/0/RSP0/CPU0:Router#LC/0/1/CPU0:May 13 10:27:25.809 : g_spa_1[181]:
%L2-T1E1_LNM-3-MAJWARNNOISE :
Interface T10/1/1/0/1/1/1, noise cleared major warning threshold

RP/0/RSP0/CPU0:Router#LC/0/1/CPU0:May 13 10:28:14.810 : g_spa_1[181]:
%L2-T1E1_LNM-3-MINWARNNOISE :
Interface T10/1/1/0/1/1/1, noise cleared minor warning threshold
```

LNM ロギング

lnm syslog コマンドを使用して LNM イベントの syslog メッセージをイネーブルにすると、システム ログおよびログ イベント バッファの両方に LNM メッセージが表示されます。ログ イベント バッファにある LNM イベントを表示するには、**show logging events buffer bistate-alarms-set** コマンドを使用し、**show logging** コマンドも使用します。これらの説明は、『*Cisco ASR 9000 Series Aggregation Services Router System Monitoring Command Reference*』にあります。

LNM は、Telcordia（ベルコア）GR-253 標準で定義されている階層レベル警告レポートをサポートします。階層警告レポートとは、上位のアラームがアサートされたときに、それよりも下位のアラーム状態が抑制されることを意味します。上位のアラームがクリアされたときに、それよりも下位のアラームの状態がまだ続いているときは、そのアラームが再度アサートされます。

LNM では、これは継続的にメジャー警告しきい値以上になり超過イベントおよびアラーム状態が発生した場合、マイナー警告アラーム状態は抑制され、安定状態に戻ることを意味します。マイナー超過イベントは、バイステート ログからも削除されます。メイン警告がクリアされると、条件がまだ存在していればマイナー警告アラームが再度アサートされます。

コントローラのバイステート ログには、メジャー警告に対する超過イベントが 1 つだけ表示されます。したがって、設定済みのしきい値を超過するノイズが存在する場合は、1 つのコントローラに対してログ メッセージが 1 つだけ表示されます。

クリア チャネル T3/E3 コントローラおよびチャネライズド T1/E1 コントローラの設定方法

T3/E3 コントローラは、Cisco IOS XR ソフトウェア のコンフィギュレーション スペースの物理層のコントロール要素で設定します。このコンフィギュレーションについては、次のタスクで説明します。

- 「クリア チャネル E3 コントローラの設定」（P.449）
- 「デフォルトの E3 コントローラ設定の変更」（P.450）
- 「クリア チャネル T3 コントローラの設定」（P.453）
- 「チャネル化された T3 コントローラの設定」（P.455）
- 「デフォルトの T3 コントローラ設定の変更」（P.457）
- 「T1 コントローラの設定」（P.459）
- 「E1 コントローラの設定」（P.463）
- 「BERT の設定」（P.467）
- 「T1 または E1 チャネルでのリンク ノイズ モニタの設定」（P.474）

クリア チャネル E3 コントローラの設定

クリア チャネル モードにある E3 コントローラは、単一シリアル インターフェイスを伝送します。

E3 コントローラを設定するには、E3 コンフィギュレーション モードを使用します。

制約事項

- コントローラ タイプに有効でないオプションを設定すると、設定をコミットするときにエラーが表示されます。
- 単一の SPA では、T3 インターフェイスと E3 インターフェイスの併用はサポートされません。
- E3 は、4 ポート チャネライズド T3/DS0 SPA ではサポートされません。

手順の概要

1. **configure**
2. **controller e3** *interface-path-id*
3. **mode serial**
4. **no shutdown**
5. **end**
   または
   commit
6. **show controllers e3** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller e3** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t3 0/1/0/0`	*rack/slot/module/port* 表記で E3 コントローラ名を指定し、E3 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	**mode serial** **例：** `RP/0/RSP0/CPU0:router(config-e3)# mode serial`	ポートのモードをクリア チャネル シリアルに設定します。 **（注）** このステップは、2 ポートおよび 4 ポート チャネライズド T3 SPA にのみ必要です。2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA は、デフォルトでシリアル モードで実行されます。
ステップ 4	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-e3)# no shutdown`	shutdown 設定を削除します。 • shutdown 設定を削除すると、コントロールに強制された管理上のダウンが解除され、コントローラを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。

	コマンドまたはアクション	目的
ステップ 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-e3)# end` または `RP/0/RSP0/CPU0:router(config-e3)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 6	**show controllers e3** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show controllers e3 0/1/0/0`	（任意）E3 コントローラに関する情報を表示します。

次の作業

- 設定した E3 コントローラ上で実行されているデフォルト設定を、このマニュアルで後述する「デフォルトの E3 コントローラ設定の変更」の説明に従って変更します。
- このモジュールで後述する「BERT の設定」の説明に従って、その完全性をテストするため、コントローラのビット誤り率テスト（BERT）を設定します。
- このマニュアルで後述する *「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」* モジュールの説明に従って、関連付けられたシリアル インターフェイスを設定します。

デフォルトの E3 コントローラ設定の変更

ここでは、このモジュールで前述した「T3 および E3 コントローラのデフォルト設定値」で説明したデフォルトの E3 コントローラ設定を変更する手順について説明します。

前提条件

このモジュールで前述した「クリア チャネル E3 コントローラの設定」の説明に従って、クリア チャネル E3 コントローラを設定する必要があります。

制約事項

- E3 は、4 ポート チャネライズド T3/DS0 SPA ではサポートされません。

手順の概要

1. **configure**
2. **controller e3** *interface-path-id*
3. **clock source** {**internal** | **line**}
4. **cablelength** *feet*
5. **framing** {**g751** | **g832**}
6. **national bits** {**disable** | **enable**}
7. **no shutdown**
8. **end**
   または
   commit
9. **show controllers e3** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller e3** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t3 0/1/0/0`	*rack/slot/module/port* 表記で E3 コントローラ名を指定し、E3 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	**clock source** {**internal** | **line**} **例：** `RP/0/RSP0/CPU0:router(config-e3)# clock source internal`	（任意）個々の E3 リンクのクロッキングを設定します。 **（注）** デフォルトのクロック ソースは **internal** です。 **（注）** シリアル リンクでクロッキングを設定する場合、一方のエンドを **internal** にし、もう一方を **line** にする必要があります。接続の両エンドに **internal** クロッキングを設定すると、フレーム同期のずれが生じます。接続の両エンドに **line** クロッキングを設定すると、ラインはアップ状態になりません。
ステップ 4	**cablelength** *feet* **例：** `RP/0/RSP0/CPU0:router(config-e3)# cablelength 250`	（任意）ルータからネットワーク装置までのケーブルの長さを指定します。 **（注）** デフォルトのケーブル長は 224 フィートです。

	コマンドまたはアクション	目的
ステップ 5	**framing** {**g751** | **g832**} **例：** `RP/0/RSP0/CPU0:router(config-e3)# framing g832`	（任意）E3 ポートのフレーム タイプを指定します。設定可能な E3 フレーム タイプは、G.751 および G.832 です。 **（注）** E3 のデフォルトのフレーム構成は G.751 です。
ステップ 6	**national bits** {**disable** | **enable**} **例：** `RP/0/RSP0/CPU0:router(config-e3)# national bits enable`	（任意）E3 ポートの 0x1F 各国用予約ビットパターンをイネーブルまたはディセーブルにします。 **（注）** E3 各国用ビットはデフォルトでイネーブルに設定され、ビットパターン値は 1 です。
ステップ 7	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-e3)# no shutdown`	shutdown 設定を削除します。 • shutdown 設定を削除すると、コントロールに強制された管理上のダウンが解除され、コントローラを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 8	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-e3)# end` または `RP/0/RSP0/CPU0:router(config-e3)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 9	**show controllers e3** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show controllers e3 0/1/0/0`	（任意）E3 コントローラに関する情報を表示します。

次の作業

- 設定した T3 コントローラ上で実行されているデフォルト設定を、このモジュールで後述する「デフォルトの T3 コントローラ設定の変更」の説明に従って変更します。
- このモジュールで後述する「BERT の設定」セクションの説明に従って、その完全性をテストするため、コントローラに BERT を設定します。

- このマニュアルで後述する*「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」*モジュールの説明に従って、関連付けられたシリアル インターフェイスを設定します。

クリア チャネル T3 コントローラの設定

クリア チャネル モードにある T3 コントローラは、単一シリアル インターフェイスを伝送します。

T3 コントローラを設定するには、T3 コンフィギュレーション モードを使用します。

前提条件

チャネライズド SPA 上でクリア チャネル T3 コントローラを設定するには、その前に、STS ストリームを T3 チャネル化するようにその SPA を設定する必要があります。詳細については、「Cisco ASR 9000 シリーズ ルータ でのチャネライズド SONET/SDH の設定」モジュールを参照してください。

制約事項

- コントローラ タイプに有効でないオプションを設定すると、設定をコミットするときにエラーが表示されます。
- 単一の SPA では、T3 インターフェイスと E3 インターフェイスの併用はサポートされません。

手順の概要

1. **configure**
2. **controller t3** *interface-path-id*
3. **mode serial**
4. **no shutdown**
5. **end**
   または
   commit
6. **show controllers t3** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller t3** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t3 0/1/0/0`	*rack/slot/module/port* 表記で T3 コントローラ名を指定し、T3 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	**mode serial** **例：** `RP/0/RSP0/CPU0:router(config-t3)# mode serial`	**（注）** ポートのモードをクリア チャネル シリアルに設定します。
ステップ 4	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-t3)# no shutdown`	shutdown 設定を削除します。 • shutdown 設定を削除すると、コントロールに強制された管理上のダウンが解除され、コントローラを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 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-t3)# end` または `RP/0/RSP0/CPU0:router(config-t3)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 6	**show controllers t3** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show controllers t3 0/1/0/0`	（任意）T3 コントローラに関する情報を表示します。

次の作業

- 設定した T3 コントローラ上で実行されているデフォルト設定を、このモジュールで後述する「デフォルトの T3 コントローラ設定の変更」の説明に従って変更します。
- このモジュールで後述する「BERT の設定」セクションの説明に従って、その完全性をテストするため、コントローラに BERT を設定します。
- *「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」*モジュールの説明に従って、関連するシリアル インターフェイスを設定します。

チャネル化された T3 コントローラの設定

チャネライズド T3 をサポートする SPA でサポートされるチャネル化は、T1、E1、および DS0 へのチャネル化です。ここでは、単一の T3 コントローラを 28 T1 コントローラまたは 21 E1 コントローラにチャネル化する手順について説明します。T1 または E1 コントローラを作成すると、次の説明に従って、それらのコントローラを DS0 タイムスロットにチャネル化することができます。

- T1 コントローラの設定
- E1 コントローラの設定

個々の T1 コントローラは、24 DS0 タイムスロットの合計をサポートします。また、個々の E1 コントローラは、31 DS0 タイムスロットの合計をサポートします。

前提条件

チャネライズド T3 コントローラを設定する前に、次の要件が満たされていることを確認します。

- 次のいずれかの SPA がインストールされていること。
  - 1 ポート チャネライズド OC-3/STM-1 SPA
  - 2 ポート チャネライズド OC-12/DS0 SPA
  - 4 ポート チャネライズド T3/DS0 SPA
- チャネライズド SONET SPA の場合は、STS ストリームを T3 用にチャネル化するように SPA を設定済みであること。詳細については、「Cisco ASR 9000 シリーズ ルータ でのチャネライズド SONET/SDH の設定」モジュールを参照してください。

（注） コントローラ タイプに有効でないオプションを設定すると、設定をコミットするときにエラーが表示されます。

手順の概要

1. **configure**
2. **controller t3** *interface-path-id*
3. **mode** [**t1** | **e1**]
4. **no shutdown**
5. **end**
   または
   commit
6. **show controllers t3** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller T3** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t3 0/1/0/0`	*rack/slot/module/port* 表記で T3 コントローラ名を指定し、T3 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	**mode t1** **例：** `RP/0/RSP0/CPU0:router(config-t3)# mode t1`	チャネル化したコントローラのモードを T1 に設定し、28 T1 コントローラを作成します。
ステップ 4	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-t3)# no shutdown`	shutdown 設定を削除します。 • shutdown 設定を削除すると、コントロールに強制された管理上のダウンが解除され、コントローラを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 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-t3)# end` または `RP/0/RSP0/CPU0:router(config-t3)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 6	**show controllers t3** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show controllers t3 0/1/0/0`	（任意）T3 コントローラに関する情報を表示します。

次の作業

- 設定した T3 コントローラ上で実行されているデフォルト設定を変更します。手順については、「デフォルトの T3 コントローラ設定の変更」（P.457）を参照してください。
- T3 コントローラを 28 個の T1 コントローラにチャネル化した場合は、これらの T1 コントローラを設定し、DS0 タイム スロットを割り当てます。手順については、*「T1 コントローラの設定」（P.459）*を参照してください。
- T3 コントローラを 21 個の E1 コントローラにチャネル化した場合は、E1 コントローラを設定し、DS0 タイム スロットを割り当てます。手順については、*「E1 コントローラの設定」（P.463）*を参照してください。

デフォルトの T3 コントローラ設定の変更

ここでは、「T3 および E3 コントローラのデフォルト設定値」（P.445）で説明したデフォルトの T3 コントローラ設定を変更する手順について説明します。

前提条件

次のいずれかの項の説明に従って、クリア チャネルまたはチャネライズド T3 コントローラを設定する必要があります。

- クリア チャネル T3 コントローラの設定
- チャネル化された T3 コントローラの設定

手順の概要

1. **configure**
2. **controller t3** *interface-path-id*
3. **clock source** {**internal** | **line**}
4. **cablelength** *feet*
5. **framing** {**auto-detect** | **c-bit** | **m23**}
6. **mdl transmit** {**idle-signal** | **path** | **test-signal**} {**disable** | **enable**}
7. **mdl string** {**eic** | **fi** | **fic** | **gen-number** | **lic** | **port-number** | **unit**} *string*
8. **no shutdown**
9. **end**
   または
   commit
10. **show controllers t3** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller T3** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t3 0/1/0/0`	*rack/slot/module/port* 表記で T3 コントローラ名を指定し、T3 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	**clock source** {**internal** | **line**} **例：** `RP/0/RSP0/CPU0:router(config-t3)# clock source internal`	（任意）T3 ポートのクロッキングを設定します。 **（注）** デフォルトのクロック ソースは **internal** です。 **（注）** シリアル リンクでクロッキングを設定する場合、一方のエンドを **internal** にし、もう一方を **line** にする必要があります。接続の両エンドに **internal** クロッキングを設定すると、フレーム同期のずれが生じます。接続の両エンドに **line** クロッキングを設定すると、ラインはアップ状態になりません。
ステップ 4	**cablelength** *feet* **例：** `RP/0/RSP0/CPU0:router(config-t3)# cablelength 250`	（任意）ルータからネットワーク装置までのケーブルの長さを指定します。 **（注）** デフォルトのケーブル長は 224 フィートです。
ステップ 5	**framing** {**auto-detect** | **c-bit** | **m23**} **例：** `RP/0/RSP0/CPU0:router(config-t3)# framing c-bit`	（任意）T3 ポートのフレーム タイプを指定します。 **（注）** T3 のデフォルトのフレーム タイプは C-bit です。2 ポート チャネライズド OC-12c/DS0 SPA では、自動検出はサポートされません。
ステップ 6	**mdl transmit** {**idle-signal** | **path** | **test-signal**} {**disable** | **enable**} **例：** `RP/0/RSP0/CPU0:router(config-t3)# mdl transmit path enable`	（任意）T3 ポートのメンテナンス データ リンク（MDL）メッセージをイネーブルにします。 **（注）** MDL メッセージは、T3 フレーム構成が C-bit パリティである場合にのみサポートされます。 **（注）** MDL メッセージはデフォルトで表示されます。
ステップ 7	**mdl string** {**eic** | **fi** | **fic** | **gen-number** | **lic** | **port-number** | **unit**} *string* **例：** `RP/0/RSP0/CPU0:router(config-t3)# mdl fi facility identification code`	（任意）MDL メッセージで送信される文字列の値を指定します。
ステップ 8	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-t3)# no shutdown`	shutdown 設定を削除します。 • shutdown 設定を削除すると、コントロールに強制された管理上のダウンが解除され、コントローラを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。

	コマンドまたはアクション	目的
ステップ 9	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-t3)# end` または `RP/0/RSP0/CPU0:router(config-t3)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 10	**show controllers t3** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show controllers t3 0/1/0/0`	（任意）T3 コントローラに関する情報を表示します。

次の作業

- クリア チャネル T3 コントローラを設定したら、次の作業を行います。
  - このモジュールで後述する「BERT の設定」（P.467）の説明に従って、その完全性をテストするため、コントローラに BERT を設定します。
  - *「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」*モジュールの説明に従って、関連するシリアル インターフェイスを設定します。
- T3 コントローラを 28 個の T1 コントローラにチャネル化した場合は、これらの T1 コントローラを設定し、DS0 タイム スロットを割り当てます。手順については、*「T1 コントローラの設定」（P.459）*を参照してください。
- T3 コントローラを 21 個の E1 コントローラにチャネル化した場合は、E1 コントローラを設定し、DS0 タイム スロットを割り当てます。手順については、*「E1 コントローラの設定」（P.463）*を参照してください。

T1 コントローラの設定

ここでは、個々の T1 コントローラを設定し、それを 24 の個別の DS0 タイムスロットにチャネル化する手順について説明します。

前提条件

T1 コントローラを設定する前に、次の要件が満たされていることを確認します。

- 次のいずれかの SPA がインストールされていること。
  - 1 ポート チャネライズド OC-3/STM-1 SPA
  - 2 ポート チャネライズド OC-12/DS0 SPA
  - 4 ポート チャネライズド T3/DS0 SPA
  - 8 ポート チャネライズド T1/E1 SPA
- 1 ポート チャネライズド OC-3/STM-1 SPA または 2 ポート チャネライズド OC-12/DS0 SPA がある場合は、次の設定を行う必要があります。
  - T3 にチャネル化した STS ストリームを設定します。詳細については、「Cisco ASR 9000 シリーズ ルータ でのチャネライズド SONET/SDH の設定」モジュールを参照してください。
  - 「チャネル化された T3 コントローラの設定」（P.455）の説明に従って T1 モードで動作するチャネライズド T3 コントローラを設定します。
- 4 ポート チャネライズド T3/DS0 SPA がある場合は、チャネライズド T3 コントローラを T1 モードで動作するように設定する必要があります。手順については、「チャネル化された T3 コントローラの設定」（P.455）を参照してください。

制約事項

コントローラ タイプに有効でないオプションを設定すると、設定をコミットするときにエラーが表示されます。

8 ポート チャネライズド T1/E1 SPA で T1 コントローラを設定する前に、次の制限事項を考慮してください。

- SPA コントローラは T1 モード用に明示的に設定されるまでは表示されません。
- 個々の SPA について、すべての SPA ポートが同じモード（すべて T1）である必要があります。

手順の概要

1. **show controllers t1** *interface-path-id*
2. **configure**
3. **controller t1** *interface-path-id*
4. **framing** {**sf** | **esf**}
5. **yellow** {**detection** | **generation**} {**disable** | **enable**}
6. **clock source** {**internal** | **line**}
7. **fdl** {**ansi** | **att**} {**enable** | **disable**}
8. **no shutdown**
9. **channel-group** *channel-group-number*
10. **timeslots** *range*
11. **speed** *kbps*
12. **exit**

13. ステップ 9 ～ 12 を繰り返し、タイムスロットをチャネル グループに割り当てます。各コントローラには、最大 24 のタイムスロットを設定できます。
14. **exit**
15. ステップ 2 ～ 14 を繰り返し、さらなるチャネル グループをコントローラに割り当てます。
16. **end**
    または
    commit

手順の詳細

ステップ 1	**show controllers t1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show controllers t3 0/1/0/0`	（任意）ステップ 3 で作成した T1 コントローラに関する情報を表示します。
ステップ 2	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**controller t1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t1 0/3/0/0/0`	T1 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	**framing** {**sf** | **esf**} **例：** `RP/0/RSP0/CPU0:router(config-t1)# framing esf`	（任意）T1 データ ラインのフレーム タイプを指定します。 • **sf**：スーパーフレーム • **esf**：拡張スーパーフレーム **（注）** T1 のデフォルトのフレーム タイプは拡張スーパーフレーム（**esf**）です。
ステップ 5	**yellow** {**detection** | **generation**} {**disable** | **enable**} **例：** `RP/0/RSP0/CPU0:router(config-t1e1)# yellow detection enable`	（任意）T1 でのイエロー アラームの検出と生成をイネーブルまたはディセーブルにします。 **（注）** デフォルトでは、T1 チャネルでイエロー アラームが検出され、生成されます。
ステップ 6	**clock source** {**internal** | **line**} **例：** `RP/0/RSP0/CPU0:router(config-t1e1)# clock source internal`	（任意）個々の T1 リンクのクロッキングを設定します。 **（注）** デフォルトのクロック ソースは **internal** です。 **（注）** シリアル リンクでクロッキングを設定する場合、一方のエンドを **internal** にし、もう一方を **line** にする必要があります。接続の両エンドに **internal** クロッキングを設定すると、フレーム同期のずれが生じます。接続の両エンドに **line** クロッキングを設定すると、ラインはアップ状態になりません。

ステップ 7	**fdl** {**ansi** | **att**} {**enable** | **disable**} **例：** `RP/0/RSP0/CPU0:router(config-t1e1)# fdl ansi enable`	（任意）ファシリティ データ リンク（FDL）を介した ANSI T1.403 または AT&T TR54016 についての秒単位のパフォーマンス レポートの伝送をイネーブルにします。 **（注）** FDL ansi および att はデフォルトでディセーブルに設定されています。
ステップ 8	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-t1e1)# no shutdown`	shutdown 設定を削除します。 • shutdown 設定を削除すると、コントロールに強制された管理上のダウンが解除され、コントローラを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 9	**channel-group** *channel-group-number* **例：** `RP/0/RSP0/CPU0:router(config-t1)# channel-group 0`	T1 チャネル グループを作成し、そのチャネル グループのチャネル 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 10	**timeslots** *range* **例：** `RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 7-12`	DS0 タイムスロットをチャネル グループに関連付けて、対応するシリアル サブインターフェイスをそのチャネル グループに対して作成します。 • 範囲は 1 ～ 24 タイムスロットです。 • 24 タイムスロットすべてを単一のチャネル グループに割り当てることも、タイムスロットを複数のチャネル グループに分割することもできます。 **（注）** 個々の T1 コントローラは、24 DS0 タイムスロットの合計をサポートします。
ステップ 11	**speed** *kbps* **例：** `RP/0/RSP0/CPU0:router(config-t1e1-channel_group)# speed 64`	（任意）DS0 の速度を Kbps 単位で指定します。有効値は 56 と 64 です。 **（注）** デフォルトの速度は 64 kbps です。
ステップ 12	**exit** **例：** `RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit`	チャネル グループ コンフィギュレーション モードを終了します。
ステップ 13	ステップ 9 ～ 12 を繰り返し、タイムスロットをチャネル グループに割り当てます。各コントローラには、最大 24 のタイムスロットを設定できます。	—
ステップ 14	**exit** **例：** `RP/0/RSP0/CPU0:router(config-t1)# exit`	T1 コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードを開始します。

ステップ 15	必要に応じて、ステップ 2 ～ 14 を繰り返してその他のチャネル グループをコントローラに割り当てます。	—
ステップ 16	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-t3)# end` または `RP/0/RSP0/CPU0:router(config-t3)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

次の作業

- 「BERT の設定」（P.467）の説明に従って、その完全性をテストするため、コントローラに BERT を設定します。
- 対応するシリアル インターフェイスを設定します。手順については、*「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」*モジュールを参照してください。

E1 コントローラの設定

ここでは、個々の E1 コントローラを設定し、それを 31 の個別の DS0 タイムスロットにチャネル化する手順について説明します。

前提条件

E1 コントローラを設定する前に、次の要件が満たされていることを確認します。

- 次のいずれかの SPA がインストールされていること。
  - 1 ポート チャネライズド OC-3/STM-1 SPA
  - 2 ポート チャネライズド OC-12/DS0 SPA
  - 4 ポート チャネライズド T3/DS0 SPA
  - 8 ポート チャネライズド T1/E1 SPA
- 1 ポート チャネライズド OC-3/STM-1 SPA または 2 ポート チャネライズド OC-12/DS0 SPA がある場合は、次の設定を行う必要があります。

  - T3 にチャネル化した STS ストリームを設定します。詳細については、「Cisco ASR 9000 シリーズ ルータ でのチャネライズド SONET/SDH の設定」モジュールを参照してください。
  - 「チャネル化された T3 コントローラの設定」（P.455）の説明に従って E1 モードで動作するチャネライズド T3 コントローラを設定します。
- 4 ポート チャネライズド T3/DS0 SPA がある場合は、チャネライズド T3 コントローラを E1 モードで動作するように設定する必要があります。手順については、「チャネル化された T3 コントローラの設定」（P.455）を参照してください。

制約事項

コントローラ タイプに有効でないオプションを設定すると、設定をコミットするときにエラーが表示されます。

8 ポート チャネライズド T1/E1 SPA で E1 コントローラを設定する前に、次の制限事項を確認してください。

- SPA コントローラは、E1 モード用に明示的に設定されるまでは認識されません。
- 個々の SPA について、すべての SPA ポートが同じモード（すべて E1）である必要があります。

手順の概要

1. **show controllers e1** *interface-path-id*
2. **configure**
3. **controller e1** *interface-path-id*
4. **clock source** {**internal** | **line**}
5. **framing** {**crc4** | **no-crc4** | **unframed**}
6. **national bits** *bits*
7. **no shutdown**
8. **channel-group** *channel-group-number*
9. **timeslots** *range*
10. **speed** *kbps*
11. **exit**
12. ステップ 8 ～ 11 を繰り返し、タイムスロットをチャネル グループに割り当てます。各コントローラには、最大 24 のタイムスロットを設定できます。
13. **exit**
14. ステップ 2 ～ 13 を繰り返し、目的に合わせて、さらなるチャネル グループをコントローラに割り当てます。
15. **end**
    または
    commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**show controllers e1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show controllers e1 0/1/0/0`	（任意）E1 コントローラに関する情報を表示します。
ステップ 2	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**controller e1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller e1 0/3/0/0/0`	E1 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	**clock source** {**internal** | **line**} **例：** `RP/0/RSP0/CPU0:router(config-e1)# clock source internal`	（任意）個々の E1 リンクのクロッキングを設定します。 **（注）** デフォルトのクロック ソースは **internal** です。 **（注）** シリアル リンクでクロッキングを設定する場合、一方のエンドを **internal** にし、もう一方を **line** にする必要があります。接続の両エンドに **internal** クロッキングを設定すると、フレーム同期のずれが生じます。接続の両エンドに **line** クロッキングを設定すると、ラインはアップ状態になりません。
ステップ 5	**framing** {**crc4** | **no-crc4** | **unframed**} **例：** `RP/0/RSP0/CPU0:router(config-e1)# framing unframed`	（任意）E1 データ ラインのフレーム タイプを指定します。E1 に有効なフレーム タイプは次のとおりです。 • **crc4**：CRC-4 エラー監視機能付きのフレーム構成 • **no-crc4**：CRC-4 エラー監視機能なしのフレーム構成 • **unframed**：フレーム化されていない E1 **（注）** E1 のデフォルトのフレーム タイプは **crc4** です。
ステップ 6	**national bits** *bits* **例：** `RP/0/RSP0/CPU0:router(config-e1)# national bits 10`	（任意）E1 ポートの各国用予約ビットを指定します。指定できる範囲は 0 ～ 31 です。 **（注）** デフォルトのビット パターンは 0 です。これは 16 進表記の *0x1f* に一致します。
ステップ 7	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-e1)# no shutdown`	shutdown 設定を削除します。 • shutdown 設定を削除すると、コントロールに強制された管理上のダウンが解除され、コントローラを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。

	コマンドまたはアクション	目的
ステップ 8	**channel-group** *channel-group-number* **例：** `RP/0/RSP0/CPU0:router(config-e1)# channel-group 0`	E1 チャネル グループを作成し、そのチャネル グループのチャネル 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 9	**timeslots** *range* **例：** `RP/0/RSP0/CPU0:router(config-e1-channel_group)# timeslots 1-16`	1 つまたは複数のタイムスロットをチャネル グループに関連付け、関連付けたシリアル サブインターフェイスをそのチャネル グループに作成します。 • 範囲は 1 ～ 31 タイムスロットです。 • 31 タイムスロットすべてを単一のチャネル グループに割り当てることも、タイムスロットを複数のチャネル グループに分割することもできます。 **（注）** 各 E1 コントローラは、31 DS0 タイムスロットの合計をサポートします。
ステップ 10	**speed** *kbps* **例：** `RP/0/RSP0/CPU0:router(config-e1-channel_group)# speed 100`	（任意）DS0 の速度を Kbps 単位で指定します。有効値は 56 と 64 です。 **（注）** デフォルトの速度は 64 kbps です。
ステップ 11	**exit** **例：** `RP/0/RSP0/CPU0:router(config-e1-channel_group)# exit`	チャネル グループ コンフィギュレーション モードを終了します。
ステップ 12	ステップ 8 ～ 11 を繰り返し、タイムスロットをチャネル グループに割り当てます。	—
ステップ 13	**exit** **例：** `RP/0/RSP0/CPU0:router(config-e1)# exit`	E1 コンフィギュレーション モードを終了します。

	コマンドまたはアクション	目的
ステップ 14	ステップ 2 ～ 13 を繰り返し、目的に合わせて、さらなるチャネル グループをコントローラに割り当てます。	—
ステップ 15	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-e3)# end` または `RP/0/RSP0/CPU0:router(config-e3)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

次の作業

- このモジュールの「BERT の設定」（P.467）の説明に従って、その完全性をテストするため、コントローラに BERT を設定します。
- *「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」*モジュール（このマニュアルで後述します）の説明に従って、対応するシリアル インターフェイスを設定します。

BERT の設定

使用するハードウェアのサポートに応じて、BERT は T3/E3 または T1/E1 コントローラのそれぞれにおいて、および DS0 チャネル グループにおいてサポートされます。これは、フレーム化されていない T3/E3 または T1/E1 信号でのみ行われ、一度に 1 つのポート上でのみ実行されます。個々のチャネル グループでもサポートされます。

BERT の結果を参照するには、EXEC モードで **show controllers t1** または **show controllers t3** コマンドを使用します。BERT の結果には次の情報が含まれます。

- 選択したテスト パターンのタイプ
- テストのステータス
- 選択したインターバル
- BER テストの残り時間
- 合計ビット エラー

- 合計受信ビット

BERT はデータ挿入型です。テストの実行中、正規のデータはラインにフローされません。BERT の進行中、ラインはアラーム状態に置かれ、BERT が完了すると正常状態に復元されます。

T3/E3 および T1/E1 コントローラでの BERT の設定

ここでは、T3/E3 ライン、T1/E1 ライン、または個々のチャネル グループでビット エラー レート テスト（BERT）のパターンをイネーブルにする手順について説明します。

前提条件

クリア チャネル T3/E3 コントローラまたはチャネライズド T3-to-T1/E1 コントローラを設定する必要があります。

制約事項

1 ポート チャネライズド OC-48/STM-16 SPA で BERT を設定する前に、次の制限事項を確認してください。

- 同時に設定できる BERT テストは STS-12 ストリームあたり 2 つだけです。
- これらのテスト パターンがサポートされます。
  - 2^15-1（O.151）
  - 2^20-1（O.151）- QRSS
  - 2^23-1（O.151）
  - 固定パターン（すべて 0s、すべて 1s など）
  - 単一ビット エラー注入
  - データ反転

4 ポート チャネライズド T3/DS0 SPA 上で BERT を設定する場合は、次の制約事項に配慮してください。

- 最大 12 個の BERT セッションがサポートされます。
- 最初の 3 つの物理ポート間での 6 つの同時 BERT セッションおよび第 4 ポートでの 6 つの同時 BERT セッションがサポートされます。
- T1 ごとに 1 つの BERT セッションだけがサポートされます。
- これらのテスト パターンが 4 ポート チャネライズド T3/DS0 SPA でサポートされます。
  - 2^11-1：T1/E1/DS0 のみ
  - 2^15-1（O.151）
  - 2^20-1（O.153）：T3 のみ
  - 2^20-1（QRSS）
  - 2^23-1（O.151）
  - 0/1 交互
  - 固定パターン（すべて 0s、すべて 1s など）
  - 1 in 8 DS1 挿入：T1/E1/DS0 のみ
  - 3 in 24 DS1 挿入：T1/E1/DS0 のみ

次のパターンが 8 ポート チャネライズド T1/E1 SPA で T1/E1/DS0 に対してサポートされます。

- 2^11-1
- 2^15-1（O.153）
- 2^20-1（QRSS）
- 2^23-1（O.151）
- 0/1 交互
- 固定パターン（すべて 0s、すべて 1s など）

他のカードの場合、すべてのコントローラとチャネル グループの有効なパターンは次のとおりです。0s、1s、2^15、2^20、2^20-QRSS、2^23 および alt-0-1。

T1 および E1 コントローラに有効なパターンには 1in8、3in24、55Daly、55Octet があります。チャネル グループに有効なパターンには 2^11、2^9、ds0-1、ds0-2、ds0-3、ds0-4 があります。

手順の概要

1. **configure**
2. **controller** [**t3** | **e3** | **t1** | **e1**] *interface-path-id*
3. **pattern** *pattern*
4. **bert interval** *time*
5. **bert error** [*number*]
6. **end**
   または
   commit
7. **exit**
8. **exit**
9. **bert** [**t3** | **e3** | **t1** | **e1**] *interface-path-id* [**channel-group** *channel-group-number*] [**error**] **start**
10. **bert** [**t3** | **e3** | **t1** | **e1**] *interface-path-id* [**channel-group** *channel-group-number*] **stop**
11. **show controllers** [**t3** | **e3** | **t1** | **e1**] *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller** [**t3** | **e3** | **t1** | **e1**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t3 0/1/0/0`	コントローラ名とインスタンスを *rack/slot/module/port* 表記で指定し、T3、E3、T1、または E1 コントローラ 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	**bert pattern** *pattern* **例：** `RP/0/RSP0/CPU0:router(config-t3)# bert pattern 2^15`	コントローラで特定のビット誤り率テスト（BERT）のパターンをイネーブルにします。 **（注）** BER テストを開始するには、EXEC モードで **bert** コマンドを使用する必要があります。
ステップ 4	**bert interval** *time* **例：** `RP/0/RSP0/CPU0:router(config-t3)# bert pattern 2^15`	（任意）T3/E3 または T1/E1 ラインでのビット誤り率テスト（BERT）パターンの長さを指定します。インターバルの値は 1 ～ 14400 の範囲で指定できます。
ステップ 5	**bert error** [*number*] **例：** `RP/0/RSP0/CPU0:router(config-t3)# bert error 10`	ビット ストリームに追加する BERT エラーの数を指定します。範囲は 1 ～ 255 です。
ステップ 6	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-t3)# end` または `RP/0/RSP0/CPU0:router(config-t3)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 7	**exit** **例：** `RP/0/RSP0/CPU0:router(config-t3)# exit`	T3/E3 または T1/E1 コントローラ コンフィギュレーション モードを終了します。
ステップ 8	**exit** **例：** `RP/0/RSP0/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。

	コマンドまたはアクション	目的
ステップ 9	**bert** [**t3** | **e3** | **t1** | **e1**] *interface-path-id* [**channel-group** *channel-group-number*] [**error**] **start** **例：** RP/0/RSP0/CPU0:router# bert t3 0/3/0/0 start RP/0/RSP0/CPU0:router# bert t3 0/3/0/0 error	指定した T3/E3 または T1/E1 コントローラでの、設定した BERT テストを開始します。 **（注）** オプションの **error** キーワードを指定して、実行中の BERT ストリームにエラーを挿入することもできます。
ステップ 10	**bert** [**t3** | **e3** | **t1** | **e1**] *interface-path-id* [**channel-group** *channel-group-number*] **stop** **例：** RP/0/RSP0/CPU0:router# bert t3 0/3/0/0 stop	指定した T3/E3 または T1/E1 コントローラでの、設定した BERT テストを停止します。
ステップ 11	**show controllers** [**t3** | **e3** | **t1** | **e1**] *interface-path-id* **例：** RP/0/RSP0/CPU0:router# show controllers t3 0/3/0/0	設定した BERT の結果を表示します。

次の作業

*「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」*モジュールの説明に従って、テストしたコントローラに関連付けられているシリアル インターフェイスを設定します。

DS0 チャネル グループでの BERT の設定

ここでは、個々の DS0 チャネル グループでビット エラー レート テスト（BERT）のパターンをイネーブルにする手順について説明します。

前提条件

クリア チャネル T1/E1 コントローラまたはチャネライズド T3-to-T1/E1 コントローラを設定する必要があります。

手順の概要

1. **configure**
2. **controller** {**t1** | **e1**} *interface-path-id*
3. **channel-group** *channel-group-number*
4. **bert pattern** *pattern*
5. **bert interval** *time*
6. **end**
   または
   commit
7. **exit**
8. **exit**

9. **exit**
10. **bert** [**t1** | **e1**] *interface-path-id* [**channel-group** *channel-group-number*][**error**] **start**
11. **bert** [**t1** | **e1**] *interface-path-id* [**channel-group** *channel-group-number*] **stop**
12. **show controllers** [**t1** | **e1**] *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller** {**t1** | **e1**} *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t3 0/1/0/0`	コントローラ名とインスタンス ID を *rack/slot/module/port* 表記で指定し、T1 または E1 コントローラ 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	**channel-group** *channel-group-number* **例：** `RP/0/RSP0/CPU0:router(config-t1)# channel-group 1` `RP/0/RSP0/CPU0:router(config-t1-channel_group)#`	特定のチャネル グループのチャネル グループ コンフィギュレーション モードを開始します。 *channel-group-number* を、BERT を設定するチャネル グループを指す番号に置き換えます。
ステップ 4	**bert pattern** *pattern* **例：** `RP/0/RSP0/CPU0:router(config-t1-channel_group)# bert pattern 2^15`	T1 ラインで特定のビット エラー レート テスト（BERT）のパターンをイネーブルにします。すべてのコントローラおよびチャネル グループに有効なパターンには、**0s**、**1s**、**2^15**、**2^20**、**2^20-QRSS**、**2^23**、**alt-0-1** があります。T1 および E1 コントローラに有効なパターンには **1in8**、**3in24**、**55Daly**、**55Octet** があります。チャネル グループに有効なパターンには **2^11**、**2^9**、**ds0-1**、**ds0-2**、**ds0-3**、**ds0-4** があります。 **（注）** BER テストを開始するには、EXEC モードで **bert** コマンドを使用する必要があります。
ステップ 5	**bert interval** *time* **例：** `RP/0/RSP0/CPU0:router(config-t1-channel_group)# bert interval 5`	（任意）T1/E1 ラインでのビット誤り率テスト（BERT）パターンの長さを分単位で指定します。インターバルの値は 1 ～ 14400 の範囲で指定できます。

	コマンドまたはアクション	目的
ステップ 6	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-t1-channel_group)# end` または `RP/0/RSP0/CPU0:router(config-t1-channel_group)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 7	**exit** **例：** `RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit`	チャネル グループ コンフィギュレーション モードを終了します。
ステップ 8	**exit** **例：** `RP/0/RSP0/CPU0:router(config-t1)# exit`	T1 または E1 コンフィギュレーション モードを終了します。
ステップ 9	**exit** **例：** `RP/0/RSP0/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 10	**bert** [**t1** | **e1**] *interface-path-id* [**channel-group** *channel-group-number*] [**error**] **start** **例：** `RP/0/RSP0/CPU0:router# bert t1 0/3/0/0/0 start` `RP/0/RSP0/CPU0:router# bert t1 0/3/0/0/0 error`	指定したチャネル グループで、設定した BERT テストを開始します。 **（注）** オプションの **error** キーワードを指定して、実行中の BERT ストリームにエラーを挿入することもできます。

	コマンドまたはアクション	目的
ステップ 11	**bert** [**t1** | **e1**] *interface-path-id* [**channel-group** *channel-group-number*] **stop** **例：** `RP/0/RSP0/CPU0:router# bert t1 0/3/0/0/0 stop`	指定したチャネル グループで、設定した BERT テストを停止します。
ステップ 12	**show controllers** [**t1** | **e1**] *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show controllers t3 0/3/0/0`	設定した BERT の結果を表示します。

次の作業

テストしたコントローラに関連付けられるシリアル インターフェイスを設定します。手順については、*「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」*モジュール（このマニュアルで後述します）を参照してください。

T1 または E1 チャネルでのリンク ノイズ モニタの設定

ここでは、Cisco ASR 9000 シリーズ ルータで T1 または E1 チャネルに対するリンク ノイズ モニタ（LNM）を設定する方法について説明します。

前提条件

Cisco ASR 9000 シリーズ ルータで LNM を設定する前に、次の要件が満たされていることを確認してください。

- 2 ポート チャネライズド OC-12c/DS0 SPA がインストールされていること。
- 2 ポート チャネライズド OC-12/DS0 SPA が、T1 または E1 モードで動作するチャネライズド T3 コントローラとして設定されていること。手順については、「チャネル化された T3 コントローラの設定」（P.455）を参照してください。
- T1 または E1 コントローラが、単一チャネルとして 24 個または 31 個の DS0 タイム スロット全体をサポートするように設定されていること。手順については、「T1 コントローラの設定」（P.459）または「E1 コントローラの設定」（P.463）を参照してください。LNM は、フラクショナル T1 または E1 リンクではサポートされません。

制約事項

Cisco ASR 9000 シリーズ ルータで LNM を設定する前に、次の制限事項を確認してください。

- **lnm major-warning** コマンドと **lnm remove** コマンドは相互に排他的です。1 つのコントローラには、これらの LNM 機能のいずれか 1 つのみ設定できます。
- **lnm minor-warning** コマンドは、1 つのコントローラに対して、**lnm major-warning** コマンドまたは **lnm remove** コマンドとともに設定できます。
- **lnm remove** コマンドが設定されている場合、**ppp multilink minimum-active links** コマンドで設定されたしきい値までの MLPPP バンドルのリンクのみ削除されます。

手順の概要

1. **configure**
2. **controller** {**t1** | **e1**} *interface-path-id*
3. **lnm {major-warning** | **remove} [clear** | **set**][**line-code-violation** *lcv-value* [**path-code-violation** *pcv-value*]][**duration** *seconds*]
4. **lnm minor-warning** [**clear** | **set**][**line-code-violation** *lcv-value* [**path-code-violation** *pcv-value*]][**duration** *seconds*]
5. **lnm syslog**
6. **end**
   または
   commit

手順の詳細

ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller** {**t1** | **e1**} *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t1 0/1/1/0/1/1`	T1 または E1 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	**lnm** {**major-warning** | **remove**}[**clear** | **set**][**line-code-violation** *lcv-value* [**path-code-violation** *pcv-value*]][**duration** *seconds*] **例：** `RP/0/RSP0/CPU0:router(config-t1)# lnm major-warning`	（任意）リンク ノイズ モニタをイネーブルにして、T1/E1 リンクでのノイズ エラーのしきい値を指定します。これは、メジャー警告イベントやリンク削除の通知と、これらのイベントからのリカバリに使用されます。 しきい値設定とクリア両方のデフォルト値は次のとおりです。 • T1 リンクの場合：**line-code-violation** は 1544、**path-code-violation** は 320、**duration** は 10。 • E1 リンクの場合：**line-code-violation** は 2048、**path-code-violation** は 831、**duration** は 10。
ステップ 4	**lnm minor-warning** [**clear** | **set**][**line-code-violation** *lcv-value* [**path-code-violation** *pcv-value*]][**duration** *seconds*] **例：** `RP/0/RSP0/CPU0:router(config-t1)# lnm minor-warning`	（任意）リンク ノイズ モニタをイネーブルにして、T1/E1 リンクでのノイズ エラーのしきい値を指定します。これは、マイナー警告イベントの通知と、このイベントからのリカバリに使用されます。 しきい値設定とクリア両方のデフォルト値は次のとおりです。 • T1 リンクの場合：**line-code-violation** は 154、**path-code-violation** は 145、**duration** は 10。 • E1 リンクの場合：**line-code-violation** は 205、**path-code-violation** は 205、**duration** は 10。

ステップ 5	**lnm syslog** **例：** `RP/0/RSP0/CPU0:router(config-t1)# lnm syslog`	（任意）リンク ノイズ モニタのメジャーおよびマイナーのイベントとアラームのロギングをイネーブルにします。 **（注）** LNM メッセージがシステム ログとログ イベント バッファの両方に表示されるようにするには、このコマンドを使用する必要があります。
ステップ 6	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-t1)# end` または `RP/0/RSP0/CPU0:router(config-t1)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

リンク ノイズ モニタリングの設定およびステータスの確認

LNM の設定、状態情報、統計情報およびイベントを確認するには、次の例に示すように、**show controllers lnm** コマンドを使用します。

（注） **lnm remove** コマンドが設定されている場合、show controllers の出力のヘッダーには「Remove」が表示され、「major-warning」および「Major-Warn」の代わりにイベントが表示されます。

```
RP/0/RSP0/CPU0:Router# show controllers t1 0/1/1/0/1/1 lnm all
Thu May 13 10:28:26.474 PDT

Controller T1 0/1/1/0/1/1

Syslog   Monitoring type  State     Thresholds (lcv/pcv/duration)
-------------------------------------------------------------------
enabled   minor-warning   stable    Set(  15/  15/   4) Clear(  15/  15/   4)
          major-warning   stable    Set( 154/ 145/   4) Clear( 154/ 145/   4)

  Monitoring type            Minor-Warn      Major-Warn
  ---------------            ------------    ------------
  Create                     1               1
  Update                     0               0
  Delete                     0               0
  Clear                      0               0
  Noise Crossed              1               1
```

```
  Noise Cleared              1              1

Last Five Events
------------------------------------------------------------------
MINWARNCROSS: Noise crossed minor-warn threshold at Thu May 13 09:54:10 2010
MAJWARNCROSS: Noise crossed major-warn threshold at Thu May 13 09:54:11 2010
MAJWARNCLEAR: Noise cleared major-warn threshold at Thu May 13 10:27:25 2010
MINWARNCLEAR: Noise cleared minor-warn threshold at Thu May 13 10:28:14 2010
```

リンク ノイズ モニタリングの状態および統計情報のクリア

clear controller lnm コマンドを使用すると、LNM 状態をリセットすることや、統計情報をクリアしてゼロにリセットすることができます。

通常、LNM コントローラの状態をクリアする必要はありません。**state** オプションを指定すると LNM 設定がリセットされ、その結果としてシステム内の現在の LNM 状態が更新されます。したがって、通常の状態では、コントローラがアラーム状態の場合、リセットはアラーム状態を報告し続けるはずです。または、コントローラのアラームがすべてクリアされれば、リセットは安定状態を示します。**clear controller lnm state** コマンドを使用しても、実際にはアラームは何もクリアされませんが、システム内のアラーム値がリフレッシュされます。したがって、このコマンドは、レポートされたコントローラの状態が実際のコントローラの状態と同期してていない場合に使用できます。

LNM の状態をリセットするには、次の例に示すように **clear controller lnm** コマンドを使用します。

```
RP/0/RSP0/CPU0:Router# clear controller t1 0/1/0/0/1/1 lnm state
```

LNM 統計情報をクリアしてカウンタをゼロにリセットするには、次の例に示すように **clear controller lnm** コマンドを使用します。

```
RP/0/RSP0/CPU0:Router# clear controller t1 0/1/0/0/1/1 lnm statistics

RP/0/RSP0/CPU0:Router# show controller T1 0/1/0/1/1/1 lnm statistics
Thu May 13 11:26:20.991 PDT

Controller T1 0/1/0/1/1/1

  Monitoring type            Minor-Warn      Major-Warn
  ---------------            ------------    ------------
  Create                     0               0
  Update                     0               0
  Delete                     0               0
  Clear                      0               0
  Noise Crossed              0               0
  Noise Cleared              0               0
```

設定例

ここでは、次の例を示します。

- 「クリア チャネル T3 コントローラの設定：例」（P.478）
- 「T3 コントローラでのチャネル化した T1 コントローラの設定：例」（P.478）
- 「T3 コントローラでの BERT の設定：例」（P.479）
- 「T1 コントローラでのリンク ノイズ モニタリングの設定：例」（P.480）
- 「T3 チャネルの QoS：例」（P.481）

クリア チャネル T3 コントローラの設定：例

次に、クリア チャネル T3 コントローラの設定例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)#controller T3 0/3/2/0
RP/0/RSP0/CPU0:router(config-t3)#clock source internal
RP/0/RSP0/CPU0:router(config-t3)#mode serial
RP/0/RSP0/CPU0:router(config-t3)#cablelength 4
RP/0/RSP0/CPU0:router(config-t3)#framing c-bit
RP/0/RSP0/CPU0:router(config-t3)#commit
```

T3 コントローラでのチャネル化した T1 コントローラの設定：例

次に、28 T1 コントローラがチャネル化されている T3 コントローラの設定例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# controller T3 0/3/0/0
RP/0/RSP0/CPU0:router(config-t3)# mode t1
RP/0/RSP0/CPU0:router(config-t3)# framing m23
RP/0/RSP0/CPU0:router(config-t3)# cablelength 11
RP/0/RSP0/CPU0:router(config-t3)# clock source line
RP/0/RSP0/CPU0:router(config-t3)#commit
RP/0/RSP0/CPU0:router(config-t3)#exit
RP/0/RSP0/CPU0:router(config)# exit
RP/0/RSP0/CPU0:router# show controllers T1 ?

0/3/0/0/0   T1 Interface Instance
  0/3/0/0/1   T1 Interface Instance
  0/3/0/0/10  T1 Interface Instance
  0/3/0/0/11  T1 Interface Instance
  0/3/0/0/12  T1 Interface Instance
  0/3/0/0/13  T1 Interface Instance
  0/3/0/0/14  T1 Interface Instance
  0/3/0/0/15  T1 Interface Instance
  0/3/0/0/16  T1 Interface Instance
  0/3/0/0/17  T1 Interface Instance
  0/3/0/0/18  T1 Interface Instance
  0/3/0/0/19  T1 Interface Instance
  0/3/0/0/2   T1 Interface Instance
  0/3/0/0/20  T1 Interface Instance
  0/3/0/0/21  T1 Interface Instance
  0/3/0/0/22  T1 Interface Instance
  0/3/0/0/23  T1 Interface Instance
  0/3/0/0/24  T1 Interface Instance
  0/3/0/0/25  T1 Interface Instance
  0/3/0/0/26  T1 Interface Instance
  0/3/0/0/27  T1 Interface Instance
  0/3/0/0/3   T1 Interface Instance
  0/3/0/0/4   T1 Interface Instance
  0/3/0/0/5   T1 Interface Instance
 --More--
!
RP/0/RSP0/CPU0:router#
RP/0/RSP0/CPU0:router(config)#configure
RP/0/RSP0/CPU0:router(config)# controller t1 0/3/0/0/0
RP/0/RSP0/CPU0:router(config-t1)# channel-group 0
RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 1-24
RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit
RP/0/RSP0/CPU0:router(config-t1)# exit
RP/0/RSP0/CPU0:router(config)# controller t1 0/3/0/0/1
```

```
RP/0/RSP0/CPU0:router(config-t1)# channel-group 0
RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 1-24
RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit
RP/0/RSP0/CPU0:router(config-t1)# exit
RP/0/RSP0/CPU0:router(config)# controller t1 0/3/0/0/2
RP/0/RSP0/CPU0:router(config-t1)# channel-group 0
RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 1-12
RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit
RP/0/RSP0/CPU0:router(config-t1)# channel-group 1
RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 13-24
RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit
RP/0/RSP0/CPU0:router(config-t1)# exit
RP/0/RSP0/CPU0:router(config)# controller t1 0/3/0/0/3
RP/0/RSP0/CPU0:router(config-t1)# channel-group 0
RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 1-6
RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit
RP/0/RSP0/CPU0:router(config-t1)# channel-group 1
RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 7-12
RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit
RP/0/RSP0/CPU0:router(config-t1)# channel-group 2
RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 13-18
RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit
RP/0/RSP0/CPU0:router(config-t1)# channel-group 3
RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 19-24
RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit
RP/0/RSP0/CPU0:router(config-t1-channel_group)#commit
```

T3 コントローラでの BERT の設定：例

次に、T3 コントローラで BERT を設定し、BERT の結果を表示する例を示します。

```
RP/0/RSP0/CPU0:router# config
RP/0/RSP0/CPU0:router(config)# controller t3 0/3/0/1
RP/0/RSP0/CPU0:router(config-t3)# bert pattern 0s

Run bert from exec mode for the bert config to take effect

RP/0/RSP0/CPU0:router(config-t3)#exit
RP/0/RSP0/CPU0:router(config)# exit

Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]
RP/0/RSP0/CPU0:router# bert t3 0/3/0/1 start


RP/0/RSP0/CPU0:router# bert t3 0/3/0/1 stop


RP/0/RSP0/CPU0:router# show controllers t3 0/3/0/1

  T30/3/0/1 is up
  No alarms detected.
  MDL transmission is disabled
    EIC: , LIC: , FIC: , UNIT:
     Path FI:
     Idle Signal PORT_NO:
     Test Signal GEN_NO:
  FEAC code received: No code is being received
  Framing is C-BIT Parity, Line Code is B3ZS, Clock Source is Internal
  Data in current interval (108 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation, 0 P-bit Err Secs
     0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
```

```
    0 Unavailable Secs, 0 Line Errored Secs
    0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
  Data in Interval 1:
    0 Line Code Violations, 0 P-bit Coding Violation
    0 C-bit Coding Violation, 0 P-bit Err Secs
    0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
    0 Unavailable Secs, 0 Line Errored Secs
    0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
  Data in Interval 2:
    0 Line Code Violations, 0 P-bit Coding Violation
    0 C-bit Coding Violation, 0 P-bit Err Secs
    0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
    0 Unavailable Secs, 0 Line Errored Secs
    0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
  Data in Interval 3:
    0 Line Code Violations, 0 P-bit Coding Violation
    0 C-bit Coding Violation, 0 P-bit Err Secs
    0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
    0 Unavailable Secs, 0 Line Errored Secs
    0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
```

T1 コントローラでのリンク ノイズ モニタリングの設定：例

次に、リンクの LNM を設定する前に、24 の DS0 タイムスロットすべてを 1 つのチャネルとして使用して、T1 コンフィギュレーション モードのチャネライズド T3 コントローラを設定する例を示します。この例では、表示される値は実際にはしきい値設定のシステム デフォルトです。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# controller T3 0/1/1/0/1
RP/0/RSP0/CPU0:router(config-t3)# mode t1
RP/0/RSP0/CPU0:router(config-t3)# framing m23
RP/0/RSP0/CPU0:router(config-t3)# cablelength 11
RP/0/RSP0/CPU0:router(config-t3)# clock source line
RP/0/RSP0/CPU0:router(config-t3)#commit
RP/0/RSP0/CPU0:router(config-t3)#exit
RP/0/RSP0/CPU0:router(config)# controller t1 0/1/1/0/1/1
RP/0/RSP0/CPU0:router(config-t1)# channel-group 0
RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 1-24
RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit
RP/0/RSP0/CPU0:router(config-t1)# lnm syslog
RP/0/RSP0/CPU0:router(config-t1)# lnm major-warning set line-code-violation 1544
path-code-violation 320 duration 10
RP/0/RSP0/CPU0:router(config-t1)# lnm minor-warning set line-code-violation 154
path-code-violation 145 duration 10
```

次に、リンクの LNM を設定する前に、24 の DS0 タイムスロットすべてを 1 つのチャネルとして使用して、T1 コンフィギュレーション モードのチャネライズド T3 コントローラを設定する例を示します。この例では、表示される値は実際に set しきい値のシステム デフォルトであり、これらのしきい値を超過すると、PPP にノイズ属性が通知されて MLPPP リンクが削除されるように LNM が設定されています。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# controller T3 0/1/1/0/1
RP/0/RSP0/CPU0:router(config-t3)# mode t1
RP/0/RSP0/CPU0:router(config-t3)# framing m23
RP/0/RSP0/CPU0:router(config-t3)# cablelength 11
RP/0/RSP0/CPU0:router(config-t3)# clock source line
RP/0/RSP0/CPU0:router(config-t3)#commit
RP/0/RSP0/CPU0:router(config-t3)#exit
RP/0/RSP0/CPU0:router(config)# controller t1 0/1/1/0/1/1
RP/0/RSP0/CPU0:router(config-t1)# channel-group 0
```

```
RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 1-24
RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit
RP/0/RSP0/CPU0:router(config-t1)# lnm syslog
RP/0/RSP0/CPU0:router(config-t1)# lnm remove set line-code-violation 1544
path-code-violation 320 duration 10
RP/0/RSP0/CPU0:router(config-t1)# lnm minor-warning set line-code-violation 154
path-code-violation 145 duration 10
```

T3 チャネルの QoS：例

T3 チャネルの QoS は、PPP および HDLC カプセル化の両方でサポートされます。次の例では、T3 インターフェイスの一般的な QoS 設定を示します。

```
class-map VOIP
match dscp EF
end-class-map
class-map OAM
match dscp AF43
end-class-map
!
Policy-map T3-no-priority
class OAM
bandwidth percent 30
!
class class-default
!
end-policy-map
!
Policy-map T3-priority
class VOIP
priority level 1
     police rate percent 60
!
class OAM
bandwidth percent 30
!
class class-default
!
end-policy-map
```

その他の関連資料

ここでは、T3 および T1 コントローラに関する参考資料について説明します。

関連資料

関連項目	参照先
Cisco IOS XR マスター コマンド リファレンス	*『Cisco IOS XR Master Commands List』*
Cisco IOS XR インターフェイス コンフィギュレーション コマンド	*『Cisco IOS XR Interface and Hardware Component Command Reference』*

関連項目	参照先
Cisco IOS XR ソフトウェアを使用した初期システムブートアップとルータの設定情報	『*Cisco IOS XR Getting Started Guide*』
Cisco IOS XR AAA サービス構成情報	『*Cisco IOS XR System Security Configuration Guide*』および『*Cisco IOS XR System Security Command Reference*』

標準

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

MIB

MIB	MIB のリンク
• IF-MIB • DS3-MIB • CISCO-DS3-MIB • DS1-MIB **（注）** 4 ポート クリア チャネル T3/E3 SPA ではサポートされていません。 • エンティティ MIB	Cisco IOS XR ソフトウェアを使用して選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	—

シスコのテクニカル サポート

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

その他の関連資料

Cisco ASR 9000 シリーズ ルータ ソフトウェアでの高密度波長分割多重コントローラの設定

このモジュールでは、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータでの高密度波長分割多重（DWDM）コントローラの設定について説明します。

DWDM は、既存の光ファイバに基づいて、帯域幅を増やすために使用される光のテクノロジーです。DWDM は、サポートされる 10 ギガビット イーサネット（GE）ラインカードに対して設定できます。DWDM コントローラを設定した後は、関連する 10 ギガビット イーサネット インターフェイスを設定できます。

DWDM コントローラ インターフェイス設定の機能履歴

リリース	変更内容
リリース 3.9.0	この機能が Cisco ASR 9000 シリーズ ルータで次のカードに対して導入されました。 • Cisco 8 ポート 10 ギガビット イーサネット ラインカード(A9K-8T-L および -E) • Cisco 2 ポート 10 ギガビット イーサネット + 20 ポート ギガビット イーサネット コンビネーション ラインカード（A9K-2T20GE-L）
リリース 3.9.1	次のカードのサポートが追加されました。 • Cisco 8 ポート 10 ギガビット イーサネット ラインカード（A9K-8T-B） • Cisco 2 ポート 10 ギガビット イーサネット + 20 ポート ギガビット イーサネット コンビネーション ラインカード（A9K-2T20GE-B および -E）
リリース 4.0.0	IPoDWDM 予防的保護のサポートが次のカードに対して追加されました。 • Cisco 8 ポート 10 ギガビット イーサネット ラインカード（A9K-8T-L、-B および -E） • Cisco 2 ポート 10 ギガビット イーサネット + 20 ポート ギガビット イーサネット コンビネーション ラインカード（A9K-2T20GE-L、-B および -E）

リリース 4.2.1	IPoDWDM 予防的保護のサポートが次のモジュール ポート アダプタに対して追加されました。 • A9K-MPA-4x10GE • A9K-MPA-2X10GE
リリース 4.2.3	IPoDWDM 予防的保護のサポートが次のモジュール ポート アダプタに対して追加されました。 • A9K-MPA-2X40GE • A9K-MPA-1X40GE

内容

DWDM コントローラ インターフェイスを設定するための前提条件

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。

DWDM コントローラを設定する前に、DWDM をサポートする次のカードのいずれかがインストールされていることを確認してください。

- Cisco 8 ポート 10 ギガビット イーサネット ラインカード
- Cisco 2 ポート 10 ギガビット イーサネット + 20 ポート ギガビット イーサネット コンビネーション ラインカード

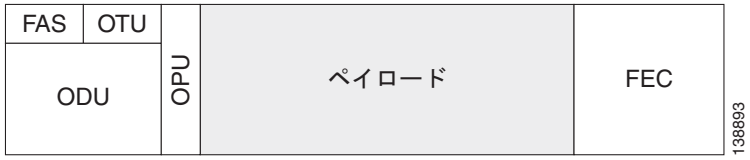
DWDM コントローラに関する情報

Cisco IOS XR ソフトウェアの DWDM のサポートは、ITU-T G.709 に規定されている光トランスポート ネットワーク（OTN）プロトコルに基づいています。この規格は、SONET/SDH テクノロジーと DWDM の多波長ネットワークの利点を兼ね備えています。また、使用するリジェネレータの数を減らすことで、ネットワーク コストを減らすことができる、前方誤り訂正（FEC）の機能も備えています。

マルチサービス トランスポートを使用するために、OTN はラップされたオーバーヘッド（OH）の概念を使用します。この構造について説明します。

- 光チャネル ペイロード ユニット（OPU）の OH 情報が情報ペイロードに追加され、OPU が形成されます。OPU OH には、クライアント信号のアダプテーションをサポートする情報が含まれます。
- 光チャネル データ ユニット（ODU）の OH が OPU に追加され、ODU が形成されます。ODU OH には、光チャネルをサポートするメンテナンス機能と操作機能の情報が含まれます。
- 光チャネル トランスポート ユニット（OTU）の OH と FEC が追加され、OTU が形成されます。OTU OH には、1 つまたは複数の光チャネル接続を経由するトランスポートをサポートする操作機能の情報が含まれます
- 光チャネル（OCh）の OH が追加され、OCh が形成されます。OCh には OTN 管理機能があり、OPU、ODU、OTU、およびフレーム整列信号（FAS）という 4 つのパートが含まれます。図 33 を参照してください。

図 33　　OTN 光チャネルの構造



IPoDWDM について

Cisco IOS XR ソフトウェアには、高密度波長分割多重（IPoDWDM）機能ガ含まれています。

IPoDWDM は、次のハードウェア デバイスでサポートされます。

- Cisco 8 ポート 10 ギガビット イーサネット ラインカード
- Cisco 2 ポート 10 ギガビット イーサネット + 20 ポート ギガビット イーサネット コンビネーション ラインカード

IPoDWDM は、現時点では次のソフトウェア機能を提供します。

- 予防的メンテナンス

予防的メンテナンス

予防的なメンテナンスは、前方誤り訂正高速再ルーティング（FEC-FRR）を自動的にトリガーします。予防的メンテナンスを行うには、レイヤ 0（L0）とレイヤ 3（L3）の間に協調型メンテナンスが必要です。
L0 は DWDM 光レイヤです。FEC-FRR は L3 保護メカニズムです。FEC-FRR は、伝送中に誘発されるか、劣化信号に起因するエラーを、発生前に検出して修正します。

システム管理者は、IPoDWDM の次の機能を設定できます。

- 光レイヤ DWDM ポート（「光レイヤ DWDM ポートの設定」（P.495）を参照）。
- DWDM の光ポートの管理状態（「DWDM 光ポートの管理状態の設定」（P.497）を参照）。
- FEC-FRR のトリガーしきい値、ウィンドウ サイズ、復帰しきい値、復帰ウィンドウ サイズ（「予防的 FEC-FRR トリガーの設定」（P.499）を参照）。

FEC-FRR のトリガー

FEC-FRR は、次のアラームによってトリガーされるように設定できます。

- ais：アラーム表示信号（AIS）

- bdi：後方障害表示（BDI）
- *bdiO：後方障害表示：オーバーヘッド（BDI-O）
- *bdiP：後方障害表示：ペイロード（BDI-P）
- *deg：劣化（DEG）
- lck：ロック（LCK）
- lof：フレーム損失（LOF）
- lom：複数フレーム損失
- los：信号消失（LOS）
- *losO：信号消失：オーバーヘッド（LOS-O）
- *losP：信号消失：ペイロード（LOS-P）
- oci：オープン接続表示（OCI）
- plm：ペイロード不一致（PLM）
- *ssf：サーバ信号障害（SSF）
- *ssfO：サーバ信号障害：オーバーヘッド（SSF-O）
- *ssfP：サーバ信号障害：ペイロード（SSF-P）
- tim：トレース ID 不一致（TIM）

信号のロギング

EC、UC、アラームなどの DWDM 統計データは、DWDM ラインカードでログ ファイルに収集され保存されます。

DWDM コントローラの設定方法

DWDM コントローラは、Cisco IOS XR ソフトウェアのコンフィギュレーション スペースの物理層のコントロール要素で設定します。この設定を行うには、**controller dwdm** コマンドを使用します。設定については、次のタスクで説明します。

- 「G.709 パラメータの設定」（P.488）

（注） ギガビット イーサネット インターフェイスのすべてのインターフェイス コンフィギュレーション タスクは、インターフェイス コンフィギュレーション モードで実行する必要があります。

G.709 パラメータの設定

ここでは、アラートと FEC のアラーム表示およびしきい値をカスタマイズする方法について説明します。デフォルト値が実際のインストールに合っていない場合にのみ、このタスクを使用してください。

前提条件

loopback コマンドおよび **g709 fec** コマンドは、コントローラがシャットダウン状態の場合にのみ使用できます。**admin-state** コマンドを使用します。

手順の概要

1. **configure**
2. **controller dwdm** *interface-path-id*
3. **admin-state maintenance**
   または
   admin-state out-of-service
4. **commit**
5. **loopback** {**internal** | **line**}
6. **g709 fec** {**disable** | **enhanced** | **standard**}
7. **g709** {**odu** | **otu**} **report** *alarm* **disable**
8. **g709 otu overhead tti** {**expected** | **sent**} {**ascii** | **hex**} *tti-string*
9. **end**
   または
   commit
10. **admin-state in-service**
11. **show controllers dwdm** *interface-path-id* **g709**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:Router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller dwdm** *interface-path-id* **例：** `RP/0/RSP0/CPU0:Router(config)# controller dwdm 0/1/0/0`	*rack/slot/module/port* 表記で DWDM コントローラ名を指定し、DWDM 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	**admin-state maintenance** または **admin-state out-of-service** **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# admin-state out-of-service`	DWDM コントローラをディセーブルにします。DWDM コンフィギュレーション コマンドを使用する前に、コントローラをディセーブルにする必要があります。
ステップ 4	**commit** **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# commit`	設定変更を保存します。これで、前のステップのシャットダウンが実行されます。コントローラがシャットダウンすると、設定を進めることができます。

	コマンドまたはアクション	目的
ステップ 5	**loopback** {**internal** | **line**} **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# loopback internal`	（任意）ループバック モードの DWDM コントローラを設定します。
ステップ 6	**g709 fec** {**disable** | **standard**} **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# g709 fec disable`	（任意）DWDM コントローラの FEC を設定します。デフォルトでは、拡張 FEC がイネーブルです。
ステップ 7	**g709** {**odu** | **otu**} **report** *alarm* **disable** **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# g709 odu bdi disable`	（任意）DWDM コントローラのコンソールに対する、選択した ODU アラームまたは OTU アラームのロギングをディセーブルにします。デフォルトでは、すべてのアラームがコンソールにロギングされます。
ステップ 8	**g709 otu overhead tti** {**expected** | **sent**} {**ascii** | **hex**} *tti-string* **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# g709 otu overhead tti expected ascii test OTU 5678`	**show controller dwdm** コマンドで表示される伝送または予想の Trail Trace Identifier（TTI）を設定します。
ステップ 9	**end** または **commit** **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# end` または `RP/0/RSP0/CPU0:Router(config-dwdm)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

	コマンドまたはアクション	目的
ステップ 10	**admin-state in-service** **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# admin-state in-service`	DWDM ポートをイン サービス（IS）にして、すべての通常の動作をサポートするようにします。
ステップ 11	**show controllers dwdm** *interface-path-id* **g709** **例：** `RP/0/RSP0/CPU0:Router# show controller dwdm 0/1/0/0 optics`	G.709 OTN プロトコルのアラームおよびビット エラーのカウンタと共に、FEC 統計情報としきい値ベースのアラートを表示します。

次の作業

ギガビット イーサネット インターフェイスのすべてのインターフェイス コンフィギュレーション タスクは、インターフェイス コンフィギュレーション モードで実行する必要があります。詳細については、本書の対応するモジュールを参照してください。

DWDM コントローラでパフォーマンス モニタリングを実行する方法

パフォーマンス モニタリング パラメータは、問題を早期に検出するためのパフォーマンス データを収集および格納し、しきい値を設定し、報告するために使用されます。しきい値は、各パフォーマンス モニタリング パラメータのエラー レベルを設定するために使用されます。蓄積サイクルで、パフォーマンス監視モニタリング パラメータの現在の値が、対応するしきい値に達した場合、または超過した場合、しきい値超過アラート（TCA）を生成できます。TCA によって、パフォーマンス低下を早期に検出できます。

パフォーマンス モニタリングの統計情報は 15 分ベースで蓄積され、各 15 分の開始時に同期されます。また、深夜 12 時に始まる日次単位でも統計情報は蓄積されます。履歴カウントは、33 回の 15 分インターバルと 2 回の日次インターバルで維持されます。

パフォーマンス モニタリングについては、次のタスクで説明します。

- 「DWDM コントローラのパフォーマンス モニタリングの設定」（P.491）

DWDM コントローラのパフォーマンス モニタリングの設定

ここでは、DWDM コントローラでパフォーマンス モニタリングを設定する方法とパフォーマンス パラメータを表示する方法について説明します。

手順の概要

1. **configure**
2. **controller dwdm** *interface-path-id*
3. **pm** {**15-min** | **24-hour**} **fec threshold** {**ec-bits** | **uc-words**} *threshold*
4. **pm** {**15-min** | **24-hour**} **optics threshold** {**lbc** | **opr** | **opt**} {**max** | **min**} *threshold*

5. **pm** {**15-min** | **24-hour**} **otn threshold** *otn-parameter threshold*
6. **pm** {**15-min** | **24-hour**} **fec report** {**ec-bits** | **uc-words**} **enable**
7. **pm** {**15-min** | **24-hour**} **optics report** {**lbc** | **opr** | **opt**} {**max-tca** | **min-tca**} **enable**
8. **pm** {**15-min** | **24-hour**} **otn report** *otn-parameter* **enable**
9. **end**
   または
   commit
10. **show controllers dwdm** *interface-path-id* **pm history** [**15-min** | **24-hour** | **fec** | **optics** | **otn**]
11. **show controllers dwdm** *interface-path-id* **pm interval** {**15-min** | **24-hour**} [**fec** | **optics** | **otn**] *index*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:Router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller dwdm** *interface-path-id* **例：** `RP/0/RSP0/CPU0:Router(config)# controller dwdm 0/1/0/0`	*rack/slot/module/port* 表記で DWDM コントローラ名を指定し、DWDM 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	**pm** {**15-min**	**24-hour**} **fec threshold** {**ec-bits**	**uc-words**} *threshold* **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min fec threshold ec-bits 49000000` `RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min fec threshold uc-words xxxxxx`	FEC 層で特定のパラメータのパフォーマンス モニタリングを設定します。
ステップ 4	**pm** {**15-min**	**24-hour**} **optics threshold** {**lbc**	**opr**	**opt**} {**max**	**min**} *threshold* **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold opt max xxx` `RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold lbc min xxx`	光ファイバ層で特定のパラメータのパフォーマンス モニタリングを設定します。

	コマンドまたはアクション	目的
ステップ 5	**pm** {**15-min** | **24-hour**} **otn threshold** *otn-parameter threshold* **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min otn threshold bbe-pm-ne xxx` `RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min otn threshold es-sm-fe xxx`	OTN 層で特定のパラメータのパフォーマンス モニタリングを設定します。次の OTN パラメータを指定できます。 • **bbe-pm-fe** - 遠端のパス モニタリングのバックグラウンド ブロック エラー（BBE-PM） • **bbe-pm-ne** - 近端のパス モニタリングのバックグラウンド ブロック エラー（BBE-PM） • **bbe-sm-fe** - 遠端のセクション モニタリングのバックグラウンド ブロック エラー（BBE-SM） • **bbe-sm-ne** - 近端のセクション モニタリングのバックグラウンド ブロック エラー（BBE-SM） • **bber-pm-fe** - 遠端のパス モニタリングのバックグラウンド ブロック エラー率（BBER-PM） • **bber-pm-ne** - 近端のパス モニタリングのバックグラウンド ブロック エラー率（BBER-PM） • **bber-sm-fe** - 遠端のセクション モニタリングのバックグラウンド ブロック エラー率（BBER-SM） • **bber-sm-ne** - 近端のセクション モニタリングのバックグラウンド ブロック エラー率（BBER-SM） • **es-pm-fe** - 遠端のパス モニタリングのエラー秒数（ES-PM） • **es-pm-ne** - 近端のパス モニタリングのエラー秒数（ES-PM） • **es-sm-fe** - 遠端のセクション モニタリングのエラー秒数（ES-SM） • **es-sm-ne** - 近端のセクション モニタリングのエラー秒数（ES-SM） • **esr-pm-fe** - 遠端のパス モニタリングのエラー秒数比（ESR-PM） • **esr-pm-ne** - 近端のパス モニタリングのエラー秒数比（ESR-PM） • **esr-sm-fe** - 遠端のセクション モニタリングのエラー秒数比（ESR-SM） • **esr-sm-ne** - 近端のセクション モニタリングのエラー秒数比（ESR-SM） • **fc-pm-fe** - 遠端のパス モニタリングの障害カウント（FC-PM） • **fc-pm-ne** - 近端のパス モニタリングの障害カウント（FC-PM） • **fc-sm-fe** - 遠端のセクション モニタリングの障害カウント（FC-SM） • **fc-sm-ne** - 近端のセクション モニタリングの障害カウント（FC-SM）

	コマンドまたはアクション	目的
		• **ses-pm-fe** - 遠端のパス モニタリングの重大エラー秒数（SES-PM） • **ses-pm-ne** - 近端のパス モニタリングの重大エラー秒数（SES-PM） • **ses-sm-fe** - 遠端のセクション モニタリングの重大エラー秒数（SES-SM） • **ses-sm-ne** - 近端のセクション モニタリングの重大エラー秒数（SES-SM） • **sesr-pm-fe** - 遠端のパス モニタリングの重大エラー秒数比（SESR-PM） • **sesr-pm-ne** - 近端のパス モニタリングの重大エラー秒数比（SESR-PM） • **sesr-sm-fe** - 遠端のセクション モニタリングの重大エラー秒数比（SESR-SM） • **sesr-sm-ne** - 近端のセクション モニタリングの重大エラー秒数比（SESR-SM） • **uas-pm-fe** - 遠端のパス モニタリングの使用不可秒数（UAS-PM） • **uas-pm-ne** - 近端のパス モニタリングの使用不可秒数（UAS-PM） • **uas-sm-fe** - 遠端のセクション モニタリングの使用不可秒数（UAS-SM） • **uas-sm-ne** - 近端のセクション モニタリングの使用不可秒数（UAS-SM）
ステップ 6	**pm** {**15-min** | **24-hour**} **fec report** {**ec-bits** | **uc-words**} **enable** **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min fec report ec-bits enable` `RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min fec report uc-words enable`	FEC 層で特定のパラメータについて TCA の生成を設定します。
ステップ 7	**pm** {**15-min** | **24-hour**} **optics report** {**lbc** | **opr** | **opt**} {**max-tca** | **min-tca**} **enable** **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min optics report opt enable` `RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min optics report lbc enable`	光ファイバ層で特定のパラメータについて TCA の生成を設定します。

	コマンドまたはアクション	目的
ステップ 8	**pm** {**15-min** | **24-hour**} **otn report** *otn-parameter* **enable** **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min otn report bbe-pm-ne enable` `RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min otn report es-sm-fe enable`	OTN 層で特定のパラメータについて TCA の生成を設定します。OTN パラメータについては、ステップ 5 を参照してください。
ステップ 9	**end** または **commit** **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# end` または `RP/0/RSP0/CPU0:Router(config-dwdm)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

Internet Protocol over Dense Wavelength-Division Multiplexing（IPoDWDM）の設定

ここでは、次の設定手順について説明します。

光レイヤ DWDM ポートの設定

光レイヤ DWDM ポートを設定するには、次の手順を使用します。

手順の概要

1. **configure**
2. **controller dwdm** *interface-path-id*
3. **network port id** *id-number*
4. **network connection id** *id-number*
5. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:Router# config`	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	**controller dwdm** *interface-path-id* **例：** `RP/0/RSP0/CPU0:Router(config)# controller dwdm 0/1/0/1`	DWDM コントローラを指定し、DWDM コントローラ モードを開始します。
ステップ 3	**network port id** *id-number* **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# network port id 1/0/1/1`	マルチサービス トランスポート プロトコル（MSTP）のポートに識別番号を割り当てます。

	コマンドまたはアクション	目的
ステップ 4	**network connection id** *id-number* **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# network connection id 1/1/1/1`	マルチサービス トランスポート プロトコル（MSTP）の接続 ID を設定します。
ステップ 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# end` または `RP/0/RSP0/CPU0:Router(config-dwdm)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

DWDM 光ポートの管理状態の設定

次の手順を使用して、管理状態を設定し、必要に応じてメンテナンス禁止フラグを設定します。

手順の概要

1. **configure**
2. **controller dwdm** *interface-path-id*
3. **admin-state** {**in-service** | **maintenance** | **out-of-service**}
4. **exit**
5. **interface tengige** *interface-path-id*
6. **maintenance disable**
7. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** RP/0/RSP0/CPU0:Router# config	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	**controller dwdm** *interface-path-id* **例：** RP/0/RSP0/CPU0:Routerconfig)# controller dwdm 0/1/0/1	DWDM コントローラを指定し、DWDM コントローラ モードを開始します。
ステップ 3	**admin-state** {**in-service** | **maintenance** | **out-of-service**} **例：** RP/0/RSP0/CPU0:Router(config-dwdm)# admin-state maintenance	トランスポート管理状態を指定します。
ステップ 4	**exit** **例：** RP/0/RSP0/CPU0:Router(config-dwdm)# exit	前のモードに戻ります。
ステップ 5	**interface pos** *interface-path-id* または **interface tengige** *interface-path-id* **例：** RP/0/RSP0/CPU0:Router(config)# interface pos 1/0/1/1 または RP/0/RSP0/CPU0:Router(config)# interface tengige 1/0/1/1	インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 6	**maintenance disable** **例：** `RP/0/RSP0/CPU0:Router(config-if)# maintenance disable`	メンテナンス禁止フラグをプロビジョニングします。これ以降は、インターフェイスに対するメンテナンス アクティビティを実行できなくなります。
ステップ 7	**end** または **commit** **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# end` または `RP/0/RSP0/CPU0:Router(config-dwdm)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

予防的 FEC-FRR トリガーの設定

前方誤り訂正高速再ルーティング（FEC-FRR）の自動トリガーを設定するには、次の手順に従います。

手順の概要

1. **configure**
2. **controller dwdm** *interface-path-id*
3. **proactive**
4. **logging signal** *file-name*
5. **proactive trigger threshold** *x-coefficient y-power*
6. **proactive trigger window** *window*
7. **proactive revert threshold** *x-coefficient y-power*
8. **proactive revert window** *window*
9. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:Router# config`	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	**controller dwdm** *interface-path-id* **例：** `RP/0/RSP0/CPU0:Router(config)# controller dwdm 0/1/0/1`	DWDM コントローラを指定し、DWDM コントローラ モードを開始します。
ステップ 3	**proactive** **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# proactive enable`	FEC-FRR の自動トリガーをイネーブルにします。
ステップ 4	**logging signal** *file-name* **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# logging signal LogFile1`	FEC-FRR の 10 ミリ秒予防的モニタリングをイネーブルにします。
ステップ 5	**proactive trigger threshold** *x-coefficient y-power* **例：** `RP/0/RSP0/CPU0:Routerconfig-dwdm)# proactive trigger threshold 1 9`	FEC-FRR のトリガーしきい値を *x*E-*y* の形式で設定します。
ステップ 6	**proactive trigger window** *window* **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# proactive trigger window 10000`	FRR がトリガーされるトリガー ウィンドウを設定します（ミリ秒単位）。
ステップ 7	**proactive revert threshold** *x-coefficient y-power* **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# proactive revert threshold 1 9`	復帰しきい値（*x*E-*y* の形式）を設定します。このしきい値は、FEC-FRR ルートから元のルートへの復帰をトリガーします。

	コマンドまたはアクション	目的
ステップ 8	**proactive revert window** *window* **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# proactive revert window 600000`	FEC-FRR ルートから元のルートへの復帰がトリガーされる復帰ウィンドウを設定します。
ステップ 9	**end** または **commit** **例：** `RP/0/RSP0/CPU0:Router(config-dwdm)# end` または `RP/0/RSP0/CPU0:Router(config-dwdm)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

設定例

ここでは、次の設定例について説明します。

- 「レーザーのオン：例」（P.501）
- 「レーザーのオフ：例」（P.502）
- 「DWDM コントローラの設定：例」（P.502）
- 「DWDM のパフォーマンス モニタリング：例」（P.502）
- 「IPoDWDM 設定：例」（P.503）

レーザーのオン：例

（注） これは必須の設定です。DWDM カードはこの設定なしでは動作しません。

次の例では、レーザーをオンにして DWDM ポートをイン サービス（IS）状態にする方法を示します。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/1
RP/0/RP0/CPU0:Router(config-dwdm)# admin-state in-service
```

```
RP/0/RP0/CPU0:Router(config-dwdm)# commit
```

レーザーのオフ：例

次に、レーザーをオフにし、すべてのトラフィックを停止して DWDM ポートをアウトオブ サービス（OOS）状態にする方法の例を示します。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:Router(config)# controller dwdm 0/1/0/1
RP/0/RP0/CPU0:Router(config-dwdm)# admin-state out-of-service
RP/0/RP0/CPU0:Router(config-dwdm)# commit
```

DWDM コントローラの設定：例

次の例は、アラートおよび FEC のアラーム表示としきい値をカスタマイズする方法です。

```
RP/0/RSP0/CPU0:Router# configure
RP/0/RSP0/CPU0:Router(config)# controller dwdm 0/1/0/0
RP/0/RSP0/CPU0:Router(config-dwdm)# maintenance out-of-service
RP/0/RSP0/CPU0:Router(config-dwdm)# commit
RP/0/RSP0/CPU0:Router(config-dwdm)# g709 disable
RP/0/RSP0/CPU0:Router(config-dwdm)# loopback internal
RP/0/RSP0/CPU0:Router(config-dwdm)# g709 fec standard
RP/0/RSP0/CPU0:Router(config-dwdm)# g709 odu bdi disable
RP/0/RSP0/CPU0:Router(config-dwdm)# maintenance in-service
RP/0/RSP0/CPU0:Router(config-dwdm)# commit
```

DWDM のパフォーマンス モニタリング：例

次の例は、optics パラメータのパフォーマンス モニタリングを設定し、設定と現在の統計情報を表示する方法です。

```
RP/0/RSP0/CPU0:Router# configure
RP/0/RSP0/CPU0:Router(config)# controller dwdm 0/2/0/0

RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold opt max 2000000
RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold opt min 200
RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold lbc max 3000000
RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold lbc min 300
RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold opr max 4000000
RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min optics threshold opr min 400
RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min optics report opt max-tca enable
RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min optics report opt min-tca enable
RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min optics report opr max-tca enable
RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min optics report opr min-tca enable
RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min optics report lbc max-tca enable
RP/0/RSP0/CPU0:Router(config-dwdm)# pm 15-min optics report lbc min-tca enable
RP/0/RSP0/CPU0:Router(config-dwdm)# exit
RP/0/RSP0/CPU0:Router(config)# exit

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

LC/0/2/CPU0:Jul 12 04:10:47.252 : plim_4p_10ge_dwdm[194]: %L1-PMENGINE-4-TCA : Port DWDM
0/2/0/0 reports OPTICS TX-PWR-MIN(NE) PM TCA with current value 0, threshold 200 in
current 15-min interval window
LC/0/2/CPU0:Jul 12 04:10:47.255 : plim_4p_10ge_dwdm[194]: %L1-PMENGINE-4-TCA : Port DWDM
0/2/0/0 reports OPTICS RX-PWR-MIN(NE) PM TCA with current value 68, threshold 400 in
current 15-min interval window
```

```
RP/0/RP1/CPU0:Jul 12 04:09:05.443 : config[65678]: %MGBL-CONFIG-6-DB_COMMIT :
Configuration committed by user 'lab'. Use 'show configuration commit changes 1000000001'
to view the changes.
RP/0/RP1/CPU0:Jul 12 04:09:05.604 : config[65678]: %MGBL-SYS-5-CONFIG_I : Configured from
console by lab

RP/0/RSP0/CPU0:Router# show controllers dwdm 0/2/0/0 pm interval 15-min optics 0

Optics in the current interval [ 4:15:00 - 04:26:02 Wed Jul 12 2006]
             MIN      AVG      MAX  Threshold  TCA  Threshold  TCA
                                     (min)   (enable) (max)   (enable)
LBC[mA ] :  3605     4948     6453     300        YES      3000000   YES
OPT[uW]  :  2593     2593     2593     200        YES      2000000   YES
OPR[uW]  :  69       69       70       400        YES      4000000   YES
```

IPoDWDM 設定：例

ここでは、次の設定例について説明します。

- 「光レイヤ DWDM のポート設定：例」（P.503）
- 「DWDM 光ポートの管理状態設定：例」（P.503）
- 「予防的 FEC-FRR トリガーの設定：例」（P.504）

光レイヤ DWDM のポート設定：例

次に、光レイヤ DWDM ポートを設定する方法の例を示します。

```
RP/0/RSP0/CPU0:Router# configure
RP/0/RSP0/CPU0:Router(config)# controller dwdm 0/1/0/1
RP/0/RSP0/CPU0:Router(config-dwdm)# network port id 1/0/1/1
RP/0/RSP0/CPU0:Router(config-dwdm)# network connection id 1/1/1/1
```

DWDM 光ポートの管理状態設定：例

次に、管理状態を設定し、オプションでメンテナンス禁止フラグを設定する方法の例を示します。

```
RP/0/RSP0/CPU0:Router# configure
RP/0/RSP0/CPU0:Router(config)# controller dwdm 0/1/0/1
RP/0/RSP0/CPU0:Router(config-dwdm)# admin-state in-service
RP/0/RSP0/CPU0:Router(config-dwdm)# exit
RP/0/RSP0/CPU0:Router(config)# interface tengige 1/0/1/1
RP/0/RSP0/CPU0:Router(config-if)# maintenance disable
RP/0/RSP0/CPU0:Router(config-if)# commit
```

予防的 FEC-FRR トリガーの設定：例

次に、前方誤り訂正高速再ルーティング（FEC-FRR）の自動トリガーを設定する方法の例を示します。

```
RP/0/RSP0/CPU0:Router# configure
RP/0/RSP0/CPU0:Router(config)# controller dwdm 0/1/0/1
RP/0/RSP0/CPU0:Router(config-dwdm)# proactive
RP/0/RSP0/CPU0:Router(config-dwdm)# logging signal LogFile1
RP/0/RSP0/CPU0:Router(config-dwdm)# proactive trigger threshold 1 9
RP/0/RSP0/CPU0:Router(config-dwdm)# proactive trigger window 10000
RP/0/RSP0/CPU0:Router(config-dwdm)# proactive revert threshold 1 9
RP/0/RSP0/CPU0:Router(config-dwdm)# proactive revert window 600000
```

その他の関連資料

ここでは、DWDM コントローラ設定に関する参考資料について説明します。

関連資料

関連項目	参照先
Cisco IOS XR マスター コマンド リファレンス	*『Cisco IOS XR Master Commands List』*
Cisco IOS XR インターフェイス コンフィギュレーション コマンド	*『Cisco IOS XR Interface and Hardware Component Command Reference』*
Cisco IOS XR ソフトウェアを使用した初期システム ブートアップとルータの設定情報	*『Cisco IOS XR Getting Started Guide』*
Cisco IOS XR AAA サービス構成情報	『*Cisco IOS XR System Security Configuration Guide*』および *『Cisco IOS XR System Security Command Reference』*

標準

標準	タイトル
ITU-T G.709/Y.1331	『Interfaces for the optical transport network (OTN)』

MIB

MIB	MIB のリンク
—	Cisco IOS XR ソフトウェアを使用して選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml
OTN-MIB	IPoDWDM MIB

RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	—

シスコのテクニカル サポート

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

その他の関連資料

Cisco ASR 9000 シリーズ ルータでの POS インターフェイスの設定

ここでは、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータでの Packet-over-SONET/SDH（POS）インターフェイスの設定について説明します。

POS インターフェイスは、Cisco ハイレベル データリンク コントロール（HDLC）プロトコルまたはポイントツーポイント プロトコル（PPP）カプセル化を使用して、SONET フレームおよび同期デジタル ハイアラーキ（SDH）フレームを介した安全で信頼性の高いデータ伝送を実現します。Cisco HDLC および PPP カプセル化に加えて、Cisco ASR 9000 シリーズ ルータはフレームリレー カプセル化もサポートします。

レイヤ 1 の POS インターフェイスを設定するコマンドについては、『*Cisco IOS XR Interface and Hardware Component Command Reference*』を参照してください。

Cisco IOS XR ソフトウェアの POS インターフェイス設定機能の履歴

リリース	変更内容
リリース 4.0.0	この機能は、Cisco ASR 9000 シリーズ ルータで次の SPA に対して導入されました。 • Cisco 1 ポート チャネライズド OC-48/STM-16 SPA • Cisco 2 ポート チャネライズド OC-12c/DS0 SPA • Cisco 1 ポート OC-192c/STM-64 POS/RPR XFP SPA • Cisco 2 ポート OC-48c/STM-16 POS/RPR SPA • Cisco 8 ポート OC-12c/STM-4 POS SPA
リリース 4.0.1	次の SPA のサポートが Cisco ASR 9000 シリーズ ルータに追加されました。 • Cisco 4 ポート OC-3c/STM-1 POS SPA • Cisco 8 ポート OC-3c/STM-1 POS SPA

内容

- 「その他の関連資料」（P.536）

POS インターフェイスを設定するための前提事項

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。

POS インターフェイスを設定する前に、次の条件を満たしていることを確認してください。

- 新しい POS インターフェイス設定に割り当てるインターフェイスの IP アドレスを調べておく必要があります。
- 「Cisco ASR 9000 シリーズ ルータ でのクリア チャネル SONET コントローラの設定」または「Cisco ASR 9000 シリーズ ルータ でのチャネライズド SONET/SDH の設定」モジュールの説明に従ってクリア チャネルまたはチャネライズド SONET コントローラを設定してある。

POS インターフェイスの設定に関する情報

POS コントローラ インターフェイスを設定するには、次の概念を理解しておく必要があります。

- 「Cisco HDLC カプセル化」（P.509）
- 「PPP Encapsulation」（P.509）
- 「キープアライブ タイマー」（P.511）
- 「フレーム リレー カプセル化」（P.511）
- 「POS インターフェイスのデフォルト設定」（P.508）

Cisco ASR 9000 シリーズ ルータでは、1 つの POS インターフェイスにおいて、PPP カプセル化、Cisco HDLC カプセル化、またはフレーム リレー カプセル化を使用するデータが伝送されます。

ルータは、POS インターフェイス アドレスを識別するために、そのインターフェイスに関連付けられた物理層インターフェイス モジュール（PLIM）カードのラック番号、スロット番号、ベイ番号、およびポート番号を使用します。POS インターフェイス下にサブインターフェイスおよび相手先固定接続（PVC）が設定されている場合、ルータは POS インターフェイス パス ID にサブインターフェイス番号を含めます。

POS インターフェイスのデフォルト設定

POS インターフェイスが始動され、追加のコンフィギュレーション コマンドが適用されない場合は、表 13 に示すデフォルト インターフェイス設定が適用されます。これらのデフォルト設定はコンフィギュレーションで変更できます。

表 13　POS モジュラ サービス カードおよび PLIM のデフォルト インターフェイス設定

パラメータ	設定ファイルのエントリ	デフォルト設定
Keepalive （注）　**keepalive** コマンドは、HDLC または PPP カプセル化を使用する POS インターフェイスに適用されます。このコマンドはフレーム リレー カプセル化を使用する POS インターフェイスには適用されません。	**keepalive** {*interval* [*retry*] | **disable**} **no keepalive**	間隔：10 秒 再試行回数： • 5（PPP カプセル化を使用） • 3（HDLC カプセル化を使用）
カプセル化	**encapsulation** [**hdlc** | **ppp** | **frame-relay** [**IETF**]]	**hdlc**
最大伝送単位（MTU）	**mtu** *bytes*	4474 バイト
巡回冗長検査（CRC）	**crc** [**16** | **32**]	**32**

（注）　デフォルト設定は、**show running-config** コマンドの出力には含まれません。

Cisco HDLC カプセル化

Cisco ハイレベル データリンク コントローラ（HDLC）は、HDLC を使用して同期シリアル リンクでデータを送信するためのシスコ独自のプロトコルです。また、Cisco HDLC は、シリアル リンクのキープアライブを維持するシリアル ライン アドレス解決プロトコル（SLARP）と呼ばれる単純な制御プロトコルも提供します。HDLC は、Cisco IOS XR ソフトウェアにおける POS インターフェイスのデフォルト カプセル化タイプです。Cisco HDLC は、効率的なパケットの説明およびエラー制御を行う、オープン システム インターコネクション（OSI）スタックのレイヤ 2（データリンク）におけるデフォルトのデータ カプセル化のデフォルト プロトコルです。

（注）　Cisco HDLC は、POS インターフェイスにおいてデフォルトでイネーブルになります。

Cisco HDLC では、「キープアライブ タイマー」（P.511）で説明するように、キープアライブを使用してリンク ステートをモニタします。

PPP Encapsulation

PPP は、同期シリアル リンクでデータを送信するために使用される標準プロトコルです。また、PPP は、リンクのプロパティをネゴシエートするリンク コントロール プロトコル（LCP）も提供します。LCP は、エコー要求および応答を使用して、リンクの継続的なアベイラビリティをモニタリングします。

（注） インターフェイスに PPP カプセル化が設定されている場合、ECHOREQ パケットを送信し、ECHOREP 応答を受信しなかった回数が 3 回に達すると、リンク ダウンが宣言され、完全な LCP ネゴシエーションが再度開始されます。

PPP は、リンク上で動作するデータ プロトコルのプロパティをネゴシエーションするプロトコルとして、以下のネットワーク制御プロトコル（NCP）を提供します。

- IP コントロール プロトコル（IPCP）: IP プロパティのネゴシエーションを行います。
- マルチプロトコル ラベル スイッチング コントロール プロセッサ（MPLSCP）: MPLS プロパティのネゴシエーションを行います。
- Cisco Discovery Protocol コントロール プロセッサ（CDPCP）: CDP プロパティのネゴシエーションを行います。
- IPv6CP : IP Version 6（IPv6）プロパティのネゴシエーションを行います。
- 開放型システム間相互接続コントロール プロセッサ（OSICP）: OSI プロパティのネゴシエーションを行います。

PPP は、キープアライブを使用してリンク ステートをモニタリングします（「キープアライブ タイマー」（P.511）を参照）。

PPP は次の認証プロトコルをサポートします。これらのプロトコルでは、接続によるデータ トラフィックのフローを許可する前にそのアイデンティティを証明するために、リモート デバイスが必要です。

- チャレンジ ハンドシェイク認証プロトコル（CHAP）: CHAP は、リモート デバイスにチャレンジ メッセージを送信します。リモート デバイスは、共有秘密を使用してチャレンジの値を暗号化し、暗号化された値とその名前を応答メッセージでローカル ルータに戻します。ローカル ルータは、リモート デバイスの名前をローカル ユーザ名またはリモート セキュリティ サーバ データベース内に保存された関連秘密に一致させようとします。保存された秘密を使用して、元のチャレンジを暗号化し、暗号化された値が一致していることを確認します。
- マイクロソフト チャレンジ ハンドシェイク認証プロトコル（MS-CHAP）: MS-CHAP は CHAP の Microsoft バージョンです。CHAP の標準バージョンと同様に、MS-CHAP は PPP 認証に使用されます。この場合、認証は、Microsoft Windows NT または Microsoft Windows 95 を使用するパーソナル コンピュータとネットワーク アクセス サーバとして機能する Cisco ルータまたはアクセス サーバの間で行われます。
- パスワード認証プロトコル（PAP）: PAP 認証では、ローカル ユーザ名データベース内またはリモート セキュリティ サーバ データベース内の一致するエントリに照らし合わせてチェックする名前とパスワードを送信するために、リモート デバイスが必要です。

（注） PPP 認証プロトコルのイネーブル化および設定の詳細については、このマニュアルで後述する「Cisco ASR 9000 シリーズ ルータ での PPP の設定」モジュールを参照してください。

POS インターフェイスで CHAP、MS-CHAP、および PAP をイネーブルにするには、インターフェイス コンフィギュレーション モードで **ppp authentication** コマンドを使用します。

（注） PPP 認証をイネーブル化またはディセーブル化しても、ローカル ルータがリモート デバイスに対して自身を認証しようとすることには変わりありません。

キープアライブ タイマー

シスコ キープアライブは、リンク ステートをモニタリングする場合に便利です。キープアライブは、キープアライブ タイマーの値によって決定される頻度で、定期的にピアに送信され、ピアから受信されます。受け入れ可能なキープアライブがピアから受信されない場合、リンクはダウン状態に移行します。ピアから受け入れ可能なキープアライブが受信されるか、キープアライブがディセーブルになると、リンクはすぐにアップ状態に移行します。

ピアにキープアライブを送信し、応答が得られなかった回数が 3 回に達すると、リンクはダウン状態に移行します。ECHOREQ パケットは、LCP ネゴシエーションが完了した場合（LCP が開いている場合など）に限り、送信されます。

（注） **keepalive** コマンドは、HDLC または PPP カプセル化を使用する POS インターフェイスに適用されます。このコマンドはフレームリレー カプセル化を使用する POS インターフェイスには適用されません。

LCP が ECHOREQ パケットをピアに送信する頻度を設定するには、インターフェイス コンフィギュレーション モードで **keepalive** コマンドを使用します。システムを 10 秒のデフォルト キープアライブ インターバルに戻すには、**keepalive** コマンドを **no** 引数とともに使用します。キープアライブをディセーブルにするには、**keepalive disable** コマンドを使用します。PPP と Cisco HDLC では、0 のキープアライブはキープアライブをディセーブルにし、**show running-config** コマンド出力では、**keepalive disable** として報告されます。

keepalive コマンドをコンフィギュレーションから完全に削除するには、**no keepalive** コマンドを使用します。**keepalive** コマンドをインターフェイス コンフィギュレーションから削除してからでなければ、そのインターフェイスでフレーム リレー カプセル化を設定することはできません。フレーム リレー インターフェイスは、キープアライブをサポートしません。

（注） MDR 中は、キープアライブ インターバルが 10 秒以上であることが必要です。

LCP がピアで動作していて、ECHOREQ パケットを受信すると、キープアライブがピアでイネーブルかどうかに関係なく、エコー応答（ECHOREP）パケットで応答します。

キープアライブは、2 つのピアの間で独立しています。一方のピアでキープアライブをイネーブルに設定し、もう一方でディセーブルに設定することもできます。キープアライブがローカルでディセーブルの場合でも、LCP は受信する ECHOREQ パケットに ECHOREP パケットで応答します。同様に、LCP は、それぞれの端のキープアライブの期間が異なる場合でも機能します。

（注） キープアライブ タイマーを設定した後で、ピアに送信される SLARP パケットの情報を表示するには、**debug chdlc slarp packet** コマンドと他の Cisco HDLC **debug** コマンドを使用します。

フレーム リレー カプセル化

Cisco ASR 9000 シリーズ ルータでは、フレーム リレー カプセル化を使用する POS インターフェイスの設定は階層形式となり、次の要素で構成されます。

1. POS メイン インターフェイスは物理インターフェイスとポートで構成されます。POS インターフェイスが Cisco HDLC カプセル化および PPP カプセル化を使用する接続をサポートしていない場合は、POS メイン インターフェイス下に PVC を持つサブインターフェイスを設定する必要があります。フレーム リレー接続は、PVC でのみサポートされます。

2. POS サブインターフェイスは POS メイン インターフェイス下に設定されます。POS サブインターフェイスは、その下に PVC を設定しなければトラフィックをアクティブに伝送しません。
3. ポイントツーポイントおよびレイヤ 2 接続回線（AC）の PVC は、POS サブインターフェイスの下で設定されます。メイン インターフェイスの下に PVC を直接設定できません。1 つのサブインターフェイスにつき、許可されるポイントツーポイントまたは L2 AC PVC は 1 つだけです。PVC はあらかじめ定義された回線パスを使用し、パスが中断されるとエラーが発生します。PVC は、回線が削除されるまでアクティブのままです。POS PVC 上の接続はフレームリレー カプセル化だけをサポートします。
4. レイヤ 3 の設定は、一般的にサブインターフェイス上で行われます。

（注） 親インターフェイスの管理状態は、サブインターフェイスとその PVC の状態を決定します。親インターフェイスまたはサブインターフェイスの管理状態が変わると、その親インターフェイスまたはサブインターフェイスの下に設定されたすべての子 PVC の管理状態も変わります。

Cisco ASR 9000 シリーズ ルータでは、次の SPA がフレーム リレー カプセル化をサポートします。

- Cisco 4 ポート OC-3c/STM-1 POS SPA
- Cisco 8 ポート OC-3c/STM-1 POS SPA
- Cisco 1 ポート OC-192c/STM-64 POS/RPR XFP SPA
- Cisco 2 ポート OC-48c/STM-16 POS/RPR SPA
- Cisco 8 ポート OC-12c/STM-4 POS SPA

POS インターフェイスでフレームリレー カプセル化を設定するには、**encapsulation frame-relay** コマンドを使用します。

フレーム リレー インターフェイスは、次の 2 つのタイプのカプセル化フレームをサポートします。

- Cisco（これがデフォルト値です）
- IETF

PVC に Cisco または IETF カプセル化を設定するには、PVC コンフィギュレーション モードで **encap** コマンドを使用します。PVC のカプセル化タイプを明示的に設定しない場合、その PVC はメイン POS インターフェイスのカプセル化タイプを継承します。

（注） MPLS に設定された POS メイン インターフェイスには、Cisco カプセル化を設定する必要があります。IETF カプセル化は、MPLS ではサポートされていません。

インターフェイスにフレーム リレーのカプセル化を設定する前に、そのインターフェイスから以前のレイヤ 3 のすべての設定が除去されていることを確認する必要があります。たとえば、メイン インターフェイスの下に直接設定されている IP アドレスがないことを確認する必要があります。IP アドレスが直接設定されていると、メイン インターフェイスの下で行われたフレーム リレー設定が実行できなくなります。

フレーム リレー インターフェイスでの LMI

ローカル管理インターフェイス（LMI）プロトコルは、PVC の追加、削除、およびステータスをモニタリングします。また、LMI は、フレーム リレー UNI インターフェイスを形成するリンクの完全性を確認します。デフォルトでは、**cisco** LMI はすべての PVC でイネーブルです。ただし、このマニュアルで後述する「インターフェイスでのデフォルト フレームリレー設定の変更」モジュールで説明するように、LMI タイプを ANSI または Q.933 に変更できます。

LMI のタイプが **cisco** （デフォルトの LMI タイプ）である場合、1 つのインターフェイスでサポートできる PVC の最大数は、メイン インターフェイスの MTU サイズに関連しています。カードまたは SPA でサポートされる PVC の最大数を計算するには、次の公式を使用します。

```
(MTU - 13)/8 = PVC の最大数
```

（注） POS インターフェイスの場合、**mtu** コマンドのデフォルト設定は 4474 バイトです。したがって、**cisco** LMI で設定された 1 つの POS インターフェイスでサポートされる PVC のデフォルトの最大数は 557 です。

（注） フレームリレー インターフェイスには LMI インターフェイス タイプを設定する必要があります。そうしなければ、POS インターフェイスはアップ状態になりません。プロバイダー エッジ（PE）ルータとカスタマー エッジ（CE）ルータとの接続では、LMI がアップ状態になるためには、PE 側が DCE であり、CE 側が DTE であることが必要です。フレーム リレー インターフェイスに対する LMI インターフェイス タイプの設定の詳細については、「Cisco ASR 9000 シリーズ ルータでのフレーム リレーの設定」モジュールを参照してください。

POS 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。

POS インターフェイスの始動

ここでは、POS インターフェイスの始動に使用するコマンドについて説明します。

前提条件

Cisco IOS XR ソフトウェアを実行するルータに POS ラインカードまたは SPA が取り付けられている必要があります。

制約事項

POS インターフェイスがアクティブになるためには、POS 接続の両端の設定が一致している必要があります。

手順の概要

1. **show interfaces**
2. **configure**
3. **interface pos** *interface-path-id*
4. **ipv4 address** *ipv4_address*/*prefix*
5. **no shutdown**
6. **end**
   または
   commit
7. **exit**
8. **exit**
9. 接続の他端でインターフェイスを始動するために、ステップ 1 ～ 8 を繰り返します。
10. **show ipv4 interface brief**
11. **show interfaces pos** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**show interfaces** **例：** `RP/0/RSP0/CPU0:router# show interfaces`	（任意）設定されているインターフェイスを表示します。 • このコマンドを使用して、ルータが PLIM カードを認識しているかどうかも確認します。
ステップ 2	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface pos** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface POS 0/3/0/0`	POS インターフェイス名と *rack/slot/module/port* 表記を指定して、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。
ステップ 4	**ipv4 address** *ipv4_address/prefix* **例：** `RP/0/RSP0/CPU0:router (config)#ipv4 address 10.46.8.6/24`	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。 **（注）** このインターフェイスにフレームリレー カプセル化を設定する場合は、このステップを省略してください。フレームリレーの場合、IP アドレスとサブネット マスクはサブインターフェイスに設定します。

	コマンドまたはアクション	目的
ステップ 5	**no shutdown** **例：** `RP/0/RSP0/CPU0:router (config-if)# no shutdown`	shutdown 設定を削除します。 **（注）** shutdown 設定を削除することにより、インターフェイスでの強制的な管理上の停止が排除されるため、インターフェイスはアップ状態またはダウン状態に移行することができます（親 SONET レイヤが管理上の停止状態に設定されていないことを前提とします）。
ステップ 6	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router (config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 7	**exit** **例：** `RP/0/RSP0/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、グローバル コンフィギュレーション モードに入ります。
ステップ 8	**exit** **例：** `RP/0/RSP0/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、EXEC 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 9	**show interfaces** **configure** **interface pos** *interface-path-id* **no shut** **exit** **exit** **commit** **例：** RP/0/RSP0/CPU0:router# show interfaces RP/0/RSP0/CPU0:router# configure RP/0/RSP0/CPU0:router (config)# interface pos 0/3/0/0 RP/0/RSP0/CPU0:router (config-if)# no shutdown RP/0/RSP0/CPU0:router (config-if)# commit RP/0/RSP0/CPU0:router (config-if)# exit RP/0/RSP0/CPU0:router (config)# exit	接続の他端でインターフェイスを始動するために、ステップ 1 ～ 8 を繰り返します。 **（注）** POS 接続の両端で設定が一致している必要があります。
ステップ 10	**show ipv4 interface brief** **例：** RP/0/RSP0/CPU0:router # show ipv4 interface brief	インターフェイスがアクティブであり、適切に設定されていることを確認します。 POS インターフェイスが適切に始動されていると、**show ipv4 interface brief** コマンドの出力結果で、そのインターフェイスの [Status] フィールドに「Up」と表示されます。
ステップ 11	**show interfaces pos** *interface-path-id* **例：** RP/0/RSP0/CPU0:router# show interfaces pos 0/3/0/0	（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。

次の作業

始動した POS インターフェイスのデフォルト設定を変更するには、「オプションの POS インターフェイス パラメータの設定」（P.516）を参照してください。

オプションの POS インターフェイス パラメータの設定

ここでは、POS インターフェイスのデフォルト設定の変更に使用できるコマンドについて説明します。

前提条件

POS インターフェイスのデフォルト設定を変更する前に、POS インターフェイスを始動して、「POS インターフェイスの始動」（P.513）で説明するように shutdown 設定を削除することをお勧めします。

制約事項

POS インターフェイスがアクティブになるためには、POS 接続の両端の設定が一致している必要があります。

手順の概要

1. **configure**
2. **interface pos** *interface-path-id*
3. **encapsulation** [**hdlc** | **ppp** | **frame-relay** [**IETF**]]
4. **pos crc** {**16** | **32**}
5. **mtu** *value*
6. **end**
   または
   commit
7. **exit**
8. **exit**
9. **show interfaces pos** [*interface-path-id*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface pos** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface POS 0/3/0/0`	POS インターフェイス名と *rack/slot/module/port* 表記を指定して、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。
ステップ 3	**encapsulation** [**hdlc** | **ppp** | **frame-relay** [**IETF**]] **例：** `RP/0/RSP0/CPU0:router(config-if)# encapsulation hdlc`	（任意）インターフェイス カプセル化パラメータおよび HDLC やポイントツーポイント プロトコル（PPP）などの詳細を設定します。 **（注）** デフォルトのカプセル化は **hdlc** です。
ステップ 4	**pos crc** {**16** | **32**} **例：** `RP/0/RSP0/CPU0:router(config-if)# pos crc 32`	（任意）インターフェイスの CRC 値を設定します。16 ビットの CRC モードを指定するには **16** キーワード、**32** ビットの CRC モードを指定するには 32 キーワードを入力します。 **（注）** デフォルト CRC は **32** です。
ステップ 5	**mtu** *value* **例：** `RP/0/RSP0/CPU0:router(config-if)# mtu 4474`	（任意）MTU 値を設定します。 • デフォルト値は 4474 です。 • POS MTU の範囲は 64 ～ 9216 です。

	コマンドまたはアクション	目的
ステップ 6	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router (config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 7	**exit** **例：** `RP/0/RSP0/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、グローバル コンフィギュレーション モードに入ります。
ステップ 8	**exit** **例：** `RP/0/RSP0/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、EXEC モードを開始します。
ステップ 9	**show interfaces pos** [*interface-path-id*] **例：** `RP/0/RSP0/CPU0:router# show interface pos 0/3/0/0`	（任意）指定した POS インターフェイスの一般情報を表示します。

次の作業

- 始動した POS インターフェイス上に PVC を持つポイントツーポイント フレームリレー サブインターフェイスを作成するには、「PVC を持つポイントツーポイント POS サブインターフェイスの作成」（P.519）を参照してください。
- PPP カプセル化がイネーブルである POS インターフェイスに PPP 認証を設定するには、このマニュアルで後述する *「Cisco ASR 9000 シリーズ ルータ での PPP の設定」* モジュールを参照してください。
- Cisco HDLC カプセル化または PPP カプセル化がイネーブルである POS インターフェイスのキープアライブ インターバルを変更するには、「POS インターフェイスでのキープアライブ インターバルの変更」（P.524）を参照してください。

- フレームリレー カプセル化がイネーブルである POS インターフェイスのデフォルトのフレームリレー設定を変更するには、このマニュアルの*「Cisco ASR 9000 シリーズ ルータでのフレーム リレーの設定」*の「インターフェイスでのデフォルト フレームリレー設定の変更」モジュールを参照してください。

PVC を持つポイントツーポイント POS サブインターフェイスの作成

ここに記載する手順では、ポイントツーポイント POS サブインターフェイスを作成し、その POS サブインターフェイスに相手先固定接続（PVC）を設定します。

（注） サブインターフェイスおよび PVC の作成は、フレームリレー カプセル化だけが設定されたインターフェイスでサポートされます。

前提条件

POS インターフェイスでサブインターフェイスを作成する前に、「POS インターフェイスの始動」（P.513）で説明するように、フレームリレー カプセル化が設定されたメイン POS インターフェイスを始動する必要があります。

制約事項

PVC は、各ポイントツーポイント POS サブインターフェイスに 1 つだけ設定できます。

手順の概要

1. **configure**
2. **interface pos** *interface-path-id.subinterface* **point-to-point**
3. **ipv4 address** *ipv4_address/prefix*
4. **pvc** *dlci*
5. **end**
   または
   commit
6. 接続の他端で POS サブインターフェイスおよび関連付けられている PVC を始動するために、ステップ 1 ～ 5 を繰り返します。

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface pos** *interface-path-id.subinterface* **point-to-point** **例：** `RP/0/RSP0/CPU0:router (config)# interface pos 0/3/0/0.1 point-to-point`	POS 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。 *subinterface* は、1 から 4294967295 の範囲のサブインターフェイス ID に置き換えてください。
ステップ 3	**ipv4 address** *ipv4_address/prefix* **例：** `RP/0/RSP0/CPU0:router (config-subif)#ipv4 address 10.46.8.6/24`	IP アドレスおよびサブネット マスクをサブインターフェイスに割り当てます。
ステップ 4	**pvc** *dlci* **例：** `RP/0/RSP0/CPU0:router (config-subif)# pvc 20`	POS 相手先固定接続（PVC）を作成し、フレームリレー PVC コンフィギュレーション サブモードを開始します。 *dlci* を 16 から 1007 の範囲の PVC ID に置き換えます。 **（注）** 各サブインターフェイスに設定できる PVC は 1 つだけです。

	コマンドまたはアクション	目的
ステップ 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router (config-fr-vc)# end` または `RP/0/RSP0/CPU0:router(config-fr-vc)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 6	**configure** **interface pos** *interface-path-id.subinterface* **pvc** *dlci* **commit** **例：** `RP/0/RSP0/CPU0:router# configure` `RP/0/RSP0/CPU0:router (config)# interface pos 0/3/0/1.1` `RP/0/RSP0/CPU0:router (config-subif)#ipv4 address 10.46.8.5/24` `RP/0/RSP0/CPU0:router (config-subif)# pvc 20` `RP/0/RSP0/CPU0:router (config-fr-vc)# commit`	接続の他端で POS サブインターフェイスおよび関連付けられている PVC を始動するために、ステップ 1 ～ 5 を繰り返します。 **（注）** DLCI（PVC ID）は、サブインターフェイス接続の両端で一致している必要があります。 **（注）** 接続の他端のサブインターフェイスに IP アドレスおよびサブネット マスクを割り当てるときには、接続の両端のアドレスが同じサブネットに属している必要があることに注意してください。

次の作業

- オプションの PVC パラメータを設定するには、「オプションの PVC パラメータの設定」（P.521）を参照してください。
- フレーム リレー カプセル化がイネーブルである POS インターフェイスのデフォルトのフレーム リレー設定を変更するには、*「Cisco ASR 9000 シリーズ ルータでのフレーム リレーの設定」*モジュールの「インターフェイスでのデフォルト フレームリレー設定の変更」を参照してください。
- レイヤ 3 QOS サービス ポリシーを PVC サブモードの PVC に付加するには、該当する Cisco IOS XR ソフトウェアのコンフィギュレーション ガイドを参照してください。

オプションの PVC パラメータの設定

ここでは、POS PVC でのデフォルト設定の変更に使用できるコマンドについて説明します。

前提条件

PVC のデフォルト設定を変更する前に、「PVC を持つポイントツーポイント POS サブインターフェイスの作成」（P.519）で説明するように POS サブインターフェイスで PVC を作成する必要があります。

制約事項

- 接続がアクティブになるためには、DLCI（PVC ID）が PVC の両端で一致している必要があります。
- PVC DLCI を変更するには、PVC を削除し、新しい DLCI を設定して PVC を追加し直す必要があります。

手順の概要

1. **configure**
2. **interface pos** *interface-path-id.subinterface*
3. **pvc** *dlci*
4. **encap** [**cisco** | **ietf**]
5. **service-policy** {**input** | **output**} *policy-map*
6. **end**
   または
   commit
7. 接続の他端で PVC を設定するために、ステップ 1 ～ 6 を繰り返します。
8. **show frame-relay pvc** *dlci-number*
9. **show policy-map interface pos** *interface-path-id.subinterface* {**input** | **output**}
   または
   show policy-map type qos interface pos *interface-path-id.subinterface* {**input** | **output**}

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface pos** *interface-path-id.subinterface* **例：** `RP/0/RSP0/CPU0:router (config)# interface pos 0/3/0/0.1`	POS 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	**pvc** *dlci* **例：** `RP/0/RSP0/CPU0:router (config-subif)# pvc 20`	PVC に対する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。 *dlci* は、PVC の識別に使用される DLCI 番号に置き換えてください。有効値の範囲は 16 ～ 1007 です。

	コマンドまたはアクション	目的
ステップ 4	**encap** [**cisco** | **ietf**] **例：** `RP/0/RSP0/CPU0:router (config-fr-vc)# encap ietf`	（任意）フレームリレー PVC のカプセル化を設定します。 **（注）** PVC のカプセル化タイプを明示的に設定しない場合、その PVC はメイン POS インターフェイスのカプセル化タイプを継承します。
ステップ 5	**service-policy** {**input** | **output**} *policy-map* **例：** `RP/0/RSP0/CPU0:router (config-fr-vc)# service-policy output policy1`	ポリシー マップを入力サブインターフェイスまたは出力サブインターフェイスに付加します。付加すると、そのサブインターフェイスのサービス ポリシーとしてポリシー マップが使用されます。 **（注）** ポリシー マップの作成と設定については、『*Cisco IOS XR Modular Quality of Service Configuration Guide*』を参照してください。
ステップ 6	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router (config-fr-vc)# end` または `RP/0/RSP0/CPU0:router(config-fr-vc)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 7	**configure** **interface pos** *interface-path-id.subinterface* **pvc** *dlci* **encap** [**cisco** | **ietf**] **commit** **例：** `RP/0/RSP0/CPU0:router# configure` `RP/0/RSP0/CPU0:router (config)# interface pos 0/3/0/1.1` `RP/0/RSP0/CPU0:router (config-subif)# pvc 20` `RP/0/RSP0/CPU0:router (config-fr-vc)# encap cisco` `RP/0/RSP0/CPU0:router (config-fr-vc)# commit`	接続の他端で POS サブインターフェイスおよび関連付けられている PVC を始動するために、ステップ 1 ～ 6 を繰り返します。 **（注）** サブインターフェイス接続の両端で設定が一致している必要があります。

	コマンドまたはアクション	目的
ステップ 8	**show frame-relay pvc** *dlci-number* **例：** `RP/0/RSP0/CPU0:router# show frame-relay pvc 20`	（任意）指定した POS インターフェイスの設定を検証します。
ステップ 9	**show policy-map interface pos** *interface-path-id.subinterface* {**input** | **output**} または **show policy-map type qos interface pos** *interface-path-id.subinterface* {**input** | **output**} **例：** `RP/0/RSP0/CPU0:router# show policy-map interface pos 0/3/0/0.1 output` または `RP/0/RSP0/CPU0:router# show policy-map type qos interface pos 0/3/0/0.1 output`	（任意）サブインターフェイスに付加された入力ポリシーおよび出力ポリシーの統計情報と設定を表示します。

次の作業

フレーム リレー カプセル化がイネーブルである POS インターフェイスのデフォルトのフレーム リレー設定を変更するには、*「Cisco ASR 9000 シリーズ ルータでのフレーム リレーの設定」*モジュールの「インターフェイスでのデフォルト フレームリレー設定の変更」を参照してください。

POS インターフェイスでのキープアライブ インターバルの変更

Cisco HDLC カプセル化または PPP カプセル化がイネーブルである POS インターフェイスのキープアライブ インターバルを変更するには、次の作業を行います。

（注） POS インターフェイスで Cisco HDLC カプセル化または PPP カプセル化をイネーブルした場合、キープアライブ インターバルはデフォルトで 10 秒に設定されます。デフォルトのキープアライブ インターバルを変更する手順は、次のとおりです。

（注） Cisco HDLC は、POS インターフェイスにおいてデフォルトでイネーブルになります。

前提条件

キープアライブ タイマーの設定を変更する前に、インターフェイスで Cisco HDLC カプセル化または PPP カプセル化がイネーブルになっていることを確認する必要があります。インターフェイスで Cisco HDLC カプセル化または PPP カプセル化をイネーブルにするには、「オプションの POS インターフェイス パラメータの設定」（P.516）で説明するように **encapsulation** コマンドを使用します。

制約事項

MDR 中は、キープアライブ インターバルが 10 秒以上であることが必要です。

手順の概要

1. **configure**
2. **interface pos** *interface-path-id*
3. **keepalive** {*seconds* [*retry-count*] | **disable**}
   または
   no keepalive
4. **end**
   または
   commit
5. **show interfaces** *type interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface pos** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface POS 0/3/0/0`	POS インターフェイス名と *rack/slot/module/port* 表記を指定して、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。
ステップ 3	**keepalive** {*seconds* [*retry-count*]	**disable**} または **no keepalive** **例：** `RP/0/RSP0/CPU0:router(config-if)# keepalive 3` または `RP/0/RSP0/CPU0:router(config-if)# no keepalive`	キープアライブ メッセージの間隔を秒数で指定します。また、リンクをダウン状態に遷移する前に、応答なしでピアに送信できるキープアライブ メッセージの数をオプションで指定します。 • **keepalive disable** コマンド、**no keepalive**、または **keepalive** コマンドに引数 0 を付けたものを使用すると、キープアライブ機能が完全にディセーブルになります。 • キープアライブがインターフェイスで設定されている場合は、そのインターフェイスでフレーム リレー カプセル化を設定する前に、**no keepalive** コマンドを使用してキープアライブ機能をディセーブルにします。

	コマンドまたはアクション	目的
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	**show interfaces pos** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router# show interfaces POS 0/3/0/0`	（任意）インターフェイスの設定を確認します。

レイヤ 2 接続回線（AC）の設定方法

レイヤ 2 接続回路（AC）の設定作業について、次の手順で説明します。

- PVC を持つレイヤ 2 フレームリレー サブインターフェイスの作成
- オプションのレイヤ 2 PVC パラメータの設定

（注） レイヤ 2 スイッチングのためのインターフェイスの設定後は、**ipv4 address** などのルーティング コマンドは使用できません。

（注） レイヤ 2 AC は、HDLC カプセル化または PPP カプセル化が設定されたインターフェイスではサポートされません。

PVC を持つレイヤ 2 フレームリレー サブインターフェイスの作成

ここに記載する手順では、PVC を持つレイヤ 2 フレームリレー サブインターフェイスを作成します。

前提条件

POS インターフェイスでサブインターフェイスを作成する前に、「POS インターフェイスの始動」（P.513）で説明するように POS インターフェイスを始動する必要があります。

（注） インターフェイスをレイヤ 2 スイッチング用に設定する場合は、「POS インターフェイスの始動」設定手順のステップ 4 を省略してください。**ipv4 address** コマンドは、フレームリレー カプセル化が設定されたインターフェイスでは使用できません。

制約事項

- 各サブインターフェイスで設定できる PVC は 1 つだけです。
- 接続が正しく動作するためには、PVC の両端で設定が一致している必要があります。
- **ipv4 address** コマンドは、フレームリレー カプセル化が設定されたインターフェイスでは使用できません。インターフェイスをレイヤ 2 トランスポート モード用に設定する前に、IP アドレスの以前の設定を削除する必要があります。
- レイヤ 2 設定は、フレームリレー PVC だけでサポートされます。レイヤ 2 設定が直接メイン POS インターフェイスに適用されるレイヤ 2 ポート モードはサポートされていません。

手順の概要

1. **configure**
2. **interface pos** *interface-path-id.subinterface* **l2transport**
3. **pvc** *dlci*
4. **end**
   または
   commit
5. AC の他端でサブインターフェイスおよび関連付けられている PVC を始動するために、ステップ 1 ～ 4 を繰り返します。

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface pos** *interface-path-id.subinterface* **l2transport** **例：** `RP/0/RSP0/CPU0:router(config)# interface pos 0/3/0/0.1 l2transport`	サブインターフェイスを作成して、そのサブインターフェイスに対する POS 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。 **（注）** *subinterface* は、1 つのメイン インターフェイスに設定された他のサブインターフェイスに対して一意である必要があります。
ステップ 3	**pvc** dlci **例：** `RP/0/RSP0/CPU0:router(config-if)# pvc 100`	フレームリレー相手先固定接続（PVC）を作成して、レイヤ 2 転送 PVC コンフィギュレーション モードを開始します。 *dlci* は、PVC の識別に使用される DLCI 番号に置き換えてください。有効値の範囲は 16 ～ 1007 です。 **（注）** 各サブインターフェイスに設定できる PVC は 1 つだけです。
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-fr-vc)# end` または `RP/0/RSP0/CPU0:router(config-fr-vc)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	AC の他端でサブインターフェイスおよび関連付けられている PVC を始動するために、ステップ 1 ～ 4 を繰り返します。	AC を始動します。 **（注）** AC の両端で設定が一致している必要があります。

次の作業

- オプションのサブインターフェイス パラメータを設定するには、「オプションのレイヤ 2 サブインターフェイス パラメータの設定」（P.531）を参照してください。
- オプションの PVC パラメータを設定するには、「オプションのレイヤ 2 PVC パラメータの設定」（P.529）を参照してください。
- Cisco ASR 9000 シリーズ ルータのレイヤ 2 サービスの設定の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide*』の「Implementing Point to Point Layer 2 Services」モジュールを参照してください。

オプションのレイヤ 2 PVC パラメータの設定

ここでは、フレームリレー レイヤ 2 PVC でのデフォルト設定の変更に使用できるコマンドについて説明します。

前提条件

「PVC を持つレイヤ 2 フレームリレー サブインターフェイスの作成」（P.527）で説明するように、レイヤ 2 サブインターフェイスで PVC を作成する必要があります。

手順の概要

1. **configure**
2. **interface pos** *interface-path-id.subinterface* **l2transport**
3. **pvc** *dlci*
4. **encap** [**cisco** | **ietf**]
5. **service-policy** {**input** | **output**} *policy-map*
6. **end**
   または
   commit
7. AC の他端で PVC を設定するために、ステップ 1 ～ 5 を繰り返します。
8. **show policy-map interface pos** *interface-path-id.subinterface* {**input** | **output**}
   または
   show policy-map type qos interface pos *interface-path-id.subinterface* {**input** | **output**}

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface pos** *interface-path-id.subinterface* **l2transport** **例：** `RP/0/RSP0/CPU0:router(config)# interface pos 0/6/0/1.10 l2transport`	レイヤ 2 フレームリレー サブインターフェイスに対する POS 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	**pvc** *dlci* **例：** `RP/0/RSP0/CPU0:router(config-if)# pvc 100`	指定した PVC に対するフレームリレー PVC コンフィギュレーション モードを開始します。 *dlci* は、PVC の識別に使用される DLCI 番号に置き換えてください。有効値の範囲は 16 ～ 1007 です。
ステップ 4	**encap** {**cisco** | **ietf**} **例：** `RP/0/RSP0/CPU0:router(config-fr-vc)# encap ietf`	フレームリレー PVC のカプセル化を設定します。 PVC の両端でカプセル化タイプが一致している必要があります。
ステップ 5	**service-policy** {**input** | **output**} *policy-map* **例：** `RP/0/RSP0/CPU0:router (config-fr-vc)# service-policy output policy1`	ポリシー マップを入力サブインターフェイスまたは出力サブインターフェイスに付加します。付加すると、そのサブインターフェイスのサービス ポリシーとしてポリシー マップが使用されます。 **（注）** ポリシー マップの作成と設定については、『*Cisco IOS XR Modular Quality of Service Configuration Guide*』を参照してください。

	コマンドまたはアクション	目的
ステップ 6	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-pos-l2transport-pvc)# end` または `RP/0/RSP0/CPU0:router(config-pos-l2transport-pvc)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 7	AC の他端で PVC を設定するために、ステップ 1 ～ 5 を繰り返します。	AC を始動します。 **（注）** 接続の両端で設定が一致している必要があります。
ステップ 8	**show policy-map interface pos** *interface-path-id.subinterface* {**input** | **output**} または **show policy-map type qos interface pos** *interface-path-id.subinterface* {**input** | **output**} **例：** `RP/0/RSP0/CPU0:router# show policy-map interface pos 0/6/0/1.10 output` または `RP/0/RSP0/CPU0:router# show policy-map type qos interface pos 0/6/0/1.10 output`	（任意）サブインターフェイスに付加された入力ポリシーおよび出力ポリシーの統計情報と設定を表示します。

オプションのレイヤ 2 サブインターフェイス パラメータの設定

ここでは、フレームリレー レイヤ 2 サブインターフェイスでのデフォルト設定の変更に使用できるコマンドについて説明します。

前提条件

PVC のデフォルト設定を変更する前に、「PVC を持つレイヤ 2 フレームリレー サブインターフェイスの作成」（P.527）で説明するようにレイヤ 2 サブインターフェイスで PVC を作成する必要があります。

制約事項

ほとんどの場合、サブインターフェイスに設定された MTU がメイン インターフェイスに設定された MTU より優先されます。このルールの例外は、サブインターフェイスの MTU がメイン インターフェイスの MTU より大きい場合です。その場合、CLI 出力にはサブインターフェイスの MTU の設定値が表示されますが、実際に有効となる MTU はメイン インターフェイスに設定された値です。レイヤ 2 接続のトラブルシューティングや最適化において混乱を避けるために、メイン インターフェイスに設定する MTU の方を大きくすることをお勧めします。

手順の概要

1. **configure**
2. **interface pos** *interface-path-id.subinterface*
3. **mtu** *value*
4. **end**
   または
   commit
5. AC の他端でサブインターフェイスを設定するために、ステップ 1 ～ 4 を繰り返します。

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface pos** *interface-path-id.subinterface* **例：** `RP/0/RSP0/CPU0:router(config)# interface pos 0/3/0/1.1`	レイヤ 2 フレームリレー サブインターフェイスに対する POS 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	**mtu** *value* **例：** `RP/0/RSP0/CPU0:router(config-if)# mtu 5000`	（任意）MTU 値を設定します。有効値の範囲は 64 ～ 65535 です。

	コマンドまたはアクション	目的
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-pos-l2transport-pvc)# end` または `RP/0/RSP0/CPU0:router(config-pos-l2transport-pvc)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	AC の他端で PVC を設定するために、手順 1 ～ 4 を繰り返します。	AC を始動します。 **（注）** 接続の両端で設定が一致している必要があります。

POS インターフェイスの設定例

ここでは、次の設定例について説明します。

- 「POS インターフェイスの始動と Cisco HDLC カプセル化の設定：例」（P.533）
- 「POS インターフェイスでのフレームリレー カプセル化の設定：例」（P.534）
- 「POS インターフェイスでの PPP カプセル化の設定：例」（P.535）

POS インターフェイスの始動と Cisco HDLC カプセル化の設定：例

次に、Cisco HDLC カプセル化を設定した基本的な POS インターフェイスの始動例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RSP0/CPU0:router(config-if)# no shutdown
RP/0/RSP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

次に、キープアライブ メッセージの間隔を 10 秒に設定する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/RSP0/CPU0:router(config-if)# keepalive 10
RP/0/RSP0/CPU0:router(config-if)# commit
```

POS インターフェイスでのフレームリレー カプセル化の設定：例

次に、ルータ 1 でフレームリレー カプセル化が設定された POS インターフェイスと PVC を持つポイントツーポイント POS サブインターフェイスを作成する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/RSP0/CPU0:router(config-if)# encapsulation frame-relay
RP/0/RSP0/CPU0:router(config-if)# no shutdown
RP/0/RSP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them?[yes]: yes

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router (config)# interface pos 0/3/0/0.1 point-to-point
RP/0/RSP0/CPU0:router (config-subif)#ipv4 address 10.20.3.1/24
RP/0/RSP0/CPU0:router (config-subif)# pvc 100
RP/0/RSP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them?[yes]: yes

RP/0/RSP0/CPU0:router# show interface POS 0/3/0/0

Wed Oct  8 04:20:30.248 PST DST
POS0/3/0/0 is up, line protocol is up
  Interface state transitions: 1
  Hardware is Packet over SONET/SDH
  Internet address is 10.20.3.1/24
  MTU 4474 bytes, BW 155520 Kbit
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation FRAME-RELAY, crc 32, controller loopback not set,
  LMI enq sent  116, LMI stat recvd 76, LMI upd recvd 0, DTE LMI up
  LMI enq recvd 0, LMI stat sent  0, LMI upd sent  0
  LMI DLCI 1023  LMI type is CISCO  frame relay DTE
  Last clearing of "show interface" counters 00:00:06
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     1 packets input, 13 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     1 packets output, 13 bytes, 0 total output drops
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
```

次に、ルータ 1 に接続されたルータ 2 でフレームリレー カプセル化が設定された POS インターフェイスと PVC を持つポイントツーポイント POS サブインターフェイスを作成する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface POS 0/3/0/1
RP/0/RSP0/CPU0:router(config-if)# encapsulation frame-relay
RP/0/RSP0/CPU0:router(config-if)# frame-relay intf-type dce
RP/0/RSP0/CPU0:router(config-if)# no shutdown
RP/0/RSP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them?[yes]: yes

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router (config)# interface pos 0/3/0/1.1 point-to-point
RP/0/RSP0/CPU0:router (config-subif)#ipv4 address 10.20.3.2/24
RP/0/RSP0/CPU0:router (config-subif)# pvc 100
RP/0/RSP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them?[yes]: yes

RP/0/RSP0/CPU0:router# show interface POS 0/3/0/1
```

```
Wed Oct  8 04:20:38.037 PST DST
POS0/3/0/1 is up, line protocol is up
  Interface state transitions: 1
  Hardware is Packet over SONET/SDH
  Internet address is 10.20.3.2/24
  MTU 4474 bytes, BW 155520 Kbit
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation FRAME-RELAY, crc 32, controller loopback not set,
  LMI enq sent  0, LMI stat recvd 0, LMI upd recvd 0
  LMI enq recvd 77, LMI stat sent  77, LMI upd sent  0 , DCE LMI up
  LMI DLCI 1023  LMI type is CISCO  frame relay DCE
  Last clearing of "show interface" counters 00:00:14
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     2 packets input, 26 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     2 packets output, 26 bytes, 0 total output drops
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
```

次に、メイン POS インターフェイスで PVC を持つレイヤ 2 POS サブインターフェイスを作成する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router (config)# interface pos 0/3/0/0.1 l2transport
RP/0/RSP0/CPU0:router (config-subif)# pvc 100
RP/0/RSP0/CPU0:router(config-subif)# commit
```

POS インターフェイスでの PPP カプセル化の設定：例

次に、POS インターフェイスを作成し、PPP カプセル化を設定する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp
RP/0/RSP0/CPU0:router(config-if)# no shutdown
RP/0/RSP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes

RP/0/RSP0/CPU0:router# show interfaces POS 0/3/0/0

POS0/3/0/0 is down, line protocol is down
  Hardware is Packet over SONET
  Internet address is 172.18.189.38/27
  MTU 4474 bytes, BW 2488320 Kbit
     reliability 0/255, txload Unknown, rxload Unknown
  Encapsulation PPP, crc 32, controller loopback not set, keepalive set (
10 sec)
  LCP Closed
  Closed: IPCP
  Last clearing of "show interface" counters never
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 total output drops
```

```
Output 0 broadcast packets, 0 multicast packets
0 output errors, 0 underruns, 0 applique, 0 resets
0 output buffer failures, 0 output buffers swapped out
0 carrier transitions
```

その他の関連資料

ここでは、POS インターフェイスの設定に関連する参考資料を示します。

関連資料

関連項目	参照先
Cisco IOS XR マスター コマンド リファレンス	*『Cisco IOS XR Master Commands List』*
Cisco IOS XR インターフェイス コンフィギュレーション コマンド	*『Cisco IOS XR Interface and Hardware Component Command Reference』*
Cisco IOS XR ソフトウェアを使用するルータを初回に起動し設定するための情報	*『Cisco IOS XR Getting Started Guide』*
Cisco IOS XR AAA サービス構成情報	*『Cisco IOS XR System Security Configuration Guide』*および*『Cisco IOS XR System Security Command Reference』*
ユーザ グループとタスク ID に関する情報	*『Cisco IOS XR Interface and Hardware Component Command Reference』*

標準

標準	タイトル
FRF.1.2	*『PVC User-to-Network Interface (UNI) Implementation Agreement - July 2000』*
ANSI T1.617 Annex D	—
ITU Q.933 Annex A	—

MIB

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

RFC

RFC	タイトル
RFC 1294	*『Multiprotocol Interconnect Over Frame Relay』*
RFC 1315	*『Management Information Base for Frame Relay DTEs』*
RFC 1490	*『Multiprotocol Interconnect Over Frame Relay』*
RFC 1586	*『Guidelines for Running OSPF Over Frame Relay Networks』*
RFC 1604	*『Definitions of Managed Objects for Frame Relay Service』*
RFC 2115	*『Management Information Base for Frame Relay DTEs Using SMIv2』*
RFC 2390	*『Inverse Address Resolution Protocol』*
RFC 2427	*『Multiprotocol Interconnect Over Frame Relay』*
RFC 2954	*『Definitions of Managed Objects for Frame Relay Service』*

シスコのテクニカル サポート

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

その他の関連資料

Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定

ここでは、Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定について説明します。

シリアル インターフェイスについて設定する前に、そのインターフェイスと関連付けられたクリア チャネル T3/E3 コントローラまたはチャネライズド T1/E1 コントローラ（DS0 チャネル）を設定する必要があります。

シリアル コントローラ インターフェイス設定の機能履歴

リリース	変更内容
リリース 3.3.0	この機能は、Cisco XR 12000 シリーズ ルータで導入されました。 次のハードウェアについて、Cisco XR 12000 シリーズ ルータでのサポートが追加されました。 • Cisco XR 12000 SIP-401 • Cisco XR 12000 SIP-501 • Cisco XR 12000 SIP-601 次の SPA について、Cisco XR 12000 シリーズ ルータでのサポートが追加されました。 • Cisco 2 ポートおよび 4 ポート チャネライズド T3/DS0 SPA • Cisco 2 ポートおよび 4 ポート T3/E3 シリアル SPA
リリース 3.4.0	次の機能のサポートが追加されました。 • 相手先固定接続（PVC）とのサブインターフェイス • 次のハードウェア上のシリアル メイン インターフェイスおよび PVC でのフレームリレー カプセル化 – Cisco 8 ポート チャネライズド T1/E1 SPA – Cisco 2 ポートおよび 4 ポート チャネライズド T3/DS0 SPA – Cisco 2 ポートおよび 4 ポート T3/E3 シリアル SPA – Cisco 1 ポート チャネライズド OC-3 SPA – Cisco 1 ポート チャネライズド OC-12 SPA – Cisco 1 ポート チャネライズド OC-48 SPA – Cisco 1 ポート チャネライズド OC-12/STM-4 ISE ラインカード

リリース 3.4.1	この機能は、Cisco CRS-1 ルータで導入されました。 次のハードウェアについて、Cisco CRS-1 ルータでのサポートが追加されました。 • Cisco CRS-1 SIP-800 • Cisco 2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA マルチリンク PPP が Cisco XR 12000 シリーズ ルータ上のシリアル インターフェイスでサポートされました。
リリース 3.5.0	次の SPA について、Cisco XR 12000 シリーズ ルータでのサポートが追加されました。 • Cisco 1 ポート チャネライズド OC-12/DS0 SPA • Cisco 1 ポート チャネライズド OC-48/STM-16 SPA
リリース 3.7.0	Cisco XR 12000 シリーズ ルータ上で、1 ポート チャネライズド OC-48/DS3 ラインカードのサポートが追加されました。
リリース 3.8.0	Cisco XR 12000 シリーズ ルータ上で、レイヤ 2 サブインターフェイス ファイルおよび次のラインカードの Quality of Service（QoS）のサポートが追加されました。 • Cisco 1 ポート チャネライズド OC-12/DS0 ラインカード • Cisco 4 ポート チャネライズド OC-12/DS3 ラインカード
リリース 3.9.0	シリアル インターフェイスのサポートが Cisco ASR 9000 シリーズ ルータで 2 ポート チャネライズド OC-12c/DS0 SPA に対して追加されました。
リリース 4.0.0	次の機能および SPA のサポートが Cisco ASR 9000 シリーズ ルータに追加されました。 • シリアル インターフェイスでの IPv4 マルチキャストのサポートが追加されました。インターフェイスでのマルチキャスト設定の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Multicast Configuration Guide*』を参照してください。 • IPHC が Cisco 2 ポート チャネライズド OC-12c/DS0 SPA に追加されました。 • Cisco 1 ポート チャネライズド OC-48/STM-16 SPA のサポートが導入されました。
リリース 4.0.0	**fragment-counter** コマンドを使用するフラグメンテーション カウンタのサポートが次の SPA に対して追加されました。 • Cisco 1 ポート チャネライズド OC-3/STM-1 SPA • Cisco 4 ポート チャネライズド T3/DS0 SPA • Cisco 8 ポート チャネライズド T1/E1 SPA

リリース 4.0.1	次の SPA のサポートが追加されました。 • Cisco 1 ポート チャネライズド OC-3/STM-1 SPA • Cisco 2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA
リリース 4.1.0	次の SPA のサポートが追加されました。 • Cisco 4 ポート チャネライズド T3/DS0 SPA • Cisco 8 ポート チャネライズド T1/E1 SPA IPHC のサポートが次の SPA に追加されました。 • Cisco 1 ポート チャネライズド OC-3/STM-1 SPA • Cisco 4 ポート チャネライズド T3/DS0 SPA • Cisco 8 ポート チャネライズド T1/E1 SPA • Cisco 2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA

内容

シリアル インターフェイスの設定の前提条件

シリアル インターフェイスを設定する前に、次のタスクと条件を満たしていることを確認してください。

- 適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。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。
- 2 ポートまたは 4 ポート クリア チャネル T3/E3 SPA がインストールされている。
- Cisco ASR 9000 シリーズ ルータに次の SIP と、次の SPA のいずれかをインストールしておく必要があります。
  - Cisco SIP 700 SPA インターフェイス プロセッサ
  - Cisco 1 ポート チャネライズド OC-3/STM-1 SPA
  - Cisco 2 ポート チャネライズド OC-12c/DS0 SPA
  - Cisco 1 ポート チャネライズド OC-48/STM-16 SPA
  - Cisco 2 ポートまたは 4 ポート クリア チャネル T3/E3 SPA
  - Cisco 4 ポート チャネライズド T3/DS0 SPA
  - Cisco 8 ポート チャネライズド T1/E1 SPA

- 使用しているハードウェアは、T3/E3 または T1/E1 コントローラおよびシリアル インターフェイスをサポートする必要があります。

  次のハードウェアは、T3/E3 コントローラおよびシリアル インターフェイスをサポートしています。

  - Cisco 2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA
  - Cisco 2 ポートおよび 4 ポート チャネライズド T3 SPA
  - Cisco 4 ポート チャネライズド OC-12/DS3 ラインカード
  - Cisco 1 ポート チャネライズド OC-48/STM-16 SPA およびラインカード

  次のハードウェアが、T1/E1 コントローラおよび DS0 チャネルをサポートしていることを確認します。

  - Cisco 2 ポートおよび 4 ポート チャネライズド T3 SPA
  - Cisco 4 ポート チャネライズド OC-12/DS3 ラインカード
  - Cisco 1 ポート チャネライズド OC-12/DS0 SPA およびラインカード
  - Cisco 1 ポート チャネライズド OC-48/DS3 SPA およびラインカード
  - Cisco 1 ポート チャネライズド OC3/STM-1 SPA
  - Cisco 8 ポート チャネライズド T1/E1 SPA

  次のハードウェアはシリアル インターフェイスをサポートします。

  - Cisco 2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA
  - Cisco 2 ポートおよび 4 ポート チャネライズド T3 SPA
  - Cisco 4 ポート チャネライズド OC-12/DS3 ラインカード
  - Cisco 1 ポート チャネライズド OC-12/DS0 SPA およびラインカード
  - Cisco 1 ポート チャネライズド OC-48/DS3 SPA およびラインカード
  - Cisco 1 ポート チャネライズド OC3/STM-1 SPA
  - Cisco 8 ポート チャネライズド T1/E1 SPA

（注） 2 ポートおよび 4 ポート チャネライズド T3 SPA は、クリア チャネル モードで動作することも、チャネル化して 28 個の T1 コントローラまたは 21 個の E1 コントローラとすることもできます。

（注） Cisco 4 ポート チャネライズド T3/DS0 SPA は、クリア チャネル モードで実行できます。または、28 T1 コントローラか 21 E1 コントローラにチャネル化できます。

- 設定するシリアル インターフェイスと関連付ける、クリア チャネル T3/E3 コントローラまたはチャネライズド T3-to-T1/E1 コントローラが設定済みであることが必要です。手順については、このマニュアルの *「Cisco ASR 9000 シリーズ ルータでのクリア チャネル T3/E3 およびチャネライズド T3 および T1/E1 コントローラの設定」* モジュールを参照してください。

（注） チャネライズド T3-to-T1/E1 コントローラでは、ユーザが T1/E1 コントローラの各 DS0 チャネル グループを設定するときに、シリアル インターフェイスが自動的に作成されます。

シリアル インターフェイスの設定に関する情報

シリアル 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。

Cisco ASR 9000 シリーズ ルータでは、単一のシリアル インターフェイスは単一のインターフェイス上でデータを伝送し、このときに PPP、Cisco HDLC、またはフレームリレーのカプセル化が使用されます。

概要：クリア チャネル SPA 上のシリアル インターフェイスの設定

表 14 は、Cisco 2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA 上で T3 シリアル インターフェイスを設定するために必要なタスクの概要です。

表 14　概要：クリア チャネル SPA 上の T3 シリアル インターフェイスの設定

ステップ	作業	Module	セクション
1.	必要に応じて、**hw-module subslot** コマンドを使用して SPA のシリアル モードを T3 に設定します。 （注）　デフォルトで、2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA は T3 モードで実行するように設定されています。	「Cisco ASR 9000 シリーズ ルータでのクリア チャネル T3/E3 およびチャネライズド T3 および T1/E1 コントローラの設定」	カード タイプの設定
2.	T3 コントローラを設定します。	「Cisco ASR 9000 シリーズ ルータでのクリア チャネル T3/E3 およびチャネライズド T3 および T1/E1 コントローラの設定」	カード タイプの設定
3.	ステップ 2 で設定した T3 コントローラに関連付けるシリアル インターフェイスを設定します。	「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」	「シリアル インターフェイスの設定方法」

表 15 は、2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA 上に E3 シリアル インターフェイスを設定するために必要なタスクの概要です。

表 15　概要：クリア チャネル SPA 上の E3 シリアル インターフェイスの設定

ステップ	作業	Module	セクション
1.	**hw-module subslot** コマンドを使用して SPA のシリアル モードを E3 に設定します。	「Cisco ASR 9000 シリーズ ルータでのクリア チャネル T3/E3 およびチャネライズド T3 および T1/E1 コントローラの設定」	カード タイプの設定
2.	E3 コントローラを設定します。	「Cisco ASR 9000 シリーズ ルータでのクリア チャネル T3/E3 およびチャネライズド T3 および T1/E1 コントローラの設定」	カード タイプの設定
3.	ステップ 2 で設定した E3 コントローラに関連付けるシリアル インターフェイスを設定します。	「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」	*シリアル インターフェイスの設定方法*

概要：チャネライズド SPA 上のシリアル インターフェイスの設定

表 16表 17 は、次の SPA およびラインカード上で T1 シリアル インターフェイスを設定するために必要なタスクの概要です。

- Cisco 2 ポートおよび 4 ポート チャネライズド T3 SPA
- Cisco 4 ポート チャネライズド OC-12/DS3 ラインカード
- Cisco 1 ポート チャネライズド OC-12/DS0 SPA およびラインカード
- Cisco 1 ポート チャネライズド OC-48/STM-16 SPA およびラインカード
- Cisco 2 ポート チャネライズド OC-12c/DS0 SPA

表 16　概要：T1 DS0 チャネル上のシリアル インターフェイスの設定

ステップ	作業	Module	セクション
1.	T3 コントローラ パラメータを設定し、SPA モードを T3 に設定します。 28 T1 コントローラが自動的に作成されます。	「Cisco ASR 9000 シリーズ ルータでのクリア チャネル T3/E3 およびチャネライズド T3 および T1/E1 コントローラの設定」	カード タイプの設定 チャネル化された T3 コントローラの設定
2.	T1 コントローラ上に、DS0 チャネル グループを作成し、設定します。	「Cisco ASR 9000 シリーズ ルータでのクリア チャネル T3/E3 およびチャネライズド T3 および T1/E1 コントローラの設定」	T1 コントローラの設定
3.	ステップ 2 で作成したチャネル グループと関連付けられたシリアル インターフェイスを設定します。	「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」	シリアル インターフェイスの設定方法

表 17　概要：T1 DS0 チャネル上のシリアル インターフェイスの設定

ステップ	作業	Module	セクション
1.	SONET コントローラ パラメータおよび STS ストリームを T3 モード用に設定します。	「Cisco ASR 9000 シリーズ ルータ でのチャネライズド SONET/SDH の設定」	SONET T3 チャネルおよび VT1.5 がマッピングされた T1 チャネルの設定
2.	T3 コントローラ パラメータを設定し、モードを T1 に設定します。 28 T1 コントローラが自動的に作成されます。	「Cisco ASR 9000 シリーズ ルータでのクリア チャネル T3/E3 およびチャネライズド T3 および T1/E1 コントローラの設定」	チャネル化された T3 コントローラの設定
3.	T1 コントローラ上に、DS0 チャネル グループを作成し、設定します。	「Cisco ASR 9000 シリーズ ルータでのクリア チャネル T3/E3 およびチャネライズド T3 および T1/E1 コントローラの設定」	T1 コントローラの設定
4.	ステップ 2 で作成したチャネル グループと関連付けられたシリアル インターフェイスを設定します。	「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」	シリアル インターフェイスの設定方法

表 18 は、次の SPA およびラインカード上で E1 シリアル インターフェイスを設定するために必要なタスクの概要です。

- 2 ポートおよび 4 ポート チャネライズド T3 SPA
- 4 ポート チャネライズド OC-12/DS3 ラインカード
- 1 ポート チャネライズド OC-12/DS0 SPA およびラインカード
- 1 ポート チャネライズド OC-48/DS3 SPA およびラインカード
- 1 ポート チャネライズド OC-3/STM-1 SPA
- 2 ポート チャネライズド OC-12c/DS0 SPA

表 18　概要：E1 DS0 チャネル上のシリアル インターフェイスの設定

ステップ	作業	Module	セクション
1.	E3 コントローラ パラメータを設定し、SPA モードを T3 に設定します。 21 E1 コントローラが自動的に作成されます。	「Cisco ASR 9000 シリーズ ルータでのクリア チャネル T3/E3 およびチャネライズド T3 および T1/E1 コントローラの設定」	チャネル化された T3 コントローラの設定
2.	E1 コントローラ上に、DS0 チャネル グループを作成し、設定します。	「Cisco ASR 9000 シリーズ ルータでのクリア チャネル T3/E3 およびチャネライズド T3 および T1/E1 コントローラの設定」	E1 コントローラの設定
3.	ステップ 2 で作成したチャネル グループと関連付けられたシリアル インターフェイスを設定します。	「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」	シリアル インターフェイスの設定方法

表 19　概要：E1 DS0 チャネル上のシリアル インターフェイスの設定

ステップ	作業	Module	セクション
1.	SONET コントローラ パラメータおよび STS ストリームを T3 モード用に設定します。	「Cisco ASR 9000 シリーズ ルータ でのチャネライズド SONET/SDH の設定」	SONET T3 チャネルおよび VT1.5 がマッピングされた T1 チャネルの設定
2.	T3 コントローラ パラメータを設定し、モードを E1 に設定します。 21 E1 コントローラが自動的に作成されます。	「Cisco ASR 9000 シリーズ ルータでのクリア チャネル T3/E3 およびチャネライズド T3 および T1/E1 コントローラの設定」	チャネル化された T3 コントローラの設定
3.	E1 コントローラ上に、DS0 チャネル グループを作成し、設定します。	「Cisco ASR 9000 シリーズ ルータでのクリア チャネル T3/E3 およびチャネライズド T3 および T1/E1 コントローラの設定」	E1 コントローラの設定
4.	ステップ 2 で作成したチャネル グループと関連付けられたシリアル インターフェイスを設定します。	「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」	シリアル インターフェイスの設定方法

表 20 は、1 ポート チャネライズド OC-48/STM-16 SPA 上に、T3 シリアル インターフェイスを設定するために必要なタスクの概要です

表 20　概要：T3 チャネル上のシリアル インターフェイスの設定

ステップ	作業	Module	セクション
1.	SONET コントローラ パラメータおよび STS ストリームを設定します。	「Cisco ASR 9000 シリーズ ルータ でのチャネライズド SONET/SDH の設定」	T3 のためのクリア チャネル SONET コントローラの設定
2.	T3 の STS ストリーム モードを設定して、T3 コントローラ パラメータを設定します。	「Cisco ASR 9000 シリーズ ルータ でのチャネライズド SONET/SDH の設定」	T3 のためのクリア チャネル SONET コントローラの設定
3.	シリアル インターフェイスを設定します。	「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」	シリアル インターフェイスの設定方法

Cisco HDLC カプセル化

Cisco ハイレベル データリンク コントローラ（HDLC）は、HDLC を使用して同期シリアル リンクでデータを送信するためのシスコ独自のプロトコルです。また、Cisco HDLC は、シリアル リンクのキープアライブを維持するシリアル ライン アドレス解決プロトコル（SLARP）と呼ばれる単純な制御プロトコルも提供します。HDLC は、Cisco IOS XR ソフトウェアにおけるシリアル インターフェイスのデフォルト カプセル化タイプです。Cisco HDLC は、効率的なパケットの説明およびエラー制御を行う、オープン システム インターコネクション（OSI）スタックのレイヤ 2（データリンク）におけるデフォルトのデータ カプセル化のデフォルト プロトコルです。

（注） Cisco HDLC は、シリアル インターフェイスのデフォルトのカプセル化タイプです。

Cisco HDLC は、キープアライブを使用してリンク ステートをモニタリングします（「キープアライブ タイマー」（P.549）を参照）。

（注） キープアライブ タイマーを設定した後で、ピアに送信される SLARP パケットの情報を表示するには、**debug chdlc slarp packet** コマンドを使用します。

PPP Encapsulation

PPP は、同期シリアル リンクでデータを送信するために使用される標準プロトコルです。また、PPP は、リンクのプロパティをネゴシエートするリンク コントロール プロトコル（LCP）も提供します。LCP は、エコー要求および応答を使用して、リンクの継続的なアベイラビリティをモニタリングします。

（注） インターフェイスに PPP カプセル化が設定されている場合、リンクがダウンしたと宣言され、ECHOREP 応答を受信せずに 5 つの ECHOREQ パケットが送信された後、完全な LCP ネゴシエーションが再開されます。

PPP は、リンク上で動作するデータ プロトコルのプロパティをネゴシエートする、次のネットワーク コントロール プロトコル（NCP）を提供します。

- IP のプロパティをネゴシエートする IP コントロール プロトコル（IPCP）
- MPLS のプロパティをネゴシエートするマルチプロトコル ラベル スイッチング コントロール プロセッサ（MPLSCP）
- CDP のプロパティをネゴシエートする Cisco Discovery Protocol コントロール プロセッサ（CDPCP）
- IP バージョン 6（IPv6）のプロパティをネゴシエートする IPv6CP
- OSI のプロパティをネゴシエートするオープン システム インターコネクション コントロール プロセッサ（OSICP）

PPP は、キープアライブを使用してリンク ステートをモニタリングします（「キープアライブ タイマー」（P.549）を参照）。

PPP は次の認証プロトコルをサポートします。これらのプロトコルでは、接続によるデータ トラフィックのフローを許可する前にそのアイデンティティを証明するために、リモート デバイスが必要です。

- チャレンジ ハンドシェイク認証プロトコル（CHAP）：CHAP は、リモート デバイスにチャレンジ メッセージを送信します。リモート デバイスは、共有秘密を使用してチャレンジの値を暗号化し、暗号化された値とその名前を応答メッセージでローカル ルータに戻します。ローカル ルータは、ローカル ユーザ名またはリモート セキュリティ サーバ データベース内に保存されたシークレットのうち、リモート デバイスの名前に対応するものを見つけます。この保存されたシークレットを使用して、元のチャレンジを暗号化し、暗号化された値が一致していることを確認します。

- マイクロソフト チャレンジ ハンドシェイク認証プロトコル（MS-CHAP）：MS-CHAP は CHAP の Microsoft バージョンです。CHAP の標準バージョンと同様に、MS-CHAP は PPP 認証に使用されます。この場合、認証は、Microsoft Windows NT または Microsoft Windows 95 を使用するパーソナル コンピュータとネットワーク アクセス サーバとして機能する Cisco ルータまたはアクセス サーバの間で行われます。
- パスワード認証プロトコル（PAP）：PAP 認証では、ローカル ユーザ名データベース内またはリモート セキュリティ サーバ データベース内の一致するエントリに照らし合わせてチェックする名前とパスワードを送信するために、リモート デバイスが必要です。

（注） PPP 認証プロトコルのイネーブル化および設定の詳細については、このマニュアルの *「Cisco ASR 9000 シリーズ ルータ での PPP の設定」* モジュールを参照してください。

シリアル インターフェイスで CHAP、MS-CHAP、および PAP をイネーブルにするには、インターフェイス コンフィギュレーション モードで **ppp authentication** コマンドを使用します。

（注） PPP 認証をイネーブル化またはディセーブル化しても、ローカル ルータがリモート デバイスに対して自身を認証しようとすることには変わりありません。

マルチリンク PPP

マルチリンク PPP（MLPPP）は、次の SPA でサポートされます。

- 1 ポート チャネライズド OC-12/DS0 SPA およびラインカード
- 2 ポートおよび 4 ポート チャネライズド T3 SPA
- 8 ポート チャネライズド T1/E1 SPA
- 1 ポート チャネライズド OC-3/STM-1 SPA
- 2 ポート チャネライズド OC-12/DS0 SPA

MLPPP は、複数の物理リンクを 1 つの論理リンクに組み合わせる方式を提供します。MLPPP の実装によって、複数の PPP シリアル インターフェイスが 1 つのマルチリンク インターフェイスにまとめられます。MLPPP は、複数の PPP リンクでデータグラムの断片化、再編成、および配列を行います。

MLPPP は、QoS を除く PPP シリアル インターフェイスでサポートされる同じ機能を提供します。また、次の追加機能も提供します。

- 128 バイト、256 バイト、および 512 バイトのフラグメント サイズ
- 長いシーケンス番号（24 ビット）
- 失われたフラグメントの検出タイムアウト期間（80 ms）
- 最小アクティブ リンクの設定オプション
- マルチリンク インターフェイスでの LCP エコー要求および応答のサポート
- フル T1 および E1 フレームおよび非フレーム リンク

シリアル インターフェイスで MLPPP を設定する方法の詳細については、このマニュアルの *「Cisco ASR 9000 シリーズ ルータ での PPP の設定」* モジュールを参照してください。

キープアライブ タイマー

シスコ キープアライブは、リンク ステートをモニタリングする場合に便利です。キープアライブは、キープアライブ タイマーの値によって決定される頻度で、定期的にピアに送信され、ピアから受信されます。受け入れ可能なキープアライブがピアから受信されない場合、リンクはダウン状態に移行します。ピアから受け入れ可能なキープアライブが受信されるか、キープアライブがディセーブルになると、リンクはすぐにアップ状態に移行します。

（注） **keepalive** コマンドは、HDLC または PPP カプセル化を使用するシリアル インターフェイスに適用されます。フレーム リレー カプセル化を使用するシリアル インターフェイスには適用されません。

各カプセル化タイプでは、ピアによって無視される特定の数のキープアライブがシリアル インターフェイスのダウン状態への移行をトリガーします。HDLC カプセル化の場合、無視されるキープアライブが 3 つあると、インターフェイスがダウン状態になります。PPP カプセル化の場合、無視されるキープアライブが 5 つあると、インターフェイスがダウン状態になります。ECHOREQ パケットは、LCP ネゴシエーションが完了した場合（LCP が開いている場合など）に限り、送信されます。

LCP が ECHOREQ パケットをピアに送信する頻度を設定するには、インターフェイス コンフィギュレーション モードで **keepalive** コマンドを使用します。システムを 10 秒のデフォルト キープアライブ インターバルに戻すには、**keepalive** コマンドを **no** 引数とともに使用します。キープアライブをディセーブルにするには、**keepalive disable** コマンドを使用します。PPP と Cisco HDLC では、0 のキープアライブはキープアライブをディセーブルにし、**show running-config** コマンド出力では、**keepalive disable** として報告されます。

（注） Minimal Disruptive Restart（MDR）の間は、キープアライブが失敗する可能性があります。したがって、キープアライブ タイマーは、両端において、ディセーブルにするか MDR の時間よりも長くなるように設定しておく必要があります。

（注） Minimal Disruptive Restart（MDR）のアップグレードを実行する前に、Cisco XR 12000 シリーズ ルータでのキープアライブを無効にすることを推奨します。

（注） Minimal Disruptive Restart（MDR）のアップグレードを実行する前に、Cisco CRS-1 ルータでのキープアライブ インターバルを 10 秒以上に設定することを推奨します。

LCP がピアで動作していて、ECHOREQ パケットを受信すると、キープアライブがピアでイネーブルかどうかに関係なく、エコー応答（ECHOREP）パケットで応答します。

キープアライブは、2 つのピアの間で独立しています。一方のピアの端ではキープアライブをイネーブルにし、もう一方の端ではディセーブルにすることができます。キープアライブがローカルでディセーブルの場合でも、LCP は受信する ECHOREQ パケットに ECHOREP パケットで応答します。同様に、LCP は、それぞれの端のキープアライブの期間が異なる場合でも機能します。

（注） キープアライブ タイマーを設定した後で、ピアに送信される SLARP パケットの情報を表示するには、**debug chdlc slarp packet** コマンドと他の Cisco HDLC **debug** コマンドを使用します。

フレーム リレー カプセル化

シリアル インターフェイスでフレーム リレー カプセル化がイネーブルの場合、インターフェイスの設定は階層型になっており、次の要素で構成されます。

1. シリアル メイン インターフェイスは、物理インターフェイスおよびポートで構成されます。Cisco HDLC および PPP カプセル化接続をサポートするシリアル インターフェイスを使用していない場合、シリアル メイン インターフェイスの下に相手先固定接続（PVC）があるサブインターフェイスを設定する必要があります。フレーム リレー接続は、PVC でのみサポートされます。
2. シリアル サブインターフェイスは、シリアル メイン インターフェイスの下に設定されます。シリアル サブインターフェイスは、シリアル サブインターフェイスの下に PVC を設定するまで、トラフィックをアクティブに伝送しません。レイヤ 3 の設定は、一般的にサブインターフェイス上で行われます。
3. ポイントツーポイント PVC は、シリアル サブインターフェイスの下に設定されます。メイン インターフェイスの下に PVC を直接設定できません。1 つのサブインターフェイスに対して 1 つのポイントツーポイント PVC を設定できます。PVC はあらかじめ定義された回線パスを使用し、パスが中断されるとエラーが発生します。PVC は、どちらかの設定から回線を削除しない限り、アクティブな状態に保たれます。シリアル PVC での接続は、フレーム リレー カプセル化だけをサポートします。

（注） 親インターフェイスの管理状態は、サブインターフェイスとその PVC の状態を決定します。親インターフェイスまたはサブインターフェイスの管理状態が変わると、その親インターフェイスまたはサブインターフェイスの下に設定されたすべての子 PVC の管理状態も変わります。

シリアル インターフェイスでフレームリレー カプセル化を設定するには、**encapsulation frame-relay** コマンドを使用します。

フレーム リレー インターフェイスは、次の 2 つのタイプのカプセル化フレームをサポートします。

- Cisco（デフォルト）
- IETF

PVC に Cisco または IETF カプセル化を設定するには、PVC コンフィギュレーション モードで **encap** コマンドを使用します。PVC にカプセル化のタイプが明示的に設定されていない場合、その PVC は、メイン シリアル インターフェイスからカプセル化のタイプを引き継ぎます。

（注） Cisco カプセル化は、MPLS に設定されたシリアル メイン インターフェイスで必要です。IETF カプセル化は、MPLS ではサポートされていません。

インターフェイスにフレーム リレーのカプセル化を設定する前に、そのインターフェイスから以前のレイヤ 3 のすべての設定が除去されていることを確認する必要があります。たとえば、メイン インターフェイスの下に直接設定されている IP アドレスがないことを確認する必要があります。IP アドレスが直接設定されていると、メイン インターフェイスの下で行われたフレーム リレー設定が実行できなくなります。

フレーム リレー インターフェイスでの LMI

ローカル管理インターフェイス（LMI）プロトコルは、PVC の追加、削除、およびステータスをモニタリングします。また、LMI は、フレーム リレー UNI インターフェイスを形成するリンクの完全性を確認します。デフォルトでは、**cisco** LMI はすべての PVC でイネーブルです。ただし、デフォルトの

LMI タイプを ANSI または Q.933 に変更できます。手順については、このマニュアルの *「Cisco ASR 9000 シリーズ ルータでのフレーム リレーの設定」*モジュールの *「インターフェイスでのデフォルト フレームリレー設定の変更」*を参照してください。

LMI のタイプが **cisco** （デフォルトの LMI タイプ）である場合、1 つのインターフェイスでサポートできる PVC の最大数は、メイン インターフェイスの MTU サイズに関連しています。カードまたは SPA でサポートされる PVC の最大数を計算するには、次の公式を使用します。

```
(MTU - 13)/8 = PVC の最大数
```

（注） シリアル インターフェイスでの **mtu** コマンドのデフォルト設定は、1504 バイトです。したがって、**cisco** LMI が設定されたシリアル インターフェイスでサポートされる PVC のデフォルト数は、186 です。

フレーム リレーでのレイヤ 2 トンネル プロトコル バージョン 3 ベースのレイヤ 2 VPN

レイヤ 2 トンネル プロトコル バージョン 3（L2TPv3）機能は、レイヤ 2 バーチャル プライベート ネットワーク（VPN）を使用する IP コア ネットワークでレイヤ 2 ペイロードをトンネリングする L2TP プロトコルを定義します。

L2TPv3 は、レイヤ 2 プロトコルを転送するために使用するトンネリング プロトコルです。いくつかの異なる設定で動作し、パケット スイッチド ネットワークでいくつかの異なるレイヤ 2 プロトコルおよび接続をトンネリングできます。

L2TPv3 を設定するには、その前に、L2TPv3 疑似配線を運用する 2 つの接続回線（AC）間の接続を設定する必要があります。Cisco IOS XR ソフトウェアは、2 つの AC が結合されているポイントツーポイント、エンドツーエンドのサービスをサポートします。

ここでは、フレームリレー カプセル化シリアル インターフェイス上でレイヤ 2 AC を設定する方法について説明します。

（注） シリアル インターフェイスは、DLCI モードのレイヤ 2 AC だけをサポートします。レイヤ 2 ポート モードの AC は、シリアル インターフェイスではサポートされていません。

ネットワークでの L2TPv3 の設定に関する詳細については、『*Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco CRS Router* 』の「*Implementing Layer 2 Tunnel Protocol Version 3*」モジュールを参照してください。L2VPN の設定の詳細については、『*Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco CRS Router*』の「*Implementing MPLS Layer 2 VPNs*」モジュールを参照してください。

ネットワークでの L2TPv3 の設定に関する詳細については、『*Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco XR 12000 Series Router* 』の「*Implementing Layer 2 Tunnel Protocol Version 3 on Cisco IOS XR Software*」モジュールを参照してください。L2VPN の設定の詳細については、『*Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco XR 12000 Series Router*』の「*Implementing MPLS Layer 2 VPNs on Cisco IOS XR Software*」モジュールを参照してください。

シリアル インターフェイス コンフィギュレーションのデフォルト設定

T3/E3 SPA でインターフェイスをイネーブルにし、追加のコンフィギュレーション コマンドを適用しない場合、デフォルトのインターフェイス設定は表 21 のようになります。これらのデフォルト設定はコンフィギュレーションで変更できます。

表 21　　シリアル インターフェイスのデフォルト設定

パラメータ	設定ファイルのエントリ	デフォルト設定
Keepalive （注）　**keepalive** コマンドは、HDLC または PPP カプセル化を使用するシリアル インターフェイスに適用されます。フレーム リレー カプセル化を使用するシリアル インターフェイスには適用されません。	**keepalive** [**disable**] **no keepalive**	10 秒のキープアライブ
カプセル化	**encapsulation** [**hdlc** | **ppp** | **frame-relay** [**ietf**]]	hdlc
最大伝送単位（MTU）	**mtu** *bytes*	1504 バイト
巡回冗長検査（CRC）	**crc** [**16** | **32**]	16
シリアル インターフェイス上のデータ ストリームの反転	**invert**	データ ストリームは反転しません。
ペイロード スクランブリング（暗号化）	**scramble**	スクランブリングはディセーブルです。
パケット間に挿入される HDLC フラグ シーケンスの数	**transmit-delay**	デフォルトは 0（ディセーブル）です。

（注）　デフォルト設定は、**show running-config** コマンドの出力には含まれません。

シリアル インターフェイスの表記方法

クリア チャネル SPA 上のシリアル インターフェイスの表記方法は、*rack*/*slot*/*module*/*port* です。次に例を示します。

```
interface serial 0/0/1/2
```

チャネライズド SPA 上の T1、E1、および DS0 インターフェイスの表記方法は、*rack*/*slot*/*module*/*port*/*channel-num*:*channel-group-number* です。次に例を示します。

```
interface serial 0/0/1/2/4:3
```

シリアル インターフェイス下にサブインターフェイスと PVC を設定すると、ルータでは、シリアル インターフェイス アドレスの末尾にサブインターフェイス番号が含まれます。この場合の表記方法は *rack*/*slot*/*module*/*port*[/*channel-num*:*channel-group-number*].*subinterface* です。次に例を示します。

```
interface serial 0/0/1/2.1
interface serial 0/0/1/2/4:3.1
```

（注） 値の間のスラッシュは、表記の一部として必要です。

シリアル インターフェイスの表記方法の構文は次のようになります。

- *rack*：ラックのシャーシ番号。
- *slot*：モジュラ サービス カードまたはラインカードの物理スロット番号。
- *module*：モジュール番号。共有ポート アダプタ（SPA）は、そのサブスロット番号から参照されます。
- *port*：コントローラの物理ポート番号。
- *channel-num*：T1 または E1 のチャネル番号。T1 チャネルの範囲は 0 ～ 23、E1 チャネルの範囲は 0 ～ 31 です。
- *channel-group-number*：タイムスロット番号。T1 タイムスロットの範囲は 1 ～ 24、E1 タイムスロットの範囲は 1 ～ 31 です。*channel-group-number* の前には、スラッシュではなくコロンを付けます。
- *subinterface*：サブインターフェイス番号。

有効なインターフェイスの選択肢一覧を表示するには、**serial** キーワードに続けて疑問符（**?**）のオンライン ヘルプ機能を使用します。

IPHC の概要

IP ヘッダー圧縮（IPHC）は、1 つの送信におけるパケットのヘッダーの大部分が、フロー全体を通して一定に保たれるという前提に基づいています。1 つのフローの中で、関連パケットのヘッダーのいくつかのフィールドだけが変更されます。

IPHC は、パケットごとに異なるフィールドだけが圧縮されたヘッダーに含まれるように、これらのヘッダーを圧縮します。どのパケットでも同じであるフィールドは、圧縮済みヘッダーでは除去されます。フル ヘッダーが、圧縮済みヘッダーの間に送信されます。

フル ヘッダーとは、未圧縮のヘッダーであり、その内容は元のヘッダー フィールドすべてと、そのフローを識別するための追加情報（コンテキスト ID）です。フル ヘッダーを圧縮パケットの間で送信する間隔を設定するには、**refresh max-period** コマンドと **refresh max-time** コマンドを使用します。

IPHC のコンテキストは、圧縮済みパケットのコンプレッサ（送信者）とデコンプレッサ（レシーバ）がフロー内のパケットをエンコード/デコードするために使用されます。コンテキストは、コンプレッサとデコンプレッサに保存され、両端でのデルタ計算に使用されます。1 つのインターフェイスに対して許可されるコンテキストの数は設定可能です。圧縮できるヘッダーの最大サイズも設定できます。

IPHC では、RTP および UDP トラフィックの圧縮と圧縮解除、および TCP および CTCP トラフィックでの CN の圧縮解除がサポートされています。

ユーザは次の種類の圧縮形式の 1 つを選択できます。

- インターネット技術特別調査委員会（IETF）標準形式。
  RFC2507 および RFC2508 の圧縮スキームを使用します。
- IPHC 形式。
  IETF に類似したオプションがあります。

表 22 に、IPHC の機能、その機能の値、およびデフォルトを示します。

表 22　IPHC の機能およびデフォルト設定

IPHC の機能	値	デフォルト
TCP コンテキスト	0 ～ 255	1
非 TCP コンテキスト	1 ～ 6000	16
圧縮形式オプション	IETF または IPHC	—
フィードバック メッセージ	Enable または Disable	Enabled
最大リフレッシュ期間サイズ	1 ～ 65535 パケット	256
最大リフレッシュ期間	0 ～ 255 秒	5
最大ヘッダー サイズ	20 ～ 40 バイト	40
Real Time Protocol（RTP）	Enable または Disable	Enabled
リフレッシュ RTP	Enable または Disable	Disable

現在、IP ヘッダーのプロトコル フィールドが UDP である IPv4 ユニキャスト パケットだけが圧縮されます。

IPHC は、インターフェイスで次のように設定されます。

- IPHC スロット レベル コマンドを設定
- IPHC プロファイルを作成
- プロファイルに IPHC 属性を設定
- インターフェイスにプロファイルを割り当て

IPHC プロファイルには、インターフェイスで Real Time Protocol（RTP）をイネーブルする **rtp** コマンドが含まれている必要があります。含まれていない場合、プロファイルはイネーブルになりません。**refresh rtp** コマンドを使用して、設定済みのリフレッシュ設定値を RTP パケットに対してイネーブルにする必要があります。デフォルトでは、リフレッシュ RTP はディセーブルであり、フローの最初のパケットだけが「フル ヘッダー」パケットとして送信されます。

一部の属性は（たとえばフィードバック メッセージ、最大リフレッシュ期間サイズ、最大リフレッシュ期間、最大ヘッダー サイズ）、プロファイルがインターフェイス上でイネーブルになったときに、そのプロファイルで属性値が設定されていなければ、デフォルト値が適用されます。

現時点では、IPHC がサポートされるのは、PPP カプセル化を使用するシリアル インターフェイスと、PPP カプセル化インターフェイスを使用するマルチリンクのみです。

IPHC は一般的に、1 つのインターフェイスのカスタマー エッジ（CE）端とプロバイダー エッジ（PE）端の間で設定されます。この機能が動作するには、インターフェイスの両端で設定されている必要があります。PPP プロトコルによってインターフェイスの両端間で IPHC 固有のパラメータがネゴシエートされ、両端で設定されている値のうち、最小の値が使用されます。

QoS および IPHC

IPHC プロファイルをあるインターフェイスに対してイネーブルにする場合に、Quality of Service（QoS）サービス ポリシーに一致するパケットだけにその IPHC プロファイルを適用するように設定することができます。この場合、QoS サービス ポリシー クラス属性によって、圧縮するパケットが決定されます。これにより、ユーザはより IPHC の精度を最適化できます。

ポリシー マップは、**service-policy** コマンドを使用してインターフェイスに割り当てます。IPHC アクションは、出力サービス ポリシーにだけ適用されます。IPHC は、入力サービス ポリシーではサポートされません。

ユーザは、次のように QoS を使用して IPHC を設定できます。

- **compress header ip** アクションで QoS **policy-map** を作成します。
- **ipv4 iphc profile** *profile_name* **mode service-policy** コマンドを使用して、IPHC プロファイルをインターフェイスに割り当てます。
- QoS **policy-map** に **compress header ip** アクションをアタッチします。これには、**service-policy output** コマンドを使用します。

QoS を使用して IPHC を設定する方法の例については、「MLPPP/LFI および QoS を使用するシリアル インターフェイスでの IPHC の設定：例」（P.589）を参照してください。

QoS の設定の詳細については、『*Cisco XR 12000 Series Router Modular Quality of Service Configuration Guide*』、『*Cisco XR 12000 Series Router Modular Quality of Service Command ReferenceCisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Configuration Guide*』、および『*Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Command Reference*』を参照してください。

シリアル インターフェイスの設定方法

チャネライズドまたはクリア チャネル T3/E3 コントローラを設定した後は（手順はこのマニュアルの「Cisco ASR 9000 シリーズ ルータでのクリア チャネル T3/E3 およびチャネライズド T3 および T1/E1 コントローラの設定」モジュールを参照してください）、そのコントローラに関連付けるシリアル インターフェイスを設定できます。

次のタスクでは、シリアル インターフェイスを設定する方法について説明します。

シリアル インターフェイスの始動

ここでは、シリアル インターフェイスの始動に使用するコマンドについて説明します。

前提条件

Cisco XR 12000 シリーズ ルータには、1 つ以上の SIP、および 1 つ以上の SPA またはラインカードがインストールされ、Cisco IOS XR ソフトウェアを実行している必要があります。

- Cisco XR 12000 SIP-401
- Cisco XR 12000 SIP-501
- Cisco XR 12000 SIP-601
- 2 ポートおよび 4 ポート T3/E3 シリアル SPA
- 2 ポートおよび 4 ポート チャネライズド T3/DS0 シリアル SPA
- 4 ポート チャネライズド OC-12/DS3 ラインカード
- 1 ポート チャネライズド OC-12/DS0 SPA およびラインカード
- 1 ポート チャネライズド OC-48/DS3 SPA およびラインカード
- 1 ポート チャネライズド OC-3/STM-1 SPA
- 8 ポート チャネライズド T1/E1 SPA

Cisco CRS-1 ルータには、次の SIP および SPA がインストールされ、Cisco IOS XR ソフトウェアを実行している必要があります。

- Cisco CRS-1 SIP-800
- 2 ポートおよび 4 ポート T3/E3 シリアル SPA

Cisco ASR 9000 シリーズ ルータには、次の SIP と、次の SPA のうち 1 つ以上がインストールされていることと、Cisco IOS XR ソフトウェアを実行していることが必要です。

- SIP 700 SPA インターフェイス プロセッサ
- 1 ポート チャネライズド OC-3/STM-1 SPA
- 2 ポート チャネライズド OC-12c/DS0 SPA
- 1 ポート チャネライズド OC-48/STM-16 SPA
- 4 ポート チャネライズド T3/DS0 SPA
- 2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA
- 8 ポート チャネライズド T1/E1 SPA

制約事項

シリアル インターフェイスがアクティブになるためには、シリアル接続の両端の設定が一致している必要があります。

手順の概要

1. **show interfaces**
2. **configure**
3. **interface serial** *interface-path-id*
4. **ipv4 address** *ip-address*
5. **no shutdown**

6. **end**
   または
   commit
7. **exit**
8. **exit**
9. 接続の他端でインターフェイスを始動するために、ステップ 1 ～ 8 を繰り返します。
10. **show ipv4 interface brief**
11. **show interfaces serial** *interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**show interfaces** **例：** `RP/0/0RP0RSP0/CPU0:router# show interfaces`	（任意）設定されているインターフェイスを表示します。 • このコマンドを使用して、ルータが PLIM カードを認識しているかどうかも確認します。
ステップ 2	**configure** **例：** `RP/0/0RP0RSP0/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 3	**interface serial** *interface-path-id* **例：** `RP/0/0RP0RSP0/CPU0:router(config)# interface serial 0/1/0/0`	シリアル インターフェイス名と *rack/slot/module/port* 表記を指定して、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。
ステップ 4	**ipv4 address** *ip-address* **例：** `RP/0/0RP0RSP0/CPU0:router(config-if)# ipv4 address 10.1.2.1 255.255.255.224`	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。 **（注）** このインターフェイスにフレームリレー カプセル化を設定する場合は、このステップを省略してください。フレームリレーの場合、IP アドレスとサブネット マスクはサブインターフェイスに設定します。
ステップ 5	**no shutdown** **例：** `RP/0/0RP0RSP0/CPU0:router (config-if)# no shutdown`	shutdown 設定を削除します。 **（注）** shutdown 設定を削除することにより、インターフェイスでの強制的な管理上の停止が排除されるため、インターフェイスはアップ状態またはダウン状態に移行することができます（親 SONET レイヤが管理上の停止状態に設定されていないことを前提とします）。

	コマンドまたはアクション	目的
ステップ 6	**end** または **commit** **例：** `RP/0/0RP0RSP0/CPU0:router (config-if)# end` または `RP/0/0RP0RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 7	**exit** **例：** `RP/0/0RP0RSP0/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、グローバル コンフィギュレーション モードに入ります。
ステップ 8	**exit** **例：** `RP/0/0RP0RSP0/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、EXEC モードを開始します。
ステップ 9	**show interfaces** **configure** **interface serial** *interface-path-id* **no shut** **exit** **exit** **例：** `RP/0/0RP0RSP0/CPU0:router# show interfaces` `RP/0/0RP0RSP0/CPU0:router# configure` `RP/0/0RP0RSP0/CPU0:router (config)# interface serial 0/1/0/1` `RP/0/0RP0RSP0/CPU0:router(config-if)# ipv4 address 10.1.2.2 255.255.255.224` `RP/0/0RP0RSP0/CPU0:router (config-if)# no shutdown` `RP/0/0RP0RSP0/CPU0:router (config-if)# commit` `RP/0/0RP0RSP0/CPU0:router (config-if)# exit` `RP/0/0RP0RSP0/CPU0:router (config)# exit`	接続の他端でインターフェイスを始動するために、ステップ 1 ～ 8 を繰り返します。 **（注）** シリアル接続の両端で設定が一致している必要があります。

	コマンドまたはアクション	目的
ステップ 10	**show ipv4 interface brief** **例：** `RP/0/0RP0RSP0/CPU0:router # show ipv4 interface brief`	インターフェイスがアクティブであり、適切に設定されていることを確認します。 シリアル インターフェイスが適切に始動されていれば、**show ipv4 interface brief** コマンドの出力の、そのインターフェイスの [Status] フィールドに [Up] と表示されます。
ステップ 11	**show interfaces serial** *interface-path-id* **例：** `RP/0/0RP0RSP0/CPU0:router# show interfaces serial 0/1/0/0`	（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。

次の作業

始動したシリアル インターフェイスのデフォルト設定を変更するには、「オプションのシリアル インターフェイス パラメータの設定」（P.559）を参照してください。

オプションのシリアル インターフェイス パラメータの設定

ここでは、シリアル インターフェイスのデフォルト設定の変更に使用するコマンドについて説明します。

前提条件

シリアル インターフェイスのデフォルト設定を変更する前に、シリアル インターフェイスを始動して、「シリアル インターフェイスの始動」（P.555）で説明するように shutdown 設定を削除することをお勧めします。

制約事項

シリアル インターフェイスがアクティブになるためには、シリアル接続の両端の設定が一致している必要があります。

手順の概要

1. **configure**
2. **interface serial** *interface-path-id*
3. **encapsulation** [**hdlc** | **ppp** | **frame-relay** [**IETF**]
4. **serial**
5. **crc** *length*
6. **invert**
7. **scramble**
8. **transmit-delay** *hdlc-flags*

9. **end**
   または
   commit
10. **exit**
11. **exit**
12. **exit**
13. **show interfaces serial** [*interface-path-id*]

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/0RP0RSP0/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 serial** *interface-path-id* **例：** `RP/0/0RP0RSP0/CPU0:router(config)# interface serial 0/1/0/0`	シリアル インターフェイス名と *rack/slot/module/port* 表記を指定して、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。
ステップ 3	**encapsulation** [**hdlc** | **ppp** | **frame-relay** [**IETF**] **例：** `RP/0/0RP0RSP0/CPU0:router(config-if)# encapsulation hdlc`	（任意）HDLC や PPP、フレームリレーなどのインターフェイス カプセル化パラメータおよび詳細を設定します。 **（注）** デフォルトのカプセル化は **hdlc** です。
ステップ 4	**serial** **例：** `RP/0/0RP0RSP0/CPU0:router(config-if)# serial`	（任意）シリアル サブモードを開始し、シリアル パラメータを設定します。
ステップ 5	**crc** *length* **例：** `RP/0/0RP0RSP0/CPU0:ios(config-if-serial)# crc 32`	（任意）インターフェイスの巡回冗長検査（CRC）の長さを指定します。16 ビットの CRC モードを指定するには **16** キーワード、**32** ビットの CRC モードを指定するには 32 キーワードを入力します。 **（注）** デフォルトの CRC の長さは 16 です。
ステップ 6	**invert** **例：** `RP/0/0RP0RSP0/CPU0:ios(config-if-serial)# inverts`	（任意）データ ストリームを反転します。
ステップ 7	**scramble** **例：** `RP/0/0RP0RSP0/CPU0:ios(config-if-serial)# scramble`	（任意）インターフェイス上でペイロード スクランブリングをイネーブルにします。 **（注）** インターフェイス上のペイロード スクランブリングはディセーブルです。

	コマンドまたはアクション	目的
ステップ 8	**transmit-delay** *hdlc-flags* **例：** `RP/0/0RP0RSP0/CPU0:ios(config-if-serial)# transmit-delay 10`	（任意）インターフェイス上の送信遅延を指定します。指定できる値は 0 ～ 128 です。 **（注）** 送信遅延はデフォルトでディセーブルです（送信遅延は **0** に設定されます）。
ステップ 9	**end** または **commit** **例：** `RP/0/0RP0RSP0/CPU0:router (config-if)# end` または `RP/0/0RP0RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 10	**exit** **例：** `RP/0/0RP0RSP0/CPU0:router(config-if-serial)# exit`	シリアル コンフィギュレーション モードを終了します。
ステップ 11	**exit** **例：** `RP/0/0RP0RSP0/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、グローバル コンフィギュレーション モードに入ります。
ステップ 12	**exit** **例：** `RP/0/0RP0RSP0/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、EXEC モードを開始します。
ステップ 13	**show interfaces serial** [*interface-path-id*] **例：** `RP/0/0RP0RSP0/CPU0:router# show interface serial 0/1/0/0`	（任意）指定したシリアル インターフェイスの一般情報を表示します。

次の作業

- 始動したシリアル インターフェイス上に PVC を持つポイントツーポイント フレームリレー サブインターフェイスを作成するには、「PVC を持つポイントツーポイント シリアル サブインターフェイスの作成」（P.562）を参照してください。
- PPP カプセル化がイネーブルであるシリアル インターフェイスに PPP 認証を設定するには、このマニュアルで後述する *「Cisco ASR 9000 シリーズ ルータ での PPP の設定」* モジュールを参照してください。
- デフォルトのキープアライブ設定を変更するには、「シリアル インターフェイスでのキープアライブ インターバルの変更」（P.567）を参照してください。
- フレーム リレー カプセル化がイネーブルであるシリアル インターフェイスのデフォルトのフレーム リレー設定を変更するには、*「Cisco ASR 9000 シリーズ ルータでのフレーム リレーの設定」* モジュールの「インターフェイスでのデフォルト フレームリレー設定の変更」セクションを参照してください。

PVC を持つポイントツーポイント シリアル サブインターフェイスの作成

ここに記載する手順では、ポイントツーポイント シリアル サブインターフェイスを作成し、そのシリアル サブインターフェイスに PVC を設定します。

（注） サブインターフェイスおよび PVC の作成は、フレームリレー カプセル化だけが設定されたインターフェイスでサポートされます。

前提条件

シリアル インターフェイスでサブインターフェイスを作成する前に、「シリアル インターフェイスの始動」（P.555）で説明するように、フレームリレー カプセル化が設定されたメイン シリアル インターフェイスを始動する必要があります。

制約事項

PVC は、各ポイントツーポイント シリアル サブインターフェイスに 1 つだけ設定できます。

手順の概要

1. **configure**
2. **interface serial** *interface-path-id.subinterface* **point-to-point**
3. **ipv4 address** *ipv4_address/prefix*
4. **pvc** *dlci*
5. **end**
   または
   commit
6. 接続の他端でシリアル サブインターフェイスおよび関連付けられている PVC を始動するために、ステップ 1 ～ 5 を繰り返します。

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/0RP0RSP0/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 serial** *interface-path-id.subinterface* **point-to-point** **例：** `RP/0/0RP0RSP0/CPU0:router (config)# interface serial 0/1/0/0.1`	シリアル 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	**ipv4 address** *ipv4_address/prefix* **例：** `RP/0/0RP0RSP0/CPU0:router (config-subif)#ipv4 address 10.46.8.6/24`	IP アドレスおよびサブネット マスクをサブインターフェイスに割り当てます。
ステップ 4	**pvc** *dlci* **例：** `RP/0/0RP0RSP0/CPU0:router (config-subif)# pvc 20`	シリアル PVC を作成し、フレームリレー PVC コンフィギュレーション サブモードを開始します。 *dlci* を 16 から 1007 の範囲の PVC ID に置き換えます。 **（注）** 各サブインターフェイスに設定できる PVC は 1 つだけです。

	コマンドまたはアクション	目的
ステップ 5	**end** または **commit** **例：** `RP/0/0RP0RSP0/CPU0:router (config-subif)# end` または `RP/0/0RP0RSP0/CPU0:router(config-subif)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 6	**configure** **interface serial** *interface-path-id* **pvc** *dlci* **commit** **例：** `RP/0/0RP0RSP0/CPU0:router# configure` `RP/0/0RP0RSP0/CPU0:router (config)# interface serial 0/1/0/1.1` `RP/0/0RP0RSP0/CPU0:router (config-subif)#ipv4 address 10.46.8.5/24` `RP/0/0RP0RSP0/CPU0:router (config-subif)# pvc 20` `RP/0/0RP0RSP0/CPU0:router (config-fr-vc)# commit`	接続の他端でシリアル サブインターフェイスおよび関連付けられている PVC を始動するために、ステップ 1 ～ 5 を繰り返します。 **（注）** DLCI（PVC ID）は、サブインターフェイス接続の両端で一致している必要があります。 **（注）** 接続の他端のサブインターフェイスに IP アドレスおよびサブネット マスクを割り当てるときには、接続の両端のアドレスが同じサブネットに属している必要があることに注意してください。

次の作業

- オプションの PVC パラメータを設定するには、「オプションのシリアル インターフェイス パラメータの設定」（P.559）を参照してください。
- フレーム リレー カプセル化がイネーブルであるシリアル インターフェイスのデフォルトのフレーム リレー設定を変更するには、*「Cisco ASR 9000 シリーズ ルータでのフレーム リレーの設定」*モジュールの「インターフェイスでのデフォルト フレームリレー設定の変更」セクションを参照してください。
- レイヤ 3 QOS サービス ポリシーを PVC サブモードの PVC に付加するには、該当する Cisco IOS XR ソフトウェアのコンフィギュレーション ガイドを参照してください。

オプションの PVC パラメータの設定

ここでは、シリアル PVC でのデフォルト設定の変更に使用できるコマンドについて説明します。

フレーム リレーのオプションに関する追加情報については、『*Cisco IOS XR Interface and Hardware Component Configuration Guide for the Cisco XR 12000 Series Router*』の「Configuring Frame Relay on Cisco IOS XR Software」モジュールを参照してください。

フレーム リレーのオプションに関する追加情報については、『*Cisco IOS XR Interface and Hardware Component Configuration Guide for the Cisco ASR 9000 Series Router*』の「Configuring Frame Relay on the Cisco ASR 9000 Series Router」モジュールを参照してください。

前提条件

PVC のデフォルト設定を変更する前に、「PVC を持つポイントツーポイント シリアル サブインターフェイスの作成」（P.562）で説明するようにシリアル サブインターフェイスで PVC を作成する必要があります。

制約事項

- 接続がアクティブになるためには、DLCI（PVI ID）が PVC の両端で一致している必要があります。
- PVC DLCI を変更するには、PVC を削除し、新しい DLCI を設定して PVC を追加し直す必要があります。

手順の概要

1. **configure**
2. **interface serial** *interface-path-id.subinterface*
3. **pvc** *dlci*
4. **encap** [**cisco** | **ietf**]
5. **service-policy** {**input** | **output**} *policy-map*
6. **end**
   または
   commit
7. 接続の他端で PVC を設定するために、ステップ 1 ～ 6 を繰り返します。
8. **show frame-relay pvc** *dlci-number*
9. **show policy-map interface serial** *interface-path-id.subinterface* {**input** | **output**}
   または
   show policy-map type qos interface serial *interface-path-id.subinterface* {**input** | **output**}

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/0RP0RSP0/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 serial** *interface-path-id.subinterface* **例：** `RP/0/0RP0RSP0/CPU0:router (config)# interface serial 0/1/0/0.1`	シリアル 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	**pvc** *dlci* **例：** `RP/0/0RP0RSP0/CPU0:router (config-subif)# pvc 20`	PVC に対する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 4	**encap** [**cisco** | **ietf**] **例：** `RP/0/0RP0RSP0/CPU0:router (config-fr-vc)# encap ietf`	（任意）フレームリレー PVC のカプセル化を設定します。 **（注）** PVC にカプセル化のタイプが明示的に設定されていない場合、その PVC は、メイン シリアル インターフェイスからカプセル化のタイプを引き継ぎます。
ステップ 5	**service-policy** {**input** | **output**} *policy-map* **例：** `RP/0/0RP0RSP0/CPU0:router (config-fr-vc)# service-policy output policy1`	ポリシー マップを入力サブインターフェイスまたは出力サブインターフェイスに付加します。付加すると、そのサブインターフェイスのサービス ポリシーとしてポリシー マップが使用されます。
ステップ 6	**end** または **commit** **例：** `RP/0/0RP0RSP0/CPU0:router (config-fr-vc)# end` または `RP/0/0RP0RSP0/CPU0:router(config-fr-vc)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

	コマンドまたはアクション	目的
ステップ 7	**configure** **interface serial** *interface-path-id.subinterface* **pvc** *dlci* **encap** [**cisco** | **ietf**] **commit** **例：** RP/0/0RP0RSP0/CPU0:router# configure RP/0/0RP0RSP0/CPU0:router (config)# interface serial 0/1/0/1.1 RP/0/0RP0RSP0/CPU0:router (config-subif)# pvc 20 RP/0/0RP0RSP0/CPU0:router (config-fr-vc)# encap cisco RP/0/0RP0RSP0/CPU0:router (config-fr-vc)# commit	接続の他端でシリアル サブインターフェイスおよび関連付けられている PVC を始動するために、ステップ 1 ～ 6 を繰り返します。 **（注）** サブインターフェイス接続の両端で設定が一致している必要があります。
ステップ 8	**show frame-relay pvc** *dlci-number* **例：** RP/0/0RP0RSP0/CPU0:router# show frame-relay pvc 20	（任意）指定したシリアル インターフェイスの設定を検証します。
ステップ 9	**show policy-map interface serial** *interface-path-id.subinterface* {**input** | **output**} または **show policy-map type qos interface serial** *interface-path-id.subinterface* {**input** | **output**} **例：** RP/0/0RP0RSP0/CPU0:router# show policy-map interface serial 0/1/0/0.1 output または RP/0/0RP0RSP0/CPU0:router# show policy-map type qos interface serial 0/1/0/0.1 output	（任意）サブインターフェイスに付加された入力ポリシーおよび出力ポリシーの統計情報と設定を表示します。

次の作業

- フレーム リレー カプセル化がイネーブルであるシリアル インターフェイスのデフォルトのフレーム リレー設定を変更するには、*「Cisco ASR 9000 シリーズ ルータでのフレーム リレーの設定」*モジュール（このマニュアル）の「インターフェイスでのデフォルト フレームリレー設定の変更」を参照してください。

シリアル インターフェイスでのキープアライブ インターバルの変更

Cisco HDLC カプセル化または PPP カプセル化がイネーブルであるシリアル インターフェイスのキープアライブ インターバルを変更するには、次の作業を行います。

（注） シリアル インターフェイスで Cisco HDLC カプセル化または PPP カプセル化をイネーブルした場合、キープアライブ インターバルはデフォルトで 10 秒に設定されます。デフォルトのキープアライブ インターバルを変更する手順は、次のとおりです。

（注） Cisco HDLC は、シリアル インターフェイスにおいてデフォルトでイネーブルになります。

前提条件

キープアライブ タイマーの設定を変更する前に、Cisco HDLC カプセル化または PPP カプセル化がインターフェイスでイネーブルになっていることを確認します。インターフェイスで Cisco HDLC カプセル化または PPP カプセル化をイネーブルにするには、「オプションのシリアル インターフェイス パラメータの設定」（P.559）で説明するように **encapsulation** コマンドを使用します。

制約事項

- Minimal Disruptive Restart（MDR）のアップグレードを実行する前に、Cisco XR 12000 シリーズ ルータでのキープアライブを無効にすることを推奨します。
- Minimal Disruptive Restart（MDR）のアップグレードを実行する前に、Cisco CRS-1 ルータでのキープアライブ インターバルを 10 秒以上に設定することを推奨します。

手順の概要

1. **configure**
2. **interface serial** *interface-path-id*
3. **keepalive** {*seconds* | **disable**}
   または
   no keepalive
4. **end**
   または
   commit
5. **show interfaces** *type interface-path-id*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/0RP0RSP0/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 serial** *interface-path-id* **例：** `RP/0/0RP0RSP0/CPU0:router(config)# interface serial 0/1/0/0`	シリアル インターフェイス名と *rack/slot/module/port* 表記を指定して、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。

	コマンドまたはアクション	目的
ステップ 3	**keepalive** {*seconds* | **disable**} または **no keepalive** **例：** `RP/0/0RP0RSP0/CPU0:router(config-if)# keepalive 3` または `RP/0/0RP0RSP0/CPU0:router(config-if)# no keepalive`	キープアライブ メッセージの間隔を秒数で指定します。 • キープアライブ機能をディセーブルにするには、**keepalive disable** コマンド、**no keepalive**、または **keepalive** コマンドを引数 0 で使用します。 • 範囲は 1 ～ 30 です。デフォルトは 10 秒です。 • キープアライブがインターフェイスに設定されている場合、そのインターフェイスでフレーム リレー カプセル化を設定する前に、**no keepalive** コマンドを使用してキープアライブ機能をディセーブルにします。
ステップ 4	**end** または **commit** **例：** `RP/0/0RP0RSP0/CPU0:router(config-if)# end` または `RP/0/0RP0RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	**show interfaces serial** *interface-path-id* **例：** `RP/0/0RP0RSP0/CPU0:router# show interfaces serial 0/1/0/0`	（任意）インターフェイスの設定を確認します。

レイヤ 2 接続回線（AC）の設定方法

レイヤ 2 接続回路（AC）の設定作業について、次の手順で説明します。

- PVC を持つシリアル レイヤ 2 サブインターフェイスの作成
- オプションのシリアル レイヤ 2 PVC パラメータの設定

（注） レイヤ 2 スイッチングのためのインターフェイスの設定後は、**ipv4 address** などのルーティング コマンドは使用できません。インターフェイスにルーティング コマンドを設定すると、**l2transport** コマンドが拒否されます。

PVC を持つシリアル レイヤ 2 サブインターフェイスの作成

ここに記載する手順では、PVC を持つレイヤ 2 サブインターフェイスを作成します。

前提条件

シリアル インターフェイスでサブインターフェイスを作成する前に、「シリアル インターフェイスの始動」（P.555）で説明するようにシリアル インターフェイスを始動する必要があります。

制約事項

各シリアル サブインターフェイスで設定できる PVC は 1 つだけです。

手順の概要

1. **configure**
2. **interface serial** *interface-path-id.subinterface* **l2transport**
3. **pvc** *vpi/vci*
4. **end**
   または
   commit
5. AC の他端でシリアル サブインターフェイスおよび関連付けられている PVC を始動するために、ステップ 1 ～ 4 を繰り返します。

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/0RP0/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 serial** *interface-path-id.subinterface* **l2transport** **例：** `RP/0/0RP0/CPU0:router(config)# interface serial 0/1/0/0.1 l2transport`	サブインターフェイスを作成して、そのサブインターフェイスに対するシリアル 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。

	コマンドまたはアクション	目的
ステップ 3	**pvc** *vpi/vci* **例：** `RP/0/0RP0/CPU0:router(config-if)# pvc 5/20`	シリアル PVC を作成して、シリアル レイヤ 2 転送 PVC コンフィギュレーション モードを開始します。 **（注）** 各サブインターフェイスに設定できる PVC は 1 つだけです。
ステップ 4	**end** または **commit** **例：** `RP/0/0RP0/CPU0:router(config-fr-vc)# end` または `RP/0/0RP0/CPU0:router(config-fr-vc)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	AC の他端でシリアル サブインターフェイスおよび関連付けられている PVC を始動するために、ステップ 1 ～ 4 を繰り返します。	AC を始動します。 **（注）** AC の両端で設定が一致している必要があります。

次の作業

- オプションの PVC パラメータを設定するには、「オプションのシリアル レイヤ 2 PVC パラメータの設定」（P.572）を参照してください。
- ネットワークでの L2TPv3 の設定に関する詳細については、『*Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco CRS Router* 』の「*Implementing Layer 2 Tunnel Protocol Version 3*」モジュールを参照してください。L2VPN の設定の詳細については、『*Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco CRS Router*』の「*Implementing MPLS Layer 2 VPNs*」モジュールを参照してください。
- ネットワークでの L2TPv3 の設定に関する詳細については、『*Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco XR 12000 Series Router* 』の「*Implementing Layer 2 Tunnel Protocol Version 3 on Cisco IOS XR Software*」モジュールを参照してください。L2VPN の設定の詳細については、『*Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco XR 12000 Series Router*』の「*Implementing MPLS Layer 2 VPNs on Cisco IOS XR Software*」モジュールを参照してください。

オプションのシリアル レイヤ 2 PVC パラメータの設定

ここでは、シリアル レイヤ 2 PVC でのデフォルト設定の変更に使用できるコマンドについて説明します。

前提条件

PVC のデフォルト設定を変更する前に、「PVC を持つシリアル レイヤ 2 サブインターフェイスの作成」（P.570）で説明するようにレイヤ 2 サブインターフェイスで PVC を作成する必要があります。

制約事項

PVC の両端での設定が、アクティブにする接続に合っている必要があります。

手順の概要

1. **configure**
2. **interface serial** *interface-path-id.subinterface* **l2transport**
3. **pvc** *dlci*
4. **encap** [**cisco** | **ietf**]
5. **service-policy** {**input** | **output**} *policy-map*
6. **fragment end-to-end** *fragment-size*
7. fragment-counter
8. **end**
   または
   commit
9. AC の他端で PVC を設定するために、ステップ 1 ～ 7 を繰り返します。
10. **show policy-map interface serial** *interface-path-id.subinterface* {**input** | **output**}
    または
    show policy-map type qos interface serial *interface-path-id.subinterface* {**input** | **output**}

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/0RP0/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 serial** *interface-path-id.subinterface* **l2transport** **例：** `RP/0/0RP0/CPU0:router(config)# interface serial 0/1/0/0.1 l2transport`	レイヤ 2 シリアル サブインターフェイスに対するシリアル 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。

	コマンドまたはアクション	目的
ステップ 3	**pvc** *dlci* **例：** `RP/0/0RP0/CPU0:router(config-if)# pvc 100`	指定した PVC に対するシリアル フレームリレー PVC 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	**encap** {**cisco** | **ietf**} **例：** `RP/0/0RP0/CPU0:router(config-fr-vc)# encapsulation aal5`	フレームリレー PVC のカプセル化を設定します。
ステップ 5	**service-policy** {**input** | **output**} *policy-map* **例：** `RP/0/0RP0/CPU0:router (config-subif)# service-policy output policy1`	ポリシー マップを入力サブインターフェイスまたは出力サブインターフェイスに付加します。付加すると、そのサブインターフェイスのサービス ポリシーとしてポリシー マップが使用されます。
ステップ 6	**fragment end-to-end** *fragment-size* **例：** `RP/0/0/CPU0:router(config-fr-vc)# fragment end-to-end 100`	インターフェイスでフレームリレー フレームのフラグメンテーションをイネーブルにします。 *fragment-size* を、発信元フレームリレー フレームのペイロード バイト数に置き換えます。これが各フラグメントのバイト数になります。この数値には、元のフレームのフレームリレー ヘッダーは含まれません。 Cisco 8 ポート チャネライズド T1/E1 SPA では、有効な値は 128、256、および 512 です。
ステップ 7	**fragment-counter** **例：** `RP/0/0/CPU0:router(config-fr-vc)# fragment-counter`	フレーム リレー サブインターフェイスおよび PVC に対してフラグメンテーション カウンタをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 8	**end** または **commit** **例：** `RP/0/0RP0/CPU0:router(config-serial-l2transport-pvc)# end` または `RP/0/0RP0/CPU0:router(config-serial-l2transport-pvc)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 9	AC の他端で PVC を設定するために、ステップ 1 ～ 7 を繰り返します。	AC を始動します。 **（注）** 接続の両端で設定が一致している必要があります。
ステップ 10	**show policy-map interface serial** *interface-path-id.subinterface* {**input** | **output**} または **show policy-map type qos interface serial** *interface-path-id.subinterface* {**input** | **output**} **例：** `RP/0/0RP0/CPU0:router# show policy-map interface pos 0/1/0/0.1 output` または `RP/0/0RP0/CPU0:router# show policy-map type qos interface pos 0/1/0/0.1 output`	（任意）サブインターフェイスに付加された入力ポリシーおよび出力ポリシーの統計情報と設定を表示します。

次の作業

- 作成した AC に対してポイントツーポイント疑似配線相互接続を設定するには、『*Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco CRS Router*』の「*Implementing Layer 2 Tunnel Protocol Version 3*」モジュールを参照してください。
- L2VPN を設定するには、『*Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco CRS Router*』の「*Implementing MPLS Layer 2 VPNs*」モジュールを参照してください。
- 作成した AC に対してポイントツーポイント疑似配線相互接続を設定するには、『*Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco XR 12000 Series Router*』の「*Implementing Layer 2 Tunnel Protocol Version 3 on Cisco IOS XR Software*」モジュールを参照してください。

- L2VPN を設定するには、『*Cisco IOS XR Virtual Private Network Configuration Guide for the Cisco XR 12000 Series Router*』の「*Implementing MPLS Layer 2 VPNs on Cisco IOS XR Software*」モジュールを参照してください。

IPHC 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。

- 「IPHC の設定の前提条件」（P.575）
- 「IPHC スロット レベル コマンドの設定」（P.576）
- 「IPHC プロファイルの設定」（P.577）
- 「IPHC プロファイルの設定」（P.580）
- 「インターフェイスでの IPHC プロファイルのイネーブル化」（P.582）

IPHC の設定の前提条件

IP ヘッダー圧縮（IPHC）は、次のカードでサポートされます。

- Cisco 1 ポート チャネライズド OC-12/STM-4
- Cisco 1 ポート チャネライズド OC-48/STM-16
- Cisco 1 ポート チャネライズド STM-1/OC-3
- Cisco 8 ポート チャネライズド T1/E1 SPA
- Cisco 2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA
- Cisco 2 ポートおよび 4 ポート チャネライズド T3 SPA
- Cisco マルチレート 10G IP サービス エンジン SIP
  - Cisco 12000-SIP-600
  - Cisco 12000-SIP-401
  - Cisco 12000-SIP-501
  - Cisco 12000-SIP-601
- SIP 700 SPA インターフェイス プロセッサ
- Cisco 2 ポート チャネライズド OC-12c/DS0 SPA
- Cisco 1 ポート チャネライズド OC-3/STM-1 SPA
- Cisco 4 ポート チャネライズド T3/DS0 SPA
- Cisco 8 ポート チャネライズド T1/E1 SPA
- Cisco 2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA

IPHC スロット レベル コマンドの設定

ここでは、IP ヘッダー圧縮（IPHC）スロット レベル コマンドの設定方法を説明します。このコマンドは、IPHC リソースを予約し、ラインカード上で IPHC をイネーブルにし、ノードに対する TCP および非 TCP 接続の最大数を定義するものです。この設定を行ってからでなければ、IPHC プロファイルを作成することはできません。

（注） IPHC スロット レベル設定は、両方のピア ルータで必要です。

手順の概要

IP ヘッダー圧縮（IPHC）スロット レベルを設定するには、次の手順を実行します。

1. **config**
2. **iphc tcp connections** *max-number* **location** *node-id*
3. **iphc non-tcp connections** *max-number* **location** *node-id*
4. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**config** **例：** `RP/0/0/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。

	コマンドまたはアクション	目的
ステップ 2	**iphc tcp connections** *max-number* **location** *node-id* **例：** `RP/0/0/CPU0:router(config)# iphc tcp connections 2000 location 0/1/cpu0`	1 つのラインカードにおいて IPHC 用に設定できる TCP 接続の最大数を設定します。 指定できる範囲は 1 ～ 2000 です。
ステップ 3	**iphc non-tcp connections** *max-number* **location** *node-id* **例：** `RP/0/0/CPU0:router(config)# iphc non-tcp connections 20000 location 0/1/cpu0`	1 つのラインカードにおいて IPHC 用に設定できる非 TCP 接続の最大数を設定します。 範囲は 1 ～ 20000 です。
ステップ 4	**end** または **commit** **例：** `RP/0/0/CPU0:router(config-if)# end` または `RP/0/0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

IPHC プロファイルの設定

ここでは、IP ヘッダー圧縮（IPHC）プロファイルを作成および設定する方法について説明します。この手順は、TCP および非 TCP 圧縮用です。

手順の概要

IP ヘッダー圧縮（IPHC）プロファイルを設定するには、次の手順を実行します。

1. **configure**
2. **iphc profile** *profile-name* **type** {**ietf** | **iphc**}
3. **tcp compression**
4. **tcp context absolute** *number-of-contexts*
5. **non-tcp compression**

6. **non-tcp context absolute** *number-of-contexts*
7. **rtp**
8. **refresh max-period** {*max-number* | **infinite**}
9. **refresh rtp**
10. **feedback disable**
11. **max-header** *number-of-bytes*
12. **end**
    または
    commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/0/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	**iphc profile** *profile-name* **type** {**ietf** | **iphc**} **例：** `RP/0/0/CPU0:router(config)# iphc profile Profile_1 type iphc`	IPHC プロファイルを作成し、圧縮形式タイプを設定します。IPHC プロファイル 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	**tcp compression** **例：** `RP/0/0/CPU0:router(config-iphc-profile)# tcp compression`	IPHC プロファイルの TCP 圧縮をイネーブルにします。
ステップ 4	**tcp context absolute** number-of-contexts **例：** `RP/0/0/CPU0:router(config-iphc-profile)# tcp context absolute 255`	ラインカード上で IPHC に使用できる TCP 接続コンテキストの最大数を設定します。
ステップ 5	**non-tcp compression** **例：** `RP/0/0/CPU0:router(config-iphc-profile)# non-tcp compression`	IPHC プロファイルの非 TCP 圧縮をイネーブルにします。
ステップ 6	**non-tcp context absolute** number-of-contexts **例：** `RP/0/0/CPU0:router(config-iphc-profile)# non-tcp context absolute 255`	ラインカード上で IPHC に使用できる非 TCP 接続コンテキストの最大数を設定します。

	コマンドまたはアクション	目的
ステップ 7	**rtp** **例：** `RP/0/0/CPU0:router(config-iphc-profile)# rtp`	インターフェイスに Real Time Protocol（RTP）プロトコルを設定します。
ステップ 8	**refresh max-period** {*max-number* | **infinite**} **例：** `RP/0/0/CPU0:router(config-iphc-profile)# refresh max-period 50`	IPHC コンテキストがリフレッシュされるまでに、リンクで交換される圧縮された IP ヘッダーの最大パケット数を設定します。
ステップ 9	**refresh rtp** **例：** `RP/0/0/CPU0:router(config-iphc-profile)# refresh rtp`	RTP パケットに対して設定されているコンテキストのリフレッシュ設定をイネーブルにします。
ステップ 10	**feedback disable** **例：** `RP/0/0/CPU0:router(config-iphc-profile)# feedback disable`	インターフェイスで IPHC コンテキスト ステータス フィードバック メッセージをディセーブルにします。
ステップ 11	**max-header** *number-of-bytes* **例：** `RP/0/0/CPU0:router(config-iphc-profile)# max-header 20`	圧縮された IP ヘッダーの最大サイズ（バイト単位）を設定します。
ステップ 12	**end** または **commit** **例：** `RP/0/0/CPU0:router(config-if)# end` または `RP/0/0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

IPHC プロファイルの設定

ここでは、IP ヘッダー圧縮（IPHC）プロファイルを作成および設定する方法について説明します。この手順は、TCP および非 TCP 圧縮用です。

手順の概要

IP ヘッダー圧縮（IPHC）プロファイルを設定するには、次の手順を実行します。

1. **configure**
2. **iphc profile** *profile-name* **type** {**cisco | ietf | iphc**}
3. **tcp compression**
4. **tcp context absolute** *number-of-contexts*
5. **non-tcp compression**
6. **non-tcp context absolute** *number-of-contexts*
7. **rtp**
8. **refresh max-period** {*max-number* | **infinite**}
9. **refresh max-time** {*max-time* | **infinite**}
10. **refresh rtp**
11. **feedback disable**
12. **max-header** *number-of-bytes*
13. **end**
    または
    commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**config** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**iphc profile** *profile-name* **type** {**cisco** | **ietf** | **iphc**} **例：** `RP/0/RSP0/CPU0:router(config)# iphc profile Profile_1 type iphc`	IPHC プロファイルを作成し、圧縮形式タイプを設定し、IPHC プロファイル 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	**tcp compression** **例：** `RP/0/RSP0/CPU0:router(config-iphc-profile)# tcp compression`	IPHC プロファイルの TCP 圧縮をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 4	**tcp context absolute** number-of-contexts **例：** RP/0/RSP0/CPU0:router(config-iphc-profile)# tcp context absolute 255	ラインカード上で IPHC に使用できる TCP 接続コンテキストの最大数を設定します。
ステップ 5	**non-tcp compression** **例：** RP/0/RSP0/CPU0:router(config-iphc-profile)# non-tcp compression	IPHC プロファイルの非 TCP 圧縮をイネーブルにします。
ステップ 6	**non-tcp context absolute** number-of-contexts **例：** RP/0/RSP0/CPU0:router(config-iphc-profile)# non-tcp context absolute 255	ラインカード上で IPHC に使用できる非 TCP 接続コンテキストの最大数を設定します。
ステップ 7	**rtp** **例：** RP/0/RSP0/CPU0:router(config-iphc-profile)# rtp	インターフェイスに Real Time Protocol（RTP）プロトコルを設定します。
ステップ 8	**refresh max-period** {*max-number* | **infinite**} **例：** RP/0/RSP0/CPU0:router(config-iphc-profile)# refresh max-period 50	IPHC コンテキストがリフレッシュされるまでに、リンクで交換される圧縮された IP ヘッダーの最大パケット数を設定します。
ステップ 9	**refresh max-time** {*max-time* | **infinite**} **例：** RP/0/RSP0/CPU0:router(config-iphc-profile)# refresh max-time 10	コンテキストのリフレッシュ間隔の最大時間を設定します。
ステップ 10	**refresh rtp** **例：** RP/0/RSP0/CPU0:router(config-iphc-profile)# refresh rtp	RTP パケットに対して設定されているコンテキストのリフレッシュ設定をイネーブルにします。
ステップ 11	**feedback disable** **例：** RP/0/RSP0/CPU0:router(config-iphc-profile)# feedback disable	インターフェイスで IPHC コンテキスト ステータス フィードバック メッセージをディセーブルにします。

	コマンドまたはアクション	目的
ステップ 12	**max-header** *number-of-bytes* **例：** `RP/0/RSP0/CPU0:router(config-iphc-profile)# max-header 20`	圧縮された IP ヘッダーの最大サイズ（バイト単位）を設定します。
ステップ 13	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

インターフェイスでの IPHC プロファイルのイネーブル化

ここでは、インターフェイスにプロファイルを直接割り当てることで、インターフェイス上で IP ヘッダー圧縮（IPHC）プロファイルをイネーブルにする方法について説明します。

手順の概要

IPHC プロファイルをインターフェイスに対してイネーブルにするように設定するには、次の手順を実行します。

1. **config**
2. **interface** *type interface-path-id*
3. **encapsulation ppp**
4. **ipv4 iphc profile** *profile-name* [**mode service-policy**]
5. **service policy input** | **output** | **type** *service-policy-name*
6. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**config** **例：** `RP/0/0RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface type** *interface-path-id* **例：** `RP/0/0RSP0/CPU0:router(config)# interface serial 0/1/0/1`	インターフェイスを指定します。 **（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。 ルータ構文の詳細については、疑問符（**?**）オンライン ヘルプ機能を使用します。
ステップ 3	**encapsulation** {**hdlc** | **ppp** | **frame-relay** | **mfr**} **例：** `RP/0/0RSP0/CPU0:router(config-if)# encapsulation ppp`	インターフェイスのレイヤ 2 カプセル化を指定します。
ステップ 4	**ipv4 iphc profile** *profile-name* [**mode service-policy**] **例：** `RP/0/0RSP0/CPU0:router(config-if)# ipv4 iphc profile Profile_1` または `RP/0/0RSP0/CPU0:router(config-if)# ipv4 iphc profile Profile_1 mode service-policy`	IPHC プロファイルをインターフェイスにアタッチします。 • *profile-name* : インターフェイスに割り当てる IPHC プロファイルのテキスト名です。 • **mode service-policy** : IPHC プロファイルが QoS サービス ポリシーにだけ適用されることを指定します。

	コマンドまたはアクション	目的
ステップ 5	**service policy output** *service-policy-name* **例：** `RP/0/0RSP0/CPU0:router(config-if)# service policy input | output | type service-policy-name`	（任意）IPHC プロファイルが適用される QoS サービス ポリシーの名前を指定します。出力サービス ポリシーだけが許可されています。 ステップ 2 で **mode service-policy** を指定した場合にだけ使用します。
ステップ 6	**end** または **commit** **例：** `RP/0/0RSP0/CPU0:router(config-if)# end` または `RP/0/0RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

シリアル インターフェイスの設定例

ここでは、次の設定例について説明します。

- 「シリアル インターフェイスの始動と Cisco HDLC カプセル化の設定：例」（P.584）
- 「シリアル インターフェイスでのフレームリレー カプセル化の設定：例」（P.585）
- 「シリアル インターフェイスでの PPP カプセル化の設定：例」（P.587）
- 「IPHC の設定：例」（P.587）

シリアル インターフェイスの始動と Cisco HDLC カプセル化の設定：例

次に、Cisco HDLC カプセル化を設定した基本的なシリアル インターフェイスの始動例を示します。

```
RP/0/0RP0RSP0/CPU0:Router#config
RP/0/0RP0RSP0/CPU0:Router(config)# interface serial 0/3/0/0/0:0
RP/0/0RP0RSP0/CPU0:Router(config-if)# ipv4 address 192.0.2.2 255.255.255.252
RP/0/0RP0RSP0/CPU0:Router(config-if)# no shutdown
RP/0/0RP0RSP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

次に、キープアライブ メッセージの間隔を 10 秒に設定する例を示します。

```
RP/0/0RP0RSP0/CPU0:router# configure
RP/0/0RP0RSP0/CPU0:router(config)# interface serial 0/3/0/0/0:0
RP/0/0RP0RSP0/CPU0:router(config-if)# keepalive 10
RP/0/0RP0RSP0/CPU0:router(config-if)# commit
```

次に、オプションのシリアル インターフェイス パラメータを変更する例を示します。

```
RP/0/0RP0RSP0/CPU0:router# configure
RP/0/0RP0RSP0/CPU0:router(config)# interface serial 0/3/0/0/0:0
RP/0/0RP0RSP0/CPU0:Router(config-if)# serial
RP/0/0RP0RSP0/CPU0:Router(config-if-serial)# crc 16
RP/0/0RP0RSP0/CPU0:Router(config-if-serial)# invert
RP/0/0RP0RSP0/CPU0:Router(config-if-serial)# scramble
RP/0/0RP0RSP0/CPU0:Router(config-if-serial)# transmit-delay 3
RP/0/0RP0RSP0/CPU0:Router(config-if-serial)# commit
```

次は、**show interfaces serial** コマンドの出力例です。

```
RP/0/0RP0RSP0/CPU0:Router# show interfaces serial 0/0/3/0/5:23
Serial0/0/3/0/5:23 is down, line protocol is down
  Hardware is Serial network interface(s)
  Internet address is Unknown
  MTU 1504 bytes, BW 64 Kbit
     reliability 143/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16,  loopback not set, keepalive set (10 sec)
  Last clearing of "show interface" counters 18:11:15
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     2764 packets input, 2816 bytes, 3046 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 broadcast packets, 0 multicast packets
              0 runts, 0 giants, 0 throttles, 0 parity
     3046 input errors, 1 CRC, 0 frame, 0 overrun, 2764 ignored, 281 abort
     2764 packets output, 60804 bytes, 0 total output drops
     Output 0 broadcast packets, 0 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

シリアル インターフェイスでのフレームリレー カプセル化の設定：例

次に、ルータ 1 上に、フレーム リレー カプセル化を設定した SPA 上および PVC を設定したシリアル サブインターフェイス上にシリアル インターフェイスを作成する例を示します。

```
RP/0/0RP0RSP0/CPU0:router# configure
RP/0/0RP0RSP0/CPU0:router(config)# interface serial 0/1/0/0
RP/0/0RP0RSP0/CPU0:router(config-if)# encapsulation frame-relay
RP/0/0RP0RSP0/CPU0:router(config-if)#frame-relay intf-type dce
RP/0/0RP0RSP0/CPU0:router(config-if)# no shutdown
RP/0/0RP0RSP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes

RP/0/0RP0RSP0/CPU0:router# configure
RP/0/0RP0RSP0/CPU0:router(config)# interface serial 0/1/0/0.1 point-to-point
RP/0/0RP0RSP0/CPU0:router (config-subif)#ipv4 address 10.20.3.1/24
RP/0/0RP0RSP0/CPU0:router (config-subif)# pvc 16
RP/0/0RP0RSP0/CPU0:router (config-fr-vc)# encapsulation ietf
RP/0/0RP0RSP0/CPU0:router (config-fr-vc)# commit
RP/0/0RP0RSP0/CPU0:router(config-fr-vc)# exit
RP/0/0RP0RSP0/CPU0:router(config-subif)# exit
```

```
RP/0/0RP0RSP0/CPU0:router(config)# exit

RP/0/0RP0RSP0/CPU0:router# show interface serial 0/1/0/0
Wed Oct  8 04:14:39.946 PST DST
Serial0/1/0/0 is up, line protocol is up
  Interface state transitions: 5
  Hardware is Serial network interface(s)
  Internet address is 10.20.3.1/24
  MTU 4474 bytes, BW 44210 Kbit
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation FRAME-RELAY, crc 16,
  Scrambling is disabled, Invert data is disabled
  LMI enq sent  0, LMI stat recvd 0, LMI upd recvd 0
  LMI enq recvd 880, LMI stat sent  880, LMI upd sent  0 , DCE LMI up
  LMI DLCI 1023  LMI type is CISCO  frame relay DCE
  Last clearing of "show interface" counters 02:23:04
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     858 packets input, 11154 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     858 packets output, 12226 bytes, 0 total output drops
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
```

次に、ルータ 1 に接続しているルータ 2 上に、フレーム リレー カプセル化を設定した SPA 上および PVC を設定したシリアル サブインターフェイス上にシリアル インターフェイスを作成する例を示します。

```
RP/0/0RP0RSP0/CPU0:router# configure
RP/0/0RP0RSP0/CPU0:router(config)# interface serial 0/1/0/1
RP/0/0RP0RSP0/CPU0:router(config-if)# encapsulation frame-relay
RP/0/0RP0RSP0/CPU0:router(config-if)# no shutdown
RP/0/0RP0RSP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes

RP/0/0RP0RSP0/CPU0:router# configure
RP/0/0RP0RSP0/CPU0:router(config)# interface serial 0/1/0/1.1 point-to-point
RP/0/0RP0RSP0/CPU0:router (config-subif)#ipv4 address 10.20.3.2/24
RP/0/0RP0RSP0/CPU0:router (config-subif)# pvc 16
RP/0/0RP0RSP0/CPU0:router (config-fr-vc)# encapsulation ietf
RP/0/0RP0RSP0/CPU0:router (config-fr-vc)# commit
```

```
RP/0/0RP0RSP0/CPU0:router(config-fr-vc)# exit
RP/0/0RP0RSP0/CPU0:router(config-subif)# exit
RP/0/0RP0RSP0/CPU0:router(config)# exit

RP/0/0RP0RSP0/CPU0:router# show interface serial 0/1/0/1
Wed Oct  8 04:13:45.046 PST DST
Serial0/1/0/1 is up, line protocol is up
  Interface state transitions: 7
  Hardware is Serial network interface(s)
  Internet address is Unknown
  MTU 4474 bytes, BW 44210 Kbit
     reliability 255/255, txload 0/255, rxload 0/255
  Encapsulation FRAME-RELAY, crc 16,
  Scrambling is disabled, Invert data is disabled
  LMI enq sent  1110, LMI stat recvd 875, LMI upd recvd 0, DTE LMI up
  LMI enq recvd 0, LMI stat sent  0, LMI upd sent  0
  LMI DLCI 1023  LMI type is CISCO  frame relay DTE
  Last clearing of "show interface" counters 02:22:09
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     853 packets input, 12153 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 0 runts, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     853 packets output, 11089 bytes, 0 total output drops
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
```

シリアル インターフェイスでの PPP カプセル化の設定：例

次に、シリアル インターフェイスを作成し、PPP カプセル化を設定する例を示します。

```
RP/0/0RP0RSP0/CPU0:router# configure
RP/0/0RP0RSP0/CPU0:router(config)# interface serial 0/3/0/0/0:0
RP/0/0RP0RSP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/0RP0RSP0/CPU0:router(config-if)# encapsulation ppp
RP/0/0RP0RSP0/CPU0:router(config-if)# no shutdown
RP/0/0RP0RSP0/CPU0:router(config-if)# ppp authentication chap MIS-access
RP/0/0RP0RSP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

次に、最初の認証が失敗した後に 2 回リトライできる（認証が失敗した場合に全部で 3 回リトライできる）ようにシリアル インターフェイス 0/3/0/0/0:0 を設定する例を示します。

```
RP/0/0RP0RSP0/CPU0:router# configuration
RP/0/0RP0RSP0/CPU0:router(config)# interface serial 0/3/0/0/0:0
RP/0/0RP0RSP0/CPU0:router(config-if)# encapsulation ppp
RP/0/0RP0RSP0/CPU0:router(config-if)# ppp authentication chap
RP/0/0RP0RSP0/CPU0:router(config-if)# ppp max-bad-auth 3
RP/0/0RP0RSP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

IPHC の設定：例

ここで紹介する例は、次のとおりです。

- 「IPHC プロファイルの設定：例」（P.588）
- 「シリアル インターフェイスでの IPHC の設定：例」（P.588）
- 「マルチリンクでの IPHC の設定：例」（P.588）

- 「MLPPP/LFI および QoS を使用するシリアル インターフェイスでの IPHC の設定：例」（P.589）

IPHC プロファイルの設定：例

次に、IPHC プロファイルを設定する例を示します。

```
config
  iphc tcp connections 6000 location 0/2/1
  iphc non-tcp connections 6000 location 0/2/1
  iphc profile Profile_1 type iphc
    tcp compression
    tcp context absolute 255
    non-tcp compression
    non-tcp context absolute 255
    rtp
    refresh max-period 50
    refresh max-time 10
    refresh rtp
    feedback disable
    max-header 20
commit
```

シリアル インターフェイスでの IPHC の設定：例

例 1

次に、インターフェイスにプロファイルを直接割り当てることによってシリアル インターフェイスで IP ヘッダー圧縮（IPHC）プロファイルをイネーブルにする例を示します。

```
config
  interface serial 0/1/0/1
    encapsulation ppp
    ipv4 iphc profile Profile_1
commit
```

例 2

次に、IPHC プロファイルを含む QoS サービス ポリシーを指定して、インターフェイスで IP ヘッダー圧縮（IPHC）プロファイルをイネーブルにする例を示します。

```
config
  interface serial 0/1/0/1:1
    encapsulation ppp
    ipv4 iphc profile Profile_2 mode service-policy
    service-policy output ip_header_compression_policy_map
commit
```

マルチリンクでの IPHC の設定：例

次に、マルチリンク インターフェイスで IP ヘッダー圧縮（IPHC）を設定する例を示します。

```
config
  interface multilink 0/4/3/0/4
    ipv4 address 10.10.10.10
    encapsulation ppp
    ipv4 iphc profile Profile_1
    commit
  interface serial 0/1/0/1:1
    encapsulation ppp
    multilink group 4
    commit
```

MLPPP/LFI および QoS を使用するシリアル インターフェイスでの IPHC の設定：例

次に、IPHC プロファイルを含む QoS サービス ポリシーを指定し、LFI を使用してシリアル インターフェイスに IP ヘッダー圧縮（IPHC）を設定する例を示します。

```
config
  interface multilink 0/4/3/0/4
    ipv4 address 10.10.10.10
      multilink
        fragment-size 128
        interleave
    ipv4 iphc profile Profile_2 mode service-policy
    service-policy output SP_2
    commit
  interface serial 0/1/0/1:2
    encapsulation ppp
    multilink group 4
    commit
```

その他の関連資料

ここでは、T3/E3 および T1/E1 コントローラおよびシリアル インターフェイスに関連する参考資料を示します。

関連資料

関連項目	参照先
Cisco IOS XR マスター コマンド リファレンス	*『Cisco IOS XR Master Commands List』*
Cisco IOS XR インターフェイス コンフィギュレーション コマンド	*『Cisco IOS XR Interface and Hardware Component Command Reference』*
Cisco IOS XR ソフトウェアを使用した初期システム ブートアップとルータの設定情報	*『Cisco IOS XR Getting Started Guide』*
Cisco IOS XR AAA サービス構成情報	『*Cisco IOS XR System Security Configuration Guide*』および *『Cisco IOS XR System Security Command Reference』*
リモートの Craft Works Interface（CWI）クライアント管理アプリケーションからの、Cisco CRS-1 ルータ上のインターフェイスとその他のコンポーネントの設定に関する情報	*『Cisco Craft Works Interface Configuration Guide』*

標準

標準	タイトル
FRF.1.2	*『PVC User-to-Network Interface (UNI) Implementation Agreement - July 2000』*
ANSI T1.617 Annex D	—
ITU Q.933 Annex A	—

MIB

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

RFC

RFC	タイトル
RFC 1294	*『Multiprotocol Interconnect Over Frame Relay』*
RFC 1315	*『Management Information Base for Frame Relay DTEs』*
RFC 1490	*『Multiprotocol Interconnect Over Frame Relay』*
RFC 1586	*『Guidelines for Running OSPF Over Frame Relay Networks』*
RFC 1604	*『Definitions of Managed Objects for Frame Relay Service』*
RFC 2115	*『Management Information Base for Frame Relay DTEs Using SMIv2』*
RFC 2390	*『Inverse Address Resolution Protocol』*
RFC 2427	*『Multiprotocol Interconnect Over Frame Relay』*
RFC 2954	*『Definitions of Managed Objects for Frame Relay Service』*

シスコのテクニカル サポート

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

Cisco ASR 9000 シリーズ ルータでのフレーム リレーの設定

このモジュールでは、フレーム リレー カプセル化を設定した Packet-over-SONET/SDH（POS）、マルチリンク、およびシリアル インターフェイスで使用できる、任意設定のフレーム リレー パラメータについて説明します。

Cisco IOS XR ソフトウェアのフレームリレー インターフェイス設定の機能履歴

リリース	変更内容
リリース 4.0.0	フレーム リレーのサポートは、次の SPA に対して追加されました。 • Cisco 2 ポート チャネライズド OC-12c/DS0 SPA • Cisco 1 ポート チャネライズド OC-48/STM-16 SPA • Cisco 8 ポート OC-12c/STM-4 POS SPA • Cisco 2 ポート OC-48c/STM-16 POS/RPR SPA • Cisco 1 ポート OC-192c/STM-64 POS/RPR XFP SPA 次のフレーム リレー機能のサポートが Cisco 2 ポート チャネライズド OC-12c/DSO SPA に対して追加されました。 • Multilink Frame Relay (FRF.16) • エンドツーエンド フラグメンテーション（FRF.12）
リリース 4.0.1	フレーム リレーのサポートは、次の SPA に対して追加されました。 • Cisco 1 ポート チャネライズド OC-3/STM-1 SPA • Cisco 2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA • Cisco 4 ポート OC-3c/STM-1 POS SPA • Cisco 8 ポート OC-3c/STM-1 POS SPA
リリース 4.1.0	フレーム リレーのサポートは、次の SPA に対して追加されました。 • Cisco 4 ポート チャネライズド T3/DS0 SPA • Cisco 8 ポート チャネライズド T1/E1 SPA

内容

フレームリレー設定の前提条件

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。

フレームリレーを設定する前に、次の条件を満たしていることを確認します。

- 使用しているハードウェアが POS インターフェイスまたはシリアル インターフェイスをサポートしている必要があります。
- 対応するモジュールの説明に従って、**encapsulation frame relay** コマンドを使用し、インターフェイスでフレームリレーのカプセル化をイネーブルにしました。
  - マルチリンク バンドル インターフェイスでフレーム リレー カプセル化をイネーブルにするには、「マルチリンク フレーム リレー バンドル インターフェイスの設定」（P.606）を参照してください。
  - POS インターフェイスでフレームリレーのカプセル化をイネーブルにするには、このマニュアルの *「Cisco ASR 9000 シリーズ ルータでの POS インターフェイスの設定」* モジュールを参照してください。
  - シリアル インターフェイスでフレームリレーのカプセル化をイネーブルにするには、このマニュアルの *「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」* モジュールを参照してください。

フレームリレー インターフェイスに関する情報

ここでは、フレームリレー インターフェイスを設定する際のさまざまな側面について説明します。

フレーム リレー カプセル化

Cisco ASR 9000 シリーズ ルータでは、フレーム リレーは、POS インターフェイス、シリアル メイン インターフェイス、およびそれらのインターフェイスで設定された PVC でサポートされます。フレームリレーのカプセル化をインターフェイスでイネーブルにするには、インターフェイス コンフィギュレーション モードで **encapsulation frame-relay** コマンドを使用します。

フレーム リレー インターフェイスは、次の 2 つのタイプのカプセル化フレームをサポートします。

- Cisco（これがデフォルト値です）
- IETF

PVC に Cisco または IETF カプセル化を設定するには、インターフェイス コンフィギュレーション モードで **encapsulation frame-relay** コマンドを使用します。

（注） **encapsulation** コマンドで PVC のカプセル化タイプが明示的に設定されていない場合、その PVC はメイン インターフェイスからカプセル化タイプを継承します。

encapsulation frame relay コマンドおよび **encap**（**PVC**）コマンドについては、次のモジュールを参照してください。

- POS インターフェイスでフレームリレーのカプセル化をイネーブルにするには、このマニュアルの *「Cisco ASR 9000 シリーズ ルータでの POS インターフェイスの設定」* モジュールを参照してください。
- シリアル インターフェイスでフレームリレーのカプセル化をイネーブルにするには、このマニュアルの *「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」* モジュールを参照してください。

フレームリレーのカプセル化でインターフェイスを設定し、その他の追加コンフィギュレーション コマンドを適用しない場合、表 23 のデフォルト インターフェイス設定が示されます。これらのデフォルト設定は、このモジュールの説明に従って設定で変更できます。

表 23　フレームリレーのカプセル化のデフォルト設定

パラメータ	設定ファイルのエントリ	デフォルト設定	コマンド モード
PVC Encapsulation	**encap** {**cisco** | **ietf**}	**cisco** （注） **encap** コマンドを設定しない場合、PVC のカプセル化タイプはフレームリレーのメイン インターフェイスから継承されます。	PVC コンフィギュレーション
Type of support provided by the interface	**frame-relay intf-type** {**dce** | **dte**}	**dte**	インターフェイス コンフィギュレーション

表 23　フレームリレーのカプセル化のデフォルト設定 （続き）

パラメータ	設定ファイルのエントリ	デフォルト設定	コマンド モード
LMI type supported on the interface	**frame-relay lmi-type** [**ansi** | **cisco** | **q933a**]	DCE の場合、デフォルト設定は **cisco** です。 DTE の場合、デフォルト設定は DCE でサポートされる LMI タイプに合わせて同期されます。 （注）　インターフェイスをデフォルトの LMI タイプに戻すには、**no frame-relay lmi-type** [**ansi** | **cisco** | **q933a**] コマンドを使用します。	インターフェイス コンフィギュレーション
Disable or enable LMI	**frame-relay lmi disable**	デフォルトでフレームリレー インターフェイスの LMI はイネーブルです。 LMI をディセーブルにした後で改めてイネーブルにするには、**no frame-relay lmi disable** コマンドを使用します。	インターフェイス コンフィギュレーション

（注）　LMI ポーリング関連のコマンドのデフォルト設定については、表 24（P.595）と表 25（P.595）を参照してください。

LMI

ローカル管理インターフェイス（LMI）プロトコルは、PVC の追加、削除、およびステータスをモニタリングします。また、フレームリレーのユーザネットワーク インターフェイス（UNI）を構成するリンクの完全性も検証します。

フレームリレー インターフェイスは、UNI で次のタイプの LMI をサポートします。

- ANSI-ANSI T1.617 Annex D
- Q.933-ITU-T Q.933 Annex A
- シスコ

インターフェイスで使用する LMI タイプを設定するには、**frame-relay lmi-type** コマンドを使用します。

（注）　使用する LMI タイプは、メイン インターフェイスに設定されている PVC と対応している必要があります。フレームリレー接続の両エンドの LMI タイプは一致する必要があります。

使用しているルータが別の非フレームリレー ルータに接続するスイッチとして機能する場合、**frame-relay intf-type dce** コマンドを使用して、データ通信機器（DCE）をサポートする LMI タイプを設定します。

ルータがフレームリレー ネットワークに接続している場合、**frame-relay intf-type dte** コマンドを使用して、データ端末装置（DTE）をサポートする LMI タイプを設定します。

（注） DTE インターフェイスでは、デフォルトで LMI タイプの自動検知がサポートされています。

システムのフレーム リレー インターフェイスの情報と統計情報を表示するには、**show frame-relay lmi** および **show frame-relay lmi-info** コマンドを EXEC モードで使用します。（*type* および *interface-path-id* 引数を指定するとき、メイン インターフェイスの情報を指定する必要があります）。エラーしきい値、イベント数とポーリング検証タイマーを変更し、フレーム リレー インターフェイスのモニタおよびトラブルシューティングを実行する際に役立つ情報を収集する **show frame-relay lmi** コマンドを使用できます。

LMI のタイプが **cisco** （デフォルトの LMI タイプ）である場合、1 つのインターフェイスでサポートできる PVC の最大数は、メイン インターフェイスの MTU サイズに関連しています。カードまたは SPA でサポートされる PVC の最大数を計算するには、次の公式を使用します。

```
(MTU - 13)/8 = PVC の最大数
```

cisco LMI で設定した POS PVC でサポートされる PVC のデフォルトの数は 557 です。また、**cisco** LMI で設定したシリアル PVC でサポートされる PVC のデフォルトの数は 186 です。

シスコ製ではない LMI タイプの場合、単一のメイン インターフェイスで最大 992 PVC がサポートされます。

（注） 特定の LMI タイプをインターフェイスに設定する場合、**no frame-relay lmi-type** [**ansi** | **cisco** | **q933a**] コマンドを使用して、インターフェイスをデフォルトの LMI タイプに戻します。

表 24 は、DCE 用に設定した PVC で LMI ポーリング オプションを変更するときに使用できるコマンドです。

表 24　DCE の LMI ポーリング コンフィギュレーション コマンド

パラメータ	設定ファイルのエントリ	デフォルト設定
Sets the error threshold on a DCE interface.	**lmi-n392dce** *threshold*	**3**
Sets the monitored event count.	**lmi-n393dce** *events*	**4**
Sets the polling verification timer on the DCE end.	**lmi-t392dce** *seconds*	**15**

表 25 は、DTE 用に設定した PVC で LMI ポーリング オプションを変更するときに使用できるコマンドです。

表 25　DTE の LMI ポーリング コンフィギュレーション コマンド

パラメータ	設定ファイルのエントリ	デフォルト設定
Set the number of Line Integrity Verification (LIV) exchanges performed before requesting a full status message.	**lmi-n391dte** *polling-cycles*	**6**
Sets the error threshold.	**lmi-n392dte** *threshold*	**3**
Sets the monitored event count.	**lmi-n393dte** *events*	**4**
Sets the polling interval (in seconds) between each status inquiry from the DTE end.	**frame-relay lmi-t391dte** *seconds*	**10**

Multilink Frame Relay (FRF.16)

マルチリンク フレーム リレー（MFR）は、次の共有ポート アダプタ（SPA）でのみサポートされます。

- Cisco 1 ポート チャネライズド STM-1/OC-3 SPA
- Cisco 2 ポート チャネライズド OC-12c/DSO SPA

マルチリンク フレーム リレー ハイ アベイラビリティ

MFR は、次のレベルのハイ アベイラビリティをサポートします。

- MFR は、プロセス再起動をサポートしていますが、一部の統計情報は、特定のプロセスの再起動時にリセットされます。
- MFR メンバー リンクは、ルート スイッチ プロセッサ（RSP）スイッチオーバー中も動作可能です。

マルチリンク フレーム リレーの設定の概要

マルチリンク フレームリレー インターフェイスは、インターフェイスでフレームリレーのカプセル化を可能にするマルチリンク バンドルの一部です。マルチリンク フレームリレー インターフェイスを作成するには、次のコンポーネントを設定します。

- MgmtMultilink コントローラ
- フレームリレーのカプセル化を可能にするマルチリンク バンドル インターフェイス
- バンドル ID 名
- マルチリンク フレームリレー サブインターフェイス
- バンドル インターフェイスの帯域幅クラス
- シリアル インターフェイス

MgmtMultilink コントローラ

次のコマンドを使用して、コントローラのマルチリンク バンドルを設定します。

controller MgmtMultilink *rack*/*slot*/*bay*/*controller-id*

bundle *bundleId*

この設定で、汎用マルチリンク バンドルのコントローラが作成されます。コントローラ ID 番号はコントローラ チップのゼロベース インデックスです。現在、マルチリンク フレームリレーをサポートする SPA には、1 ベイごとに 1 コントローラしかないため、コントローラの ID 番号は常にゼロ（0）です。

マルチリンク バンドル インターフェイス

マルチリンク バンドルを作成した後は、次のコマンドを使用して、フレームリレーのカプセル化を可能にするマルチリンク バンドル インターフェイスを作成します。

interface multilink *interface-path-id*

encapsulation frame-relay

この設定で、マルチリンク バンドル インターフェイスにマルチリンク フレームリレー サブインターフェイスを作成できます。

（注） マルチリンク バンドル インターフェイス上のカプセル化をフレームリレーに設定した後は、マルチリンク バンドルに関連付けられたメンバー リンクがインターフェイスにある場合、カプセル化は変更できません。

バンドル ID 名

（注） バンドル ID 名は、フレームリレー フォーラム 16.1（FRF 16.1）でのみ設定できます。

バンドル ID（**bid**）名は、インターフェイスの両エンドポイントのバンドル インターフェイスを識別します。バンドル ID 名は、一貫したリンクの割り当てを確保するために情報要素で交換されます。

デフォルトで、インターフェイス名（たとえば Multilink 0/4/1/0/1）がバンドル ID 名として使用されます。ただし、オプションで **frame-relay multilink bid** コマンドを使用して名前を作成することもできます。

（注） デフォルトの名前を使用するか、frame-relay multilink bid コマンドを使用して名前を作成するかにかかわらず、各バンドルに固有の名前を指定することを推奨します。

バンドル ID 名の長さは、ヌルの終端文字を含めて 50 文字までです。バンドル ID 名はバンドル インターフェイス レベルで設定され、各メンバー リンクに適用されます。

バンドル ID 名を設定するには、次のコマンドを使用します。

interface multilink *interface-path-id*

frame-relay multilink bid *bundle-id-name*

マルチリンク フレームリレー サブインターフェイス

マルチリンク フレームリレー サブインターフェイスを設定するには、次のコマンドを使用します。

interface multilink *interface-path-id*[*.subinterface* {**l2transport** | **point-to-point**}]

1 つのマルチリンク バンドル インターフェイスには最大 992 サブインターフェイスを設定できます。

（注） サブインターフェイス レベルで特定のフレームリレー インターフェイス機能を設定します。

マルチリンク フレームリレー サブインターフェイス機能

次のコマンドは、マルチリンク フレームリレー バンドル サブインターフェイスで特定の機能を設定するために使用できます。

- **mtu** *MTU size*
- **description**
- **shutdown**
- **bandwidth** *bandwidth*
- **service-policy** {**input** | **output**} *policymap-name*

（注） **service-policy** コマンドを入力すると、ポリシー マップをマルチリンク フレームリレー バンドル サブインターフェイスに付加できるようになりますが、フレームリレー PVC コンフィギュレーション モードで次の操作を実行する必要があります。詳細については、「マルチリンク フレーム リレー バンドル インターフェイスの設定」（P.606）を参照してください。

バンドル インターフェイスの帯域幅クラス

（注） 帯域幅クラスは、マルチリンク バンドル インターフェイスでのみ設定できます。

マルチリンク フレームリレー インターフェイスでは、3 タイプの帯域幅クラスのいずれかを設定できます。

- a — 帯域幅クラス A
- b — 帯域幅クラス B
- c — 帯域幅クラス C

帯域幅クラス A を設定し、1 つまたは複数のメンバー リンクがアップ（PH_ACTIVE）の場合、バンドル インターフェイスもアップで、BL_ACTIVATE がフレームリレー接続にシグナリングされます。すべてのメンバー リンクがダウンの場合、バンドル インターフェイスはダウンで、BL_DEACTIVATE がフレームリレー接続にシグナリングされます。

帯域幅クラス B を設定し、すべてのメンバー リンクがアップ（PH_ACTIVE）の場合、バンドル インターフェイスはアップで、BL_ACTIVATE がフレームリレー接続にシグナリングされます。いずれかのメンバー リンクがダウンの場合、バンドル インターフェイスはダウンで、BL_ACTIVATE がフレームリレー接続にシグナリングされます。

帯域幅クラス C を設定する場合、バンドルのリンクのしきい値を 1 ～ 255 に設定する必要があります。このしきい値は、バンドル インターフェイスをアップにするため、およびフレームリレー接続に BL_ACTIVATE をシグナリングするために必要な、アップ（PH_ACTIVE）にするリンクの最小数です。アップ状態のリンク数がこのしきい値未満の場合、バンドル インターフェイスがダウンになり、BL_DEACTIVATE がフレームリレー接続にシグナリングされます。しきい値に 1 を入力した場合の動作は、帯域幅クラス A と同じです。アップ状態のメンバー リンク数よりも多いしきい値を入力すると、バンドルはダウンのままです。

フレームリレー マルチリンク バンドル インターフェイスの帯域幅クラスを設定するには、次のコマンドを使用します。

interface multilink *interface-path-id*

frame-relay multilink bandwidth-class {**a** | **b** | **c** [*threshold*]}

デフォルトは a（帯域幅クラス A）です。

シリアル インターフェイス

T3 コントローラと T1 コントローラを設定した後は、マルチリンク フレームリレー バンドル サブインターフェイスにシリアル インターフェイスを追加できます。この場合、シリアル インターフェイスを設定し、マルチリンク フレームリレー（mfr）としてカプセル化し、それをバンドル インターフェイス（マルチリンク グループ番号によって指定されます）に割り当て、リンクの名前を設定します。MFR 確認応答タイムアウト値、再送信の再試行回数および hello 間隔もバンドル リンクに対して設定できます。

マルチリンク フレームリレー シリアル インターフェイスを設定するには、次のコマンドを使用します。

interface serial *rack/slot/module/port/t1-num:channel-group-number*

encapsulation mfr

multilink group *group number*

frame-relay multilink lid *link-id name*

frame-relay multilink ack *ack-timeout*

frame-relay multilink hello *hello-interval*

frame-relay multilink retry *retry-count*

（注） MFR バンドルのすべてのシリアル リンクは、マルチリンク インターフェイスの **mtu** コマンドの値を継承します。そのため、MFR バンドルのメンバーとして設定する前に、シリアル インターフェイスで **mtu** コマンドを設定しないでください。デフォルト以外の MTU 値を設定している場合、MFR バンドルのメンバーとしてシリアル インターフェイスを設定しようとすると、Cisco IOS XR ソフトウェアによってブロックされます。また、MFR バンドルのメンバーとして設定されているシリアル インターフェイスの **mtu** コマンド値を変更しようとした場合もブロックされます。

show コマンド

マルチリンク フレームリレー シリアル インターフェイスの設定を検証するには、次の **show** コマンドを使用します。

show frame-relay multilink location ***node id***

show frame-relay multilink interface serial ***interface-path-id*** **[detail | verbose]**

次の例は、**show frame-relay multilink location** コマンドの表示出力です。

```
RP/0/RSP0/CPU0:router# show frame-relay multilink  location  0/4/cpu0
Member interface: Serial0/4/2/0/9:0, ifhandle 0x05007b00
HW state = Up, link state = Up
Member of bundle interface Multilink0/4/2/0/2 with ifhandle 0x05007800

Bundle interface: Multilink0/4/2/0/2, ifhandle 0x05007800
    Member Links: 4 active, 0 inactive
    State = Up,   BW Class = C (threshold   3)
    Member Links:
    Serial0/4/2/0/12:0, HW state = Up, link state = Up
    Serial0/4/2/0/11:0, HW state = Up, link state = Up
    Serial0/4/2/0/10:0, HW state = Up, link state = Up
    Serial0/4/2/0/9:0, HW state = Up, link state = Up

Member interface: Serial0/4/2/0/10:0, ifhandle 0x05007c00
HW state = Up, link state = Up
Member of bundle interface Multilink0/4/2/0/2 with ifhandle 0x05007800

Member interface: Serial0/4/2/0/11:0, ifhandle 0x05007d00
HW state = Up, link state = Up
Member of bundle interface Multilink0/4/2/0/2 with ifhandle 0x05007800

Member interface: Serial0/4/2/0/12:0, ifhandle 0x05007e00
HW state = Up, link state = Up
Member of bundle interface Multilink0/4/2/0/2 with ifhandle 0x05007800
```

次の例は、コマンドの表示出力です。

```
RP/0/RSP0/CPU0:router# show frame-relay multilink interface serial 0/4/2/0/10:0

Member interface: Serial0/4/2/0/10:0, ifhandle 0x05007c00
HW state = Up, link state = Up
```

```
Member of bundle interface Multilink0/4/2/0/2 with ifhandle 0x05007800
```

エンドツーエンド フラグメンテーション（FRF.12）

データリンク接続識別子（DLCI）を使用して、FRF.12 エンドツーエンド フラグメンテーション接続を設定することができます。ただし、チャネライズド フレームリレー シリアル インターフェイスで設定する必要があります。

（注） **fragment end-to-end** コマンドは、POS インターフェイス、またはマルチリンク フレームリレー バンドル インターフェイスの DLCI では使用できません。

FRF.12 エンドツーエンド フラグメンテーションを DLCI 接続で設定するには、次のコマンドを使用します。

fragment end-to-end *fragment-size*

fragment-size 引数には、シリアル インターフェイスのフラグメント サイズをバイト単位で定義します。

（注） DLCI 接続では、優先順位の高いフラグメントと低いフラグメントのインターリービングが発生するように、パケットを優先順位の高低で分類する出力サービス ポリシーを設定することを強くお勧めします。

フレーム リレーの設定

次の項では、フレームリレー インターフェイスの設定方法について説明します。

- 「インターフェイスでのデフォルト フレームリレー設定の変更」（P.600）
- 「フレームリレーのカプセル化を設定したインターフェイスでの LMI のディセーブル」（P.603）
- 「マルチリンク フレーム リレー バンドル インターフェイスの設定」（P.606）
- 「チャネライズド フレームリレー シリアル インターフェイスでの FRF.12 エンドツーエンド フラグメンテーションの設定」（P.612）

インターフェイスでのデフォルト フレームリレー設定の変更

このタスクは、Packet-over-SONET/SDH（POS）、マルチリンク、またはシリアルのインターフェイスでフレームリレー カプセル化が設定されている場合にデフォルトのフレームリレー パラメータを変更するために実行します。

前提条件

デフォルトのフレームリレー設定を変更する前に、次のモジュールの説明に従ってインターフェイスでフレームリレーをイネーブルにする必要があります。

- POS インターフェイスでフレームリレーのカプセル化をイネーブルにするには、このマニュアルの *「Cisco ASR 9000 シリーズ ルータでの POS インターフェイスの設定」* モジュールを参照してください。

- シリアル インターフェイスでフレームリレーのカプセル化をイネーブルにするには、このマニュアルの*「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」*モジュールを参照してください。

（注） POS インターフェイスまたはシリアル インターフェイスでフレーム リレー カプセル化をイネーブルにする前に、そのインターフェイスに割り当て済みの IP アドレスがないことを確認します。IP アドレスが割り当て済みの場合、フレームリレーのカプセル化をイネーブルにすることはできません。フレームリレーの場合、IP アドレスとサブネット マスクはサブインターフェイスで設定します。

制約事項

- LMI タイプは、アクティブにする接続の両エンドで一致する必要があります。
- インターフェイスでフレームリレーのカプセル化を削除し、そのインターフェイスを PPP または HDLC のカプセル化で設定し直す前に、すべてのインターフェイス、サブインターフェイス、LMI、およびそのインターフェイスのフレームリレー設定を削除する必要があります。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **frame-relay intf-type** {**dce** | **dte**}
4. **frame-relay lmi-type** [**ansi** | **cisco** | **q933a**]
5. **encap** {**cisco** | **ietf**}
6. **end**
   または
   commit
7. **show interfaces** [**summary** | [*type interface-path-id*] [**brief** | **description** | **detail** | **accounting** [**rates**]]] [**location** *node-id*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**interface** *type interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface pos 0/4/0/1`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**frame-relay intf-type** {**dce** | **dte**} **例：** `RP/0/RSP0/CPU0:router(config-if)# frame-relay intf-type dce`	インターフェイスで提供するサポートのタイプを設定します。 • 使用しているルータが別のルータに接続するスイッチとして機能する場合、**frame-relay intf-type dce** コマンドを使用して、DCE をサポートする LMI タイプを設定します。 • ルータがフレームリレー ネットワークに接続している場合、**frame-relay intf-type dte** コマンドを使用して、DTE をサポートする LMI タイプを設定します。 **（注）** デフォルトのインターフェイス タイプは DTE です。
ステップ 4	**frame-relay lmi-type** [**ansi** | **q933a** | **cisco**] **例：** `RP/0/RSP0/CPU0:router(config-if)# frame-relay lmi-type ansi`	インターフェイスでサポートする LMI タイプを選択します。 • ANSI T1.617a-1994 Annex D の定義に従って LMI を使用するには、**frame-relay lmi-type ansi** コマンドを入力します。 • Cisco の定義（標準ではありません）に従って LMI を使用するには、**frame-relay lmi-type cisco** コマンドを使用します。 • ITU-T Q.933 (02/2003) Annex A の定義に従って LMI を使用するには、**frame-relay lmi-type q933a** コマンドを使用します。 **（注）** デフォルトの LMI タイプは Cisco です。
ステップ 5	**encap** {**cisco** | **ietf**} **例：** `RP/0/RSP0/CPU0:router (config-fr-vc)# encap ietf`	フレームリレー PVC のカプセル化を設定します。 **（注）** PVC のカプセル化タイプを明示的に設定しない場合、その PVC はメイン インターフェイスからカプセル化タイプを継承します。

	コマンドまたはアクション	目的
ステップ 6	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 7	**show interfaces** [**summary** | [*type interface-path-id*] [**brief** | **description** | **detail** | **accounting** [**rates**]]] [**location** *node-id*] **例：** `RP/0/RSP0/CPU0:router# show interface pos 0/4/0/1`	（任意）指定したインターフェイスの設定を検証します。

フレームリレーのカプセル化を設定したインターフェイスでの LMI のディセーブル

フレームリレーのカプセル化が設定されたインターフェイスで LMI をディセーブルにするには、次のタスクを実行します。

（注） フレームリレーのカプセル化がイネーブルなインターフェイスでは、デフォルトで LMI がイネーブルです。インターフェイスの LMI をディセーブルにした後で改めてイネーブルにするには、インターフェイス コンフィギュレーション モードで **no frame-relay lmi disable** コマンドを使用します。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **frame-relay lmi disable**

4. **end**
   または
   commit
5. **show interfaces** [**summary** | [*type interface-path-id*] [**brief** | **description** | **detail** | **accounting** [**rates**]]] [**location** *node-id*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface POS 0/4/0/1`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**frame-relay lmi disable** **例：** `RP/0/RSP0/CPU0:router(config-if)# frame-relay lmi disable`	指定したインターフェイスで LMI をディセーブルにします。

	コマンドまたはアクション	目的
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	**show interfaces** [**summary** | [*type interface-path-id*] [**brief** | **description** | **detail | accounting** [**rates**]]] [**location** *node-id*] **例：** `RP/0/RSP0/CPU0:router# show interfaces POS 0/1/0/0`	（任意）指定したインターフェイスで LMI がディセーブルになっていることを確認します。

マルチリンク フレーム リレー バンドル インターフェイスの設定

次に示す手順は、マルチリンク フレームリレー（MFR）バンドル インターフェイスとそのサブインターフェイスを設定するときに実行します。

前提条件

MFR バンドルを設定する前に、次の SPA がインストールされていることを確認してください。

- 1 ポート チャネライズド STM-1/OC-3 SPA
- 2 ポート チャネライズド OC-12c/DS0 SPA

制約事項

- マルチリンク フレーム リレー バンドル インターフェイスのすべてのメンバ リンクは、同じタイプにする必要があります（たとえば、T1 または E1）。メンバー リンクは、ポイントツーポイントなど、同じフレーム構成タイプにし、同じ帯域幅クラスにする必要があります。
- すべてのメンバ リンクがフル T1 または E1 である必要があります。DS0 など、フラクショナル リンクはサポートされません。
- すべてのメンバ リンクは、同じ SPA 上にある必要があります。そうでなければ、関連しないバンドルと見なされます。
- すべてのメンバ リンクは、遠端でも同じラインカードまたは SPA に接続している必要があります。
- サポートされる DLCI の範囲 16 ～ 1007 に基づいて、最大 992 の MFR サブインターフェイスが各メイン インターフェイスでサポートされます。
- Cisco 1 ポート チャネライズド OC-3/STM-1 SPA および 2 ポート チャネライズド OC-12c/DS0 SPA には次の追加の注意事項があります。
  - ラインカードごとに最大 700 の MFR バンドルがサポートされます。
  - システムごとに最大 2600 の MFR バンドルがサポートされます。
  - ラインカードごとに最大 4000 のフレーム リレー レイヤ 3 サブインターフェイスがサポートされます。
  - システムごとに最大 8000 のフレーム リレー レイヤ 3 サブインターフェイスがサポートされます。
- MLFR バンドルの一部であるフレーム リレー サブインターフェイスでのフラグメンテーションはサポートされません。
- MFR バンドルのすべてのシリアル リンクは、マルチリンク インターフェイスの **mtu** コマンドの値を継承します。そのため、MFR バンドルのメンバーとして設定する前に、シリアル インターフェイスで **mtu** コマンドを設定しないでください。Cisco IOS XR ソフトウェアは次の処理をブロックします。
  - インターフェイスがデフォルト以外の MTU 値で設定されている場合、MFR バンドルのメンバーとしてシリアル インターフェイスを設定しようとする処理。
  - MFR バンドルのメンバーとして設定されているシリアル インターフェイスの **mtu** コマンド値を変更しようとする処理。

手順の概要

1. **configure**

2. **controller MgmtMultilink** *rack*/*slot*/*bay*/*controller-id*
3. **exit**
4. **controller t3** *interface-path-id*
5. **mode** *type*
6. **clock source** {**internal** | **line**}
7. **exit**
8. **controller** {**t1** | **e1**} *interface-path-id*
9. **channel-group** *channel-group-number*
10. **timeslots** *range*
11. **exit**
12. **exit**
13. **interface multilink** *interface-path-id*[*.subinterface* {**l2transport** | **point-to-point**}]
14. **encapsulation frame-relay**
15. **frame-relay multilink bid** *bundle-id-name*
16. **frame-relay multilink bandwidth-class** {**a** | **b** | **c** [*threshold*]}
17. **exit**
18. **interface multilink** *interface-path-id*[*.subinterface* {**l2transport** | **point-to-point**}]
19. **ipv4 address** *ip-address*
20. **pvc** *dlci*
21. **service-policy** {**input** | **output**} *policy-map*
22. **exit**
23. **exit**
24. **interface serial** *interface-path-id*
25. **encapsulation mfr**
26. **multilink group** *group-id*
27. **frame-relay multilink lid** *link-id name*
28. **frame-relay multilink ack** *ack-timeout*
29. **frame-relay multilink hello** *hello-interval*
30. **frame-relay multilink retry** *retry-count*
31. **exit**
32. **end**
    または
    commit
33. **exit**
34. **show frame-relay multilink interface** *type interface-path-id* [**detail** | **verbose**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# config`	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	**controller MgmtMultilink** *rack/slot/bay/controller-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller MgmtMultilink 0/1/0/0`	*rack/slot/bay/controller-id* 表記で汎用マルチリンク バンドルのコントローラを作成し、マルチリンク管理コンフィギュレーション モードを開始します。コントローラ ID 番号はコントローラ チップのゼロ ベース インデックスです。現在、マルチリンク フレームリレーをサポートする SPA には、1 ベイごとに 1 コントローラしかないため、コントローラの ID 番号は常にゼロ（0）です。
ステップ 3	**exit** **例：** `RP/0/RSP0/CPU0:router(config-mgmtmultilink)# exit`	マルチリンク管理コンフィギュレーション モードを終了します。
ステップ 4	**controller t3** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t3 0/1/0/0`	*rack/slot/module/port* 表記で T3 コントローラ名を指定し、T3 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	**mode** *type* **例：** `RP/0/RSP0/CPU0:router(config-t3)# mode t1`	チャネライズするマルチリンクのタイプを設定します（たとえば、28 T1）。
ステップ 6	**clock source** {**internal** | **line**} **例：** `RP/0/RSP0/CPU0:router(config-t3)# clock source internal`	（任意）個々の E3 リンクのクロッキングを設定します。 **（注）** デフォルトのクロック ソースは **internal** です。 **（注）** シリアル リンクでクロッキングを設定する場合、一方のエンドを **internal** にし、もう一方を **line** にする必要があります。接続の両エンドに **internal** クロッキングを設定すると、フレーム同期のずれが生じます。接続の両エンドに **line** クロッキングを設定すると、ラインはアップ状態になりません。
ステップ 7	**exit** **例：** `RP/0/RSP0/CPU0:router(config-t3)# exit`	T3/E3 コントローラ コンフィギュレーション モードを終了します。

	コマンドまたはアクション	目的
ステップ 8	**controller** {**t1** | **e1**} *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t1 0/1/0/0/0`	T1 または E1 コンフィギュレーション モードを開始します。
ステップ 9	**channel-group** *channel-group-number* **例：** `RP/0/RSP0/CPU0:router(config-t1)# channel-group 0`	T1 チャネル グループを作成し、そのチャネル グループのチャネル 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 10	**timeslots** *range* **例：** `RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 1-24`	1 つまたは複数の DS0 タイムスロットをチャネル グループに関連付け、関連付けたシリアル サブインターフェイスをそのチャネル グループに作成します。 • T1 コントローラの場合、範囲は 1 ～ 24 タイムスロットです。 • E1 コントローラの場合、範囲は 1 ～ 31 タイムスロットです。 • すべてのタイムスロットを単一のチャネル グループに割り当てることも、タイムスロットを複数のチャネル グループに分割することもできます。
ステップ 11	**exit** **例：** `RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit`	チャネル グループ コンフィギュレーション モードを終了します。
ステップ 12	**exit** **例：** `RP/0/RSP0/CPU0:router(config-t1)# exit`	T1 コンフィギュレーション モードを終了します。
ステップ 13	**interface multilink** *interface-path-id*[*.subinterface* {**l2transport** | **point-to-point**}] **例：** `RP/0/RSP0/CPU0:router(config)# interface Multilink 0/1/0/0/100`	バンドルのフレームリレーのカプセル化を指定するマルチリンク バンドル インターフェイスを作成します。マルチリンク バンドル インターフェイスにマルチリンク フレームリレー サブインターフェイスを作成します。
ステップ 14	**encapsulation frame-relay** **例：** `Router(config-if)# encapsulation frame-relay`	フレームリレーのカプセル化タイプを指定します。
ステップ 15	**frame-relay multilink bid** *bundle-id-name* **例：** `Router(config-if)# frame-relay multilink bid MFRBundle`	**（注）** （任意）デフォルトで、インターフェイス名（たとえば Multilink 0/4/1/0/1）がバンドル ID 名として使用されます。ただし、オプションで **frame-relay multilink bid** コマンドを使用して名前を作成することもできます。

	コマンドまたはアクション	目的
ステップ 16	**frame-relay multilink bandwidth-class** {**a** | **b** | **c** [*threshold*]} **例：** `Router(config-if)# frame-relay multilink bandwidth-class a`	マルチリンク フレームリレー インターフェイスでは、3 タイプの帯域幅クラスのいずれかを設定します。 • a — 帯域幅クラス A • b — 帯域幅クラス B • c — 帯域幅クラス C デフォルトは **a**（帯域幅クラス A）です。
ステップ 17	**exit** **例：** `RP/0/RSP0/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 18	**interface multilink** *interface-path-id*[*.subinterface* {**l2transport** | **point-to-point**}] **例：** `RP/0/RSP0/CPU0:router(config)# interface Multilink 0/1/0/0/100.16 point-to-point`	*rack*/*slot*/*bay*/*controller-id bundleId.subinterace* [**point-to-point** | **l2transport**] 表記でマルチリンク サブインターフェイスを作成し、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。 • **l2transport** - 接続回線として扱います。 • **point-to-point** - ポイントツーポイント リンクとして扱います。 1 つのマルチリンク バンドル インターフェイスには最大 992 サブインターフェイスを設定できます。DLCI は 16 ～ 1007 です。
ステップ 19	**ipv4 address** *ip-address* **例：** `RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 3.1.100.16 255.255.255.0`	次の形式でインターフェイスに IP アドレスおよびサブネット マスクを割り当てます。 *A.B.C.D/prefix* または *A.B.C.D/mask*
ステップ 20	**pvc** *dlci* **例：** `RP/0/RSP0/CPU0:router (config-subif)# pvc 16`	POS 相手先固定接続（PVC）を作成し、フレームリレー PVC コンフィギュレーション サブモードを開始します。 *dlci* を 16 から 1007 の範囲の PVC ID に置き換えます。 **（注）** 各サブインターフェイスに設定できる PVC は 1 つだけです。
ステップ 21	**service-policy** {**input** | **output**} *policy-map* **例：** `RP/0/RSP0/CPU0:router(config-fr-vc)# service-policy output policy-mapA`	ポリシー マップを入力サブインターフェイスまたは出力サブインターフェイスに付加します。付加すると、そのサブインターフェイスのサービス ポリシーとしてポリシー マップが使用されます。 **（注）** ポリシー マップの作成と設定については、『*Cisco IOS XR Modular Quality of Service Configuration Guide*』を参照してください。
ステップ 22	**exit** **例：** `RP/0/RSP0/CPU0:router(config-fr-vc)# exit`	フレームリレー仮想回線モードを終了します。

	コマンドまたはアクション	目的
ステップ 23	**exit** **例：** `RP/0/RSP0/CPU0:router(config-subif)# exit`	サブインターフェイス コンフィギュレーション モードを終了します。
ステップ 24	**interface serial** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface serial 0/1/0/0/0/0:0`	完全なインターフェイス番号を *rack/slot/module/port/T3Num/T1num:instance* 表記で指定します。
ステップ 25	**encapsulation mfr** **例：** `RP/0/RSP0/CPU0:router(config)# encapsulation mfr`	シリアル インターフェイスでマルチリンク フレーム リレーをイネーブルにします。
ステップ 26	**multilink group** *group-id* **例：** `RP/0/RSP0/CPU0:router(config-if)# multilink group 100`	このインターフェイスのマルチリンク グループ ID を指定します。
ステップ 27	**frame-relay multilink lid** *link-id name* **例：** `RP/0/RSP0/CPU0:router(config-if)# frame-relay multilink lid sj1`	**（注）** フレームリレー マルチリンク バンドル リンクの名前を設定します。
ステップ 28	**frame-relay multilink ack** *ack-timeout* **例：** `RP/0/RSP0/CPU0:router(config-if)# frame-relay multilink ack 5`	フレーム リレー マルチリンク バンドル リンクの確認応答タイムアウト値を設定します。
ステップ 29	**frame-relay multilink hello** *hello-interval* **例：** `RP/0/RSP0/CPU0:router(config-if)# frame-relay multilink hello 60`	フレーム リレー マルチリンク バンドル リンクの hello 間隔を設定します。
ステップ 30	**frame-relay multilink retry** *retry-count* **例：** `RP/0/RSP0/CPU0:router(config-if)# frame-relay multilink retry 2`	フレーム リレー マルチリンク バンドル リンクの再送信の再試行回数を設定します。
ステップ 31	**exit** **例：** `RP/0/RSP0/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。

	コマンドまたはアクション	目的
ステップ 32	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 33	**exit** **例：** `RP/0/RSP0/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 34	**show frame-relay multilink interface** *type interface-path-id* [**detail** | **verbose**] **例：** `RP/0/RSP0/CPU0:router# show frame-relay multilink interface Multilink 0/5/1/0/1`	バンドル固有の情報やフレームリレー情報など、インターフェイス記述ブロック（IDB）から取得した情報を表示します。

チャネライズド フレームリレー シリアル インターフェイスでの FRF.12 エンドツーエンド フラグメンテーションの設定

チャネライズド フレームリレー シリアル インターフェイスで FRF.12 エンドツーエンド フラグメンテーションを設定するには、次の手順で操作します。

手順の概要

1. **config**
2. **controller t3** *interface-path-id*
3. **mode** *type*
4. **clock source** {**internal** | **line**}

5. **exit**
6. **controller t1** *interface-path-id*
7. **channel-group** *channel-group-number*
8. **timeslots** *range*
9. **exit**
10. **exit**
11. **interface serial** *interface-path-id*
12. **encapsulation frame-relay**
13. **exit**
14. **interface serial** *interface-path-id*
15. **ipv4 address** *ip-address*
16. **pvc** *dlci*
17. **service-policy** {**input** | **output**} *policy-map*
18. **fragment end-to-end** *fragment-size*
19. **exit**
20. **exit**
21. **exit**
22. **end**
    または
    commit
23. **exit**
24. **show frame-relay pvc** [**dlci** | **interface** | **location**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**config** **例：** `RP/0/RSP0/CPU0:router# config`	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	**controller t3** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t3 0/1/0/0`	*rack/slot/module/port* 表記で T3 コントローラ名を指定し、T3 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	**mode** *type* **例：** `RP/0/RSP0/CPU0:router(config-t3)# mode t1`	チャネライズするマルチリンクのタイプを設定します（たとえば、28 T1）。

	コマンドまたはアクション	目的
ステップ 4	**clock source** {**internal** | **line**} **例：** `RP/0/RSP0/CPU0:router(config-t3)# clock source internal`	（任意）個々の E3 リンクのクロッキングを設定します。 **（注）** デフォルトのクロック ソースは **internal** です。 **（注）** シリアル リンクでクロッキングを設定する場合、一方のエンドを **internal** にし、もう一方を **line** にする必要があります。接続の両エンドに **internal** クロッキングを設定すると、フレーム同期のずれが生じます。接続の両エンドに **line** クロッキングを設定すると、ラインはアップ状態になりません。
ステップ 5	**exit** **例：** `RP/0/RSP0/CPU0:router(config-t3)# exit`	T3/E3 または T1/E1 コントローラ コンフィギュレーション モードを終了します。
ステップ 6	**controller t1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t1 0/1/0/0/0`	T1 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	**channel-group** *channel-group-number* **例：** `RP/0/RSP0/CPU0:router(config-t1)# channel-group 0`	T1 チャネル グループを作成し、そのチャネル グループのチャネル 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 8	**timeslots** *range* **例：** `RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 1-24`	1 つまたは複数の DS0 タイムスロットをチャネル グループに関連付け、関連付けたシリアル サブインターフェイスをそのチャネル グループに作成します。 • 範囲は 1 ～ 24 タイムスロットです。 • 24 タイムスロットすべてを単一のチャネル グループに割り当てることも、タイムスロットを複数のチャネル グループに分割することもできます。 **（注）** 個々の T1 コントローラは、24 DS0 タイムスロットの合計をサポートします。
ステップ 9	**exit** **例：** `RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit`	チャネル グループ コンフィギュレーション モードを終了します。
ステップ 10	**exit** **例：** `RP/0/RSP0/CPU0:router(config-t1)# exit`	T1 コンフィギュレーション モードを終了します。

	コマンドまたはアクション	目的
ステップ 11	**interface serial** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface serial 0/1/0/0/0/0:0`	完全なインターフェイス番号を *rack/slot/module/port/T3Num/T1num:instance* 表記で指定します。
ステップ 12	**encapsulation frame-relay** **例：** `RP/0/RSP0/CPU0:Router(config-if)# encapsulation frame-relay`	フレームリレーのカプセル化タイプを指定します。
ステップ 13	**exit** **例：** `RP/0/RSP0/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 14	**interface serial** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface serial 1/0/0/0/0:0.1`	*rack*/*slot*/*module*/*port*[/*channel-num:channel-group-number*].*subinterface* 表記で、完全なサブインターフェイス番号を指定します。
ステップ 15	**ipv4 address** *ip-address* **例：** `RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 3.1.100.16 255.255.255.0`	次の形式でインターフェイスに IP アドレスおよびサブネット マスクを割り当てます。 *A.B.C.D/prefix* または *A.B.C.D/mask*
ステップ 16	**pvc** *dlci* **例：** `RP/0/RSP0/CPU0:router (config-subif)# pvc 100`	POS 相手先固定接続（PVC）を作成し、フレームリレー PVC コンフィギュレーション サブモードを開始します。 *dlci* を 16 から 1007 の範囲の PVC ID に置き換えます。 **（注）** 各サブインターフェイスに設定できる PVC は 1 つだけです。
ステップ 17	**service-policy** {**input** | **output**} *policy-map* **例：** `RP/0/RSP0/CPU0:router(config-fr-vc)# service-policy output policy-mapA`	ポリシー マップを入力サブインターフェイスまたは出力サブインターフェイスに付加します。付加すると、そのサブインターフェイスのサービス ポリシーとしてポリシー マップが使用されます。 **（注）** 効率的な FRF.12 機能（具体的にはインターリーブ）のためには、出力サービス ポリシーに優先順位を設定します。 **（注）** ポリシー マップの作成と設定については、『*Cisco IOS XR Modular Quality of Service Configuration Guide*』を参照してください。

	コマンドまたはアクション	目的
ステップ 18	**fragment end-to-end** *fragment-size* **例：** `RP/0/RSP0/CPU0:router(config-fr-vc)# fragment end-to-end 100`	（任意）インターフェイスでのフレーム リレー フレームのフラグメンテーションをイネーブルにし、元のフレームから各フラグメントに入れるペイロードのサイズ（バイト単位）を指定します。この数値には、元のフレームのフレームリレー ヘッダーは含まれません。 使用するハードウェアによって、有効値は 64 ～ 512 です。
ステップ 19	**exit** **例：** `RP/0/RSP0/CPU0:router(config-fr-vc)# exit`	フレームリレー仮想回線モードを終了します。
ステップ 20	**exit** **例：** `RP/0/RSP0/CPU0:router(config-subif)# exit`	サブインターフェイス コンフィギュレーション モードを終了します。
ステップ 21	**exit** **例：** `RP/0/RSP0/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 22	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

	コマンドまたはアクション	目的
ステップ 23	**exit** **例：** RP/0/RSP0/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 24	**show frame-relay pvc** [**dlci** | **interface** | **location**] **例：** RP/0/RSP0/CPU0:router# show frame-relay pvc 100	指定した PVC DLCI、インターフェイス、または場所の情報を表示します。

フレーム リレーの設定例

ここでは、次の設定例について説明します。

- 「オプションのフレームリレー パラメータ：例」（P.617）
- 「マルチリンク フレームリレー：例」（P.620）
- 「エンドツーエンド フラグメンテーション：例」（P.620）

オプションのフレームリレー パラメータ：例

次の例は、フレームリレーのカプセル化を設定した POS インターフェイスを始動および設定する方法です。この例では、インターフェイスが DCE で ANSI T1.617a-1994 Annex D LMI をサポートするように、デフォルトのフレームリレー設定を変更します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/RSP0/CPU0:router(config-if)# encapsulation frame-relay IETF
RP/0/RSP0/CPU0:router(config-if)# frame-relay intf-type dce
RP/0/RSP0/CPU0:router(config-if)# frame-relay lmi-type ansi
RP/0/RSP0/CPU0:router(config-if)# no shutdown
RP/0/RSP0/CPU0:router(config-if)# end

Uncommitted changes found, commit them? [yes]: yes

RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router (config)# interface pos 0/3/0/0.10 point-to-point
RP/0/RSP0/CPU0:router (config-subif)#ipv4 address 10.46.8.6/24
RP/0/RSP0/CPU0:router (config-subif)# pvc 20
RP/0/RSP0/CPU0:router (config-fr-vc)# encap ietf
RP/0/RSP0/CPU0:router(config-subif)# commit
```

次の例では、フレーム リレー カプセル化を設定した POS インターフェイスで LMI をディセーブルにする方法を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface
RP/0/RSP0/CPU0:router(config)# interface pos 0/3/0/0
RP/0/RSP0/CPU0:router(config-if)# frame-relay lmi disable
RP/0/RSP0/CPU0:router(config-if)# end

Uncommitted changes found, commit them? [yes]: yes
```

次の例は、シリアル インターフェイスで LMI を再度イネーブルにする方法です。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface
RP/0/RSP0/CPU0:router(config)# interface serial 0/3/0/0
RP/0/RSP0/CPU0:router(config-if)# no frame-relay lmi disable
RP/0/RSP0/CPU0:router(config-if)# end

Uncommitted changes found, commit them? [yes]: yes
```

次の例では、すべてのインターフェイスの LMI のフレーム リレー統計情報を表示する方法を示します。

```
RP/0/RSP0/CPU0:router# show frame-relay lmi

LMI Statistics for interface POS0/1/0/0/ (Frame Relay DCE) LMI TYPE = ANSI
Invalid Unnumbered Info 0             Invalid Prot Disc 0
Invalid Dummy Call Ref 0              Invalid Msg Type 0
Invalid Status Message 0              Invalid Lock Shift 9
Invalid Information ID 0              Invalid Report IE Len 0
Invalid Report Request 0              Invalid Keep IE Len 0
Num Status Enq. Rcvd 9444             Num Status Msgs Sent 9444
Num Full Status Sent 1578             Num St Enq. Timeouts 41
Num Link Timeouts 7

LMI Statistics for interface POS0/1/0/1/ (Frame Relay DCE) LMI TYPE = CISCO
Invalid Unnumbered Info 0             Invalid Prot Disc 0
Invalid Dummy Call Ref 0              Invalid Msg Type 0
Invalid Status Message 0              Invalid Lock Shift 0
Invalid Information ID 0              Invalid Report IE Len 0
Invalid Report Request 0              Invalid Keep IE Len 0
Num Status Enq. Rcvd 9481             Num Status Msgs Sent 9481
Num Full Status Sent 1588             Num St Enq. Timeouts 16
Num Link Timeouts 4
```

次の例は、メイン シリアル インターフェイス上の PVC でシリアル サブインターフェイスを作成する方法です。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface serial 0/3/0/0/0:0.10 point-to-point
RP/0/RSP0/CPU0:router (config-subif)#ipv4 address 10.46.8.6/24
RP/0/RSP0/CPU0:router (config-subif)# pvc 20
RP/0/RSP0/CPU0:router (config-fr-vc)# encapsulation ietf
RP/0/RSP0/CPU0:router(config-subif)# commit
```

次の例は、システムに設定されているすべての PVC に関する情報を表示する方法です。

```
RP/0/RSP0/CPU0router# show frame-relay pvc

PVC Statistics for interface Serial0/3/2/0 (Frame Relay DCE)

              Active     Inactive      Deleted       Static
  Local          4           0            0             0
  Switched       0           0            0             0
  Dynamic        0           0            0             0

DLCI = 612, DLCI USAGE = LOCAL, ENCAP = CISCO, INHERIT = TRUE, PVC STATUS = ACTI
VE, INTERFACE = Serial0/3/2/0.1
  input pkts 0          output pkts 0           in bytes 0
  out bytes 0           dropped pkts 0          in FECN packets 0
  in BECN pkts 0        out FECN pkts 0         out BECN pkts 0
  in DE pkts 0          out DE pkts 0
  out bcast pkts 0      out bcast bytes 0
  pvc create time 00:00:00      last time pvc status changed 00:00:00
```

```
DLCI = 613, DLCI USAGE = LOCAL, ENCAP = CISCO, INHERIT = TRUE, PVC STATUS = ACTI
VE, INTERFACE = Serial0/3/2/0.2
  input pkts 0          output pkts 0           in bytes 0
  out bytes 0           dropped pkts 0          in FECN packets 0
  in BECN pkts 0        out FECN pkts 0         out BECN pkts 0
  in DE pkts 0          out DE pkts 0
  out bcast pkts 0      out bcast bytes 0
  pvc create time 00:00:00      last time pvc status changed 00:00:00

DLCI = 614, DLCI USAGE = LOCAL, ENCAP = CISCO, INHERIT = TRUE, PVC STATUS = ACTI
VE, INTERFACE = Serial0/3/2/0.3
  input pkts 0          output pkts 0           in bytes 0
  out bytes 0           dropped pkts 0          in FECN packets 0
  in BECN pkts 0        out FECN pkts 0         out BECN pkts 0
  in DE pkts 0          out DE pkts 0
  out bcast pkts 0      out bcast bytes 0
  pvc create time 00:00:00      last time pvc status changed 00:00:00

DLCI = 615, DLCI USAGE = LOCAL, ENCAP = CISCO, INHERIT = TRUE, PVC STATUS = ACTI
VE, INTERFACE = Serial0/3/2/0.4
  input pkts 0          output pkts 0           in bytes 0
  out bytes 0           dropped pkts 0          in FECN packets 0
  in BECN pkts 0        out FECN pkts 0         out BECN pkts 0
  in DE pkts 0          out DE pkts 0
  out bcast pkts 0      out bcast bytes 0
  pvc create time 00:00:00      last time pvc status changed 00:00:00
```

次の例では、DTE 用に設定されている PVC の LMI ポーリング オプションを変更してから、**show frame-relay lmi** コマンドと **show frame-relay lmi-info** コマンドを使用してインターフェイスのモニタリングとトラブルシューティングの情報を表示する方法を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface pos 0/3/0/0
RP/0/RSP0/CPU0:router(config-if)# frame-relay lmi-n391dte 10
RP/0/RSP0/CPU0:router(config-if)# frame-relay lmi-n391dte 5
RP/0/RSP0/CPU0:router(config-if)# frame-relay lmi-t391dte 15
RP/0/RSP0/CPU0:router(config-subif)# commit

RP/0/RSP0/CPU0:router# show frame-relay lmi interface pos 0/3/0/0

LMI Statistics for interface pos 0/3/0/0 (Frame Relay DTE) LMI TYPE = ANSI
Invalid Unnumbered Info 0             Invalid Prot Disc 0
Invalid Dummy Call Ref 0              Invalid Msg Type 0
Invalid Status Message 0              Invalid Lock Shift 9
Invalid Information ID 0              Invalid Report IE Len 0
Invalid Report Request 0              Invalid Keep IE Len 0
Num Status Enq. Rcvd 9444             Num Status Msgs Sent 9444
Num Full Status Sent 1578             Num St Enq. Timeouts 41
Num Link Timeouts 7

RP/0/RSP0/CPU0:router# show frame-relay lmi-info interface pos 0/3/0/0

LMI IDB Info for interface POS0/3/0/0
  ifhandle:             0x6176840
  Interface type:       DTE
  Interface state:      UP
  Line Protocol:        UP
  LMI type (cnf/oper):  AUTO/CISCO
  LMI type autosense:   OFF
  Interface MTU:        1504
  -------------- DTE -------------
  T391:                 15s
  N391: (cnf/oper):     5/5
```

```
  N392: (cnf/oper):     3/0
  N393:                 4
  My seq#:              83
  My seq# seen:         83
  Your seq# seen:       82
  -------------- DCE --------------
  T392:                 15s
  N392: (cnf/oper):     3/0
  N393:                 4
  My seq#:              0
  My seq# seen:         0
  Your seq# seen:       0
```

マルチリンク フレームリレー：例

次の例は、シリアル インターフェイスでマルチリンク フレームリレーを設定する方法です。

```
RP/0/RSP0/CPU0:router# config
RP/0/RSP0/CPU0:router(config)# controller MgmtMultilink 0/3/1/0
RP/0/RSP0/CPU0:router(config-mgmtmultilink)# bundle 100
RP/0/RSP0/CPU0:router(config-mgmtmultilink)# exit

RP/0/RSP0/CPU0:router(config)# controller T3 0/3/1/0
RP/0/RSP0/CPU0:router(config-t3)# mode t1
RP/0/RSP0/CPU0:router(config-t3)# clock source internal
RP/0/RSP0/CPU0:router(config-t3)# exit

RP/0/RSP0/CPU0:router(config)# controller T1 0/3/1/0/0
RP/0/RSP0/CPU0:router(config-t1)# channel-group 0
RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 1-24
RP/0/RSP0/CPU0:router(config-t1-channel_group)#  exit
RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit
RP/0/RSP0/CPU0:router(config-t1)# exit

RP/0/RSP0/CPU0:router(config)# interface Multilink 0/3/1/0/100
RP/0/RSP0/CPU0:router(config-if)# encapsulation frame-relay
RP/0/RSP0/CPU0:router(config-if)# exit

RP/0/RSP0/CPU0:router(config)# interface Multilink 0/3/1/0/100.16 point-to-point
RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 3.1.100.16 255.255.255.0
RP/0/RSP0/CPU0:router(config-subif)# pvc 16
RP/0/RSP0/CPU0:router(config-fr-vc)# service-policy output policy-mapA
RP/0/RSP0/CPU0:router(config-fr-vc)# exit
RP/0/RSP0/CPU0:router(config-subif)# exit

RP/0/RSP0/CPU0:router(config)# interface Serial 0/3/1/0/0:0
RP/0/RSP0/CPU0:router(config-if)# encapsulation mfr
RP/0/RSP0/CPU0:router(config-if)# multilink group 100
RP/0/RSP0/CPU0:router(config-if)# frame-relay multilink lid sj1
RP/0/RSP0/CPU0:router(config-if)# frame-relay multilink ack 5
RP/0/RSP0/CPU0:router(config-if)# frame-relay multilink hello 60
RP/0/RSP0/CPU0:router(config-if)# frame-relay multilink retry 2
RP/0/RSP0/CPU0:router(config-if)# exit
RP/0/RSP0/CPU0:router(config)#
```

エンドツーエンド フラグメンテーション：例

次の例は、チャネライズド フレームリレー シリアル インターフェイスで FRF.12 エンドツーエンド フラグメンテーションを設定する方法です。

```
RP/0/RSP0/CPU0:router# config
RP/0/RSP0/CPU0:router(config)# controller T30/3/1/0
RP/0/RSP0/CPU0:router(config-t3)# mode t1
RP/0/RSP0/CPU0:router(config-t3)# clock source internal
RP/0/RSP0/CPU0:router(config-t3)# exit
RP/0/RSP0/CPU0:router(config-t3)# controller T10/3/1/0/0
RP/0/RSP0/CPU0:router(config-t1)# channel-group 0
RP/0/RSP0/CPU0:router(config-t1-channel_group)#  timeslots 1-24
RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit
RP/0/RSP0/CPU0:router(config-t1-channel_group)# interface Serial 0/3/1/0/0:0
RP/0/RSP0/CPU0:router(config-if)# encapsulation frame-relay
RP/0/RSP0/CPU0:router(config-if)# exit
RP/0/RSP0/CPU0:router(config-if)# interface Serial 0/3/1/0/0:0.100 point-to-point
RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 3.1.1.1 255.255.255.0
RP/0/RSP0/CPU0:router(config-subif)#  pvc 100
RP/0/RSP0/CPU0:router(config-fr-vc)#  service-policy output LFI
RP/0/RSP0/CPU0:router(config-fr-vc)# fragment end-to-end 256
```

その他の関連資料

ここでは、フレーム リレーに関する関連資料について説明します。

関連資料

関連項目	参照先
Cisco IOS XR マスター コマンド リファレンス	『*Cisco IOS XR Master Commands List*』
Cisco IOS XR インターフェイス コンフィギュレーション コマンド	『*Cisco IOS XR Interface and Hardware Component Command Reference*』
Cisco IOS XR ソフトウェアを使用した初期システム ブートアップとルータの設定情報	『*Cisco IOS XR Getting Started Guide*』
Cisco IOS XR AAA サービス構成情報	『*Cisco IOS XR System Security Configuration Guide*』および『*Cisco IOS XR System Security Command Reference*』

標準

標準	タイトル
FRF.12	*フレームリレー フォーラム .12*
FRF.16	*フレームリレー フォーラム .16*
ANSI T1.617 Annex D	*American National Standards Institute T1.617 Annex D*
ITU Q.933 Annex A	*International Telecommunication Union Q.933 Annex A*

MIB

MIB	MIB のリンク
FRF.16 MIB Cisco Frame Relay MIB IF-MIB 『Management Information Base for Frame Relay DTEs』 Management Information Base for Frame Relay DTEs Using SMIv2	Cisco IOS XR ソフトウェアを使用して MIB を検索およびダウンロードするには、http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml にある Cisco MIB Locator を使用し、[Cisco Access Products] メニューからプラットフォームを選択します。

RFC

RFC	タイトル
RFC 1294	『*Multiprotocol Interconnect Over Frame Relay*』
RFC 1315	『*Management Information Base for Frame Relay DTEs*』
RFC 1490	『*Multiprotocol Interconnect Over Frame Relay*』
RFC 1586	『*Guidelines for Running OSPF Over Frame Relay Networks*』
RFC 1604	『*Definitions of Managed Objects for Frame Relay Service*』
RFC 2115	『*Management Information Base for Frame Relay DTEs Using SMIv2*』
RFC 2390	『*Inverse Address Resolution Protocol*』

RFC	タイトル
RFC 2427	『*Multiprotocol Interconnect Over Frame Relay*』
RFC 2954	『*Definitions of Managed Objects for Frame Relay Service*』
RFC 3020	『*RFC for FRF.16 MIB*』

シスコのテクニカル サポート

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

その他の関連資料

Cisco ASR 9000 シリーズ ルータ での PPP の設定

このモジュールでは、Cisco ASR 9000 シリーズ ルータでの POS およびシリアル インターフェイスでのポイントツーポイント プロトコル（PPP）の設定について説明します。

PPP インターフェイス設定の機能履歴

リリース	変更内容
リリース 3.9.0	PPP、および PPP および MLPPP の ICSSO が Cisco ASR 9000 シリーズ ルータで導入されました。
リリース 3.9.1	T3 チャネライズド SONET のサポートが追加されました。
リリース 4.0.0	次の機能のサポートが、2 ポート チャネライズド OC-12c/DS0 SPA に対して追加されました。 • IPHC over PPP、MLPPP、および MLPPP/LFI • NxDS0 シリアル インターフェイス PPP のサポートが次の SPA に対して導入されました。 • 1 ポート チャネライズド OC-48/STM-16 SPA • 1 ポート OC-192c/STM-64 POS/RPR XFP SPA • 2 ポート OC-48c/STM-16 POS/RPR SPA • 8 ポート OC-12c/STM-4 POS SPA

リリース 4.0.1	Cisco ASR 9000 シリーズ ルータでの PPP サポートが次の SPA に対して追加されました。 • Cisco 1 ポート チャネライズド OC-3/STM-1 SPA（MLPPP もサポート） • Cisco 2 ポートおよび 4 ポート クリア チャネル T3/E3 SPA • Cisco 4 ポート OC-3c/STM-1 SPA • Cisco 8 ポート OC-3c/STM-1 SPA
リリース 4.1.0	ノイズ属性のサポートが PPP に対して追加されました。リンクにおいてリンク ノイズ モニタリング（LNM）しきい値を超えたときに MLPPP バンドル上 のリンクを削除できるようにするためです。 PPP のサポート（T1/E1 チャネルでの MLPPP のサポートなど）が、次の SPA で導入されました。 • Cisco 4 ポート チャネライズド T3 SPA • Cisco 8 ポート チャネライズド T1/E1 SPA

内容

PPP 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。

POS インターフェイスまたはシリアル インターフェイスで PPP 認証を設定する前に、次のタスクと条件を満たしていることを確認します。

- 使用しているハードウェアが POS インターフェイスまたはシリアル インターフェイスをサポートしている必要があります。
- 対応するモジュールの説明に従って、**encap ppp** コマンドを使用し、インターフェイスで PPP のカプセル化をイネーブルにしました。
  - POS インターフェイスで POS のカプセル化をイネーブルにするには、このマニュアルの *「Cisco ASR 9000 シリーズ ルータでの POS インターフェイスの設定」* モジュールを参照してください。
  - シリアル インターフェイスで POS のカプセル化をイネーブルにするには、このマニュアルの *「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」* モジュールを参照してください。

PPP について

PPP および関連機能を設定するには、この項で説明する情報を理解している必要があります。

PPP 認証

インターフェイスに PPP 認証が設定されている場合、ホストは、PPP 接続を確立する前に他のホストがセキュア パスワードを使用して自身を一意に識別することを求めます。このパスワードは一意で、両方のホストで認識されています。

PPP は、次の認証プロトコルをサポートします。

- チャレンジ ハンドシェイク認証プロトコル（CHAP）
- Microsoft による CHAP プロトコルの拡張版（MS-CHAP）
- パスワード認証プロトコル（PAP）

POS インターフェイスまたはシリアル インターフェイス上で初めて PPP をイネーブルにしたときは、対象のインターフェイスで CHAP、MS-CHAP、PAP のいずれかのシークレット パスワードを設定するまで、そのインターフェイスでの認証はイネーブルになりません。インターフェイスで PPP を設定する場合、次の点に気を付けてください。

- CHAP、MS-CHAP、PAP は単一のインターフェイスに設定できますが、一度に使用される認証方式は 1 つだけです。使用される認証プロトコルの順序は、LCP ネゴシエーション中のピアによって決定されます。使用される最初の認証方式は、ピアによってもサポートされます。
- PAP は、POS インターフェイスおよびシリアル インターフェイスで使用可能な最小のセキュア認証プロトコルです。POS インターフェイスおよびシリアル インターフェイス経由で送信される情報について、より高レベルのセキュリティを確保するため、PAP 認証に加えて CHAP または MS-CHAP 認証を設定することをお勧めします。
- PPP 認証をイネーブル化またはディセーブル化しても、ローカル ルータがリモート デバイスに対して自身を認証しようとすることには変わりありません。
- **ppp authentication** コマンドは、インターフェイス上で CHAP、MS-CHAP、PAP 認証が選択される順序を指定するときにも使用されます。CHAP、MS-CHAP、PAP は、任意の順序でイネーブル化できます。3 つのすべての方式をイネーブル化すると、リンク ネゴシエーションでは、最初に指定された方式が要求されます。ピアが 2 番目の方式の使用を提案した場合、または最初の方式を拒否した場合は、2 番目の方式が試行されます。リモート装置の中には、1 つの方式しかサポートしないものがあります。方式の順序は、適切な方式で正しくネゴシエーションするためにリモート デバイスの機能で指定された方式と、求められるデータ ライン セキュリティのレベルに基づいて決定されます。PAP ユーザ名とパスワードはクリア テキスト文字列として送信されます。この文字列は、代行受信や再利用が可能です。

注意 **aaa authentication ppp** コマンドを使わずに設定した *list-name* 値を使用すると、インターフェイスはピアを認証できません。**ppp** キーワードを指定した **aaa authentication** コマンドの実装についての詳細は、『*Cisco IOS XR System Security Command* Reference』の「*Authentication,*

Authorization, and Accounting Commands on Cisco IOS XR Software」および『*Cisco IOS XR System Security Configuration Guide*』の「*Configuring AAA Services on Cisco IOS XR Software*」を参照してください。

PAP 認証

PAP は、リモート ノードに対し、2 ウェイ ハンドシェイクを使用してそのアイデンティティを確立するためのシンプルな方式を提供します。2 台のホスト間で PPP リンクが確立した後、ユーザ名とパスワードのペアは認証が確認されるまで、または接続が終了するまで、リモート ノードによってリンクを経由して（クリア テキストで）繰り返し送信されます。

PAP はセキュアな認証プロトコルではありません。パスワードはリンクを経由してクリア テキストで送信され、プレイバック攻撃やトライアルアンドエラー攻撃からの保護機能はありません。リモート ノードは、ログイン試行の頻度とタイミングを管理しています。

CHAP 認証

CHAP は RFC 1994 で定義され、3 ウェイ ハンドシェイクを使用してピアのアイデンティティを確認します。次の手順に、CHAP プロセスの概要を示します。

ステップ 1 CHAP オーセンティケータがピアにチャレンジ メッセージを送信します。

ステップ 2 ピアは 1 ウェイ ハッシュ関数で算出された値で応答します。

ステップ 3 オーセンティケータは、応答を、独自の計算で予測したハッシュ値と照合します。値が一致すると、認証は成功します。値が一致しないと、接続は終了します。

この認証方式は、オーセンティケータとピアでのみ認識されている CHAP パスワードによって決まります。CHAP パスワードは、リンク経由では送信されません。認証は 1 ウェイですが、相互認証に同じ CHAP パスワード セットを使用することで、CHAP のネゴシエーションを双方向に行うことができます。

（注） 有効な CHAP 認証には、両方のホストの CHAP パスワードが同一であることが必要です。

MS-CHAP 認証

Microsoft チャレンジ ハンドシェイク認証プロトコル（MS-CHAP）は、Microsoft バージョンの CHAP で、RFC 1994 の拡張です。MS-CHAP では、CHAP と同じ認証プロセスが使用されます。ただし、認証は、Microsoft Windows NT または Microsoft Windows 95 を実行する PC と、ネットワーク アクセス サーバ（NAS）として動作する Cisco ルータまたはアクセス サーバとの間で行われます。

（注） 有効な MS-CHAP 認証には、両方のホストの MS-CHAP パスワードが同一であることが必要です。

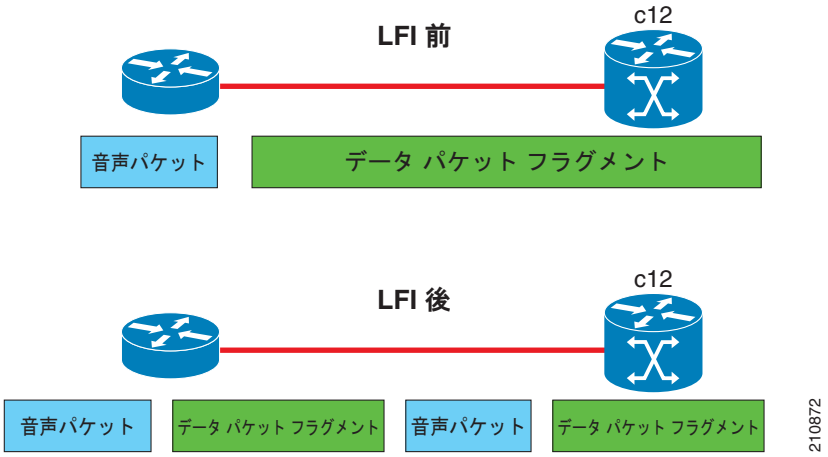
マルチリンク PPP

マルチリンク ポイントツーポイント プロトコル（MLPPP）は、複数の物理リンクを組み合わせて 1 つの論理リンクを構成する機能を持ちます。実装によって、複数の PPP インターフェイスが結合されて 1 つのマルチリンク インターフェイスとなります。MLPPP は、複数の PPP リンクでデータグラムの断片化、再編成、および配列を行います。

リンク フラグメンテーション/インターリーブ（LFI）は、MLPPP インターフェイス用に設計されており、低速インターフェイス上の音声およびデータを統合するときに必要です。

LFI は、データと同じ回線上を移動する音声やビデオなど、遅延の影響を受けやすいトラフィックを安定させます。ネットワークが低速インターフェイスの大きなパケットを処理しているとき、音声は増大した遅延およびジッターの影響を受けやすくなります。LFI は、大きなデータグラムを分割（フラグメント）し、これらを低遅延のトラフィック パケットにインターリーブすることで、遅延やジッタを軽減します。

図 34　リンク フラグメンテーション/インターリーブ



MLPPP の機能概要

Cisco IOS XR での MLPPP は、PPP シリアル インターフェイスでサポートされているのと同じ機能（ただし、QoS を除きます）を提供します。また、次の追加機能も提供します。

- 長いシーケンス番号（24 ビット）。
- 失われたフラグメントの検出タイムアウト期間（1 秒）
- 最小アクティブ リンクの設定オプション。
- マルチリンク インターフェイスでの LCP エコー要求および応答のサポート。
- フル T1 および E1 フレームおよび非フレーム リンク。
- Cisco 2 ポート チャネライズド OC-12c/DS0 SPA での、T1/E1 リンクでのノイズ エラーのしきい値設定のサポート。これは、PPP にノイズ属性を通知して MLPPP バンドル リンクを削除させるために使用されます。LNM の詳細については、『*Cisco ASR 9000 アグリゲーション サービス ルータ 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*』の「Cisco ASR 9000 シリーズ ルータでのクリア チャネル T3/E3 およびチャネライズド T3 および T1/E1 コントローラの設定」モジュールを参照してください。

IPHC Over MLPPP

2 ポート チャネライズド OC-12c/DS0 SPA は、IPHC over PPP、MLPPP、および MLPPP/LFI をサポートします。IPHC の詳細と設定方法については、『*Cisco ASR 9000 アグリゲーション サービス ルータ 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*』の「Cisco ASR 9000 シリーズルータでのシリアル インターフェイスの設定」モジュールを参照してください。

PPP および MLPPP の ICSSO

（注） SR-APS および MR-APS は、Cisco 1 ポート チャネライズド OC-48/STM-16 SPA でサポートされません。

Cisco ASR 9000 シリーズ ルータでのシャーシ間ステートフル スイッチオーバー（ICSSO）の機能は、ポイントツーポイント プロトコル（PPP）やマルチリンク PPP（MLPPP）のセッションを、マルチルータ自動保護スイッチング（MR-APS）現用ルータから MR-APS 保護ルータへの MR-APS スイッチオーバー時にも維持することです。

ICSSO によって、新しい MR-APS アクティブ ルータとリモート PPP/MLPPP ピア デバイス間のリンク制御プロトコル（LCP）または IP 制御プロトコル（IPCP）再ネゴシエーションの必要なしに、MR-APS スイッチオーバーが可能になります。ICSSO の主な目的は、MR-APS スイッチオーバー中に加入者セッションおよびデータ損失を最小限に抑えることです。

ICSSO は、アクティブ ルータの PPP および MLPPP の状態情報とバックアップ ルータの状態情報を同期して、バックアップ ルータが MR-APS スイッチオーバーの後すぐにトラフィックを転送する準備が必ずできているようにします。

ICSSO は次に示す他のソフトウェア コンポーネントと連携します。

- 「マルチルータ自動保護スイッチング（MR-APS）」（P.630）
- 「セッション状態冗長プロトコル（SSRP）」（P.631）
- 「冗長グループ マネージャ（RG-MGR）」（P.631）
- 「IP の高速再ルーティング（IP-FRR）」（P.631）
- 「VPN ルーティングおよび転送（VRF）」（P.632）
- 「Open Shortest Path First（OSPF）」（P.632）

マルチルータ自動保護スイッチング（MR-APS）

マルチルータ自動保護スイッチング（MR-APS）は、設備や機器の障害に対してレイヤ 1 を保護するためのシスコの機能です。この機能を使用するには、2 台のそれぞれ異なるルータに配置された SONET コントローラの保護のペアを設定します。冗長バックアップ ルータはアクティブ ルータと同じように設定されていて、MR-APS スイッチオーバー時にトラフィックをただちに転送する準備ができています。

保護ペアの通信には、SONET ダウンストリーム接続からのレイヤ 1（k1/k2）シグナリング バイト（Bellcore 仕様 GR-253-CORE に従う）とレイヤ 3 シグナリング メッセージが使用されます。これには、Protect Group Protocol（PGP）が使用されます。MR-APS は、バックアップ ルート使用に切り替える IP-FRR アップデートを間接的にトリガーするような障害の原因の多くを検出します。

MR-APS の設定では、異なるルータ上の 2 台のインターフェイスは、現用インターフェイスまたは保護インターフェイスのロールを割り当てられます。これらのロールはオペレータによって設定されます。通常の状態では、現用インターフェイスがアクティブ トラフィックを伝送します。現用インターフェイスに障害が発生した場合は、保護インターフェイスがただちにアクティブ トラフィックを引き継ぐので、PPP トラフィックが失われることはありません。

セッション状態冗長プロトコル（SSRP）

MR-APS に設定された SONET コントローラのペアは、セッション状態冗長プロトコル（SSRP）保護グループの一部です。SSRP は、アクティブとスタンバイのルータ間でインターフェイスとシステムの状態情報を伝達します。SSRP には、キープアライブ プロトコルとしての役割もあります。

SSRP を設定するには、SONET コントローラにシャーシ間冗長グループを関連付け、MR-APS ピア ルータによる各アクティブ SONET コントローラでの PPP セッション ステート同期化をイネーブルにします。

PPP セッションは、次の 3 つの状態のいずれかになれます。

- Active : PPP セッションがアクティブ状態となるのは、PPP セッション ネゴシエーションが完了し、関連付けられたルートがインストール済みで、関連付けられた隣接関係が作成済みのときです。Active 状態の PPP セッションは、スタンバイ ルータのピアにデータを複製します。
- Standby Up : スタンバイ ルータ上の PPP セッションが Standby Up 状態となるのは、複製された状態情報がアクティブ ルータから受信済みで、関連付けられた PPP ルートがインストール済みで、関連付けられた隣接関係が作成済みのときです。Standby Up 状態の PPP セッションは、MR-APS スイッチオーバー直後からトラフィックを転送できる状態になっています。
- Standby Down : スタンバイ ルータ上の PPP セッションが Standby Down 状態となるのは、関連付けられたルートがインストール済みではなく、隣接関係も作成されていないときです。

SSRP は MR-APS ピア ルータ間で動作し、TCP/IP を使用します。1 つの SSRP セッションは、冗長 SONET コントローラの各ペアで実行されます。これは、複数の SSRP セッションが MR-APS 冗長ルータの 1 つのペアで実行できることを意味しています。

（注） SSRP は冗長性制御プロトコルではなく、状態情報同期プロトコルです。

冗長グループ マネージャ（RG-MGR）

冗長グループ マネージャ（RG-MGR）は保護インターフェイスのバックアップ ルートを設定します。RG-MGR は保護された SONET コントローラでのイベントを登録し、ルーティング情報ベース（RIB）コンポーネントに IP 高速再ルーティング（IP-FRR）更新情報を渡します。

IP の高速再ルーティング（IP-FRR）

（注） IC-SSO で使用する場合、IP-FRR は PPP カプセル化だけでサポートされます。HDLC カプセル化との組み合わせではサポートされません。

IP 高速再ルーティング（IP-FRR）の特徴は、MR-APS スイッチオーバー後に、きわめて高速に PPP/MLPPP トラフィックの再ルーティングができることです。

IP-FRR はプライマリおよびバックアップ ルートを制御します。各ルートは、ルーティング情報ベース（RIB）内でマッピングされます。MR-APS スイッチオーバー後にトラフィックを転送するためにどのバックアップ パスが使用されるかは、IP-FRR によって制御されます。

MR-APS スイッチオーバーが発生すると、IP-FRR アップデートがトリガーされます。これによって、保護 SONET コントローラ上のバックアップ ルートがアクティブになります。現用 SONET コントローラが復元されると、別の IP-FRR アップデートがトリガーされ、トラフィックがプライマリ ルートに再ルーティングされます。

IP-FRR の詳細については、『*Cisco IOS XR MPLS Configuration Guide*』の「Implementing MPLS Traffic Engineering on Cisco IOS XR Software」モジュールを参照してください。

VPN ルーティングおよび転送（VRF）

ICSSO は、VPN ルーティングおよび転送（VRF）とともに使用できます。異なるサービス タイプごとにトラフィック ストリームを分離する場合、ユーザは VRF テクノロジーを使用して実行できます。VRF を利用すると、複数の独立したルーティングおよび転送データベースを作成して維持することができます。「ICSSO で使用するマルチリンクの VRF の設定：例」（P.674）および「ICSSO で使用するためのイーサネットの VRF の設定：例」（P.675）を参照してください。VRF の設定に関する詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*』を参照してください。

Open Shortest Path First（OSPF）

PPP セッションの終端がリモート ピアとなっている場合は、集約ルータはそのリモート ピアが使用可能かどうかを Open Shortest Path First（OSPF）を使用してネットワーク上でアドバタイズする必要があります。OSPF は、リモート PPP ピアが使用可能かどうかを ICSSO ピア ルータにアドバタイズするために必要です。「ICSSO で使用する OSPF の設定：例」（P.675）を参照してください。OSPF の設定方法の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*』を参照してください。

ICSSO の設定の概要

ICSSO は次のように設定されます。

- MR-APS の設定
- SSRP プロファイルを設定
- SSRP グループを設定
- PPP カプセル化のシリアル インターフェイスへの設定
- マルチリンク インターフェイスを設定
- ICSSO 設定を確認

このモジュールの「PPP および MLPPP の ICSSO の設定」（P.660）で、ICSSO を設定する手順を説明しています。

「PPP および MLPPP の ICSSO の設定：例」（P.670）では、ICSSO および関連コンポーネントの設定の具体的な例を示しています。

QoS を使用するマルチクラス MLPPP

マルチクラス マルチリンク ポイントツーポイント プロトコル（MLPPP）は、Quality of Service（QoS）と組み合わせて使用できます。設定するには、ポリシー マップ内の特定のクラスの下で **encap-sequence** コマンドを使用します。

encap-sequence コマンドは、MQC 定義クラス内のパケットの MLPPP MCMP クラス ID を指定します。

encap-sequence ID 番号の有効値は、**none**、0、1、2、3 です。**none** 値は、**priority level** が 1 のときだけ適用でき、MLPPP カプセル化がないことを示します。1、2、または 3 の値は、プライオリティ 1 もしくは 2 のクラスまたはキューイング アクションを含むその他のクラスで使用できます。
encap-sequence ID 番号の値のうち、0 はデフォルト クラス用に予約されており、他のクラスで指定することはできません。

（注） **encap-sequence** ID 番号は、番号順に設定する必要があります。たとえば、1 と 2 をすでに割り当てていない限り、ID 番号 3 は割り当てることができません。

encap-sequence ID 番号の数は、マルチリンク ヘッダーによってピア間でネゴシエーションされる MLPPP クラスの数未満でなければなりません。システムによってこれが確認されないため、ユーザは設定がこれに合っていることを確認する必要があります。

ppp multilink multiclass remote apply コマンドは、これを確認する方法を提供します。
encap-sequence ID 番号（デフォルト値の 0 を含む）を使用するクラスの数が、**ppp multilink multiclass remote apply** コマンドの *min-number* 値よりも小さいことを確認します。たとえば、ppp multilink multiclass remote apply コマンドの min-number 値が 4 の場合は、encap-sequence ID 番号を持つクラスは 3 つ以下となります。

QoS ポリシーは、次の条件を検証します。これらの条件が満たされていない場合、ポリシーは拒否されます。

- **encap-sequence** ID 番号が 1 ～ 3 という許容値の範囲内である。
- **encap-sequence** がポリシー マップ内でいずれかのクラスに対して設定されている場合は、そのポリシー マップ内のクラスのうち、**プライオリティ レベル 1** のものすべてに **encap-sequence** ID 番号も指定されていることが必要になります。
- **encap-sequence** を **none** に設定できるのは、**プライオリティ レベル**が 1 のクラスに限定されます。
- class-default には **encap-sequence** 設定は含まれていません。
- キューイング アクションを含むクラスだけが **encap-sequence** 設定を持ちます。

（注） 同じ **encap-sequence** ID 番号を共有するクラスは、プライオリティが同じである必要があります。

QoS ポリシー マップは、次のとおりに設定されます。

```
config
    policy-map type qos policy-name
        class class-name
            アクション
            アクション
            アクション
...
```

次に、MLPPP のポリシー マップを設定する例を示します。

```
config
    policy-map foo
        class ip-prec-1
            encap-sequence none
            police rate percent 10
            priority level 1
    !
        class ip-prec-2
```

```
        encap-sequence 1
        shape average percent 80
!
    class ip-prec-3
        encap-sequence 1
        bandwidth percent 10
!
    class class-default
!
end-policy-map
!
```

QoS および QoS コマンド設定の詳細については、『*Cisco ASR 9000 Series Aggregation Services Routers Modular Quality of Service Configuration Guide*』および『*Cisco ASR 9000 Series Aggregation Services Routers Modular Quality of Service Command Reference*』を参照してください。

T3 SONET チャネル

Cisco ASR 9000 シリーズ ルータは、次のハードウェアで T3 チャネライズド SONET をサポートします。

- SIP 700 SPA インターフェイス プロセッサ
- 1 ポート チャネライズド OC-3/STM-1 SPA
- 2 ポート チャネライズド OC-12c/DS0 SPA
- 1 ポート チャネライズド OC-48/STM-16 SPA

チャネライズド SONET によって、複数の T3 チャネルを同じ物理リンク上で転送できるようになります。

チャネライズド SONET、T3 および T1 コントローラ、シリアル インターフェイス、および SONET APS の設定の詳細については、次の関連モジュールを参照してください。

- 「Cisco ASR 9000 シリーズ ルータ でのチャネライズド SONET/SDH の設定」
- 「Cisco ASR 9000 シリーズ ルータ でのクリア チャネル SONET コントローラの設定」
- 「Cisco ASR 9000 シリーズ ルータでのクリア チャネル T3/E3 およびチャネライズド T3 および T1/E1 コントローラの設定」
- 「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」

PPP の設定方法

ここでは、次の手順について説明します。

- 「デフォルトの PPP 設定の変更」（P.635）
- 「PPP 認証の設定」（P.638）
- 「認証プロトコルのディセーブル化」（P.648）
- 「マルチリンク PPP の設定」（P.652）
- 「PPP および MLPPP の ICSSO の設定」（P.660）

デフォルトの PPP 設定の変更

インターフェイスで初めて PPP をイネーブルにすると、次のデフォルト設定が適用されます。

- 認証が失敗すると、ただちに、インターフェイスは自身をリセットします。
- 応答がなくても許可される設定要求の最大数は 10 で、この数を超えるとすべての要求が停止されます。
- 否定応答（CONFNAK）が連続して返される場合、それが許可される最大数は 5 で、この数を超えるとネゴシエーションが終了されます。
- 応答がなくても許可される終了要求（TermReq）の最大数は 2 で、この数を超えるとリンク制御プロトコル（LCP）またはネットワーク制御プロトコル（NCP）は終了されます。
- 認証パケットに対する応答の最大待機時間は 10 秒です。
- PPP ネゴシエーション中の応答の最大待機時間は 3 秒です。

ここでは、PPP カプセル化がイネーブルになっているシリアル インターフェイスまたは POS インターフェイスで基本的な PPP 設定を変更する手順について説明します。ここで使用するコマンドは、PPP（CHAP、MS-CHAP、PAP）によってサポートされるすべての種類の認証に適用されます。

前提条件

encapsulation ppp コマンドを使用し、インターフェイスで PPP カプセル化をイネーブルにする必要があります。

- POS インターフェイスで POS のカプセル化をイネーブルにするには、このマニュアルの *「Cisco ASR 9000 シリーズ ルータでの POS インターフェイスの設定」* モジュールを参照してください。
- インターフェイスで POS のカプセル化をイネーブルにするには、このマニュアルの *「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」* モジュールを参照してください。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp max-bad-auth** *retries*
4. **ppp max-configure** *retries*
5. **ppp max-failure** *retries*
6. **ppp max-terminate** *number*
7. **ppp timeout authentication** *seconds*
8. **ppp timeout retry** *seconds*
9. **end**
   または
   commit
10. **show ppp interfaces** {*type interface-path-id* | **all** | **brief** {*type interface-path-id* | **all** | **location** *node-id*} | **detail** {*type interface-path-id* | **all** | **location** *node-id*} | **location** *node-id*}

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface serial 0/4/0/1`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ppp max-bad-auth** *retries* **例：** `RP/0/RSP0/CPU0:router(config-if)# ppp max-bad-auth 3`	（任意）PPP 認証が失敗した後、インターフェイスで許可する認証のリトライ回数を設定します。 • 許可する認証のリトライ回数を指定しない場合、認証が失敗すると、ただちに、ルータは自身をリセットします。 • *retries* 引数を、0 ～ 10 の範囲でリトライ回数に置き換えます。この回数を超えると、インターフェイスは自身をリセットします。 • デフォルトのリトライ回数は 0 回です。 • **ppp max-bad-auth** コマンドは、PPP カプセル化がイネーブルになっている任意のインターフェイスに適用できます。
ステップ 4	**ppp max-configure** *retries* **例：** `RP/0/RSP0/CPU0:router(config-if)# ppp max-configure 4`	（任意）（応答なしで）試行される設定要求の最大数を指定します。この数を超えると、要求は停止されます。 • *retries* 引数を、4 ～ 20 の範囲で設定要求がリトライする最大回数に置き換えます。 • デフォルトの設定要求の最大数は 10 です。 • 設定要求の最大回数分だけ送信されないうちに設定要求メッセージが応答を受け取った場合、以降の設定要求は放棄されます。
ステップ 5	**ppp max-failure** *retries* **例：** `RP/0/RSP0/CPU0:router(config-if)# ppp max-failure 3`	（任意）否定応答（CONFNAK）が連続して返される場合に、それが許可される最大数を設定します。この数を超えるとネゴシエーションは終了されます。 • *retries* 引数を、2 ～ 10 の範囲で CONFNAK の最大数に置き換えます。この数を超えるとネゴシエーションは終了されます。 • デフォルトの CONFNAK の最大数は 5 です。

	コマンドまたはアクション	目的
ステップ 6	**ppp max-terminate** *number* **例：** `RP/0/RSP0/CPU0:router(config-if)# ppp max-terminate 5`	（任意）応答がなくても送信される終了要求（TermReq）の最大数を設定します。この数を超えるとリンク制御プロトコル（LCP）またはネットワーク制御プロトコル（NCP）は終了されます。 • *number* 引数を、応答がなくても送信される TermReq の最大数に置き換えます。この数を超えると LCP または NCP は終了されます。範囲は 2 ～ 10 です。 • デフォルトの TermReq の最大数は 2 です。
ステップ 7	**ppp timeout authentication** *seconds* **例：** `RP/0/RSP0/CPU0:router(config-if)# ppp timeout authentication 20`	（任意）PPP 認証タイムアウト パラメータを設定します。 • *seconds* 引数を、認証パケットに対する応答を待機する最大時間（秒）に置き換えます。範囲は 3 ～ 30 秒です。 • デフォルトの認証タイムアウトは 10 秒です。この時間には、リモート ルータが接続を認証して許可し、応答するまでの時間を組み込む必要があります。ただし、この処理に 10 秒かからないこともあります。そのような場合は **ppp timeout authentication** コマンドを使用してタイムアウト時間を短くし、認証応答が失われる場合の接続時間を改善します。
ステップ 8	**ppp timeout retry** *seconds* **例：** `RP/0/RSP0/CPU0:router(config-if)# ppp timeout retry 8`	（任意）PPP 認証タイムアウト リトライ パラメータを設定します。 • *seconds* 引数を、PPP ネゴシエーション時に応答を待機する最大時間（秒）に置き換えます。範囲は 1 ～ 10 秒です。 • デフォルトは 3 秒です。

	コマンドまたはアクション	目的
ステップ 9	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 10	**show ppp interfaces** {*type interface-path-id* | **all** | **brief** {*type interface-path-id* | **all** | **location** *node-id*} | **detail** {*type interface-path-id* | **all** | **location** *node-id*} | **location** *node-id*} **例：** `RP/0/RSP0/CPU0:router# show ppp interfaces serial 0/2/0/0`	インターフェイスまたは PPP カプセル化がイネーブルになっているすべてのインターフェイスの PPP 設定を確認します。

PPP 認証の設定

ここでは、次の手順について説明します。

- 「PAP、CHAP、MS-CHAP 認証のイネーブル化」（P.638）
- 「PAP 認証パスワードの設定」（P.642）
- 「CHAP 認証パスワードの設定」（P.644）
- 「MS-CHAP 認証パスワードの設定」（P.646）

PAP、CHAP、MS-CHAP 認証のイネーブル化

ここでは、シリアル インターフェイスまたは POS インターフェイスで PAP、CHAP、MS-CHAP 認証をイネーブルにする手順について説明します。

前提条件

次のモジュールの説明に従って、**encapsulation ppp** コマンドを使用し、インターフェイスで PPP のカプセル化をイネーブルにする必要があります。

- POS インターフェイスで POS のカプセル化をイネーブルにするには、このマニュアルの *「Cisco ASR 9000 シリーズ ルータでの POS インターフェイスの設定」* モジュールを参照してください。
- インターフェイスで POS のカプセル化をイネーブルにするには、このマニュアルの *「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」* モジュールを参照してください。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp authentication** *protocol* [*protocol* [*protocol*]] [*list-name* | **default**]
4. **end**
   または
   commit
5. **show ppp interfaces** {*type interface-path-id* | **all** | **brief** {*type interface-path-id* | **all** | **location** *node-id*} | **detail** {*type interface-path-id* | **all** | **location** *node-id*} | **location** *node-id*}

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**interface** *type interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface serial 0/4/0/1`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ppp authentication** *protocol* [*protocol* [*protocol*]] [*list-name* | **default**] **例：** `RP/0/RSP0/CPU0:router(config-if)# ppp authentication chap pap MIS-access`	インターフェイスで CHAP、MS-CHAP、または PAP をイネーブルにし、インターフェイスで CHAP、MS-CHAP、PAP 認証が選択される順序を指定します。 • *protocol* 引数を、**pap**、**chap**、または **ms-chap** に置き換えます。 • *list name* 引数を、使用する認証方式のリストの名前に置き換えます。リストを作成するには、『*Cisco IOS XR System Security Command* Reference』の「*Authentication, Authorization, and Accounting Commands on Cisco IOS XR Software*」モジュールに記載されている説明に従って **aaa authentication ppp** コマンドを使用します。 • リスト名を指定しない場合は、デフォルト名が使用されます。デフォルトのリストは、『*Cisco IOS XR System Security Command* Reference』の「*Authentication, Authorization, and Accounting Commands on Cisco IOS XR Software*」モジュールに記載されている説明に従って **aaa authentication ppp** コマンドで指定します。

	コマンドまたはアクション	目的
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	**show ppp interfaces** {*type interface-path-id* | **all** | **brief** {*type interface-path-id* | **all** | **location** *node-id*} | **detail** {*type interface-path-id* | **all** | **location** *node-id*} | **location** *node-id*} **例：** `RP/0/RSP0/CPU0:router# show ppp interfaces serial 0/2/0/0`	インターフェイスの PPP ステート情報を表示します。 • *type interface-path-id* 引数を入力すると、特定のインターフェイスの PPP 情報が表示されます。 • **brief** キーワードを入力すると、ルータのすべてのインターフェイス、特定のインターフェイス インスタンス、または特定のノードのすべてのインターフェイスの簡易出力が表示されます。 • **all** キーワードを入力すると、ルータにインストールされているすべてのノードの詳細な PPP 情報が表示されます。 • **location** *node-id* キーワード引数を入力すると、指定したノードの詳細な PPP 情報が表示されます。 リンク制御プロトコル（LCP）またはネットワーク制御プロトコル（NCP）に適用される PPP ステートには、7 つのステートがあります。

関連情報

対応する項の説明に従って、PAP、CHAP、または MS-CHAP 認証のパスワードを設定します。

- インターフェイスで PAP をイネーブルにする場合は、「PAP 認証パスワードの設定」（P.642）の説明に従って PAP 認証のユーザ名とパスワードを設定します。
- インターフェイスで CHAP をイネーブルにする場合は、「CHAP 認証パスワードの設定」（P.644）の説明に従って CHAP 認証のユーザ名とパスワードを設定します。
- インターフェイスで MS-CHAP をイネーブルにする場合は、「MS-CHAP 認証パスワードの設定」（P.646）の説明に従って MS-CHAP 認証のユーザ名とパスワードを設定します。

PAP 認証パスワードの設定

ここでは、シリアル インターフェイスまたは POS インターフェイスで PAP 認証をイネーブルにして設定する手順について説明します。

（注） PAP は、POS およびインターフェイスで使用可能な最小のセキュア認証プロトコルです。POS およびインターフェイス経由で送信される情報について、より高レベルのセキュリティを確保するため、PAP 認証に加えて CHAP または MS-CHAP 認証を設定することをお勧めします。

前提条件

「PAP、CHAP、MS-CHAP 認証のイネーブル化」（P.638）の説明に従って、**ppp authentication** コマンドを使用し、インターフェイスで PAP 認証をイネーブルにする必要があります。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp pap sent-username** *username* **password** [**clear** | **encrypted**] *password*
4. **end**
   または
   commit
5. **show running-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**interface** *type interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface serial 0/4/0/1`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ppp pap sent-username** *username* **password** [**clear** | **encrypted**] *password* **例：** `RP/0/RSP0/CPU0:router(config-if)# ppp pap sent-username xxxx password notified`	インターフェイスでリモートのパスワード認証プロトコル（PAP）サポートをイネーブルにし、ピアに対する PAP 認証要求に **sent-username** コマンドと **password** コマンドを含めます。 • *username* 引数を、PAP 認証要求で送信するユーザ名に置き換えます。 • **password clear** を入力してパスワードのクリア テキスト暗号化を選択するか、パスワードがすでに暗号化されている場合は **password encrypted** を入力します。 • **ppp pap sent--username** コマンドを使用すると、複数の username および password コンフィギュレーション コマンドを、インターフェイス上にあるこのコマンドの単一コピーに置き換えることができます。 • **ppp pap sent-username** コマンドは、インターフェイスごとに設定する必要があります。 • リモートの PAP サポートでは、デフォルトでディセーブルになっています。

	コマンドまたはアクション	目的
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	**show running-config** **例：** `RP/0/RSP0/CPU0:router# show running-config`	PPP カプセル化がイネーブルになっているインターフェイスの PPP 認証情報を確認します。

CHAP 認証パスワードの設定

ここでは、CHAP 認証をイネーブルにし、シリアル インターフェイスまたは POS インターフェイスで CHAP パスワードを設定する手順について説明します。

前提条件

「PAP、CHAP、MS-CHAP 認証のイネーブル化」（P.638）の説明に従って、**ppp authentication** コマンドを使用し、インターフェイスで CHAP 認証をイネーブルにする必要があります。

制約事項

両ホストのエンドポイントに同じ CHAP パスワードを設定する必要があります。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp chap password** [**clear** | **encrypted**] *password*
4. **end**
   または
   commit

5. **show running-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface serial 0/4/0/1`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ppp chap password** [**clear** | **encrypted**] *password* **例：** `RP/0/RSP0/CPU0:router(config-if)# ppp chap password clear xxxx`	指定したインターフェイスで CHAP 認証をイネーブルにし、インターフェイス固有の CHAP パスワードを定義します。 • **clear** を入力してクリア テキスト暗号化を選択するか、パスワードがすでに暗号化されている場合は **encrypted** を入力します。 • *password* 引数を、クリア テキストまたはすでに暗号化されているパスワードに置き換えます。このパスワードは、ルータのコレクション間のセキュアな通信の認証に使用されます。 • **ppp chap password** コマンドはリモート CHAP 認証のみに使用され（ピアに対するルータ認証の場合）、ローカルの CHAP 認証では有効になりません。このコマンドは、このコマンドをサポートしないピアを認証しようとする場合に使用すると便利です（古い Cisco IOS XR ソフトウェア イメージを実行しているルータなど）。 • CHAP シークレット パスワードは、不明なピアからのチャレンジに応答するためにルータによって使用されます。

	コマンドまたはアクション	目的
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	**show running-config** **例：** `RP/0/RSP0/CPU0:router# show running-config`	PPP カプセル化がイネーブルになっているインターフェイスの PPP 認証情報を確認します。

MS-CHAP 認証パスワードの設定

ここでは、MS-CHAP 認証をイネーブルにし、シリアル インターフェイスまたは POS インターフェイスで MS-CHAP パスワードを設定する手順について説明します。

前提条件

「PAP、CHAP、MS-CHAP 認証のイネーブル化」（P.638）の説明に従って、**ppp authentication** コマンドを使用し、インターフェイスで MS-CHAP 認証をイネーブルにする必要があります。

制約事項

両ホストのエンドポイントに同じ MS-CHAP パスワードを設定する必要があります。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp ms-chap password** [**clear** | **encrypted**] *password*
4. **end**
   または
   commit

5. **show running-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface serial 0/4/0/1`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ppp ms-chap password** [**clear** | **encrypted**] *password* **例：** `RP/0/RSP0/CPU0:router(config-if)# ppp ms-chap password clear xxxx`	ルータのコレクションを呼び出すルータをイネーブルにし、共通の Microsoft チャレンジ ハンドシェイク認証（MS-CHAP）シークレット パスワードを設定します。 MS-CHAP シークレット パスワードは、不明なピアからのチャレンジに応答するためにルータによって使用されます。
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	**show running-config** **例：** `RP/0/RSP0/CPU0:router# show running-config`	PPP カプセル化がイネーブルになっているインターフェイスの PPP 認証情報を確認します。

認証プロトコルのディセーブル化

ここでは、次の手順について説明します。

- 「インターフェイスでの PAP 認証のディセーブル化」（P.648）
- 「インターフェイスでの CHAP 認証のディセーブル化」（P.649）
- 「インターフェイスでの MS-CHAP 認証のディセーブル化」（P.651）

インターフェイスでの PAP 認証のディセーブル化

ここでは、シリアル インターフェイスまたは POS インターフェイスで PAP 認証をディセーブルにする手順について説明します。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp pap refuse**
4. **end**
   または
   commit
5. **show running-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface serial 0/4/0/1`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ppp pap refuse** **例：** `RP/0/RSP0/CPU0:router(config-if)# ppp pap refuse`	認証を要求するピアからのパスワード認証プロトコル（PAP）認証を拒否します。 • 発信チャレンジ ハンドシェイク認証プロトコル（CHAP）が（**ppp authentication** コマンドを使用して）設定されている場合、拒否パケットでの認証方式として CHAP が提案されます。 • PAP 認証は、デフォルトではディセーブルに設定されています。

	コマンドまたはアクション	目的
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	**show running-config** **例：** `RP/0/RSP0/CPU0:router# show running-config`	PPP カプセル化がイネーブルになっているインターフェイスの PPP 認証情報を確認します。

インターフェイスでの CHAP 認証のディセーブル化

ここでは、シリアル インターフェイスまたは POS インターフェイスで CHAP 認証をディセーブルにする手順について説明します。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp chap refuse**
4. **end**
   または
   commit
5. **show running-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface serial 0/4/0/1`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ppp chap refuse** **例：** `RP/0/RSP0/CPU0:router(config-if)# ppp chap refuse`	認証を要求するピアからの CHAP 認証を拒否します。指定したインターフェイスで **ppp chap refuse** コマンドを入力すると、CHAP を使用してユーザ認証を強制しようとしたピアの試行はすべて拒否されます。 • CHAP 認証は、デフォルトではディセーブルに設定されています。 • 発信パスワード認証プロトコル（PAP）が（**ppp authentication** コマンドを使用して）設定されている場合、拒否パケットでの認証方式として PAP が提案されます。
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	**show running-config** **例：** `RP/0/RSP0/CPU0:router# show running-config`	PPP カプセル化がイネーブルになっているインターフェイスの PPP 認証情報を確認します。

インターフェイスでの MS-CHAP 認証のディセーブル化

ここでは、シリアル インターフェイスまたは POS インターフェイスで MS-CHAP 認証をディセーブルにする手順について説明します。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ppp ms-chap refuse**
4. **end**
   または
   commit
5. **show running-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface serial 0/4/0/1`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ppp ms-chap refuse** **例：** `RP/0/RSP0/CPU0:router(config-if)# ppp ms-chap refuse`	認証を要求するピアからの MS-CHAP 認証を拒否します。指定したインターフェイスで **ppp chap refuse** コマンドを入力すると、MS-CHAP を使用してユーザ認証を強制しようとしたピアの試行はすべて拒否されます。 • MS-CHAP 認証は、デフォルトではディセーブルに設定されています。 • 発信パスワード認証プロトコル（PAP）が（**ppp authentication** コマンドを使用して）設定されている場合、拒否パケットでの認証方式として PAP が提案されます。

	コマンドまたはアクション	目的
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# end` または `RP/0/RSP0/CPU0:router(config-if)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	**show running-config** **例：** `RP/0/RSP0/CPU0:router# show running-config`	PPP カプセル化がイネーブルになっているインターフェイスの PPP 認証情報を確認します。

マルチリンク PPP の設定

ここでは、次の手順について説明します。

前提条件

MLPPP および LFI は、1 ポート チャネライズド OC-3/STM-1 SPA および 2 ポート チャネライズド OC-12/DS0 SPA でサポートされます。

制約事項

Cisco IOS XR ソフトウェアの MLPPP には、次の制限があります。

- サポートされるのはフル レート T1 のみです。

- バンドルのすべてのリンクは、同じ SPA に属します。
- バンドルのすべてのリンクは、同じ速度で動作する必要があります。
- バンドルごとに最大 10 のリンクがサポートされます。
- ラインカードごとに最大 700 のバンドルがサポートされます。
- システムごとに最大 2600 のバンドルがサポートされます。
- DS0 リンク メンバでは MLPPP インターフェイスはサポートされません。
- T3 チャネルをメンバとする場合、MLPPP インターフェイスはサポートされません。したがって、LFI も T3 チャネルではサポートされません。
- MLPPP バンドルのすべてのシリアル リンクは、マルチリンク インターフェイスの **mtu** コマンドの値を継承します。そのため、MLPPP バンドルのメンバーとして設定する前に、シリアル インターフェイスで **mtu** コマンドを設定しないでください。Cisco IOS XR ソフトウェアは、以下をブロックします。
  - インターフェイスがデフォルト以外の MTU 値で設定されている場合、MLPPP バンドルのメンバーとしてシリアル インターフェイスを設定しようとする処理。
  - MLPPP バンドルのメンバーとして設定されているシリアル インターフェイスの **mtu** コマンド値を変更しようとする処理。

Cisco IOS XR ソフトウェアでのマルチリンク処理は、マルチリンク コントローラと呼ばれるハードウェア モジュールによって制御されます。このコントローラは、ASIC、ネットワーク プロセッサ、CPU の連係動作で成り立ちます。MgmtMultilink コントローラにより、マルチリンク インターフェイスはチャネライズド SPA のシリアル インターフェイスのように動作します。

コントローラの設定

コントローラを設定するには、次の作業を行います。

手順の概要

1. **configure**
2. **controller** *type interface-path-id*
3. **mode** *type*
4. **clock source** {**internal** | **line**}
5. **exit**
6. **controller t1** *interface-path-id*
7. **channel-group** *channel-group-number*
8. **timeslots** *range*
9. **exit**
10. **exit**
11. **controller mgmtmultilink** *interface-path-id*
12. **bundle** *bundle-id*
13. **end**
    または
    commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**controller** *type interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t3 0/1/0/0`	コントローラ コンフィギュレーション サブモードを開始して、コントローラ名とインスタンス ID を *rack/slot/module/port* 表記で指定します。
ステップ 3	**mode** *type* **例：** `RP/0/RSP0/CPU0:router# mode t1`	チャネライズするマルチリンクのタイプを設定します（たとえば、28 T1）。
ステップ 4	**clock source** {**internal** | **line**} **例：** `RP/0/RSP0/CPU0:router(config-t3)# clock source internal`	（任意）ポートのクロッキングを設定します。 **（注）** デフォルトのクロック ソースは **internal** です。
ステップ 5	**exit** **例：** `RP/0/RSP0/CPU0:router(config-t3)# exit`	コントローラ コンフィギュレーション モードを終了します。
ステップ 6	**controller t1** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller t1 0/1/0/0/1`	T1 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	**channel-group** *channel-group-number* **例：** `RP/0/RSP0/CPU0:router(config-t1)# channel-group 0`	T1 チャネル グループを作成し、そのチャネル グループのチャネル グループ コンフィギュレーション モードを開始します。チャネル グループ番号は、0 ～ 23 の範囲で設定できます。
ステップ 8	**timeslots** *range* **例：** `RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 1-24`	1 つまたは複数の DS0 タイムスロットをチャネル グループに関連付け、関連付けたシリアル サブインターフェイスをそのチャネル グループに作成します。 • 範囲は 1 ～ 24 タイムスロットです。 **（注）** タイムスロットの範囲は、1 ～ 24 にする必要があります。これは、結果として構築されるシリアル インターフェイスが MLPPP バンドルに受け入れられるようにするためです。

	コマンドまたはアクション	目的
ステップ 9	**exit** **例：** `RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit`	チャネル グループ コンフィギュレーション モードを終了します。
ステップ 10	**exit** **例：** `RP/0/RSP0/CPU0:router(config-t1)# exit`	T1 コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードを開始します。
ステップ 11	**controller mgmtmultilink** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# controller mgmtmultilink 0/1/0/0`	マルチリンク インターフェイスの管理用にコントローラ コンフィギュレーション サブモードを開始します。コントローラ名とインスタンス ID を *rack/slot/module/port* 表記で指定します。
ステップ 12	**bundle** *bundle-id* **例：** `RP/0/RSP0/CPU0:router(config-mgmtmultilink)# bundle 20`	指定したバンドル ID でマルチリンク インターフェイスを作成します。
ステップ 13	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-t3)# end` または `RP/0/RSP0/CPU0:router(config-t3)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

インターフェイスの設定

インターフェイスを設定するには、次の作業を行います。

制約事項

- MLPPP バンドルのすべてのシリアル リンクは、マルチリンク インターフェイスの **mtu** コマンドの値を継承します。そのため、MLPPP バンドルのメンバーとして設定する前に、シリアル インターフェイスで **mtu** コマンドを設定しないでください。Cisco IOS XR ソフトウェアは、以下をブロックします。
  - インターフェイスがデフォルト以外の MTU 値で設定されている場合、MLPPP バンドルのメンバーとしてシリアル インターフェイスを設定しようとする処理。
  - MLPPP バンドルのメンバーとして設定されているシリアル インターフェイスの **mtu** コマンド値を変更しようとする処理。

手順の概要

1. **configure**
2. **interface multilink** *interface-path-id*
3. **ipv4 address** *address/mask*
4. **multilink fragment-size** *bytes*
   または
   multilink fragment delay *delay-ms*
5. **keepalive** {*interval* | **disable**}[*retry*]
6. **exit**
7. **interface** *type interface-path-id*
8. **encapsulation** *type*
9. **multilink group** *group-id*
10. **end**
    または
    commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface multilink** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface multilink 0/1/0/0/1`	マルチリンク インターフェイス名とインスタンス ID を *rack/slot/module/port/bundle-id* 表記で指定して、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。

	コマンドまたはアクション	目的
ステップ 3	**ipv4 address** *ip-address* **例：** `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 80.170.0.1/24`	次の形式でインターフェイスに IP アドレスおよびサブネット マスクを割り当てます。 *A.B.C.D/prefix* または *A.B.C.D/mask*
ステップ 4	**multilink fragment-size** *bytes* または **multilink fragment delay** *delay-ms* **例：** `RP/0/RSP0/CPU0:router(config-if)# multilink fragment-size 350` または `RP/0/RSP0/CPU0:router(config-if)# multilink fragment delay 2`	（任意）マルチリンク フラグメントのサイズを指定します（128 バイトなど）。フラグメント サイズによっては、サポートされない場合があります。デフォルトは no fragments です。 または （任意）ミリ秒単位でのマルチリンク フラグメント遅延を指定します。これは、個々のメンバ リンク（帯域幅 1536000bps/192000Bps の T1）の送信時間遅延と同じ長さになるように、MLPPP フラグメント サイズを設定します。 ユーザが **fragment delay 2** を指定する場合、フラグメント サイズは（192000*0.002）=384B です。このコマンドの使用は **fragment size** での使用に限定されます。どちらのコマンドも、他方よりも優先されます。
ステップ 5	**keepalive** {*interval* | **disable**}[*retry*] **例：** `RP/0/RSP0/CPU0:router(config-if)# keepalive disable`	チャネルのキープアライブ タイマーを設定します。ここで、 • *interval*：キープアライブ メッセージ間の秒数（1 ～ 30）。デフォルトは 10 です。 • **disable**：キープアライブ タイマーをオフにします。 • *retry*：（任意）リンクがダウン状態に遷移する前に、応答なしでピアに送信できるキープアライブ メッセージの数（1 ～ 255）。デフォルトは 3 です。 **（注）** Cisco IOS デバイスによっては、そのデバイスに接続するにはマルチリンク キープアライブを両方のデバイスでディセーブルにする必要があります。
ステップ 6	**exit** **例：** `RP/0/RSP0/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、グローバル コンフィギュレーション モードに入ります。
ステップ 7	**interface** *type interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface serial 0/1/0/0/1:0`	インターフェイス名とインスタンス ID を *rack/slot/module/port/t1-number:channel-group* 表記で指定して、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。
ステップ 8	**encapsulation** *type* **例：** `RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp`	カプセル化のタイプを指定します。ここでは、PPP を指定します。

	コマンドまたはアクション	目的
ステップ 9	**multilink group** *group-id* **例：** `RP/0/RSP0/CPU0:router(config-if)# multilink group 20`	このインターフェイスのマルチリンク グループ ID を指定します。
ステップ 10	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-t3)# end` または `RP/0/RSP0/CPU0:router(config-t3)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

MLPPP オプション機能の設定

次のいずれかのオプション機能を設定するには、次のタスクを実行します。

- アクティブ リンクの最大数
- マルチリンク インターリーブ

（注） アクティブ リンクの最大数は、両方のエンドポイントで設定する必要があります。

手順の概要

1. **configure**
2. **interface multilink** *interface-path-id*
3. **multilink**
4. **ppp multilink minimum-active links** *value*
5. **multilink interleave**
6. **no shutdown**

7. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface multilink** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface multilink 0/1/0/0/1`	マルチリンク インターフェイス名とインスタンス ID を *rack/slot/module/port/bundle-id* 表記で指定して、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。
ステップ 3	**multilink** **例：** `RP/0/RSP0/CPU0:router(config-if)# multilink`	インターフェイス マルチリンク コンフィギュレーション モードを開始します。
ステップ 4	**ppp multilink minimum-active links** *value* **例：** `RP/0/RSP0/CPU0:router(config-if-multilink)# ppp multilink minimum-active links 12`	（任意）マルチリンク インターフェイスのアクティブ リンクの最小数を指定します。 **（注）** リンクの LNM しきい値を超えたとき、MLPPP バンドルのリンクを削除するように PPP にシグナリングするようにノイズ属性のサポートが設定されている場合、リンクはこの miminum-active しきい値未満では削除されません。
ステップ 5	**multilink interleave** **例：** `RP/0/RSP0/CPU0:router(config-if-multilink)# multilink interleave`	（任意）マルチリンク インターフェイスでインターリーブをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 6	**no shutdown** **例：** `RP/0/RSP0/CPU0:router(config-if-mutlilink)# no shutdown`	shutdown 設定を削除します。 • shutdown 設定を削除すると、コントロールに強制された管理上のダウンが解除され、コントローラを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 7	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-t3)# end` または `RP/0/RSP0/CPU0:router(config-t3)# commit`	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` – **yes** と入力すると、実行コンフィギュレーション ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

PPP および MLPPP の ICSSO の設定

この項では、次の ICSSO の設定手順について説明します。

- 「前提条件」（P.660）
- 「制約事項」（P.661）
- 「基本 ICSSO 実装の設定」（P.661）
- 「MR-APS の設定」（P.662）
- 「シリアルおよびマルチリンク インターフェイスの SSRP の設定」（P.664）

前提条件

Cisco ASR 9000 シリーズ ルータは、次の MR-APS、最小装置、ハードウェア構成で ICSSO をサポートします。

- 6 スロットまたは 8 スロット シャーシ 2 台
- ルート/スイッチ プロセッサ（RSP）4 台、シャーシあたり 2 台（信頼性を高める）
- 2 つの 20G SIP、シャーシごとに 1 つ
- 次の SPA タイプのうち 2 つ、シャーシごとに 1 つ

  - 2 ポート チャネライズド OC-12/DS0 SPA
  - 4 ポート チャネライズド T3 SPA
  - 8 ポート チャネライズド T1/E1 SPA
- 2 つの 40 ギガビット イーサネット ラインカード、シャーシごとに 2 つ
- 2 つの 4 ポート 10 ギガビット イーサネット ラインカード、シャーシごとに 1 つ
- 1 ポート チャネライズド OC-3/STM-1 SPA（SPA-1XCHSTM1/OC3）

制約事項

次の制約事項は、PPP および MLPPP の ICSSO に適用されます。

- ICSSO は 2 つの独立したルータだけでサポートされます。
  同じルータ上の 2 枚のラインカードに対しては、ICSSO はサポートされません。
- ICSSO ピア ルータ間の IOS XR システム設定の自動同期または検証は利用できません。
- 次の制約事項は、2 ポート チャネライズド OC-12/DS0 SPA の ICSSO に適用されます。
  - ICSSO は、T1/T3 PPP および T1/MLPPP インターフェイスだけでサポートされます。
  - T1 メンバ リンクは、同じ SPA で終端する必要があります。
  - MR-APS で保護されている MLPPP バンドルのメンバ リンクはすべて、MR-APS 保護ペアの一部である同じ SONET ポートに含まれている必要があります。
  - OC-12 SONET インターフェイス上の T1/PPP、T3/PPP および MLPPP カプセル化されたインターフェイスは保護できます。
- 次の制約事項は、1 ポート チャネライズド T3 SPA の ICSSO に適用されます。
  - T3、T1、E1 チャネルだけの PPP でサポートされます。
  - E1 チャネルだけの MLPPP のメンバ リンクでサポートされます。
- 次の制約事項は、8 ポート チャネライズド T1/E1 SPA の ICSSO に適用されます。
  - T1 および E1 チャネルだけの PPP でサポートされます。
  - E1 チャネルだけの MLPPP のメンバ リンクでサポートされます。

基本 ICSSO 実装の設定

ICSSO の単純バージョンを設定するには、次の手順を使用します。

手順の概要

1. **config**
2. **redundancy**
3. **multi-router aps**
4. **group** *group_number*
5. **controller sonet** *path*
6. **member ipv4** *address* **backup-interface**
7. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**config** **例：** `RP/0/RSP0/CPU0:router# config`	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	**redundancy** **例：** `RP/0/RSP0/CPU0:router(config)# redundancy`	冗長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	**multi-router aps** **例：** `RP/0/RSP0/CPU0:router(config-redundancy)# multi-router aps`	Multi-Router APS 冗長を設定して、APS 冗長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	**group** *group_number* **例：** `RP/0/RSP0/CPU0:router(config-redundancy-aps)# group 1`	APS 冗長グループを設定し、グループ番号を割り当てます。
ステップ 5	**controller sonet** *path* **例：** `RP/0/RSP0/CPU0:router(config-redundancy-aps-group)# controller sonet 0/1/0/0`	APS 冗長バックアップとして SONET コントローラを指定します。
ステップ 6	**member ipv4** *address* **backup-interface type** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config-redundancy-group-controller)# member ipv4 10.10.10.10 backup-interface GigabitEthernet 0/6/0/1`	IP-FRR で使用されるバックアップ インターフェイスの IP アドレスを指定します。
ステップ 7	**commit** **例：** `RP/0/RSP0/CPU0:router(config-redundancy-group-controller)# commit`	設定を保存します。
ステップ 8	**show running config** **例：** `RP/0/RSP0/CPU0:router# show running config`	設定を確認するために MR-APS、SONET コントローラおよび IP アドレス情報を含むルータの現在の設定を表示します。

MR-APS の設定

MR-APS を設定するには、次の手順に従います。

手順の概要

1. **config**
2. **aps group** *number*
3. **channel {0 | 1} remote** *ip-address*
4. **channel {0 | 1} local sonet** *interface-path-id*
5. **exit**
6. **aps rprplus**
7. **interface GigabitEthernet** *interface-path-id*
8. **description** *text*
9. **ipv4 address** *ipv4-address mask*
10. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**config** **例：** `RP/0/RSP0/CPU0:router# config`	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	**aps group** *number* **例：** `RP/0/RSP0/CPU0:router(config)# aps group 1`	自動保護スイッチング（APS）グループを追加して、APS 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 3	**channel {0 | 1} remote** *ip-address* **例：** `RP/0/RSP0/CPU0:router(config-aps)# channel 0 remote 99.10.1.2`	リモート ルータに物理的に配置されたポートとインターフェイスを SONET APS チャネルとして割り当てます。 • 0 は保護チャネルにチャネルを指定します。 • 1 は現用チャネルとしてチャネルを指定します。
ステップ 4	**channel {0 | 1} local sonet** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config-aps)# channel 1 local SONET 0/1/0/0`	ローカル SONET 物理ポートを SONET APS チャネルとして割り当てます。 • 0 は保護チャネルにチャネルを指定します。 • 1 は現用チャネルとしてチャネルを指定します。
ステップ 5	**exit** **例：** `RP/0/RSP0/CPU0:router(config-aps)# exit`	前のモードに戻ります。
ステップ 6	**aps rprplus** **例：** `RP/0/RSP0/CPU0:router(config-aps)# aps rprplus`	スイッチオーバーの APS ホールド タイマーを拡張します。

	コマンドまたはアクション	目的
ステップ 7	**interface GigabitEthernet** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/6/0/0`	ギガビット イーサネット インターフェイスを MR-APS ピアへのパスとして作成し、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	**description** *text* **例：** `RP/0/RSP0/CPU0:router(config-if)# description MR-APS PGP interface for aps group 1`	このインターフェイスにテキスト説明を追加します。
ステップ 9	**ipv4 address** *ipv4-address mask* **例：** `RP/0/RSP0/CPU0:router(config-if )# ipv4 address 99.10.1.1 255.255.255.0`	インターフェイスのプライマリ IPv4 アドレスとサブネット マスクを設定します。
ステップ 10	**commit** **例：** `RP/0/RSP0/CPU0:router(config-if)# commit`	現在の設定を保存します。

シリアルおよびマルチリンク インターフェイスの SSRP の設定

シリアルおよびマルチリンク インターフェイスの SSRP を設定するには、次の手順を実行します。

手順の概要

1. **config**
2. **ssrp profile** ***profile-name***
3. **peer ipv4 address** ***A.B.C.D***
4. **exit**
5. **ssrp location** ***node_id***
6. **group** ***group-id*** **profile** ***profile_name***
7. **group** ***group-id*** **profile** ***profile_name***
8. **exit**
9. **interface serial** *interface-path-id*
10. **ssrp group** ***group-number*** **id** ***id-number*** **ppp**
11. **encapsulation ppp**
12. **multilink**
13. **group** ***group-id***
14. **exit**
15. **keepalive disable**
16. **exit**

17. **interface serial** *interface-path-id*
18. **ssrp group** ***group-number*** **id** ***id-number*** **ppp**
19. **encapsulation ppp**
20. **multilink**
21. **group** ***group-id***
22. **exit**
23. **keepalive disable**
24. **exit**
25. **interface multilink interface-path-id**
26. **ipv4 address** *ipv4-address mask*
27. **ssrp group** ***group-number*** **id** ***id-number*** **ppp**
28. **encapsulation ppp**
29. **shutdown**
30. **keepalive disable**
31. **exit**
32. **controller MgmtMultilink** *interface-path-id*
33. **bundle** *bundleID*
34. **bundle** *bundleID*
35. **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**config** **例：** `RP/0/RSP0/CPU0:router# config`	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	**ssrp profile** *profile-name* **例：** `RP/0/RSP0/CPU0:router(config)# ssrp profile Profile_1`	セッション状態冗長プロトコル（SSRP）プロファイルを設定し、SSRP 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	**peer ipv4 address** *A.B.C.D* **例：** `RP/0/RSP0/CPU0:router(config)# peer ipv4 address 10.10.10.10`	セッション状態冗長プロトコル（SSRP）ピアの IPv4 アドレスを設定します。
ステップ 4	**exit** **例：** `RP/0/RSP0/CPU0:router(config-aps)# exit`	前のモードに戻ります。

	コマンドまたはアクション	目的
ステップ 5	**ssrp location** *node_id* **例：** RP/0/RSP0/CPU0:router(config)# ssrp location 0/1/CPU0	セッション状態冗長プロトコル（SSRP）グループを作成するノードを指定し、SSRP ノード 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	**group** *group-id* **profile** *profile_name* **例：** RP/0/RSP0/CPU0:router(config-ssrp)# group 1 profile Profile_1	セッション状態冗長プロトコル（SSRP）グループを作成し、プロファイルに関連付けます。
ステップ 7	**group** *group-id* **profile** *profile_name* **例：** RP/0/RSP0/CPU0:router(config-ssrp-node)# group 2 profile Profile_2	2 つ目のセッション状態冗長プロトコル（SSRP）グループを作成し、それをプロファイルに関連付けます。
ステップ 8	**exit** **例：** RP/0/RSP0/CPU0:router(config-ssrp-node)# exit	前のモードに戻ります。
ステップ 9	**interface serial** *interface-path-id[.subinterface]* **例：** RP/0/RSP0/CPU0:router(config)# interface serial 0/1/0/0/1/1:0	物理インターフェイスまたは仮想インターフェイス。 **（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。 ルータ構文の詳細については、疑問符（**?**）オンライン ヘルプ機能を使用します。
ステップ 10	**ssrp group** *group-number* **id** *id-number* **ppp** **例：** RP/0/RSP0/CPU0:router(config-if)# ssrp group 1 id 1 ppp	SSRP グループをインターフェイス上でアタッチします。
ステップ 11	**encapsulation ppp** **例：** RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp	ポイントツーポイント プロトコル（PPP）を使用してルータと通信するためのカプセル化をイネーブルにします。
ステップ 12	**multilink** **例：** RP/0/RSP0/CPU0:router(config-if)# multilink	マルチリンク 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 13	**group** *group-id* **例：** RP/0/RSP0/CPU0:router(config-if)# group 1	セッション状態冗長プロトコル（SSRP）グループをこのインターフェイスにアタッチします。

	コマンドまたはアクション	目的
ステップ 14	**exit** **例：** RP/0/RSP0/CPU0:router(config)# exit	前のモードに戻ります。
ステップ 15	**keepalive disable** **例：** RP/0/RSP0/CPU0:router(config)# keepalive disable	このインターフェイスのキープアライブ タイマーをディセーブルにします。
ステップ 16	**exit** **例：** RP/0/RSP0/CPU0:router(config-if)# exit	前のモードに戻ります。
ステップ 17	**interface serial** *interface-path-id[.subinterface]* **例：** RP/0/RSP0/CPU0:router(config)# interface serial 0/1/0/0/1/2:0	物理インターフェイスまたは仮想インターフェイス。 **（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。 ルータ構文の詳細については、疑問符（**?**）オンライン ヘルプ機能を使用します。
ステップ 18	**ssrp group** *group-number* **id** *id-number* **ppp** **例：** RP/0/RSP0/CPU0:router(config-if)# ssrp group 1 id 2 ppp	SSRP グループをインターフェイス上でアタッチします。
ステップ 19	**encapsulation ppp** **例：** RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp	ポイントツーポイント プロトコル（PPP）を使用してルータと通信するためのカプセル化をイネーブルにします。
ステップ 20	**multilink** **例：** RP/0/RSP0/CPU0:router(config-if)# multilink	マルチリンク 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 21	**group** *group-id* **例：** RP/0/RSP0/CPU0:router(config-if)# group 1	セッション状態冗長プロトコル（SSRP）グループをこのインターフェイスにアタッチします。
ステップ 22	**exit** **例：** RP/0/RSP0/CPU0:router(config-if)# exit	前のモードに戻ります。

	コマンドまたはアクション	目的
ステップ 23	**keepalive disable** **例：** `RP/0/RSP0/CPU0:router(config-if)# keepalive disable`	このインターフェイスのキープアライブ タイマーをディセーブルにします。
ステップ 24	**exit** **例：** `RP/0/RSP0/CPU0:router(config-if)# exit`	前のモードに戻ります。
ステップ 25	**interface multilink** *interface-path-id* **例：** `RP/0/RSP0/CPU0:router(config)# interface Multilink 0/1/0/0/1`	物理インターフェイスまたは仮想インターフェイス。 **（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。 ルータ構文の詳細については、疑問符（**?**）オンライン ヘルプ機能を使用します。
ステップ 26	**ipv4 address** *ipv4-address mask* **例：** `RP/0/RSP0/CPU0:router(config-if )# ipv4 address 10.10.10.10 255.255.255.0`	インターフェイスのプライマリ IPv4 アドレスとサブネット マスクを設定します。
ステップ 27	**ssrp group** *group-number* **id** *id-number* **ppp** **例：** `RP/0/RSP0/CPU0:router(config-if)# ssrp group 1 id 3 ppp`	SSRP グループをインターフェイス上でアタッチします。
ステップ 28	**encapsulation ppp** **例：** `RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp`	ポイントツーポイント プロトコル（PPP）を使用してルータと通信するためのカプセル化をイネーブルにします。
ステップ 29	**shutdown** **例：** `RP/0/RSP0/CPU0:router(config-if)# shutdown`	インターフェイスを設定のために管理上のダウン状態にします。
ステップ 30	**keepalive disable** **例：** `RP/0/RSP0/CPU0:router(config-if)# keepalive disable`	このインターフェイスのキープアライブ タイマーをディセーブルにします。
ステップ 31	**exit** **例：** `RP/0/RSP0/CPU0:router(config-if)# exit`	前のモードに戻ります。

	コマンドまたはアクション	目的
ステップ 32	**controller MgmtMultilink** *interface-path-id* **例：** RP/0/RSP0/CPU0:router(config)# controller MgmtMultilink 0/1/0/0	汎用マルチリンク バンドルのコントローラを設定し、MgmtMultilink コンフィギュレーション モードを開始します。
ステップ 33	**bundle** *bundleID* **例：** RP/0/RSP0/CPU0:router(config-mgmtmultilink)# bundle 1	マルチリンク インターフェイス バンドルを作成します。
ステップ 34	**bundle** *bundleID* **例：** RP/0/RSP0/CPU0:router(config-mgmtmultilink)# bundle 2	マルチリンク インターフェイス バンドルを作成します。
ステップ 35	**commit** **例：** RP/0/RSP0/CPU0:router(config-mgmtmultilink)# commit	現在の設定を保存します。

PPP の設定例

ここでは、次の設定例について説明します。

- 「POS インターフェイスでの PPP カプセル化の設定：例」（P.669）
- 「シリアル インターフェイスでの PPP カプセル化の設定：例」（P.670）
- 「PPP および MLPPP の ICSSO の設定：例」（P.670）
- 「マルチリンク PPP 設定の確認」（P.678）

POS インターフェイスでの PPP カプセル化の設定：例

次に、POS インターフェイスを作成し、PPP カプセル化を設定する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface POS 0/3/0/0
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp
RP/0/RSP0/CPU0:router(config-if)# no shutdown
RP/0/RSP0/CPU0:router(config-if)# ppp pap sent-username P1_TEST-8 password xxxx
RP/0/RSP0/CPU0:router(config-if)# ppp authentication chap pap MIS-access
RP/0/RSP0/CPU0:router(config-if)# ppp chap password encrypted xxxx
RP/0/RSP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

次に、最初の認証が失敗した後に 2 回リトライできる（認証が失敗した場合に全部で 3 回リトライできる）ように POS インターフェイス 0/3/0/1 を設定する例を示します。

```
RP/0/RSP0/CPU0:router# configure
```

```
RP/0/RSP0/CPU0:router(config)# interface POS 0/3/0/1
RP/0/RSP0/CPU0:router(config-if)# ppp max-bad-auth 3
```

シリアル インターフェイスでの PPP カプセル化の設定：例

次に、PPP MS-CHAP をカプセル化したシリアル インターフェイスを作成して設定する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface serial 0/3/0/0/0:0
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 172.18.189.38 255.255.255.224
RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp
RP/0/RSP0/CPU0:router(config-if)# no shutdown
RP/0/RSP0/CPU0:router(config-if)# ppp authentication ms-chap MIS-access
RP/0/RSP0/CPU0:router(config-if)# ppp ms-chap password encrypted xxxx
RP/0/RSP0/CPU0:router(config-if)# end
Uncommitted changes found, commit them? [yes]: yes
```

MLPPP の設定：例

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# controller t3 0/1/0/0/1
RP/0/RSP0/CPU0:router# mode t1
RP/0/RSP0/CPU0:router(config-t3)# clock source internal
RP/0/RSP0/CPU0:router(config-t3)# exit
RP/0/RSP0/CPU0:router(config)# controller t1 0/1/0/0/1/1
RP/0/RSP0/CPU0:router(config-t1)# channel-group 0
RP/0/RSP0/CPU0:router(config-t1-channel_group)# timeslots 1-24
RP/0/RSP0/CPU0:router(config-t1-channel_group)# exit
RP/0/RSP0/CPU0:router(config-t1)# exit
RP/0/RSP0/CPU0:router(config)# controller mgmtmultilink 0/1/0/0
RP/0/RSP0/CPU0:router(config-mgmtmultilink)# bundle 20
RP/0/RSP0/CPU0:router(config-t3)# commit
RP/0/RSP0/CPU0:router(config-t3)# exit

RP/0/RSP0/CPU0:router(config)# interface multilink 0/1/0/0/20
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 80.170.0.1/24
RP/0/RSP0/CPU0:router(config-if)# multilink fragment-size 128
RP/0/RSP0/CPU0:router(config-if)# keepalive disable
RP/0/RSP0/CPU0:router(config-if)# exit
RP/0/RSP0/CPU0:router(config)# interface serial 0/1/0/0/1/1:0
RP/0/RSP0/CPU0:router(config-if)# encapsulation ppp
RP/0/RSP0/CPU0:router(config-if)# multilink group 20
RP/0/RSP0/CPU0:router(config-t3)# commit
RP/0/RSP0/CPU0:router(config-t3)# exit

RP/0/RSP0/CPU0:router(config)# interface multilink 0/1/0/0/1
RP/0/RSP0/CPU0:router(config-if)# multilink
RP/0/RSP0/CPU0:router(config-if-multilink)# ppp multilink minimum-active links 10
RP/0/RSP0/CPU0:router(config-if-multilink)# multilink interleave
RP/0/RSP0/CPU0:router(config-if-mutlilink)# no shutdown
RP/0/RSP0/CPU0:router(config-t3)# commit
```

PPP および MLPPP の ICSSO の設定：例

ここでは ICSSO 設定および関連の設定に関する次の例を示します。

- 「ICSSO の設定：例」（P.672）

ICSSO の設定：例

次に、SONET コントローラで ICSSO を設定する例を示します。

```
config
    redundancy
        multi-router aps
        group 1
        controller sonet 0/1/0/0
            member ipv4 10.10.10.10 backup-interface GigabitEthernet 0/6/0/1
            commit
show running config
```

ICSSO とともに使用するためのチャネライズド SONET コントローラの設定：例

次の例では、ICSSO とともに使用するためのチャネライズド SONET コントローラの設定方法を示します。

```
config
    controller SONET0/7/1/0
        framing sonet
        sts 1
        mode t3
!
        sts 2
        mode t3
!
        sts 3
        mode t3
!
    controller T3 0/7/0/1
        mode t1
        framing auto-detect
!
    controller T1 0/7/0/1/1
        channel-group 0
        timeslots 1-24
```

MR-APS の設定：例

次に、MR-APS の設定例を示します。

```
config
    aps group 1
        channel 0 remote 99.10.1.2
        channel 1 local SONET0/1/0/0
!
    aps rprplus
!
    interface GigabitEthernet0/6/0/0
        description MR-APS PGP interface for aps group 1
        ipv4 address 99.10.1.1 255.255.255.0
```

次に、冗長グループ マネージャを設定する例を示します。

```
// mr-aps part:
aps group 1
 channel 0 remote 99.10.1.2
```

```
 channel 1 local SONET0/1/0/0
!
// ssrp part:
ssrp location 0/1/CPU0
 group 1 profile TEST
!
ssrp profile TEST
 peer ipv4 address 99.10.1.2
!
// redundancy group manager part:
redundancy
 multi-router aps
  group 1
   controller SONET0/1/0/0
    member ipv4 99.30.1.2 backup-interface GigabitEthernet0/6/0/4
   !

// ospf part:
router ospf 1
 nsr
 nsf ietf
 redistribute connected instance IPCP
 redistribute static
 area 0
  interface GigabitEthernet0/6/0/4
  !
!
!

show redundancy-group multi-router aps
```

シリアルおよびマルチリンク インターフェイスの SSRP の設定：例

次の例では、SSRP をシリアル インターフェイス（PPP カプセル化あり）とマルチリンク インターフェイスで設定する方法を示します。

```
config
    ssrp profile TEST
        peer ipv4 address 99.10.1.2
!
    ssrp location 0/1/CPU0
        group 1 profile TEST
!
    interface Serial0/1/0/0/1/1:0
        ssrp group 1 id 1 ppp
        encapsulation ppp
        multilink
        group 1
!
    keepalive disable
!
    interface Serial0/1/0/0/1/2:0
        ssrp group 1 id 2 ppp
        encapsulation ppp
        multilink
        group 1
!
    keepalive disable
!
    interface Multilink0/1/0/0/1
```

```
        ipv4 address 51.1.1.1 255.255.255.0
        ssrp group 1 id 3 ppp
        encapsulation ppp
        shutdown
!
keepalive disable
!
    controller MgmtMultilink0/1/0/0
        bundle 1
```

（注） シリアル インターフェイスの設定の詳細については、このマニュアルの *「Cisco ASR 9000 シリーズ ルータでのシリアル インターフェイスの設定」* モジュールを参照してください。

（注） マルチリンクの設定の詳細については、「マルチリンク PPP の設定」（P.652）を参照してください。

ICSSO で使用するマルチリンクの VRF の設定：例

次に、ICSSO で使用するためのマルチリンク インターフェイスの VPN ルーティングおよび転送（VRF）を設定する例を示します。

```
config
    vrf EvDO-vrf
        address-family ipv4 unicast
!
    interface Multilink 0/0/0/0/1
        description To EvDO BTS Number 1
        vrf EvDO-vrf
        ipv4 address 150.0.1.3 255.255.255.0
        encapsulation ppp
!
```

（注） VRF の設定に関する詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*』を参照してください。マルチリンクの設定の詳細については、「マルチリンク PPP の設定」（P.652）を参照してください。

ICSSO で使用するためのイーサネットの VRF の設定：例

次に、ICSSO で使用するためのイーサネット インターフェイスの VPN ルーティングおよび転送（VRF）を設定する例を示します。

```
config
    vrf EvDO-vrf
        address-family ipv4 unicast
!
    interface GigabitEthernet 1/0/0/0.20
        description Inter-ASR9000 EvDO VLAN
        vrf EvDO-vrf
        encapsulation dot1q 20
```

（注） VRF の設定に関する詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*』を参照してください。イーサネットの設定の詳細については、このマニュアルの「Cisco ASR 9000 シリーズ ルータのイーサネット OAM の設定」モジュールを参照してください。

ICSSO で使用する OSPF の設定：例

一連のセル サイトで PPP セッションが終端する集約ルータは、Open Shortest Path First（OSPF）を使用して LAN スイッチに自身のアベイラビリティをアドバタイズします。次に、ICSSO で使用するために OSPF を設定する例を示します。

```
config
    router ospf 1
        nsr
        nsf ietf
        redistribute connected instance IPCP
        redistribute static
        area 0
    interface GigabitEthernet 0/6/0/1
!
```

（注） OSPF の設定方法の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*』を参照してください。

ICSSO 設定の確認：例

次に、ICSSO 設定を確認する例を示します。

- 「SSRP グループの確認：例」（P.675）
- 「ICSSO ステータスの確認：例」（P.676）
- 「MR-APS 設定の確認：例」（P.676）
- 「OSPF 設定の確認：例」（P.677）

SSRP グループの確認：例

次の例では、SSRP グループ設定を確認する方法を示します。

```
RP/0/RSP0/CPU0:Router# show ssrp groups all det loc 0/1/cpu0
```

```
Tue Nov 10 16:57:55.911 UTC

Group ID: 1
  Conn (ACT,SB): UP,UP
  Profile:  TEST
  Peer:     99.10.1.2
  Max-hops: 255
  Sessions: 3
  Channels Created
  Client:            PPP
    Active Init:      TRUE
    Standby Init:     TRUE
    Active State:     IDT-End-Sent
    Standby State:    IDT-End-Received
    Auth-Req Pending: FALSE
    Active   ID Out:              93
    Active   ID In:               93
    Active   Last Reply In:       93
    Active   Counter:              5

    Standby  ID Out:              50
    Standby  ID In:               50
    Standby  Last Reply In:       50
    Standby  Counter:              5

    Session   Interface
    ----------------------------
    1         Se0/1/0/0/1/1:0
    2         Se0/1/0/0/1/2:0
    3         Mu0/1/0/0/1
```

ICSSO ステータスの確認：例

次に、ICSSO ステータスを確認する例を示します。

```
RP/0/RSP0/CPU0:Router# show ppp sso sum loc 0/1/cpu0
Tue Nov 10 16:59:00.253 UTC

Not-Ready    : The session is not yet ready to run as Active or Standby
Stby-UnNegd  : In Standby mode, no replication state received yet
Act-Down     : In Active mode, lower layer not yet up
Deactivating : Session was Active, now going Standby
Act-UnNegd   : In Active mode, not fully negotiated yet
Stby-Negd    : In Standby mode, replication state received and pre-programmed
Activating   : Session was Standby and pre-programmed, now going Active
Act-Negd     : In Active mode, fully negotiated and up
-            : This layer not running

                      Not-   Stby-  Act- Deactiv- Act-   Stby-  Activ- Act-
Layer        | Total  Ready  UnNegd Down   ating  UnNegd Negd   ating  Negd
-------------+------- ------ ------ ------ ------ ------ ------ ------ ------
LCP          |      6      0      0      0      0      0      0      0      6
of-us-auth   |      6      0      0      0      0      0      0      0      6
of-peer-auth |      6      0      0      0      0      0      0      0      6
IPCP         |      2      0      0      0      0      0      0      0      2
```

MR-APS 設定の確認：例

次に、MR-APS の設定を確認する例を示します。

例 1：

```
RP/0/RSP0/CPU0:Router# show redundancy-group multi-router aps all

Tue Nov 10 17:00:14.018 UTC

Interchassis Group: 1
             State: FRR ADD SENT
        Controller: SONET0/1/0/0                        0x2000080
  Backup Interface: GigabitEthernet0/6/0/1              0x10000180
  Next Hop IP Addr: 10.10.10.10

Interchassis Group: Not Configured
             State: WAIT CONFIG
        Controller: SONET0/1/0/1                        0x20003c0
  Backup Interface: None                                0x0
  Next Hop IP Addr: 0.0.0.0
```

例 2：

```
RP/0/RSP0/CPU0:Router# show cef adj rem loc 0/6/cpu0

Tue Nov 10 17:00:30.471 UTC
Display protocol is ipv4
Interface    Address                                          Type    Refcount

SO0/1/0/0    Ifhandle: 0x2000080                              remote  2
             Adjacency: PT:0xa47c9cf4
             Interface: SO0/1/0/0
             Interface Type: 0x0, Base Flags: 0x110000 (0xa4a00494)
             Nhinfo PT: 0xa4a00494, Idb PT: 0xa4cd60d8, If Handle: 0x2000080
             Ancestor If Handle: 0x0

             Protect FRR: 0xa4a8a040
             Backup FRR: 0xa4a89f34
             Backup NH: 0xa4a00a74
             Backup IFH: 0x10000180
             Backup Interface: Gi0/6/0/1
             Backup IP: 10.10.10.10

             FRR Active: 0
```

OSPF 設定の確認：例

次に、OSPF の設定を確認する例を示します。

例 1：

```
RP/0/RSP0/CPU0:Router# show route back
Tue Nov 10 17:01:48.974 UTC

Codes: C - connected, S - static, R - RIP, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, su - IS-IS summary null, * - candidate default
       U - per-user static route, o - ODR, L - local, G  - DAGR
       A - access/subscriber

C    51.1.1.2/32 is directly connected, 00:10:03, Multilink0/1/0/0/1
                 Backup  O E2 [110/20] via 10.10.10.10, GigabitEthernet0/6/0/1
```

```
C    52.1.1.2/32 is directly connected, 00:11:47, Multilink0/1/0/0/2
                 Backup  O E2 [110/20] via 10.10.10.10, GigabitEthernet0/6/0/1
S    110.0.0.2/32 [1/0] via 51.1.1.2, 00:11:40
                 Backup  O E2 [110/20] via 10.10.10.10, GigabitEthernet0/6/0/1
```

例 2：

```
RP/0/RSP0/CPU0:Router# show route 51.1.1.2
Tue Nov 10 17:02:26.507 UTC

Routing entry for 51.1.1.2/32
  Known via "connected IPCP", distance 0, metric 0 (connected)
  Installed Nov 10 16:51:45.703 for 00:10:40
  Routing Descriptor Blocks
    51.1.1.2 directly connected, via Multilink0/1/0/0/1
      Route metric is 0
  No advertising protos.
```

マルチリンク PPP 設定の確認

次のコマンドを使用して、マルチリンク設定を確認し、トラブルシューティングを行うことができます。

- 「show multilink interfaces：例」（P.678）
- 「show ppp interfaces multilink：例」（P.681）
- 「show ppp interface serial：例」（P.681）
- 「show imds interface multilink：例」（P.681）

show multilink interfaces：例

```
RP/0/RSP0/CPU0:Router# show multilink interfaces Serial 0/4/3/1/10:0
Mon Sep 21 09:24:19.604 UTC

Serial0/4/3/1/10:0 is up, line protocol is up
      Encapsulation: PPP
      Multilink group id: 6
      Member status: ACTIVE

RP/0/RSP0/CPU0:Router# show multilink interfaces Multilink 0/4/3/0/3
Mon Sep 21 09:17:12.131 UTC

Multilink0/4/3/0/3 is up, line protocol is up
  Fragmentation: disabled
  Interleave: disabled
  Encapsulation: PPP
  Member Links: 1 active, 1 inactive
    - Serial0/4/3/1/5:0 is up, line protocol is up
      Encapsulation: PPP
      Multilink group id: 3
      Member status: ACTIVE

    - Serial0/4/3/1/6:0 is administratively down, line protocol is administratively down
Encapsulation: PPP
      Multilink group id: 3
      Member status: INACTIVE : LCP has not been negotiated

  Fragmentation Statistics
```

```
  Input Fragmented packets 0            Input Fragmented bytes 0
  Output Fragmented packets 0           Output Fragmented bytes 0
  Input Unfragmented packets 0          Input Unfragmented bytes 0
  Output Unfragmented packets 0         Output Unfragmented bytes 0
  Input Reassembled packets 0           Input Reassembled bytes 0

RP/0/5/CPU0:Mav-IOX-Rahul#sho multilink interfaces Serial 0/4/3/1/10:0
Mon Sep 21 09:24:19.604 UTC

Serial0/4/3/1/10:0 is up, line protocol is up
      Encapsulation: PPP
      Multilink group id: 6
      Member status: ACTIVE

RP/0/RSP0/CPU0:Router# show multilink interfaces
Mon Sep 21 09:15:10.679 UTC

Multilink0/4/3/0/1 is up, line protocol is up
  Fragmentation: disabled
  Interleave: disabled
  Encapsulation: FR
  Member Links: 1 active, 1 inactive
   - Serial0/4/3/1/2:0: INACTIVE : Down  (Member link idle)
   - Serial0/4/3/1/1:0: ACTIVE : Up


Multilink0/4/3/0/10 is up, line protocol is down
  Fragmentation: disabled
  Interleave: disabled
  Encapsulation: PPP
  Member Links: 0 active, 0 inactive
  Fragmentation Statistics
  Input Fragmented packets 0            Input Fragmented bytes 0
  Output Fragmented packets 0           Output Fragmented bytes 0
  Input Unfragmented packets 0          Input Unfragmented bytes 0
  Output Unfragmented packets 0         Output Unfragmented bytes 0
  Input Reassembled packets 0           Input Reassembled bytes 0

Multilink0/4/3/0/100 is administratively down, line protocol is administratively down
  Fragmentation: disabled
  Interleave: disabled
  Encapsulation: PPP
  Member Links: 0 active, 0 inactive
  Fragmentation Statistics
  Input Fragmented packets 0            Input Fragmented bytes 0
  Output Fragmented packets 0           Output Fragmented bytes 0
  Input Unfragmented packets 0          Input Unfragmented bytes 0
  Output Unfragmented packets 0         Output Unfragmented bytes 0
  Input Reassembled packets 0           Input Reassembled bytes 0

Multilink0/4/3/0/2 is up, line protocol is up
  Fragmentation: disabled
  Interleave: disabled
  Encapsulation: FR
  Member Links: 2 active, 0 inactive
   - Serial0/4/3/1/4:0: ACTIVE : Up
   - Serial0/4/3/1/3:0: ACTIVE : Up


Multilink0/4/3/0/3 is up, line protocol is up
  Fragmentation: disabled
  Interleave: disabled
  Encapsulation: PPP
  Member Links: 1 active, 1 inactive
```

```
    - Serial0/4/3/1/5:0: ACTIVE
    - Serial0/4/3/1/6:0: INACTIVE : LCP has not been negotiated
   Fragmentation Statistics
   Input Fragmented packets 0            Input Fragmented bytes 0
   Output Fragmented packets 0           Output Fragmented bytes 0
   Input Unfragmented packets 0          Input Unfragmented bytes 0
   Output Unfragmented packets 0         Output Unfragmented bytes 0
   Input Reassembled packets 0           Input Reassembled bytes 0

Multilink0/4/3/0/4 is up, line protocol is up
   Fragmentation: disabled
   Interleave: disabled
   Encapsulation: PPP
   Member Links: 2 active, 0 inactive
    - Serial0/4/3/1/8:0: ACTIVE
    - Serial0/4/3/1/7:0: ACTIVE
   Fragmentation Statistics
   Input Fragmented packets 0            Input Fragmented bytes 0
   Output Fragmented packets 0           Output Fragmented bytes 0
   Input Unfragmented packets 0          Input Unfragmented bytes 0
   Output Unfragmented packets 0         Output Unfragmented bytes 0
   Input Reassembled packets 0           Input Reassembled bytes 0

Multilink0/4/3/0/5 is up, line protocol is up
   Fragmentation: disabled
   Interleave: enabled
   Encapsulation: PPP
   Member Links: 1 active, 0 inactive
    - Serial0/4/3/1/9:0: ACTIVE
   Fragmentation Statistics
   Input Fragmented packets 0            Input Fragmented bytes 0
   Output Fragmented packets 0           Output Fragmented bytes 0
   Input Unfragmented packets 0          Input Unfragmented bytes 0
   Output Unfragmented packets 0         Output Unfragmented bytes 0
   Input Reassembled packets 0           Input Reassembled bytes 0

Multilink0/4/3/0/6 is up, line protocol is up
   Fragmentation: disabled
   Interleave: enabled
   Encapsulation: PPP
   Member Links: 1 active, 0 inactive
    - Serial0/4/3/1/10:0: ACTIVE
   Fragmentation Statistics
   Input Fragmented packets 0            Input Fragmented bytes 0
   Output Fragmented packets 0           Output Fragmented bytes 0
   Input Unfragmented packets 0          Input Unfragmented bytes 0
   Output Unfragmented packets 0         Output Unfragmented bytes 0
   Input Reassembled packets 0           Input Reassembled bytes 0

Multilink0/4/3/0/7 is up, line protocol is down
   Fragmentation: disabled
   Interleave: enabled
   Encapsulation: PPP
   Member Links: 0 active, 1 inactive
    - Serial0/4/3/1/11:0: INACTIVE : LCP has not been negotiated
   Fragmentation Statistics
   Input Fragmented packets 0            Input Fragmented bytes 0
   Output Fragmented packets 0           Output Fragmented bytes 0
   Input Unfragmented packets 0          Input Unfragmented bytes 0
   Output Unfragmented packets 0         Output Unfragmented bytes 0
   Input Reassembled packets 0           Input Reassembled bytes 0

Multilink0/4/3/0/8 is up, line protocol is down
   Fragmentation: disabled
```

```
  Interleave: enabled
  Encapsulation: PPP
  Member Links: 0 active, 1 inactive
   - Serial0/4/3/1/12:0: INACTIVE : LCP has not been negotiated
  Fragmentation Statistics
  Input Fragmented packets 0            Input Fragmented bytes 0
  Output Fragmented packets 0           Output Fragmented bytes 0
  Input Unfragmented packets 0          Input Unfragmented bytes 0
  Output Unfragmented packets 0         Output Unfragmented bytes 0
  Input Reassembled packets 0           Input Reassembled bytes 0
```

show ppp interfaces multilink：例

```
RP/0/RSP0/CPU0:Router# show ppp interfaces multilink 0/3/1/0/1

Multilink 0/3/1/0/1 is up, line protocol is up
LCP: Open
     Keepalives disabled
  IPCP: Open
     Local IPv4 address: 1.1.1.2
     Peer IPv4 address:  1.1.1.1
  Multilink
     Member Links: 2 active, 1 inactive (min-active 1)
      - Serial0/3/1/0/0:0: ACTIVE
      - Serial0/3/1/0/1:0: ACTIVE
      - Serial0/3/1/0/2:0: INACTIVE : LCP has not been negotiated
```

show ppp interface serial：例

```
RP/0/RSP0/CPU0:Router# show ppp interface Serial 0/3/1/0/0:0

Serial 0/3/1/0/0:0 is up, line protocol is up
  LCP: Open
     Keepalives disabled
     Local MRU: 1500 bytes
     Peer  MRU: 1500 bytes
     Local Bundle MRRU: 1596 bytes
     Peer  Bundle MRRU: 1500 bytes
     Local Endpoint Discriminator: 1b61950e3e9ce8172c8289df0000003900000001
     Peer  Endpoint Discriminator: 7d046cd8390a4519087aefb90000003900000001
  Authentication
     Of Peer: <None>
     Of Us:   <None>
  Multilink
     Multilink group id: 1
     Member status: ACTIVE
```

show imds interface multilink：例

```
RP/0/RSP0/CPU0:Router# show imds interface Multilink 0/3/1/0/1

IMDS INTERFACE DATA (Node 0x0)

Multilink0_3_1_0_1 (0x04001200)
-----------------------
flags: 0x0001002f    type: 55 (IFT_MULTILINK)    encap: 52 (ppp)
state: 3 (up)     mtu: 1600    protocol count: 3
control parent: 0x04000800     data parent: 0x00000000
       protocol         capsulation           state        mtu
```

```
--------------- -------------------- --------------- --------
12 (ipv4)
                26 (ipv4)            3 (up)          1500
                47 (ipcp)            3 (up)          1500
16 (ppp_ctrl)
                53 (ppp_ctrl)        3 (up)          1500
0 (Unknown)
                139 (c_shim)         3 (up)          1600
                52 (ppp)             3 (up)          1504
                56 (queue_fifo)      3 (up)          1600
                60 (txm_nopull)      3 (up)          1600
```

その他の関連資料

ここでは、PPP カプセル化に関する参考資料について説明します。

関連資料

関連項目	参照先
Cisco IOS XR マスター コマンド リファレンス	*『Cisco IOS XR Master Commands List』*
Cisco IOS XR インターフェイス コンフィギュレーション コマンド	*『Cisco IOS XR Interface and Hardware Component Command Reference』*
Cisco IOS XR ソフトウェアを使用した初期システム ブートアップとルータの設定情報	*『Cisco IOS XR Getting Started Guide』*
Cisco IOS XR AAA サービス構成情報	*『Cisco IOS XR System Security Configuration Guide』*および *『Cisco IOS XR System Security Command Reference』*

標準

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

MIB

MIB	MIB のリンク
この機能によってサポートされる新しい MIB または変更された MIB はありません。またこの機能による既存 MIB のサポートに変更はありません。	Cisco IOS XR ソフトウェアを使用して選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
RFC-1661	*『The Point-to-Point Protocol (PPP)』*
RFC- 1994	*『PPP Challenge Handshake Authentication Protocol (CHAP)』*

シスコのテクニカル サポート

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

その他の関連資料

Cisco ASR 9000 シリーズ ルータでの 802.1Q VLAN インターフェイスの設定

このモジュールでは、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータの 802.1Q VLAN インターフェイスの設定と管理について説明します。

IEEE 802.1Q 仕様は、VLAN メンバーシップ情報のあるタグ付きイーサネット フレームの標準方式を確立し、ブリッジド LAN インフラストラクチャ内にある VLAN トポロジーの定義、操作、および管理ができる VLAN ブリッジの動作を定義します。

802.1Q 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することを目的としています。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 VLAN インターフェイス設定の機能履歴

リリース	変更内容
リリース 3.7.2	この機能は、Cisco ASR 9000 シリーズ ルータで導入されました。
リリース 3.9.0	レイヤ 2 dot1q が更新されました。カプセル化 dot1q が追加されました。

内容

802.1Q VLAN インターフェイス設定の前提条件

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。

802.1Q VLAN インターフェイスを設定する前に、次の条件を満たしていることを確認してください。

- ギガビット イーサネット インターフェイス、10 ギガビット イーサネット インターフェイス、またはイーサネット バンドル インターフェイスの設定が完了している必要があります。

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おく必要があります。

802.1Q VLAN の概要

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

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 IOS XR ソフトウェアは、ギガビット イーサネットおよび 10 ギガビット イーサネット インターフェイスでの VLAN サブインターフェイスの設定をサポートしています。

802.1Q タグ付きフレーム

IEEE 802.1Q タグ ベースの VLAN は、MAC ヘッダーの特別なタグを使用し、ブリッジでのフレームの VLAN メンバーシップを識別できます。このタグは、VLAN および Quality of Service（QoS）のプライオリティの識別に使用されます。VLAN は、手動での入力によってスタティックに作成することも、Generic Attribute Registration Protocol（GARP）VLAN Registration プロトコル（GVRP）を介してダイナミックに作成することもできます。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 VLAN インターフェイスの CFM

802.1Q VLAN インターフェイスをモニタするための接続障害管理（CFM）の設定方法は、イーサネット インターフェイスをモニタするための CFM の設定と同じです。

イーサネット インターフェイスの CFM の設定については、「Cisco ASR 9000 シリーズ ルータのイーサネット OAM の設定」モジュールの次のセクションを参照してください。

- 「イーサネット CFM」（P.68）
- 「イーサネット CFM の設定」（P.109）
- 「イーサネット CFM サービスの設定：例」（P.167）
- 「イーサネット CFM の show コマンド：例」（P.168）

サブインターフェイス

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

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

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

サブインターフェイス MTU

サブインターフェイスの最大伝送単位（MTU）は、物理インターフェイスから継承されます。これには、802.1Q VLAN タグに許可されている追加の 4 バイトも含まれます。

ネイティブ VLAN

Cisco ASR 9000 シリーズ ルータは、ネイティブ VLAN をサポートしません。ただし、同等の機能は、次のように **encapsulation** コマンドを使用して達成されます。

```
encapsulation dot1q TAG-ID, untagged
```

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。Cisco ASR 9000 シリーズ ルータでは、EFP は VLAN カプセル化を使用した L2 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。

VLAN インターフェイスでのレイヤ 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、L2VPN 操作の 3 つ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。
- Q-in-Any AC：AC は、内部 VLAN タグが L3 終端でない限り、特定の外部 VLAN タグおよび任意の内部 VLAN タグで送受信されるすべてのフレームに対応します。Q-in-Any は、ワイルドカード化を使用して任意の 2 番目のタグに一致させる QinQ の拡張です。

（注） Q-in-Any モードは、基本の Dot1Q モードを変化させたものです。Q-in-Any モードでは、フレームは基本の QinQ カプセル化が行われていますが、Q-in-Any モードでは内部タグは意味を持ちません。ただし、いくつかの特定の内部 VLAN タグは、特定のサービスに使用されます。たとえば、一般的なインターネット アクセスに L3 サービスを提供するために、あるタグが使用されることがあります。

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 に L2VPN を設定するには、「VLAN での接続回線の設定」（P.691）を参照してください。

VLAN に L2VPN を設定する場合は、次の事項に注意する必要があります。

- Cisco IOS XR ソフトウェアは LC ごとに 4k AC をサポートします。
- ポイントツーポイント接続では、2 つの AC を同じタイプにするべきではありません。たとえば、ポート モードのイーサネット AC を、Dot1Q イーサネット AC に接続することができます。
- 疑似回線は、VLAN モードまたはポート モードで実行できます。VLAN モードで実行される疑似回線に単一の Dot1Q タグを設定することができますが、ポート モードで実行される疑似回線にタグを設定することはできません。これらの異なるタイプの回路を接続するには、インターワーキングが必要です。この場合のインターワーキングは、タグのポップ、プッシュ、書き換えの形を取ります。レイヤ 2 VPN を使用するメリットは、まったく異なるタイプのメディアを接続するのに必要なインターワーキングを簡素化できることにあります。
- MPLS 疑似回線の両側にある AC は、異なるタイプでもかまいません。この場合は、AC の一方または両方のエンドで、疑似回線接続を行うための適切な変換が行われます。

AC と疑似回線の情報を表示するには、**show interfaces** コマンドを使用します。

（注） L2VPN ネットワークの設定の詳細については、『*Cisco ASR 9000 Series Router Multiprotocol Label Switching Configuration Guide*』の「*Implementing MPLS Layer 2* VPNs」モジュールを参照してください。

他のレイヤ 2 VPN 機能

次のレイヤ 2 VPN 機能の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide*』および『*Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Command Reference*』を参照してください。

- プロバイダー バックボーン ブリッジ（PBB）802.1ah
- Policy-Based Forwarding（PBF）
- MVRP 802.1（MVRP-lite）

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」（P.689）
- 「VLAN での接続回線の設定」（P.691）
- 「802.1Q VLAN サブインターフェイスの削除」（P.694）

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、このモジュールの「802.1Q VLAN サブインターフェイスの削除」を参照してください。

手順の概要

1. **configure**
2. **interface** {**GigabitEthernet** | **TenGigE** |**Bundle-Ether**} *interface-path-id.subinterface*
3. **encapsulation dot1q**
4. **ipv4 address** *ip-address mask*
5. **exit**
6. ステップ 2 ～ 5 を繰り返し、残りの VLAN サブインターフェイスを定義します。
7. **end**
   または
   commit
8. **show ethernet trunk bundle-ether** *instance*（任意）

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** {**GigabitEthernet** | **TenGigE** | **Bundle-Ether**} *interface-path-id.subinterface* **例：** `RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/2/0/4.10`	サブインターフェイス コンフィギュレーション モードを開始し、インターフェイス タイプ、ロケーション、サブインターフェイス番号を指定します。 • *interface-path-id* 引数を、次のいずれかのインスタンスに置き換えます。 – 物理イーサネット インターフェイス インスタンスまたはイーサネット バンドル インスタンス。名前の表記は *rack*/*slot*/*module*/*port* の形式で、表記の一部として値をスラッシュで区切る必要があります。 – 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値に置き換えます。範囲は 0 ～ 4095 です。 • 名前表記は *interface-path-id.subinterface* で、表記の一部として引数をピリオドで区切る必要があります。
ステップ 3	**encapsulation dot1q** **例：** `RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100, untagged`	インターフェイスのレイヤ 2 カプセル化を設定します。 **（注）** **dot1q vlan** コマンドは、Cisco ASR 9000 シリーズ ルータでは **encapsulation dot1q** コマンドに置き換えられます。引き続き、下位互換性のために使用可能ですが、レイヤ 3 インターフェイスだけが対象です。
ステップ 4	**ipv4 address** *ip-address mask* **例：** `RP/0/RSP0/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。 – スラッシュ（/）と数字による表記。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。

	コマンドまたはアクション	目的
ステップ 5	**exit** **例：** `RP/0/RSP0/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/RSP0/CPU0:router(config)# end` または `RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 8	**show ethernet trunk bundle-ether** *instance* **例：** `RP/0/RSP0/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。

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。

手順の概要

1. **configure**
2. **interface** {**GigabitEthernet** | **TenGigE** | **Bundle-Ether**] *interface-path-id.subinterface* **l2transport**
3. **encapsulation dot1q**
4. **l2protocol cpsv** {**tunnel** | **reverse-tunnel**}

5. **end**
   または
   commit
6. **show interfaces** [**GigabitEthernet** | **TenGigE**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure 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 2	**interface** [**GigabitEthernet** | **TenGigE** | **Bundle-Ether** | **TenGigE**] *interface-path*] *id.subinterface* **l2transport** **例：** `RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/1/0/0.1 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。 • 引数を、次のいずれかのインスタンスで置き換えます。 – 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。名前の表記は *rack*/*slot*/*module*/*port* の形式で、表記の一部として値をスラッシュで区切る必要があります。 – 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。 （注） 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** **例：** `RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100, untagged`	インターフェイスのレイヤ 2 カプセル化を設定します。 （注） **dot1q vlan** コマンドは、Cisco ASR 9000 シリーズ ルータでは **encapsulation dot1q** コマンドに置き換えられます。引き続き、下位互換性のために使用可能ですが、レイヤ 3 インターフェイスだけが対象です。

	コマンドまたはアクション	目的
ステップ 4	**l2protocol cpsv** {**tunnel** | **reverse-tunnel**} **例：** `RP/0/RSP0/CPU0:router(config-if-l2)# l2protocol cpsv tunnel`	プロトコル CDP、PVST+、STP、VTP のイーサネット インターフェイスでのレイヤ 2 プロトコル トンネリングとプロトコル データ ユニット（PDU）フィルタリングを設定します。 • **tunnel**：インターフェイスに入るときのフレームの L2PT カプセル化と、インターフェイスから出るときのフレームのカプセル化解除を指定します。 • **reverse-tunnel**：インターフェイスから出るときのフレームの L2PT カプセル化と、インターフェイスに入るときのフレームのカプセル化解除を指定します。
ステップ 5	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config-if-l2)# end` または `RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 6	**show interfaces** [**GigabitEthernet** | **TenGigE**] *interface-path-id.subinterface* **例：** `RP/0/RSP0/CPU0:router# show interfaces TenGigE 0/3/0/0.1`	（任意）ルータ上のインターフェイスに関する統計情報を表示します。

次の作業

- ポイントツーポイント疑似回線相互接続を AC 上で設定するには、『*Cisco ASR 9000 Series Router Multiprotocol Label Switching Configuration Guide*』の「*Implementing MPLS Layer 2* VPNs」モジュールを参照してください。
- マルチプロトコル ラベル スイッチング（MPLS）や QoS などのレイヤ 3 サービス ポリシーを VLAN に付加する方法については、該当する Cisco ASR 9000 シリーズ ルータのコンフィギュレーション ガイドを参照してください。

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タスクで設定した 802.1Q VLAN サブインターフェイスを削除する手順について説明します。

手順の概要

1. **configure**
2. **no interface** {**GigabitEthernet** | **TenGigE** | **Bundle-Ether**] *interface-path-id.subinterface*
3. ステップ 2 を繰り返し、その他の VLAN サブインターフェイスを削除します。
4. **end**
   または
   commit
5. **show ethernet trunk bundle-ether** *instance*（任意）

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**no interface** {**GigabitEthernet** | **TenGigE** | **Bundle-Ether**] *interface-path-id.subinterface* **例：** `RP/0/RSP0/CPU0:router(config)# no interface TenGigE 0/2/0/4.10`	サブインターフェイスを削除すると、そのサブインターフェイスに適用されているすべての設定も自動的に削除されます。 • *instance* 引数を、次のいずれかのインスタンスで置き換えます。 – 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。名前の表記は *rack*/*slot*/*module*/*port* の形式で、表記の一部として値をスラッシュで区切る必要があります。 – 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。

	コマンドまたはアクション	目的
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/RSP0/CPU0:router(config)# end` または `RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	**show ethernet trunk bundle-ether** *instance* **例：** `RP/0/RSP0/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。

VLAN インターフェイスの設定例

ここでは、次の例について説明します。

「VLAN サブインターフェイス：例」（P.695）

VLAN サブインターフェイス：例

次の例では、3 つの VLAN サブインターフェイスを一度に作成する方法を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface TenGigE 0/2/0/4.1
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 10.0.10.1/24
RP/0/RSP0/CPU0:router(config-subif)# interface TenGigE0/2/0/4.2
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 101
RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 10.0.20.1/24
RP/0/RSP0/CPU0:router(config-subif)# interface TenGigE0/2/0/4.3
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 102
```

```
RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 10.0.30.1/24
RP/0/RSP0/CPU0:router(config-subif)# commit
RP/0/RSP0/CPU0:router(config-subif)# exit
RP/0/RSP0/CPU0:router(config)# exit

RP/0/RSP0/CPU0:router# show ethernet trunk bundle-Ether 1
Trunk                                Sub types        Sub states
Interface     St Ly    MTU    Subs       L2      L3       Up    Down  Ad-Down
BE1           Up L3   1514    1000        0    1000     1000       0       0

Summary                       1000        0    1000     1000       0       0
```

次の例では、1 つのイーサネット バンドルに対して 2 個の VLAN サブインターフェイスを作成する方法を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface bundle-ether 2
RP/0/RSP0/CPU0:router(config-if)# ipv4 address 192.168.2.1/24
RP/0/RSP0/CPU0:router(config-if)# exit
RP/0/RSP0/CPU0:router(config)# interface bundle-ether 2.1
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 192.168.100.1/24
RP/0/RSP0/CPU0:router(config-subif)# exit
RP/0/RSP0/CPU0:router(config)# interface bundle-ether 2.2
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 200
RP/0/RSP0/CPU0:router(config-subif)# ipv4 address 192.168.200.1/24
RP/0/RSP0/CPU0:router(config-subif)# exit
RP/0/RSP0/CPU0:router(config)# commit
```

次に、基本的な dot1Q AC を作成する例を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/0.1
RP/0/RSP0/CPU0:router(config-subif)# l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 100
RP/0/RSP0/CPU0:router(config-subif)# commit
RP/0/RSP0/CPU0:router(config-subif)# exit
RP/0/RSP0/CPU0:router(config)# exit
```

次の例では、Q-in-Q AC を作成する方法を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/0.2
RP/0/RSP0/CPU0:router(config-subif)# l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 200 second-dot1q 201
RP/0/RSP0/CPU0:router(config-subif)# commit
RP/0/RSP0/CPU0:router(config-subif)# exit
RP/0/RSP0/CPU0:router(config)# exit
```

次の例では、Q-in-Any AC を作成する方法を示します。

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/0.3
RP/0/RSP0/CPU0:router(config-subif)# l2transport
RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 300 second-dot1q any
RP/0/RSP0/CPU0:router(config-subif)# commit
RP/0/RSP0/CPU0:router(config-subif)# exit
RP/0/RSP0/CPU0:router(config)# exit
```

その他の関連資料

ここでは、VLAN インターフェイスの設定に関連する参考資料を示します。

関連資料

関連項目	参照先
Cisco ASR 9000 シリーズ ルータ マスター コマンド リファレンス	*『Cisco ASR 9000 Series Router Master Commands List』*
Cisco ASR 9000 シリーズ ルータ インターフェイス コンフィギュレーション コマンド	*『Cisco ASR 9000 Series Router Interface and Hardware Component Command Reference』*
Cisco IOS XR ソフトウェアを使用する Cisco ASR 9000 シリーズ ルータの初期システム ブートアップと設定に関する情報。	*『Cisco ASR 9000 Series Router Getting Started Guide』*
ユーザ グループとタスク ID に関する情報	*『Cisco ASR 9000 Series Router Interface and Hardware Component Command Reference』*

標準

標準	タイトル
この機能によってサポートされる新しい規格または変更された規格はありません。またこの機能による既存規格のサポートに変更はありません。	—

MIB

MIB	MIB のリンク
このモジュールに適用できる MIB はありません。	Cisco IOS XR ソフトウェアを使用して選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	—

シスコのテクニカル サポート

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

Cisco ASR 9000 シリーズ ルータでのサテライト ネットワーク仮想化（nV）システムの設定

このモジュールでは、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータでのサテライト ネットワーク仮想化システムの設定について説明します。

Cisco ASR 9000 シリーズ ルータでのサテライト システム設定の機能履歴

リリース	変更内容
リリース 4.2.1	• サテライト ネットワーク仮想化（サテライト nV）サービスのサポートが Cisco ASR 9000 シリーズ ルータで追加されました。
リリース 4.2.3	• 36 ポート 10 ギガビット イーサネット ラインカードのサポートが追加されました。
リリース 4.3.0	• ホストとしての Cisco ASR 9001 および Cisco ASR 9922 シリーズ ルータのサポートが追加されました。 • サテライト デバイスとしての Cisco ASR 901 および Cisco ASR 903 のサポートが追加されました。

内容

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。

サテライト nV システムを設定する前に、次のハードウェアおよびソフトウェアがシャーシにインストールされている必要があります。

- ハードウェア: Cisco ASR 9000 シリーズ アグリゲーション サービス ルータと ASR 9000 Enhanced イーサネット ラインカード（シャーシ間リンクのロケーションとして）、および Cisco ASR9000v、Cisco ASR 901、Cisco ASR 903 ルータ（サテライト ボックスとして）。
- ソフトウェア：Cisco IOS XR ソフトウェア Release 4.2.1 以降。

その他のハードウェア要件の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Hardware Installation Guide*』を参照してください。

サテライト nV スイッチング システムの概要

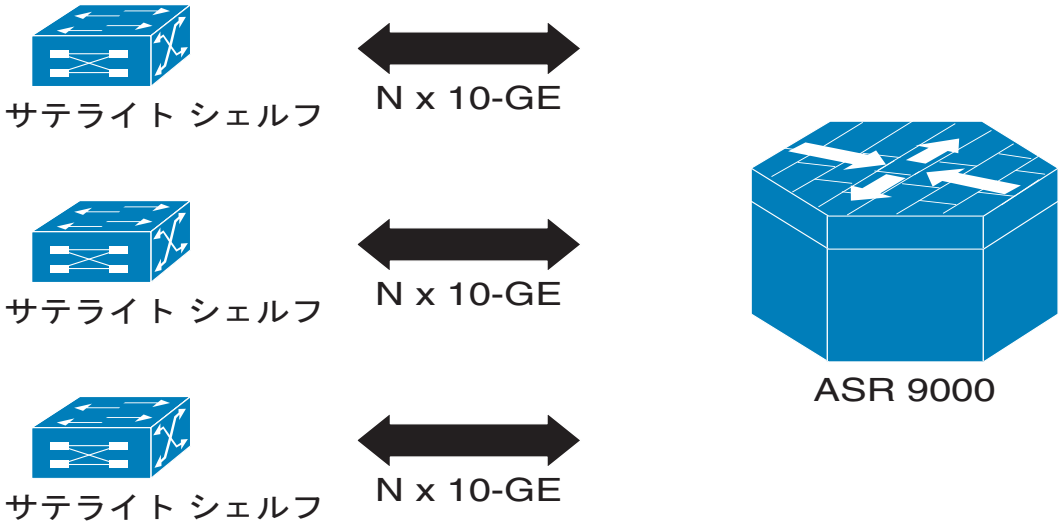
Cisco ASR 9000 シリーズ ルータ サテライト ネットワーク仮想化（nV）サービスまたはサテライト スイッチング システムを使用すると、1 つ以上のサテライト スイッチが 1 つ以上の Cisco ASR 9000 シリーズ ルータを補完するトポロジを構成でき、全体で 1 つの仮想スイッチング システムを実現することができます。このシステムでは、サテライト スイッチはルータによる管理制御の下で動作します。サテライト シャーシおよび機能の設定および管理はすべて、Cisco ASR 9000 シリーズ ルータのコントロール プレーンおよび管理プレーンを通して実行され、このルータはホストと呼ばれます。

（注） Cisco ASR 9001 および Cisco ASR 9922 シリーズ ルータも、サテライト nV サービスのホストとして使用できます。

Cisco ASR 9000 シリーズ ルータとそのサテライトとの相互接続は、標準のイーサネット インターフェイスを介して行われます。サテライト nV サービスが Cisco IOS XR Release 4.2.x で導入されたときは、
Cisco ASR 9000v がサテライト デバイスとして使用されていました。4 個の 10 ギガビット ポートが ICL として使用されていました。一方、Cisco ASR 901 では、2 個の 1 ギガビット ポートが ICL として使用されます。一般的に、ホストで使用されるインターフェイスのタイプは、使用するサテライト デバイスに基づいて決定されます。図 35 を参照してください。

（注） 1 Gig と 10 GigE の両方のイーサネット インターフェイスを ICL として使用することは、Cisco ASR 9000 Enhanced イーサネット ラインカードではサポートされますが、Cisco ASR 9000 イーサネット ラインカードではサポートされません。

図 35　Cisco ASR 9000 シリーズ サテライト nV スイッチング システム



このタイプのアーキテクチャは、キャリア イーサネット転送ネットワークで実現できます。サテライト スイッチは、アクセス スイッチとして使用することも、プリアグリゲーション/アグリゲーション スイッチとして使用することもできます。これらのスイッチからの接続先は、より高度な L2、L3 サービスがプロビジョニングされた、Cisco ASR 9000 シリーズ ルータや Cisco CRS 3 ルータなどのエッジ ルータです。

このモデルは、FTTB（Fiber To The Business）ネットワーク アプリケーションにも利用できます。このようなアプリケーションでは、企業向けインターネットおよび VPN サービスが商用ベースで提供されます。他にも、ワイヤレス/RAN バックホール集約ネットワークなどに使用できます。

サテライト nV システムの利点

Cisco ASR 9000 シリーズ サテライト nV システムには、次の利点があります。

1. **ポートの拡張性と密度の向上**：400 個を超える物理ギガビット イーサネット ポートから 1 つの仮想ラインカードを作成できます。生成される論理的な Cisco ASR 9000 シリーズ ルータの、イーサネット ポート密度が大幅に増加します。たとえば、Cisco ASR 9000 シリーズ ルータの 24 ポート TenGigE ラインカード 1 つで最大 24 台のサテライト スイッチ（1 台につき 44 個の GigE ポート）を統合できるので、実質的なポート密度は、Cisco ASR 9000 シリーズ ルータのラインカード スロットあたりギガビット イーサネット ポート 1056 個となります。その他の構成では、さらに高いポート密度を達成できます。このことが利点となるのは、Cisco ASR 9000 シリーズ ルータではスロットあたりのノンブロッキング キャパシティが最大 400 Gbps ですが（適切な RSP を使用する場合）、物理的に数百個ものギガビット イーサネット ポート/SFP を 1 つの Cisco ASR 9000 シリーズ ラインカードの前面プレートに収容する手段は他にはないからです。その結果、Cisco ASR 9000 シリーズ ラインカードのキャパシティを最大限に活用するには、イーサネット ポートを物理的に分けると同時に、論理的な管理制御を維持することが必要になります。このようにすると、すべてのポートが物理的に Cisco ASR 9000 シリーズ ルータの 1 つの大きなラインカード上に存在しているように見えます。
2. **コスト削減**：Cisco IOS XR ソフトウェアのエッジ ルーティング能力およびアプリケーション機能のすべてを、低コストのアクセス スイッチで使用できます。

3. **運用費の削減**：ソフトウェア イメージをシームレスにアップグレードできます。また、シャーシおよびサービスを 1 か所から管理できます。たとえば、全体を 1 つの論理的ルータとして見ることができ、CLI または XML のインターフェイスを 1 か所からスイッチのシステム全体に対して適用できます。また、スイッチのシステム全体のモニタリングや、システム全体のイメージ管理とソフトウェア アップグレードを、1 か所から行うことができます。
4. **機能の一貫性の向上**：Cisco ASR 9000 シリーズ ルータの通常の GigE ポートのすべての機能を、サテライト アクセス スイッチのアクセス ポートでも使用でき、機能的な統一を図ることができます。サテライト システムは、ネットワークのアクセス レイヤや集約レイヤで使用されるのが一般的です。アクセス スイッチを集約またはコア スイッチとともに統合することによって、アクセス スイッチと集約/コア スイッチとの間の機能のギャップをなくすことができます。すべての機能（キャリア イーサネット機能、QoS、および OAM など）が、アクセスでもコアでも同じように動作するのは、この統合アプローチが採用されているからです。
5. **各機能導入のスピードの向上**：このサテライト ソリューションでは、Cisco ASR 9000 シリーズ ルータ上で実装されるすべての機能が、即座にアクセス スイッチでも利用可能になるので、エッジ スイッチに対して理想的な速さで各機能が導入されるようになります。
6. **復元力の向上**：nV サテライト ソリューションでは、マルチシャーシ復元力が高まり、エンドツーエンド QoS も向上します。QoS 機能の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router QoS Configuration Guide*』を参照してください。

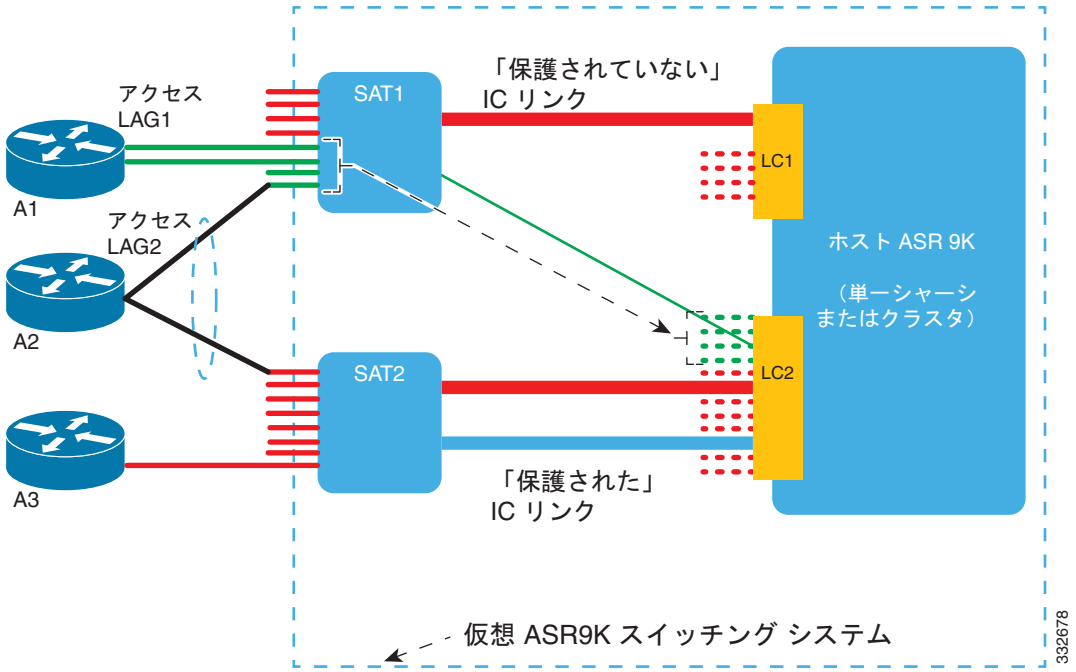
ポート エクステンダ モデルの概要

ポート エクステンダ サテライト スイッチング システムでは、サテライト スイッチはその親である Cisco ASR 9000 シリーズ ルータに、物理イーサネット ポートを通してアタッチされます。

（注） Cisco IOS XR ソフトウェア Release 4.2.1 よりも後のリリースでは、このポート エクステンダ モデルを越えるアタッチ モデルもサポートされます。

このモデルでは、親の Cisco ASR 9000 シリーズ ルータはホストと呼ばれます。管理やプロビジョニングという点で見ると、サテライト スイッチの物理アクセス ポートは、Cisco ASR 9000 シリーズ ルータ上の物理イーサネット ポートと同等です。サテライト スイッチング システムを管理するために、専用のコンソール接続を用意する必要はありません（デバッグ目的の場合を除く）。サテライトのインターフェイスおよびシャーシ レベルの機能は、ホスト上で稼働する Cisco IOS XR ソフトウェアのコントロール プレーンで認識されます。したがって、サテライトおよびホストを 1 つの論理ルータとして完全に管理できます。

図 36　ポート エクステンダ サテライト スイッチング システム



このモデルでは、単一の Cisco ASR 9000 シリーズ ルータが 2 台のサテライト スイッチ SAT1 および SAT2 をホスティングし、全体的な仮想 Cisco ASR 9000 スイッチング システムを形成します。図 36 の Cisco ASR 9000 シリーズ ルータ、SAT1、および SAT2 を囲む点線がこれを表しています。

この構造は、外部ネットワークからは実質的に単一の論理的な Cisco ASR 9000 シリーズ ルータに見えます。外部アクセス スイッチ A1、A2、A3 をこの全体的な仮想スイッチに接続するには、通常のイーサネット リンクを使用して SAT1 および SAT2 に物理的に接続します。サテライト スイッチと Cisco ASR 9000 シリーズ ルータとの間のリンクはイーサネット リンクであり、シャーシ間リンク（ICL）と呼ばれます。Cisco ASR 9000 シリーズ ルータは、このシステムではホストと呼ばれます。シャーシ間リンク上で輻輳が発生したときは、内蔵 QoS 保護メカニズムをトラフィックに対して使用できます。

（注） SAT1、SAT2 およびホスト Cisco ASR 9000 シリーズ ルータを、地理的に同じ場所に配置する必要はありません。つまり、ICL の距離は、特定の場所または建物の中だけで使用するための公称長である必要はありません。ICL の長さは数十、数百、あるいは数千 km とすることもできるので、地理的に広範囲に広がる論理サテライト スイッチを作成できます。

（注） Cisco ASR 9000 シリーズ ルータ マルチシャーシ クラスタ システムでは、複数の Cisco ASR 9000 シリーズ ルータ システムが 1 つの仮想スイッチ システム内に存在します。ただし、論理的には、これも単一ホスト システムと見なされます。

サテライト nV システムでサポートされる機能

ここでは、サテライト システムの機能の詳細を示します。

サテライト システムの物理トポロジ

サテライト システムでは、サテライト スイッチとホスト Cisco ASR 9000 シリーズ ルータとの間の ICL に対して、ポイントツーポイント ハブ アンド スポーク物理トポロジがサポートされます。このトポロジでは、サテライトから Cisco ASR 9000 シリーズ ルータへの物理イーサネット MAC レイヤ接続が可能です。これを実現するには、Ethernet over Fiber または Ethernet over Optical を介しての直接的な転送を利用します（Ethernet over SONET/SDH/CWDM/DWDM ネットワークなど）。

このトポロジでは、サテライト スイッチを地理的にホスト Cisco ASR 9000 シリーズ ルータとは別の場所に配置することもできます。距離の制限はなく、このソリューションはサテライトがホストから数十、数百、あるいは数千 km 離れた場所に配置されていても機能します。

シャーシ間リンク冗長モードとロード バランシング

Cisco ASR 9000 シリーズ サテライト システムでは、次の冗長モードがサポートされます。

- **非冗長シャーシ間リンク モード**：このモードでは、1 つのサテライトの複数のシャーシ間リンクの間にリンク レベルの冗長性はありません。
- **冗長シャーシ間リンク モード**：このモードでは、シャーシ間リンクの間のリンク レベルの冗長性を、単一リンク集約（LAG）バンドルを使用して実現します。

冗長 ICL モードでは、IC バンドルのメンバー間のトラフィックのロード バランシングを行うために、サテライト アクセス ポート ID に基づく単純なハッシュ関数が使用されます（パケットからの L2 または L3 ヘッダーの内容を使用するフロー ベース ハッシュに基づくものではありません）。したがって、1 つのサテライト アクセス ポートのすべてのパケットで確実に同じ ICL が使用されるようになります。その結果、QoS などの機能のアクションを適用するときに、同じサテライト アクセス ポートに属するすべてのパケットが対象となります。

（注） Cisco IOS XR ソフトウェアは、冗長と非冗長の両方の ICL モードが同じ nV サテライト シェルフに共存することをサポートしています（Cisco IOS XR ソフトウェア Release 4.3.x 以降）。

QoS の適用と ICL に対する設定の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Configuration Guide*』を参照してください。

サテライト検出および制御プロトコル

シスコ独自の検出と制御のプロトコルが、サテライト スイッチとホスト Cisco ASR 9000 シリーズ ルータ デバイスの間で使用されます。これは、サテライト デバイスの検出、プロビジョニング、およびモニタリングを、ホスト Cisco ASR 9000 シリーズ サテライト システムからインバンドで ICL を介して行うためのものです。サテライト検出および制御（SDAC）プロトコルは、サテライト デバイスとそのホストの間の関係の動作、意味、構文的定義です。

サテライト検出および制御プロトコルの IP 接続

SDAC プロトコルの接続は、ICL を経由する通常のインバンド IP ルーテッド パスを通して行われ、これにはキャリア ネットワークに応じて適切なプライベートとパブリックの IP アドレスが使用されます。

ホスト CLI で、各サテライト スイッチの管理 IP アドレスや、ICL 上の対応する IP アドレスを設定できます。プライベート IPv4 アドレス空間からアドレスを選択できます（たとえば 10.0.0.0/8 や 192.1.168.0/24）。これは、IPv4 FIB で使用されている通常のサービス レベル IPv4 アドレスとの競合を防ぐためです。サテライト管理トラフィックだけに使用するプライベート VRF を設定することもできます。サテライトに割り当てられる IP アドレスがこのプライベート VRF 内に収まるようになります。このようにすると、アドレス競合のリスクや IP アドレス管理の複雑さは、ルータで使用される他の IP アドレスや VRF と比べて軽減されます。

レイヤ 2 および L2VPN の機能

物理イーサネットまたはバンドル イーサネット インターフェイスでサポートされる L2 および L2VPN の機能はすべて、サテライト イーサネット インターフェイスでもサポートされます。Cisco ASR 9000 シリーズ サテライト システムでサポートされる L2VPN 機能の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide*』を参照してください。

レイヤ 3 および L3VPN の機能

イーサネット インターフェイスでサポートされる MPLS L3VPN の機能（GRE、NetFlow など）はすべて、Cisco ASR 9000 シリーズ サテライト システムでもサポートされます。これらの機能の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router MPLS Layer 3 VPN Configuration Guide*』および『*Cisco ASR 9000 Series Aggregation Services Router Netflow Configuration Guide*』を参照してください。

レイヤ 2 およびレイヤ 3 マルチキャストの機能

レイヤ 2 およびレイヤ 3 マルチキャストの機能（IGMP、IGMP スヌーピング、PIM、mLDP、MVPN、P2MP TE など）はすべて、サテライト イーサネット インターフェイスでも、標準のイーサネットおよびバンドル イーサネット インターフェイスと同様にサポートされます。サテライト システムでサポートされるこれらの機能の詳細については、『*Cisco ASR 9000 Series Aggregation Services Routers Multicast Configuration Guide*』を参照してください。

Quality of Service

レイヤ 2、レイヤ 3 QoS および ACL の機能のほとんどは、サテライト イーサネット インターフェイスでも、通常の物理イーサネット インターフェイスと同様にサポートされますが、キューイング アクションを伴う入力ポリシーを除きます。ただし、QoS については、動作における機能的な相違があります。Cisco IOS XR ソフトウェア Release 4.2.1 では、ユーザによって設定された MQC ポリシーは Cisco ASR 9000 シリーズ ルータでは適用されますが、サテライト スイッチ インターフェイスでは適用されないからです。QoS ポリシーの属性、機能、および設定の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Configuration Guide*』を参照してください。

（注） ユーザ設定の QoS ポリシーは、IC リンク輻輳およびオーバーサブスクリプションのシナリオを処理するために適用される、デフォルトのポート レベル QoS からは独立しています。サテライト システム ポートに適用されるデフォルトのポート レベル QoS に加えて、Cisco ASR 9000 シリーズ ルータ側では、サテライト イーサネット ポートとの間で送受信される入力および出力のトラフィックに対して適用されるデフォルト QoS もあります。

クラスタのサポート

Cisco ASR 9000 シリーズ ルータのクラスタは、サテライト モードとともにサポートされます。単一クラスタ システムは、サテライト スイッチのグループのための、1 個の論理 Cisco ASR 9000 シリーズ ルータ ホスト システムであるかのように機能します。クラスタ システムでは、1 つのサテライト スイッチで、ある ICL をラック 0 に接続し、他の ICL をラック 1 に接続することもできます。詳細については、「*Cisco ASR 9000 シリーズ ルータでの nV エッジ システムの設定*」の章を参照してください。

（注） サテライト イーサネット インターフェイスは、クラスタのラック間リンクとしては使用できません。

時刻の同期

サテライト スイッチの時刻パラメータは、ホストの時刻と同期されます。これによって、デバッグ メッセージやその他のサテライト イベント ログのタイムスタンプがホストとの間で、およびネットワーク上のすべてのサテライト スイッチとの間で整合していることを保証できます。このことは、ICL が検出されたときに、ホストからのサテライト スイッチへの SDAC 検出プロトコルを通して実現されます。

サテライト シャーシ管理

サテライトのシャーシ レベルの管理は、ホストを介して行われます。サテライト スイッチは、全体的な仮想スイッチの論理的な一部分であるからです。したがって、サービス プロバイダーは、ボックス レベル管理に加えて、サービス レベルを含むあらゆる面を単一の論理デバイスとして管理できるようになります。その結果、ネットワークの運用が単純になります。この運用とは、サテライト シャーシのインベントリ管理、環境センサー モニタリング、障害/アラーム モニタリングなどであり、ホスト Cisco ASR 9000 シリーズ ルータの、対応する CLI、SNMP および XML インターフェイスを介して行われます。

（注） サテライト システム ハードウェアの機能、SFP のサポート、および互換性のあるトポロジについては、『*Cisco ASR 9000 Series Aggregation Services Router Hardware Installation Guide*』を参照してください。

サテライト nV システムの制限事項

現在のソフトウェア リリースでは、サテライト nV システムに関して次のような制限事項があります。これらの制限は、将来のリリースで解消される予定です。

- シャーシ間リンク冗長性がサポートされるのはスタティック EtherChannel のみであり、LACP ベース リンクではサポートされません。最小および最大リンク コマンドは、ICL がバンドルの場合には適用されません。
- サテライト システムが冗長 ICL モードで動作している場合は、どのような形態のリンク バンドルも（LACP の有無に関係なく）、その同じサテライト スイッチのアクセス ポート上では設定できません。
- サテライト システムが冗長 ICL モードで動作している場合は、そのサテライトのアクセス ポートではイーサネット OAM の機能はサポートされません。
- マルチシャーシ リンク集約がサポートされるのは、2 つの独立した Cisco ASR 9000 シリーズ ルータが、PoA（接続ポイント）として機能している場合です。これらのそれぞれに専用のサテライト スイッチがあり、各サテライト スイッチを通して DHD（デュアル ホーム デバイス）接続が行われます。ただし、MC-LAG は、1 つのサテライト スイッチが 2 つの独立した Cisco ASR 9000 シリーズ ルータに ICL LAG を介して接続されている場合はサポートされません。

（注） ソフトウェアのその他の制限事項については、『*Cisco ASR 9000 Series Aggregation Services Router Release Notes*』を参照してください。

サテライト nV システムの実装

インターフェイス コントロール プレーン エクステンダ（ICPE）インフラストラクチャには、サテライト デバイスに物理的に存在するインターフェイスのコントロール プレーンを Cisco IOS XR ソフトウェアの中で提供するメカニズムがあります。このインフラストラクチャが確立された後は、インターフェイスはルータ上の他の物理イーサネット インターフェイスと同様に機能します。

ICPE の設定は、これらの機能領域をカバーします。これらの設定は、サテライト デバイスとの接続を設定するのに必要です。

- 「サテライト nV システムの定義」（P.707）
- 「ホスト IP アドレスの設定」（P.710）
- 「シャーシ間リンクと IP 接続の設定」（P.711）
- 「プラグ アンド プレイ サテライト nV スイッチの起動（Rack, Plug, and Go インストール）」（P.714）

サテライト nV システムの定義

Cisco IOS XR ソフトウェアにアタッチされるサテライトはそれぞれ、ホスト上で設定されている必要があり、それぞれに一意の ID も必要です。設定および機能を適切に検証するには、サテライトのタイプとその機能も指定する必要があります。

さらに、サテライトと接続できるようにするには、IP アドレスを設定する必要があります。このアドレスはサテライトまで、検出プロトコルを通してプッシュされ、これによって制御プロトコルの接続が可能になります。

ここでは、サテライト システムを定義するために ID と基本的な識別情報を割り当てる方法について説明します。

手順の概要

1. **configure**
2. **nv**
3. **satellite** <*Satellite ID*>
4. **serial-number** <*string*>（任意）
5. **description** <*string*>（任意）
6. **type** <*type*>
7. **ipv4 address** <*address*>
8. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**nv** **例：** `RP/0/RSP0/CPU0:router(config)# nv`	nV コンフィギュレーション サブモードを開始します。
ステップ 3	**satellite** *id* **例：** `RP/0/RSP0/CPU0:router(config-nV)# satellite <100-65534>`	ホストにアタッチされる新しいサテライトを宣言し、サテライト コンフィギュレーション サブモードを開始します。
ステップ 4	**serial-number** *<string>* **例：** `RP/0/RSP0/CPU0:router(config-nV)# serial-number CAT1521B1BB`	（任意）シリアル番号は、サテライト認証に使用されます。
ステップ 5	**description** *id* **例：** `RP/0/RSP0/CPU0:router(config-nV)# description Milpitas Building12`	（任意）サテライトに関連付けられる説明の文字列がある場合は指定します（場所など）。

	コマンドまたはアクション	目的
ステップ 6	**type** *type_name* **例：** `RP/0/RSP0/CPU0:router(config-nV)# satellite 200 type ?` `asr9000v  Satellite type`	アタッチされるサテライトのタイプを定義します。サテライトのタイプは、ASR9000v、ASR901v、および ASR 903v です。
ステップ 7	**ipv4 address** *address* **例：** `RP/0/RSP0/CPU0:router(config-nV)# ipv4 address 10.22.1.2`	サテライトに割り当てる IP アドレスを指定します。指定した IP アドレスへの接続ルートが ICPE によってセットアップされます。このルートは、設定済みのすべての ICL を経由します。
ステップ 8	**end** または **commit** **例：** `RP/0/0RSP0/CPU0:router(config)# end` または `RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ホスト IP アドレスの設定

この手順では、ループバック インターフェイスのホスト IP アドレスを設定する手順を提供します。

手順の概要

1. **configure**
2. **interface** *Loopback0*
3. **ipv4 address** 8.8.8.8 255.255.255.255
4. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *loopback0* **例：** `RP/0/RSP0/CPU0:router(config)# interface loopback0`	インターフェイスのループバック アドレスを指定します。

	コマンドまたはアクション	目的
ステップ 3	**ipv4 address** **例：** `RP/0/RSP0/CPU0:router(config-int)# ipv4 address 8.8.8.8 255.255.255.255`	ループバック インターフェイスのホスト IP アドレスを設定します。
ステップ 4	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config)# end` または `RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

シャーシ間リンクと IP 接続の設定

シャーシ間リンク（ICL）は、明示的に設定する必要があります。どのサテライトが接続されるかを指定するためです。また、アクセス ポートも指定する必要があります。これはダウンストリーム GigE ポートであり、設定済みの ICL を介してホストにクロスリンクされます。ホストとサテライトの間の接続を確立するには、適切な IP アドレスを両側で設定する必要があります。サテライト IP アドレスは、検出プロトコルを通して転送されます。この設定については、「サテライト nV システムの定義」（P.707）で説明しています。

（注） この設定では、グローバル デフォルト VRF が使用されます。推奨されるオプションは、プライベート VRF を nV の IP アドレスに使用することです。サテライト システムの設定：例のプライベート VRF を使用するサテライト管理を参照してください。

手順の概要

1. **configure**
2. **interface** *<interface_name>*
3. **description** *To Sat5 1/46*
4. **ipv4** *point-to-point*
5. **ipv4 unnumbered** *Loopback0*

6. **nv**
7. **satellite-fabric-link satellite** *<id>*
8. **remote-ports** *interface-type*
9. **end**
   または
   commit

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** **例：** `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *interface-name* **例：** `RP/0/RSP0/CPU0:router(config)# interface TenGigE0/2/1/0`	サポートされるシャーシ間リンク インターフェイスのタイプは、サポートされるサテライトでの接続方法によって決まります。GigabitEthernet、TenGigE および Bundle-Ether インターフェイスのみが ICL タイプをサポートします。
ステップ 3	**description** **例：** `RP/0/RSP0/CPU0:router(config-interface)# description To Sat5 1/46`	サポートされるシャーシ間リンク インターフェイス タイプの説明を指定します。
ステップ 4	**ipv4** *point-to-point* **例：** `RP/0/RSP0/CPU0:router(config-interface)# ipv4 point-to-point`	IPv4 ポイント ツー ポイント アドレスを設定します。
ステップ 5	**ipv4 unnumbered** *loopback0* **例：** `RP/0/RSP0/CPU0:router(config-interface)# interface unnumbered loopback0`	インターフェイスの IPv4 ループバック アドレスを設定します。
ステップ 6	**nv** **例：** `RP/0/RSP0/CPU0:router(config)# nv`	nV コンフィギュレーション サブ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 7	**satellite-fabric-link satellite** *<id>* **例：** `RP/0/RSP0/CPU0:router(config-int-nv)# satellite-fabric-link satelite 200`	インターフェイスが ICPE シャーシ間リンクであることを指定します。
ステップ 8	**remote-ports** *interface type* **例：** `RP/0/RSP0/CPU0:router(config-int-nv)# remote-ports GigabitEthernet 0/0/0-30`	リモート サテライト ポート 0 ～ 30 を設定します。
ステップ 9	**end** または **commit** **例：** `RP/0/RSP0/CPU0:router(config)# end` または `RP/0/RSP0/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、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 ファイルに設定変更が保存され、コンフィギュレーション セッションが終了し、ルータが EXEC モードに戻ります。 – **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。 – **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

（注） ICL に対する QoS 設定については、『*Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Configuration Guide*』を参照してください。

サテライト nV アクセス インターフェイスの設定

サテライトのアクセス GigabitEthernet インターフェイスを Cisco IOS XR ソフトウェアの中でローカルに表現するために、GigabitEthernet という名前のインターフェイスが使用されます。これは、サテライトではない他の GigabitEthernet インターフェイスに似ています。唯一の違いは、サテライト アクセス GigabitEthernet インターフェイスに使用されるラック ID は、そのサテライトに対して設定されたサテライト ID であることです。

このインターフェイスでは、GigabitEthernet インターフェイス（物理 IC リンクを介して実行される場合）または Bundle-Ether インターフェイス（仮想 IC リンクを介して実行される場合）で通常設定されるすべての機能がサポートされます。

プラグ アンド プレイ サテライト nV スイッチの起動（Rack, Plug, and Go インストール）

1. サテライト ラックを開梱し、スタックし、電源ケーブルに接続します。
2. 正しいタイプの認定光モジュールを 1 つ以上の SFP+ スロットに差し込み、適切な認定光モジュールをホストの SFP+ または XFP スロットに差し込みます。SMF/MMF ファイバで接続します。

（注） ホストからの 10GigE ファイバを、サテライト デバイスの任意の 10G SFP+ ポートに任意の順序で接続します。

（注） サテライト nV サービスでは、Cisco ASR 9000 シリーズ ルータまたは Cisco ASR 9001 および Cisco ASR 9922 シリーズ ルータをホストとして使用できます。Cisco ASR 9000v、Cisco ASR 901、または Cisco ASR 903 ルータをサテライト デバイスとして使用できます。

3. サテライト nV システムの設定を、ホストの 10GigE ポートで CLI または XML を使用して行います。ホストを nV 動作用に設定します。手順については、サテライト nV システムの定義、ホスト IP アドレスの設定、およびシャーシ間リンクと IP 接続の設定を参照してください。
4. サテライト デバイスのシャーシの電源を投入します。
5. サテライト シャーシのステータスは、前面プレートのシャーシ エラー LED に基づいてチェックできます。
   - 重大エラー LED が点灯しているときは、重大なハードウェア障害を示します。
   - メジャー エラー LED が点灯しているときは、ハードウェアは機能しているがホストに接続できないことを示します。
   - 重大とメジャーの LED が消灯しているときは、サテライト デバイスは稼働中であり、ホストに接続されています。
   - 必要に応じて、サテライト イーサネット ポート パケット ループバック テストを、ホストを通して行います。この目的は、エンド ツー エンドのデータ パスをチェックすることです。

（注） サテライト ソフトウェアのアップグレードが必要な場合は、ホストに通知が送られます。必要に応じて、ホストからのインバンド ソフトウェア アップグレードを実行できます。サテライトのステータスをチェックするには、ホストで **show nv satellite status** を使用します。

サテライト nV ソフトウェアのアップグレードおよび管理

サテライト ソフトウェア イメージは、Cisco ASR 9000 シリーズ ルータ パッケージの中にある **asr9k-9000v-nV-p.pie** という PIE の中にバンドルされています。Cisco IOS XR ソフトウェア プロダクション SMU ツールを使用すると、サテライト イメージのパッチを現場で生成できます。バグ修正や小規模な機能強化を、正式なソフトウェア アップグレードを必要とせずに行うことができます。

（注） Cisco ASR 901 ルータを使用するサテライト ソフトウェア イメージについては、『*Network Virtualization Using Cisco ASR 901 Series Aggregation Services Router as a Satellite*』を参照してください。

ここでは、サテライト nV ソフトウェアを管理するコマンドを示します。

前提条件

プラグ アンド プレイ サテライト インストール手順を使用してサテライトのインストールが完了している必要があります。詳細については、プラグ アンド プレイ サテライト nV スイッチの起動（Rack, Plug, and Go インストール）を参照してください。

サテライトのインストール

サテライトにソフトウェア イメージをダウンロードしてアクティブにするには、**install nv satellite** *<satellite ID / all>* **transfer/activate** コマンドを使用します **transfer** コマンドは、サテライトにイメージをダウンロードします。**transfer** コマンドの後に **activate** コマンドを実行すると、ソフトウェアがそのサテライト上でアクティブになります。

例

```
RP/0/RSP0/CPU0:sat-host#install nv satellite 100 transfer
Install operation initiated successfully.
RP/0/RSP0/CPU0:sat-host#RP/0/RSP0/CPU0:May  3 20:12:46.732 : icpe_gco[1146]:
%PKT_INFRA-ICPE_GCO-6-TRANSFER_DONE : Image transfer completed on Satellite 100

RP/0/RSP0/CPU0:sat-host#install nv satellite 100 activate
Install operation initiated successfully.
LC/0/2/CPU0:May  3 20:13:50.363 : ifmgr[201]: %PKT_INFRA-LINK-3-UPDOWN : Interface
GigabitEthernet100/0/0/28, changed state to Down
RP/0/RSP0/CPU0:May  3 20:13:50.811 : invmgr[254]: %PLATFORM-INV-6-OIROUT : OIR: Node 100
removed
```

（注） **activate** コマンドを直接実行した場合は、ソフトウェア イメージがサテライトに転送されて、アクティブ化も行われます。

例

```
RP/0/RSP0/CPU0:sat-host#install nv satellite 101 activate
Install operation initiated successfully.

RP/0/RSP0/CPU0:sat-host#RP/0/RSP0/CPU0:May  3 20:06:33.276 : icpe_gco[1146]:
%PKT_INFRA-ICPE_GCO-6-TRANSFER_DONE : Image transfer completed on Satellite 101
RP/0/RSP0/CPU0:May  3 20:06:33.449 : icpe_gco[1146]: %PKT_INFRA-ICPE_GCO-6-INSTALL_DONE :
Image install completed on Satellite 101
```

```
RP/0/RSP0/CPU0:May  3 20:06:33.510 : invmgr[254]: %PLATFORM-INV-6-OIROUT : OIR: Node 101
removed
```

（注） サテライト イメージのアップグレードを適切に行うには、管理プレーン CLI が Cisco ASR 9000 シリーズ ルータ上で設定されていないことを確認してください。設定されている場合は、この例外を 10GigE インターフェイス（これはサテライト ICL です）ごとに追加する必要があります。

この例外を追加するには、次の CLI を使用します。

```
control-plane
 management-plane
  inband
    !
   !
    interface TenGigE0/0/0/5  <=== To enable TFTP on nV satellite ICL
allow TFTP
```

この例外が追加されていない場合は、イメージのダウンロードとアップグレードは失敗します。

サテライト ソフトウェアのモニタリング

- 基本的なステータス チェックを実行するには、**show nv satellite status brief** コマンドを使用します。

```
RP/0/RSP0/CPU0:shanghai# show nv satellite status brief

Sat-ID  Type      IP Address    MAC address     State
------  --------  ------------  --------------  --------------------------------
100     asr9000v  101.102.103.105  dc7b.9426.1594  Connected (Stable)
200     asr9000v  101.102.103.106  0000.0000.0000  Halted; Conflict: no links configured
400               194.168.9.9   0000.0000.0000  Halted; Conflict: satellite has no type
configured
```

- サテライトでのアップグレードが必要かどうかを調べるには、**show nv satellite status satellite** *satellite_id* を実行します。

例

```
RP/0/RSP0/CPU0:sat-host#show nv satellite status satellite 100

Satellite 100
-------------
  State: Connected (Stable)
  Type: asr9000v
  Description: sat-test
  MAC address: dc7b.9427.47e4
  IPv4 address: 100.1.1.1
  Configured Serial Number: CAT1521B1BB
  Received Serial Number: CAT1521B1BB
  Remote version: Compatible (latest version)
    ROMMON: 125.0 (Latest)
    FPGA: 1.13 (Latest)
    IOS: 200.8 (Latest)
  Configured satellite fabric links:
    TenGigE0/2/0/6
    --------------
```

```
    State: Satellite Ready
    Port range: GigabitEthernet0/0/0-9
  TenGigE0/2/0/13
  ---------------
    State: Satellite Ready
    Port range: GigabitEthernet0/0/30-39
  TenGigE0/2/0/9
  --------------
    State: Satellite Ready
    Port range: GigabitEthernet0/0/10-19
```

（注） この例に示した出力では、**Remote version**、**ROMMON**、**FPGA**、および **IOS** に最新バージョンが表示されている必要があります。そうでない場合は、サテライトでのアップグレードが必要です。表示されるバージョン番号は、ASR 90000v にインストールされているバージョンです。前述の出力で、latest というキーワードの代わりにバージョン番号が表示されている場合は、その番号はサテライト PIE の中の ASR9000v イメージ バンドルに対応しています。

サテライト プロトコル ステータスのモニタリング

- サテライト検出プロトコルのステータスを調べるには、**show nv satellite protocol discovery** コマンドを使用します。

```
RP/0/RSP0/CPU0:router# show nv satellite protocol discovery brief
Interface       Sat-ID  Status                          Discovered links
--------------  ------  ------------------------------  ----------------------
Te0/1/0/0       100     Satellite Ready                 Te0/1/0/0
Te0/1/0/1       100     Satellite Ready                 Te0/1/0/1
```

（または）

```
RP/0/RSP0/CPU0:router# show nv satellite protocol discovery interface TenGigE 0/1/0/0

  Satellite ID: 100
  Status: Satellite Ready
  Remote ports: GigabitEthernet0/0/0-15
  Host IPv4 Address: 101.102.103.104
  Satellite IPv4 Address: 101.102.103.105
  Vendor: cisco, ASR9000v-DC-E
  Remote ID: 2
  Remote MAC address: dc7b.9426.15c2
  Chassis MAC address: dc7b.9426.1594
```

- サテライト制御プロトコルのステータスを調べるには、**show nv satellite protocol control** コマンドを使用します。

```
RP/0/RSP0/CPU0:shanghai# sh nv satellite protocol control brief
Sat-ID  IP Address     Protocol state  Channels
------  ------------   --------------  ----------------------------------
    101.102.103.105  Connected     Ctrl, If-Ext L1, If-Ext L2, X-link, Soft Reset,
Inventory, EnvMon, Alarm


RP/0/RSP0/CPU0:shanghai# sh nv satellite protocol control
Satellite 100
-------------
  IP address: 101.102.103.105
  Status: Connected
  Channels:
    Control
```

```
    -------
      Channel status: Open
      Messages sent: 24 (24 control), received: 23 (23 control).
    Interface Extension Layer 1
    ---------------------------
      Channel status: Open
      Messages sent: 7 (3 control), received: 14 (2 control).
    Interface Extension Layer 2
    ---------------------------
      Channel status: Open
      Messages sent: 11 (3 control), received: 10 (2 control).
    Interface Extension Cross-link
    ------------------------------
      Channel status: Open
      Messages sent: 4 (3 control), received: 3 (2 control).
….
```

サテライト インベントリのモニタリング

show inventory chassis、**show inventory fans**、**show environment temperatures** の各コマンドを管理コンフィギュレーション モードで使用すると、サテライト インベントリのステータスをモニタできます。

```
RP/0/RSP0/CPU0:shanghai(admin)# show inventory chassis

NAME: "module 0/RSP0/CPU0", DESCR: "ASR9K Fabric, Controller, 4G memory"
PID: A9K-RSP-4G, VID: V02, SN: FOC143781GJ
...
NAME: "fantray SAT100/FT0/SP", DESCR: "ASR9000v"
PID: ASR-9000v-FTA, VID: V00 , SN: CAT1507B228

NAME: "module SAT100/0/CPU0", DESCR: "ASR-9000v GE-SFP Line Card"
PID: ASR-9000v, VID: N/A, SN:
NAME: "module mau GigabitEthernet100/0/CPU0/8", DESCR: "CISCO-AVAGO     "
PID: SFP-GE-S, VID: V01, SN: AGM1424P08N

NAME: "module mau TenGigE100/0/CPU0/3", DESCR: "CISCO-FINISAR   "
PID: SFP-10G-SR, VID: V02, SN: FNS144502Y3

NAME: "power-module SAT100/PM0/SP", DESCR: "ASR-9000v Power Module"
PID: ASR-9000v, VID: N/A, SN:
NAME: "Satellite Chassis ASR-9000v ID 100", DESCR: "ASR9000v"
PID: ASR-9000v-AC-A, VID: V00 , SN: CAT12345678
```

```
RP/0/RSP0/CPU0:sat-host (admin)# show inventory fans

NAME: "fantray 0/FT0/SP", DESCR: "ASR-9006 Fan Tray"
PID: ASR-9006-FAN, VID: V02, SN: FOX1519XHU8

NAME: "fantray 0/FT1/SP", DESCR: "ASR-9006 Fan Tray"
PID: ASR-9006-FAN, VID: V02, SN: FOX1519XHTM

NAME: "fantray SAT100/FT0/SP", DESCR: "ASR9000v"
PID: ASR-9000v-FTA, VID: V01 , SN: CAT1531B4TC
```

```
NAME: "fantray SAT101/FT0/SP", DESCR: "ASR9000v"
PID: ASR-9000v-FTA, VID: V01 , SN: CAT1542B0LJ

NAME: "fantray SAT102/FT0/SP", DESCR: "ASR9000v"
PID: ASR-9000v-FTA, VID: V01 , SN: CAT1531B4T7


RP/0/RSP0/CPU0:sat-host(admin)# show inventory | b GigabitEthernet100/

NAME: "module mau GigabitEthernet100/0/CPU0/0", DESCR: "CISCO-FINISAR   "
PID: SFP-GE-S, VID:  , SN: FNS11350L5E

NAME: "module mau GigabitEthernet100/0/CPU0/1", DESCR: "CISCO-FINISAR   "
PID: SFP-GE-S, VID: V01, SN: FNS0934M290

NAME: "module mau GigabitEthernet100/0/CPU0/2", DESCR: "CISCO-FINISAR   "
PID: SFP-GE-S, VID:  , SN: FNS12280L59


RP/0/RSP0/CPU0:sat-host(admin)# show environment temperatures

R/S/I   Modules Sensor                  (deg C)
0/RSP0/*
        host    Inlet0                  33.1
        host    Hotspot0                46.9

0/RSP1/*
        host    Inlet0                  32.1
        host    Hotspot0                45.9

0/0/*
        host    Inlet0                  37.3
        host    Hotspot0                52.3

0/1/*
        spa0    InletTemp               34.0
        spa0    Hotspot                 34.5

        spa1    LocalTemp               38.0
        spa1    Chan1Temp               36.0
        spa1    Chan2Temp               39.0
        spa1    Chan3Temp               39.0
        spa1    Chan4Temp               48.0

        host    Inlet0                  36.1
        host    Hotspot0                64.0

0/2/*
        host    Inlet0                  39.2
        host    Hotspot0                54.6

0/3/*
        host    Inlet0                  41.3
        host    Hotspot0                48.5

0/FT0/*
        host    Inlet0                  42.3
        host    Hotspot0                36.1

0/FT1/*
        host    Inlet0                  40.4
        host    Hotspot0                35.8

SAT100/FT0/*
```

```
        host    Hotspot0                53.0


SAT101/FT0/*
        host    Hotspot0                56.0


SAT102/FT0/*
        host    Hotspot0                53.0
```

サテライト デバイスのリロード

サテライト デバイスをリロードするには、**hw-module satellite** *satellite id/all* **reload** コマンドを使用します。

例

```
RP/0/RSP0/CPU0:sat-host# hw-module satellite 101 reload

Reload operation completed successfully.
RP/0/RSP0/CPU0:May  3 20:26:51.883 : invmgr[254]: %PLATFORM-INV-6-OIROUT : OIR: Node 101
removed
```

サテライトで設定されるポート レベル パラメータ

これらは、サテライト nV システムで設定できるポート レベル パラメータです。

- 管理状態（shut および no shut）
- イーサネット MTU
- イーサネット MAC アドレス。
- イーサネット リンクのオートネゴシエーション、たとえば
  - 半/全二重
  - Link speed
  - フロー制御
- オートネゴシエーション パラメータの静的設定（速度、デュプレックス、フロー制御など）
- キャリア遅延
- レイヤ 1 パケット ループバック、たとえば
  - ライン ループバック
  - 内部ループバック
- Cisco ASR 9000 シリーズ ルータでのすべてのサテライト アクセス ポート機能。

サテライト ポートでのループバック タイプ

サテライト ポートで設定できるループバック インターフェイスのタイプは 2 つあります。具体的には次のとおりです。

- ライン ループバック
- 内部ループバック

次の図は、サテライトでのこれらのループバック インターフェイス タイプの機能を示します。

図 37　ライン ループバック

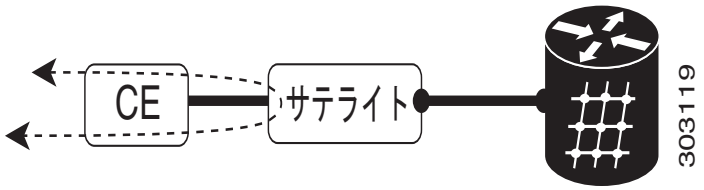
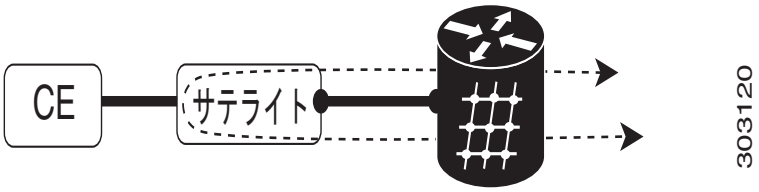


図 38　内部ループバック



次の例に示すように、使用するタイプのループバックを指定できます。

```
Interface GigabitEthernet 100/0/0/0
loopback line | internal
```

サテライト nV システムの設定例

ここでは、次の例について説明します。

- 「サテライト システムの設定：例」（P.721）
  - 「サテライト グローバル コンフィギュレーション」（P.721）
  - 「ICL（サテライト ファブリック リンク）インターフェイス設定」（P.722）
  - 「サテライト インターフェイス設定」（P.722）
  - 「プライベート VRF を使用するサテライト管理」（P.723）

サテライト システムの設定：例

次の例では、サテライト システムの接続を設定するための設定例を示します。

サテライト グローバル コンフィギュレーション

サテライト ID、タイプ、シリアル番号、説明、およびサテライト IP アドレスは、サテライト グローバル コンフィギュレーション サブモードで設定されます。

```
nv
 satellite 100
  type asr9000v
  serial-number CAT1521B1BB
  description milpitas bldg20
  ipv4 address 10.0.0.100
 !
```

```
!
```

ICL（サテライト ファブリック リンク）インターフェイス設定

サテライトに接続されたインターフェイス上で（TenGig またはバンドル インターフェイス）、サテライト ID に関連付けられたポートを指定する必要があります。同じサテライトに接続されているすべてのファブリック リンクに、同じ（ホスト）IPv4 アドレスを使用する必要があります。同じホストからさまざまなサテライトへの接続に使用するホスト IPv4 アドレスは、すべて同一でも、それぞれ異なっていてもかまいません。

```
interface Loopback1000
 ipv4 address 10.0.0.1 255.0.0.0
interface TenGigE0/2/1/0
 description To Sat5 1/46
 ipv4 point-to-point
 ipv4 unnumbered Loopback1000
 nv
  satellite-fabric-link satellite 200
   remote-ports GigabitEthernet 0/0/0-30
  !
 !
!
```

（注） これらの例では、ルータのグローバル VRF からの IP アドレスをサテライト管理トラフィックに使用する方法を示しています。サテライト検出および制御プロトコルの IP 接続で説明したように、これはプライベート VRF を使用して行うこともできます。このようにすると、グローバル VRF との IP アドレスの競合を防止できます。この場合は、例に示したループバック インターフェイスおよび ICL インターフェイスを、サテライト管理トラフィック専用のプライベート VRF に割り当てる必要があります。

サテライト インターフェイス設定

サテライト インターフェイスは、他の通常の GigabitEthernet インターフェイスと同様に使用できます。

```
interface GigabitEthernet200/0/0/0
l2transport
!
!

interface GigabitEthernet200/0/0/0
ip address 99.0.0.1 255.255.255.0
!
!

interface GigabitEthernet200/0/0/2
bundle id 100 mode active
!
!
```

プライベート VRF を使用するサテライト管理

サテライト管理トラフィックに使用されるループバック インターフェイスと ICL を設定するときに、グローバル デフォルト ルーティング テーブルの代わりに特別なプライベート VRF を使用することができます。この VRF の IP アドレスは、そのルータで使用される他のアドレスと競合することはありません。

```
router(config)# vrf NV_MGMT_VRF
router(config)# address ipv4 unicast

router(config)# interface Loopback 1000
router(config)# vrf NV_MGMT_VRF
router(config)# ipv4 address 10.0.0.1 / 24

router(config)# interface TenGige 0/1/0/3
router(config)# vrf NV_MGMT_VRF
router(config)# ipv4 point-to-point
router(config)# ipv4 unnumbered Loopback 1000
router(config)# nv
router(config-nv)# satellite-fabric-link satellite 500
router(config-nv)# remote-ports GigabitEthernet 0/0/28-39
router(config)# nv satellite 500
router(config)# ipv4 address 10.0.0.2 / 24
```

その他の関連資料

ここでは、関連資料の参照先を示します。

関連資料

関連項目	参照先
Cisco IOS XR マスター コマンド リファレンス	『*Cisco IOS XR Master Commands List*』
Cisco IOS XR ソフトウェアでのサテライト システム ソフトウェアのアップグレードとダウングレード	『*Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*』
Cisco IOS XR インターフェイス コンフィギュレーション コマンド	『*Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Command Reference*』
Cisco IOS XR ソフトウェアのサテライト QoS 設定情報	『*Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Configuration Guide*』
サテライト システムでのレイヤ 2 および L2VPN 機能	『*Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide*』
サテライト システムでのレイヤ 3 および L3VPN 機能	『*Cisco ASR 9000 Series Aggregation Services Router MPLS Layer 3 VPN Configuration Guide*』
サテライト システムでのマルチキャスト機能	『*Cisco ASR 9000 Series Aggregation Services Router Multicast Configuration Guide*』
サテライト システムでのブロードバンド ネットワーク ゲートウェイ機能	『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Configuration Guide*』
AAA 関連の情報およびサテライト システムでの設定	『*Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide*』

関連項目	参照先
Cisco ASR 901 ルータの設定	『*Network Virtualization Using Cisco ASR 901 Series Aggregation Services Router as a Satellite*』
ユーザ グループとタスク ID に関する情報	『*Cisco IOS XR System Security Configuration Guide*』の「*Configuring AAA Services on Cisco IOS XR* Software」モジュール

標準

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

MIB

MIB	MIB のリンク
このモジュールに適用できる MIB はありません。	Cisco IOS XR ソフトウェアを使用して選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
なし	該当なし

シスコのテクニカル サポート

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

Cisco ASR 9000 シリーズ ルータでの nV エッジ システムの設定

このモジュールでは、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータでの nV エッジ システムの設定について説明します。

Cisco ASR 9000 シリーズ ルータでの nV エッジ システム設定の機能履歴

リリース	変更内容
リリース 4.2.1	• nV エッジ システムのサポートが Cisco ASR 9000 シリーズ ルータに追加されました。

内容

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。

nV エッジ システムを設定する前に、次のハードウェアおよびソフトウェアがシャーシにインストールされている必要があります。

- ハードウェア：Cisco ASR 9000 シリーズ SPA Interface Processor-700 および Cisco ASR 9000 Enhanced イーサネット ラインカードがサポートされます。Cisco ASR 9000 Enhanced Ethernet ラインカード 10 GigE リンクは、IRL（ラック間リンク）として使用されます。
- ソフトウェア：Cisco ASR 9000 シリーズ ルータの Cisco IOS XR ソフトウェア Release 4.2.1 以上。

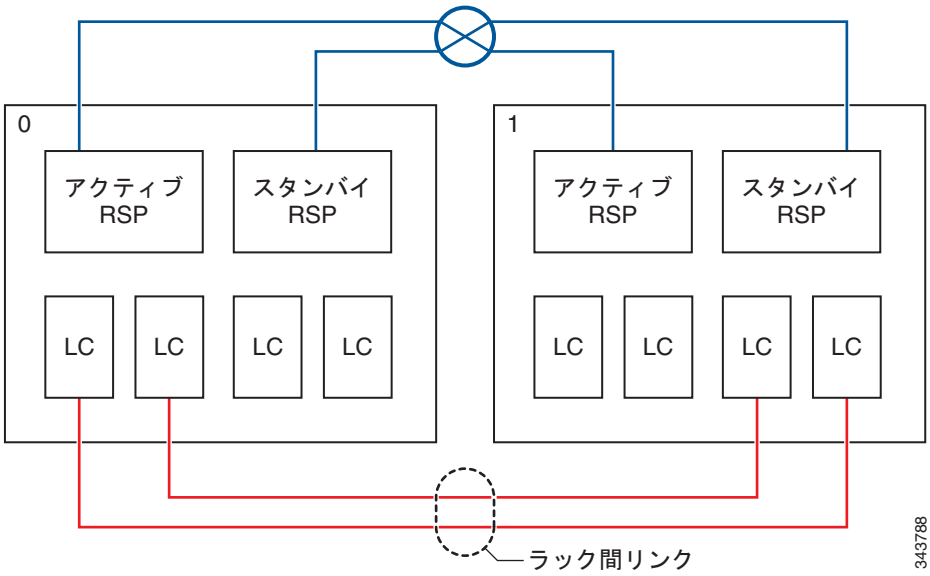
ハードウェア要件の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Hardware Installation Guide*』を参照してください。

Cisco ASR 9000 nV エッジ アーキテクチャの概要

Cisco ASR 9000 シリーズ nV エッジは、2 台以上の Cisco ASR 9000 シリーズ ルータ シャーシで構成されます。これらのシャーシが結合されて、単一の論理的なスイッチングまたはルーティング エンティティを形成します。2 台の Cisco ASR 9000 シリーズ ルータ プラットフォームを単一の仮想 Cisco ASR 9000 シリーズ システムとして運用することができます。実質的に、2 台の物理シャーシが共有コントロール プレーンで論理的にリンクされるので、2 台のルート スイッチ プロセッサ（RSP）が単一のシャーシに収容されているのと同じことになります。図 39 を参照してください。上の青色の線は内部の EOBC 相互接続を表し、下の赤色の線はデータ プレーン相互接続を表します。

その結果、単一ノードの帯域幅容量が倍になり、ハイ アベイラビリティのための複雑なプロトコル ベースの方式は必要なくなります。したがって、サービスとスケーラビリティの要件が最も厳しいときであっても、フェールオーバーを 50 ミリ秒未満で達成できます。

図 39　Cisco ASR 9000 nV エッジ アーキテクチャ



（注）　Cisco IOS XR ソフトウェア Release 4.2.x では、nV エッジ システムのスケーラビリティはシャーシ 2 台までに制限されます。

（注）　Cisco ASR 9000 シリーズ ルータでは、タイプの異なるシャーシが同じ nV エッジ システムに参加できます。Cisco ASR 9000 シリーズ シャーシのすべての組み合わせ（Cisco ASR 9010、Cisco ASR 9006、Cisco ASR 9922、Cisco ASR 9001 など）がサポートされます。

図 40　Cisco ASR 9000 nV エッジ システムにおける EOBC リンク

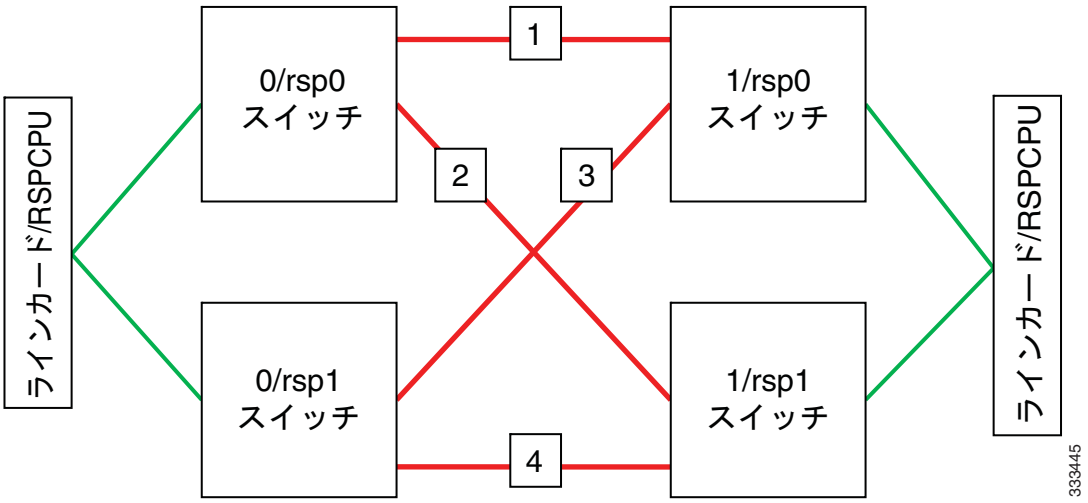


図 40 に示すように、2 台の物理的シャーシがレイヤ 1 10 Gbps 接続でリンクされます。RSP の通信にはレイヤ 1 またはレイヤ 2 イーサネット アウトオブバンド チャネル（EOBC）拡張が使用され、これによって単一の仮想コントロール プレーンが作成されます。各 RSP に 2 個の EOBC ポートと冗長 RSP があるため、シャーシ間の接続は 4 本となります。

シスコの仮想化ネットワーク アーキテクチャは、nV エッジ システムとサテライト デバイスを組み合わせてサテライト nV アーキテクチャを提案しています。サテライト nV モデルの詳細については、「*Cisco ASR 9000 シリーズ ルータでのサテライト ネットワーク仮想化（nV）システムの設定*」の章を参照してください。

Cisco ASR 9000 シリーズ nV エッジ システムのラック間リンク

IRL（ラック間リンク）接続が必要になるのは、あるシャーシから転送されるトラフィックが、nV エッジ システム内の別のシャーシ部分のインターフェイスから送出される場合です。IRL は、10 GigE リンクであることと、直接 L1 接続であることが必要です。IRL が転送パケットに使用されるのは、その入力と出力のインターフェイスがそれぞれ別のラックにある場合です。このようなシャーシ間リンクの最大数は 16 です。2 本以上のリンクが必要であり、これらのリンクは 2 枚のラインカードに存在する必要があります。これは、何らかの障害が原因で一方のラインカードが停止した場合の復元力を高めるためです。「Cisco ASR 9000 nV エッジ アーキテクチャ」を参照してください。

（注） IRL の QoS の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Modular QoS Configuration Guide*』を参照してください。

Cisco ASR 9000 シリーズ nV エッジ システムでの障害検出

Cisco ASR 9000 シリーズ nV エッジ システムでは、プライマリ DSC ノードに障害が発生すると、バックアップ DSC ノードの RSP がプライマリになります。これは、マスター RSP としての処理を実行し、コントロール プレーンのプロセスのアクティブなセットをホスティングします。nV エッジ システムの通常のシナリオでは、プライマリおよびバックアップの DSC ノードがそれぞれ別のラックでホスティングされ、プライマリ DSC の障害検出は、ラック間の通信を介して行われます。

ラック境界を越えて RSP 障害を検出するために、次のメカニズムが使用されます。

- 同じシャーシ内のピア RSP によって検出された FPGA ステート情報が、コントロール リンクを介してブロードキャストされます。この情報が送信されるのは、状態変更が発生したときであり、200 ミリ秒間隔で送信されます。
- ラック間コントロールまたはデータ リンクの UDLD 状態がリモート ラックに送信されます。障害検出の間隔は 500 ミリ秒です。
- キープアライブ メッセージが RSP カード間で、ラック間コントロール リンクを介して送信されます。障害検出時間は 10 秒です。

スプリット ブレインとは、Cisco ASR 9000 シリーズ nV エッジ システム内のルータ同士を結ぶラック間リンクに障害が発生し、その結果として両方のルータがプライマリ ノードとして動作し始める状態です。したがって、スプリット ブレインを検出するために、これらのラック間でメッセージが送信されます。これは、ラック間データ リンクを介して 200 ミリ秒間隔で行われます。

ハイ アベイラビリティのシナリオ

次に示すのは、障害検出のシナリオの例です。

1. プライマリ DSC ノードでの単一 RSP 障害：同じシャーシ内のスタンバイ RSP が最初に、バックプレーン FPGA を通してこの障害を検出します。障害が検出された場合は、この RSP がアクティブ ステートに移行し、障害のことをバックアップ DSC ノードに通知します。これには、シャーシ間コントロール リンク メッセージングが使用されます。
2. プライマリ DSC ノードとスタンバイ ピア RSP の障害：このシナリオが発生する状況にはさまざまなものがあります。たとえば、プライマリ DSC ラックでの電源再投入や、プライマリ ラック内の両方の RSP カードの同時ソフト リセットです。
   a. リモート ラックの障害は最初に、ラック間コントロール リンクでの UDLD 障害によって検出されます。バックアップ DSC ノードは、ラック間データ リンクの UDLD ステートを調べます。データ リンクも障害状態であることからラック障害が確定した場合は、バックアップ DSC ノードがアクティブになります。
   b. UDLD 障害検出は 500 ミリ秒間隔で行われますが、コントロール リンクとデータ リンクの障害の間隔は一定ではありません。これらは RSP およびラインカードによって検出された、それぞれ別の障害であるからです。最大 2 秒間のウィンドウ期間が必要です。これは、コントロール リンクとデータ リンクの障害を相互に関連付けるためと、スプリット ブレイン検出メッセージを受信できるようにするためです。RSP 間のキープアライブ メッセージングには、UDLD 検出によって RSP カードのリセットを検出できなかった場合の冗長検出メカニズムとしての役割があります。
3. ラック間コントロール リンクの障害（スプリット ブレイン）：この障害は最初に、ラック間コントロール リンクでの UDLD プロトコルによって検出されます。この場合は、バックアップ DSC が引き続き、UDLD およびキープアライブ メッセージをラック間データ リンク経由で受信します。シナリオ 2 で説明したように、2 秒間のウィンドウ期間の間に、コントロール リンクとデータ リンクの障害の同期化を行うことができます。データ リンクが障害状態でない場合、つまりスプリット ブレイン パケットが管理 LAN を介して受信されている場合は、スプリット ブレイン状態を回避するためにバックアップ DSC ラックがリロードされます。

Cisco ASR 9000 シリーズ nV エッジ システムの利点

Cisco ASR 9000 シリーズ nV エッジ システムのアーキテクチャには、次の利点があります。

1. Cisco ASR 9000 シリーズ nV エッジ システムは、ネイバー デバイスからは単一のスイッチまたはルータのように見えます。
2. 2 台の物理シャーシを共有コントロール プレーンで論理的にリンクできるので、2 台のルート スイッチ プロセッサ（RSP）が単一のシャーシに収容されているのと同じことになります。その結果、単一ノードの帯域幅容量が倍になり、ハイ アベイラビリティのための複雑なプロトコル ベースの方式は必要なくなります。
3. サービスとスケーラビリティの要件が最も厳しいときであっても、フェールオーバーを 50 ミリ秒未満で達成できます。
4. このクラスタは、2 つのエンティティではなく、単一のエンティティとして管理できます。シャーシが互いを保護するので、復元力が高まります。

5. Cisco nV テクノロジーによって、Cisco ASR 9000 シリーズ ルータの能力を、物理シャーシを超えて拡張できます。これには、リモート仮想ラインカードを使用します。この Small Form-Factor（SFF）Cisco ASR 9000v カードは、数百本のギガビット イーサネット接続をアクセス レイヤおよび集約レイヤで集約できます。
6. ギガビット イーサネット インターフェイス数が数千という規模に拡張するときも、数百あるいは数千のアクセス プラットフォームを個別にプロビジョニングする必要はありません。これはネットワーク アーキテクチャを簡素化し、運用コスト（OpEx）を削減するのに役立ちます。
7. Cisco IOS XR ソフトウェアのマルチシャーシ機能が使用されます。この機能が拡張されて、シャーシの復元力がさらに高まります。これには、データ プレーン、コントロール プレーン、および管理プレーンの保護も含まれており、Cisco ASR 9000 シリーズ nV エッジ システム内のシャーシの 1 つが完全に停止した場合にも備えることができます。
8. 疑似配線を冗長化するのに必要な疑似配線の数を減らすことができます。
9. nV エッジ システムでは、新しいシャーシをシームレスに追加できます。シャーシをシステムに追加したときに、トラフィックの中断やコントロール セッション フラップが発生することはありません。

Cisco ASR 9000 シリーズ nV エッジ システムの制約事項

Cisco ASR 9000 nV エッジ システムには、次のような制約事項があります。

- 第 1 世代 Cisco ASR 9000 イーサネット ラインカードはサポートされません。
- 類似していないタイプのシャーシ同士を接続して nV エッジ システムを形成することはできません。
- シスコがサポートする SFP だけが、すべてのラック間接続に使用できます。
- 10 GigE SFP は、EOBC ポートではサポートされません。
- nV エッジ コントロール プレーンのリンクは、直接物理接続であることが必要です。ネットワークまたは中間ルーティングまたはスイッチング デバイスが間に存在していてはなりません。
- nV エッジ システムは、速度が異なるリンクの混在をサポートしません。

Cisco ASR 9000 シリーズ nV エッジ システムの実装

ここでは、Cisco ASR 9000 シリーズ nV エッジ システムの実装を説明します。

- 「Cisco ASR 9000 nV エッジ システムの設定」（P.731）

Cisco ASR 9000 nV エッジ システムの設定

Cisco ASR 9000 nV クラスタを起動するには、次のサブセクションで説明する手順を実行する必要があります。

単一シャーシからクラスタへの移行

2 台のシャーシがあり、Cisco IOS XR ソフトウェア Release 4.2.x イメージを実行しているとします。この手順では、これらを rack0 および rack1 と呼びます。すでに Cisco IOS XR ソフトウェア Release 4.2.1 以降を実行している場合は、最初の 2 つのステップを省略できます。

1. 各シャーシを個別に Cisco IOS XR ソフトウェア Release 4.2.1 **でターボ ブートする必要があります。**
2. Field Programmable Device（FPD）をアップグレードします。このステップが必要であるのは、Cisco ASR 9000 シリーズ nV エッジの要件として、少なくとも RSP ROMmon が Cisco IOS XR ソフトウェア Release 4.2.1 に対応していることが求められているからです。
3. **情報を収集する**：クラスタに追加する各ラックのシャーシ シリアル番号がわかっていることが必要です。オペレーティング システムで、この情報を **show inventory chassis** コマンドから取得できます。ROMmon のシステムで、シリアル番号を **bpcookie** から取得できます。
4. rack0 の管理コンフィギュレーションを設定するために、次のとおりに入力します。

```
(admin config) # nv edge control serial <rack 0 serial> rack 0
(admin config) # nv edge control serial <rack 1 serial> rack 1
(admin config) # commit
```

5. ラック 0 をリロードします。

6. ラック 1 の電源を切ります。
7. 物理的にルータを接続します。RSP カードの前面パネルにあるシャーシ間コントロール リンク（SFP+ 0 および SFP+ 1 というラベルがあります）を相互に接続します。Rack0-RSP0 を Rack1-RSP0 に接続し、RSP1 についても同様に接続します。ラック 1 が稼働状態になった後で、接続を確認するには、**show nv edge control interface** *loc 0/RSP0/CPU0* コマンドを使用します。

```
RP/0/RSP0/CPU0:ios# show nv edge control switch interface  loc 0/RSP0/CPU0

Priority lPort             Remote_lPort      UDLD STP
======== =====             ============      ==== ========
0        0/RSP0/CPU0/12    1/RSP0/CPU0/12    UP   Forwarding
1        0/RSP0/CPU0/13    1/RSP1/CPU0/13    UP   Blocking
2        0/RSP1/CPU0/12    1/RSP1/CPU0/12    UP   On Partner RSP
3        0/RSP1/CPU0/13    1/RSP0/CPU0/13    UP   On Partner RSP
```

（注） シャーシ間コントロール リンクに対して明示的なコマンドは必要なく、このリンクはデフォルトでオンになっています。

8. ラック 1 を起動します。
9. シャーシ間データ リンクも接続する必要があります。シャーシ間データ リンク インターフェイスとなるように設定する必要があります。設定するには、**nv edge interface** コンフィギュレーション コマンドを 10 ギガビット インターフェイスの下で使用します（10 ギガビットのみ）。この設定がシャーシ間データ リンクの両側（rack0 と rack1）で行われたことを確認します。

（注） シャーシ間データ リンクの動作を確認するには、**show nv edge data forwarding** コマンドを使用します。

10. Rack0 と Rack1 が完全に起動してすべての RSP とラインカードが **XR-RUN** 状態になった後は、**show dsc** コマンドと **show redundancy summary** コマンドの出力は nV エッジ システム設定：例 に示すコマンド出力と類似したものとなっていることが必要です。

nV エッジ システムの設定例

ここでは、次の例を示します。

- 「nV エッジ システム設定：例」（P.732）

nV エッジ システム設定：例

次の例では、Cisco ASR 9000 シリーズ nV エッジ システムの接続を設定するための設定例を示します。

IRL（ラック間リンク）インターフェイス設定

```
interfacetenGigE 0/1/1/1
nv
edge
interface
!
```

10 ギガビット インターフェイスからの Cisco nV エッジ IRL リンク サポート

この例では、te0/2/0/0 と te1/2/0/0 を使用してラック間データ リンクを形成します。

```
RP/0/RSP0/CPU0:cluster_router#show runn interface te1/2/0/0

interface TenGigE1 /2/0/0
nv
edge
data
interface
!

RP/0/RSP0/CPU0:cluster_router#show runn interface te0/2/0/0

interface TenGigE0 /2/0/0
nv
edge
data
interface
!
```

その他の関連資料

次の項では、関連マニュアルの参照先を示します。

関連資料

関連項目	参照先
Cisco IOS XR マスター コマンド リファレンス	『*Cisco IOS XR Master Commands List*』
Cisco IOS XR ソフトウェアでのサテライト システム ソフトウェアのアップグレードとダウングレード	『*Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*』
Cisco IOS XR インターフェイス コンフィギュレーション コマンド	『*Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Command Reference*』
Cisco IOS XR ソフトウェアのサテライト Qos 設定情報	『*Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Configuration Guide*』
サテライト システムでのレイヤ 2 および L2VPN 機能	『*Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide*』
サテライト システムでのレイヤ 3 および L3VPN 機能	『*Cisco ASR 9000 Series Aggregation Services Router MPLS Layer 3 VPN Configuration Guide*』
サテライト システムでのマルチキャスト機能	『*Cisco ASR 9000 Series Aggregation Services Router Multicast Configuration Guide*』
サテライト システムでのブロードバンド ネットワーク ゲートウェイ機能	『*Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Configuration Guide*』
ユーザ グループとタスク ID に関する情報	『*Cisco IOS XR System Security Configuration Guide*』の「*Configuring AAA Services on Cisco IOS XR* Software」モジュール

標準

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

MIB

MIB	説明
CISCO-RF-MIB	DSC シャーシ アクティブ/スタンバイ ノード ペアの情報を提供します。nV エッジ シナリオでは、DSC プライマリ/バックアップ RP 情報およびスイッチオーバー通知を提供します。 Cisco IOS XR ソフトウェアを使用して選択したプラットフォームの MIB を検索およびダウンロードするには、次の URL の Cisco MIB Locator を使用します。 http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml
ENTITY-STATE-MIB	各ノードの冗長性ステート情報を提供します。

MIB	説明
CISCO-ENTITY-STATE-EXT-MIB	冗長性ステータス変更に関する通知（トラップ）を定義する ENTITY-STATE-MIB に対する拡張です。
CISCO-ENTITY-REDUNDANCY-MIB	冗長性グループ タイプ（たとえばノード冗長性グループ タイプ）とプロセス冗長グループ タイプを定義します。

RFC

RFC	タイトル
なし	該当なし

シスコのテクニカル サポート

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

その他の関連資料

INDEX

HC	Cisco IOS XR Interface and Hardware Component Configuration Guide
IC	Cisco IOS XR IP Addresses and Services Configuration Guide
MCC	Cisco IOS XR Multicast Configuration Guide
MNC	Cisco IOS XR System Monitoring Configuration Guide
MPC	Cisco IOS XR MPLS Configuration Guide
QC	Cisco IOS XR Modular Quality of Service Configuration Guide
RC	Cisco IOS XR Routing Configuration Guide
SBC	Cisco IOS XR Session Border Controller Configuration Guide
SC	Cisco IOS XR System Security Configuration Guide
SMC	Cisco IOS XR System Management Configuration Guide
VFC	Cisco IOS XR Virtual Firewall Configuration Guide

A

B

C

D

E

F

H

I

K

L

M

N

O

P

R

S

T

U

V

Y

い

か

き

し

す

ち

て

と

ぬ

ね

の

ふ

ほ

ま

め

り

る

れ

ろ

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/