Cisco ASR 9000 シリーズ アグリゲーション サービス ルータ IP アドレスおよびサービス コンフィギュレーションガイド リリース 4.2

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/

Text Part Number: OL-26068-01-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。

目 次

はじめに

『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Configuration Guide*』の「はじめに」には、次の項があります。

マニュアルの変更履歴

この表に、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-26068-01-J	2011 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。

第 1 章

アクセス リストおよびプレフィックス リストの実装

アクセス コントロール リスト（ACL）は、ネットワーク トラフィック プロファイルをまとめて定義する 1 つ以上のアクセス コントロール エントリ（ACE）です。 このプロファイルはその後、トラフィック フィルタリング、ルート フィルタリング、QoS 分類、アクセス コントロールなど、Cisco IOS XR ソフトウェアの機能で参照できます。 各 ACL には、送信元アドレス、宛先アドレス、プロトコル、およびプロトコルに固有のパラメータなどの基準に基づく、アクション要素（許可または拒否）やフィルタ要素が含まれています。

プレフィックス リストはルート マップおよびルート フィルタリング操作に使用されるほか、ボーダー ゲートウェイ プロトコル（BGP）の多くのルート フィルタリング コマンドではアクセス リストの代わりに使用できます。 プレフィックスは IP アドレスの一部であり、左端のオクテットの左端のビットから始まります。 アドレスの何ビットがプレフィックスに属するかを正確に指定すると、プレフィックスを使用してアドレスを集約し、そのアドレスに対して再配布（フィルタ ルーティング アップデート）などの機能を実行できるようになります。

この章では、次の製品にアクセス リストおよびプレフィックス リストを実装するのに必要な新規のタスクおよび改訂されたタスクについて説明します： Cisco ASR 9000 シリーズ ルータ

（注） この章に記載されているアクセス リストおよびプレフィックス リストのコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』を参照してください。この章で使用される他のコマンドの説明については、コマンド リファレンスのマスター索引を参照するか、またはオンラインで検索してください。

アクセス リストおよびプレフィックス リストの実装の機能履歴

リリース	変更内容
リリース 3.7.2	この機能が導入されました。

アクセスリストおよびプレフィックスリストの実装の前提条件

アクセス リストおよびプレフィックス リストの実装には、次の前提条件が適用されます。

すべてのコマンド タスク ID は、それぞれのコマンド リファレンスと、『Cisco IOS XR Task ID Reference Guide』に記載されています。タスク グループの割り当てについて支援が必要である場合は、システム管理者にお問い合わせてください。

アクセスリストおよびプレフィックスリストの実装の制約事項

アクセス リストおよびプレフィックス リストの実装には、次の制約事項が適用されます。

- IPv4 ACL は、ループバック インターフェイスおよびインターフレックス インターフェイスではサポートされません。
- IPv6 ACL は、ループバック、インターフレックス、および L2 イーサネット フロー ポイント（EFP）のメインまたはサブインターフェイスではサポートされません。

ACL ベース転送（ABF）の実装には、次の制約事項が適用されます。

- ネクスト ホップ オプションを持つ ACL を出方向に接続する設定、ネクスト ホップを持ち出方向に接続された ACL を変更する設定、ネクスト ホップを持つ ACE を拒否する設定のネクスト ホップ設定はサポートされていません。

- リリース 4.2.0 では、A9K-SIP-700 LC および ASR 9000 Enhanced Ethernet LC は ABFv4 および ABFv6 をサポートします。 リリース 4.2.0 では、ASR 9000 Ethernet LC は ABFv6 をサポートせず、ABFv4 のみをサポートします。

（注） これには例外が 1 つあります。 IP to TAG の場合、入力 LC が（ABF ネクスト ホップに基づいて）ラベルを提供するため、パケットはタグ パケットとしてファブリックを横断します。 このようなパケットは、A9K-SIP-700 によって問題なく処理されます。

- 低速パスでは ABF がサポートされないため、NPU から LC CPU へと入力方向にパントされたパケットは ABF では処理されません。
- フラグメンテーションを必要とする IP パケットは、ABF で処理されません。 そのようなパケットは、従来の方法で転送されます。 フラグメント化されたパケットは受信後、ABF によって処理されます。

ハードウェアの制限

- ABF のサポートは、IPv4 およびイーサネット ラインカードのみが対象です。 IPv6 とその他のインターフェイスはサポートされません。
- ABF は入力ラインカードの機能であるため、出力ラインカードは ABF に対応している必要があります。

アクセスリストおよびプレフィックスリストの実装に関する情報

アクセスリストおよびプレフィックスリストを実装するには、次の概念を理解する必要があります。

アクセス リストおよびプレフィックス リスト機能のハイライト

ここでは、アクセス リストとプレフィックス リストの機能のハイライトを示します。

- Cisco IOS XR ソフトウェア 特定のシーケンス番号を指定して、アクセス リストまたはプレフィックス リストのカウンタをクリアできます。
- Cisco IOS XR ソフトウェア 既存のアクセス リストまたはプレフィックス リストの内容を別のアクセス リストまたはプレフィックス リストにコピーできます。

- Cisco IOS XR ソフトウェア permit ステートメントまたは deny ステートメントにシーケンス番号を適用して、名前付きのアクセス リストまたはプレフィックス リストでこのようなステートメントの並べ替え、追加、または削除を実行できます。

（注） 並べ替えは、IPv4 プレフィックス リストのみが対象です。

- Cisco IOS XR ソフトウェア 標準アクセス リストと拡張アクセス リストとを区別しません。標準アクセス リストをサポートしているのは、下位互換性を確保するためです。

IP アクセス リストの目的

アクセスリストは、パケットフィルタリングを実行して、ネットワークを介して移動するパケットとその場所を制御します。この処理は、ネットワークトラフィックを制限したり、ユーザやデバイスによるネットワークへのアクセスを制限したりするのに役立ちます。アクセスリストの用途は多様なので、多くのコマンドの構文でアクセス リストが参照されます。アクセス リストを使用して、次のようなことを実行できます。

- インターフェイスでの着信パケットのフィルタリング
- インターフェイスでの発信パケットのフィルタリング
- ルーティング アップデートの内容の制限
- アドレスまたはプロトコルに基づくデバッグ出力の制限
- vty へのアクセスの制御
- 輻輳回避、輻輳管理、プライオリティ キューイング、カスタム キューイングなどの高度な機能に使用されるトラフィックの特定または分類

IP アクセス リストの機能

アクセス リストは、permit ステートメントと deny ステートメントで構成される順次リストです。これらのステートメントは、IP アドレス、場合によっては上位層 IP プロトコルに適用されます。アクセス リストには、参照に使用される名前があります。多くのソフトウェア コマンドは、構文の一部としてアクセス リストを受け取ります。

アクセスリストを設定して名前を付けることは可能ですが、アクセスリストを受け取るコマンドによってアクセスリストが参照されるまで、有効にはなりません。複数のコマンドから同じアクセス リストを参照できます。アクセス リストで、ルータに到達するトラフィック、またはルータ経由で送信されるトラフィックは制御できますが、ルータが送信元のトラフィックは制御できません。

IP アクセス リストのプロセスとルール

IP アクセス リストを設定するときは、次のプロセスとルールを使用してください。

- アクセス リストの条件に対してフィルタリングされる各パケットの送信元アドレスや宛先アドレス、またはプロトコルがテストされます。一度に 1 つの条件（permit ステートメントまたは deny ステートメント）がテストされます。
- パケットがアクセス リストのステートメントに一致しないと、そのパケットはリスト内の次のステートメントに対してテストされます。
- パケットとアクセス リストのステートメントが一致すると、リスト内の残りのステートメントはスキップされ、パケットは一致したステートメントに指定されたとおりに許可または拒否されます。パケットが許可されるか拒否されるかは、パケットが一致する最初のエントリによって決まります。 つまり、一致すると、それ以降のエントリは考慮されません。
- アクセス リストでアドレスまたはプロトコルが拒否されると、パケットは廃棄され、インターネット制御メッセージ プロトコル（ICMP）ホスト到達不能メッセージが返されます。ICMP は、Cisco IOS XR ソフトウェアで設定できます。
- 各アクセス リストの最後には暗黙の deny ステートメントがあるため、一致する条件がない場合は、パケットはドロップされます。 つまり、各ステートメントに対してテストするときまでにパケットを許可または拒否しないと、パケットは拒否されます。
- アクセス リストには permit ステートメントを 1 つ以上含める必要があります。そうしないと、パケットはすべて拒否されます。
- 最初に一致が見つかった後は条件のテストが終了するため、条件の順序は重要です。 同じ permit ステートメントまたは deny ステートメントでも、順序が異なる場合、ある状況では通過し、別の状況では拒否されるパケットが生じる可能性があります。
- コマンドでアクセス リストを名前によって参照したときに、そのアクセス リストが存在しない場合は、すべてのパケットが通過します。
- 1 つのインターフェイス、1 つのプロトコル、1 つの方向につき、許可されるアクセス リストは 1 つだけです。
- インバウンド アクセス リストは、ルータに到達するパケットを処理します。 着信パケットの処理後に、アウトバウンド インターフェイスへのルーティングが行われます。 インバウンド アクセス リストが効率的なのは、フィルタリング テストで拒否されたことでパケットが廃棄される場合、ルーティング検索のオーバーヘッドが抑えられるためです。パケットがテストで許可されると、そのパケットに対してルーティングの処理が実施されます。インバウンドリストの場合、permit とは、インバウンドインターフェイスでパケットの受信後に処理が続行されることを示します。**deny** とは、パケットが廃棄されることを示します。
- アウトバウンド アクセス リストの場合、パケットの処理後にルータから送信されます。 着信パケットはアウトバウンドインターフェイスにルーティングされてから、アウトバウンド アクセス リストで処理されます。 アウトバウンド リストの場合、許可とは、出力バッファに対して送信されることを示し、拒否とは、パケットが廃棄されることを示します。

- アクセス リストは、使用中のアクセス グループによって適用されている場合には削除できません。 アクセス リストを削除するには、まずアクセス リストを参照しているアクセス グループを削除してから、アクセス リストを削除します。
- **ipv4 access group** コマンドを使用するには、アクセス リストが存在している必要があります。

IP アクセス リストを作成する際に役立つヒント

IP アクセス リストを作成する場合は、次の事項を考慮してください。

- アクセス リストは、インターフェイスに適用する前に作成します。
-
- ネットワークまたはサブネットのより具体的な参照が、より全般的な参照よりも前に出現するように、アクセス リストを構成します。
- 個々のステートメントの用途をひと目で確認および理解しやすくするために、ステートメントの前または後に役立つ注記を書き込みます。

送信元アドレスと宛先アドレス

送信元アドレスと宛先アドレスは、IP パケットの最も一般的な 2 つのフィールドで、アクセス リストの基礎となります。 送信元アドレスを指定して、特定のネットワーキング デバイスまたはホストからのパケットを制御します。宛先アドレスを指定して、特定のネットワーキング デバイスまたはホストに送信されるパケットを制御します。

ワイルドカード マスクと暗黙のワイルドカード マスク

アドレス フィルタリングでは、アクセス リスト エントリ内のアドレス ビットとアクセス リストに送信されるパケットを比較するときに、ワイルドカードマスクを使用して、対応する IP アドレス ビットを確認するか無視するかを指定します。 管理者は、ワイルドカード マスクを慎重に設定することにより、許可または拒否のテストに 1 つまたは複数の IP アドレスを選択できます。

IP アドレス ビット用のワイルドカード マスクでは、数値 1 と数値 0 を使用して、対応する IP アドレス ビットをどのように扱うかを指定します。 1 と 0 は、サブネット（ネットワーク）マスクで意味する内容が逆になるため、ワイルドカード マスクは逆マスクとも呼ばれます。

- ワイルドカード マスク ビット 0 は、対応するビット値を確認することを示します。
- ワイルドカード マスクのビット 1 は、対応するビット値を無視することを意味します。

アクセス リスト ステートメントでは、送信元アドレスまたは宛先アドレスにワイルドカード マスクを指定する必要はありません。 **host** キーワードを使用した場合は、ワイルドカードマスクとして 0.0.0.0 を指定したものと見なされます。

サブネット マスクでは、ネットワークとサブネットを示す隣接ビットをマスクにする必要がありますが、それとは異なり、ワイルドカード マスクではマスクに非隣接ビットを使用できます。IPv6 アクセス リストでは、隣接ビットのみがサポートされます。

ワイルドカード ビットの代わりに、CIDR 形式（/x）を使用することもできます。 たとえば、アドレス 1.2.3.4 0.255.255.255 は 1.2.3.4/8 と表すことができます。

トランスポート層の情報

トランスポート層の情報（パケットが TCP、UDP、ICMP、IGMP のいずれのパケットであるかなどの情報）に基づいてパケットをフィルタリングできます。

IP アクセス リスト エントリ シーケンス番号

IP アクセス リスト エントリにシーケンス番号を適用する機能によって、アクセス リストの変更が簡易になります。 この機能がない頃は、アクセスリスト内のエントリの位置を指定する方法はありませんでした。 以前は、既存のリストの途中にエントリ（ステートメント）を挿入する場合、目的の位置の後にあるすべてのエントリを削除してから、新しいエントリを追加し、削除したすべてのエントリを再入力する必要がありました。 これは手間がかかり、エラーが起こりやすい方法です。

IP アクセス リスト エントリ シーケンス番号機能を使用すると、アクセス リスト エントリにシーケンス番号を追加し、リスト内のエントリを並べ替えることができます。 新しいエントリを追加する場合、アクセスリストの目的の位置に挿入されるようにシーケンス番号を選択します。 必要に応じて、アクセス リストの現在のエントリを並べ替えて、新しいエントリを挿入できる場所を作成できます。

シーケンス番号の動作

ここでは、シーケンス番号の動作を詳しく説明します。

- シーケンス番号のないエントリを複数適用すると、最初のエントリにシーケンス番号 10 が割り当てられ、それ以降のエントリには 10 ずつ増分したシーケンス番号が割り当てられます。 最大シーケンス番号は 2147483646 です。 生成したシーケンス番号がこの最大値を超えると、次のメッセージが表示されます。

```
Exceeded maximum sequence number.
```

- シーケンス番号のないエントリを 1 つ指定すると、アクセス リストの最後のシーケンス番号に 10 を加えたシーケンス番号が割り当てられ、リストの末尾に配置されます。
- ACL エントリは、トラフィック フローにもハードウェアのパフォーマンスにも影響を及ぼすことなく追加できます。
- グローバル コンフィギュレーション モードで新しいアクセス リストを入力すると、そのアクセス リストのシーケンス番号が自動的に生成されます。

- ルート プロセッサ（RP）のエントリとラインカード（LC）のエントリのシーケンス番号を常に同期できるように、分散機能がサポートされています。
- この機能は、名前付きの標準および拡張 IP アクセス リストと連動します。 アクセス リストの名前を番号として指定できるため、番号も使用できます。

IP アクセス リスト ログ メッセージ

Cisco IOS XR ソフトウェア 標準 IP アクセス リストで許可または拒否されたパケットに関するログ メッセージが表示されます。 つまり、パケットがアクセス リストに一致すると、そのパケットに関するログ メッセージ情報がコンソールに送信されます。 ログをコンソールに送信するメッセージのレベルは、グローバル コンフィギュレーション モードの **logging console** コマンドで制御します。

最初にパケットがアクセスリストをトリガーすると、すぐにログメッセージが生成されます。その後、5 分間隔でパケットが収集されて表示または記録されます。 ログ メッセージにはアクセス リスト番号、パケットの許可または拒否に関する状況、パケットの送信元 IP アドレス、および直前の 5 分間に許可または拒否された送信元からのパケット数が示されます。

ただし、{ **ipv4 | ipv6** } **access-list log-update threshold** コマンドを使用すると、アクセス リストに一致したパケットを許可または拒否する際に、ログ メッセージを生成するパケットの数を設定できます。 この手順は、5 分間隔よりも短い頻度でログ メッセージを受信する場合に実行することを推奨します。

注意 *number-of-matches* 引数を 1 に設定すると、ログ メッセージはキャッシュされずにただちに送信されます。この場合、アクセス リストに一致するすべてのパケットについてログ メッセージが生成されます。 大量のログ メッセージでシステムが過負荷になる可能性があるため、1 に設定することは推奨されません。

{ ipv4 | ipv6} access-list log-update threshold コマンドを使用する場合でも、5 分タイマーは有効なままなので、各キャッシュのメッセージ数に関係なく、5 分が経過すると各キャッシュは空になります。 ログメッセージを送信するタイミングに関係なく、しきい値が指定されていない場合と同様に、ログ メッセージのキャッシュは消去され、カウントは 0 にリセットされます。

（注） ログ メッセージが多すぎて処理できない場合や、1 秒以内に 2 つ以上のログ メッセージを処理した場合には、ログ メッセージ パケットの一部がドロップされることがあります。 この動作により、ログを生成するパケットの数が多くなっても、ルータが CPU サイクルを過度に使用することはありません。 したがって、ロギング機能は課金ツールや、アクセス リストとの一致数を正確に把握するための情報源として使用しないでください。

フラグメント制御付き拡張アクセス リスト

この機能が導入される前、非フラグメント パケットと、パケットの先頭フラグメントは、IP 拡張アクセスリストで処理していました（このようなアクセスリストを適用した場合）が、先頭以外のフラグメントはデフォルトで許可されていました。 フラグメント制御付き IP 拡張アクセス リスト機能により、先頭以外のパケットもさらにきめ細かく制御できるようになりました。 IP 拡張アクセスリストを適用するときに、パケットの先頭以外の IP フラグメントを調べるかどうかを指定できます。

先頭以外のフラグメントにはレイヤ 3 情報のみが含まれているため、レイヤ 3 情報のみが含まれるアクセス リスト エントリを先頭以外のフラグメントに適用できるようになりました。 フラグメントにはフィルタリングに必要な情報がすべて揃っており、それでエントリをフラグメントに適用できるというわけです。

この機能により、オプションの **fragments** キーワードが、IP アクセス リスト コマンドの **deny (IPv4)**、**permit (IPv4)**、**deny (IPv6)**、**permit (IPv6)** に追加されています。 アクセス リスト エントリに **fragments** キーワードを指定することにより、その特定のアクセス リスト エントリは、パケットの先頭以外のフラグメントにのみ適用されます。フラグメントは、指定内容に応じて許可または拒否されます。

fragments キーワードの有無に応じたアクセス リスト エントリの動作をまとめると、次のようになります。

アクセス リスト エントリの状態	結果
fragments キーワードがなく、すべてのアクセス リスト エントリ情報が一致する	アクセス リスト エントリにレイヤ 3 情報のみが含まれている場合： • エントリは、非フラグメント パケット、先頭フラグメント、先頭以外のフラグメントに適用されます。 アクセス リスト エントリにレイヤ 3 情報とレイヤ 4 情報が含まれている場合： • エントリは、非フラグメント パケットと先頭フラグメントに適用されます。 ◦ エントリが一致し、かつ permit ステートメントである場合、パケットまたはフラグメントは許可されます。 ◦ エントリが一致し、かつ deny ステートメントである場合、パケットまたはフラグメントは拒否されます。 • エントリは、次の方法で先頭以外のフラグメントにも適用されます。 先頭以外のフラグメントにはレイヤ 3 情報のみが含まれているため、アクセス リスト エントリのレイヤ 3 の部分のみが適用されます。 アクセス リスト エントリのレイヤ 3 の部分が一致し、 ◦ エントリが permit ステートメントである場合、先頭以外のフラグメントは許可されます。 ◦ エントリが deny ステートメントの場合は、次のアクセスリスト エントリが処理されます。 （注） 先頭以外のフラグメントと非フラグメントや先頭フラグメントとでは、deny ステートメントの処理が異なることに注意してください。

アクセス リスト エントリの状態	結果
fragments キーワードがあり、すべてのアクセス リスト エントリ情報が一致する	アクセス リスト エントリは、先頭以外のフラグメントにのみ適用されます。 （注） レイヤ 4 情報を含むアクセス リスト エントリに **fragments** キーワードは設定できません。

すべてのアクセス リスト エントリに **fragments** キーワードを追加しないでください。IP パケットの先頭フラグメントは非フラグメントと見なされ、それ以降のフラグメントとは独立して扱われるためです。 先頭フラグメントは **fragments** キーワードが含まれているアクセス リスト permit エントリまたは deny エントリとは一致しないため、パケットは次のアクセス リスト エントリと比較されます。この比較は、**fragments** キーワードが含まれていないアクセス リスト エントリによってパケットが許可または拒否されるまで続きます。 したがって、deny エントリごとに、2 つのアクセス リスト エントリが必要になる場合があります。 ペアの最初の deny エントリには **fragments** キーワードは含まれず、初期フラグメントに適用されます。 ペアの 2 番めの deny エントリには **fragments** キーワードは含まれ、以降のフラグメントに適用されます。 同じホストに複数の **deny** アクセス リスト エントリがあり、それぞれのレイヤ 4 ポートが異なる場合、そのホストに追加する必要があるのは、**fragments** キーワードを指定した deny アクセス リスト エントリ 1 つだけです。 このように、パケットのすべてのフラグメントは、アクセス リストによって同様に扱われます。

IP データグラムのパケット フラグメントは個々のパケットと見なされ、各フラグメントはアクセス リスト アカウンティングとアクセス リスト違反カウントの 1 つのパケットとして個別にカウントされます。

（注） アクセス リストおよび IP フラグメントに関するあらゆるケースを **fragments** キーワードで解決できるわけではありません。

ポリシー ルーティング

ポリシー ルーティングが **match ip address** コマンドに基づくものであり、アクセス リストのエントリがレイヤ 4 ～ レイヤ 7 の情報に一致した場合、フラグメンテーションとフラグメント制御機能はポリシー ルーティングに影響を及ぼします。 先頭フラグメントがポリシー ルーティングされなかった場合でも、先頭以外のフラグメントがアクセス リストを通過し、ポリシー ルーティングされることがあります。その逆もまた同じです。

前に説明したようにアクセス リスト エントリに **fragments** キーワードを使用すると、先頭フラグメントと先頭以外のフラグメントに対するアクションの照合を改善できるため、ポリシー ルーティングが想定どおりに機能する可能性が高くなります。

アクセス リストのエントリに関するコメント

remark アクセス リスト コンフィギュレーション コマンドを使用すると、名前付き IP アクセス リストにエントリに関するコメント（注釈）を含めることができます。 コメントを含めると、ネットワーク管理者がアクセス リストを理解し、精査しやすくなります。 1 つのコメント行の最大長は 255 文字です。

コメントは、**permit** ステートメントまたは **deny** ステートメントの前後どちらにでも配置できます。 コメントがどの **permit** ステートメントまたは **deny** ステートメントの説明であるのかが明確になるように、コメントの位置に関して一貫性を保つようにしてください。 たとえば、あるコメントは対応する **permit** または **deny** ステートメントの前にあり、他のコメントは対応するステートメントの後ろにあると、混乱を招きます。 コメントに順番を付けることができます。

アクセスリストの作成後、アクセスリストをインターフェイスまたは端末回線に適用することを忘れないでください。 詳細については、アクセス リストの適用, （18 ページ）を参照してください。

アクセス コントロール リスト カウンタ

Cisco IOS XR ソフトウェアでは、ACL カウンタがハードウェアとソフトウェアの両方で維持されます。 ハードウェア カウンタは、アクセス グループをインターフェイスに適用するなど、パケット フィルタリングの用途に使用します。 ソフトウェア カウンタは、主にソフトウェア パケット処理に関するあらゆる用途に使用できます。

パケット フィルタリングでは、ACE ごとに 64 ビットのハードウェア カウンタが使用されます。同じラインカードにある所定の方向のインターフェイスに同じアクセス グループを適用した場合、ACL のハードウェア カウンタは 2 つのインターフェイス間で共有されます。

特定のアクセス グループのハードウェア カウンタを表示するには、EXEC モードで **show access-lists ipv4** [*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**location** *node-id*}] コマンドを使用します。

ハードウェア カウンタをクリアするには、EXEC モードで **clear access-list ipv4** *access-list-name* [**hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**location** *node-id*}] コマンドを使用します。

わずかながらパフォーマンスが低下するため、IPv4 ACL に対するハードウェア カウントはデフォルトでは無効になっています。ハードウェア カウント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 **ipv4 access-group** *access-list-name* {**ingress** | **egress**} [**hardware-count**] コマンドを使用します。 このコマンドは必要に応じて使用できるため、カウントは指定したインターフェイスに対してのみイネーブルになります。

ソフトウェア カウンタは、ソフトウェアがパケットを処理すると更新されます。たとえば、例外パケットを LC CPU にパントして処理した場合や、ルーティング プロトコルが ACL を使用した場合などです。 維持されるソフトウェア カウンタというのは、その ACL を使用するすべてのソフトウェア アプリケーションの集合体です。 ソフトウェア専用の ACL カウンタを表示するには、EXEC モードで **show access-lists ipv4** *access-list-name* [**sequence** *number*] コマンドを使用します。

ここに挙げた情報は、ハードウェア カウントが常にイネーブルになっていることを除いて、すべてIPv6にも当てはまります。IPv6アクセス グループのコマンドライン インターフェイス（CLI）には **hardware-count** オプションがありません。

プレフィックス リストを使用した BGP フィルタリング

プレフィックス リストは、BGP ルート フィルタリング コマンドの多くでアクセス リストの代わりに使用できます。 プレフィックス リストを使用した場合の利点は次のとおりです。

- サイズの大きなリストをロードしてルート ルックアップを実施する場合のパフォーマンスが大幅に向上します。
- 差分更新がサポートされます。
- CLI の使い勝手が向上します。 アクセス リストを使用して BGP 更新をフィルタリングするためのCLIは、パケット フィルタリング形式を使用しているため、わかりにくく使い勝手もよくありません。
- 柔軟性が高まります。

コマンドでプレフィックス リストを使用するには、あらかじめプレフィックス リストをセットアップしておく必要があります。プレフィックス リストのエントリには、シーケンス番号を割り当ててください。

プレフィックス リストでトラフィックをフィルタリングする仕組み

プレフィックス リストによるフィルタリングでは、ルートのプレフィックスが、プレフィックス リストに記載されているプレフィックスと照合されます。 一致すると、一致したルートが使用されます。 具体的には、プレフィックスを許可するか、拒否するかは次のルールに基づきます。

- 空のプレフィックス リストはすべてのプレフィックスを許可します。
- 特定のプレフィックスがプレフィックス リストのどのエントリとも一致しなかった場合、暗黙の deny が適用されます。
- プレフィックス リストの複数のエントリが特定のプレフィックスと一致したときは、最も長く、最も具体的な一致が選択されます。

シーケンス番号は自動的に生成されます。ただし、この自動生成をディセーブルにしている場合を除きます。 シーケンス番号の自動生成をディセーブルにしている場合は、IPv4 または IPv6 のプレフィックス リスト コンフィギュレーション コマンドの **permit** コマンドおよび **deny** コマンドで *sequence-number* 引数を使用して、各エントリのシーケンス番号を指定する必要があります。プレフィックス リストのエントリを削除するには、*sequence-number* 引数を指定した **permit** コマンドまたは **deny** コマンドの **no** 形式を使用してください。

show コマンドの出力には、シーケンス番号が含まれます。

ACL ベース転送の実装に関する情報

アクセスリストおよびプレフィックスリストを実装するには、次の概念を理解する必要があります。

ACL ベース転送の概要

統合ネットワークは、音声、ビデオ、およびデータを伝送します。 トラフィックによっては、ルーティング プロトコルが算出したパスを使用するのではなく、特定のパスにルーティングすることが必要になる場合があります。 これを実現するための簡単なソリューションは、ACL設定にネクストホップ アドレスを指定することです。これで、パケットベースで宛先アドレスをルックアップするのではなく、ACLに設定したネクストホップアドレスを使用して指定の宛先にパケットを転送できるようになります。 ACL設定でネクストホップを使用して転送するというこの機能は、ACL ベース転送（ABF）と呼ばれます。

ACL ベース転送を使用すると、ブロードキャスト TV over IP、IP テレフォニー、データなどを対象としたサービスを複数のプロバイダーから選択することが可能になり、カフェテリア形式でインターネットにアクセスできます。 サービス プロバイダーは、ユーザ トラフィックをさまざまなコンテンツ プロバイダーに迂回させることができます。

ABF-OT

ユーザが適切なネクスト ホップを柔軟に選択できるようにするため、ABF の機能が強化され、オブジェクト トラッキング（OT）と情報をやり取りできるようになりました。これは、次の機能に影響を及ぼします。

- CEF でのプレフィックスのトラッキング
- ライン ステート プロトコルのトラッキング
- IPSLA（IP サービス レベル契約）

オブジェクト トラッキングでの IPSLA のサポート

OT モジュールは、IPSLA モジュールとやり取りして到達可能性情報を取得します。 ルータは、IPSLA を使って定期的に測定を実施します。

アクセスリストおよびプレフィックスリストの実装方法

Cisco ASR 9000 SIP 700 ラインカードおよび ASR 9000 イーサネット ラインカードで IPv6 ACL をサポートするようになりました。 これに関連する基準は次のとおりです。

- ACL 対応のインターフェイス：1000（各方向 500 ずつ）、ASR 9000 イーサネット ラインカードの場合は 4000
- 一意の ACL：512（それぞれに 5 個の ACE）、ASR 9000 イーサネット ラインカードの場合は 2000
- ACL あたりの最大 ACE 数：8000（ASR 9000 イーサネット ラインカードの場合は、LC モデルに基づいて 16000、8000、4000 のいずれか）
- IPv6 ACL ログも、今後サポートする予定です。

ここでは、次の手順について説明します。

拡張アクセス リストの設定

このタスクでは、拡張 IPv4 または IPv6 アクセス リストを設定します。

手順の概要

1. **configure**
2. {**ipv4** | **ipv6**} **access-list** *name*
3. [*sequence-number*] **remark** *remark*
4. 次のいずれかを実行します。
   - [*sequence-number*]{**permit** | **deny**} *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
   - [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix*/*prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* {*port* | *protocol-port*}] [**dscp** *value*] [*routing*] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
5. 必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
7. **show access-lists** {**ipv4** | **ipv6**} [*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter 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	{**ipv4** | **ipv6**} **access-list** *name*
	
	例：
	
	`RP/0/RSP0/CPU0:router(config)# ipv4 access-list acl_1`
	または
	
	`RP/0/RSP0/CPU0:router(config)# ipv6 access-list acl_2`
		IPv4 または IPv6 アクセス リスト コンフィギュレーション モードを開始し、名前付きアクセス リストを設定します。

ステップ 3	[*sequence-number*] **remark** *remark* 例： `RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 remark Do not allow user1 to telnet out`	（任意）名前付きのアクセス リストに **permit** ステートメントまたは **deny** ステートメントに関するコメントを書くことができます。 • 注釈は最大 255 文字まで可能で、これより長い文字は切り捨てられます。 • **permit** ステートメントまたは **deny** ステートメントの前後どちらにも設定できますが、どちらかの位置に統一することを推奨します。
ステップ 4	次のいずれかを実行します。 • [*sequence-number*]{**permit** | **deny**} *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**] • [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix*/*prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* {*port* | *protocol-port*}] [**dscp** *value*] [*routing*] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]	IPv4 アクセス リスト acl_1 に許可または拒否の条件を 1 つ以上指定します。 • 任意の **log** キーワードを指定すると、エントリに一致したパケットに関する情報ログ メッセージがコンソールに送信されます。 • 任意の **log-input** キーワードは、ログ メッセージに入力インターフェイスも含まれることを除いて、**log** キーワードと同じように機能します。 または IPv6 アクセス リスト acl_2 に許可または拒否の条件を 1 つ以上指定します。 • IPv6 オプション ヘッダーおよび任意の上位層プロトコル タイプ情報に基づいて IPv6 トラフィックをフィルタリングする方法の詳細については、**deny**（IPv6）コマンドおよび **permit**（IPv6）コマンドを参照してください。

	コマンドまたはアクション	目的

	例：
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit 172.16.0.0 0.0.255.255`
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 deny 192.168.34.0 0.0.0.255`
	または
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# 20 permit icmp any any`
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# 30 deny tcp any any gt 5000`
		（注） どの IPv6 アドレス リストにも、ネイバー アドバタイズメントおよび送信要求に使用される暗黙の permit 2 つあります。それは暗黙的ネイバー探索ネイバーアドバイタイズメント（NDNA）と暗黙的ネイバー探索ネイバー送信要求（NDNS）です。
		（注） どの IPv6 アクセス リストにも最後の一致条件として暗黙の **deny ipv6 any any** ステートメントがあります。1 つの IPv6 アクセス リストには、暗黙の **deny ipv6 any any** ステートメントを有効にするために少なくとも 1 つのエントリが含まれる必要があります。

ステップ 5	必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。	アクセス リストは変更できます。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **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 7	**show access-lists** {**ipv4** | **ipv6**} [*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location**	（任意）現在の IPv4 または IPv6 アクセス リストの内容を表示します。

	コマンドまたはアクション	目的
	node-id} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter location** *node-id*}]] 例： `RP/0/RSP0/CPU0:router# show access-lists ipv4 acl_1`	• 特定のアクセス リストの内容を表示するには、*access-list-name* 引数を使用します。 • 方向（入力または出力）とアクセスリストを指定して、それを使用するすべてのインターフェイスのハードウェアの内容とカウンタを表示するには、**hardware**、**ingress** または **egress**、および **location** または **sequence** の各キーワードを使用します。 インターフェイスのアクセス グループを設定するには、イネーブルにするアクセス リスト ハードウェア カウンタに対して **ipv4 access-group** コマンドを使用します。 • 現在の IPv4 または IPv6 アクセス リストをまとめたサマリーを表示するには、**summary** キーワードを使用します。 • インターフェイスの統計情報を表示するには、**interface** キーワードを使用します。

次の作業

アクセス リストを作成したら、回線またはインターフェイスに適用する必要があります。アクセス リストを適用する方法については、アクセス リストの適用, （18 ページ）を参照してください。

一意のアクセス リスト エントリ（ACE）の追加または削除中に、ACL コミットが失敗します。これは、割り当てられたマネージャ プロセスが存在しないために発生します。 config-ipv4-acl モードを終了してコンフィギュレーション モードに戻り、再び config-ipv4-acl モードを開始してから、最初の ACE を追加してください。

アクセス リストの適用

作成したアクセス リストを機能させるには、そのアクセス リストを参照する必要があります。アクセス リストは、発信インターフェイスまたは着信インターフェイスに適用できます。 ここでは、端末回線とネットワーク インターフェイスの両方に対してこのタスクを実行するためのガイドラインを示します。

すべての仮想端末回線にユーザが接続する可能性があるため、すべての仮想端末回線に同じ制約を設定する必要があります。

着信アクセス リストの場合、パケットの受信後、Cisco IOS XR ソフトウェアはアクセス リストに照らしてそのパケットの送信元アドレスをチェックします。アクセス リストがアドレスを許可している場合は、パケットの処理を継続します。 アクセス リストがアドレスを拒否している場合

は、パケットを廃棄し、ICMPホスト到達不能メッセージを返します。 ICMPメッセージは設定可能です。

発信アクセス リストの場合、パケットを受信して管理下のインターフェイスに転送した後、アクセス リストに照らしてパケットの送信元アドレスをチェックします。 アクセス リストがアドレスを許可している場合は、パケットを送信します。アクセスリストがアドレスを拒否している場合は、パケットを廃棄し、ICMP ホスト到達不能メッセージを返します。

まだ定義されていないアクセスリストをインターフェイスに適用すると、アクセスリストがまだインターフェイスに適用されていないものと解釈し、すべてのパケットを容認します。 ネットワークで未定義のアクセス リストをセキュリティの手段として使用する場合は、この動作に留意してください。

インターフェイスへのアクセスの制御

このタスクでは、アクセス リストをインターフェイスに適用して、そのインターフェイスへのアクセスを制限します。

アクセス リストは、発信インターフェイスまたは着信インターフェイスに適用できます。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. 次のいずれかを実行します。
   - **ipv4 access-group** *access-list-name* {**ingress** | **egress**} [**hardware-count**] [**interface-statistics**]
   - **ipv6 access-group** *access-list-name* {**ingress** | **egress**} [**interface-statistics**]
4. 次のいずれかを実行します。
   - end
   - 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。

	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 gigabitethernet 0/2/0/2`	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。 • *type* 引数には、インターフェイス タイプを指定します。 インターフェイス タイプの詳細については、疑問符（?）オンライン ヘルプ機能を使用してください。 • *instance* 引数には、物理インターフェイス インスタンスまたは仮想インスタンスを指定します。 ◦物理インターフェイス インスタンスの表記方法は *rack/slot/module/port* です。 値を区切るスラッシュ（/）は、表記の一部として必要です。 ◦仮想インターフェイス インスタンスの数値範囲は、インターフェイス タイプによって異なります。

ステップ 3	次のいずれかを実行します。
	• **ipv4 access-group** *access-list-name* {**ingress** | **egress**} [**hardware-count**] [**interface-statistics**]
	• **ipv6 access-group** *access-list-name* {**ingress** | **egress**} [**interface-statistics**]
	
	例：
	`RP/0/RSP0/CPU0:router(config-if)# ipv4 access-group p-in-filter in`
	
	`RP/0/RSP0/CPU0:router(config-if)# ipv4 access-group p-out-filter out`
		インターフェイスへのアクセスを制御します。
		• 特定の IPv4 または IPv6 アクセス リストを指定するには、*access-list-name* 引数を使用します。
		• 着信パケットをフィルタリングするには **in** キーワードを使用し、発信パケットをフィルタリングするには **out** キーワードを使用します。
		• IPv4 アクセス グループのハードウェア カウンタをイネーブルにするには、**hardware-count** キーワードを使用します。
		◦IPv6 アクセス グループのハードウェア カウンタは、自動的にイネーブルになります。
		• ハードウェアにインターフェイスごとの統計情報を指定するには、**interface-statistics** キーワードを使用します。
		
		この例では、GigabitEthernet 0/2/0/2 から発着信されるパケットにフィルタを適用します。

ステップ 4	次のいずれかを実行します。 • end • commit 例： `RP/0/RSP0/CPU0:router(config-if)# end`	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]:`

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config-if)# commit`	◦ **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。

回線へのアクセスの制御

このタスクでは、回線にアクセス リストを適用して、その回線へのアクセスを制御します。

手順の概要

1. **configure**
2. **line** {**aux** | **console** | **default** | **template** *template-name*}
3. **access-class** *list-name*{**ingress** | **egress**}
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **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。

	コマンドまたはアクション	目的
ステップ 2	**line** {**aux** | **console** | **default** | **template** *template-name*} 例： `RP/0/RSP0/CPU0:router(config)# line default`	補助、コンソール、デフォルト、またはユーザ定義の回線テンプレートを指定し、回線テンプレート コンフィギュレーション モードを開始します。 • ライン テンプレートは、物理端末回線接続（コンソール ポートおよびAUX ポート）およびVTY 接続を設定して管理するために使用する属性のコレクションです。 Cisco IOS XR ソフトウェアでは、次のテンプレートを使用できます。 ◦補助回線テンプレート：補助回線に適用される回線テンプレート。 ◦コンソール ライン テンプレート：コンソール回線に適用されます。 ◦デフォルト ライン テンプレート：物理および仮想端末回線に適用されます。 ◦ユーザ定義ライン テンプレート：仮想端末回線の範囲に適用できます。
ステップ 3	**access-class** *list-name*{**ingress** | **egress**} 例： `RP/0/RSP0/CPU0:router(config-line)# access-class acl_2 out`	IPv4 または IPv6 アクセス リストを使用して、着信接続および発信接続を制限します。 • 例では、IPv6 アクセス リスト acl_2 を使用して、デフォルトの回線テンプレートの発信接続をフィルタリングしています。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		• 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。

プレフィックス リストの設定

このタスクでは、IPv4 または IPv6 プレフィックス リストを設定します。

手順の概要

1. configure
2. {**ipv4** | **ipv6**} **prefix-list** *name*
3. [*sequence-number*] **remark** *remark*
4. [*sequence-number*] {**permit** | **deny**} *network/length* [**ge** *value*] [**le** *value*] [**eq** *value*]
5. 必要に応じてステップ 4 を繰り返します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。
6. 次のいずれかを実行します。
   - end
   - commit
7. 次のいずれかを実行します。
   - **show prefix-list ipv4** [*name*] [*sequence-number*]
   - **show prefix-list ipv6** [*name*] [*sequence-number*] [*summary*]
8. **clear** {**ipv4** | **ipv6**} **prefix-list** *name* [*sequence-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。

	コマンドまたはアクション	目的

ステップ 2	{**ipv4** | **ipv6**} **prefix-list** *name*
	
	例：
	`RP/0/RSP0/CPU0:router(config)# ipv4 prefix-list pfx_1`
	または
	`RP/0/RSP0/CPU0:router(config)# ipv6 prefix-list pfx_2`
		IPv4 または IPv6 プレフィックス リスト コンフィギュレーションモードを開始し、名前付きプレフィックス リストを設定します。
		• プレフィックス リストを作成するには、少なくとも 1 つの **permit** 句または **deny** 句を入力する必要があります。
		• プレフィックス リストのエントリをすべて削除するには、**no** {**ipv4** | **ipv6**} **prefix-list** *name* コマンドを使用します。

ステップ 3	[*sequence-number*] **remark** *remark*
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv4_pfx)# 10 remark Deny all routes with a prefix of 10/8`
	`RP/0/RSP0/CPU0:router(config-ipv4_pfx)# 20 deny 10.0.0.0/8 le 32`
		（任意）名前付きのプレフィックス リストに次の **permit** ステートメントまたは **deny** ステートメントに関するコメントを書くことができます。
		• 注釈は最大 255 文字まで可能で、これより長い文字は切り捨てられます。
		• **permit** ステートメントまたは **deny** ステートメントの前後どちらにも設定できますが、どちらかの位置に統一することを推奨します。

ステップ 4	[*sequence-number*] {**permit** | **deny**} *network/length* [**ge** *value*] [**le** *value*] [**eq** *value*] 例： `RP/0/RSP0/CPU0:router(config-ipv6_pfx)# 20 deny 128.0.0.0/8 eq 24`	名前付きプレフィックス リストに許可または拒否の条件を 1 つ以上指定します。 • この例では、プレフィックス リスト pfx_2 の 128.0.0.0/8 の /24 に一致するプレフィックスをすべて拒否します。
ステップ 5	必要に応じてステップ 4 を繰り返します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。	プレフィックス リストは変更できます。

ステップ 6	次のいずれかを実行します。
	• end
	• commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv6_pfx)# end`
	または
	`RP/0/RSP0/CPU0:router(config-ipv6_pfx)# 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

	コマンドまたはアクション	目的
		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 7	次のいずれかを実行します。
	• **show prefix-list ipv4** [*name*] [*sequence-number*]
	• **show prefix-list ipv6** [*name*] [*sequence-number*] [*summary*]
	例：
	`RP/0/RSP0/CPU0:router# show prefix-list ipv4 pfx_1`
	または
	`RP/0/RSP0/CPU0:router# show prefix-list ipv6 pfx_2 summary`
		（任意）現在の IPv4 または IPv6 プレフィックス リストの内容を表示します。
		• 特定のプレフィックス リストの内容を表示するには、*name* 引数を使用します。
		• プレフィックス リスト エントリのシーケンス番号を指定するには、*sequence-number* 引数を使用します。
		• プレフィックス リストの内容のサマリーを表示するには、**summary** キーワードを使用します。

ステップ 8	**clear** {**ipv4** | **ipv6**} **prefix-list** *name* [*sequence-number*] **例**： `RP/0/RSP0/CPU0:router# clear prefix-list ipv4 pfx_1 30`	（任意）IPv4 または IPv6 プレフィックス リストのヒット カウントをクリアします。 （注） ヒット カウントは、特定のプレフィックス リスト エントリに一致する数を示す値です。

標準アクセス リストの設定

このタスクでは、標準 IPv4 アクセス リストを設定します。

標準アクセス リストでは、照合操作に送信元アドレスを使用します。

手順の概要

1. **configure**
2. **ipv4 access-list** *name*
3. [*sequence-number*] **remark** *remark*
4. [*sequence-number*] {**permit** | **deny**} *source* [*source-wildcard*] [**log** | **log-input**]
5. 必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。
6. 次のいずれかを実行します。
   - end
   - commit
7. **show access-lists** [**ipv4** | **ipv6**] [*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter 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	**ipv4 access-list** *name* 例： `RP/0/RSP0/CPU0:router# ipv4 access-list acl_1`	IPv4 アクセス リスト コンフィギュレーション モードを開始し、アクセス リスト acl_1 を設定します。
ステップ 3	[*sequence-number*] **remark** *remark* 例： `RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 remark Do not allow user1 to telnet out`	（任意）名前付きのアクセス リストに次の **permit** ステートメントまたは **deny** ステートメントに関するコメントを書くことができます。 • 注釈は最大 255 文字まで可能で、これより長い文字は切り捨てられます。 • **permit** ステートメントまたは **deny** ステートメントの前後どちらにも設定できますが、どちらかの位置に統一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 4	[*sequence-number*] {**permit** | **deny**} *source* [*source-wildcard*] [**log** | **log-input**]
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 permit 172.16.0.0 0.0.255.255`
	または
	`RRP/0/RSP0/CPU0:routerrouter(config-ipv4-acl)# 30 deny 192.168.34.0 0.0.0.255`
		パケットの通過またはドロップを決定する許可または拒否の条件を 1 つ以上指定します。
		• パケットの送信元のネットワークまたはホストの番号を指定するには、*source* 引数を使用します。
		• 送信元に適用するワイルドカード ビットを指定するには、任意の *source-wildcard* 引数を使用します。
		• 任意の **log** キーワードを指定すると、エントリに一致したパケットに関する情報ログ メッセージがコンソールに送信されます。
		• 任意の **log-input** キーワードは、ログ メッセージに入力インターフェイスも含まれることを除いて、**log** キーワードと同じように機能します。

ステップ 5	必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。エントリを削除するには、**no** *sequence-number* コマンドを使用します。	アクセス リストは変更できます。

ステップ 6	次のいずれかを実行します。
	• end
	• commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# end`
	または
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 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。

	コマンドまたはアクション	目的
		• 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 access-lists** [**ipv4** | **ipv6**] [*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter location** *node-id*}]] 例： `RP/0/RSP0/CPU0:router# show access-lists ipv4 acl_1`	（任意）名前付き IPv4 アクセス リストの内容を表示します。 • IPv4 標準アクセス リストの内容は、拡張アクセス リスト形式で表示されます。

次の作業

標準アクセス リストの作成後、それを回線またはインターフェイスに適用する必要があります。アクセス リストを適用する方法については、アクセス リストの適用，（18 ページ）を参照してください。

アクセス リストのコピー

このタスクでは、IPv4 または IPv6 アクセス リストをコピーします。

手順の概要

1. **copy access-list** {**ipv4** | **ipv6**}**source-acl destination-acl**
2. **show access-lists** {**ipv4** | **ipv6**}[*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter 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	**copy access-list** {**ipv4** | **ipv6**}**source-acl destination-acl** 例： `RP/0/RSP0/CPU0:router# copy ipv6 access-list list-1 list-2`	既存の IPv4 または IPv6 アクセス リストのコピーを作成します。 • コピーするアクセス リストの名前を指定するには、*source-acl* 引数を使用します。 • 送信元アクセス リストの内容のコピー先を指定するには、*destination-acl* 引数を使用します。 ◦ *destination-acl* 引数は一意の名前である必要があります。アクセス リストに *destination-acl* 引数名が存在する場合、そのアクセス リストはコピーされません。
ステップ2	**show access-lists** {**ipv4** | **ipv6**}[*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter location** *node-id*}]] 例： `RP/0/RSP0/CPU0:router# show access-lists ipv4 list-2`	（任意）名前付きの IPv4 または IPv6 アクセス リストの内容を表示します。たとえば、コピー先の内容を検証して、宛先アクセス リスト list-2 に送信元アクセス リスト list-1 の情報がすべて含まれていることを確認できます。

アクセス リスト エントリの順序付けとアクセス リストの変更

このタスクでは、名前付きアクセス リストのエントリにシーケンス番号を割り当てる方法と、アクセス リストに対してエントリの追加または削除を行う方法について説明します。 アクセス リストを変更することを前提に説明します。 アクセス リストの並べ替えは任意です。

手順の概要

1. **resequence access-list** {**ipv4** | **ipv6**} *name* [*base* [*increment*]]
2. configure
3. {**ipv4** | **ipv6**} **access-list** *name*
4. 次のいずれかを実行します。
   - [*sequence-number*] {**permit** | **deny**} *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
   - [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* {*port* | *protocol-port*}] [**dscp** *value*] [**routing**] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
5. 必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。
6. 次のいずれかを実行します。
   - end
   - commit
7. **show access-lists** [**ipv4** | **ipv6**] [*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter 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	**resequence access-list** {**ipv4** | **ipv6**} *name* [*base* [*increment*]] 例： `RP/0/RSP0/CPU0:router# resequence access-list ipv4 acl_3 20 15`	（任意）開始シーケンス番号と、シーケンス番号の増分値を使用して、指定した IPv4 または IPv6 アクセス リストを並べ替えます。 • この例では、acl_3 という名前の IPv4 アクセス リストを並べ替えます。 開始シーケンス番号は 20、増分は 15 です。 増分値を選択しないと、デフォルトの増分値 10 が使用されます。
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。

	コマンドまたはアクション	目的

ステップ 3	{**ipv4** | **ipv6**} **access-list** *name*
	
	例：
	`RP/0/RSP0/CPU0:router(config)# ipv4 access-list acl_1`
	または
	`RP/0/RSP0/CPU0:router(config)# ipv6 access-list acl_2`
		IPv4 または IPv6 アクセス リスト コンフィギュレーション モードを開始し、名前付きアクセス リストを設定します。

ステップ 4	次のいずれかを実行します。
	• [*sequence-number*] {**permit** | **deny**} *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
	• [*sequence-number*] {**permit** | **deny**} *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* {*port* | *protocol-port*}] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*} [*operator* {*port* | *protocol-port*}] [**dscp** *value*] [**routing**] [**authen**] [**destopts**] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit 172.16.0.0 0.0.255.255`
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 deny 192.168.34.0 0.0.0.255`
	または
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# 20 permit icmp any any`
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# 30 deny tcp any any gt 5000`
		IPv4 アクセス リスト acl_1 に許可または拒否の条件を 1 つ以上指定します。
		• 任意の **log** キーワードを指定すると、エントリに一致したパケットに関する情報ログ メッセージがコンソールに送信されます。
		• 任意の **log-input** キーワードは、ログ メッセージに入力インターフェイスも含まれることを除いて、**log** キーワードと同じように機能します。
		• このアクセスリストでは **permit** ステートメントを最初に使用していますが、必要なステートメントの順序に応じて、**deny** ステートメントが最初に使用される可能性もあります。
		
		または
		IPv6 アクセス リスト acl_2 に許可または拒否の条件を 1 つ以上指定します。
		• IPv6 オプション ヘッダーと、ICMP、TCP、UDP などの上位層プロトコルに基づいて IPv6 トラフィックをフィルタリングする方法の詳細については、**permit**（IPv6）コマンドおよび **deny**（IPv6）コマンドを参照してください。
		
		（注） どの IPv6 アクセス リストにも最後の一致条件として暗黙の **deny ipv6 any any** ステートメントがあります。 1 つの IPv6 アクセス リストには、暗黙の **deny ipv6 any any** ステートメントを有効にするために少なくとも 1 つのエントリが含まれる必要があります。

ステップ 5	必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加し	アクセス リストは変更できます。

	コマンドまたはアクション	目的
	ます。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。	

ステップ 6	次のいずれかを実行します。
	• end
	• commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# end`
	または
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 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 7	**show access-lists** [**ipv4** | **ipv6**] [*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter location** *node-id*}]] 例： `RP/0/RSP0/CPU0:router# show access-lists ipv4 acl_1`	（任意）名前付きの IPv4 または IPv6 アクセス リストの内容を表示します。 • 出力をレビューして、アクセス リストに最新情報が含まれていることを確認します。

次の作業

アクセス リストがまだインターフェイスまたは回線に適用されていないか、または他の方法で参照されている場合は、アクセス リストを適用します。 アクセス リストを適用する方法については、アクセス リストの適用, （18 ページ）を参照してください。

プレフィックス リストのコピー

このタスクでは、IPv4 または IPv6 プレフィックス リストをコピーします。

手順の概要

1. **copy prefix-list** {**ipv4** | **ipv6**} *source-name destination-name*
2. 次のいずれかを実行します。
   - **show prefix-list ipv4** [*name*] [*sequence-number*]
   - **show prefix-list ipv6** [*name*] [*sequence-number*] [**summary**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**copy prefix-list** {**ipv4** | **ipv6**} *source-name destination-name* 例： `RP/0/RSP0/CPU0:router# copy prefix-list ipv6 list_1 list_2`	既存の IPv4 または IPv6 プレフィックス リストのコピーを作成します。 • コピーするプレフィックス リストの名前を指定するには *source-name* 引数を使用し、コピー元のプレフィックス リストの内容のコピー先を指定するには、*destination-name* 引数を使用します。 • *destination-name* 引数は、一意の名前である必要があります。*destination-name* 引数名がプレフィックス リストに存在する場合、そのプレフィックス リストはコピーされません。
ステップ 2	次のいずれかを実行します。 • **show prefix-list ipv4** [*name*] [*sequence-number*] • **show prefix-list ipv6** [*name*] [*sequence-number*] [**summary**]	（任意）現在の IPv4 または IPv6 プレフィックス リストの内容を表示します。 • 出力をレビューして、プレフィックス リスト list_2 に list_1 のエントリが含まれていることを確認します。

	コマンドまたはアクション	目的
	例： `RP/0/RSP0/CPU0:router# show prefix-list ipv6 list_2`	

プレフィックス リスト エントリの順序付けとプレフィックス リストの変更

このタスクでは、名前付きプレフィックス リストのエントリにシーケンス番号を割り当てる方法と、プレフィックス リストに対してエントリの追加または削除を行う方法について説明します。プレフィックス リストを変更することを前提に説明します。 プレフィックス リストの並べ替えは任意です。

はじめる前に

（注） IPv6 プレフィックス リストの並べ替えはサポートされません。

手順の概要

1. **resequence prefix-list ipv4** *name* [*base* [*increment*]]
2. configure
3. {**ipv4** | **ipv6**} **prefix-list** *name*
4. [*sequence-number*] {**permit** | **deny**} *network/length* [**ge** *value*] [**le** *value*] [**eq** *value*]
5. 必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。
6. 次のいずれかを実行します。
   - end
   - commit
7. 次のいずれかを実行します。
   - **show prefix-list ipv4** [*name*] [*sequence-number*]
   - **show prefix-list ipv6** [*name*] [*sequence-number*] [**summary**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**resequence prefix-list ipv4** *name* [*base* [*increment*]] 例： `RP/0/RSP0/CPU0:router# resequence prefix-list ipv4 pfx_1 10 15`	（任意）開始シーケンス番号と、シーケンス番号の増分値を使用して、指定したIPv4プレフィックス リストを並べ替えます。 • この例では、pfx_1 というプレフィックス リストを並べ替えます。 開始シーケンス番号は 10、増分は 15 です。
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	{**ipv4** | **ipv6**} **prefix-list** *name* 例： `RP/0/RSP0/CPU0:router(config)# ipv6 prefix-list pfx_2`	IPv4またはIPv6プレフィックス リスト コンフィギュレーション モードを開始し、名前付きプレフィックス リストを設定します。
ステップ 4	[*sequence-number*] {**permit** | **deny**} *network/length* [**ge** *value*] [**le** *value*] [**eq** *value*] 例： `RP/0/RSP0/CPU0:router(config-ipv6_pfx)# 15 deny 128.0.0.0/8 eq 24`	名前付きプレフィックス リストに許可または拒否の条件を 1 つ以上指定します。
ステップ 5	必要に応じてステップ 4 を繰り返し、計画したシーケンス番号でステートメントを追加します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。	プレフィックス リストは変更できます。

ステップ 6	次のいずれかを実行します。
	• end
	• commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv6_pfx)# end`
	または
	`RP/0/RSP0/CPU0:router(config-ipv6_pfx)# 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。

	コマンドまたはアクション	目的
		◦ **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	次のいずれかを実行します。 • **show prefix-list ipv4** [*name*] [*sequence-number*] • **show prefix-list ipv6** [*name*] [*sequence-number*] [**summary**] 例： `RP/0/RSP0/CPU0:router# show prefix-list ipv6 pfx_2`	（任意）現在の IPv4 または IPv6 プレフィックス リストの内容を表示します。 • 出力をレビューして、プレフィックス リスト pfx_2 に新しい情報がすべて含まれていることを確認します。

ACL ベース転送を実装する方法

ここでは、次の手順について説明します。

セキュリティ ACL での ACL ベース転送の設定

セキュリティ ACL で ACL ベース転送を設定するには、次のタスクを実行します。

手順の概要

1. configure
2. **ipv4 access-list** *name*
3. [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**default nexthop1** [**ipv4** *ipv4-address1*] **nexthop2**[**ipv4** *ipv4-address2*] **nexthop3**[**ipv4** *ipv4-address3*]] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**] [[**track** *track-name*] [**ttl** *ttl* [*value1 ... value2*]]
4. 次のいずれかを実行します。
   - end
   - commit
5. **show access-list ipv4** [[*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter 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	**ipv4 access-list** *name* 例： `RP/0/RSP0/CPU0:router(config)# ipv4 access-list security-abf-acl`	IPv4 アクセス リスト コンフィギュレーション モードを開始し、指定したアクセス リストを設定します。

ステップ 3	[*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**default nexthop1** [**ipv4** *ipv4-address1*] **nexthop2**[**ipv4** *ipv4-address2*] **nexthop3**[**ipv4** *ipv4-address3*]] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log**	**log-input**] [[**track** *track-name*] [**ttl** *ttl* [*value1 ... value2*]]
		
		例：
		`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 permit ipv4 10.0.0.0 0.255.255.255 any nexthop 50.1.1.2`
		`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 15 permit ipv4 30.2.1.0 0.0.0.255 any`
		`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 permit ipv4 30.2.0.0 0.0.255.255 any nexthop 40.1.1.2`
		`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 25 permit ipv4 any any`
			IPv4 アクセス リストの条件を設定します。 設定例では、セキュリティ ACL で ACL ベース転送を設定する方法を示しています。
			• **nexthop** キーワードは、このエントリに指定されたネクスト ホップに転送します。

	コマンドまたはアクション	目的

ステップ 4	次のいずれかを実行します。
	• end
	• commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# end`
	または
	`RP/0/RSP0/CPU0:router(config-ipv4-acl)# 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 5	**show access-list ipv4** [[*access-list-name* **hardware** {**ingress** | **egress**} [**interface** *type interface-path-id*] {**sequence** *number* | **location** *node-id*} | **summary** [*access-list-name*] | *access-list-name* [*sequence-number*] | **maximum** [**detail**] [**usage** {**pfilter location** *node-id*}]] 例： `RP/0/RSP0/CPU0:router# show access-lists ipv4 security-abf-acl`	ACL ソフトウェアに関する情報を表示します。

IPSLA-OT の実装

ここでは、次の手順について説明します。

• トラック モードのイネーブル化, （39 ページ）

トラック モードのイネーブル化

手順の概要

1. configure
2. **track** *track-name*
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**track** *track-name* 例： `RP/0/RSP0/CPU0:router(config)# track t1`	トラック コンフィギュレーション モードを開始します。
ステップ 3	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	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。

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	◦ **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。

トラック タイプの設定

ネクストホップ デバイスの可用性をトラッキングするメカニズムには、さまざまな種類があります。 トラッキング タイプには 4 つのタイプがあり、次のものを使用します。

- 回線プロトコル
- リスト
- ルート
- IPSLA

トラッキング タイプの設定（回線プロトコル）

回線プロトコルは、オブジェクトトラッカーコンポーネントがトラッキングできるオブジェクトタイプの 1 つです。 このオブジェクト タイプでは、インターフェイスからの状態変化通知をトラッキングするためのオプションを利用できます。 インターフェイス状態変化通知に基づいて、トラック状態を UP にするか、DOWN にするかを決定します。

手順の概要

1. configure
2. **track** *track-name*
3. **type line-protocol state interface** *type interface-path-id*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**track** *track-name* 例： `RP/0/RSP0/CPU0:router(config)# track t1`	トラック コンフィギュレーション モードを開始します。
ステップ 3	**type line-protocol state interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-track)# type line-protocol state interface tengige 0/4/4/0`	状態変化通知のためにトラッキングする必要があるインターフェイスを設定します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

トラック タイプ（リスト）の設定

リストは、ブール オブジェクト タイプです。 ブールとは、オブジェクト トラッカーでサポートされているさまざまなオブジェクト タイプの組み合わせに対して、ブール AND 演算またはブール OR 演算を実行する機能のことです。

手順の概要

1. configure
2. **track** *track-name*
3. **type list boolean and**
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**track** *track-name* 例： `RP/0/RSP0/CPU0:router(config)# track t1`	トラック コンフィギュレーション モードを開始します。
ステップ 3	**type list boolean and** 例： `RP/0/RSP0/CPU0:router(config-track)# type list boolean and`	ブール AND 演算またはブール OR 演算を実行できるトラック オブジェクトのリストを設定します。
ステップ 4	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	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

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	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。

トラッキング タイプ（ルート）の設定

ルートは、ルート オブジェクト タイプです。 オブジェクト トラッカーは、FIB 通知をトラッキングして、ルート到達可能性およびトラック状態を判断します。

手順の概要

1. configure
2. **track** *track-name*
3. **type route reachability**
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **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。

	コマンドまたはアクション	目的
ステップ 2	**track** *track-name* 例： `RP/0/RSP0/CPU0:router(config)# track t1`	トラック コンフィギュレーション モードを開始します。
ステップ 3	**type route reachability** 例： `RP/0/RSP0/CPU0:router(config-track)# type route reachability`	到達可能性状態を動的に学習する必要があるルートを設定します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

トラッキング タイプの設定（rtr）

IPSLA は、ipsla オブジェクト タイプです。 オブジェクト トラッカーは、ipsla 操作の戻りコードをトラッキングして、トラック状態の変化を判断します。

手順の概要

1. configure
2. **track** *track-name*
3. **type rtr** *ipsla operation id* **reachability**
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**track** *track-name* 例： `RP/0/RSP0/CPU0:router(config)# track t1`	トラック コンフィギュレーション モードを開始します。
ステップ 3	**type rtr** *ipsla operation id* **reachability** 例： `RP/0/RSP0/CPU0:routertype rtr 100 reachability`	到達可能性のためにトラッキングする必要がある ipsla 操作 id を設定します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		◦ **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。

IPv6 ACL 用のピュア ACL ベース転送の設定

手順の概要

1. configure
2. {**ipv6** } **access-list** *name*
3. [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]] [**ttl** *ttl value* [*value1 ... value2*]][**default] nexthop1** [**vrf** *vrf-name1*][**ipv6** *ipv6-address1*] [**nexthop2** [**vrf** *vrf-name2*] [**ipv6** *ipv6-address2*] [nexthop3 [vrf vrf-name3] [**ipv6***ipv6-address3*]]]
4. 次のいずれかを実行します。
   - end
   - 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	{**ipv6** } **access-list** *name* 例： `RP/0/RSP0/CPU0:router(config)# ipv6 access-list security-abf-acl`	IPv6 アクセス リスト コンフィギュレーション モードを開始し、指定したアクセス リストを設定します。

	コマンドまたはアクション	目的
ステップ 3	[*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**dscp** *dscp*] [**fragments**] [*packet-length operator packet-length value*] [**log** | **log-input**]] [**ttl** *ttl value* [*value1 ... value2*]][**default**] **nexthop1** [**vrf** *vrf-name1*][**ipv6** *ipv6-address1*] [**nexthop2** [**vrf** *vrf-name2*] [**ipv6** *ipv6-address2*] [nexthop3 [vrf vrf-name3] [**ipv6***ipv6-address3*]]] 例： `RP/0/RSP0/CPU0:router(config-ipv6-acl)# 10 permit ipv6 any any default nexthop1 vrf vrf_A ipv6 11::1 nexthop2 vrf vrf_B ipv6 nexthop3 vrf vrf_C ipv6 33::3`	IPv6 アクセス リストの条件を設定します。 設定例では、ACL 用にピュア ACL ベース転送を設定する方法を示しています。 • このエントリに指定されたネクストホップに転送します。

ステップ 4	次のいずれかを実行します。
	• end
	• commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# end`
	または
	`RP/0/RSP0/CPU0:router(config-ipv6-acl)# 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。

アクセスリストおよびプレフィックスリストの実装の設定例

ここでは、次の設定例について説明します。

アクセス リストのエントリの並べ替え：例

次に、アクセス リストを並べ替える例を示します。 並べ替え後のアクセス リストの開始値は 10 で、増分値は 20 です。 後続のエントリはユーザ指定の増分値に基づいて並べられています。範囲は 1 ～ 2147483646 です。

シーケンス番号のないエントリが入力されると、デフォルトで、アクセス リストの最後のエントリのシーケンス番号に 10 を加えたシーケンス番号が割り当てられます。

```
ipv4 access-list acl_1
10 permit ip host 10.3.3.3 host 172.16.5.34
20 permit icmp any any
30 permit tcp any host 10.3.3.3
40 permit ip host 10.4.4.4 any
60 permit ip host 172.16.2.2 host 10.3.3.12
70 permit ip host 10.3.3.3 any log
80 permit tcp host 10.3.3.3 host 10.1.2.2
100 permit ip any any

configure
 ipv4 access-list acl_1
 end
resequence ipv4 access-list acl_1 10 20

ipv4 access-list acl_1
10 permit ip host 10.3.3.3 host 172.16.5.34
30 permit icmp any any
50 permit tcp any host 10.3.3.3
70 permit ip host 10.4.4.4 any
90 permit ip host 172.16.2.2 host 10.3.3.12
110 permit ip host 10.3.3.3 any log
130 permit tcp host 10.3.3.3 host 10.1.2.2
150 permit ip any any

ipv4 access-list acl_1
10 permit ip host 10.3.3.3 host 172.16.5.34
20 permit icmp any any
30 permit tcp any host 10.3.3.3
40 permit ip host 10.4.4.4 any
60 permit ip host 172.16.2.2 host 10.3.3.12
70 permit ip host 10.3.3.3 any log
80 permit tcp host 10.3.3.3 host 10.1.2.2
100 permit ip any any

configure
ipv6 access-list acl_1
end
resequence ipv6 access-list acl_1 10 20

ipv4 access-list acl_1
10 permit ip host 10.3.3.3 host 172.16.5.34
30 permit icmp any any
50 permit tcp any host 10.3.3.3
70 permit ip host 10.4.4.4 any
90 Dynamic test permit ip any any
110 permit ip host 172.16.2.2 host 10.3.3.12
```

```
130 permit ip host 10.3.3.3 any log
150 permit tcp host 10.3.3.3 host 10.1.2.2
170 permit ip host 10.3.3.3 any
190 permit ip any any
```

シーケンス番号を指定したエントリの追加：例

次の例では、新しいエントリを IPv4 アクセス リスト acl_5 に追加しています。

```
ipv4 access-list acl_5
 2 permit ipv4 host 10.4.4.2 any
 5 permit ipv4 host 10.0.0.44 any
 10 permit ipv4 host 10.0.0.1 any
 20 permit ipv4 host 10.0.0.2 any
configure
ipv4 access-list acl_5
 15 permit 10.5.5.5 0.0.0.255
end
ipv4 access-list acl_5
 2 permit ipv4 host 10.4.4.2 any
 5 permit ipv4 host 10.0.0.44 any
 10 permit ipv4 host 10.0.0.1 any
 15 permit ipv4 10.5.5.5 0.0.0.255 any
 20 permit ipv4 host 10.0.0.2 any
```

シーケンス番号を指定しないエントリの追加：例

次に、シーケンス番号が指定されていないエントリをアクセス リストの末尾に追加する方法を示します。 シーケンス番号のないエントリを追加すると、自動的にシーケンス番号が割り当てられ、アクセス リストの末尾に配置されます。 デフォルトの増分値は 10 であるため、エントリには、既存のアクセス リストの最後のエントリのシーケンス番号に 10 を加えたシーケンス番号が割り当てられます。

```
configure
ipv4 access-list acl_10
permit 10
.1.1.1 0.0.0.255
permit 10
.2.2.2 0.0.0.255
permit 10
.3.3.3 0.0.0.255
end

ipv4 access-list acl_10
 10 permit ip 10
.1.1.0 0.0.0.255 any
 20 permit ip 10
.2.2.0 0.0.0.255 any
 30 permit ip 10
.3.3.0 0.0.0.255 any

configure
ipv4 access-list acl_10
permit 10
.4.4.4 0.0.0.255
end

ipv4 access-list acl_10
 10 permit ip 10
.1.1.0 0.0.0.255 any
 20 permit ip 10
.2.2.0 0.0.0.255 any
 30 permit ip 10
```

```
.3.3.0 0.0.0.255 any
 40 permit ip 10
.4.4.0 0.0.0.255 any
```

その他の参考資料

ここでは、アクセス リストおよびプレフィックス リストの実装に関連する資料を示します。

関連資料

関連項目	参照先
アクセス リスト コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*Access List Commands*」の章
プレフィックス リスト コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*Prefix List Commands*」の章
端末サービス コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『*Cisco ASR 9000 Series Aggregation Services Router System Management Command Reference*』の「*Terminal Services Commands*」の章

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
—	MIB を特定してダウンロードするには、次の URL にある Cisco MIB Locator を使用し、[Cisco Access Products] メニューからプラットフォームを選択します。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

第 2 章

ARP の設定

アドレス解決は、ネットワーク アドレスをメディア アクセス コントロール（MAC）アドレスにマッピングするプロセスです。 このプロセスを実現するのに使用されるのが、アドレス解決プロトコル（ARP）です。 この章では、Cisco ASR 9000 シリーズのアグリゲーション サービス ルータに ARP プロセスを設定する方法について説明します。

（注） この章に記載されている ARP コマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』を参照してください。この章に記載されている他のコマンドのドキュメントについては、コマンド リファレンスのマスター インデックスを使用するか、またはオンラインで検索してください。

ARP 設定の機能履歴

リリース	変更内容
リリース 3.7.2	この機能が導入されました。

ARP 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。

ARP の設定に関する制約事項

ARP の設定には、次の制約事項が適用されます。

- 逆アドレス解決プロトコル（RARP）はサポートされません。
- ARP スロットリングはサポートされません。

（注） ARP スロットリングとは、転送情報ベース（FIB）で ARP パケットのレートを制限するものです。

Cisco ASR 9000 シリーズ ルータに Direct Attached Gateway Redundancy（DAGR）機能を設定するときには、次の制約事項も適用されます。

- IPv6 はサポートされていません。
- イーサネット バンドルはサポートされません。
- Non-Ethernet インターフェイスはサポートされていません。
- 無中断 ARP プロセス再起動はサポートされません。
- 無中断 RSP フェールオーバーはサポートされません。

ARP の設定に関する情報

ARP 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。

IP アドレッシングの概要

IP のデバイスは、ローカル アドレス（ローカル セグメントまたは LAN のデバイスを一意に識別）とネットワーク アドレス（デバイスが属するネットワークを識別）の両方を持つことができます。 ローカル アドレスは、より正確にはデータリンク アドレスとして知られています。その理由は、ローカル アドレスはパケット ヘッダーのデータリンク層（OSI モデルの第 2 層）の部分にあり、データリンク デバイス（ブリッジやすべてのデバイス インターフェイスなど）によって読み取られるからです。 データリンク層内の MAC 副層がその層用にアドレスを処理するため、技術志向が強い人ほどローカル アドレスを *MAC* アドレスと呼びます。

たとえば、イーサネットでデバイスと通信するには、Cisco IOS XR ソフトウェアがまずそのデバイスの 48 ビットの MAC アドレスまたはローカル データリンク アドレスを特定する必要があり

ます。 IP アドレスからローカル データリンク アドレスを決定する処理は、アドレス解決と呼ばれています。

単一の LAN でのアドレス解決

次のプロセスでは、送信元デバイスと宛先デバイスが同じLANに接続されている場合のアドレス解決について説明します。

1. エンド システム A は、エンド システム B の MAC アドレスを学習しようとして、ARP 要求を LAN にブロードキャストします。
2. ブロードキャストは、エンド システム B を含め LAN 上のすべてのデバイスで受信され、処理されます。
3. エンド システム B のみが、ARP 要求に応答します。 ARP 応答に自身の MAC アドレスを含めてエンド システム A に送信します。
4. エンド システム A は、応答を受信し、自身の ARP キャッシュにエンド システム B の MAC アドレスを保存します （ARP キャッシュ内で、ネットワーク アドレスが MAC アドレスに関連付けられます）。
5. エンドシステムAはエンドシステムBとの通信が必要になるたびに、ARPキャッシュをチェックし、エンド システム B の MAC アドレスを探し、フレームを直接送信します。最初に ARP 要求を使用する必要はありません。

ルータによって相互接続されている場合のアドレス解決

次のプロセスでは、送信元デバイスと宛先デバイスが、ルータによって相互接続された異なる LAN に接続されている場合のアドレス解決について説明します（プロキシ ARP が有効になっている場合のみ）。

1. エンド システム Y は、エンド システム Z の MAC アドレスを学習しようとして、ARP 要求を LAN にブロードキャストします。
2. ブロードキャストは、ルータXを含めLAN上のすべてのデバイスで受信され、処理されます。
3. ルータ X は、自身のルーティング テーブルをチェックし、エンド システム Z が別の LAN にあることを認識します。
4. このため、ルータ X はエンド システム Z のプロキシとして機能します。 自身がエンド システム Z に属しているかのように、エンド システム Y からの ARP 要求に応答し、ARP 応答に自身の MAC アドレスを含めて送信します。
5. エンド システム Y は、ARP 応答を受信し、自身の ARP キャッシュにあるエンド システム Z のエントリにルータ X の MAC アドレスを保存します。
6. エンド システム Y はエンド システム Z との通信が必要になると、ARP キャッシュをチェックし、ルータ X の MAC アドレスを探し、フレームを直接送信します。ARP 要求を使用する必要はありません。

7 ルータ X は、エンド システム Y からのトラフィックを受信して、それを他の LAN 上にあるエンド システム Z に転送します。

ARP およびプロキシ ARP

Cisco IOS XR ソフトウェアでは、2 つの形式のアドレス解決がサポートされています。アドレス解決プロトコル（ARP）とプロキシ ARP で、それぞれ RFC 826 と RFC 1027 で定義されています。

ARP は、IP アドレスをメディアや MAC アドレスに関連付けるために使用されます。 ARP は IP アドレスを入力とし、関連するメディアのアドレスを決定します。 メディアまたは MAC アドレスが決定すると、IP アドレスまたはメディア アドレスの関連付けは、すぐ取得できるように ARP のキャッシュに保管されます。 その後、IP データグラムがリンク層フレームにカプセル化され、ネットワークを通じて送信されます。

プロキシ ARP がディセーブルされると、ネットワーキング デバイスは、次のいずれかの条件が満たされる場合に限り、インターフェイスに受信された ARP 要求に応答します。

- ARP 要求のターゲット IP アドレスは、要求が受信されたインターフェイス IP アドレスと同じです。
- ARP 要求のターゲット IP アドレスには、静的に設定された ARP エイリアスがあります。

プロキシ ARP がイネーブルになると、ネットワーキング デバイスは、次の条件すべてを満たす ARP 要求にも応答します。

- ターゲット IP アドレスが、要求を受信した同一の物理ネットワーク（LAN）上にない。
- ネットワーキング デバイスに、ターゲット IP アドレスまでのルートが 1 つ以上存在する。
- ターゲット IP アドレスまでのルートすべてが、要求を受信したインターフェイスとは別のインターフェイスを通過する。

ARP キャッシュ エントリ

ARP は、ネットワーク アドレス（IP アドレスなど）とイーサネット ハードウェア アドレスの間の通信を確立します。 各通信の記録は、キャッシュ内に事前定義された期間だけ保持された後、廃棄されます。

また、明示的に削除するまで保持されるスタティック（永続）エントリを ARP キャッシュに追加することもできます。

Direct Attached Gateway Redundancy

Direct Attached Gateway Redundancy（DAGR）により、接続済みのデバイス上でサードパーティ冗長性スキームが機能して Gratuitous ARP をフェールオーバー シグナルとして使用できるようになります。これにより、ARP プロセスはルーティング情報ベース（RIB）に新しいタイプのルート

をアドバタイズできます。 このようなルートは、Open Shortest Path First（OSPF）によって配布されます。

IP ネットワークの部分によっては、ルーティング プロトコルのない冗長性が必要になることがあります。 典型的な例が、モバイル環境で見られます。モバイル環境では、ベース ステーション コントローラやマルチメディア ゲートウェイなどのデバイスをペアで導入して冗長性を確保し、アグレッシブ フェールオーバー要件（サブセカンド以下）を満たしています。しかし、これらのデバイスには一般に、OSPF や Intermediate System-to-Intermediate System（IS-IS）プロトコルなどのネイティブのレイヤ 3 プロトコルを使用して、この冗長性を管理する機能がありません。 その代わりに、イーサネット スイッチ経由で隣接の IP デバイスに接続されるものと認識し、仮想ルータ冗長プロトコル（VRRP）によく似た独自のメカニズムを使用してレイヤ 2 で冗長性を管理します。 このためには復元力を持つイーサネット スイッチング機能が必要であり、実現できるかどうかは MAC ラーニングや MAC フラッディングなどのメカニズムに左右されます。

DAGR は、このようなデバイスの多くがイーサネット スイッチを介さずに直接 Cisco ASR 9000 シリーズ ルータに接続できるようにする機能です。 DAGR を使用すると、レイヤ 3 ソリューションを使用して、サブセカンド フェールオーバー要件を満たすことができます。 MAC ラーニングもフラッディングもスイッチングも必要ありません。

（注） モバイル デバイスの 1 対 1 レイヤ 2 冗長性メカニズムは独自のメカニズムであるため、必ずしも標準に準拠しているとは限りません。 IP モバイル機器のほとんどは DAGR と互換性がありますが、DAGR とのインターフェイスとなるレイヤ 2 メカニズムが独自のものである場合があるため、相互運用性を確保するには資格が必要になります。

その他のガイドライン

次に、DAGR を設定するときに考慮すべき追加のガイドラインを示します。

- システムごとに最大 40 組の DAGR ピアがサポートされます。各ピアは、同じインターフェイスでも異なるインターフェイスでもかまいません。
- DAGR ルートでは、ARP 応答パケットを受け取ってから 500 ミリ秒以内でフェールオーバーが実施されます。
- ARP プロセスの再開時に、DAGR グループが再初期化されます。

ARP の設定方法

ここでは、次のタスクの手順を示します。

スタティック ARP キャッシュ エントリの定義

ARPをはじめとするアドレス解決プロトコルを使用すると、IPアドレスとメディアアドレスとをダイナミックにマッピングできます。ホストのほとんどがダイナミックアドレス解決をサポートしているため、一般にスタティック ARP キャッシュ エントリを指定する必要はありません。 スタティックARPキャッシュエントリを定義する必要がある場合は、グローバルに定義できます。このタスクを実行すると、ARP キャッシュにエントリが永続的にインストールされます。Cisco IOS XR ソフトウェア は、このエントリを使用して、32 ビットの IP アドレスを 48 ビットのハードウェア アドレスに変換します。

また、ARP キャッシュにエイリアス エントリを作成して、指定された IP アドレスの所有者であるかのように ARP 要求に応答することもできます。

手順の概要

1. configure
2. 次のいずれかを実行します。
   - **arp** [**vrf** *vrf-name*] *ip-address hardware-address encapsulation-type*
   - **arp** [**vrf** *vrf-name*] *ip-address hardware-address encapsulation-type* **alias**
3. 次のいずれかを実行します。
   - end
   - 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	次のいずれかを実行します。 • **arp** [**vrf** *vrf-name*] *ip-address hardware-address encapsulation-type* • **arp** [**vrf** *vrf-name*] *ip-address hardware-address encapsulation-type* **alias**	指定された 32 ビットの IP アドレスを指定された 48 ビットのハードウェア アドレスに関連付けるスタティック ARP キャッシュ エントリを作成します。 （注） **alias** エントリを作成すると、エントリが対応付けられたインターフェイスは、指定されたアドレスの所有者であるかのように機能します。つまり、エントリ内のデータリンク層アドレスを持つネットワーク層アドレスに代わって ARP 要求パケットに応答します。

	コマンドまたはアクション	目的

	例：
	`RP/0/RSP0/CPU0:router(config)# arp 192.168.7.19 0800.0900.1834 arpa`
	または
	`RP/0/RSP0/CPU0:router(config)# arp 192.168.7.19 0800.0900.1834 arpa alias`	

ステップ 3	次のいずれかを実行します。
	• end
	• 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。

プロキシ ARP のイネーブル化

Cisco IOS XR ソフトウェア （RFC 1027 で定義されている）プロキシ ARP を使用して、ルーティングに必要な情報を持たないホストでも他のネットワークやサブネット上のホストのメディア アドレスを判別できるようにします。たとえば、ARP 要求の送信元と異なるインターフェイス上のホストに宛てた ARP 要求をルータが受信した場合、そのルータに他のインターフェイスを経由してそのホストに至るすべてのルートが格納されていれば、ルータは自身のローカルデータ リンク

アドレスを示すプロキシ ARP 応答パケットを生成します。 ARP 要求を送信したホストはルータにパケットを送信し、ルータはパケットを目的のホストに転送します。プロキシARPはデフォルトではディセーブルになっています。このタスクでは、ディセーブルになっているプロキシ ARP 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

1. configure
2. **interface** *type number*
3. proxy-arp
4. 次のいずれかを実行します。
   - end
   - 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** *type number* 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 モードを開始します。
ステップ 3	proxy-arp 例： `RP/0/RSP0/CPU0:router(config-if)# proxy-arp`	インターフェイス上でプロキシ ARP をイネーブルにします。

ステップ 4	次のいずれかを実行します。
	• end
	• 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

	コマンドまたはアクション	目的
		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。

DAGR の設定

次の手順に従って、Cisco ASR 9000 シリーズ ルータに DAGR グループを作成します。

手順の概要

1. configure
2. **interface** *type interface-path-id*
3. arp dagr
4. **peer ipv4** *address*
5. **route distance normal** *normal- distance* **priority** *priority-distance*
6. **route metric normal** *normal- metric* **priority** *priority-metric*
7. **timers query** *query-time* **standby** *standby-time*
8. **priority-timeout** *time*
9. 次のいずれかを実行します。
   - end
   - commit
10. **show arp dagr** [*interface* [*IP-address*]]

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 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、インターフェイスを設定します。
ステップ 3	arp dagr 例： `RP/0/RSP0/CPU0:router(config-if)# arp dagr`	DAGR 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	**peer ipv4** *address* 例： `RP/0/RSP0/CPU0:router(config-if-dagr)# peer ipv4 10.0.0.100`	仮想 IP アドレス用に DAGR グループを新規に作成します。
ステップ 5	**route distance normal** *normal- distance* **priority** *priority-distance* 例： `RP/0/RSP0/CPU0:router(config-if-dagr-peer)# route distance normal 140 priority 3`	（任意）DAGR グループのルート ディスタンスを設定します。
ステップ 6	**route metric normal** *normal- metric* **priority** *priority-metric* 例： `RP/0/RSP0/CPU0:router(config-if-dagr-peer)# route metric normal 84 priority 80`	（任意）DAGR グループのルート メトリックを設定します。
ステップ 7	**timers query** *query-time* **standby** *standby-time* 例： `RP/0/RSP0/CPU0:router(config-if-dagr-peer)# timers query 2 standby 19`	（任意）仮想 IP アドレスに向けて ARP 要求を連続して送信する場合の各要求間の間隔を秒単位で設定します。

	コマンドまたはアクション	目的
ステップ 8	**priority-timeout** *time* 例： `RP/0/RSP0/CPU0:router(config-if-dagr-peer)# priority-timeout 25`	（任意）高優先順位の DAGR ルートから通常の優先順位に戻るまで待機する時間の長さを秒単位で計測するタイマーを設定します。

ステップ 9	次のいずれかを実行します。
	• end
	• commit
	
	例：
	`RP/0/RSP0/CPU0:router(config-if-dagr)# end`
	または
	`RP/0/RSP0/CPU0:router(config-if-dagr)# 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 10	**show arp dagr** [*interface* [*IP-address*]] 例： `RP/0/RSP0/CPU0:router# show arp dagr`	（任意）すべての DAGR グループの動作状態を表示します。 任意の *interface* 引数と *IP-address* 引数を使用すると、特定のインターフェイスまたは仮想 IP アドレスへの出力を制限できます。

第 3 章

シスコ エクスプレス フォワーディングの実装

シスコ エクスプレス フォワーディング（CEF）は、拡張レイヤ 3 IP スイッチング テクノロジーです。CEF によって、インターネットや、Web ベースのアプリケーションまたは対話型セッションが集中的に使用されるネットワークなどの、大規模でダイナミックなトラフィック パターンを持つネットワークのパフォーマンスおよびスケーラビリティが最適化されます。

この章では、Cisco ASR 9000 シリーズ アグリゲーション サービス ルータに CEF を実装するのに必要なタスクについて説明します。

（注） この章に記載されている CEF コマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』を参照してください。 設定タスクの実行中に示される他のコマンドのドキュメントについては、オンラインでマスター コマンド インデックスを検索してください。

CEF の実装の機能履歴

リリース	変更内容
リリース 3.7.2	この機能が導入されました。

シスコエクスプレスフォワーディングの実装の前提条件

シスコエクスプレスフォワーディングを実装するには、次の前提条件を満たす必要があります。

- 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。

シスコ エクスプレス フォワーディング ソフトウェアの実装に関する情報

このドキュメントで示すシスコエクスプレスフォワーディング機能を実装するには、次の概念を理解する必要があります。

シスコ エクスプレス フォワーディング実装でサポートされている主要な機能

Cisco IOS XR ソフトウェア上の CEF では、次の機能をサポートしています。

- ボーダー ゲートウェイ プロトコル（BGP）ポリシー アカウンティング
- リバース パス転送（RPF）
- 仮想インターフェイス サポート
- マルチパス サポート
- ルート整合性
- パッケージング、再起動性、リソース不足（OOR）処理などのハイ アベイラビリティ機能
- OSPFv2 SPF プレフィックス優先順位付け
- BGP 属性ダウンロード

CEF の利点

CEF には、次の利点があります。

- パフォーマンス向上：CEF は、高速スイッチング ルート キャッシングよりも CPU を消費しません。 より多くの CPU 処理能力を Quality of Service（QoS）や暗号化などのレイヤ 3 サービスに向けることができます。

- スケーラビリティ：CEF では、各モジュラ サービス カード（MSC）でスイッチング能力を最大限に活用できます。
- 復元力：CEF では、大規模な動的ネットワーク上で比類ないレベルのスイッチング一貫性と安定性を実現します。動的ネットワークでは、ルーティング変更のために、高速にスイッチングされるキャッシュ エントリが頻繁に無効化されます。 ルーティング変更により、ルート キャッシュを使用した高速スイッチングではなく、ルーティング テーブルを使用したトラフィックのプロセス スイッチングが行われることがあります。 転送情報ベース（FIB）ルックアップ テーブルにはルーティング テーブルに存在する既知のルートがすべて含まれているため、ルート キャッシュのメンテナンスが不要になるほか、高速スイッチングまたはプロセス スイッチング フォワーディングのシナリオも必要ありません。 CEF では、一般的なデマンド キャッシング スキームよりも効率よくトラフィックを切り替えることができます。

CEF コンポーネント

Cisco IOS XR ソフトウェアCEF は、2 つの別個のコンポーネントとともに常に CEF モードで動作します。転送情報ベース（FIB）データベースと、隣接関係テーブル、つまりプロトコル独立型の Adjacency Information Base（AIB）です。

CEF は、Cisco IOS XR ソフトウェアにとって主要な IP パケット転送データベースです。 CEF の役割は次の機能を果たすことです。

- ソフトウェア スイッチング パス
- ソフトウェアおよびハードウェア転送エンジンの転送テーブルおよび隣接関係テーブルのメンテナンス（AIB によるメンテナンス）

Cisco IOS XR ソフトウェアでは、次の CEF 転送テーブルがメンテナンスされます。

- IPv4 CEF データベース
- IPv6 CEF データベース
- MPLS LFD データベース
- マルチキャスト転送テーブル（MFD）

プロトコル独立型の FIB プロセスが、Route Switch Processor（RSP）の IPv4 および IPv6 ユニキャスト用の転送テーブルと、各 MSC 用の転送テーブルをメンテナンスします。

各ノード上の FIB が、ルーティング情報ベース（RIB）を更新し、ルート解決を実行し、RSP および各 MSC の FIB テーブルを個別にメンテナンスします。 各ノード上の FIB テーブルに格納されている情報は、テーブルによって若干異なることがあります。 隣接 FIB エントリがメンテナンスされるのはローカル ノードに限られるため、FIB エントリにリンクされている隣接エントリが異なるものになることがあります。

ボーダー ゲートウェイ プロトコルのポリシー アカウンティング

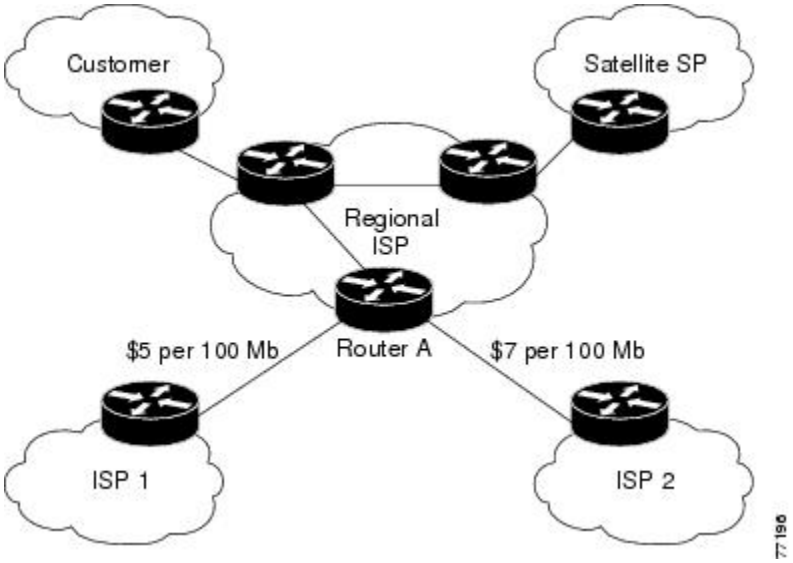
ボーダー ゲートウェイ プロトコル（BGP）ポリシーアカウンティングは、異なるピア間で送受信される IP トラフィックを測定および分類します。 ポリシー アカウンティングは、入力インターフェイスまたは出力インターフェイス単位でイネーブル化されます。また、IP トラフィックを識別するために、コミュニティ リスト、自律システム番号、自律システム パスなどのパラメータに基づいてカウンタが割り当てられます。

（注） ルート ポリシーには、2 つのタイプがあります。 1 つは通常の BGP ルート ポリシーで、BGP リンクに対してアドバタイズされる BGP ルートをフィルタリングするために使用します。 このタイプのルート ポリシーは、特定の BGP ネイバーに適用します。 もう 1 つは特殊なルート ポリシーで、BGP プレフィックスのトラフィック インデックスをセットアップするために使用します。 RIB テーブルに BGP ルートを挿入するときに、このルート ポリシーをグローバル BGP IPv4 アドレス ファミリに適用すると、トラフィック インデックスをセットアップできます。 BGP ポリシーアカウンティングでは、2 つ目のタイプのルート ポリシーが使用されます。

BGP ポリシーアカウンティングを使用して、通過するルートに基づいてトラフィックのアカウンティングを行うことができます。サービスプロバイダーは、すべてのトラフィックをカスタマー別に識別してアカウンティングを実施し、それに応じて課金できます。図 1 : BGP ポリシー アカウンティングのトポロジ例, （69 ページ） では、BGP ポリシー アカウンティングはルータ A で実装され、自律システム バケットにおけるパケットおよびバイト ボリュームを測定します。 カスタマーは、国内、海外、または衛星経由の送信元からルーティングされたトラフィックに応じて適切に課金されます。

（注） BGP ポリシー アカウンティングは、BGP プレフィックスに限って IP トラフィックを測定し、分類します。

図 *1*：*BGP* ポリシー アカウンティングのトポロジ例



BGP ポリシーアカウンティングは、指定されたルート ポリシーに基づいて、インターフェイスに関連付けられたトラフィック インデックス（バケット）を各プレフィックスに割り当てます。BGP プレフィックスは、トラフィック インデックスとともに RIB から FIB にダウンロードされます。

BGP プレフィックスに割り当てることができるトラフィック インデックス（バケット番号）が全部で 63 個（1 ～ 63）あります。 システム内部ではトラフィック インデックスにアカウンティング テーブルが関連付けられており、このテーブルは入力インターフェイスおよび出力インターフェイスごとに作成されます。 トラフィック インデックスを使用すると、送信元 IP アドレスまたは宛先 IP アドレス、あるいはその両方が BGP プレフィックスである場合に、IP トラフィックのアカウンティングを行うことができます。

（注） トラフィック インデックス 0 には、Interior Gateway Protocol（IGP）ルートを使用して、パケット数が含められます。

リバース パス転送（ストリクトとルーズ）

ユニキャスト IPv4 および IPv6 リバース パス転送（uRPF）は、ストリクトとルーズのどちらのモードでも、検証可能な IP 送信元アドレスを欠いている IP パケットを廃棄することにより、不

正な形式の IP 送信元アドレスまたはスプーフィングされた IP 送信元アドレスがネットワークに導入された場合にもたらされる問題を軽減します。 ユニキャスト RPF は、CEF テーブルの逆引きを行うことでこれを確認します。 このため、ユニキャスト リバース パス転送が可能になるのは、ルータで CEF がイネーブルになっている場合だけです。

IPv6 uRPF をサポートしているのは、ASR 9000-SIP-700 LC、ASR 9000 Ethernet LC、および ASR 9000 Enhanced Ethernet LC です。

（注） ユニキャスト RPF は、ブートストラップ プロトコルおよびダイナミック ホスト コンフィギュレーション プロトコル（DHCP）機能が正しく動作するように、送信元アドレスが 0.0.0.0 で宛先アドレスが 255.255.255.255 のパケットの通過を許可します。

ストリクト uRPF がイネーブルになっていると、FIB でそのパケットの送信元アドレスがチェックされます。 パケットを受信したインターフェイスが、トラフィックをパケットの送信元に転送するのに使用されたのと同じインターフェイスである場合、パケットはチェックを通過し、パケットに対してさらに処理が実施されます。それ以外の場合、パケットはドロップされます。 ストリクト uRPF を適用するのは、自然の対称性または設定された対称性がある場合だけにしてください。 内部インターフェイスによってはルーティングが非対称になってパケットの送信元へのルートが複数存在することがあるため、ネットワーク内部にあるインターフェイスにはストリクト uRPF を実装しないでください。

（注） ストリクト RPF の動作は、プラットフォーム、再帰レベルの数、および等コスト マルチパス（ECMP）シナリオに含まれるパスの数によって若干異なります。 ストリクト RPF が設定されている場合でも、プラットフォームによってはプレフィックスの一部または全部に対してルーズ RPF チェックに切り替わることがあります。

ルーズ uRPF がイネーブルになっていると、FIB でそのパケットの送信元アドレスがチェックされます。 送信元アドレスが存在し、有効な転送エントリに一致する場合、パケットはチェックを通過し、パケットに対してさらに処理が実施されます。それ以外の場合、パケットはドロップされます。

ストリクト モードの uRPF では、プレフィックスの uRPF インターフェイス リストをメンテナンスする必要があります。 リストには、ストリクト モードの uRPF が設定されたインターフェイスで、かつプレフィックス パスが指すインターフェイスのみが含まれています。 uRPF インターフェイス リストは、可能な限りプレフィックス間で共有されます。 このリストのサイズは、ASR 9000 イーサネット ラインカードでは 12、統合 20G SIP カードでは 64 です。 リストがサポートされている最大値を超えると、uRPF がストリクト モードからルーズ モードにフォールバックします。

ルーズおよびストリクトの uRPF は、2 つのオプションをサポートしています。**allow self-ping** と **allow default** です。 **self-ping** オプションでは、パケットの送信元が自身に ping を実行できます。 **allow default** オプションでは、デフォルトのルーティング エントリに合わせてルックアップ結果を生成できます。 uRPF がストリクト モードで、**allow default** オプションがイネーブルになっているときには、パケットがデフォルトのインターフェイス経由で届いた場合にのみ、パケットに対してさらに処理が実施されます。

BGP 属性ダウンロード

BGP 属性ダウンロード機能を使用すると、CEF にインストールした BGP 属性を表示できます。CEF にインストールした BGP 属性を表示するには、**show cef bgp-attribute** コマンドを設定します。 **show cef bgp-attribute attribute-id** コマンドおよび **show cef bgp-attribute local-attribute-id** コマンドを使用すると、特定の BGP 属性を属性 ID およびローカル属性 ID 別に参照できます。

CEF の実装方法

ここでは、次のタスクの手順を示します。

CEF の確認

このタスクを実行すると、CEF を検証できます。

手順の概要

1. **show cef** {**ipv4** | **ipv6**}
2. **show cef** {**ipv4** | **ipv6**} **summary**
3. **show cef** {**ipv4** | **ipv6**} **detail**
4. show adjacency detail

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 cef** {**ipv4** | **ipv6**} 例： `RP/0/RSP0/CPU0:router# show cef ipv4`	IPv4 または IPv6 CEF テーブルを表示します。 ネクストホップおよび転送インターフェイスがプレフィックスごとに表示されます。 （注） **show cef** コマンドの出力は、場所によって異なります。
ステップ 2	**show cef** {**ipv4** | **ipv6**} **summary** 例： `RP/0/RSP0/CPU0:router# show cef ipv4 summary`	IPv4 または IPv6 CEF テーブルのサマリーを表示します。
ステップ 3	**show cef** {**ipv4** | **ipv6**} **detail** 例： `RP/0/RSP0/CPU0:router# show cef ipv4 detail`	IPv4 または IPv6 CEF テーブルの詳細な情報を表示します。

	コマンドまたはアクション	目的
ステップ 4	show adjacency detail 例： `RP/0/RSP0/CPU0:router# show adjacency detail`	インターフェイスごとのレイヤ 2 情報など詳細な隣接情報を表示します。 （注） **show adjacency** コマンドの出力は、場所によって異なります。

BGP ポリシー アカウンティングの設定

このタスクを実行すると、BGP ポリシー アカウンティングを設定できます。

（注） ルート ポリシーには、2 つのタイプがあります。 BGP ポリシー アカウンティングでは、BGP プレフィックスのトラフィック インデックスをセットアップするために使用するタイプを使用します。 RIB テーブルに BGP ルートを挿入するときに、このルート ポリシーをグローバル BGP IPv4 アドレス ファミリに適用すると、トラフィック インデックスをセットアップできます。

BGP ポリシー アカウンティングでは、送信元 IP アドレス（BGP プレフィックス）および宛先 IP アドレス（BGP プレフィックス）に割り当てられたトラフィック インデックスに基づいて、入力および出力 IP トラフィックのアカウンティングをインターフェイス単位で行うことができます。 Routing Policy Language（RPL）を使用して、次のパラメータに従って BGP プレフィックスのトラフィック インデックスを割り当てることができます。

- prefix-set
- AS-path-set
- community-set

（注） BGP ポリシー アカウンティングは、IPv4 プレフィックスでのみサポートされます。

2 つの設定タスクを実行すると、prefix-set、AS-path-set、または community-set パラメータに従って、RIB の BGP プレフィックスを分類できます。

1. prefix-set、AS-path-set、または community-set に基づいてトラフィック インデックスのセットアップに関するポリシーを定義するには、**route-policy** コマンドを使用します。
2. 定義済みのルート ポリシーをグローバル BGP IPv4 ユニキャスト アドレス ファミリに適用するには、BGP **table-policy** コマンドを使用します。

route-policy コマンドおよび **table-policy** コマンドについては、『*Cisco ASR 9000 Series Aggregation Services Router Routing Command Reference*』を参照してください。

各インターフェイスでBGPポリシーアカウンティングをイネーブルにするには、次のオプションを使用します。

- ipv4 bgp policy accounting コマンドに次のいずれかのキーワード オプションを指定します。
  - input source-accounting
  - input destination-accounting
  - input source-accounting destination-accounting
- ipv4 bgp policy accounting コマンドに次のいずれかのキーワード オプションを指定します。
  - output source-accounting
  - output destination-accounting
  - output source-accounting destination-accounting
- **ipv4 bgp policy accounting** コマンドに用意されているキーワードを任意に組み合わせて使用します。

はじめる前に

BGP ポリシー アカウンティング機能を使用するには、ルータで BGP をイネーブルにする必要があります（デフォルトでは CEF がイネーブルになっています）。 BGP をイネーブルにする方法については、『*Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*』を参照してください。

手順の概要

1. configure
2. as-path-set
3. exit
4. **prefix-set** *name*
5. exit
6. **route-policy** *policy-name*
7. end
8. configure
9. **router bgp** *autonomous-system-number*
10. **address-family ipv4** {**unicast** | **multicast** }
11. **table policy** *policy-name*
12. end
13. configure
14. **interface** *type interface-path-id*
15. **ipv4 bgp policy accounting** {**input** | **output** {**destination-accounting** [**source-accounting**] | **source-accounting** [**destination-accounting**]}}
16. 次のいずれかを実行します。
    - end
    - 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	as-path-set
	
	例：
	
	`RP/0/RSP0/CPU0:router(config)# as-path-set as107`
	`RP/0/RSP0/CPU0:router(config-as)# ios-regex '107$'`
	`RP/0/RSP0/CPU0:router(config-as)# end-set`
	
	`RP/0/RSP0/CPU0:router(config)# as-path-set as108`
	`RP/0/RSP0/CPU0:router(config-as)# ios-regex '108$'`
	`RP/0/RSP0/CPU0:router(config-as)# end-set`
		ポリシー 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	exit 例： `RP/0/RSP0/CPU0:router(config-as)# exit`	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 4	**prefix-set** *name* 例： `RP/0/RSP0/CPU0:router(config)# prefix-set RT-65`	プレフィックス リストを定義します。
ステップ 5	exit 例： `RP/0/RSP0/CPU0:router(config-pfx)# exit`	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 6	**route-policy** *policy-name* 例： `RP/0/RSP0/CPU0:router(config)# route-policy rp501b`	ルート ポリシー名を指定します。
ステップ 7	end 例： `RP/0/RSP0/CPU0:router(config-rpl)# end`	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。

	コマンドまたはアクション	目的
ステップ 8	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 9	**router bgp** *autonomous-system-number* 例： `RP/0/RSP0/CPU0:router(config)# router bgp 1`	BGP ルーティング プロセスを設定できます。
ステップ 10	**address-family ipv4** {**unicast** | **multicast** } 例： `RP/0/RSP0/CPU0:router(config-bgp)# address-family ipv4 unicast`	BGP ルーティング セッションの設定中に、アドレス ファミリ コンフィギュレーション モードを開始できます。
ステップ 11	**table policy** *policy-name* 例： `RP/0/RSP0/CPU0:router(config-bgp-af)# table-policy set-traffic-index`	ルーティング テーブルにインストールされるルートにルーティング ポリシーを適用します。
ステップ 12	end 例： `RP/0/RSP0/CPU0:router(config-bgp-af)# end`	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。

	コマンドまたはアクション	目的
ステップ 13	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 14	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface TenGigE0/1/0/2`	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 15	**ipv4 bgp policy accounting** {**input** | **output** {**destination-accounting** [**source-accounting**] | **source-accounting** [**destination-accounting**]}} 例： `RP/0/RSP0/CPU0:router(config-if)# ipv4 bgp policy accounting output destination-accounting`	BGP ポリシー アカウンティングをイネーブルにします。

ステップ 16	次のいずれかを実行します。
	• end
	• 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

BGP ポリシー アカウンティングの確認

このタスクを実行すると、BGP ポリシー アカウンティングを検証できます。

（注） BGP ポリシー アカウンティングは、IPv4 プレフィックスでサポートされます。

はじめる前に

BGP ポリシー アカウンティングを設定する必要があります。BGP ポリシー アカウンティングの設定, （72 ページ）を参照してください。

手順の概要

1. **show route bgp**
2. **show bgp summary**
3. **show bgp** *ip-address*
4. **show route ipv4** *ip-address*
5. **show cef ipv4** *prefix*
6. **show cef ipv4** *prefix* **detail**
7. **show cef ipv4 interface** *type interface-path-id* **bgp-policy-statistics**

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 route bgp** 例： `RP/0/RSP0/CPU0:router# show route bgp`	トラフィック インデックスがある BGP ルートをすべて表示します。
ステップ 2	**show bgp summary** 例： `RP/0/RSP0/CPU0:router# show bgp summary`	すべての BGP ネイバーの状況を表示します。

	コマンドまたはアクション	目的
ステップ 3	**show bgp** *ip-address* 例： `RP/0/RSP0/CPU0:router# show bgp 40.1.1.1`	BGP 属性がある BGP プレフィックスを表示します。
ステップ 4	**show route ipv4** *ip-address* 例： `RP/0/RSP0/CPU0:router# show route ipv4 40.1.1.1`	RIB にトラフィック インデックスがある特定の BGP ルートを表示します。
ステップ 5	**show cef ipv4** *prefix* 例： `RP/0/RSP0/CPU0:router# show cef ipv4 40.1.1.1`	RP FIB にトラフィック インデックスがある特定の BGP プレフィックスを表示します。
ステップ 6	**show cef ipv4** *prefix* **detail** 例： `RP/0/RSP0/CPU0:router# show cef ipv4 40.1.1.1 detail`	RP FIB に詳細な情報がある特定の BGP プレフィックスを表示します。
ステップ 7	**show cef ipv4 interface** *type interface-path-id* **bgp-policy-statistics** 例： `RP/0/RSP0/CPU0:router# show cef ipv4 interface TenGigE 0/2/0/4 bgp-policy-statistics`	特定のインターフェイスの BGP ポリシー アカウンティング統計情報を表示します。

ルート パージ遅延の設定

このタスクを実行すると、ルート パージ遅延を設定できます。パージ遅延を設定すると、RIB または関連する他のプロセスで障害が発生したときにルートがパージされるようになります。

手順の概要

1. **configure**
2. **cef purge-delay** *seconds*
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**cef purge-delay** *seconds* 例： `RP/0/RSP0/CPU0:router(config)# cef purge-delay 180`	ルーティング情報ベース（RIB）または関連する他のプロセスで障害が発生したときにルートをパージする際の遅延を設定します。

ステップ 3	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

ユニキャスト RPF チェックの設定

このタスクを実行すると、ユニキャスト リバース パス転送（uRPF）RPF チェックを設定できます。 ユニキャストRPFチェックを使用すると、不正な形式または偽装（スプーフィング）されたIP送信元アドレスがルータを通過したために発生する問題を軽減できます。 変形または偽造（ス

プーフィング）された送信元アドレスは、送信元 IP アドレスのスプーフィングに基づいたサービス拒絶（DoS）攻撃を示す場合があります。

手順の概要

1. configure
2. **interface** *type interface-path-id*
3. {**ipv4** | **ipv6**} **verify unicast source reachable-via** {**any** | **rx**} [**allow-default**] [**allow-self-ping**]
4. 次のいずれかを実行します。
   - 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** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/0`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	{**ipv4** | **ipv6**} **verify unicast source reachable-via** {**any** | **rx**} [**allow-default**] [**allow-self-ping**] 例： `RP/0/RSP0/CPU0:router(config-if)# ipv4 verify unicast source reachable-via rx`	IPv4 または IPv6 uRPF チェックをイネーブルにします。 • **rx** キーワードを指定すると、厳密なユニキャスト RPF チェックを実行できます。 ストリクト ユニキャスト RPF がイネーブルの場合、パケットは、その送信元プレフィックスがルーティング テーブルに存在し、出力インターフェイスがパケットの受信インターフェイスと一致しない限り転送されません。 • **allow-default** キーワードを指定すると、デフォルト ルートの照合を実行できます。 このオプションは、ルーズおよびストリクトの両方の RPF に適用されます。 • **allow-self-ping** キーワードを指定すると、ルータがインターフェイスに ping を実行できます。 このオプションは、ルーズおよびストリクトの両方の RPF に適用されます。
ステップ 4	次のいずれかを実行します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• 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`
		• **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。

モジュラ サービス カードとルート プロセッサ管理イーサネット インターフェイス間のスイッチングの設定

このタスクを実行すると、MSC と RP 管理イーサネット インターフェイス間のスイッチングをイネーブルにすることができます。

手順の概要

1. **configure**
2. **rp mgmtethernet forwarding**
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**rp mgmtethernet forwarding** 例： `RP/0/RSP0/CPU0:router(config)# rp mgmtethernet forwarding`	MSC からルート プロセッサ管理イーサネット インターフェイスへのスイッチングをイネーブルにします。

ステップ 3	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

BGP 属性ダウンロードの設定

このタスクを実行すると、BGP 属性ダウンロード機能を設定できます。

BGP 属性ダウンロードの設定

手順の概要

1. **configure**
2. **cef bgp attribute** {*attribute-id* | *local-attribute-id* }
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**cef bgp attribute** {*attribute-id*	*local-attribute-id* } 例： `RP/0/RSP0/CPU0:router(config)# cef bgp attribute 508`	CEF BGP 属性を設定します。

ステップ 3	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		• 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。

ルータ ソフトウェアでの CEF の実装の設定例

ここでは、次の設定例について説明します。

BGP ポリシー アカウンティングの設定：例

次に、BGP ポリシー アカウンティングを設定する例を示します。

BGP ルータ ID 用にループバック インターフェイスを設定します。

```
interface Loopback1
ipv4 address 10
.1.1.1 255.255.255.255
```

BGP ポリシー アカウンティング オプションでインターフェイスを設定します。

```
interface TenGigE0/2/0/2
 mtu 1514
 ipv4 address 10
.1.0.1 255.255.255.0
 proxy-arp
 ipv4 directed-broadcast
 ipv4 bgp policy accounting input source-accounting destination-accounting
 ipv4 bgp policy accounting output source-accounting destination-accounting
!
interface TenGigE0/2/0/2.1
 ipv4 address 10
.1.1.1 255.255.255.0
 ipv4 bgp policy accounting input source-accounting destination-accounting
 ipv4 bgp policy accounting output source-accounting destination-accounting
 dot1q vlan 1
!
interface TenGigE0/2/0/4
 mtu 1514
 ipv4 address 10
.1.0.1 255.255.255.0
 proxy-arp
 ipv4 directed-broadcast
 ipv4 bgp policy accounting input source-accounting destination-accounting
 ipv4 bgp policy accounting output source-accounting destination-accounting
!
interface TenGigE0/2/0/4.1
 ipv4 address 10
.1.2
.1 255.255.255.0
 ipv4 bgp policy accounting input source-accounting destination-accounting
 ipv4 bgp policy accounting output source-accounting destination-accounting
 dot1q vlan 1
!
interface gigabitethernet 0/0/0/4
```

```
 mtu 4474
 ipv4 address 10
.1.0.40
255.255.0.0
 ipv4 directed-broadcast
 ipv4 bgp policy accounting input source-accounting destination-accounting
 ipv4 bgp policy accounting output source-accounting destination-accounting
 encapsulation ppp
 gigabitethernet
  crc 32
 !
 keepalive disable
!
interface gigabitethernet0/0/0/8
 mtu 4474
 ipv4 address 18
.8
.0.1 255.255.0.0
 ipv4 directed-broadcast
 ipv4 bgp policy accounting input source-accounting destination-accounting
 ipv4 bgp policy accounting output source-accounting destination-accounting
 gigabitethernet
  crc 32
 !
 keepalive disable
!
```

コントローラを設定します。

```
controller gigabitethernet0/0/0/4
 ais-shut
 path
  ais-shut
 !
 threshold sf-ber 5
!
controller SONET0/0/0/8
 ais-shut
 path
  ais-shut
 !
 threshold sf-ber 5
!
```

AS-path-set および prefix-set を設定します。

```
as-path-set as107
  ios-regex '107$'
end-set

as-path-set as108
  ios-regex '108$'
end-set

prefix-set RT-65.0
  65.0.0.0/16 ge 16 le 32
end-set

prefix-set RT-66.0
  66.0.0.0/16 ge 16 le 32
end-set
```

各プレフィックス、AS-path-set、および prefix-set に基づいてトラフィック インデックスをセットアップするように、ルート ポリシー（テーブル ポリシー）を設定します。

```
route-policy bpa1

  if destination in  (10
.1.1.0/24) then
    set traffic-index 1
  elseif destination in  (10
.1.2.0/24) then
```

```
    set traffic-index 2
  elseif destination in  (10
.1.3.0/24) then
    set traffic-index 3
  elseif destination in  (10
.1.4.0/24) then
    set traffic-index 4
  elseif destination in  (10
.1.5.0/24) then
    set traffic-index 5
  endif

  if destination in  (10
.1.1.0/24) then
    set traffic-index 6
  elseif destination in  (10
.1.2.0/24) then
    set traffic-index 7
  elseif destination in  (10
.1.3.0/24) then
    set traffic-index 8
  elseif destination in  (10
.1.4.0/24) then
    set traffic-index 9
  elseif destination in  (10
.1.5.0/24) then
    set traffic-index 10
  endif

  if as-path in as107 then
       set traffic-index 7
  elseif as-path in as108 then
       set traffic-index 8
  endif

  if destination in RT-65.0 then
     set traffic-index 15
  elseif destination in RT-66.0 then
     set traffic-index 16
  endif

end-policy
```

すべての BGP ルートを通過させるか、またはドロップするように、通常の BGP ルート ポリシーを設定します。

```
route-policy drop-all
  drop
end-policy
!
route-policy pass-all
  pass
end-policy
!
```

BGP ルータを設定し、テーブル ポリシーをグローバル ipv4 アドレス ファミリに適用します。

```
router bgp 100
 bgp router-id Loopback1
 bgp graceful-restart
 bgp as-path-loopcheck
 address-family ipv4 unicast
  table-policy bpa1
  maximum-paths 8
  bgp dampening
 !
```

BGP ネイバー グループを設定します。

```
neighbor-group ebgp-peer-using-int-addr
  address-family ipv4 unicast
   policy pass-all in
```

```
   policy drop-all out
  !
 !
 neighbor-group ebgp-peer-using-int-addr-121
  remote-as 121
  address-family ipv4 unicast
   policy pass-all in
   policy drop-all out
  !
 !
 neighbor-group ebgp-peer-using-int-addr-pass-out
  address-family ipv4 unicast
   policy pass-all in
   policy pass-all out
  !
 !
```

BGP ネイバーを設定します。

```
neighbor 10
.4
.0.2
  remote-as 107
  use neighbor-group ebgp-peer-using-int-addr
 !
 neighbor 10
.8
.0.2
  remote-as 108
  use neighbor-group ebgp-peer-using-int-addr
 !
 neighbor 10
.7
.0.2
  use neighbor-group ebgp-peer-using-int-addr-121
 !
 neighbor 10
.1.7
.2
  use neighbor-group ebgp-peer-using-int-addr-121
 !
neighbor 10
.18
.0.2
  remote-as 122
  use neighbor-group ebgp-peer-using-int-addr
 !
 neighbor 10
.18
.1.2
  remote-as 1221
  use neighbor-group ebgp-peer-using-int-addr
 !
end
```

BGP ポリシー統計情報の確認：例

次に、入力インターフェイスと出力インターフェイスに各 BGP プレフィックスおよび BGP ポリシーアカウンティング統計情報用のトラフィックインデックスがセットアップされていることを確認する例を示します。 この例では、次のトラフィック ストリームを設定します。

- TenGigE0/2/0/4 から入って TenGigE0/2/0/2 の下の 5 つの VLAN サブインターフェイスに出ていくトラフィック

- GigabitEthernet 0/0/08 から入って GigabitEthernet 0/0/0/4 に出ていくトラフィック

```
show cef ipv4 interface gigabitethernet 0/0/0/8 bgp-policy-statistics

gigabitethernet0/0/0/8 is up
Input BGP policy accounting on dst IP address enabled
  buckets      packets       bytes
  7            5001160   500116000
  15          10002320  1000232000
Input BGP policy accounting on src IP address enabled
  buckets      packets       bytes
  8            5001160   500116000
  16          10002320  1000232000
Output BGP policy accounting on dst IP address enabled
  buckets      packets       bytes
  0                 15         790
Output BGP policy accounting on src IP address enabled
  buckets      packets       bytes
  0                 15         790

show cef ipv4 interface gigabitethernet 0/0/0/4 bgp-policy-statistics

gigabitethernet0/0/0/4 is up
Input BGP policy accounting on dst IP address enabled
  buckets      packets       bytes
Input BGP policy accounting on src IP address enabled
  buckets      packets       bytes
Output BGP policy accounting on dst IP address enabled
  buckets      packets       bytes
  0                 13         653
  7            5001160   500116000
  15          10002320  1000232000
Output BGP policy accounting on src IP address enabled
  buckets      packets       bytes
  0                 13         653
  8            5001160   500116000
  16          10002320  1000232000

show cef ipv4 interface TenGigE0/2/0/4 bgp-policy-statistics

TenGigE0/2/0/4 is up
Input BGP policy accounting on dst IP address enabled
  buckets      packets       bytes
  1            3297102   329710200
  2            3297102   329710200
  3            3297102   329710200
  4            3297101   329710100
  5            3297101   329710100
Input BGP policy accounting on src IP address enabled
  buckets      packets       bytes
  6            3297102   329710200
  7            3297102   329710200
  8            3297102   329710200
  9            3297101   329710100
  10           3297101   329710100
Output BGP policy accounting on dst IP address enabled
  buckets      packets       bytes
  0                 15         733
Output BGP policy accounting on src IP address enabled
  buckets      packets       bytes
  0                 15         733

show cef ipv4 interface TenGigE0/2/0/2.1 bgp-policy-statistics

TenGigE0/2/0/2.1 is up
Input BGP policy accounting on dst IP address enabled
  buckets      packets       bytes
Input BGP policy accounting on src IP address enabled
  buckets      packets       bytes
Output BGP policy accounting on dst IP address enabled
  buckets      packets       bytes
```

```
  0                15         752
  1             3297102   329710200
  2             3297102   329710200
  3             3297102   329710200
  4             3297101   329710100
  5             3297101   329710100
Output BGP policy accounting on src IP address enabled
  buckets      packets       bytes
  0                15         752
  6             3297102   329710200
  7             3297102   329710200
  8             3297102   329710200
  9             3297101   329710100
  10            3297101   329710100
```

次に、BGP ルートおよびトラフィック インデックスを確認する例を示します。

```
show route bgp

B     10
.1.1.0/24 [20/0] via 10
.17
.1.2, 00:07:09
      Traffic Index 1
B     10
.1.2.0/24 [20/0] via 10
.17
.1.2, 00:07:09
      Traffic Index 2
B    10
.1.3.0/24 [20/0] via 10
.17
.1.2, 00:07:09
      Traffic Index 3
B    10
.1.4.0/24 [20/0] via 10
.17
.1.2, 00:07:09
      Traffic Index 4
B    10
.1.5.0/24 [20/0] via 10
.17
.1.2, 00:07:09
      Traffic Index 5
B    10
.18
.1.0/24 [20/0] via 10
.18
.1.2, 00:07:09
      Traffic Index 6
B    10
.18
.2.0/24 [20/0] via 10
.18
.1.2, 00:07:09
      Traffic Index 7
B    10
.18
.3.0/24 [20/0] via 10
.18
.1.2, 00:07:09
      Traffic Index 8
B    10
.28
.4.0/24 [20/0] via 10
.18
.1.2, 00:07:09
      Traffic Index 9
B    10
.28
.5.0/24 [20/0] via 10
.18
.1.2, 00:07:09
```

```
      Traffic Index 10
B    10
.65
.1.0/24 [20/0] via 10
.45
.0.2, 00:07:09
      Traffic Index 15
B    10
.65
.2.0/24 [20/0] via 10
.45
.0.2, 00:07:09
      Traffic Index 15
B    10
.65
.3.0/24 [20/0] via 10
.45
.0.2, 00:07:09
      Traffic Index 15
B    10
.65
.65
.0/24 [20/0] via 10
.45
.0.2, 00:07:09
      Traffic Index 15
B    10
.65
.5.0/24 [20/0] via 10
.45
.0.2, 00:07:09
      Traffic Index 15
B    10
.65
.6.0/24 [20/0] via 10
.45
.0.2, 00:07:09
      Traffic Index 15
B    10
.65
.7.0/24 [20/0] via 10
.45
.0.2, 00:07:09
      Traffic Index 15
B    10
.65
.8.0/24 [20/0] via 10
.45
.0.2, 00:07:09
      Traffic Index 15
B    10
.65
.9.0/24 [20/0] via 10
.45
.0.2, 00:07:09
      Traffic Index 15
B    10
.65
.10.0/24 [20/0] via 10
.45
.0.2, 00:07:09
      Traffic Index 15
B    10
.66
.1.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 16
B    10
.66
.2.0/24 [20/0] via 10
.32
.0.2, 00:07:09
```

```
      Traffic Index 16
B    10
.66
.3.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 16
B    10
.66
.4.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 16
B    10
.66
.5.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 16
B    10
.66
.6.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 16
B    10
.66
.7.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 16
B    10
.66
.8.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 16
B    10
.66
.9.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 16
B    10
.66
.10.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 16
B    10
.67
.1.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 7
B    10
.67
.2.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 7
B    10
.67
.3.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 7
B    10
.67
.4.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 7
```

```
B    10
.67
.5.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 7
B    10
.67
.6.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 7
B    10
.67
.7.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 7
B    10
.67
.8.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 7
B    10
.67
.9.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 7
B    10
.67
.10.0/24 [20/0] via 10
.32
.0.2, 00:07:09
      Traffic Index 7
B    10
.68
.1.0/24 [20/0] via 10
.8
.0.2, 00:07:09
      Traffic Index 8
B    10
.68
.2.0/24 [20/0] via 10
.8
.0.2, 00:07:09
      Traffic Index 8
B    10
.68
.3.0/24 [20/0] via 10
.8
.0.2, 00:07:09
      Traffic Index 8
B    10
.68
.4.0/24 [20/0] via 10
.8
.0.2, 00:07:09
      Traffic Index 8
B    10
.68
.5.0/24 [20/0] via 10
.8
.0.2, 00:07:09
      Traffic Index 8
B    10
.68
.6.0/24 [20/0] via 10
.8
.0.2, 00:07:09
      Traffic Index 8
B    10
```

```
.68
.7.0/24 [20/0] via 10
.8
.0.2, 00:07:09
      Traffic Index 8
B    10
.68
.8.0/24 [20/0] via 10
.8
.0.2, 00:07:09
      Traffic Index 8
B    10
.68
.9.0/24 [20/0] via 10
.8
.0.2, 00:07:09
      Traffic Index 8
B    10
.68
.10.0/24 [20/0] via 10
.8
.0.2, 00:07:09
      Traffic Index 8

show bgp summary

BGP router identifier 192
.0
.2
.0
, local AS number 100
BGP generic scan interval 60 secs
BGP main routing table version 151
Dampening enabled
BGP scan interval 60 secs
BGP is operating in STANDALONE mode.

Process         RecvTblVer    bRIB/RIB  SendTblVer
Speaker                151         151         151

Neighbor        Spk    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  St/PfxRcd
10
.4
.0.2           0   107      54      53      151    0    0 00:25:26       20
10
.1.0.2           0   108      54      53      151    0    0 00:25:28       20
10
.1.0.2          0   121      53      54      151    0    0 00:25:42        0
10
.1.1.2          0   121      53      53      151    0    0 00:25:06        5
10
.1.2.2          0   121      52      54      151    0    0 00:25:04        0
10
.1.3.2          0   121      52      53      151    0    0 00:25:26        0
10
.1.4.2          0   121      53      54      151    0    0 00:25:41        0
10
.1.5.2          0   121      53      54      151    0    0 00:25:43        0
10
.1.6.2          0   121      51      53      151    0    0 00:24:59        0
10
.1.7.2          0   121      51      52      151    0    0 00:24:44        0
10
.1.8.2          0   121      51      52      151    0    0 00:24:49        0
10
.2
.0.2          0   122      52      54      151    0    0 00:25:21        0
10
.2
.1.2          0  1221      54      54      151    0    0 00:25:43        5
10
.2
.2.2          0  1222      53      54      151    0    0 00:25:38        0
```

```
10
.2
.3.2          0  1223      52      53      151    0    0 00:25:17        0
10
.2
.4.2          0  1224      51      52      151    0    0 00:24:57        0
10
.2
.5.2          0  1225      52      53      151    0    0 00:25:14        0
10
.2
.6.2          0  1226      52      54      151    0    0 00:25:04        0
10
.2
.7.2          0  1227      52      54      151    0    0 00:25:13        0
10
.2
.8.2          0  1228      53      54      151    0    0 00:25:36        0

show bgp 27.1.1.1

BGP routing table entry for 27.1.1.0/24
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                102         102
Paths: (1 available, best #1)
  Not advertised to any peer
  Received by speaker 0
  121
    10
.1.1.2 from 10
.1.1.2 (10
.1.1.2)
      Origin incomplete, localpref 100, valid, external, best
      Community: 27:1 121:1

show bgp 10
.1.1.1

BGP routing table entry for 10
.1.1.0/24
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                107         107
Paths: (1 available, best #1)
  Not advertised to any peer
  Received by speaker 0
  1221
      10
.2
.1.2 from 10
.2
.1.2 (18.1.1.2)
      Origin incomplete, localpref 100, valid, external, best
      Community: 28:1 1221:1

show bgp 10
.0.1.1

BGP routing table entry for 10
.0.1.0/24
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                112         112
Paths: (1 available, best #1)
  Not advertised to any peer
  Received by speaker 0
  107
    10
.1.0.2 from 10
.1.0.2 (10
.1.0.2)
      Origin incomplete, localpref 100, valid, external, best
```

```
      Community: 107:65

show bgp 10
.2
.1.1

BGP routing table entry for 10
.2
.1.0/24
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                122         122
Paths: (1 available, best #1)
  Not advertised to any peer
  Received by speaker 0
  108
    8.1.0.2 from 8.1.0.2 (8.1.0.2)
      Origin incomplete, localpref 100, valid, external, best
      Community: 108:66

show bgp 67.0.1.1

BGP routing table entry for 67.0.1.0/24
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                132         132
Paths: (1 available, best #1)
  Not advertised to any peer
  Received by speaker 0
  107
    4.1.0.2 from 4.1.0.2 (4.1.0.2)
      Origin incomplete, localpref 100, valid, external, best
      Community: 107:67

show bgp 68.0.1.1

BGP routing table entry for 68.0.1.0/24
Versions:
  Process           bRIB/RIB  SendTblVer
  Speaker                142         142
Paths: (1 available, best #1)
  Not advertised to any peer
  Received by speaker 0
  108
    8.1.0.2 from 8.1.0.2 (8.1.0.2)
      Origin incomplete, localpref 100, valid, external, best
      Community: 108:68

show route ipv4 27.1.1.1

Routing entry for 27.1.1.0/24
  Known via "bgp 100", distance 20, metric 0
  Tag 121, type external, Traffic Index 1
  Installed Nov 11 21:14:05.462
  Routing Descriptor Blocks
    17.1.1.2, from 17.1.1.2
      Route metric is 0
  No advertising protos.

show route ipv4 28.1.1.1

Routing entry for 28.1.1.0/24
  Known via "bgp 100", distance 20, metric 0
  Tag 1221, type external, Traffic Index 6
  Installed Nov 11 21:14:05.462
  Routing Descriptor Blocks
    18.1.1.2, from 18.1.1.2
      Route metric is 0
  No advertising protos.

show route ipv4 65.0.1.1

Routing entry for 65.0.1.0/24
```

```
  Known via "bgp 100", distance 20, metric 0
  Tag 107, type external, Traffic Index 15
  Installed Nov 11 21:14:05.462
  Routing Descriptor Blocks
    4.1.0.2, from 4.1.0.2
      Route metric is 0
  No advertising protos.

show route ipv4 66.0.1.1

Routing entry for 66.0.1.0/24
  Known via "bgp 100", distance 20, metric 0
  Tag 108, type external, Traffic Index 16
  Installed Nov 11 21:14:05.462
  Routing Descriptor Blocks
    8.1.0.2, from 8.1.0.2
      Route metric is 0
  No advertising protos.

show route ipv4 67.0.1.1

Routing entry for 67.0.1.0/24
  Known via "bgp 100", distance 20, metric 0
  Tag 107, type external, Traffic Index 7
  Installed Nov 11 21:14:05.462
  Routing Descriptor Blocks
    4.1.0.2, from 4.1.0.2
      Route metric is 0
  No advertising protos.

show route ipv4 68.0.1.1

Routing entry for 68.0.1.0/24
  Known via "bgp 100", distance 20, metric 0
  Tag 108, type external, Traffic Index 8
  Installed Nov 11 21:14:05.462
  Routing Descriptor Blocks
    8.1.0.2, from 8.1.0.2
      Route metric is 0
  No advertising protos.

show cef ipv4 27.1.1.1

27.1.1.0/24, version 263, source-destination sharing
Prefix Len 24, Traffic Index 1, precedence routine (0)
  via 17.1.1.2, 0 dependencies, recursive
    next hop 17.1.1.2/24, TenGigE0/2/0/2.1 via 17.1.1.0/24
    valid remote adjacency
  Recursive load sharing using 17.1.1.0/24

show cef ipv4 28.1.1.1

28.1.1.0/24, version 218, source-destination sharing
Prefix Len 24, Traffic Index 6, precedence routine (0)
  via 18.1.1.2, 0 dependencies, recursive
    next hop 18.1.1.2/24, TenGigE0/2/0/4.1 via 18.1.1.0/24
    valid remote adjacency
  Recursive load sharing using 18.1.1.0/24

show cef ipv4 65.0.1.1

65.0.1.0/24, version 253, source-destination sharing
Prefix Len 24, Traffic Index 15, precedence routine (0)
  via 4.1.0.2, 0 dependencies, recursive
    next hop 4.1.0.2/16, gigabitethernet0/0/0/4 via 4.1.0.0/16
    valid remote adjacency
  Recursive load sharing using 4.1.0.0/16

show cef ipv4 66.0.1.1

66.0.1.0/24, version 233, source-destination sharing
Prefix Len 24, Traffic Index 16, precedence routine (0)
  via 8.1.0.2, 0 dependencies, recursive
```

```
    next hop 8.1.0.2/16, gigabitethernet 0/0/0/8 via 8.1.0.0/16
    valid remote adjacency
  Recursive load sharing using 8.1.0.0/16

show cef ipv4 67.0.1.1

67.0.1.0/24, version 243, source-destination sharing
Prefix Len 24, Traffic Index 7, precedence routine (0)
  via 4.1.0.2, 0 dependencies, recursive
    next hop 4.1.0.2/16, gigabitethernet 0/0/0/4 via 4.1.0.0/16
    valid remote adjacency
  Recursive load sharing using 4.1.0.0/16

show cef ipv4 68.0.1.1

68.0.1.0/24, version 223, source-destination sharing
Prefix Len 24, Traffic Index 8, precedence routine (0)
  via 8.1.0.2, 0 dependencies, recursive
    next hop 8.1.0.2/16, gigabitethernet0/0/0/8 via 8.1.0.0/16
    valid remote adjacency
  Recursive load sharing using 8.1.0.0/16

show cef ipv4 27.1.1.1 detail

27.1.1.0/24, version 263, source-destination sharing
Prefix Len 24, Traffic Index 1, precedence routine (0)
  via 17.1.1.2, 0 dependencies, recursive
    next hop 17.1.1.2/24, TenGigE0/2/0/2.1 via 17.1.1.0/24
    valid remote adjacency

  Recursive load sharing using 17.1.1.0/24
  Load distribution: 0 (refcount 6)

  Hash  OK  Interface                 Address  Packets
  1     Y   TenGigE0/2/0/2.1          (remote)               0

show cef ipv4 28.1.1.1 detail

28.1.1.0/24, version 218, source-destination sharing
Prefix Len 24, Traffic Index 6, precedence routine (0)
  via 18.1.1.2, 0 dependencies, recursive
    next hop 18.1.1.2/24, TenGigE0/2/0/4.1 via 18.1.1.0/24
    valid remote adjacency

  Recursive load sharing using 18.1.1.0/24
  Load distribution: 0 (refcount 6)

  Hash  OK  Interface                 Address  Packets
  1     Y   TenGigE0/2/0/4.1          (remote)               0

show cef ipv4 65.0.1.1 detail

65.0.1.0/24, version 253, source-destination sharing
Prefix Len 24, Traffic Index 15, precedence routine (0)
  via 4.1.0.2, 0 dependencies, recursive
    next hop 4.1.0.2/16, gigabitethernet0/0/0/4 via 4.1.0.0/16
    valid remote adjacency

  Recursive load sharing using 4.1.0.0/16
  Load distribution: 0 (refcount 21)

  Hash  OK  Interface                 Address  Packets
  1     Y   gigabitethernet0/0/0/4               (remote)               0

show cef ipv4 66.0.1.1 detail

66.0.1.0/24, version 233, source-destination sharing
Prefix Len 24, Traffic Index 16, precedence routine (0)
  via 8.1.0.2, 0 dependencies, recursive
    next hop 8.1.0.2/16, gigabitethernet0/0/0/8 via 8.1.0.0/16
    valid remote adjacency

  Recursive load sharing using 8.1.0.0/16
```

```
  Load distribution: 0 (refcount 21)

  Hash  OK  Interface                 Address  Packets
  1     Y   gigabitethernet 0/0/0/8               (remote)               0

show cef ipv4 67.0.1.1 detail

67.0.1.0/24, version 243, source-destination sharing
Prefix Len 24, Traffic Index 7, precedence routine (0)
  via 4.1.0.2, 0 dependencies, recursive
    next hop 4.1.0.2/16, gigabitethernet 0/0/0/4 via 4.1.0.0/16
    valid remote adjacency

  Recursive load sharing using 4.1.0.0/16
  Load distribution: 0 (refcount 21)

  Hash  OK  Interface                 Address  Packets
  1     Y   gigabitethernet 0/0/0/4                (remote)               0

show cef ipv4 68.0.1.1 detail

68.0.1.0/24, version 223, source-destination sharing
Prefix Len 24, Traffic Index 8, precedence routine (0)
  via 8.1.0.2, 0 dependencies, recursive
    next hop 8.1.0.2/16, gigabitethernet 0/0/0/8 via 8.1.0.0/16
    valid remote adjacency

  Recursive load sharing using 8.1.0.0/16
  Load distribution: 0 (refcount 21)

  Hash  OK  Interface                 Address  Packets
  1     Y   gigabitethernet 0/0/0/8                (remote)               0
```

ユニキャスト RPF チェックの設定：例

次に、ユニキャスト RPF チェックを設定する例を示します。

```
configure
interface gigabitethernet 0/0/0/1
ipv4 verify unicast source reachable-via rx
end
```

モジュラ サービス カードからルート プロセッサ上の管理イーサネット インターフェイスへのスイッチングの設定：例

次に、MSC からルート プロセッサ上の管理イーサネット インターフェイスへのスイッチングを設定する例を示します。

```
configure
rp mgmtethernet forwarding
end
```

BGP 属性ダウンロードの設定：例

次に、BGP 属性ダウンロード機能を設定する例を示します。

```
router configure
show cef bgp attribute {attribute-id| local-attribute-id}
```

その他の参考資料

ここでは、CEF の実装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
CEF コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*Cisco Express Forwarding Commands*」の章
BGP コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『*Cisco ASR 9000 Series Aggregation Services Router Routing Command Reference*』の「*BGP Commands*」の章
リンク構築用コマンド：コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『*Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Command Reference*』の「*Link Bundling Commands*」の章

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
—	MIB を特定してダウンロードするには、次の URL にある Cisco MIB Locator を使用し、[Cisco Access Products] メニューからプラットフォームを選択します。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

第 4 章

ダイナミック ホスト コンフィギュレーション プロトコルの実装

この章では、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）の設定に使用する概念およびタスクについて説明します。

（注） この章に記載されている DHCP コマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』を参照してください。 この章で使用される他のコマンドの説明については、コマンド リファレンスのマスター索引を参照するか、またはオンラインで検索してください。

ダイナミック ホスト コンフィギュレーション プロトコルの実装の機能履歴

リリース	変更内容
リリース 3.7.2	この機能が導入されました。

DHCP リレー エージェントの設定の前提条件

DHCP リレー エージェントを設定するには、次の前提条件を満たす必要があります。

- 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。
- 設定済みで動作している DHCP クライアントおよび DHCP サーバ
- リレー エージェントと DCHP サーバとの間の接続

DHCP リレー エージェントに関する情報

DHCP リレー エージェントは、共有の物理サブネットに存在しないクライアントとサーバとの間で DHCP パケットを転送するホストです。 リレー エージェント転送は、IP ルータの通常の転送とは異なります。通常の転送では、IP データグラムがネットワーク間で透過的にスイッチングされます。

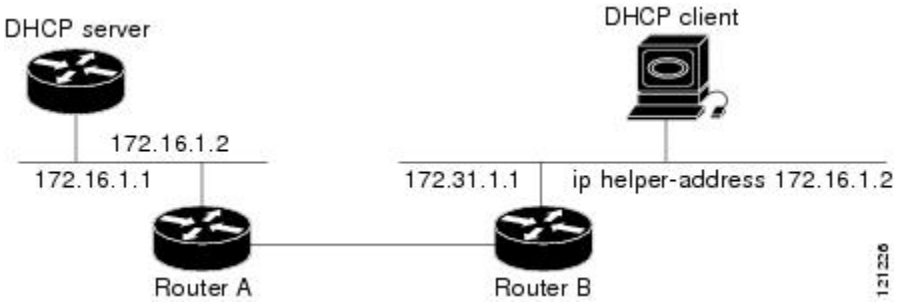
DHCP クライアントは、自身の所属先のネットワークに関する情報を保持していないときには、ユーザ データグラム プロトコル（UDP）ブロードキャストを使用して、DHCPDISCOVER メッセージを送信します。

サーバが含まれていないネットワーク セグメント上にクライアントがある場合、DHCP パケットが別のネットワーク セグメント上のサーバに届くようにするには、そのネットワーク セグメントにリレー エージェントが必要です。 ほとんどのルータはブロードキャスト トラフィックを転送するように設定されていないため、UDP ブロードキャスト パケットは転送されません。 DHCP リレー プロファイルを設定することにより DHCP パケットをリモート サーバに転送するように DHCP リレー エージェントを設定し、そこに 1 つ以上のヘルパー アドレスを設定できます。 プロファイルをインターフェイスまたは VRF に割り当てることができます。

図 2：ヘルパー アドレスを使用した UDP ブロードキャストの DHCP サーバへの転送, （105 ページ） に、そのプロセスを示します。 DHCP クライアントが、IP アドレスの要求と追加設定パラメータをローカル LAN 上でブロードキャストしています。 DHCP リレー エージェントとして機能するルータ B は、ブロードキャストを取得し、宛先アドレスを DHCP サーバのアドレスに変更し、別のインターフェイスにメッセージを送信します。 リレー エージェントは、DHCP クライアントのパケットを受け取ったインターフェイスの IP アドレスを DHCP パケットのゲートウェイ アドレス（giaddr）フィールドに挿入します。これにより、DHCP サーバは、どのサブネットがオファーを受信するかを判断し、適切な IP アドレス範囲を特定できます。 リレー エージェントは、

メッセージを（リレープロファイルのヘルパーアドレスによって指定される）サーバアドレス、この場合は 172.16.1.2 にユニキャストします。

図 *2*：ヘルパー アドレスを使用した *UDP* ブロードキャストの *DHCP* サーバへの転送



DHCP リレー エージェントを設定およびイネーブルにする方法

ここでは、次のタスクについて説明します。

DHCP リレー エージェントの設定およびイネーブル化

このタスクでは、DHCP リレー エージェントを設定し、イネーブル化する方法について説明します。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **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。

	コマンドまたはアクション	目的
ステップ 2	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	DHCP IPv4 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	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

DHCP リレー プロファイルの設定

このタスクでは、DHCP リレー エージェントを設定し、イネーブル化する方法について説明します。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **relay**
4. **helper-address** [**vrf** *vrf- name*] *address*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	DHCP IPv4 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	**profile** *profile-name* **relay** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile client relay`	DHCP IPv4 プロファイル リレー サブモードを開始します。
ステップ 4	**helper-address** [**vrf** *vrf- name*] *address* 例： `RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf vrf1 10.10.1.1`	BOOTP や DHCP など、UDP ブロードキャストを転送します。 • *address* 引数の値には、特定の DHCP サーバアドレスまたはネットワーク アドレス（宛先ネットワーク セグメントに他にも DHCP サーバがある場合）を指定できます。 ネットワーク アドレスを使用することで、他のサーバも DHCP 要求に応答できるようになります。 • サーバが複数ある場合は、各サーバにヘルパー アドレスを 1 つ設定してください。

	コマンドまたはアクション	目的

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

DHCPv6（ステートレス）リレー エージェントの設定

クライアント メッセージの転送先のアドレスを指定し、インターフェイスで IPv6 リレー サービス用にダイナミック ホスト コンフィギュレーション プロトコル（DHCP）をイネーブルにするには、このタスクを実行します。

手順の概要

1. **configure**
2. **dhcp ipv6**
3. **interface** *type interface-path-id* **relay**
4. **destination** *ipv6-address*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**dhcp ipv6**
	
	例：
	`RP/0/RSP0/CPU0:router(config) # dhcp ipv6`
	`RP/0/RSP0/CPU0:router(config-dhcpv6)#`
		DHCP for IPv6 をイネーブルにし、DHCP IPv6 コンフィギュレーション モードを開始します。

ステップ 3	**interface** *type interface-path-id* **relay** 例： `RP/0/RSP0/CPU0:router(config-dhcpv6) # interface tenGigE 0/5/0/0 relay`	インターフェイス タイプおよびインターフェイス パス ID を指定し、ルータをインターフェイス コンフィギュレーション モードに設定し、インターフェイスで DHCPv6 リレー サービスをイネーブルにします。
ステップ 4	**destination** *ipv6-address* 例：	クライアント パケットの転送先のアドレスを指定します。 インターフェイスでリレー サービスがイネーブルになっているときは、そのインターフェイスに届いた DHCP for IPv6 メッセージは設定済みのすべてのリレー宛先に転送されます。 着信

	コマンドまたはアクション	目的
	`RP/0/RSP0/CPU0:router(config-dhcpv6-if) # destination 10:10::10`	DHCP for IPv6 メッセージが、そのインターフェイス上のクライアントから届く場合や、別のリレーエージェントによってリレーされる場合があります。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

インターフェイスでの DHCP リレー エージェントのイネーブル化

このタスクでは、インターフェイスで Cisco IOS XR DHCP リレー エージェントをイネーブルにする方法について説明します。

（注） Cisco IOS XR ソフトウェアでは、DHCP リレー エージェントはデフォルトではディセーブルになっています。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. **interface type name relay profile** *profile-name*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	DHCP IPv4 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	**interface type name relay profile** *profile-name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# interface gigabitethernet 0/0/0 /0 relay profile client`	リレープロファイルをインターフェイスにアタッチします

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		◦ **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 DHCP リレーのディセーブル化

このタスクでは、インターフェイスにプロファイルを割り当てないことにより、インターフェイスで DHCP リレーをディセーブルにする方法について説明します。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. **interface** *type name* **none**
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	DHCP IPv4 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 3	**interface** *type name* **none** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# interface gigabitethernet 0/1/4/1 none`	インターフェイスで DHCP リレーをディセーブルにします。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

VRF での DHCP リレーのイネーブル化

このタスクでは、VRF で DHCP リレー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

1. **configure**
2. **dhcp ipv4**
3. **vrf** *vrf-name* **relay profile** *profile-name*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	DHCP IPv4 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	**vrf** *vrf-name* **relay profile** *profile-name* 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# vrf default relay profile client`	VRF で DHCP リレーをイネーブルにします。
ステップ 4	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	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

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	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。

リレー エージェント情報機能の設定

このタスクでは、DHCP リレー エージェント情報オプション処理機能を設定する方法について説明します。

DHCP リレー エージェントは、すでにリレー情報を持つ別の DHCP リレー エージェントからのメッセージを受信する場合があります。 デフォルトでは、1 つ前のリレー エージェントからのリレー情報が（置換オプションを使用して）置換されます。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **relay**
4. **relay information option**
5. **relay information check**
6. **relay information policy** {**drop** | **keep**}
7. **relay information option allow-untrusted**
8. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	DHCP IPv4 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	**profile** *profile-name* **relay** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile client relay`	DHCP IPv4 プロファイル リレー サブモードを開始します。
ステップ 4	**relay information option** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# relay information option`	DHCP サーバへの転送された BOOTREQUEST メッセージに、システムが DHCP リレー エージェント情報オプション（Option 82 フィールド）を挿入できるようにします。 • このオプションは、クライアントが発信した DHCP パケットをサーバに転送する際に、リレー エージェントによって挿入されます。 このオプションを認識するサーバは、その情報を使用して、IP アドレスや他のパラメータ割り当てポリシーを実装できます。 DHCP サーバは応答時に、リレー エージェントにオプションをエコーします。 リレー エージェントは、クライアントに応答を転送する前に、オプションを削除します。 • リレーエージェント情報は、サブオプションが 1 つ以上含まれている単一の DHCP オプションとして編成されます。 これらのオプションには、リレー エージェントが認識する情報が含まれています。 サポートされているサブオプションは次のとおりです。 ◦リモート ID ◦回線 ID

	コマンドまたはアクション	目的
		（注） この機能は、デフォルトではディセーブルになっています。
ステップ 5	**relay information check** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# relay information check`	（任意）転送されたBOOTREPLYメッセージ内のリレーエージェント情報オプションが有効かどうかをチェックするようにDHCPを設定します。リレーエージェントは、無効なメッセージを受信した場合には、そのメッセージをドロップします。有効なメッセージを受信した場合には、リレーエージェント情報オプションフィールドを削除し、パケットを転送します。 • DHCPは、デフォルトではDHCPサーバから受信したDHCP応答パケットのリレーエージェント情報オプション フィールドが有効であるかどうかをチェックしません。 （注） ディセーブルになっていたこの機能を再びイネーブルにするには、**relay information check** コマンドを使用します。
ステップ 6	**relay information policy** {**drop** | **keep**} 例： `RP/0/RSP0/CPU0:router(config)# dhcp relay information policy drop`	（任意）DHCPリレーエージェントの再転送ポリシー、つまりリレーエージェントがリレー情報をドロップするのか、保持するのかを設定します。 DHCPリレーエージェントは、デフォルトではリレー情報オプションを置換します。
ステップ 7	**relay information option allow-untrusted** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# relay information option allow-untrusted`	（任意）既存のリレー情報オプションがあり、かつ giaddr がゼロに設定されている BOOTREQUEST パケットを廃棄しないように DHCP IPv4 Relay を設定します。
ステップ 8	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	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]:`

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	◦ **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。

リレー エージェント giaddr ポリシーの設定

このタスクでは、すでにゼロ以外の giaddr 属性が含まれている受信した BOOTREQUEST パケットに対して DHCP リレー エージェントの処理機能を設定する方法について説明します。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. **profile relay**
4. **giaddr policy** {**replace** | **drop**}
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	DHCP IPv4 コンフィギュレーション サブモードをイネーブルにします。
ステップ 3	**profile relay** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile client relay`	プロファイル リレー サブモードをイネーブルにします。
ステップ 4	**giaddr policy** {**replace** | **drop**} 例： `RP/0/RSP0/CPU0:router(config-dhcpv4-relay-profile)# giaddr policy drop`	giaddr ポリシーを指定します。 • 置換：既存の giaddr 値を、生成された値に置き換えます。 • ドロップ：既存のゼロ以外の giaddr 値を持つパケットをドロップします。 DHCP リレーエージェントは、デフォルトでは既存の giaddr 値を保持します。 •

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **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

	コマンドまたはアクション	目的
		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。

プレフィックス委任の DHCPv6 リレー エージェント通知

プレフィックス委任の DHCPv6 リレー エージェント通知を使用すると、DHCPv6 リレー エージェントとして動作するルータは、リレー エージェントからクライアントに中継される DHCPv6 RELAY-REPLY パケットの内容を確認することによって、プレフィックス委任オプションを見つけることができます。リレーエージェントは、プレフィックス委任オプションを検出すると、委任されるプレフィックスに関する情報を抽出し、プレフィックス委任情報と一致する IPv6 加入者ルートをリレーエージェントに挿入します。その後リレー経由でそのプレフィックスに宛てられたパケットは、プレフィックス委任に含まれる情報に基づいて転送されます。 IPv6 加入者ルートは、プレフィックス委任のリース期間が経過するか、またはリレー エージェントがプレフィックス委任を解放するクライアントから解放パケットを受信するまで、ルーティング テーブルに保持されます。

リレー エージェントは、自動的に加入者ルート管理を行います。

IPv6 ルートは、リレー エージェントが RELAY-REPLY パケットを中継すると追加され、プレフィックス委任のリース期限が経過するか、リレー エージェントが解放メッセージを受信すると削除されます。プレフィックス委任のリース期間を延長するときに、リレーエージェントのルーティング テーブル内の IPv6 加入者ルートを更新できます。

この機能により、IPv6 ルートはリレー エージェントのルーティング テーブルに保持されます。この登録された IPv6 アドレスを使用すると、ユニキャスト RPF（uRPF）の動作が可能になりますが、そのためには、リバース ルックアップを実行するルータがリレー エージェント上の IPv6 アドレスが正しく、スプーフィングされていないことを確認できるようにします。リレーエージェントのルーティング テーブル内の IPv6 ルートを他のルーティング プロトコルに再配布して、サブネットを他のノードにアドバタイズできます。 クライアントが DHCP_DECLINE メッセージを送信すると、ルートは削除されます。

プレフィックス委任のための DHCPv6 ステートフル リレー エージェントの設定

プレフィックス委任用にダイナミック ホスト コンフィギュレーション プロトコル（DHCP）IPv6 リレー エージェント通知を設定するには、このタスクを実行します。

手順の概要

1. **configure**
2. **dhcp ipv6**
3. **profile** *profile-name* **proxy**
4. **helper-address** *ipv6-address* **interface** *type interface-path-id*
5. **exit**
6. **interface** *type interface-path-id* **proxy**
7. **profile** *profile-name*
8. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**dhcp ipv6**
	例：
	`RP/0/RSP0/CPU0:router(config) # dhcp ipv6`
	`RP/0/RSP0/CPU0:router(config-dhcpv6)#`
		IPv6 の DHCP をイネーブルにし、DHCP IPv6 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* **proxy**
	例：
	`RP/0/RSP0/CPU0:router(config-dhcpv6)# profile downstream proxy`
	`RP/0/RSP0/CPU0:router(config-dhcpv6-profile)#`
		プロキシ 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。

	コマンドまたはアクション	目的

ステップ 4	**helper-address** *ipv6-address* **interface** *type interface-path-id*
	
	例：
	
	`RP/0/RSP0/CPU0:router(config-dhcpv6-profile)# helper-address 2001:db8::1 GigabitEthernet 0/1/0/1`
	`RP/0/RSP0/CPU0:router(config-dhcpv6-profile)`
		DHCP IPv6 リレー エージェントを設定します。

ステップ 5	**exit**
	
	例：
	
	`RP/0/RSP0/CPU0:router(config-dhcpv6-profile)# exit`
	`RP/0/RSP0/CPU0:router(config-dhcpv6)#`
		プロファイル コンフィギュレーション モードを終了します。

ステップ 6	**interface** *type interface-path-id* **proxy**
	
	例：
	
	`RP/0/RSP0/CPU0:router(config-dhcpv6)# interface GigabitEthernet 0/1/0/0 proxy`
	`RP/0/RSP0/CPU0:router(config-dhcpv6-if)#`
		インターフェイスで IPv6 DHCP をイネーブルにし、IPv6 DHCP ステートフル リレー エージェントとして機能します。

ステップ 7	**profile** *profile-name*
	
	例：
	
	`RP/0/RSP0/CPU0:router(config-dhcpv6-if)# profile downstream`
	`RP/0/RSP0/CPU0:router(config-dhcpv6-if)#`
		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 8	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		◦ **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。

DHCP リレー エージェントの設定例

ここでは、次の設定例について説明します。

DHCP リレー プロファイル：例

次に、Cisco IOS XR リレー プロファイルを設定する例を示します。

```
dhcp ipv4
 profile client relay
  helper-address vrf foo 10.10.1.1
 !
! ...
```

インターフェイス上の DHCP リレー：例

次に、インターフェイスで DHCP リレー エージェントをイネーブルにする例を示します。

```
dhcp ipv4
 interface gigabitethernet 0/1/1/0 relay profile client
!
```

VRF 上の DHCP リレー：例

次に、VRF で DHCP リレー エージェントをイネーブルにする例を示します。

```
dhcp ipv4
 vrf default relay profile client
!
```

リレー エージェント情報オプションのサポート：例

次に、リレー エージェントと、DHCP リレー情報オプションの挿入および削除をイネーブルにする例を示します。

```
dhcp ipv4
 profile client relay
relay information option

 !
!
```

リレー エージェント giaddr ポリシー：例

次に、リレー エージェント giaddr ポリシーを設定する例を示します。

```
dhcp ipv4
 profile client relay
  giaddr policy drop
 !
!
```

DHCP スヌーピングの実装

DHCP スヌーピングの設定の前提条件

DHCP IPv4 スヌーピング リレー エージェント ブロードキャスト フラグ ポリシーを設定するには、次の前提条件を満たす必要があります。

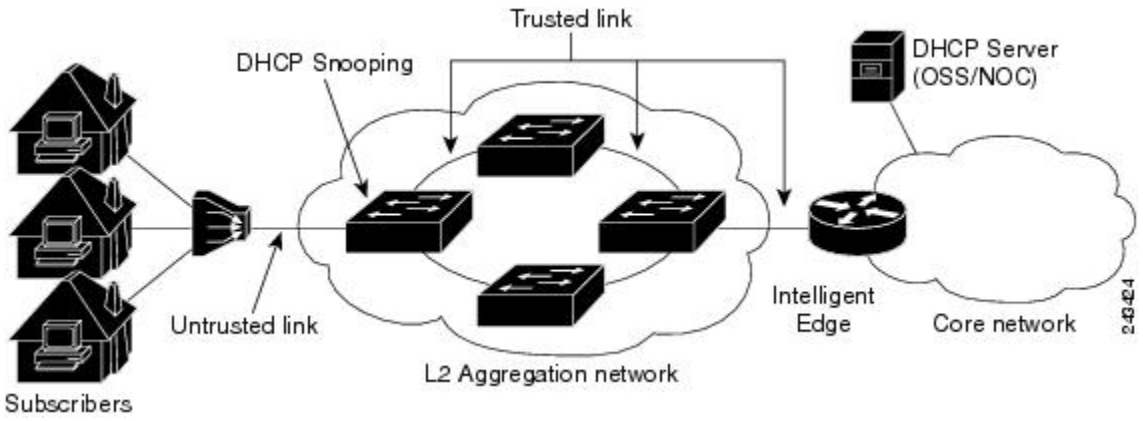
- 適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。 このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。
- Cisco IOS XR ソフトウェアが動作している Cisco ASR 9000 シリーズ ルータ。
- 設定済みで動作している DHCP クライアントおよび DHCP サーバ。

DHCP スヌーピングに関する情報

DHCP スヌーピングは、アグリゲーション ネットワークのエッジで着目されている機能です。 加入者向けの入口にセキュリティ機能が適用されます。 リレーエージェント情報のオプション情報を使用して、加入者の回線が識別されます。回線は、加入者の自宅に至る DSL 回線か、アグリゲーション ネットワークの最初のポートのいずれかになります。

DHCP スヌーピングの中心となる考えは、信頼できるリンクと信頼できないリンクという考えです。 信頼できるリンクとは、そのリンク上のトラフィックに安全にアクセスできるリンクです。 信頼できないリンクでは、加入者のアイデンティティおよび加入者のトラフィックを判別できません。 DHCP スヌーピングを信頼できないリンクで実行すると、加入者のアイデンティティを提供できます。 図 3：アグリゲーション ネットワークでの DHCP スヌーピング，（125 ページ） に、アグリゲーション ネットワークを示します。 DSLAM からアグリゲーション ネットワークに至るリンクは信頼できないリンクであり、DHCP スヌーピングのポイント オブ プレゼンスです。 アグリゲーション ネットワーク内のスイッチ同士を接続するリンクおよびアグリゲーション ネットワークからインテリジェント エッジに至るリンクは、信頼できるリンクであると考えられます。

図 *3*：アグリゲーション ネットワークでの *DHCP* スヌーピング



信頼できるポートおよび信頼できないポート

信頼できるポートでは、DHCP スヌーピングによって DHCP BOOTREQUEST パケットが転送されます。クライアントのアドレス リースはトラッキングされず、クライアントはポートにバインドされません。 DHCP BOOTREPLY パケットは転送されます。

クライアントから信頼できないポートに最初の DHCP BOOTREQUEST パケットが届くと、DHCP スヌーピングはクライアントをブリッジポートにバインドし、クライアントのアドレス リースをトラッキングします。 そのアドレス リースが期限切れになると、クライアントはデータベースから削除され、ブリッジポートからアンバインドされます。バインドが存在する限り、このクライアントからこのブリッジポートに届いたパケットは処理されて転送されます。 このクライアントから別のブリッジポートに届いたパケットは、バインドが存在しても、ドロップされます。 DHCP スヌーピングは、このクライアントがバインドされているブリッジポートにクライアントの DHCP

BOOTREPLY パケットのみを転送します。 信頼できないポートに届いた DHCP BOOTREPLY パケットは転送されません。

ブリッジ ドメインでの DHCP スヌーピング

ブリッジ ドメインで DHCP スヌーピングをイネーブルにするには、少なくとも 2 つのプロファイル、信頼できるプロファイルと信頼できないプロファイルが必要になります。 信頼できないプロファイルは、クライアント側ポートに割り当てられ、信頼できるプロファイルはサーバ側ポートに割り当てられます。 ほとんどの場合、クライアント側ポートが数多くあり、サーバ側ポートはごくわずかです。 最も簡単な例が、クライアント側ポートとサーバ側ポートという 2 つのポートがあり、信頼できないプロファイルがクライアント側ポートに明示的に割り当てられ、信頼できるプロファイルがサーバ側ポートに割り当てられている例です。

ブリッジ ドメインへのプロファイルの割り当て

通常はクライアント側ポートが数多くあり、サーバ側ポートが少数であるため、オペレータは信頼できないプロファイルをブリッジドメインに割り当てます。この設定では、信頼できないプロファイルがブリッジドメイン内のあらゆるポートに効果的に割り当てられます。このアクションにより、オペレータは信頼できないプロファイルをすべてのクライアント側ポートに明示的に割り当てる手間を省くことができます。 DHCP スヌーピングが正しく機能するためには、サーバ側ポートに信頼できる DHCP スヌーピング プロファイルも必要になるため、サーバ側ポートに信頼できる DHCP スヌーピング プロファイルを明確に設定して、サーバ側ポートに対するこの信頼できない DHCP スヌーピング プロファイルの割り当てをオーバーライドします。 ブリッジ ドメインに DHCP スヌーピングを必要としないポートがある場合、それらのポートには **none** プロファイルを割り当ててください。これにより、DHCP スヌーピングがディセーブルになります。

リレー情報オプション

クライアント ポートに割り当てられるときにのみ、リレー情報オプション（Option 82）を DHCP クライアント パケットに挿入するように、DHCP スヌーピング プロファイルを設定できます。 DHCP クライアント パケットに受信時点ですでにヌルの giaddr およびリレー情報オプションがあるときには、**relay information option allow-untrusted** コマンドで対処します。 これは、DHCP スヌーピングの信頼できる/信頼できないポートとは別の条件です。 **relay information option allow-untrusted** コマンドは、DHCP スヌーピング アプリケーションが信頼できないリレー情報オプションをどのように処理するかを決定するものです。

DHCP スヌーピングを設定する方法

ここでは、次のタスクについて説明します。

ブリッジ ドメインでの DHCP スヌーピングのイネーブル化

次の設定では、クライアント側ポートとサーバ側ポートという 2 つのポートを作成します。 ステップ 1 ～ 8 では、信頼できない DHCP スヌーピング プロファイルをクライアント ブリッジ ポートに割り当て、信頼できる DHCP スヌーピング プロファイルをサーバ ブリッジ ポートに割り当てます。 ステップ 9 ～ 18 では、信頼できない DHCP スヌーピング プロファイルをブリッジ ドメインに割り当て、信頼できる DHCP スヌーピング プロファイルをサーバ ブリッジ ポートに割り当てます。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. **profile** *untrusted-profile-name* **snoop**
4. **exit**
5. **dhcp ipv4**
6. **profile** *profile-name* **snoop**
7. **trusted**
8. **exit**
9. **l2vpn**
10. **bridge group** *group-name*
11. **bridge-domain** *bridge-domain-name*
12. **interface** *type interface-path-id*
13. **dhcp ipv4 snoop profile** *untrusted-profile-name*
14. **interface** *type interface-path-id*
15. **dhcp ipv4 snoop profile** *trusted-profile-name*
16. **exit**
17. **exit**
18. 次のいずれかのコマンドを使用します。
    - **end**
    - **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。

	コマンドまたはアクション	目的
ステップ 2	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	DHCP IPv4 プロファイル 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	**profile** *untrusted-profile-name* **snoop** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile untrustedClientProfile snoop`	クライアント ポート用に信頼できない DHCP スヌーピング プロファイルを設定します。
ステップ 4	**exit** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# exit`	DHCP IPv4 プロファイル コンフィギュレーション モードを終了します。
ステップ 5	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	DHCP for IPv4 をイネーブルにし、DHCP IPv4 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 6	**profile** *profile-name* **snoop** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile trustedServerProfile snoop`	サーバ ポート用に信頼できる DHCP スヌーピング プロファイルを設定します。
ステップ 7	**trusted** 例： `RP/0/RSP0/CPU0:router(config-dhcv4)# trusted`	DHCP スヌーピング プロファイルを信頼できるものとして設定します。
ステップ 8	**exit** 例： `RP/0/RSP0/CPU0:router(config-dhcv4)# exit`	DHCP IPv4 プロファイル コンフィギュレーション モードを終了します。
ステップ 9	**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 10	**bridge group** *group-name* 例： `RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group ccc`	ブリッジ グループを作成してブリッジ ドメインを含め、l2vpn ブリッジ グループ 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 11	**bridge-domain** *bridge-domain-name* 例： `RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain ddd`	ブリッジ ドメインを確立します。
ステップ 12	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface gigabitethernet 0/1/0/0`	インターフェイスを識別します。
ステップ 13	**dhcp ipv4 snoop profile** *untrusted-profile-name* 例： `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# dhcp ipv4 snoop profile untrustedClientProfile`	信頼できない DHCP スヌーピング プロファイルをブリッジ ポートにアタッチします。
ステップ 14	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# gigabitethernet 0/1/0/1`	インターフェイスを識別します。
ステップ 15	**dhcp ipv4 snoop profile** *trusted-profile-name* 例： `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# dhcp ipv4 snoop profile trustedServerProfile`	信頼できる DHCP スヌーピング プロファイルをブリッジ ポートにアタッチします。
ステップ 16	**exit** 例： `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-ac)# exit`	l2vpn ブリッジ グループ ブリッジドメイン 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 17	**exit** 例： `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# exit`	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 サブモードを終了します。
ステップ 18	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **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。

特定のブリッジ ポートでの DHCP スヌーピングのディセーブル化

次の設定では、ブリッジ ポート GigabitEthernet 0/1/0/1 および GigabitEthernet 0/1/0/2 を除いて、ブリッジ ドメイン ISP1 のすべてのブリッジ ポートで DHCP がパケットをスヌーピングできるようにします。 DHCP スヌーピングは、ブリッジ ポート GigabitEthernet 0/1/0/1 でディセーブルになっています。 ブリッジ ポート GigabitEthernet 0/1/0/2 は、サーバに接続する信頼できるポートです。この例では、他にイネーブルになっている機能はなく、DHCP スヌーピングのみが実行されています。

手順の概要

1. **configure**
2. **l2vpn**
3. **bridge group** *group-name*
4. **bridge-domain** *bridge-domain-name*
5. **dhcp ipv4 snoop profile** *profile-name*
6. **interface** *type interface-path-id*
7. **dhcp ipv4 none**
8. **interface** *type interface-path-id*
9. **dhcp ipv4 snoop profile** *profile-name*
10. **exit**
11. **exit**
12. 次のいずれかのコマンドを使用します。
    - **end**
    - **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`	2vpn 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	**bridge group** *group-name* 例： `RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group GRP1`	ブリッジ グループを作成してブリッジ ドメインを含め、l2vpnブリッジグループ 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	**bridge-domain** *bridge-domain-name* 例： `RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain ISP1`	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 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 5	**dhcp ipv4 snoop profile** *profile-name* 例： `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# dhcp ipv4 snoop profile untrustedClientProfile`	信頼できない DHCP スヌーピング プロファイルをブリッジ ドメインにアタッチします。
ステップ 6	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface gigabitethernet 0/1/0/1`	インターフェイスを特定し、l2vpn ブリッジ グループ ブリッジ ドメイン 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。
ステップ 7	**dhcp ipv4 none** 例： `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd-if)# dhcp ipv4 none`	ポートで DHCP スヌーピングをディセーブルにします。
ステップ 8	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# interface gigabitethernet 0/1/0/2`	インターフェイスを特定し、l2vpn ブリッジ グループ ブリッジ ドメイン 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。
ステップ 9	**dhcp ipv4 snoop profile** *profile-name* 例： `RP/0/RSP0/CPU0:router(config-l2vpn-bg-bd)# dhcp ipv4 snoop profile trustedServerProfile`	信頼できる DHCP スヌーピング プロファイルをポートにアタッチします。
ステップ 10	**exit** 例： `RP/0/RSP0/CPU0:router(config-l2vpn-bd-bg)# exit`	l2vpn ブリッジ ドメイン ブリッジ グループ 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 11	**exit** 例： `RP/0/RSP0/CPU0:router(config-l2vpn-bg)# exit`	l2vpn ブリッジ ドメイン サブモードを終了します。
ステップ 12	次のいずれかのコマンドを使用します。 • **end** • **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]:`

	コマンドまたはアクション	目的

	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		◦ **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。

リレー情報オプションの使用方法

このタスクでは、リレー情報コマンドを使用して、リレー情報オプション（Option 82）を DHCP クライアント パケットに挿入し、信頼できないリレー情報オプションとともに DHCP パケットを転送する方法を示します。

手順の概要

1. **configure**
2. **dhcp ipv4**
3. **profile** *profile-name* **snoop**
4. **relay information option**
5. **relay information option allow-untrusted**
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**dhcp ipv4** 例： `RP/0/RSP0/CPU0:router(config)# dhcp ipv4`	DHCP IPv4 プロファイル 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	**profile** *profile-name* **snoop** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4)# profile untrustedClientProfile snoop`	クライアント ポート用に信頼できない DHCP スヌーピング プロファイルを設定します。
ステップ 4	**relay information option** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4-snoop-profile)# relay information option`	DHCP サーバに転送される BOOTREQUEST メッセージに DHCP リレー情報オプション フィールドが挿入されるようにします。
ステップ 5	**relay information option allow-untrusted** 例： `RP/0/RSP0/CPU0:router(config-dhcpv4-snoop-profile)# relay information option allow-untrusted`	リレー情報オプションがすでにあり、かつ giaddr がゼロに設定されている BOOTREQUEST パケットを廃棄しないように DHCP IPv4 Relay を設定します。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		◦ **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。

DHCP スヌーピングの設定例

ここでは、次の設定例について説明します。

ブリッジ ドメインへの DHCP プロファイルの割り当て：例

次に、ブリッジ ドメインで DHCP スヌーピングをイネーブルにする例を示します。

```
l2vpn
 bridge group GRP1
  bridge-domain ISP1
   dhcp ipv4 profile untrustedClientProfile snoop
```

特定のブリッジ ポートでの DHCP スヌーピングのディセーブル化：例

次に、特定のブリッジ ポートで DHCP スヌーピングをディセーブルにする例を示します。

```
interface gigabitethernet 0/1/0/1
 dhcp ipv4 none
```

信頼できるブリッジ ポート用の DHCP プロファイルの設定：例

次に、信頼できるブリッジ ポート用に DHCP プロファイルを設定する例を示します。

```
dhcp ipv4 profile trustedServerProfile snoop
 trusted
```

ブリッジ ドメインでの信頼できないプロファイルの設定：例

次に、プロファイルをブリッジ ドメインにアタッチし、ブリッジ ポートでスヌーピングをディセーブルにする例を示します。

```
l2vpn
 bridge group GRP1
  bridge-domain ISP1
   dhcp ipv4 profile untrustedClientProfile snoop
    interface gigabitethernet 0/1/0/1
     dhcp ipv4 none
```

信頼できるブリッジ ポートの設定：例

次に、信頼できる DHCP スヌーピング プロファイルをブリッジ ポートに割り当てる例を示します。

```
l2vpn
 bridge group GRP1
  bridge-domain ISP1
   interface gigabitethernet 0/1/0/2
    dhcp ipv4 profile trustedServerProfile snoop
```

その他の参考資料

ここでは、Cisco IOS XR DHCP リレー エージェントおよび DHCP スヌーピング機能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
Cisco IOS XR DHCP コマンド	『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*DHCP Commands*」の章
スタートアップ資料	『*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 ASR 9000 Series Aggregation Services Router System Security Configuration Guide*』の「*Configuring AAA Services*」の章

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
—	MIB を特定してダウンロードするには、次の URL にある Cisco MIB Locator を使用し、[Cisco Access Products] メニューからプラットフォームを選択します。http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
RFC 2131	『*Dynamic Host Configuration Protocol*』

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

第 5 章

ホスト サービスとアプリケーションの実装

Cisco IOS XR ソフトウェアルータのホスト サービスとアプリケーション機能は主に、ネットワーク接続性およびパケットが宛先に達するまでにたどるルートをチェックし、ホスト名を IP アドレスに、または IP アドレスをホスト名にマッピングし、ルータと UNIX ワークステーションとの間でファイルを転送する目的で使用します。

（注） この章に記載されているホスト サービスとアプリケーションのコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』を参照してください。 この章で使用される他のコマンドの説明については、コマンド リファレンスのマスター索引を参照するか、またはオンラインで検索してください。

ホスト サービスとアプリケーションの実装の機能履歴

リリース	変更内容
リリース 3.7.2	この機能が導入されました。

ホスト サービスとアプリケーションの実装の前提条件

Cisco IOS XR ソフトウェアホスト サービスとアプリケーションを実装するには、次の前提条件を満たす必要があります。

- 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。

ホストサービスとアプリケーションの実装に関する情報

このドキュメントで説明する Cisco IOS XR ソフトウェアのホスト サービスとアプリケーションの機能を実装するには、次の概念を理解する必要があります。

ネットワーク接続性ツール

ネットワーク接続性ツールを使用すると、ネットワーク上のデバイスに対して traceroute と ping を実行して、デバイス接続性をチェックできます。

ping

ping コマンドは、デバイスのアクセシビリティのトラブルシューティングに広く使用されている方法です。 これは、2 つのインターネット制御メッセージ プロトコル（ICMP）クエリー メッセージ、ICMP エコー要求、および ICMP エコー応答を使用して、リモート ホストがアクティブであるかどうかを判断します。 **ping** コマンドでは、エコー応答を受信するまでにかかる時間も測定します。

ping コマンドは、最初に 1 つのアドレスに対してエコー要求パケットを送信し、応答を待機します。 ping が正常に完了するのは、エコー要求が宛先に届き、定義済みの時間内に宛先が ping の送信元にエコー応答（ホスト名が存続している）を返すことができる場合だけです。

bulk オプションが導入されたため、複数の宛先の到達可能性をチェックできるようになりました。宛先は、CLI から直接入力します。 このオプションは、ipv4 の宛先でのみサポートされます。

traceroute

ping コマンドを使用してデバイス間の接続性を検証できる場合は、**traceroute** コマンドを使用してパケットがリモート接続先までにたどるパスおよびルーティングに障害がある場所を検出できます。

traceroute コマンドは、各 ICMP "time-exceeded" メッセージの送信元を記録して、パケットが宛先に達するまでにたどったパスを示すことができます。 IP **traceroute** コマンドを使用すると、パケットがネットワーク経由でたどるパスをホップバイホップで特定できます。 このコマンドを実行すると、トラフィックが宛先に到達するまでに通過するルータなどのすべてのネットワーク層（レイヤ 3）デバイスが表示されます。

traceroute コマンドは、IP ヘッダーの存続可能時間（TTL）フィールドを使用して、ルータとサーバで特定のリターン メッセージが生成されるようにします。 **traceroute** コマンドは、TTL フィールドが 1 に設定されている宛先ホストにユーザ データグラム プロトコル（UDP）データグラムを

送信します。 ルータは 1 または 0 の TTL 値を検出すると、データグラムをドロップし、送信元に ICMP の time-exceeded メッセージを戻します。 traceroute 機能は、ICMP time-exceeded メッセージの送信元アドレス フィールドを調べ、最初のホップのアドレスを判別します。

ネクスト ホップを識別するために、**traceroute** コマンドは TTL 値が 2 の UDP パケットを送信します。 1 番めのルータは、TTL フィールドの値から 1 を差し引いて次のルータにデータグラムを送信します。 2 番めのルータは、TTL 値が 1 の UDP パケットを受け取り、データグラムを廃棄して、送信元に time-exceeded メッセージを戻します。 このように、データグラムが宛先ホストに到達するまで（または TTL の最大値に達するまで）TTL の値は増分され、処理が続けられます。

データグラムが宛先に到達したことを判断するために、**traceroute** コマンドはデータグラムの UDP 宛先ポートを宛先ホストが使用すると予測される非常に大きな値に設定します。 ホストは、この未知のポート番号を持つデータグラムを受信すると、送信元に ICMP port unreachable error メッセージを戻します。 このメッセージにより、宛先に到達したことを traceroute 機能に伝えます。

ドメイン サービス

Cisco IOS XR ソフトウェア ドメイン サービスは、Berkeley Standard Distribution（BSD）ドメイン リゾルバとして機能します。 ドメイン サービスは、Telnet などのアプリケーション、および **ping** や **traceroute** などのコマンドで使用されているホスト名とアドレスのマッピングのローカル キャッシュを維持します。 ローカル キャッシュにより、ホスト名からアドレスへの変換の速度が向上します。 ローカル キャッシュには、2 つのタイプのエントリが存在します。スタティックとダイナミックです。 **domain ipv4 host** コマンドまたは **domain ipv6 host** コマンドを使用して設定されるエントリはスタティック エントリとして追加され、ネーム サーバから届いたエントリはダイナミック エントリとして追加されます。

ネーム サーバは、World Wide Web（WWW）でネットワーク ノードの名前をアドレスに変換するために使用されます。 ネーム サーバは、DNS サーバから DNS プロトコルを使用して、ホスト名を IP アドレスにマッピングする分散データベースを維持します。 **domain name-server** コマンドを使用して、1 つ以上のネーム サーバを指定できます。

アプリケーションでホストの IP アドレスまたは IP アドレスのホスト名が必要になると、ドメイン サービスに対してリモート プロシージャ コール（RPC）が実行されます。 ドメイン サービスは、キャッシュ内で IP アドレスまたはホスト名を探し、エントリが見つからない場合にはネーム サーバに DNS クエリーを送信します。

ドメイン名要求を完了するために Cisco IOS XR ソフトウェアで使用されるデフォルト ドメイン名を指定できます。 単一のドメインまたはドメイン名のリストを指定することもできます。IP ホスト名にドメイン名が含まれていない場合には、ホスト テーブルに追加される前に指定のドメイン名が付加されます。 1 つまたは複数のドメイン名を指定するには、**domain name** コマンドまたは **domain list** コマンドを使用します。

TFTP サーバ

サーバとしてだけ機能するマシンをネットワークの各セグメントに配置するのは、コストがかかり、非効率的です。 しかし、すべてのセグメントにサーバがあるのではない場合、ネットワーク セグメントを超えたネットワークの操作によって相当の遅延が引き起こされることがあります。

ルータを TFTP サーバとして機能するように設定すると、ルータの通常の機能を使用しながらコストと遅延時間を削減できます。

一般に、TFTP サーバとして設定されたルータは、フラッシュ メモリから他のルータにシステム イメージまたはルータ コンフィギュレーション ファイルを提供します。 他のタイプのサービス要求に応答するようにルータを設定することもできます。

ファイル転送サービス

ファイル転送プロトコル（FTP）、Trivial File Transfer Protocol（TFTP）、リモート コピー プロトコル（RCP）の各クライアントは、ファイル システムまたはリソース マネージャとして実装されます。 たとえば、tftp:// で始まるパス名は、TFTP リソース マネージャによって処理されます。

ファイル システム インターフェイスは、URL を使用して、ファイルの場所を指定します。 URL は、WWW でファイルまたは場所を指定するのに広く使用されています。 ただし、Cisco ルータの URL には、ルータまたはリモート ファイル サーバ上のファイルの場所も指定されます。

ルータがクラッシュしたときは、ルータのメモリ内容全体のコピーを取得するのが便利です（これをコア ダンプと言います）。テクニカル サポート担当者が、クラッシュの原因を特定するのに使用します。 FTP、TFTP、または rcp を使用すると、コア ダンプをリモート サーバに保存できます。 コア ダンプの実行については、『*Cisco ASR 9000 Series Aggregation Services Router System Management Configuration Guide*』を参照してください。

RCP

リモートコピープロトコル（RCP）のコマンドは、リモート システム上のリモート シェル（rsh）サーバ（またはデーモン）を利用します。 RCP を使用してファイルをコピーする場合、TFTP と異なり、ファイル配布用のサーバを作成する必要はありません。 必要なのは、rsh をサポートするサーバへのアクセスだけです。 ある場所から別の場所にファイルをコピーするため、コピー元のファイルに対する読み取り権限とコピー先のディレクトリに対する書き込み権限が必要です。 コピー先ファイルが存在しない場合は、rcp により作成されます。

シスコの rcp 実装は UNIX の rcp 実装（ネットワーク上のシステム間でファイルをコピー）の関数をエミュレートしたものですが、シスコのコマンド構文は UNIX の rcp コマンド構文とは異なります。 Cisco IOS XR ソフトウェア には、rcp をトランスポート メカニズムとして使用するコピー コマンドのセットが用意されています。 これらの **rcp copy** コマンドは、Cisco IOS XR ソフトウェアの TFTP copy コマンドに類似していますが、高速で信頼性の高いデータ配信を実現する代替方法を備えている点が異なります。 これらの改善は、rcp のトランスポート メカニズムがコネクション型の TCP/IP スタック上に構築されており、これを使用しているために可能になりました。 rcp コマンドを使用して、ルータからネットワーク サーバなどへシステム イメージおよびコンフィギュレーション ファイルをコピーできます。

FTP

ファイル転送プロトコル（FTP）は、TCP/IP プロトコル スタックの一部であり、ネットワーク ノード間でファイルを転送するのに使用します。 FTP は、RFC 959 で定義されています。

TFTP

Trivial File Transfer Protocol（TFTP）は FTP の簡易版で、ネットワークを介して 1 つのコンピュータから別のコンピュータにファイルを転送できます。通常は、クライアント認証（ユーザ名とパスワードなど）を使用しません。

Cisco inetd

Cisco インターネット サービス プロセス デーモン（Cinetd）は、システムのブート後にシステム マネージャによって開始されるマルチスレッド サーバ プロセスです。 Cinetd は、Telnet サービスや TFTP サービスなどのインターネット サービスをリッスンします。 Cinetd が特定のサービスをリッスンするかどうかは、ルータ コンフィギュレーションによって異なります。 たとえば、**tftp server** コマンドを入力すると、Cinetd は TFTP サービスのリッスンを開始します。 要求が届くと、Cinetd はサービスに関連付けられたサーバ プログラムを実行します。

Telnet

Telnet をイネーブルにすると、ネットワーキング デバイスで着信 Telnet 接続が許可されます。

ホスト サービスとアプリケーションを実装する方法

ここでは、次の手順について説明します。

ネットワーク接続の確認

基本的なネットワーク接続性の診断を支援する手段として、多くのネットワーク プロトコルがエコープロトコルをサポートしています。プロトコルでは、宛先ホストに特殊なデータグラムを送信し、そのホストからの応答データグラムを待ちます。このエコープロトコルからの結果は、ホストに至るパスの信頼性、パスの遅延、およびホストに到達できるのか、ホストが機能しているのかを評価するのに役立ちます。

手順の概要

1. **ping** [**ipv4** | **ipv6** | **vrf** *vrf-name*] [*host-name* | *ip-address*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**ping** [**ipv4** | **ipv6** | **vrf** *vrf-name*] [*host-name* | *ip-address*]	接続性のテストに使用される ping ツールを開始します。

	コマンドまたはアクション	目的
	例： `RP/0/RSP0/CPU0:router# ping`	（注） **ping** コマンドと同じ回線上にあるホスト名または IP アドレスを入力しないと、ターゲット IP アドレスと、他のコマンドパラメータもいくつか指定するように求められます。ターゲットの IP アドレスを指定すると、残りのパラメータに対する代替値を指定できます。あるいは表示された各パラメータのデフォルト値を受け入れることも可能です。

複数の宛先に対するネットワーク接続性のチェック

bulk オプションを使用すると、複数の宛先への到達可能性をチェックできます。 宛先は、CLI から直接入力します。 このオプションは、ipv4 の宛先でのみサポートされます。

手順の概要

1. **ping bulk ipv4** [**input cli** { **batch** | **inline** }]
2. [**vrf** *vrf-name*] [*host-name* | *ip-address*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**ping bulk ipv4** [**input cli** { **batch**	**inline** }] 例： `RP/0/RSP0/CPU0:router# ping bulk ipv4 input cli`	接続性のテストに使用される ping ツールを開始します。

ステップ 2	[**vrf** *vrf-name*] [*host-name*	*ip-address*]
		例：
		`Please enter input via CLI with one destination per line:`
		`vrf myvrf1 1.1.1.1`
		`vrf myvrf2 2.2.2.2`
		`vrf myvrf1 myvrf1.cisco.com`
		`vrf myvrf2 myvrf2.cisco.com`
		
		`Starting pings...`
		`Type escape sequence to abort.`
		`Sending 1, 100-byte ICMP Echos to 1.1.1.1, vrf is myvrf1:`
		`!`
		`Success rate is 100 percent (1/1), round-trip min/avg/max = 1/1/1 ms`
		`Sending 2, 100-byte ICMP Echos to 2.2.2.2, vrf is myvrf2:`
		`!!`
			[Enter] ボタンを押し、宛先アドレスを 1 行に 1 つずつ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

	`Success rate is 100 percent (2/2), round-trip min/avg/max = 1/1/1 ms`
	`Sending 1, 100-byte ICMP Echos to 1.1.1.1, vrf is myvrf1:`
	`!`
	`Success rate is 100 percent (1/1), round-trip min/avg/max = 1/4/1 ms`
	`Sending 2, 100-byte ICMP Echos to 2.2.2.2, vrf is myvrf2:`
	`!!`
	`Success rate is 100 percent (2/2), round-trip min/avg/max = 1/3/1 ms`	

パケット ルートのチェック

traceroute コマンドを使用すると、パケットが宛先に移動するときに実際にたどるルートをトレースできます。

手順の概要

1. **traceroute** [**ipv4** | **ipv6** | **vrf** *vrf-name*] [*host-name* | *ip-address*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**traceroute** [**ipv4**	**ipv6**	**vrf** *vrf-name*] [*host-name*	*ip-address*] 例： `RP/0/RSP0/CPU0:router# traceroute`	ネットワーク経由でパケット ルートをトレースします。 （注） **traceroute** コマンドと同じ回線上にあるホスト名または IP アドレスを入力しないと、ターゲット IP アドレスと、他のコマンド パラメータもいくつか指定するように求められます。 ターゲットの IP アドレスを指定すると、残りのパラメータに対する代替値を指定できます。あるいは表示された各パラメータのデフォルト値を受け入れることも可能です。

ドメイン サービスの設定

このタスクを実行すると、ドメイン サービスを設定できます。

はじめる前に

デフォルトでは、DNSによるホスト名からアドレスへの変換がイネーブルになっています。**domain lookup disable** コマンドを使用してホスト名からアドレスへの変換をディセーブルにしていた場合は、**no domain lookup disable** コマンドを使用して変換を再びイネーブルにします。 **domain lookup**

disable コマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』を参照してください。

手順の概要

1. **configure**
2. 次のいずれかを実行します。
   - **domain name** *domain-name*
   - または
   - **domain list** *domain-name*
3. **domain name-server** *server-address*
4. **domain** {**ipv4** | **ipv6**} **host** *host-name* {*ipv4address* | *ipv6address*}
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	次のいずれかを実行します。
	• **domain name** *domain-name*
	• または
	• **domain list** *domain-name*
	
	例：
	`RP/0/RSP0/CPU0:router(config)# domain name cisco.com`
	`or`
	`RP/0/RSP0/CPU0:router(config)# domain list domain1.com`
		修飾されていないホスト名を完全なホスト名にするために使用されるデフォルト ドメイン名を定義します。

	コマンドまたはアクション	目的
ステップ 3	**domain name-server** *server-address* 例： `RP/0/RSP0/CPU0:router(config)# domain name-server 192.168.1.111`	名前/アドレス解決に使用するネーム サーバ（名前情報を提供するホスト）を指定します。 （注） 最大6つのアドレスを入力できますが、各コマンドでは 1 つずつしか指定できません。
ステップ 4	**domain** {**ipv4** | **ipv6**} **host** *host-name* {*ipv4address* | *ipv6address*} 例： `RP/0/RSP0/CPU0:router(config)# domain ipv4 host1 192.168.7.18`	（任意）IPv4 または IPv6 を使用して、ホスト キャッシュにホスト名とアドレスのスタティックなマッピングを定義します。 （注） ホスト名 1 つに、最大 8 つの関連するアドレスをバインドできます。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

TFTP サーバとしてのルータの設定

このタスクを実行すると、ルータを TFTP サーバとして設定できます。これにより、TFTP クライアントとして機能する他のデバイスは、slot0: や /tmp などの特定のディレクトリ（TFTP ホーム ディレクトリ）の下にあるファイルをルータに対して読み書きできます。

（注） セキュリティを確保するため、ファイルがすでに存在していないと、TFTP サーバでは書き込み要求を正常に完了できません。

はじめる前に

TFTP 機能の実装前に、サーバとクライアント ルータは互いに到達可能である必要があります。**ping** コマンドを使用してサーバとクライアント ルータ間の接続を（どちらの方向でも）テストして、この接続を検証します。

手順の概要

1. **configure**
2. **tftp** {**ipv4** | **ipv6**} **server** {**homedir** *tftp-home-directory*} {**max-servers** *number*} [**access-list** *name*]
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
4. show cinetd services

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	**tftp** {**ipv4** | **ipv6**} **server** {**homedir** *tftp-home-directory*} {**max-servers** *number*} [**access-list** *name*] 例： `RP/0/RSP0/CPU0:router(config)# tftp ipv4 server access-list listA homedir disk0`	次のものを指定します。 • IPv4 または IPv6 アドレス プレフィックス（必須） • ホーム ディレクトリ（必須） • 同時 TFTP サーバの最大数（必須） • 関連付けられたアクセス リストの名前（任意）

	コマンドまたはアクション	目的

ステップ 3	次のいずれかのコマンドを使用します。
	• **end**
	• **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 4	show cinetd services 例： `RP/0/RSP0/CPU0:router# show cinetd services`	各プロセスのネットワーク サービスを表示します。 TFTP サーバが設定されている場合、service 列には TFTP と表示されます。

rcp 接続を使用するためのルータの設定

このタスクを実行すると、rcp を使用するようにルータを設定できます。

はじめる前に

rcp コピー要求が正常に実行されるためには、ネットワーク サーバ上でリモート ユーザ名のアカウントが定義されている必要があります。

サーバに対して読み書きする場合は、ルータ上のユーザからの rcp 読み書き要求を受け入れるように rcp サーバが正しく設定されている必要があります。 UNIX システムの場合は、rcp サーバ上のリモート ユーザの hosts ファイルに対しエントリを追加する必要があります。

手順の概要

1. **configure**
2. **rcp client username** *username*
3. **rcp client source-interface** *type interface-path-id*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**rcp client username** *username* 例： `RP/0/RSP0/CPU0:router(config)# rcp client username netadmin1`	rcp サーバ上のリモート ユーザの名前を指定します。 この名前は、rcp を使用したリモート コピーを要求するときに使用されます。 rcp サーバにディレクトリ構造が存在する場合、コピー対象のすべてのファイルおよびイメージは、リモート ユーザのアカウント内のサーバ ディレクトリに該当する場所で検索されるか書き込まれます。
ステップ 3	**rcp client source-interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# rcp client source-interface gigabitethernet 1/0/2/1`	インターフェイスの IP アドレスをすべての rcp 接続の送信元として設定します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		◦**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。

トラブルシューティングのヒント
rcp を使用してコピー元からコピー先にファイルをコピーするときは、次のパス形式を使用します。

copy rcp:*//username@*{*hostname* | *ipaddress*}*/directory-path/pie-name target-device*

IPv6 rcp サーバを使用するときは、次のパス形式を使用します。

copy rcp:*//username@*`[ipv6-address]`*/directory-path/pie-name*

rcp プロトコルで **copy** コマンドを使用する方法の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router System Management Command Reference*』の **copy** コマンドを参照してください。

FTP 接続使用時のルータ設定

このタスクを実行すると、FTP 接続を使用してネットワーク上のシステム間でファイルを転送するようにルータを設定できます。 Cisco ASR 9000 シリーズ ルータに実装された FTP により、次の FTP 特性を設定できます。

- パッシブ モード FTP
- パスワード
- IP アドレス

手順の概要

1. **configure**
2. **ftp client passive**
3. **ftp client anonymous-password** *password*
4. **ftp client source-interface** *type interface-path-id*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**ftp client passive** 例： `RP/0/RSP0/CPU0:router(config)# ftp client passive`	パッシブ FTP 接続のみを使用できます。
ステップ 3	**ftp client anonymous-password** *password* 例： `RP/0/RSP0/CPU0:router(config)# ftp client anonymous-password xxxx`	匿名ユーザ用のパスワードを指定します。
ステップ 4	**ftp client source-interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# ftp client source-interface gigabitethernet 0/1/2/1`	FTP 接続の発信元 IP アドレスを指定します。
ステップ 5	次のいずれかのコマンドを使用します。 • **end** • **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]:`

	コマンドまたはアクション	目的

	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		◦ **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。

トラブルシューティングのヒント

FTP を使用してコピー元からコピー先にファイルをコピーするときは、次のパス形式を使用します。

copy ftp:*//username:password@{hostname | ipaddress}/directory-path/pie-name target-device*

IPv6 FTP サーバを使用するときは、次のパス形式を使用します。

copy ftp:*//username*:*password@*`[ipv6-address]`*/directory-path/pie-name*

ユーザ名、パスワード、ホスト名などに安全でない文字または予約された文字を含める場合は、エンコードする必要があります（RFC 1738）。

安全でない文字は次のとおりです。

`"<", ">", "#", "%" "{", "}", "|", "□", "~", "[", "]", and "\"`

予約された文字は次のとおりです。

`":", "/" "?", ":", "@", and "&"`

directory-path は、ユーザのホームディレクトリからの相対パスです。 絶対パスを指定するには、スラッシュ（/）を %2f としてエンコードする必要があります。 次に例を示します。

```
ftp://user:password@hostname/%2fTFTPboot/directory/pie-name
```

FTP プロトコルで **copy** コマンドを使用する方法の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router System Management Command Reference*』の **copy** コマンドを参照してください。

TFTP 接続使用時のルータ設定

このタスクを実行すると、TFTP 接続を使用するようにルータを設定できます。 TFTP 接続用の送信元 IP アドレスを指定する必要があります。

手順の概要

1. **configure**
2. **tftp client source-interface** *type*
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**tftp client source-interface** *type* 例： `RP/0/RSP0/CPU0:router(config)# tftp client source-interface gigabitethernet 1/0/2/1`	TFTP 接続用の送信元 IP アドレスを指定します。
ステップ 3	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	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。

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	◦ **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。

トラブルシューティングのヒント
TFTP を使用してコピー元からコピー先にファイルをコピーするときは、次のパス形式を使用します。

copy tftp:`//{`*hostname* | *ipaddress*`}/`*directory-path*`/`*pie-name target-device*
IPv6 TFTP サーバを使用するときは、次のパス形式を使用します。

copy tftp:`//[ipv6-address]/`*directory-path*`/`*pie-name*
TFTP プロトコルで **copy** コマンドを使用する方法の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router System Management Command Reference*』の **copy** コマンドを参照してください。

Telnet サービスの設定

このタスクを実行すると、Telnet サービスを設定できます。

手順の概要

1. **configure**
2. **telnet** [**ipv4** | **ipv6** | **vrf** *vrf-name*] **server max-servers 1**
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**telnet** [**ipv4** | **ipv6** | **vrf** *vrf-name*] **server max-servers 1** 例： `RP/0/RSP0/CPU0:router(config)# telnet ipv4 server max-servers 1`	ルータ上で着信 Telnet サーバを 1 つイネーブルにします。 （注） このコマンドは、ルータへの着信 Telnet 接続にのみ作用します。

ステップ 3	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

ホスト サービスとアプリケーションの実装の設定例

ここでは、次の設定例について説明します。

ネットワーク接続の確認：例

次に、ルータ A イーサネット 0 インターフェイスを送信元とし、ルータ B イーサネット インターフェイスを宛先とする拡張 **ping** コマンドの例を示します。 この ping が成功する場合、ルーティング上の問題がないことを示します。ルータ A はルータ B のイーサネットに到達する方法を認識し、ルータ B はルータ A のイーサネットに到達する方法を認識しています。 また、どちらのホストにも、デフォルト ゲートウェイが正しく設定されています。

ルータ A からの拡張 **ping** コマンドが失敗する場合、ルーティング上の問題があることを意味します。 3 つのルータのいずれでもルーティングに関する問題が発生する可能性があります。ルータ A には、ルータ B のイーサネットのサブネットまたはルータ C とルータ B との間にあるサブネットに至るルートが存在しない可能性があります。ルータ B には、ルータ A のサブネットのサブネットまたはルータ C とルータ A との間にあるサブネットに至るルートが存在しない可能性があります。ルータ C には、ルータ A またはルータ B のイーサネット セグメントのサブネットに至るルートが存在しない可能性があります。 ルーティングに関する問題を修正してから、ホスト 1 からホスト 2 への ping を実行する必要があります。 ホスト 1 からホスト 2 への ping を実行できない場合は、両方のホストのデフォルト ゲートウェイを確認してください。 ルータ A のイーサネットとルータ B のイーサネットの間の接続は、拡張 **ping** コマンドを使用してチェックします。

ルータ A からルータ B のイーサネット インターフェイスへの通常の ping では、ping パケットの送信元アドレスは、発信インターフェイスのアドレス、つまりシリアル 0 インターフェイスのアドレス（172.31.20.1）になります。 ルータ B が ping パケットに応答するとき、送信元アドレス（つまり、172.31.20.1）に応答します。 このように、ルータ A のシリアル 0 インターフェイス（172.31.20.1）とルータ B のイーサネット インターフェイス（192.168.40.1）の間の接続だけがテストされます。

ルータ A のイーサネット 0（172.16.23.2）とルータ B のイーサネット 0（192.168.40.1）との間の接続をテストするには、拡張 **ping** コマンドを使用します。 拡張 **ping** コマンドには、**ping** パケットの送信元アドレスを指定するオプションがあります。

この例では、拡張 **ping** コマンドは 10.0.0.2 と 10.0.0.1 という 2 つの IP アドレス間の IP 接続を確認します。

```
ping

Protocol [ip]:
Target IP address: 10.0.0.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands? [no]: yes
Source address or interface: 10.0.0.2
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]: yes
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes? [no]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.25.58.21, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/11/49 ms
```

traceroute コマンドは、パケットがリモート接続先に至るまでにたどるパスおよびルーティングに障害がある場所を検出するために使用されます。 **traceroute** コマンドは、2 つの IP アドレス間のパスを示すものであり、パスの問題は示しません。

```
traceroute

Protocol [ip]:
Target IP address: ena-view3
Source address: 10.0.58.29
Numeric display? [no]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]:

Type escape sequence to abort.
Tracing the route to 171.71.164.199

1 sjc-jpollock-vpn.cisco.com (10.25.0.1) 30 msec 4 msec 4 msec
 2  15lab-vlan525-gw1.cisco.com (172.19.72.2) 7 msec  5 msec  5 msec
 3  sjc15-00lab-gw1.cisco.com (172.24.114.33) 5 msec  6 msec  6 msec
 4  sjc5-lab4-gw1.cisco.com (172.24.114.89) 5 msec  5 msec  5 msec
 5  sjc5-sbb4-gw1.cisco.com (171.71.241.162) 5 msec  6 msec  6 msec
 6  sjc5-dc5-gw1.cisco.com (171.71.241.10) 6 msec  6 msec  5 msec
 7  sjc5-dc1-gw1.cisco.com (171.71.243.2) 7 msec  8 msec  8 msec
 8 ena-view3.cisco.com (171.71.164.199) 6 msec * 8 msec
```

ドメイン サービスの設定：例

次に、ルータにドメイン サービスを設定する例を示します。

ドメイン ホストの定義

```
configure

domain ipv4 host host1 192.168.7.18
domain ipv4 host bost2 10.2.0.2 192.168.7.33
```

ドメイン名の定義

```
configure
domain name cisco.com
```

ネーム サーバのアドレスの指定

```
configure

domain name-server 192.168.1.111
domain name-server 192.168.1.2
```

rcp、FTP、または TFTP 接続を使用するためのルータの設定：例

次に、rcp、FTP、または TFTP 接続を使用するようにルータを設定する例を示します。

rcp の使用

```
configure

rcp client username netadmin1
rcp client source-interface gigabitethernet 1/0/2/1
```

FTP の使用

```
configure

ftp client passive
ftp client anonymous-password xxxx
ftp client source-interface gigabitethernet 0/1/2/1
```

TFTP の使用

```
configure
tftp client source-interface gigabitethernet 1/0/2/1
```

その他の参考資料

ここでは、Cisco ASR 9000 シリーズ ルータでのホスト サービスおよびアドレス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
ホストサービスとアプリケーションのコマンド	『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*Host Services and Applications Commands*」の章

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 IOS XR ソフトウェアを使用している MIB を特定してダウンロードするには、次の URL にある Cisco MIB Locator を使用し、[Cisco Access Products] メニューからプラットフォームを選択します。http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
RFC-959	『File Transfer Protocol』
RFC-1738 および RFC-2732	『Uniform Resource Locators (URL)』
RFC-783	『Trivial File Transfer Protocol』

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

第 6 章

HSRP の実装

ホットスタンバイ ルータ プロトコル（HSRP）は、ファーストホップ IP ルータで透過的にフェールオーバーが発生する事態を考慮するように設計された IP ルーティング冗長プロトコルです。ネットワーク上のホストからの IP トラフィックをルーティングするときに単一ルータの可用性に依存しないため、HSRP では、高度なネットワーク可用性が提供されます。ルータのグループで HSRP を使用して、アクティブ ルータとスタンバイ ルータを選択します（アクティブ ルータとは、パケット転送用に選択されているルータのことです。スタンバイ ルータとは、アクティブ ルータで障害が発生したときや、プリセット条件が満たされたときに、ルーティング処理を引き継ぐルータのことです）。

HSRP の実装の機能履歴

リリース 3.7.2	この機能が導入されました。
リリース 3.9.0	次の機能に対するサポートが追加されました。 • HSRP 用の BFD。 • HSRP 用のホット リスタート。
リリース 4.2.0	HSRP 用のマルチ グループ オプティマイゼーション（MGO）機能が追加されました。

HSRP の実装の前提条件

- 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。

HSRP の実装の制約事項

HSRP は、イーサネット インターフェイス、イーサネット サブインターフェイス、およびイーサネット リンク バンドルでサポートされています。

HSRP の実装に関する情報

Cisco IOS XR ソフトウェアのソフトウェアに HSRP を実装するには、次の概念を理解する必要があります。

HSRP の概要

HSRP は、ルータ ディスカバリ プロトコル（Internet Control Message Protocol [ICMP] Router Discovery Protocol [IRDP] など）をサポートしないホスト、および選択したルータがリロードしたときやルータの電源が失われたときに新しいルータに切り替えることができないホストに便利です。 また、既存の TCP セッションはフェールオーバーが発生しても存続するため、このプロトコルでは IP トラフィックをルーティングするためにネクスト ホップを動的に選択するホストの回復をさらに透過的に実行できます。

HSRP をネットワーク セグメントに設定すると、HSRP が動作するルータのグループで仮想 MAC アドレスと IP アドレスを共有できるようになります。 この HSRP ルータ グループのアドレスが仮想 *IP* アドレスと呼ばれます。 このようなデバイスの 1 つが、アクティブ ルータとしてプロトコルによって選択されます。 アクティブ ルータは、グループの MAC アドレス宛のパケットを受信してルーティングします。 *n* 台のルータで HSRP が稼動している場合、*n* + 1 個の IP アドレスおよび MAC アドレスが割り当てられます。

HSRP が指定アクティブ ルータの障害を検出すると、選択されているスタンバイ ルータが HSRP グループの MAC アドレスと IP アドレスの制御を引き継ぎます。 この時点で新しいスタンバイ ルータも選択されます。

HSRP を実行しているルータは、ユーザ データグラム プロトコル（UDP）ベースのマルチキャスト hello パケットを送受信して、ルータの障害を検出したり、アクティブ ルータとスタンバイ ルータを指定したりします。

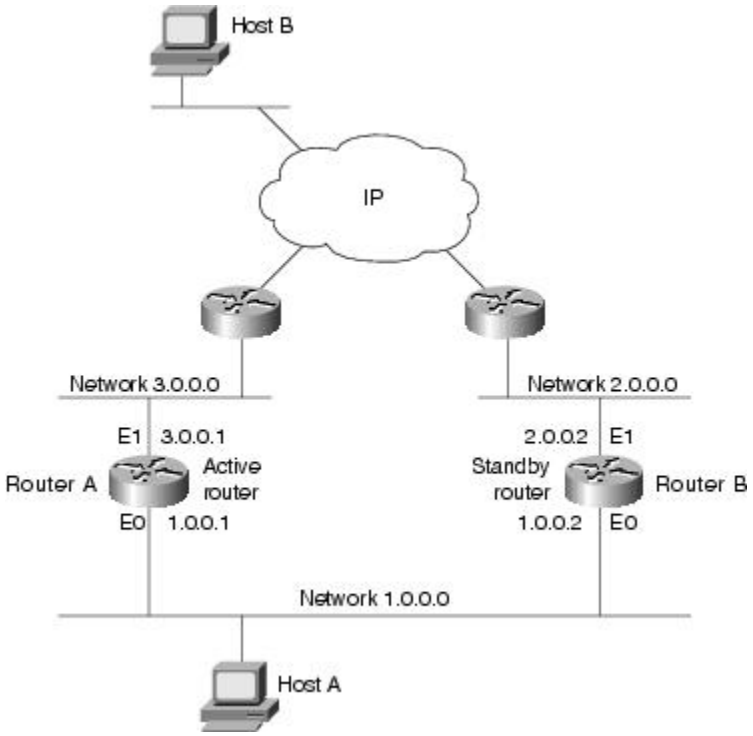
HSRP グループ

HSRP グループは、HSRP を実行し、かつ互いにホットスタンバイ サービスを提供するように設定されている複数のルータで構成されています。 HSRP は、プライオリティ スキームを使用して、HSRP によって設定されたどのルータをデフォルトのアクティブ ルータにするかを決定します。 ルータをアクティブ ルータとして設定するには、他のすべての HSRP 設定済みルータのプライオリティよりも高いプライオリティをそのルータに割り当てます。 デフォルトのプライオリティは 100 です。したがって、100 よりも高いプライオリティを持つルータを 1 つだけ設定した場合、そのルータがデフォルトのアクティブ ルータになります。

HSRP は、HSRP グループ間でプライオリティをアドバタイズするマルチキャスト メッセージを交換することによって機能します。 アクティブ ルータが設定された時間内に hello メッセージを送信できなかった場合は、最高のプライオリティのスタンバイ ルータがアクティブ ルータになります。 このようにパケット転送機能が別のルータに移行しても、ネットワークのいずれのホストにもまったく影響はありません。

図 4：HSRP グループとして設定されたルータ，（163 ページ） に、単一の HSRP グループのメンバとして設定されたルータを示します。

図 *4*：*HSRP* グループとして設定されたルータ



ネットワーク上のホストはすべて、仮想ルータの IP アドレス（この場合 1.0.0.3）をデフォルト ゲートウェイとして使用するように設定されています。

1 つのルータ インターフェイスを複数の HSRP グループに属するように設定することもできます。図 5：複数の HSRP グループのメンバとして設定されたルータ，（164 ページ） に、複数の HSRP グループのメンバとして設定されたルータを示します。

図 *5*：複数の *HSRP* グループのメンバとして設定されたルータ

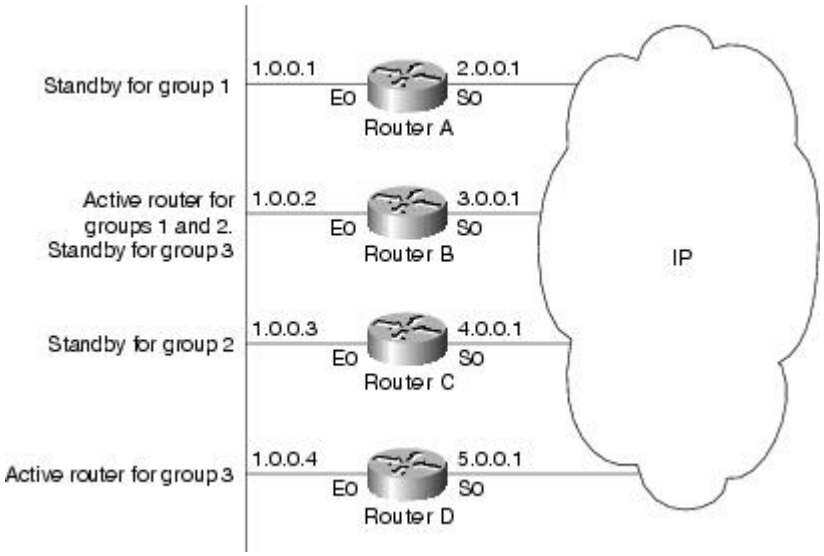


図 5：複数の HSRP グループのメンバとして設定されたルータ，（164 ページ） では、ルータ A のイーサネット インターフェイス 0 は、グループ 1 に属します。 ルータ B のイーサネット インターフェイス 0 は、グループ 1、2、および 3 に属します。 ルータ C のイーサネット インターフェイス 0 は、グループ 2、ルータ D のイーサネット インターフェイス 0 はグループ 3 に属します。 グループを作成するときは、部門の編成に従うことをお勧めします。 この場合、グループ 1 はエンジニアリング部門、グループ 2 は製造部門、グループ 3 は財務部門をサポートします。

ルータ B は、グループ 1 と 2 のアクティブ ルータ、およびグループ 3 のスタンバイ ルータとして設定されています。 ルータ D は、グループ 3 のアクティブ ルータとして設定されています。 何らかの理由でルータ D で障害が発生すると、ルータ B がルータ D のパケット転送機能を引き継ぐため、財務部門のユーザは引き続き他のサブネット上のデータにアクセスできます。

（注） サブインターフェイスごとに異なる仮想 MAC アドレス（VMAC）が必要になります。 VMAC は、グループ ID に基づいて決定されます。 このため、VMAC を明示的に設定する場合を除いて、設定するサブインターフェイスごとに固有のグループ ID が必要です。

HSRP と ARP

HSRP グループのルータは、アクティブになると、仮想 IP アドレスと仮想 MAC アドレスが含まれている ARP 応答を数多く送信します。 このような ARP 応答は、スイッチおよびラーニング ブリッジが自身のポートと MAC のマッピングを更新するのに役立ちます。 このような ARP 応答により、（事前に割り当てられた MAC アドレスまたは機能アドレスではなく）インターフェイスのバーンドイン アドレスを仮想 MAC アドレスとして使用するようにルータを設定できます。これは、仮想 IP アドレスの ARP エントリを更新するための手段となります。 インターフェイスがアップ状態になったときにそのインターフェイス IP アドレスを特定するために送信される Gratuitous ARP 応答と異なり、HSRP ルータ ARP 応答パケットはパケット ヘッダーで仮想 MAC アドレスを伝送します。 IP アドレスおよびメディア アドレスの ARP データ フィールドには、仮想 IP アドレスおよび仮想 MAC アドレスが含まれています。

プリエンプション

HSRPプリエンプション機能を使用すると、プライオリティの最も高いルータがただちにアクティブルータになることができます。プライオリティはまず設定したプライオリティ値に従って決定され、次にIPアドレスに従って決定されます。 どちらの場合も、値の大きい方がプライオリティが高くなります。

プライオリティの高いルータが、プライオリティの低いルータをプリエンプション処理すると、coupメッセージを送信します。プライオリティの低いアクティブルータが、プライオリティの高いアクティブ ルータから coup メッセージまたは hello メッセージを受信すると、スピーク状態に変わり、resign メッセージを送信します。

ICMP リダイレクト メッセージ

ICMP は、エラーをレポートするためのメッセージ パケットや IP 処理に関連する他の情報を提供する、ネットワーク層インターネット プロトコルです。 ICMP は多くの診断機能を備えており、ホストへのエラー パケットの送信およびリダイレクトが可能です。 HSRP を実行しているときは、HSRP グループに属するルータのインターフェイス（または実際の）MAC アドレスをホストが検出しないようにすることが重要です。 ICMP によってホストがルータの実際の MAC アドレスへリダイレクトされて、そのルータに障害が発生した場合、ホストからのパケットは消失します。

HSRPが設定されたインターフェイスでは、ICMPリダイレクトメッセージが自動的にイネーブルになります。 この機能は、ネクスト ホップ IP アドレスが HSRP 仮想 IP アドレスに変更されることのある HSRP で発信 ICMP リダイレクト メッセージをフィルタリングすることによって効果を発揮します。

ICMP リダイレクトをサポートするために、HSRP 経由で redirect メッセージがフィルタリングされます。これにより、ネクストホップ IP アドレスが HSRP 仮想アドレスに変更されます。 HSRP リダイレクトが有効になっていると、HSRPが動作するICMPインターフェイスはこのフィルタリングを行います。 HSRP は、アドバタイズメントを送信し、実 IP アドレスと仮想 IP アドレスの

マッピングを維持してリダイレクトのフィルタリングを実行することにより、すべてのHSRPルータの状況を把握します。

HSRP の実装方法

ここでは、次のタスクの手順を示します。

HSRP のイネーブル化

hsrp ipv4 コマンドは、設定済みのインターフェイスでHSRP をアクティブにします。 IP アドレスを指定した場合は、IP アドレスがホットスタンバイ グループの指定アドレスとして使用されます。 IP アドレスが指定されていない場合は、仮想アドレスがアクティブ ルータから学習されます。 HSRP が指定ルータを選択できるようにするには、ホットスタンバイ グループ内の少なくとも 1 つのルータに指定アドレスを指定しておくか、またはルータが指定アドレスを学習する必要があります。アクティブルータ上の指定アドレスを設定すると、常に現在使用されている指定アドレスが上書きされます。

手順の概要

1. **configure**
2. **router hsrp**
3. **interface** type interface-path-id
4. **hsrp** [*group-number*] **ipv4** [ip-address [secondary]]
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router hsrp** 例： `RP/0/RSP0/CPU0:router(config)# router hsrp`	HSRP コンフィギュレーション モード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** type interface-path-id 例： `RP/0/RSP0/CPU0:router(config-hsrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスでHSRPインターフェイス コンフィギュレーション モードをイネーブルにします。
ステップ 4	**hsrp** [*group-number*] **ipv4** [ip-address [secondary]] 例： `RP/0/RSP0/CPU0:router(config-hsrp)# hsrp 1 ipv4`	設定済みのインターフェイスで HSRP をアクティブにします。 • IP アドレスを指定した場合は、IP アドレスがホットスタンバイ グループの指定アドレスとして使用されます。 IP アドレスが指定されていない場合は、仮想アドレスがアクティブ ルータから学習されます。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

HSRP グループの属性の設定

ローカル ルータが HSRP に関与する仕組みに影響を与える他のホットスタンバイ グループ属性を設定するには、必要に応じてインターフェイス コンフィギュレーション モードで次の手順を使用します。

手順の概要

1. **configure**
2. **router hsrp**
3. **interface** type interface-path-id
4. **hsrp** [*group-number*] **priority** *priority*
5. **hsrp** [*group-number*] **track** *type* **instance** [*priority-decrement*]
6. **hsrp** [*group-number*] **preempt** [**delay** *seconds*]
7. **hsrp** [*group-number*] **authentication** *string*
8. **hsrp use-bia**
9. **hsrp** [*group-number*] **mac-address** *address*
10. 次のいずれかのコマンドを使用します。
    - **end**
    - **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	**router hsrp** 例： `RP/0/RSP0/CPU0:router(config)# router hsrp`	HSRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** type interface-path-id 例： `RP/0/RSP0/CPU0:router(config-hsrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスで HSRP インターフェイス コンフィギュレーション モードをイネーブルにします。
ステップ 4	**hsrp** [*group-number*] **priority** *priority* 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# hsrp priority 100`	（任意）HSRP プライオリティを設定します。 • *group-number* を指定しないと、設定はルータ上のすべての HSRP グループに適用されます。 • 割り当てられたプライオリティは、アクティブ ルータとスタンバイ ルータを選択するために使用されます。 プリエンプションがイネーブルである場合は、プライオリティが最高のルータが指定されたアクティブ ルータになります。 プライオリティ

	コマンドまたはアクション	目的
		が等しい場合、プライマリ IP アドレスが比較され、大きい IP アドレスが優先されます。 • インターフェイスが **hsrp track** コマンドによって設定されている場合、デバイス上の別のインターフェイスがダウンすると、デバイスのプライオリティが動的に変更されることもあります。 • **hsrp** [*group-number*] **preempt** コマンドを使用してプリエンプションをイネーブルにしていない場合、ルータは他の HSRP ルータよりもプライオリティが高い場合でもアクティブにならないことがあります。 • デフォルトの HSRP プライオリティ値を復元するには、**no hsrp** コマンドを使用します。
ステップ 5	**hsrp** [*group-number*] **track** *type* **instance** [*priority-decrement*] 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# hsrp track TenGigE 0/3/0/1`	（任意）他のインターフェイスの可用性に基づいてホットスタンバイ プライオリティが変わるように、インターフェイスを設定します。 • *group-number* を指定しないと、設定はルータ上のすべての HSRP グループに適用されます。 • トラッキング対象のインターフェイスがダウンすると、ホットスタンバイ プライオリティが 10 だけ減少します。インターフェイスがトラッキングされていなければ、ステートが変化した場合でもホット スタンバイ プライオリティに影響することはありません。ホットスタンバイ用に設定されたインターフェイスごとに、トラッキングするインターフェイスのリストを個別に設定できます。 • オプションの *priority-decrement* 引数には、トラッキング対象のインターフェイスがダウンした場合にホット スタンバイ プライオリティをどれだけ減らすかを指定します。トラッキング対象のインターフェイスが再びアップ状態になると、プライオリティは同じ値だけ段階的に増えていきます。 • トラッキング対象の複数のインターフェイスがダウンした場合、*priority-decrement* 引数が設定されていれば、設定されているプライオリティの減分値が累積されます。トラッキング対象のインターフェイスがダウンし、どのオブジェクトにもプライオリティの減分値が設定されていなければ、デフォルトの減分値は 10 で、累積されます。 • 常に最適なルータを使用してパケットが転送されるようにするには、グループ内のすべてのルータ上でこのコマンドとともに

	コマンドまたはアクション	目的
		hsrp preempt コマンドを使用する必要があります。 **hsrp preempt** コマンドを使用しないと、他の HSRP ルータの現在のプライオリティに関係なく、アクティブルータがアクティブのままになります。 • トラッキングを解除するには、**no hsrp** コマンドを使用します。
ステップ 6	**hsrp** [*group-number*] **preempt** [**delay** *seconds*] 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# hsrp preempt`	（任意）HSRP プリエンプションとプリエンプション遅延を設定します。 • *group-number* の値を指定しないと、設定はルータ上のすべての HSRP グループに適用されます。 • **hsrp preempt** コマンドでプリエンプションおよびプリエンプション遅延を設定した場合、ローカル ルータに現在のアクティブ ルータよりも高いホットスタンバイ プライオリティが設定されているときには、そのローカル ルータはアクティブ ルータとして制御を引き継ごうとします。 **hsrp preempt** コマンドを設定していない場合、ローカル ルータは、（指定ルータとして機能する）現在アクティブ状態のルータがないことを示す情報を受信した場合にのみ、アクティブ ルータとして制御を引き継ぎます。 • ルータが最初に起動したとき、ルータのルーティング テーブルは完全ではありません。プリエンプション処理するように設定されている場合にはアクティブ ルータになりますが、まだ十分なルーティング処理はできません。 この問題を解決するには、プリエンプション処理する側のルータが現在アクティブなルータを実際にプリエンプション処理するまでの遅延を設定します。 • 現在アクティブ状態のルータがない場合は、プリエンプションの *delay seconds* の値は適用されません。 この場合、ローカル ルータは、プリエンプション遅延の秒数に関係なく、該当するタイムアウトが経過したあと（**hsrp timers** コマンドを参照）、アクティブになります。 • HSRP プリエンプションおよびプリエンプション遅延値をデフォルトに戻すには、**no hsrp** コマンドを使用します。
ステップ 7	**hsrp** [*group-number*] **authentication** *string* 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# hsrp 1 authentication company1`	（任意）ホットスタンバイ ルータ プロトコル（HSRP）用の認証ストリングを設定します。 • *group-number* の値を指定しないと、設定はルータ上のすべての HSRP グループに適用されます。

	コマンドまたはアクション	目的
		• 認証ストリングはすべての HSRP メッセージで暗号化されずに送信されます。 相互運用性を確保するには、LAN 上のすべてのルータおよびアクセス サーバに同じ認証ストリングを設定する必要があります。 • 認証ストリングが一致しないと、デバイスは、HSRP で設定された他のルータから、指定されたホットスタンバイ IP アドレスおよびホットスタンバイ タイマー値を学習できません。 • 認証ストリングが一致しないと、あるルータが指定ルータを引き継ぐというようなプロトコル イベントを回避できません。 • 認証ストリングを削除するには、**no hsrp** コマンドを使用します。
ステップ 8	**hsrp use-bia** 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# hsrp use-bia`	（任意）事前に割り当てられた MAC アドレスまたは機能アドレスではなく、インターフェイスのバーンドイン アドレスを仮想 MAC アドレスとして使用するように、HSRP を設定します。 • 送信元ハードウェア アドレスが機能アドレスに設定されたアドレス解決プロトコル（ARP）応答を拒否するデバイスがあるときは、インターフェイスで **use-bia** コマンドを入力します。 • デフォルトの仮想 MAC アドレスに戻すには、**no hsrp use-bia** コマンドを使用します。
ステップ 9	**hsrp** [*group-number*] **mac-address** *address* 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# hsrp 5 mac-address 4000.1000.1060`	（任意）HSRP 用の仮想 MAC アドレスを指定します。 • *group-number* 引数の値を指定しないと、設定はルータ上のすべての HSRP グループに適用されます。 • ファーストホップ冗長が仮想 MAC アドレスを使用できることに基づき、かつイーサネット スイッチに接続されている PC ではファーストホップ アドレスを変更できない IBM ネットワーク環境を除いて、このコマンドは推奨しません。 • HSRP を使用すると、エンド ステーションで IP ルーティングのファーストホップ ゲートウェイを見つけるのに役立ちます。エンドステーションは、デフォルトのゲートウェイで設定されます。 ただし、HSRP はその他のプロトコルにファーストホップの冗長性を提供できます。 拡張分散ネットワーク機能（APPN）などの一部のプロトコルでは、MAC アドレスを使用して、ルーティングのためにファーストホップを特定します。この場合、仮想 MAC アドレスの指定が必要になることがよくあります。これらのプロトコルにとって仮想 IP アドレスは重

	コマンドまたはアクション	目的
		要ではありません。 仮想 MAC アドレスを指定するには、**hsrp mac-address** コマンドを使用します。 • ルータがアクティブな場合、指定された MAC アドレスが仮想 MAC アドレスとして使用されます。 • **hsrp mac-address** コマンドは、特定の APPN 設定向けのコマンドです。 • APPN ネットワークでは、エンドノードは隣接するネットワーク ノードの MAC アドレスを使用して設定するのが通常です。仮想 MAC アドレスをエンド ノードで使用される値に設定するには、ルータで **hsrp mac-address** コマンドを使用します。 • 標準の仮想 MAC アドレス（0000.0C07.ACn）に戻すには、**no hsrp** [*group-number*] **mac-address** コマンドを使用します。

ステップ 10	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

HSRP アクティベーション遅延の設定

HSRP のアクティベーション遅延は、インターフェイスがアップ状態になったときに、ステートマシンの起動を遅らせることを目的としています。これにより、ネットワーク タイムが安定し、リンクがアップ状態になったあとの早い段階で不必要に状態が変化するのを防ぐことができます。

手順の概要

1. **configure**
2. **router hsrp**
3. **interface** *type interface-path-id*
4. **hsrp** [*group-number*] **ipv4** [*ip-address* [*secondary*]]
5. 次のいずれかを実行します。
   - **hsrp delay** [**minimum** *seconds*] [**reload** *seconds*]
   -
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router hsrp** 例： `RP/0/RSP0/CPU0:router(config)# router hsrp`	HSRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-hsrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスでHSRPインターフェイス コンフィギュレーション モード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	**hsrp** [*group-number*] **ipv4** [*ip-address* [*secondary*]] 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# hsrp 1 ipv4`	設定済みのインターフェイスで HSRP をアクティブにします。 • IP アドレスを指定した場合は、IP アドレスがホットスタンバイ グループの指定アドレスとして使用されます。IP アドレスが指定されていない場合は、仮想アドレスがアクティブ ルータから学習されます。
ステップ 5	次のいずれかを実行します。 • **hsrp delay** [**minimum** *seconds*] [**reload** *seconds*] • 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)#hsrp delay minimum 2 reload 10`	ネットワークが安定する時間を確保し、リンクの起動後すぐに不要な状態変更がないように、インターフェイス起動時にステート マシンの起動を遅らせます。 リロード遅延は、最初のインターフェイス起動イベント後に適用される遅延です。最小遅延は、後続の（インターフェイスがフラップする場合の）インターフェイス起動イベントに適用される遅延です。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

ICMP リダイレクト メッセージの HSRP サポートのイネーブル化

デフォルトでは、ICMP リダイレクト メッセージの HSRP フィルタリングは、HSRP が実行されているルータでイネーブルになっています。

ディセーブルになっているこの機能の再イネーブル化をルータに設定するには、インターフェイス コンフィギュレーション モードで **hsrp redirects** コマンドを使用します。

手順の概要

1. **configure**
2. **router hsrp**
3. **interface** *type interface-path-id*
4. **hsrp** [*group-number*] **ipv4** [*ip-address* [*secondary*]]
5. **hsrp redirects disable**
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router hsrp** 例： `RP/0/RSP0/CPU0:router(config)# router hsrp`	HSRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-hsrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスで HSRP インターフェイス コンフィギュレーション モードをイネーブルにします。
ステップ 4	**hsrp** [*group-number*] **ipv4** [*ip-address* [*secondary*]]	設定済みのインターフェイスで HSRP をアクティブにします。 • IP アドレスを指定した場合は、IP アドレスがホットスタンバイ グループの指定アドレスとして使用されます。IP アド

	コマンドまたはアクション	目的
	例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# hsrp 1 ipv4`	レスが指定されていない場合は、仮想アドレスがアクティブ ルータから学習されます。
ステップ 5	**hsrp redirects disable** 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# hsrp redirects`	インターフェイスにホットスタンバイ ルータ プロトコル（HSRP）が設定されているときに送信する Internet Control Message Protocol（ICMP）リダイレクト メッセージを設定します。 • **hsrp redirects** コマンドは、インターフェイスごとに設定できます。 インターフェイス上で最初に HSRP を設定する場合、このインターフェイスの設定ではグローバル値を継承します。 ICMP リダイレクトをインターフェイスで明示的にディセーブルにしている場合は、グローバル コマンドではその機能を再びイネーブルにすることができません。 • **hsrp redirects** コマンドがイネーブルである場合、リダイレクト パケットのネクストホップ アドレスの実 IP アドレスが仮想 IP アドレスに置き換えられて（それが HSRP に認識されている場合）、ICMP リダイレクト メッセージがフィルタリングされます。 • デフォルト（ICMP メッセージがイネーブル）に戻すには、**no hsrp redirects** コマンドを使用します。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		• 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。

HSRP のマルチ グループ オプティマイゼーション（MGO）

マルチグループオプティマイゼーションは、多くのサブインターフェイスで構成される配置で制御トラフィックを削減するためのソリューションです。HSRP制御トラフィックの実行をセッションの 1 つに限ることにより、冗長性要件が同じサブインターフェイスでは制御トラフィックが減少します。他のすべてのセッションはこのプライマリ セッションのスレーブになり、プライマリ セッションから状態を継承します。

HSRP のカスタマイズ

HSRP 動作のカスタマイズは任意です。 HSRP グループをイネーブルにすると、そのグループはすぐに動作します。

手順の概要

1. **configure**
2. **router hsrp**
3. **interface** *type interface-path-id*
4. **address-family ipv4**
5. **hsrp** *group-no*
6. **name** *name*
7. **address** { **learn** | *address*}
8. **address** *address* **secondary**
9. **authentication** *string*
10. **bfd fast-detect**
11. **mac-address** *address*
12. **hsrp** *group-no* **slave**
13. **follow** *mgo-session-name*
14. **address** *ip-address*
15. 次のいずれかのコマンドを使用します。
    - **end**
    - **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	**router hsrp** 例： `RP/0/RSP0/CPU0:router(config)# router hsrp`	HSRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-hsrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスで HSRP インターフェイス コンフィギュレーション モードをイネーブルにします。
ステップ 4	**address-family ipv4** 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# address-family ipv4`	特定のインターフェイスで HSRP アドレス ファミリ コンフィギュレーション モードをイネーブルにします。
ステップ 5	**hsrp** *group-no* 例： `RP/0/RSP0/CPU0:router(config-hsrp-ipv4)# hsrp 1`	特定のインターフェイスで HSRP グループ コンフィギュレーション モードをイネーブルにします。
ステップ 6	**name** *name* 例： `RP/0/RSP0/CPU0:router(config-hsrp-gp)# name s1`	HSRP セッション名を設定します。
ステップ 7	**address** { **learn** | *address*} 例： `RP/0/RSP0/CPU0:router(config-hsrp-gp)# address learn`	IP のホットスタンバイ プロトコルをイネーブルにします。 • IP アドレスを指定した場合は、IP アドレスがホットスタンバイ グループの指定アドレスとして使用されます。IP アドレスが指定されていない場合は、仮想アドレスがアクティブ ルータから学習されます。

	コマンドまたはアクション	目的
ステップ 8	**address** *address* **secondary** 例： `RP/0/RSP0/CPU0:router(config-hsrp-gp)# address 10.20.30.1 secondary`	ルータのセカンダリ仮想 IPv4 アドレスを設定します。
ステップ 9	**authentication** *string* 例： `RP/0/RSP0/CPU0:router(config-hsrp-gp)# authentication company1`	ホットスタンバイ ルータ プロトコル（HSRP）用の認証ストリングを設定します。
ステップ 10	**bfd fast-detect** 例： `RP/0/RSP0/CPU0:router(config-hsrp-gp)# bfd fast-detect`	HSRPインターフェイスで高速の双方向転送検出（BFD）をイネーブルにします。
ステップ 11	**mac-address** *address* 例： `RP/0/RSP0/CPU0:router(config-hsrp-gp)# mac-address 4000.1000.1060`	ホットスタンバイ ルータ プロトコル（HSRP）用の仮想 MAC アドレスを指定します。
ステップ 12	**hsrp** *group-no* **slave** 例： `RP/0/RSP0/CPU0:router(config-hsrp-gp)# hsrp 2 slave`	特定のインターフェイスでHSRP スレーブ コンフィギュレーション モードをイネーブルにします。
ステップ 13	**follow** *mgo-session-name* 例： `RP/0/RSP0/CPU0:router(config-hsrp-slave)# follow s1`	指定のグループから状態を継承するようにスレーブ グループに指示します。

	コマンドまたはアクション	目的
ステップ 14	**address** *ip-address* 例： `RP/0/RSP0/CPU0:router(config-hsrp-slave)# address 10.3.2.2`	スレーブ グループ用にプライマリ仮想 IPv4 アドレスを設定します。

ステップ 15	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

プライマリ仮想 IPv4 アドレスの設定

IP のホットスタンバイ プロトコルをイネーブルにするには、HSRP グループ サブモードで **address (hsrp)** コマンドを使用します。

手順の概要

1. **configure**
2. **router hsrp**
3. **interface** *type interface-path-id*
4. **address-family ipv4**
5. **hsrp** *group-no*
6. **address** { **learn** | *address*}
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router hsrp** 例： `RP/0/RSP0/CPU0:router(config)# router hsrp`	HSRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-hsrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスでHSRPインターフェイス コンフィギュレーション モードをイネーブルにします。
ステップ 4	**address-family ipv4** 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# address-family ipv4`	特定のインターフェイスで HSRP アドレス ファミリ コンフィギュレーション モードをイネーブルにします。
ステップ 5	**hsrp** *group-no* 例： `RP/0/RSP0/CPU0:router(config-hsrp-ipv4)# hsrp 1`	特定のインターフェイスでHSRPグループ コンフィギュレーション モード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	**address** { **learn** | *address*} 例： `RP/0/RSP0/CPU0:router(config-hsrp-ipv4)# address learn`	IP のホットスタンバイ プロトコルをイネーブルにします。 • IP アドレスを指定した場合は、IP アドレスがホットスタンバイ グループの指定アドレスとして使用されます。 IP アドレスが指定されていない場合は、仮想アドレスがアクティブ ルータから学習されます。

ステップ 7	次のいずれかのコマンドを使用します。
	
	 • **end**
	 • **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。

セカンダリ仮想 IPv4 アドレスの設定

ルータのセカンダリ仮想 IPv4 アドレスを設定するには、ホットスタンバイ ルータ プロトコル（HSRP）仮想ルータ サブモードで **address secondary** コマンドを使用します。

手順の概要

1. **configure**
2. **router hsrp**
3. **interface** *type interface-path-id*
4. **address-family ipv4**
5. **hsrp** *group-no*
6. **address** *address* **secondary**
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router hsrp** 例： `RP/0/RSP0/CPU0:router(config)# router hsrp`	HSRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-hsrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスで HSRP インターフェイス コンフィギュレーション モードをイネーブルにします。
ステップ 4	**address-family ipv4** 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# address-family ipv4`	特定のインターフェイスで HSRP アドレス ファミリ コンフィギュレーション モードをイネーブルにします。
ステップ 5	**hsrp** *group-no* 例： `RP/0/RSP0/CPU0:router(config-hsrp-ipv4)# hsrp 1`	特定のインターフェイスで HSRP グループ コンフィギュレーション モード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	**address** *address* **secondary** 例： `RP/0/RSP0/CPU0:router(config-hsrp-ipv4)# address 10.20.30.1 secondary`	ルータのセカンダリ仮想 IPv4 アドレスを設定します。

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

スレーブ フォローの設定

指定のグループから状態を継承するようにスレーブ グループに指示するには、HSRP スレーブ サブモード モードで **slave follow** コマンドを使用します。

手順の概要

1. **configure**
2. **router hsrp**
3. **interface** *type interface-path-id*
4. **address-family ipv4**
5. **hsrp** *group-no* **slave**
6. **follow** *mgo-session-name*
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router hsrp** 例： `RP/0/RSP0/CPU0:router(config)# router hsrp`	HSRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-hsrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスで HSRP インターフェイス コンフィギュレーション モードをイネーブルにします。
ステップ 4	**address-family ipv4** 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# address-family ipv4`	特定のインターフェイスで HSRP アドレス ファミリ コンフィギュレーション モードをイネーブルにします。
ステップ 5	**hsrp** *group-no* **slave** 例： `RP/0/RSP0/CPU0:router(config-hsrp-ipv4)# hsrp 2 slave`	特定のインターフェイスで HSRP スレーブ コンフィギュレーション モード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	**follow** *mgo-session-name* 例： `RP/0/RSP0/CPU0:router(config-hsrp-slave)# follow m1`	指定のグループから状態を継承するようにスレーブ グループに指示します。

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

スレーブ プライマリ仮想 IPv4 アドレスの設定

スレーブ グループのプライマリ仮想 IPv4 アドレスを設定するには、HSRP スレーブ サブモードで **slave primary virtual IPv4 address** コマンドを使用します。

手順の概要

1. **configure**
2. **router hsrp**
3. **interface** *type interface-path-id*
4. **address-family ipv4**
5. **hsrp** *group-no* **slave**
6. **address** *ip-address*
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router hsrp** 例： `RP/0/RSP0/CPU0:router(config)# router hsrp`	HSRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-hsrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスで HSRP インターフェイス コンフィギュレーション モードをイネーブルにします。
ステップ 4	**address-family ipv4** 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# address-family ipv4`	特定のインターフェイスで HSRP アドレス ファミリ コンフィギュレーション モードをイネーブルにします。
ステップ 5	**hsrp** *group-no* **slave** 例： `RP/0/RSP0/CPU0:router(config-hsrp-ipv4)# hsrp 2 slave`	特定のインターフェイスで HSRP スレーブ コンフィギュレーション モード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	**address** *ip-address* 例： `RP/0/RSP0/CPU0:router(config-hsrp-slave)# address 10.2.3.2`	スレーブ グループ用にプライマリ仮想 IPv4 アドレスを設定します。

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

スレーブ セカンダリ仮想 IPv4 アドレスの設定

スレーブ グループのセカンダリ仮想 IPv4 アドレスを設定するには、HSRP スレーブ サブモードで **slave secondary virtual IPv4 address** コマンドを使用します。

手順の概要

1. **configure**
2. **router hsrp**
3. **interface** *type interface-path-id*
4. **address-family ipv4**
5. **hsrp** *group-no* **slave**
6. **address** *address* **secondary**
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router hsrp** 例： `RP/0/RSP0/CPU0:router(config)# router hsrp`	HSRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-hsrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスで HSRP インターフェイス コンフィギュレーション モードをイネーブルにします。
ステップ 4	**address-family ipv4** 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# address-family ipv4`	特定のインターフェイスで HSRP アドレス ファミリ コンフィギュレーション モードをイネーブルにします。
ステップ 5	**hsrp** *group-no* **slave** 例： `RP/0/RSP0/CPU0:router(config-hsrp-ipv4)# hsrp 2 slave`	特定のインターフェイスで HSRP スレーブ コンフィギュレーション モード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	**address** *address* **secondary** 例： `RP/0/RSP0/CPU0:router(config-hsrp-slave)# address 10.20.30.1 secondary`	ルータのセカンダリ仮想 IPv4 アドレスを設定します。

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

スレーブ仮想 MAC アドレスの設定

スレーブ グループの仮想 MAC アドレスを設定するには、HSRP スレーブ サブモードで **slave virtual mac address** コマンドを使用します。

手順の概要

1. **configure**
2. **router hsrp**
3. **interface** *type interface-path-id*
4. **address-family ipv4**
5. **hsrp** *group-no* **slave**
6. **mac-address** *address*
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router hsrp** 例： `RP/0/RSP0/CPU0:router(config)# router hsrp`	HSRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-hsrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスで HSRP インターフェイス コンフィギュレーション モードをイネーブルにします。
ステップ 4	**address-family ipv4** 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# address-family ipv4`	特定のインターフェイスで HSRP アドレス ファミリ コンフィギュレーション モードをイネーブルにします。
ステップ 5	**hsrp** *group-no* **slave** 例： `RP/0/RSP0/CPU0:router(config-hsrp-ipv4)# hsrp 2 slave`	特定のインターフェイスで HSRP スレーブ コンフィギュレーション モード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	**mac-address** *address* 例： `RP/0/RSP0/CPU0:router(config-hsrp-slave)# mac-address 10.20.30`	スレーブ グループの仮想 MAC アドレスを設定します。

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

HSRP セッション名の設定

HSRP セッション名を設定するには、HSRP グループ サブモードで **session name** コマンドを使用します。

手順の概要

1. **configure**
2. **router hsrp**
3. **interface** *type interface-path-id*
4. **address-family ipv4**
5. **hsrp** *group-no*
6. **name** *name*
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router hsrp** 例： `RP/0/RSP0/CPU0:router(config)# router hsrp`	HSRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-hsrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスで HSRP インターフェイス コンフィギュレーション モードをイネーブルにします。
ステップ 4	**address-family ipv4** 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# address-family ipv4`	特定のインターフェイスで HSRP アドレス ファミリ コンフィギュレーション モードをイネーブルにします。
ステップ 5	**hsrp** *group-no* 例： `RP/0/RSP0/CPU0:router(config-hsrp-ipv4)# hsrp 1`	特定のインターフェイスで HSRP グループ コンフィギュレーション モード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	**name** *name* 例： `RP/0/RSP0/CPU0:router(config-hsrp-ipv4)# name s1`	HSRP セッション名を設定します。

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

HSRP 用 BFD

双方向フォワーディング検出（BFD）は、2 つのフォワーディング エンジン間の障害の検出に使用されるネットワーク プロトコルです。 BFD セッションは、非同期モードまたはデマンド モードという 2 つのモードのいずれかで動作できます。 非同期モードでは、両方のエンドポイントが互いに hello パケットを定期的に送信します。 これらのパケットを複数回受信しない場合は、セッションがダウンしていると見なされます。 デマンド モードでは、hello パケットの交換は必須ではなく、必要に応じてそれぞれのホストが hello メッセージを送信できます。 シスコでは、BFD 非同期モードをサポートしています。

BFD の利点

- BFD は、1 秒未満で障害を検出します。
- BFD では、すべてのタイプのカプセル化をサポートしています。
- BFDは、特定のルーティングプロトコルに限定されることなく、ほとんどすべてのルーティング プロトコルをサポートします。

BFD プロセス

HSRP は、BFD を使用して、リンク障害を検出し、制御パケットのオーバーヘッドを過度に発生させることなく、フェールオーバーにかかる時間を短縮します。

HSRPプロセスは、必要に応じてBFDセッションを確立します。 BFDセッションがダウンしたときは、セッションをモニタしている各スタンバイ グループがアクティブ状態に遷移します。

HSRPは、BFDセッションのダウンによって引き起こされたアクティブ状態への遷移後10秒間、状態の選択に関与しません。

BFD の設定

HSRP の場合、既存の HSRP インターフェイス サブモードの下で設定が適用されます。HSRP グループごとにBFD高速障害検出が設定可能であり、インターフェイスごとにタイマー（最小インターフェイスと乗数）が設定可能です。 BFD高速障害検出は、デフォルトでディセーブルになっています。

BFD のイネーブル化

手順の概要

1. **configure**
2. **router hsrp**
3. **interface** *type interface-path-id*
4. **address-family ipv4**
5. **hsrp** [*group number*] **bfd fast-detect** [**peer ipv4** *ipv4-address interface-type interface-path-id*]
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router hsrp** 例： `RP/0/RSP0/CPU0:router(config)# router hsrp`	HSRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-hsrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスでHSRPインターフェイス コンフィギュレーション モードをイネーブルにします。
ステップ 4	**address-family ipv4** 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# address-family ipv4`	特定のインターフェイスで HSRP アドレス ファミリ コンフィギュレーション モードをイネーブルにします。
ステップ 5	**hsrp** [*group number*] **bfd fast-detect** [**peer ipv4** *ipv4-address interface-type interface-path-id*] 例： `RP/0/RSP0/CPU0:router(config-hsrp-ipv4)# hsrp 1 bfd fast-detect peer ipv4 10.3.5.2 tenGigE 0/3/4/2`	特定のインターフェイスで高速障害検出をイネーブルにします。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		◦ **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。

BFD タイマー（最小間隔）の変更

最小間隔により、BFD ピアへの BFD パケットの送信頻度（ミリ秒単位）が決まります。 デフォルトの最小間隔は 15 ミリ秒です。

手順の概要

1. **configure**
2. **router hsrp**
3. **interface** *type interface-path-id*
4. **hsrp bfd minimum-interval** *interval*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **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。

	コマンドまたはアクション	目的
ステップ 2	**router hsrp** 例： `RP/0/RSP0/CPU0:router(config)# router hsrp`	HSRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-hsrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスで HSRP インターフェイス コンフィギュレーション モードをイネーブルにします。
ステップ 4	**hsrp bfd minimum-interval** *interval* 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# hsrp bfd minimum-interval`	最小間隔を指定の間隔に設定します。 間隔はミリ秒で、範囲は 15 ～ 30000 ミリ秒です。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

BFD タイマー（乗数）の変更

乗数は、ピアが利用不可であると宣言するまでに許容される、BFD ピアから連続して紛失される BFD パケットの数です。 デフォルトの乗数は 3 です。

手順の概要

1. **configure**
2. **router hsrp**
3. **interface** *type interface-path-id*
4. **hsrp bfd multiplier** *multiplier*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router hsrp** 例： `RP/0/RSP0/CPU0:router(config)# router hsrp`	HSRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-hsrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスで HSRP インターフェイス コンフィギュレーション モードをイネーブルにします。
ステップ 4	**hsrp bfd multiplier** *multiplier* 例： `RP/0/RSP0/CPU0:router(config-hsrp-if)# hsrp bfd multiplier`	値に乗数を設定します。 範囲は 2 ～ 50 です。
ステップ 5	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **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。

HSRP のホット リスタート

1 つのアクティブ グループで HSRP プロセスの障害が発生した場合には、ピア HSRP アクティブ ルータ グループで強制的にフェールオーバーが行われないようにする必要があります。 ホット リスタートはウォーム RP フェールオーバーをサポートしており、ピア HSRP アクティブ ルータ グループへの強制的なフェールオーバーは発生しません。

ソフトウェアでの HSRP の実装の設定例

ここでは、次の HSRP 設定例について説明します。

HSRP グループの設定：例

次に、インターフェイスで HSRP をイネーブルにし、HSRP グループ属性を設定する例を示します。

```
configure
router hsrp
interface TenGigE 0/2/0/1
address-family ipv4
hsrp 1
name s1
address 10.0.0.5
timers 100 200
preempt delay 500
priority 20
track TenGigE 0/2/0/2
authentication company0
use-bia
commit
hsrp 2 slave
follow s1
address 10.3.2.2
commit
```

複数の HSRP グループ用のルータの設定：例

次に、複数の HSRP グループ用にルータを設定する例を示します。

```
configure
router hsrp
interface TenGigE 0/2/0/3
address family ipv4
hsrp 1
address 1.0.0.5
priority 20
preempt
authentication sclara
hsrp 2
address 1.0.0.6
priority 110
preempt
authentication mtview
hsrp 3
address 1.0.0.7
preempt
authentication svale
commit
```

その他の参考資料

ここでは、HSRP の関連資料について説明します。

関連資料

関連項目	参照先
QoS コマンド：コマンド構文の詳細、コマンド モード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『*Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Command Reference*』の「*Quality of Service Commands*」
クラスベースのトラフィック シェーピング、トラフィック ポリシング、低遅延キューイング、および Modified Deficit Round Robin（MDRR）	『*Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Configuration Guide*』の「*Configuring Modular Quality of Service Congestion Management*」
WRED、RED、およびテール ドロップ	『*Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Configuration Guide*』の「*Configuring Modular QoS Congestion Avoidance*」
HSRP コマンド	『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*HSRP Commands*」
マスター コマンド リファレンス	『*Cisco ASR 9000 Series Aggregation Services Router Commands Master List*』
スタートアップ資料	『*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 ASR 9000 Series Aggregation Services Router System Security Configuration Guide*』の「*Configuring AAA Services*」

標準および RFC

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

シスコのテクニカル サポート

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

第 7 章

LPTS の実装

Local Packet Transport Services（LPTS）では、セキュア ドメイン ルータ（SDR）宛てのすべてのパケット フローを記述するテーブルを保持し、これにより、意図した宛先に確実にパケットが配信されます。

この章に記載されている LPTS コマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「LPTS Commands」の章を参照してください。

LPTS の実装の機能履歴

リリース	変更内容
リリース 3.9.0	LPTS が追加されました。

LPTS の実装の前提条件

次に、LPTS を実装するための前提条件を示します。

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。

LPTS の実装について

このマニュアルで説明している LPTS 機能を実装するには、次の概念を理解しておく必要があります。

LPTS の概要

LPTS では、ポート アービトレータおよびフロー マネージャという 2 つのコンポーネントを使用して、このタスクを実行します。 ポート アービトレータおよびフロー マネージャは、Internal Forwarding Information Base（IFIB）と呼ばれる、論理ルータ用のパケット フローを記述するテーブルを保持するプロセスです。 IFIB は、受信したパケットを正しいルート プロセッサまたはラインカードにルーティングして処理するために使用します。

LPTS は、ルータ外からパケットを受信するすべてのアプリケーションと内部的にインターフェイスします。 LPTS は、カスタマー設定の必要なく機能します。 ただし、カスタマーが LPTS のフローマネージャおよびポートアービトレータのアクティビティとパフォーマンスをモニタリングできるように、LPTS の **show** コマンドが提供されています。

LPTS ポリサー

Cisco IOS XR では、ルート プロセッサ（RP）宛ての制御パケットは、着信ラインカード内の一連の入力ポリサーを使用してポリシングされます。 これらのポリサーは、ブートアップ時に LPTS コンポーネントによって静的にプログラミングされます。 これらのポリサーは、着信制御トラフィックのフロー タイプに基づいて適用されます。 フロー タイプは、パケット ヘッダーを調べることで決定されます。 これらの静的入力ポリサーのポリサー レートは、コンフィギュレーション ファイルで定義され、ブートアップ時にラインカード上にプログラミングされます。

これらの一連の入力ポリサーのフロータイプに基づいて、ポリサー値を変更できます。 ポリサーごとのレートは、コマンドライン インターフェイス（CLI）を使用してノード単位で（ローカルに）およびグローバルに設定できるため、静的なポリサー値を上書きできます。

LPTS の実装方法

ここでは、次のタスクの手順について説明します。

LPTS ポリサーの設定

このタスクによって、LPTS ポリサーを設定できます。

手順の概要

1. **configure**
2. **lpts pifib hardware police** [**location** *node-id*]
3. **flow** {*flow_type*} {**rate** *rate*}
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
5. **show lpts pifib hardware police** [**location** {**all** | *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	**lpts pifib hardware police** [**location** *node-id*]
	例：
	`RP/0/RSP0/CPU0:router(config)# lpts pifib hardware police location 0/2/CPU0`
	`RP/0/RSP0/CPU0:router(config-pifib-policer-per-node)#`
		入力ポリサーを設定し、pifib policer グローバル コンフィギュレーション モードまたは pifib policer ノードごとコンフィギュレーション モードを開始します。
		次に、pifib policer ノードごとコンフィギュレーション モードの例を示します。

ステップ 3	**flow** {*flow_type*} {**rate** *rate*} 例： `RP/0/RSP0/CPU0:router(config-pifib-policer-per-node)# flow ospf unicast default rate 20000`	LPTS フロー タイプのポリサーを設定します。 次に、ospf フロー タイプのポリサーを設定する方法を示します。 • *flow_type* 引数を使用して、該当するフロー タイプを選択します。 フロータイプの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』を参照してください。 • **rate** キーワードを使用して、レートをパケット/秒（PPS）単位で指定します。 範囲は 0 ～ 4294967295 です。
ステップ 4	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **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 5	**show lpts pifib hardware police** [**location** {**all** | *node_id*}] 例： `RP/0/RSP0/CPU0:router# show lpts pifib hardware police location 0/2/cpu0`	ポリサー設定値セットを表示します。 • （任意）**location** キーワードを使用して、指定したノードの Pre-Internal Forwarding Information Base（IFIB）情報を表示します。 *node-id* 引数は、*rack/slot/module* の形式で入力します。 • **all** キーワードを使用して、すべての場所を指定します。

LPTS ポリサーの実装の設定例

ここでは、次の設定例について説明します。

LPTS ポリサーの設定：例

次に、LPTS ポリサーを設定する例を示します。

```
configure
 lpts pifib hardware police
  flow ospf unicast default rate 200
  flow bgp configured rate 200
  flow bgp default rate 100
 !
 lpts pifib hardware police location 0/2/CPU0
  flow ospf unicast default rate 100
  flow bgp configured rate 300
 !
show lpts pifib hardware police location 0/2/CPU0

FT - Flow type ID; PPS - Packets per second configured rate

FT Flow type                  Rate (PPS) Accept/Drop
-- -------------------------- ---------- ----------------
0  unconfigured-default       101        0/0
0

unconfigured-default

101
       0/0
1

Fragment

1000       0
/0
2
 OSPF-mc-known

1500

32550
/0
3
 OSPF-mc
-default

250
       0/0
4
 OSPF-uc-known

2000

0
/0
5

OSPF
-uc-default

 101

 1
/0
6
 ISIS-known
              250   1500        0/0
7

ISIS
-default
```

```
250

 0
/0
8
 BGP-known

 2000       17612
/0
9
BGP-default cfg-peer               203

5
/0
10 BGP
-default

500

 4
/0
11
PIM-mcast
                 1500       0/0
12 PIM-ucast
                 1500       0/0
13 IGMP

      1500
      0/0
14
ICMP-local
                1046       0/0
15
ICMP-app
              1000     1046       0/0
16
ICMP-control

1000
      0/0
17 ICMP
-default

1046       0
/0
18
LDP-TCP-known
           1500       9965
/0
19
LDP-TCP-cfg-peer

1500
0/0
20
LDP-TCP-default

250

 0
/0
21 LDP
-UDP

1000

59759
/0
22 All
-routers
             1500       0/0
```

```
23
LMP-TCP-known
           1500       0/0
24
LMP-TCP-cfg-peer

1500
0/0
25
LMP-TCP-default

250
       0/0
26 LMP
-UDP                      1000       0/0
27 RSVP-UDP
                    1000       0/0
28 RSVP
1000       0/0
29 IKE
                1000       0/0
30
IPSEC-known

1000
0/0
31 IPSEC
-default

 250
       0/0
32
MSDP-known
                  1000       0/0
33
MSDP-cfg-peer

 1000
0/0
34 MSDP-default

250
       0/0
35 SNMP

1000
0/0
36 NTP

500
       0/0
37
SSH-known
                1000       0/0
38 SSH
-default
                1000       0/0
39
HTTP-known
               1000       0/0
40 HTTP
-default
1000       0/0
41
SHTTP-known
              1000       0/0
42 SHTTP
-default
              1000       0/0
43
TELNET-known
             500   1000        0/0
44 TELNET
```

```
-default

500
       0/0
45
CSS-known

1000
0/0
46 CSS
-default

500
       0/0
47
RSH-known

1000
0/0
48 RSH
-default

500
       0/0
49
UDP-known

 2000
      0/0
50
UDP-listen
              1500       0/0
51
UDP-cfg-peer

1500

0
/0
52 UDP
-default

101

 653
/0
53
TCP-known
                 2000       0/0
54
TCP-listen
              2000       0/0
55
TCP-cfg-peer

2000

0
/0
56 TCP
-default

101

 6
/0
57
Mcast-known

2000
0/0
58 Mcast
-default
```

```
 101
       0/0
59
Raw-listen
                  250         0/0
60 Raw
-default

 250
       0/0
61 ip-sla

1000
      0/0
62 EIGRP
                    1500        0/0
63 RIP
                   2398     1500        0/0
64
PCEP                         101         0/0
```

その他の参考資料

ここでは、LPTS の実装に関する関連資料について説明します。

関連資料

関連項目	参照先
Cisco IOS XR LPTS コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*Cisco LPTS Commands*」の章

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
—	MIB を特定してダウンロードするには、次の URL にある Cisco MIB Locator を使用し、[Cisco Access Products] メニューからプラットフォームを選択します。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

第 8 章

ネットワーク スタック IPv4 および IPv6 の実装

ネットワーク スタック IPv4 および IPv6 機能は、インターネット プロトコル バージョン 4（IPv4）およびインターネット プロトコル バージョン 6（IPv6）の設定とモニタリングに使用します。

この章では、ネットワーク スタック IPv4 および IPv6 を Cisco IOS XR ネットワークに実装するために必要な新規タスクおよび変更されたタスクについて説明します。

（注） ネットワーク スタック IPv4 および IPv6 のコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*Network Stack IPv4 and IPv6 Commands*」の章を参照してください。この章に記載されている他のコマンドのドキュメントについては、『*Cisco ASR 9000 Series Aggregation Services Router Commands Master List*』やオンライン検索を利用して参照してください。

ネットワーク スタック IPv4 および IPv6 の実装の機能履歴

リリース	変更内容
リリース 3.7.2	この機能が導入されました。
リリース 3.9.0	IPv4 用の GRE 機能が追加されました。

ネットワーク スタック IPv4 および IPv6 の実装の前提条件

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。

ネットワーク スタック IPv4 および IPv6 の実装の制約事項

IPv6 をサポートするすべての Cisco IOS XR ソフトウェアリリースで、複数の IPv6 グローバル アドレスを 1 つのインターフェイス上に設定できます。 ただし、1 つのインターフェイス上での複数の IPv6 リンクローカル アドレスはサポートされません。

ネットワーク スタック IPv4 および IPv6 の実装について

ネットワーク スタック IPv4 および IPv6 を実装するには、次の概念を理解しておく必要があります。

ネットワーク スタック IPv4 および IPv6 の例外

Cisco IOS XR ソフトウェアでのネットワーク スタック機能には、次の例外があります。

- Cisco IOS XR ソフトウェアでは、**clear ipv6 neighbors** コマンドと **show ipv6 neighbors** コマンドに **location** *node-id* キーワードが含まれています。 場所を指定した場合、指定した場所の隣接エントリのみが表示されます。
- **ipv6 nd scavenge-timeout** コマンドは、stale 状態の隣接エントリの有効期間を設定します。 隣接エントリの廃棄タイマーの有効期間が切れると、そのエントリはクリアされます。
- Cisco IOS XR ソフトウェアでは、**show ipv4 interface** コマンドと **show ipv6 interface** コマンドに **location** *node-id* キーワードが含まれています。 場所を指定した場合、指定した場所のインターフェイス エントリのみが表示されます。

- Cisco IOS XR ソフトウェア では、設定するときに、競合する IP アドレス エントリを許可します。 アクティブな 2 つのインターフェイスの間に IP アドレス競合が存在する場合、Cisco IOS XR ソフトウェアは、設定されている競合ポリシーに従って、インターフェイスを停止します（デフォルト ポリシーでは、より高いインターフェイス インスタンスを停止します）。 たとえば、GigabitEthernet 0/1/0/1 が GigabitEthernet 0/2/0/1 と競合した場合、GigabitEthernet 0/2/0/1 上の IPv4 プロトコルが停止され、GigabitEthernet 0/1/0/1 上の IPv4 はアクティブなままになります。

IPv4 および IPv6 機能

Cisco IOS XR ソフトウェアが IPv4 と IPv6 の両方のアドレスを使用して設定されている場合、インターフェイスは IPv4 と IPv6 の両方のネットワーク上のデータを送受信できます。

IPv6 のアーキテクチャは、エンドツーエンドのセキュリティ、Quality of Service（QoS）、グローバルに一意なアドレスなどのサービスを提供する一方で、既存の IPv4 ユーザが IPv6 に簡単に移行できるように設計されています。 拡大された IPv6 アドレス空間により、ネットワークのスケーラビリティが可能となり、グローバルな到達可能性が提供されます。 簡素化された IPv6 パケットヘッダー形式により、パケットの処理効率が向上しています。 IPv6 プレフィックス集約、簡略化されたネットワーク リナンバリング、および IPv6 サイト マルチホーミング機能によって、より効率的なルーティングを実現する IPv6 アドレッシング階層が提供されます。 IPv6 では、Open Shortest Path First（OSPF）、マルチプロトコル ボーダー ゲートウェイ プロトコル（BGP）などの広く導入されているルーティング プロトコルをサポートしています。

IPv6 ネイバー探索（nd）プロセスでは、インターネット制御メッセージ プロトコル（ICMP）および送信要求ノードマルチキャストアドレスを使用して、同じネットワーク（ローカル リンク）上のネイバーのリンク層アドレスを判断し、ネイバーに到達可能かどうかを確認し、隣接ルータを追跡します。

Cisco IOS XR ソフトウェアの IPv6

以前は IPng（次世代）と呼ばれていた IPv6 は、インターネットプロトコル（IP）の最新バージョンです。 IP は、デジタル ネットワーク上のデータ、音声、およびビデオ トラフィックの交換に使用されるパケットベースのプロトコルです。 IP バージョン 4（IPv4）の 32 ビット アドレッシング方式ではインターネットの成長の需要を十分に満たせないことが明らかになったときに、IPv6 が提案されました。 長い議論の後で、IP を IPng のベースにするが、はるかに大きなアドレス空間と、簡略化されたメイン ヘッダーや拡張ヘッダーなどの改善を追加することが決定されました。 IPv6 は、Internet Engineering Task Force（IETF）から発行されている RFC 2460、『*Internet Protocol, Version 6 (IPv6) Specification*』で最初に規定されました。 IPv6 でサポートされるアーキテクチャとサービスについては他の RFC で規定されています。

拡大された IPv6 アドレス空間

グローバルに一意な IP アドレスの需要は今後増加すると予想され、その需要を満たす必要があることが、IPv6 の主な目的です。 モバイルインターネット対応デバイス（携帯情報端末（PDA）、

電話、車両など）、Home Area Network（HAN）、ワイヤレス データ サービスなどのアプリケーションによって、グローバルに一意なIPアドレスの需要が増大しています。 IPv6は、ネットワーク アドレス ビット数を（IPv4 での）32 ビットの 4 倍の 128 ビットにしているため、地球上のすべてのネットワーク デバイスにグローバルに一意な IP アドレスを十分に提供できます。 IPv6 アドレスをグローバルに一意にすることで、ネットワーク デバイスのグローバルな到達可能性とエンドツーエンドのセキュリティが実現されます。これは、アドレスの需要を喚起するアプリケーションとサービスに不可欠な機能です。 また、柔軟性の高い IPv6 アドレス空間により、プライベートアドレスの必要性とネットワーク アドレス変換（NAT）の使用が低減されます。したがって、IPv6 を使用すると、ネットワーク エッジにある境界ルータによる特別な処理を必要としない新しいアプリケーション プロトコルがイネーブルになります。

IPv6 アドレス形式

IPv6 アドレスは、x:x:x:x:x:x:x:x のようにコロン（:）で区切られた一連の 16 ビットの 16 進フィールドで表されます。 次に、IPv6 アドレスの例を 2 つ示します。

2001:0DB8:7654:3210:FEDC:BA98:7654:3210

2001:0DB8:0:0:8:800:200C:417A

IPv6 アドレスには、通常、連続するゼロの 16 進フィールドが含まれます。 IPv6 アドレスを扱いやすくするために、2つのコロン（::）を使用して、IPv6アドレスの先頭、中間、最後の部分の連続したゼロの16進フィールドを圧縮できます。（これらのコロンは、連続したゼロの16進フィールドを表します）。表 2：圧縮された IPv6 アドレス形式, （218 ページ）に、圧縮された IPv6 アドレス形式を示します。

連続する 16 ビット値がゼロとして指定されている場合は、2 つのコロンを *ipv6-address* 引数の一部として使用できます。インターフェイスごとに複数のIPv6アドレスを設定できますが、設定できるリンクローカル アドレスは 1 つだけです。

（注） IPv6アドレスでは、最も長く連続するゼロの16進フィールドを表すために2つのコロン（::）を 1 回だけ使用できます。

IPv6 アドレスの 16 進文字は大文字と小文字が区別されません。

表 *2*：圧縮された *IPv6* アドレス形式

IPv6 アドレス タイプ	優先形式	圧縮形式
ユニキャスト	2001:0:0:0:0DB8:800:200C:417A	1080::0DB8:800:200C:417A
マルチキャスト	FF01:0:0:0:0:0:0:101	FF01::101
ループバック	0:0:0:0:0:0:0:1	::1
未指定	0:0:0:0:0:0:0:0	::

ノードは、表 2：圧縮された IPv6 アドレス形式, （218 ページ）に示されているループバック アドレスを使用して、IPv6 パケットを自身に送信できます。 IPv6 のループバック アドレスは、IPv4 のループバック アドレス（127.0.0.1）と同じように機能します。

（注） IPv6 ループバック アドレスは、物理インターフェイスに割り当てることができません。 IPv6 ループバック アドレスを送信元アドレスまたは宛先アドレスとするパケットは、そのパケットを作成したノード内に留まっている必要があります。 IPv6 ルータは、IPv6 ループバック アドレスを送信元アドレスまたは宛先アドレスとするパケットを転送しません。

表 2：圧縮された IPv6 アドレス形式, （218 ページ）に示されている未指定アドレスは、IPv6 アドレスがないことを示します。 たとえば、IPv6 ネットワーク上で新しく初期化されたノードは、IPv6 アドレスを受信するまで、パケットで未指定アドレスを送信元アドレスとして使用できます。

（注） IPv6 未指定アドレスは、インターフェイスに割り当てることができません。 未指定 IPv6 アドレスを IPv6 パケットまたは IPv6 ルーティング ヘッダーで宛先アドレスとして使用することはできません。

IPv6 アドレス プレフィックスは、*ipv6-prefix/prefix-length* の形式で、アドレス空間全体のビット連続ブロックを表すために使用できます。 *ipv6-prefix* 引数は、RFC 2373 に記載された形式にする必要があり、16 ビット値をコロンで区切った 16 進でアドレスを指定します。 プレフィックス長は、アドレスの高次の連続ビットのうち、何個がプレフィックス（アドレスのネットワーク部分）を構成しているかを指定する 10 進数値です。 たとえば、2001:0DB8:8086:6502::/32 は有効な IPv6 プレフィックスです。

IPv6 アドレス タイプ：ユニキャスト

IPv6ユニキャストアドレスは、単一ノード上の単一インターフェイスの識別子です。ユニキャスト アドレスに送信されたパケットは、そのアドレスが示すインターフェイスに配信されます。 Cisco IOS XR ソフトウェア では、次の IPv6 ユニキャスト アドレス タイプがサポートされています。

- 集約可能グローバル アドレス
- サイトローカル アドレス（IETF では廃止を提案しています）
- リンクローカル アドレス
- IPv4 互換 IPv6 アドレス

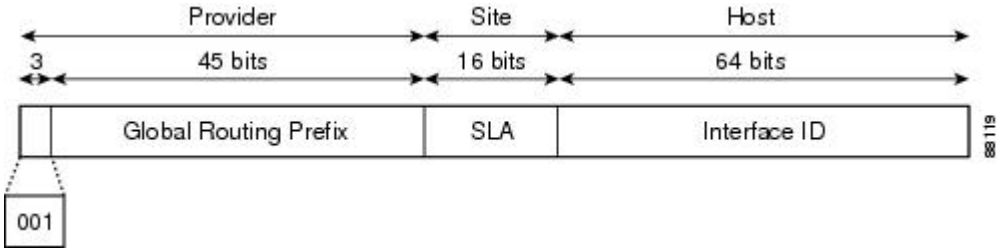
集約可能グローバル アドレス

集約可能グローバル アドレスは、集約可能なグローバル ユニキャスト プレフィックスによる IPv6 アドレスです。 集約可能グローバル ユニキャスト アドレスの構造により、グローバル ルーティング テーブル内のルーティング テーブル エントリ数を制限するルーティング プレフィックスの

厳密な集約が可能になります。集約可能グローバルアドレスは、組織を上に向かって、最終的にインターネット サービス プロバイダー（ISP）まで集約されるリンクで使用されます。

集約可能グローバル IPv6 アドレスは、グローバル ルーティング プレフィックス、サブネット ID、およびインターフェイス ID により定義されます。 バイナリ 000 から開始するアドレスを除き、すべてのグローバル ユニキャスト アドレスには 64 ビットのインターフェイス ID があります。現在のグローバルユニキャストアドレスの割り当てには、バイナリ値 001（2000::/3）から始まるアドレスの範囲が使用されます。 図 6：集約可能グローバル アドレス形式, （220 ページ）に、集約可能グローバル アドレスの構造を示します。

図 *6*：集約可能グローバル アドレス形式



2000::/3（001）～ E000::/3（111）のプレフィックスを持つアドレスには、Extended Universal Identifier（EUI）64 形式の 64 ビット インターフェイス識別子が必要です。 インターネット割り当て番号局（IANA）は、2000::/16 の範囲の IPv6 アドレス空間を地域レジストリに割り当てます。

集約可能グローバルアドレスは、通常、48 ビットのグローバル ルーティング プレフィックスと、16 ビットのサブネット ID またはサイトレベル集約（SLA）で構成されます。 RFC 2374（IPv6 集約可能グローバルユニキャストアドレス形式に関するドキュメント）では、グローバル ルーティング プレフィックスに Top-Level Aggregator（TLA）と Next-Level Aggregator（NLA）という他の 2 つの階層構造フィールドが含まれていました。IETF は、TLS フィールドと NLA フィールドがポリシーベースのフィールドであるため、これらのフィールドを RFC から削除することに決定しました。 この変更の前に展開された既存の IPv6 ネットワークの中には、依然として古いアーキテクチャに基づくネットワークを使用しているものもあります。

個々の組織では、サブネット ID と呼ばれる 16 ビットのサブネット フィールドを使用して、独自のローカル アドレッシング階層を作成したり、サブネットを識別したりできます。 サブネット ID は IPv4 でのサブネットに似ていますが、IPv6 サブネット ID を持つ組織では最大 65,535 個のサブネットをサポートできるという点が異なります。

インターフェイス ID は、リンク上のインターフェイスの識別に使用されます。 インターフェイス ID は、リンク上で一意である必要があります。 より広い範囲で一意にすることもできます。多くの場合、インターフェイス ID は、インターフェイスのリンク層アドレスと同じか、リンク層アドレスに基づいています。 集約可能グローバル ユニキャストおよびその他の IPv6 アドレス タイプで使用されるインターフェイス ID は、長さが 64 ビットの変更された EUI-64 形式で構築されている必要があります。

インターフェイス ID は、次のいずれかに該当する変更済みの EUI-64 形式で構築されています。

- すべての IEEE 802 インターフェイス タイプ（イーサネット インターフェイス、FDDI インターフェイスなど）の場合、最初の 3 オクテット（24 ビット）は、そのインターフェイスの

48 ビット リンク層アドレス（MAC アドレス）の組織固有識別子（OUI）から取得され、4 番めと 5 番めのオクテット（16 ビット）は、FFFE の固定 16 進数値です。最後の 3 オクテット（24 ビット）は、MAC アドレスの最後の 3 オクテットから取得されます。インターフェイス ID の構成は、最初のオクテットの 7 番めのビットである Universal/Local（U/L）ビットを 0 または 1 の値に設定することで完成します。値 0 はローカルに管理されている識別子を示し、値 1 はグローバルに一意の IPv6 インターフェイス識別子を示します。

- その他のすべてのインターフェイス タイプ（シリアル、ループバック、ATM、フレームリレー、トンネル インターフェイス タイプなど。ただし、IPv6 オーバーレイ トンネルで使用されるトンネル インターフェイスを除く）の場合、インターフェイス ID は IEEE 802 インターフェイス タイプのインターフェイス ID と同様に構築されますが、ルータの MAC アドレス プールからの最初の MAC アドレスを使用して識別子が構築される点が異なります（インターフェイスが MAC アドレスを持たないため）。
- IPv6 オーバーレイ トンネルで使用されるトンネル インターフェイス タイプの場合、インターフェイス ID は、識別子の上位 32 ビットがすべてゼロであるトンネル インターフェイスに割り当てられた IPv4 アドレスです。

（注） ポイントツーポイント プロトコル（PPP）を使用するインターフェイスの場合は、接続の両端のインターフェイスが同じ MAC アドレスを持つ可能性があるため、接続の両端で使用されるインターフェイス識別子は、両方の識別子が一意になるまでネゴシエーション（ランダムに選択され、必要に応じて再構築）されます。ルータの最初の MAC アドレスが、PPP を使用するインターフェイスの識別子の構築に使用されます。

ルータに IEEE 802 インターフェイス タイプがない場合は、ルータのインターフェイスでリンクローカル IPv6 アドレスが次のシーケンスで生成されます。

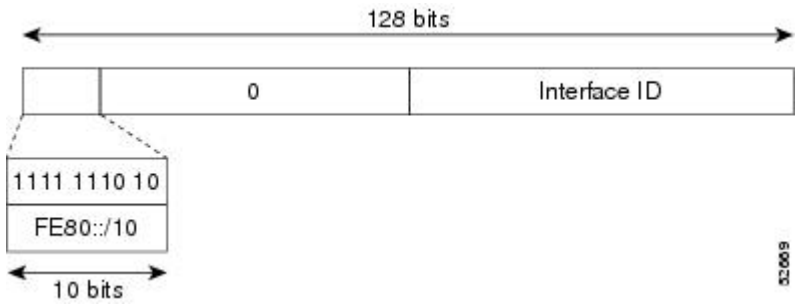
1. ルータに MAC アドレスが（ルータの MAC アドレス プールから）照会されます。
2. 使用できる MAC アドレスがない場合は、ルート プロセッサ（RP）またはラインカード（LC）のシリアル番号を使用して、リンクローカル アドレスを形成します。

リンクローカル アドレス

リンクローカル アドレスは、リンクローカル プレフィックス FE80::/10（1111 1110 10）と変更された EUI-64 形式のインターフェイス識別子を使用するすべてのインターフェイスを自動的に設定できる IPv6 ユニキャスト アドレスです。リンクローカル アドレスは、ネイバー探索プロトコルとステートレス自動設定プロセスで使用されます。ローカル リンク上のノードは、リンクローカルアドレスを使用して通信できます。ノードの通信にサイトローカルアドレスまたはグローバルに一意のアドレスは不要です。図 7：リンクローカル アドレス形式，（222 ページ）に、リンクローカル アドレスの構造を示します。

IPv6 ルータでは、送信元または宛先がリンクローカル アドレスであるパケットを他のリンクに転送できません。

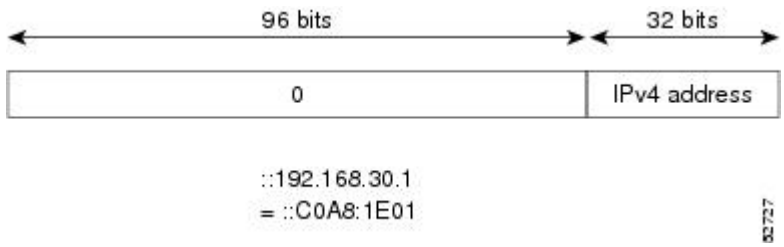
図 7：リンクローカル アドレス形式



IPv4 互換 IPv6 アドレス

IPv4 互換 IPv6 アドレスは、アドレスの上位 96 ビットがゼロであり、アドレスの下位 32 ビットが IPv4 アドレスである IPv6 ユニキャスト アドレスです。 IPv4 互換 IPv6 アドレスの形式は、0:0:0:0:0:0:A.B.C.D または ::A.B.C.D です。 IPv4 互換 IPv6 アドレスの 128 ビット全体がノードの IPv6 アドレスとして使用され、下位 32 ビットに埋め込まれた IPv4 アドレスがノードの IPv4 アドレスとして使用されます。 IPv4 互換 IPv6 アドレスは、IPv4 と IPv6 の両方のプロトコル スタックをサポートするノードに割り当てられ、自動トンネルで使用されます。 図 8：IPv4 互換 IPv6 アドレス形式, （222 ページ）に、IPv4 互換 IPv6 アドレスの構造と、許容されるいくつかのアドレス形式を示します。

図 8：*IPv4* 互換 *IPv6* アドレス形式

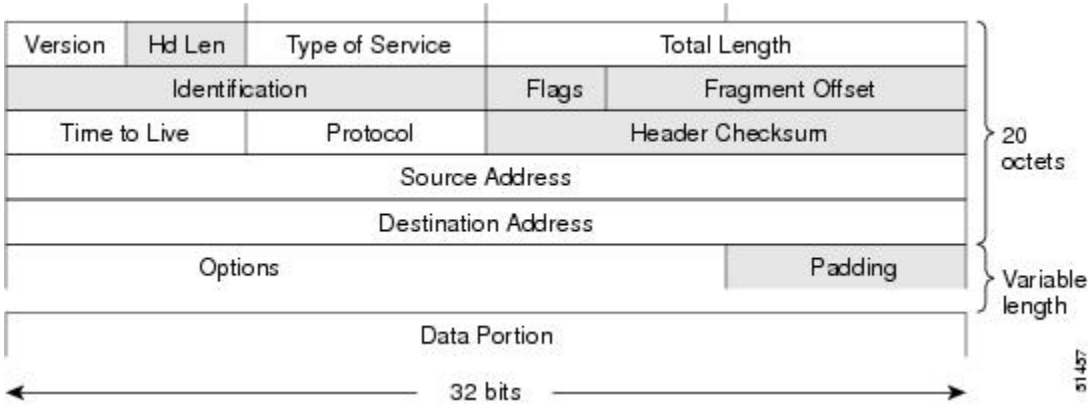


簡易 IPv6 パケット ヘッダー

基本 IPv4 パケット ヘッダーには、合計サイズが 20 オクテット（160 ビット）の 12 のフィールドがあります。 この 12 個のフィールドの後にはオプション フィールドが続く場合があり、さらにその後に、通常はトランスポートレイヤ パケットであるデータ部分が続きます。 可変長のオプション フィールドは、IPv4 パケット ヘッダーの合計サイズに加算されます。 IPv4 パケット ヘッ

ダーのグレーの部分のフィールドは、IPv6 パケット ヘッダーに含まれません（図 9：IPv4 パケット ヘッダー形式, （223 ページ）を参照）。

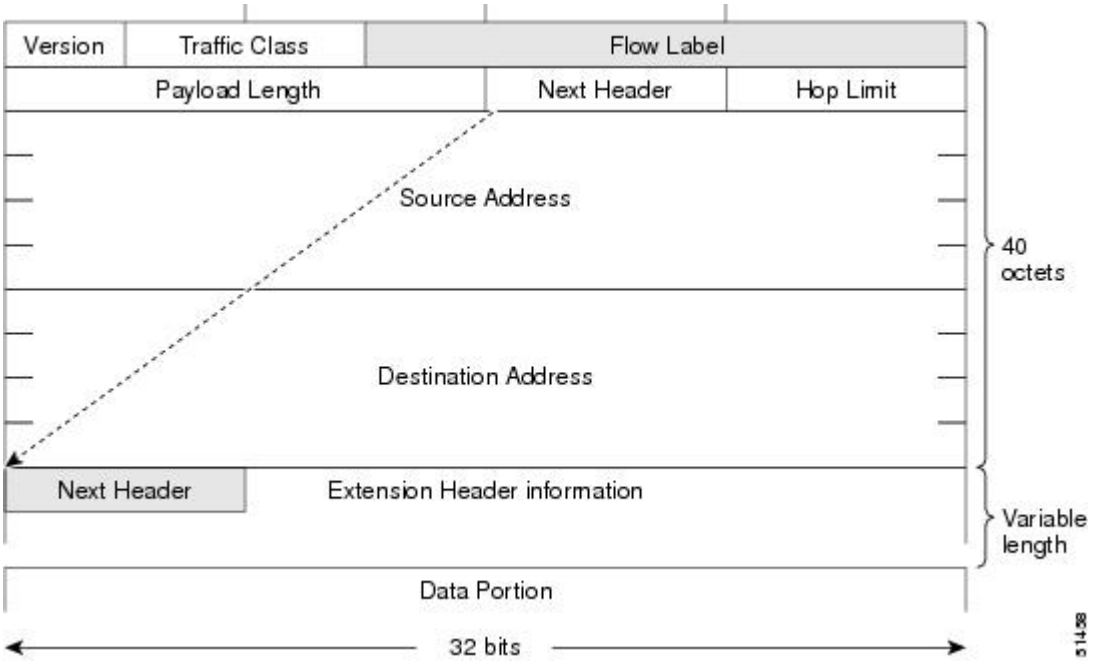
図 *9*：*IPv4* パケット ヘッダー形式



基本 IPv6 パケット ヘッダーには、合計サイズが 40 オクテット（320 ビット）の 8 つのフィールドがあります（図 10：IPv6 パケット ヘッダー形式, （224 ページ）を参照）。IPv6 では、フラグメンテーションはルータによって処理されず、チェックサムはネットワーク層で使用されないため、IPv6 ヘッダーからフィールドが除去されました。 代わりに、IPv6 のフラグメンテーションはパケットの送信元によって処理され、チェックサムはデータ リンク層とトランスポート層で使用されます （IPv4 では、ユーザ データグラム プロトコル（UDP）トランスポート層でオプションのチェックサムが使用されます。 IPv6 では、内部パケットの整合性をチェックするために UDP

チェックサムを使用する必要があります）。また、基本 IPv6 パケット ヘッダーとオプション フィールドは 64 ビットに揃えられるため、IPv6 パケットの処理が簡単になります。

図 *10*：*IPv6* パケット ヘッダー形式



次の表に、基本 IPv6 パケット ヘッダーのフィールドをリストします。

表 *3*：基本 *IPv6* パケット ヘッダー フィールド

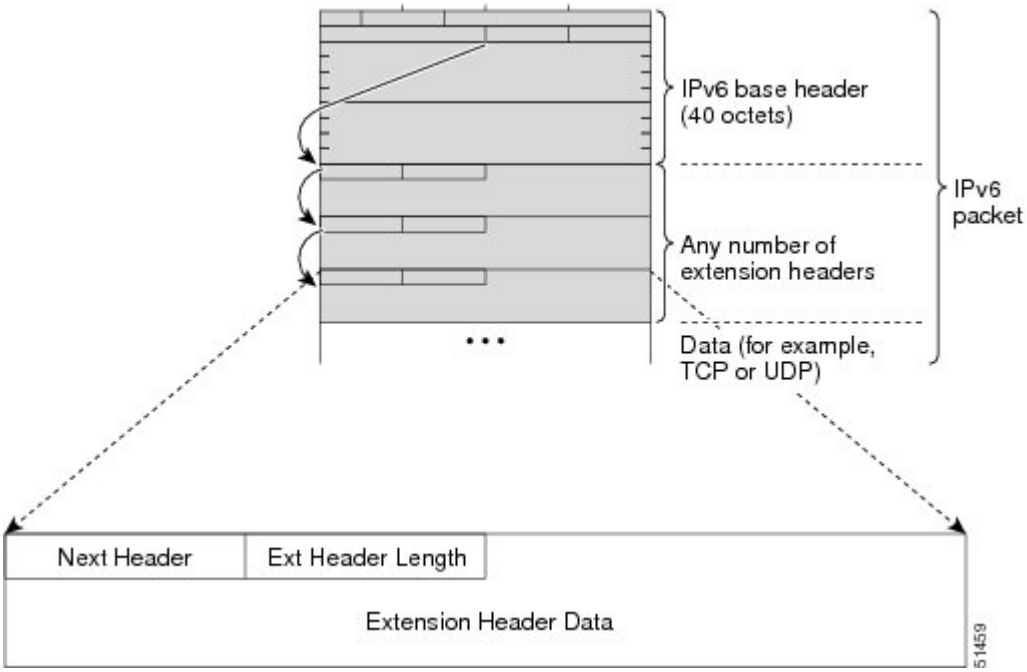
フィールド	説明
バージョン	IPv4 パケット ヘッダーのバージョン フィールドと同様ですが、IPv4 を意味する数字 4 の代わりに IPv6 を意味する数字 6 が示されます。
トラフィック クラス	IPv4 パケット ヘッダーのタイプ オブ サービス フィールドと同様です。 トラフィック クラス フィールドは、差別化されたサービスで使用されるトラフィック クラスのタグをパケットに付けます。
フロー ラベル	IPv6 パケット ヘッダーの新しいフィールドです。 フロー ラベル フィールドは、ネットワーク層でパケットを差別化する特定のフローのタグをパケットに付けます。

フィールド	説明
ペイロード長	IPv4 パケット ヘッダーの合計長フィールドと同様です。ペイロード長フィールドは、パケットのデータ部分の合計長を示します。
次ヘッダー	IPv4 パケット ヘッダーのプロトコル フィールドと同様です。 次ヘッダー フィールドの値により、基本 IPv6 ヘッダーに続く情報のタイプが決まります。 基本 IPv6 ヘッダーに続く情報のタイプは、図 11：IPv6 拡張ヘッダー形式, （226 ページ）に示すように、TCP や UDP パケットなどのトランスポートレイヤ パケット、または拡張ヘッダーです。
ホップ リミット	IPv4 パケット ヘッダーの存続可能時間フィールドと同様です。 ホップ リミット フィールドの値は、IPv6 パケットが無効と見なされる前に通過できるルータの最大数です。 各ルータを通過するたびに、この値が 1 つずつ減少します。 IPv6 ヘッダーにはチェックサムがないため、ルータは値を減らすたびにチェックサムを再計算する必要がなく、処理リソースが節約されます。
送信元アドレス	IPv4 パケット ヘッダーの送信元アドレス フィールドと同様ですが、IPv4 の 32 ビット送信元アドレスの代わりに、IPv6 では 128 ビットの送信元アドレスが含まれます。
宛先アドレス	IPv4 パケット ヘッダーの宛先アドレス フィールドと同様ですが、IPv4 の 32 ビット宛先アドレスの代わりに、IPv6 では 128 ビットの宛先アドレスが含まれます。

基本 IPv6 パケット ヘッダーの 8 つのフィールドの後に、オプションの拡張ヘッダーおよびパケットのデータ部分が続きます。 存在する場合は、各拡張ヘッダーが 64 ビットに揃えられます。 IPv6 パケットの拡張ヘッダーの数は固定されていません。 拡張ヘッダーがまとまってヘッダーのチェーンを形成します。 各拡張ヘッダーは、前のヘッダーの次ヘッダー フィールドによって識別されます。 通常は、最後の拡張ヘッダーに、TCP や UDP などのトランスポートレイヤ プロトコルの次

ヘッダー フィールドがあります。 図 11：IPv6 拡張ヘッダー形式, （226 ページ）に、IPv6 拡張ヘッダー形式を示します。

図 *11*：*IPv6* 拡張ヘッダー形式



次の表に、拡張ヘッダー タイプとその次ヘッダー フィールド値をリストします。

表 *4*：*IPv6* 拡張ヘッダー タイプ

ヘッダー タイプ	次ヘッダーの値	説明
ホップバイホップ オプション ヘッダー	0	このヘッダーは、パケットのパス上のすべてのホップで処理されます。 存在する場合、ホップバイホップ オプション ヘッダーは、常に基本 IPv6 パケット ヘッダーの直後に続きます。

ヘッダー タイプ	次ヘッダーの値	説明
宛先オプション ヘッダー	60	宛先オプション ヘッダーは、任意のホップバイホップ オプション ヘッダーの後に続くことがあります。その場合、宛先オプション ヘッダーは、最終的な宛先と、ルーティングヘッダーで指定された各通過アドレスでも処理されます。 また、宛先オプション ヘッダーは、任意のカプセル化セキュリティペイロード（ESP）ヘッダーの後に続くこともあります。その場合、宛先オプションヘッダーは、最終的な宛先でだけ処理されます。
ルーティング ヘッダー	43	ルーティング ヘッダーは送信元のルーティングに使用されます。
フラグメント ヘッダー	44	フラグメント ヘッダーは、送信元が、送信元と宛先の間のパスの最大伝送ユニット（MTU）よりも大きいパケットをフラグメント化する必要がある場合に使用されます。 フラグメント ヘッダーは、フラグメント化された各パケットで使用されます。
認証ヘッダー および ESP ヘッダー	51 50	認証ヘッダーと ESP ヘッダーは、パケットの認証、整合性、および機密性を提供するために IP セキュリティ プロトコル（IPSec）内で使用されます。これらのヘッダーは、IPv4 と IPv6 の両方で同一です。

ヘッダー タイプ	次ヘッダーの値	説明
上位層ヘッダー	6（TCP） 17（UDP）	上位層（トランスポート）ヘッダーは、データを転送するためにパケットの内部で使用される典型的なヘッダーです。2つの主要なトランスポート プロトコルは TCP と UDP です。
モビリティ ヘッダー	IANA で実行	バインディングの作成と管理に関連するすべてのメッセージで、モバイル ノード、通信ノード、およびホーム エージェントによって使用される拡張ヘッダーです。

IPv6 のパス MTU ディスカバリ

IPv4 の場合と同様に、IPv6 のパス MTU ディスカバリを使用すると、特定のデータ パス上のすべてのリンクの MTU サイズの差をホストが動的に検出し、調整できます。 ただし、IPv6 では、特定のデータ パス上の 1 つのリンクのパス MTU がパケットのサイズに十分に対応できる大きさでない場合に、フラグメンテーションはパケットの送信元によって処理されます。 IPv6 ホストでパケット フラグメンテーションを処理すると、IPv6 ルータの処理リソースが節約され、IPv6 ネットワークの効率が向上します。

（注） IPv4 では、最小リンク MTU が 68 オクテットであるため、特定のデータ パスに沿うすべてのリンクの MTU サイズが少なくとも 68 オクテットの MTU サイズをサポートする必要があります。

IPv6 では、最小リンク MTU は 1280 オクテットです。 IPv6 リンクには、1500 オクテットの MTU 値の使用をお勧めします。

（注） パス MTU ディスカバリは、TCP トランスポートを使用するアプリケーションでのみサポートされます。

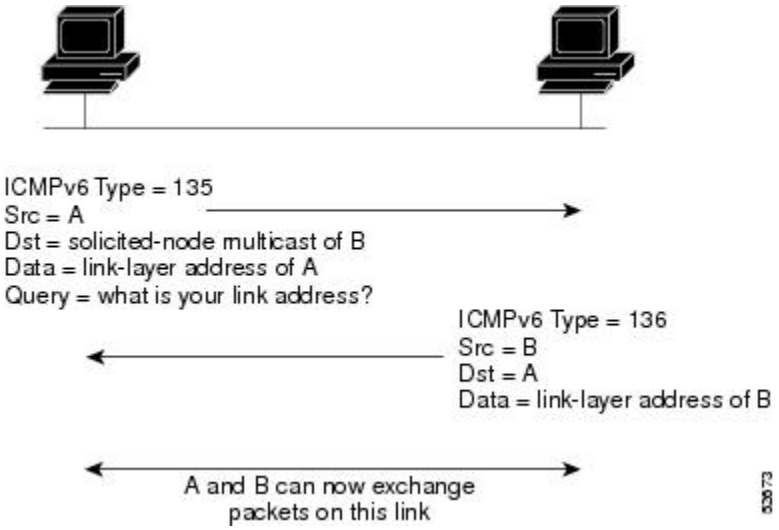
IPv6 ネイバー探索

IPv6 のネイバー探索プロセスは、ICMP メッセージと送信要求ノード マルチキャスト アドレスを使用して、同じネットワーク（ローカル リンク）上のネイバーのリンク層アドレスを判別し、ネイバーの到達可能性を確認して、隣接ルータの状況を把握します。

IPv6 ネイバー送信要求メッセージ

ICMP パケット ヘッダーのタイプ フィールドの値 135 は、ネイバー送信要求メッセージを示します。ネイバー送信要求メッセージは、ノードが同じローカル リンク上の別のノードのリンク層アドレスを決定するときに、ローカル リンク上で送信されます（図 12：IPv6 ネイバー探索 - ネイバー送信要求メッセージ, （229 ページ）を参照）。ノードで別のノードのリンク層アドレスを特定する必要がある場合、ネイバー送信要求メッセージの送信元アドレスは、ネイバー送信要求メッセージを送信するノードの IPv6 アドレスになります。ネイバー送信要求メッセージ内の宛先アドレスは、宛先ノードの IPv6 アドレスに対応する送信要求ノード マルチキャスト アドレスです。ネイバー送信要求メッセージには、送信元ノードのリンク層アドレスも含まれます。

図 *12*：*IPv6* ネイバー探索 - ネイバー送信要求メッセージ



ネイバー送信要求メッセージを受信した後に、宛先ノードは、ICMP パケット ヘッダーのタイプ フィールドに値 136 を含むネイバー アドバタイズメント メッセージをローカル リンクに送信することで応答します。ネイバー アドバタイズメント メッセージの送信元アドレスは、ネイバー アドバタイズメント メッセージを送信するノードの IPv6 アドレス（具体的には、ノード インターフェイスの IPv6 アドレス）です。ネイバー アドバタイズメント メッセージ内の宛先アドレスは、ネイバー送信要求メッセージを送信したノードの IPv6 アドレスです。ネイバー アドバタイズメント メッセージのデータ部分には、ネイバー アドバタイズメント メッセージを送信するノードのリンク層アドレスが含まれます。

送信元ノードがネイバー アドバタイズメントを受信すると、送信元ノードと宛先ノードが通信できるようになります。

ネイバー送信要求メッセージは、ネイバーのリンク層アドレスが識別された後に、ネイバーの到達可能性の確認にも使用されます。 ノードがネイバーの到達可能性を確認するときに、ネイバー送信要求メッセージの宛先アドレスは、ネイバーのユニキャスト アドレスです。

ネイバー アドバタイズメント メッセージは、ローカル リンク上のノードのリンク層アドレスが変更されたときにも送信されます。そのような変更があった場合、ネイバーアドバタイズメントの宛先アドレスは全ノード マルチキャスト アドレスになります。

ネイバー送信要求メッセージは、ネイバーのリンク層アドレスが識別された後に、ネイバーの到達可能性の確認にも使用されます。 ネイバー到達不能検出では、ネイバーの障害またはネイバーへの転送パスの障害が識別されます。この検出は、ホストとネイバー ノード（ホストまたはルータ）間のすべてのパスで使用されます。ネイバー到達不能検出は、ユニキャストパケットだけが送信されるネイバーに対して実行され、マルチキャスト パケットが送信されるネイバーに対しては実行されません。

ネイバーは、（以前にネイバーに送信されたパケットが受信され、処理されたことを示す）肯定確認応答がネイバーから返された場合に、到達可能と見なされます。 上位層プロトコル（TCP など）からの肯定確認応答は、接続で転送が順調に進行している（宛先に到達しつつある）こと、またはネイバー送信要求メッセージに対する応答でネイバーアドバタイズメントメッセージが受信されたことを示します。 パケットがピアに到達している場合、それらのパケットは送信元のネクストホップ ネイバーにも到達しています。 したがって、転送の進行により、ネクストホップ ネイバーが到達可能であることも確認されます。

ローカルリンク上にない宛先の場合、転送の進行は、ファーストホップルータが到達可能であることを暗に意味します。 上位層プロトコルからの確認応答がない場合、ノードは、ユニキャスト ネイバー送信要求メッセージを使用してネイバーを探し、転送パスがまだ機能していることを確認します。 送信要求ネイバー アドバタイズメント メッセージがネイバーから返されることは、転送パスがまだ機能していることを示す肯定確認応答です。 （送信要求フラグが値 1 に設定されたネイバーアドバタイズメントメッセージは、ネイバー送信要求メッセージへの応答でのみ送信されます）。非送信請求メッセージは送信元から宛先ノードへの一方向パスのみを確認し、送信要求ネイバー アドバタイズメント メッセージはパスが両方向で機能していることを示します。

（注） 送信要求フラグが値 0 に設定されたネイバー アドバタイズメント メッセージは、転送パスがまだ機能していることを示す肯定確認応答とは見なされません。

ネイバー送信要求メッセージは、ユニキャスト IPv6 アドレスがインターフェイスに割り当てられる前にそのアドレスが一意であることを確認するために、ステートレス自動設定プロセスでも使用されます。 アドレスがインターフェイスに割り当てられる前に、重複アドレス検出がまず新しいリンクローカル IPv6 アドレスで実行されます（重複アドレス検出の実行中、この新しいアドレスは一時的な状態のままになります）。具体的には、ノードは、メッセージ本体に未指定の送信元アドレスと一時的なリンクローカル アドレスが含まれたネイバー送信要求メッセージを送信します。 そのアドレスが別のノードですでに使用されている場合、ノードは一時的なリンクローカル アドレスを含むネイバー アドバタイズメント メッセージを返します。 別のノードが同じアドレスの一意性を同時に検証している場合は、そのノードもネイバー送信要求メッセージを返しま

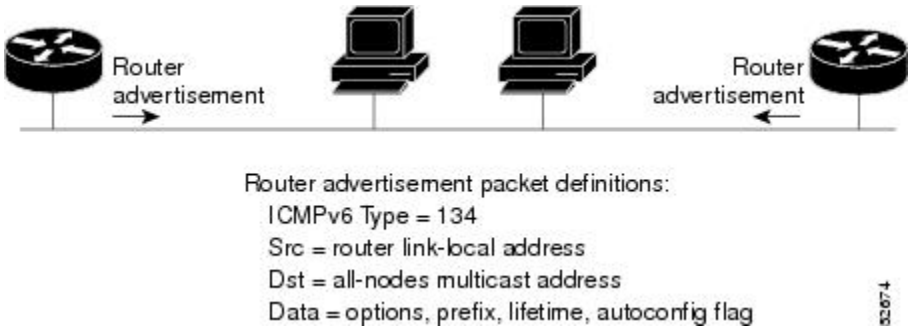
す。 ネイバー送信要求メッセージの返信としてネイバー アドバタイズメント メッセージが受信されず、同じ一時アドレスの検証を試行している他のノードからのネイバー送信要求メッセージも受信されない場合、最初のネイバー送信要求メッセージを送信したノードは、一時的なリンクローカル アドレスを一意であると見なし、そのアドレスをインターフェイスに割り当てます。

IPv6 ユニキャスト アドレス（グローバルまたはリンクローカル）はすべてリンクでの一意性を確認する必要があります。ただし、リンクローカル アドレスの一意性が確認されるまで、リンクローカルアドレスに関連付けられた他の IPv6 アドレスに対して重複アドレス検出は実行されません。 Cisco IOS XR ソフトウェアでの重複アドレス検出のシスコ実装では、64 ビット インターフェイス識別子から生成されるエニーキャスト アドレスまたはグローバル アドレスの一意性は確認されません。

IPv6 ルータ アドバタイズメント メッセージ

ルータ アドバタイズメント（RA）メッセージは、ICMP パケット ヘッダーのタイプ フィールドが値 134 であり、IPv6 ルータの設定済みの各インターフェイスへ定期的に送信されます。 ルータアドバタイズメント メッセージは全ノード マルチキャスト アドレスに送信されます（図 13：IPv6 ネイバー探索 - ルータ アドバタイズメント メッセージ, （231 ページ）を参照）。

図 *13*：*IPv6* ネイバー探索 - ルータ アドバタイズメント メッセージ



ルータ アドバタイズメント メッセージには、通常、次の情報が含まれています。

- ローカル リンク上のノードがその IPv6 アドレスの自動設定に使用できる 1 つ以上のオンリンク IPv6 プレフィックス
- アドバタイズメントに含まれる各プレフィックスのライフタイム情報
- 完成可能な自動設定のタイプ（ステートレスまたはステートフル）を示すフラグのセット
- デフォルト ルータ情報（アドバタイズメントを送信しているルータをデフォルト ルータとして使用する必要があるかどうか、および、その場合は、ルータがデフォルト ルータとして使用される秒単位の時間）
- ホストが発信するパケットで使用する必要のあるホップ リミットや MTU など、ホストに関する詳細情報

ルータアドバタイズメントは、ルータ送信要求メッセージへの応答としても送信されます。ICMP パケット ヘッダーの Type フィールドの値が 133 であるルータ送信要求メッセージは、システム始動時にホストによって送信されるため、ホストは次のスケジュールされたルータ アドバタイズメントメッセージを待機することなくすぐに自動設定できます。ルータ送信要求メッセージが通常システム起動時にホストによって送信される（ホストにユニキャスト アドレスが設定されていない）場合、ルータ送信要求メッセージの送信元アドレスは、通常は未指定の IPv6 アドレス（0:0:0:0:0:0:0:0）です。ホストに設定済みのユニキャストアドレスがある場合、ルータ送信要求メッセージを送信するインターフェイスのユニキャスト アドレスが、メッセージ内の送信元アドレスとして使用されます。 ルータ送信要求メッセージの宛先アドレスは、スコープがリンクである全ルータ マルチキャスト アドレスです。 ルータ送信要求に応答してルータ アドバタイズメントが送信される場合、ルータアドバタイズメントメッセージ内の宛先アドレスはルータ送信要求メッセージの送信元のユニキャスト アドレスです。

次のルータ アドバタイズメント メッセージ パラメータを設定できます。

- ルータ アドバタイズメント メッセージの定期的な時間間隔
- （特定のリンク上のすべてのノードで使用される）デフォルトルータとしてのルータの実用性を示す「ルータ ライフタイム」値
- 特定のリンクで使用されているネットワーク プレフィックス
- （特定のリンクで）ネイバー送信要求メッセージが再送信される時間の間隔
- ノードによってネイバーが到達可能である（特定のリンク上のすべてのノードで使用できる）と見なされるまでの時間

設定されたパラメータはインターフェイスに固有です。（デフォルト値を使用した）ルータアドバタイズメント メッセージの送信は、グローバル コンフィギュレーション モードで **ipv6 unicast-routing** コマンドが設定されている場合、イーサネットおよびFDDIインターフェイスで自動的にイネーブルになります。 その他のインターフェイス タイプの場合、ルータ アドバタイズメント メッセージの送信は、グローバル コンフィギュレーション モードで **no ipv6 nd suppress-ra** コマンドを使用して手動で設定する必要があります。 ルータ アドバタイズメント メッセージの送信は、インターフェイス コンフィギュレーション モードで **ipv6 nd suppress-ra** コマンドを使用して個々のインターフェイスでディセーブルにすることができます。

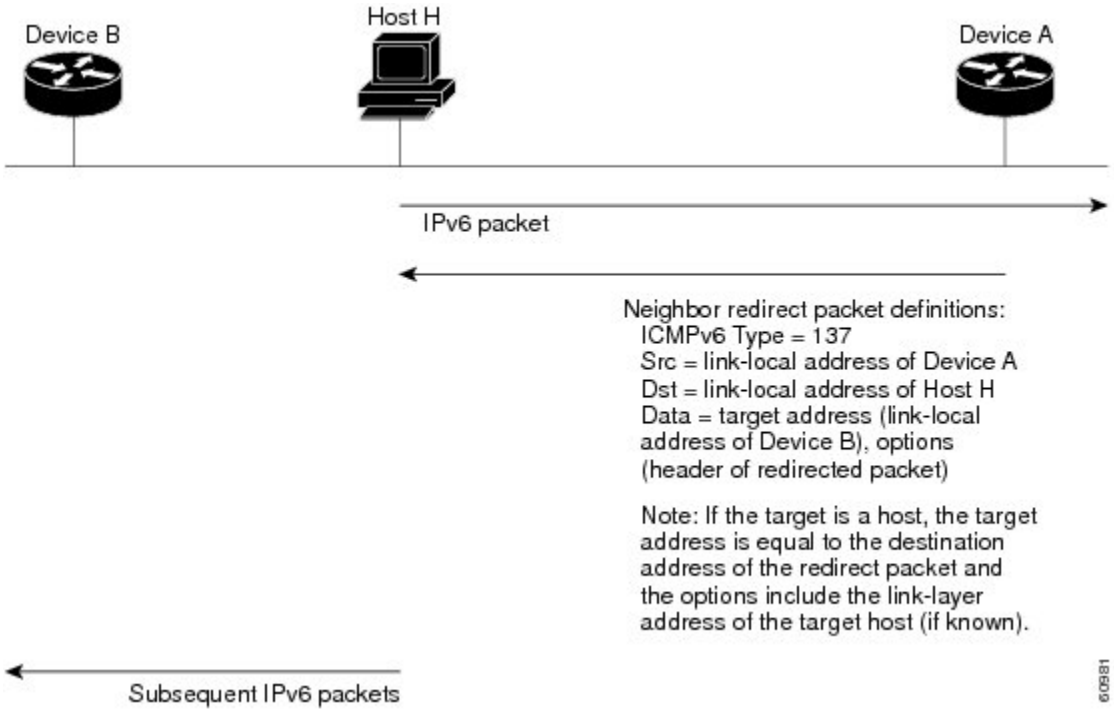
（注） ステートレス自動設定が正しく機能するには、ルータ アドバタイズメント メッセージでアドバタイズされたプレフィックス長が常に 64 ビットである必要があります。

IPv6 ネイバー リダイレクト メッセージ

ICMP パケット ヘッダーのタイプ フィールドの値 137 は、IPv6 ネイバー リダイレクト メッセージを示します。 ルータは、ネイバー リダイレクト メッセージを送信して、宛先へのパス上のよ

り適切なファーストホップ ノードをホストに通知します（図 14：IPv6 ネイバー探索 - ネイバー リダイレクト メッセージ，（233 ページ）を参照）。

図 *14*：*IPv6* ネイバー探索 - ネイバー リダイレクト メッセージ



（注） リダイレクト メッセージ内のターゲット アドレス（最終的な宛先）によって隣接ルータのリンクローカル アドレスが確実に識別されるように、ルータは各隣接ルータのリンクローカル アドレスを判断できる必要があります。 スタティック ルーティングの場合、ネクストホップ ルータのアドレスは、ルータのリンクローカルアドレスを使用して指定する必要があります。ダイナミック ルーティングの場合は、すべての IPv6 プロトコルが隣接ルータのリンクローカル アドレスを交換する必要があります。

パケットの転送後に、次の条件が満たされる場合、ルータはパケットの送信元にリダイレクト メッセージを送信する必要があります。

- パケットの宛先アドレスがマルチキャスト アドレスではない。
- パケットがルータにアドレッシングされていなかった。
- パケットが、そのパケットを受信したインターフェイスから送信されようとしている。
- ルータが、パケットにより適したファーストホップ ノードはパケットの送信元と同じリンク上にあると判断した。
- パケットの送信元アドレスが、同じリンク上のネイバーのグローバルIPv6アドレス、またはリンクローカル アドレスである。

ネイバー リダイレクト メッセージなどのすべての IPv6 ICMP エラー メッセージをルータが生成するレートを制限するには、**ipv6 icmp error-interval** グローバル コンフィギュレーション コマンドを使用します。これにより、リンク層の輻輳が最終的に低減されます。

（注） ルータはネイバー リダイレクト メッセージを受信してもそのルーティング テーブルを更新せず、ホストはネイバー リダイレクト メッセージを発信しません。

IPv6 の ICMP

IPv6 の Internet Control Message Protocol（ICMP）の機能は、IPv4 の ICMP と同じです。ICMP は、ICMP 宛先到達不能メッセージのようなエラー メッセージ、および ICMP エコー要求や応答メッセージのような情報メッセージを生成します。 また、IPv6 の ICMP パケットは、IPv6 ネイバー探索プロセス、パス MTU ディスカバリ、および Multicast Listener Discovery（MLD）プロトコル for IPv6 で使用されます。 MLD は、直接接続されているリンク上のマルチキャスト リスナー（特定のマルチキャストアドレスを宛先としたマルチキャストパケットを受信するために使用するノード）を検出するために IPv6 ルータで使用されます。 MLD は、バージョン 2 の Internet Group Management Protocol（IGMP）for IPv4 をベースとしています。

基本 IPv6 パケット ヘッダーの次ヘッダー フィールドの値 58 は、IPv6 ICMP パケットを示します。 IPv6 の ICMP パケットは、すべての拡張ヘッダーに続いて IPv6 パケットの末尾に配置される点でトランスポートレイヤ パケットに似ています。 IPv6 ICMP パケット内の ICMPv6 タイプ フィールドと ICMPv6 コード フィールドは、ICMP メッセージ タイプなどの IPv6 ICMP パケットの詳細を示します。チェックサム フィールドの値は、（送信側で計算し、受信側がチェックすることにより）IPv6 ICMP パケットと IPv6 疑似ヘッダーのフィールドから抽出されます。 ICMPv6 データ フィールドには、IP パケット処理に関連するエラー情報または診断情報が含まれます。

Address Repository Manager

IPv4 および IPv6 の Address Repository Manager（IPARM）は、システムで設定されたグローバル IP アドレスの一意性を強制適用し、IP アドレスを消費するアプリケーション プログラム インターフェイス（API）を使用して、グローバル IP アドレス情報（アンナンバード インターフェイス情報を含む）をルート プロセッサ（RP）およびラインカード（LC）上のプロセスに伝達します。

アドレス競合解決

競合解決には、競合データベースおよび競合セット定義という 2 つの部分があります。

競合データベース

IPARM では、グローバル競合データベースを保持します。 互いに競合する IP アドレスは、競合セットと呼ばれるリストに保持されます。 これらの競合セットは、グローバル競合データベースを構成します。

IP アドレスのセットは、そのセット内の少なくとも 1 つのプレフィックスが、同じセットに属する他のすべてのIPアドレスと競合する場合に、競合セットの一部であると見なされます。 たとえば、次の 4 つのアドレスは、単一の競合セットの一部です。

アドレス 1：10.1.1.1/16

アドレス 2：10.2.1.1/16

アドレス 3：10.3.1.1/16

アドレス 4：10.4.1.1/8

競合するIPアドレスが競合セットに追加されると、アルゴリズムによってそのセット全体が調べられ、そのセット内の最も優先度の高いアドレスが判別されます。

この競合ポリシー アルゴリズムは決定論的アルゴリズムであり、つまり、ユーザは、インターフェイス上のいずれのアドレスがイネーブルまたはディセーブルであるかがわかります。 イネーブルなインターフェイス上のアドレスは、その競合セットの最も優先度の高いアドレスとして宣言されます。

競合ポリシー アルゴリズムは、セット内の最も優先度の高い IP アドレスを判別します。

複数の IP アドレス

IPARM 競合処理アルゴリズムにより、複数の IP アドレスを 1 つのセット内でイネーブルにすることができます。 複数のアドレスが、最も高い優先度の IP アドレスになる場合があります。

interface GigabitEthernet 0/2/0/0：10.1.1.1/16

interface GigabitEthernet 0/3/0/0：10.1.1.2/8

interface GigabitEthernet 0/4/0/0：10.2.1.1/16

GigabitEthernet 0POS0/2/0/0 上の IP アドレスは、最も低いラック/スロット ポリシーに従って最も高い優先度として宣言され、イネーブルになります。 ただし、interface GigabitEthernet 0/4/0/0 上のアドレスは、現在の最も高い優先度の IP アドレスと競合しないため、GigabitEthernet 0/4/0/0 上のアドレスも同様にイネーブルになります。

競合セットの再帰的解決

次の例では、GigabitEthernet 0/2/0/0 のインターフェイス上のアドレスの優先度が最も高くなり、これは、最も低いラック/スロットであるためです。 ところが、現在は GigabitEthernet 0/4/0/0 上のアドレスも GigabitEthernet 0/5/0/0 上のアドレスも GigabitEthernet 0/2/0/0 上の最も高い優先度の IP アドレスと競合していません。 ただし、Gigabitethernet 0/4/0/0 上のアドレスと GigabitEthernet 0/5/0/0 上のアドレスが競合しているとすると、どちらがイネーブルになるでしょうか。 競合解決ソフトウェアは、現在イネーブルであるインターフェイスを、イネーブルのままである必要があるとして維持しようとします。 両方のインターフェイスがディセーブルの場合、ソフトウェアは、現在の競合ポリシーに基づいてアドレスをイネーブルにします。 GigabitEthernet 0/4/0/0 は、より低いラック/スロット上にあるため、イネーブルです。

interface GigabitEthernet 0/2/0/0：10.1.1.1/16

interface GigabitEthernet 0/3/0/0：10.1.1.2/8

interface GigabitEthernet 0/4/0/0：10.2.1.1/16

interface GigabitEthernet 0/5/0/0：10.2.1.2/16

接続ルートに対する Route-Tag のサポート

接続ルートに対する Route-Tag のサポート機能では、インターフェイスの IPv4 および IPv6 アドレスすべてにタグを付加します。 このタグは、IPv4 および IPv6 の管理エージェント（MA）から、IPv4 および IPv6 の Address Repository Manager（ARM）およびルーティング プロトコルに伝搬されるため、ユーザは、Routing Policy Language（RPL）スクリプトを使用してルート タグを調べることで、接続ルートの再配布を制御します。 これにより、ルート ポリシーのルート タグを確認して、一部のインターフェイスの再配布を回避できます。

このルート タグ機能は、ルート タグがポリシーに一致し、再配布を回避できるスタティック ルートおよび接続ルート（インターフェイス）ですでに利用可能です。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. 次のいずれかを実行します。
   - **ipv4 address** *ipv4-address mask* [**secondary**]
4. route-tag *[route-tag value]*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **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** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface POS 0/1/0/1`	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。

	コマンドまたはアクション	目的
ステップ 3	次のいずれかを実行します。 • **ipv4 address** *ipv4-address mask* [**secondary**] 例： `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 192.168.1.27 255.0.0.0`	インターフェイスのプライマリ（またはセカンダリ）IPv4アドレスアドレスを指定します。
ステップ 4	route-tag *[route-tag value]* 例： `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 192.168.1.27 255.0.0.0 route-tag 100`	設定されているアドレスに関連付けられているルート タグがそのアドレスにあることを指定します。 Route-Tag 値の範囲は、1 ～ 4294967295 です。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

ネットワーク スタック IPv4 および IPv6 の実装方法

ここでは、次の手順について説明します。

ネットワーク インターフェイスへの IPv4 アドレスの割り当て

このタスクでは、IPv4 アドレスを個々のネットワーク インターフェイスに割り当てます。

IPv4 アドレス

IP を設定するための基本的かつ必須のタスクは、IPv4 アドレスをネットワーク インターフェイスに割り当てることです。 こうすることで、インターフェイスがイネーブルになり、IPv4 を使用するこれらのインターフェイスでホストとの通信が可能になります。 IP アドレスは IP データグラムの送信先を特定します。 インターフェイスには、1 つのプライマリ IP アドレスと複数のセカンダリアドレスを設定できます。 ソフトウェアにより生成されるパケットは、必ずプライマリ IPv4 アドレスを使用します。 そのため、セグメントのすべてのネットワーキングデバイスは、同じプライマリ ネットワーク番号を共有する必要があります。

このタスクに関連付けられているのは、IP アドレスのサブネット化およびマスキングに関する決定です。 マスクで、IP アドレス中のネットワーク番号を示すビットが識別できます。 マスクを使用してネットワークをサブネット化した場合、そのマスクはサブネット マスクと呼ばれます。

（注） シスコでは、ネットワーク フィールドに対して左寄せの連続ビットを使用するネットワーク マスクのみをサポートしています。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ipv4 address** *ipv4-address mask* [**secondary**]
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
5. show ipv4 interface

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 gigabitethernet 0/1/0/1`	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	**ipv4 address** *ipv4-address mask* [**secondary**]
	
	例：
	`RP/0/RSP0/CPU0:router(config-if)# ipv4 address 192.168.1.27 255.0.0.0`
	`RP/0/RSP0/CPU0:router(config-if)# ipv4 address 192.168.1.27/8`
		インターフェイスに対するプライマリ IPv4 アドレスまたはセカンダリ IPv4 アドレスを指定します。
		• 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、1 に等しい各ビットが、ネットワーク アドレスに属した対応するアドレス ビットを意味することを示します。
		• ネットワーク マスクは、スラッシュ（/）および数字、つまり、プレフィックス長として示される場合もあります。プレフィックス長は、アドレスの高次の連続ビットのうち、何個がプレフィックス（アドレスのネットワーク部分）を構成しているかを指定する 10 進数値です。 スラッシュは 10 進数値の前に置かれ、IP アドレスとスラッシュの間にスペースは入りません。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		◦ **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 ipv4 interface 例： `RP/0/RSP0/CPU0:router# show ipv4 interface`	（任意）IPv4 用に設定されたインターフェイスの使用可能性ステータスを表示します。

IPv4 仮想アドレス

IPv4 仮想アドレスを設定することにより、いずれのルートプロセッサ（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、RP フェールオーバー状況間で維持されます。 このようにするには、仮想 IPv4 アドレスが、両方の RP の管理イーサネット インターフェイスで共通 IPv4 サブネットを共有する必要があります。

vrf キーワードは、VRF 単位の仮想アドレスをサポートします。

use-as-src-addr キーワードを使用すると、管理アプリケーションのために、ループバック インターフェイスを送信元インターフェイス（つまり、更新送信元）として設定する必要がなくなります。更新送信元が設定されていない場合、トランスポート プロセス（TCP、UDP、raw_ip）は、管理アプリケーションを使用して適切な送信元アドレスを選択できます。 トランスポート プロセスは、FIB を参照して、適切な送信元アドレスを選択します。 管理イーサネットの IP アドレスが送信元アドレスとして選択されており、**use-as-src-addr** キーワードが設定されている場合、トランスポートでは、管理イーサネットの IP アドレスを関連する仮想 IP アドレスに置き換えます。 この機能は、RP スイッチオーバー全体で機能します。 **use-as-src-addr** が設定されていない場合、トランスポートで選択された送信元アドレスはフェールオーバー後に変更される可能性があり、NMS ソフトウェアがこの状況を管理できなくなるおそれがあります。

（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。

IPv6 アドレッシングの設定

このタスクでは、IPv6 アドレスを個々のルータ インターフェイスに割り当て、ルータ上で IPv6 トラフィックのグローバルな転送を可能にします。 デフォルトでは、IPv6 アドレスは設定されていません。

（注） **ipv6 address** コマンドの *ipv6-prefix* 引数は、RFC 2373 に記載された形式にする必要があり、16 ビット値をコロンで区切った 16 進でアドレスを指定します。

ipv6 address コマンドの **/prefix-length** 引数は 10 進数の値で、プレフィックスを構成しているアドレスの連続する上位ビット数（アドレスのネットワーク部）を指定します。10 進値の前にはスラッシュが必要です。

ipv6 address link-local コマンドの *ipv6-address* 引数は、RFC 2373 に記載された形式にする必要があり、16 ビット値をコロンで区切った 16 進でアドレスを指定します。

ネットワーク インターフェイスへの複数の IP アドレスの割り当て

このタスクでは、複数の IP アドレスをネットワーク インターフェイスに割り当てます。

セカンダリ IPv4 アドレス

Cisco IOS XR ソフトウェアは、インターフェイスごとに複数の IP アドレスをサポートしています。 セカンダリ アドレスは無制限に指定できます。 セカンダリ IP アドレスは、さまざまな状況で使用できます。 次に、一般的な使用状況を示します。

- 特定のネットワーク セグメントに十分なホスト アドレスがない場合があります。たとえば、サブネット化により、論理サブネットあたり最大 254 のホストを使用できますが、1 つの物理サブネットでは、300 のホスト アドレスが必要になるとします。 ルータまたはアクセス サーバでセカンダリ IP アドレスを使用すると、2 つの論理サブネットで 1 つの物理サブネットを使用できます。
- 多くの旧式ネットワークは、レベル 2 ブリッジを使用して構築され、サブネット化されませんでした。セカンダリ アドレスは、慎重に使用することで、サブネット化されたルータベー

ス ネットワークへの移行に役立ちます。 旧式のブリッジ セグメントのルータで、そのセグメントに複数のサブネットがあることを簡単に認識されるようにできます。

- 1 つのネットワークの 2 つのサブネットは、別の方法で、別のネットワークにより分離できる場合があります。別のネットワークによって物理的に分離された複数のサブネットから、セカンダリ アドレスを使用して、1 つのネットワークを作成できます。 このような場合、最初のネットワークは、2 番めのネットワークの上に拡張されます。つまり、上の階層となります。 サブネットは、同時に複数のアクティブなインターフェイス上に表示できないことに注意してください。

（注） ネットワーク セグメント上の任意のルータがセカンダリ IPv4 アドレスを使用した場合、同一のセグメント上にある他のルータもすべて、同一のネットワークまたはサブネットからセカンダリ アドレスを使用する必要があります。

注意 ネットワーク セグメント上のセカンダリ アドレスの使用に矛盾があると、ただちにルーティング ループが引き起こされる可能性があります。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ipv4 address** *ipv4-address mask* [**secondary**]
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **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** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/3`	インターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**ipv4 address** *ipv4-address mask* [**secondary**] 例： `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 192.168.1.27 255.255.255.0 secondary`	設定されているアドレスが、セカンダリ IPv4 アドレスであることを指定します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

IPv4 および IPv6 プロトコル スタックの設定

このタスクでは、IPv4 と IPv6 の両方のプロトコル スタックをサポートするようにシスコのネットワーク デバイスのインターフェイスを設定します。

シスコのネットワーク デバイスのインターフェイスが IPv4 アドレスと IPv6 アドレスの両方で設定されている場合、インターフェイスは IPv4 トラフィックと IPv6 トラフィックの両方を転送します。インターフェイスは、IPv4 ネットワークと IPv6 ネットワークの両方でデータを送受信できます。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ipv4 address** *ip-address mask* [**secondary**]
4. **ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**]
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **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** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/1`	インターフェイスのタイプおよび番号を指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ipv4 address** *ip-address mask* [**secondary**] 例： `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 192.168.99.1 255.255.255.0`	インターフェイスに対するプライマリ IPv4 アドレスまたはセカンダリ IPv4 アドレスを指定します。
ステップ 4	**ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**] 例： `RP/0/RSP0/CPU0:router(config-if)# ipv6 address 2001:0DB8:c18:1::3/64`	インターフェイスに割り当てられた IPv6 アドレスを指定し、インターフェイスでの IPv6 処理をイネーブルにします。 • スラッシュ記号（/）は、*prefix-length* の前に置かれ、*ipv6-prefix* とスラッシュ記号の間にスペースは入りません。
ステップ 5	次のいずれかのコマンドを使用します。 • **end** • **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]:`

	コマンドまたはアクション	目的

	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		◦ **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。

アンナンバード インターフェイス上での IPv4 処理のイネーブル化

このタスクでは、アンナンバード インターフェイス上での IPv4 処理をイネーブルにします。

アンナンバード インターフェイス上での IPv4 処理

ここでは、明示的なIPアドレスをインターフェイスに割り当てることなく、IPv4ポイントツーポイント インターフェイスをイネーブルにするプロセスについて説明します。 アンナンバード インターフェイスがパケットを生成する場合（たとえば、ルーティング アップデートのため）は必ず、IP パケットの送信元アドレスとして指定したインターフェイスのアドレスが使用されます。また、アンナンバードインターフェイスを介してアップデートを送信するルーティングプロセスを判別する場合、指定されたインターフェイスのアドレスが使用されます。 その制限を次に示します。

- High-Level Data Link Control（HDLC）、PPP、およびフレーム リレーのカプセル化を使用するシリアル インターフェイスには、アンナンバードを設定できません。 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もアンナンバードを設定できますが、そのインターフェイスはポイントツーポイント サブインターフェイスである必要があります。
- インターフェイスがIPアドレスを持たないため、インターフェイスがアップ状態かどうかを判断するために **ping** EXEC コマンドは使用できません。 簡易ネットワーク管理プロトコル（SNMP）は、インターフェイス ステータスのリモートでのモニタリングに使用できます。
- IPセキュリティ オプションは、アンナンバードインターフェイス上でサポートできません。

Intermediate System-to-Intermediate System（IS-IS）をシリアル回線全体で設定する場合、シリアル インターフェイスをアンナンバードとして設定し、それにより、各インターフェイス上で IP アドレスは必須ではないことを規定している RFC 1195 に準拠することができます。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ipv4 unnumbered** *interface-type interface-instance*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **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** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/1`	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	**ipv4 unnumbered** *interface-type interface-instance* 例： `RP/0/RSP0/CPU0:router(config-if)# ipv4 unnumbered loopback 5`	明示的な IPv4 アドレスをインターフェイスに割り当てることなく、ポイントツーポイント インターフェイス上での IPv4 処理をイネーブルにします。 • 指定したインターフェイスは、別のアンナンバードインターフェイスではなく、任意の IP アドレスを持つ、ルータの別のインターフェイスの名前である必要があります。 • *interface-type* および *interface-instance* 引数で指定されたインターフェイスは、イネーブルにされている必要があります（**show interfaces** コマンド出力に「up」と表示）。
ステップ 4	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **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。

ICMP レート制限の設定

このタスクでは、IPv4 または IPv6 の ICMP レート制限の設定方法について説明します。

IPv4 ICMP レート制限

IPv4 ICMP レート制限機能では、IPv4 ICMP 宛先到達不能メッセージが生成されるレートを制限します。 Cisco IOS XR ソフトウェアは、通常の宛先到達不能メッセージ用と DF 宛先到達不能メッセージ用の 2 つのタイマーを保守します。 これらは同じ時間制限およびデフォルトを共有します。 **DF** キーワードが設定されていない場合、**icmp ipv4 rate-limit unreachable** コマンドによって、DF 宛先到達不能メッセージの時間値が設定されます。 **DF** キーワードが設定されている場合、その時間値は、通常の宛先到達不能メッセージの時間値とは無関係のままになります。

IPv6 ICMP レート制限

IPv6 ICMP レート制限機能によって、IPv6 ICMP エラー メッセージがネットワークへ送信されるレートを制限するためのトークン バケット アルゴリズムが実装されます。 IPv6 ICMP レート制限の初期の実装では、エラー メッセージ間に固定の間隔が定義されていましたが、traceroute などの

一部のアプリケーションでは、間断なく送信される要求のグループへの返信が必要になる場合があります。 エラー メッセージ間の固定間隔は、traceroute などのアプリケーションで動作するのに十分な柔軟性がなく、アプリケーションが失敗する原因となることがあります。 トークン バケット方式を実装すると、複数のトークンを仮想バケットに格納できます。トークンごとに 1 つのエラー メッセージを送信できます。 バケットに格納できるトークンの最大数を指定でき、エラー メッセージが送信されるたびに 1 つのトークンがバケットから削除されます。 一連のエラー メッセージが生成された場合は、バケットが空になるまでエラー メッセージを送信できます。トークンのバケットが空になると、新しいトークンがバケットに配置されるまで、IPv6 ICMP エラー メッセージは送信されません。 トークン バケット アルゴリズムは、レート制限の平均時間間隔を増やさず、固定時間間隔方式よりも柔軟性が高くなります。

手順の概要

1. **configure**
2. 次のいずれかを実行します。
   - **icmp ipv4 rate-limit unreachable** [**DF**] *milliseconds*
   - **ipv6 icmp error-interval** *milliseconds* [*bucketsize*]
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
4. 次のいずれかを実行します。
   - **show ipv4 traffic** [**brief**]
   - **show ipv6 traffic** [**brief**]

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	次のいずれかを実行します。 • **icmp ipv4 rate-limit unreachable** [**DF**] *milliseconds* • **ipv6 icmp error-interval** *milliseconds* [*bucketsize*]	IPv4 ICMP 宛先到達不能メッセージが生成されるレートを制限します。 • **DF** キーワードは、コード 4 フラグメンテーションが必要で、データ フラグメンテーション（DF）が設定されているときに、ICMP 宛先到達不能メッセージの IP ヘッダーに指定されている

	コマンドまたはアクション	目的

	例：
	`RP/0/RSP0/CPU0:router(config)# icmp ipv4 rate-limit unreachable 1000`
	または
	`RP/0/RSP0/CPU0:router(config)# ipv6 icmp error-interval 50 20`
		ように、ICMP宛先到達不能メッセージが送信されるレートを制限します。
		• *milliseconds* 引数では、ICMP 宛先到達不能メッセージを送信する間隔を指定します。
		または
		IPv6 ICMP エラー メッセージの間隔とバケット サイズを設定します。
		• *milliseconds* 引数では、トークンがバケットに追加される間隔を指定します。
		• オプションの *bucketsize* 引数では、バケットに格納されるトークンの最大数を定義します。

ステップ 3	次のいずれかのコマンドを使用します。
	• **end**
	• **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 4	次のいずれかを実行します。 • **show ipv4 traffic** [**brief**] • **show ipv6 traffic** [**brief**]	（任意）ICMP到達不能情報を含む、IPv4 トラフィックに関する統計情報を表示します。 • **brief** キーワードを使用して、IPv4 および ICMPv4 のトラフィック統計情報のみを表示します。

	コマンドまたはアクション	目的

	例：
	`RP/0/RSP0/CPU0:router# show ipv4 traffic`
	または
	`RP/0/RSP0/CPU0:router# show ipv6 traffic`
		または
		（任意）IPv6 ICMP レート制限カウンタを含む、IPv6 トラフィックに関する統計情報を表示します。
		• **brief** キーワードを使用して、IPv6 および ICMPv6 のトラフィック統計情報のみを表示します。

IPARM 競合解決の設定

このタスクでは、IP Address Repository Manager（IPARM）アドレス競合解決のパラメータを設定します。

静的ポリシー解決

静的ポリシー解決の設定により、新しいアドレス設定が現在実行中のインターフェイスに影響するのを防ぎます。

手順の概要

1. **configure**
2. {**ipv4** | **ipv6**} **conflict-policy static**
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **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。

	コマンドまたはアクション	目的

ステップ 2	{**ipv4** | **ipv6**} **conflict-policy static**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# ipv4 conflict-policy static`
	または
	`RP/0/RSP0/CPU0:router(config)# ipv6 conflict-policy static`
		競合ポリシーを静的に設定します。つまり、新しいインターフェイス アドレスが現在実行中のインターフェイスに影響するのを防ぎます。

ステップ 3	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

最長プレフィックス アドレス競合解決

この競合解決ポリシーでは、最も長いプレフィックス長を持つIPアドレスに最も高い優先度を付与することを試みます。

手順の概要

1. **configure**
2. **{ ipv4 | ipv6 } conflict-policy longest-prefix**
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**{ ipv4 | ipv6 } conflict-policy longest-prefix**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# ipv4 conflict-policy longest-prefix`
	または
	`RP/0/RSP0/CPU0:router(config)# ipv6 conflict-policy longest-prefix`
		競合ポリシーを最長プレフィックスに設定します。つまり、競合セット内の、現在実行中のインターフェイスの最長プレフィックスアドレスと競合しないすべてのアドレスは同様に実行することが許可されます。

ステップ 3	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		• 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。

最大 IP アドレス競合解決

この競合解決ポリシーでは、最大値を持つ IP アドレスに最も高い優先度を付与することを試みます。

手順の概要

1. **configure**
2. **{ ipv4 | ipv6 } conflict-policy highest-ip**
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**{ ipv4 | ipv6 } conflict-policy highest-ip**
	例：
	`RP/0/RSP0/CPU0:router(config)# ipv4 conflict-policy highest-ip`
	または
	`RP/0/RSP0/CPU0:router(config)# ipv6 conflict-policy highest-ip`
		競合ポリシーを最も高い IP 値に設定します。つまり、値が最大の IP アドレスが優先されます。

ステップ 3	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **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。

総称ルーティング カプセル化

総称ルーティング カプセル化（GRE）トンネリング プロトコルでは、カプセル化によって、1 つのプロトコルから別のプロトコルにパケットを転送する、簡易で一般的なアプローチを提供します。 転送する必要のあるパケットは、まず GRE ヘッダーでカプセル化され、さらに IPv4 や IPv6 などの別のプロトコルでカプセル化されてから、宛先に転送されます。

一般的な GRE カプセル化パケットには次のものが含まれます。

- 配信ヘッダー
- GRE ヘッダー
- ペイロード パケット

カプセル化され、宛先に送信する必要があるパケットがシステムに存在します。 これが、ペイロード パケットです。 ペイロードは、まず GRE パケットにカプセル化されます。 この GRE パケットは、次に別のプロトコルでカプセル化されてから、転送されます。 この外部プロトコルは、配信プロトコルと呼ばれます。

（注） IPv4 が GRE ペイロードとして実行される場合、Protocol Type フィールドは 0x800 に設定されている必要があります。

配信プロトコルまたはペイロード プロトコルあるいはその両方としての IPv6 は、現在配布されている GRE バージョンには含まれていません。

GRE トンネル上の IPv4 転送

GREトンネル上をトンネリングされるパケットは、通常のIPパケットとしてルータに入ります。このパケットは、この IP パケットの宛先アドレスを使用して転送（ルーティング）されます。Equal Cost Multi Path（ECMP）シナリオでは、出力インターフェイスや隣接は、プラットフォーム固有の L3 ロード バランス（LB）ハッシュに基づいて選択されます。 CRS のような 2 段階の転送プラットフォームの場合、選択した出力インターフェイスの受信隣接を使用して、そのインターフェイスをホスティングする出力ラインカードにパケットを送信します。 選択した出力インターフェイスが GRE インターフェイスである場合、入力ラインカードでは、GRE トンネル宛先への到達に使用できる実際の物理インターフェイスを決定する必要があります。 このために、2 番めのルーティング（転送）の決定が、（L3 ロード バランス ハッシュが物理インターフェイスを決定するために再度適用される）GREトンネル宛先アドレスに基づいて行われます。出力物理インターフェイスが判明すると、パケットは、GRE ヘッダーでまずカプセル化され、続いて物理インターフェイスのL2書き換えヘッダーでカプセル化された後に、そのインターフェイスから送信されます。 GRE カプセル化パケットがリモート トンネル エンドポイント ルータに到達した後、GRE パケットのカプセル化が解除されます。 外側の IP ヘッダーの宛先アドレスのルックアップ（トンネル宛先アドレスと同じ）では、入力ラインカード上のローカル アドレス（受信）エントリを検出します。

GREカプセル化解除の最初の手順は、GREパケットがルータに入ることを許可する前に、トンネルの送信元（外側の IP ヘッダーの送信元 IP アドレスと同じ）とトンネルの宛先（外側の IP ヘッダーの宛先IPアドレスと同じ）の組み合わせに基づいてトンネルエンドポイントが適格であるか調べることです。受信したパケットは、トンネルアドミタンス認定チェックに失敗すると、カプセル化解除ルータによってドロップされます。 トンネル アドミタンス チェックに成功すると、カプセル化解除により、外側の IP ヘッダーと GRE ヘッダーがパケットから取り除かれ、次に内部ペイロード パケットの処理が通常のパケットとして開始されます。

トンネル エンドポイントが、IPv4 パケットをペイロードとして持つ GRE パケットをカプセル化解除する場合、IPv4 ペイロード パケット内の宛先アドレスを使用してそのパケットを転送し、ペイロード パケットの TTL が減少する必要があります。 そのようなパケットを転送する場合は注意する必要があります。ペイロードパケットの宛先アドレスがパケットのエンカプスレータ（トンネルの反対側など）である場合、ループが発生する可能性があります。 この場合、そのパケットを廃棄する必要があります。

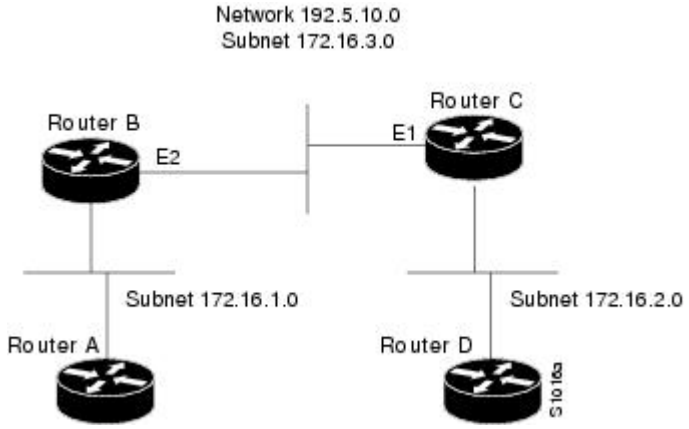
ネットワーク スタック IPv4 および IPv6 の実装の設定例

ここでは、次の設定例について説明します。

分離されたサブネットからのネットワークの作成：例

次の例では、ネットワーク 172.16.0.0 のサブネット 1 および 2 が、図 15：分離されたサブネットからのネットワークの作成，（256 ページ）に示すように、バックボーンによって分離されています。 これら 2 つのネットワークは、セカンダリ アドレスを使用して同じ論理ネットワークに入れられます。

図 *15*：分離されたサブネットからのネットワークの作成



次に、ルータ B および C の設定例を示します。

ルータ B の設定

```
configure
interface gigabitethernet 0/0/0/2
ipv4 address 192.5.10.1 255.255.255.0
ipv4 address 172.16.3.1 255.255.255.0 secondary
```

ルータ C の設定

```
configure
interface gigabitethernet 0/0/0/1
ipv4 address 192.5.10.2 255.255.255.0
ipv4 address 172.16.3.2 255.255.255.0 secondary
```

アンナンバード インターフェイスの割り当て：例

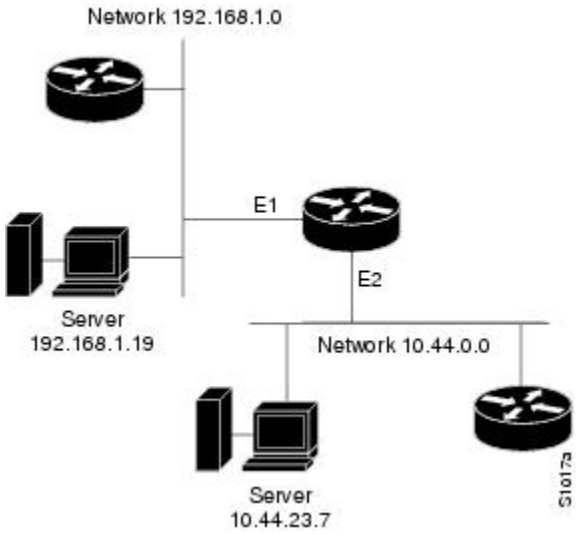
次の例では、2 番めのインターフェイス（GigabitEthernet 0/1/0/1）にループバック インターフェイス 0 のアドレスが付与されています。 このループバック インターフェイスはアンナンバードです。

```
interface loopback 0
 ipv4 address 192.168.0.5 255.255.255.0
interface gigabitethernet 0/1/0/1
 ipv4 unnumbered loopback 0
```

ヘルパー アドレスの設定：例

次の例では、1 つのルータがネットワーク 192.168.1.0 上にあり、別のルータはネットワーク 10.44.0.0 上にあり、いずれかのネットワーク セグメント上のホストからの IP ブロードキャストが両方のサーバに到達できるようにする必要があります。図 16：IP ヘルパー アドレス，（257 ページ）に、ネットワーク 10.44.0.0 をネットワーク 192.168.1.0 に接続するルータを設定する方法を示します。

図 *16*：*IP* ヘルパー アドレス



次に、設定例を示します。

```
!
interface gigabitethernet 0/0/0/1
 ipv4 helper-address 10.44.23.7
interface gigabitethernet 0/0/0/2
 ipv4 helper-address 192.168.1.19
```

VRF big モードの設定

次のタスクを実行して、VRF の big モードを設定します。

手順の概要

1. **configure**
2. **vrf** *vrf-name*
3. **mode big**
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**vrf** *vrf-name*
	
	例：
	
	`RP/0/RSP0/CPU0:router(config)# vrf v1`
	`RP/0/RSP0/CPU0:router(config-vrf)#`
		VRF コンフィギュレーション モードを開始します。

ステップ 3	**mode big**
	
	例：
	
	`RP/0/RSP0/CPU0:router(config-vrf)# mode big`
	`RP/0/RSP0/CPU0:router(config-vrf)#`
		対応する VRF の big モードを開始します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		• 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。

その他の参考資料

ここでは、ネットワーク スタック IPv4 および IPv6 の実装に関する関連資料について説明します。

関連資料

関連項目	参照先
アドレス解決の設定タスク	このマニュアルの「*ARP* の設定」の章。
ホスト名の IP アドレスへのマッピング	『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*Host Services and Applications Commands*」の章
ネットワーク スタック IPv4 および IPv6 のコマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*Network Stack IPv4 and IPv6 Commands*」の項

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 IOS XR ソフトウェアを使用している MIB を特定してダウンロードするには、次の URL にある Cisco MIB Locator を使用し、[Cisco Access Products] メニューからプラットフォームを選択します。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

第 9 章

トランスポートの設定

この章では、ノンストップルーティング（NSR）、伝送制御プロトコル（TCP）、およびユーザデータグラム プロトコル（UDP）トランスポート（Cisco ASR 9000 シリーズ アグリゲーション サービス ルータ上について説明します。

特別な要件に基づいて NSR、TCP、または UDP の値を調整する必要がある場合は、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*Transport Stack Commands*」を参照してください。

（注） この章に記載されているトランスポートコンフィギュレーションコマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』を参照してください。 この章で使用される他のコマンドの説明については、コマンド リファレンスのマスター索引を参照するか、またはオンラインで検索してください。

次の製品における NSR、SCTP、TCP、UDP、および UDP RAW トランスポートの設定の機能履歴：Cisco ASR 9000 シリーズ ルータ

リリース	変更内容
リリース 3.7.2	この機能が導入されました。

NSR、TCP、UDP トランスポートの設定の前提条件

次に、NSR、TCP、UDP トランスポートを実装するための前提条件を示します。

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。

NSR、TCP、UDP トランスポートの設定について

NSR、TCP、および UDP トランスポートを設定するには、次の概念を理解しておく必要があります。

NSR の概要

ノンストップ ルーティング（NSR）は、Open Shortest Path First（OSPF）およびラベル配布プロトコル（LDP）プロトコル用に、次のイベントのために提供されています。

- ルート プロセッサ（RP）フェールオーバー
- OSPF、LDP、または TCP でのプロセスの再開
- インサービス ソフトウェア アップグレード（ISSU）

RP フェールオーバーの場合、NSR は、TCP およびアプリケーション（OSPF または LDP）の両方に対して実現できます。

NSR は、ルーティング プロトコルのハイ アベイラビリティ（HA）を実現するための方法です。 RP フェールオーバーの後、TCP 接続およびルーティング プロトコル セッションは、ピアに通知されることなく、アクティブ RP からスタンバイ RP に移行されます。 現在、スタンバイ RP がアクティブになると、セッションが終了し、スタンバイ RP 上で実行されているプロトコルによってセッションが再確立されます。 グレースフル リスタート（GR）拡張を NSR の代わりに使用して、RP フェールオーバー時のトラフィック損失を回避できますが、GR にはいくつかの短所があります。

nsr process-failures switchover コマンドを使用して、アクティブ TCP またはアクティブ LDP の再起動時に RP フェールオーバーがリカバリ アクションとして使用されるようにします。 スタンバイ TCP または LDP が再起動すると、スタンバイ インスタンスが起動し、セッションが再同期化されるまで NSR 機能は失われますが、セッションはダウンしません。 アクティブ OSPF のプロセス障害の場合は、障害管理ポリシーが使用されます。 詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*』の「*Implementing OSPF*」を参照してください。

TCP の概要

TCP は、2 つのコンピュータ システムがデータを転送するために交換する、データおよび確認応答の形式が指定されたコネクション型プロトコルです。 また、TCP では、データを正しく到達させるために、コンピュータが使用する手順も指定されています。 TCP では、アプリケーション プログラム間の着信トラフィックのすべての逆多重化を処理するため、TCP を使用すると、1 つのシステム上の複数のアプリケーションが同時に通信できます。

TCP あるいは UDP 以外のすべての IP プロトコルは、RAW プロトコルと考えられています。

ほとんどのサイトでは、TCP、UDP、およびRAW トランスポートのデフォルト設定を変更する必要はありません。

UDP の概要

ユーザ データグラム プロトコル（UDP）は、IP ファミリに属するコネクションレス型トランスポートレイヤ プロトコルです。 UDP は、ネットワーク ファイル システム（NFS）、簡易ネットワーク管理プロトコル（SNMP）、ドメイン ネーム システム（DNS）、TFTP などの一般的なアプリケーション層プロトコルのための、トランスポート プロトコルです。

TCP、UDP 以外のすべての IP プロトコルは、RAW プロトコルとして知られています。

ほとんどのサイトでは、TCP、UDP、およびRAW トランスポートのデフォルト設定を変更する必要はありません。

NSRのリカバリアクションとしてのフェールオーバーの設定方法

ここでは、次の手順について説明します。

NSR のリカバリ アクションとしてのフェールオーバーの設定

このタスクでは、アクティブなインスタンスの障害を処理するリカバリ アクションとしてフェールオーバーを設定できます。

アクティブな TCP、またはアクティブな TCP の NSR クライアントが終了または再起動すると、TCPセッションはダウンします。 NSR の提供を継続するには、リカバリ アクションとしてフェールオーバーを設定する必要があります。 フェールオーバーが設定されている場合、アクティブな TCP またはアクティブなアプリケーション（LDP、OSPF など）が再起動または終了すると、スイッチオーバーが開始されます。

MPLS ラベル配布プロトコル（LDP）をNSR に設定する方法の詳細については、『*Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide*』を参照してください。

各 OSPF プロセスに対してプロセス レベル単位で NSR を設定する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。

手順の概要

1. **configure**
2. nsr process-failures switchover
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	nsr process-failures switchover 例： `RP/0/RSP0/CPU0:router(config)# nsr process-failures switchover`	ノンストップ ルーティング（NSR）を維持するために、アクティブなインスタンスをスタンバイ ルート プロセッサ（RP）または分散ルート プロセッサ（DRP）に切り替えるためのリカバリ アクションとしてフェールオーバーを設定します。

ステップ 3	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		◦**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。

その他の参考資料

ここでは、NSR、TCP、および UDP トランスポート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 ASR 9000 シリーズ ルータ のトランスポート スタック コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*Transport Stack Commands*」
Cisco ASR 9000 シリーズ ルータ の MPLS LDP コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『*Cisco ASR 9000 Series Aggregation Services Router MPLS Command Reference*』の「*MPLS Label Distribution Protocol Commands*」
Cisco ASR 9000 シリーズ ルータ の OSPF コマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『*Cisco ASR 9000 Series Aggregation Services Router Routing Command Reference*』の「*OSPF Commands*」
MPLS ラベル配布プロトコルの機能情報	『*Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide*』の「*Implementing MPLS Label Distribution Protocol*」
OSPF の機能情報	『*Cisco ASR 9000 Series Aggregation Services Router Routing Configuration Guide*』の「*Implementing OSPF*」

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
—	MIB を特定してダウンロードするには、次の URL にある Cisco MIB Locator を使用し、[Cisco Access Products] メニューからプラットフォームを選択します。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

第 10 章

VRRP の実装

仮想ルータ冗長プロトコル（VRRP）機能を使用すると、ファーストホップ IP ルータでの透過的なフェールオーバーが可能になり、ルータ グループが単一の仮想ルータを形成できるようになります。

（注） この章に記載されている VRRP コマンドの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』を参照してください。 この章で使用される他のコマンドの説明については、コマンド リファレンスのマスター索引を参照するか、またはオンラインで検索してください。

VRRP の実装の機能履歴

リリース	変更内容
リリース 3.7.2	この機能が導入されました。
リリース 3.9.0	• VRRP 用の BFD 機能が追加されました。 • MIB の VRRP サポート機能が追加されました。
リリース 4.1.0	IPv6 上の VRRP 機能が追加されました。

VRRP の実装の前提条件：Cisco IOS XR ソフトウェア

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。

VRRP の実装の制約事項： Cisco IOS XR ソフトウェア

次に、VRRP を実装する場合の制約事項を示します。

- ICMP リダイレクトはサポートされていません。

VRRP の実装について

Cisco IOS XR ソフトウェアで VRRP を実装するには、次の概念を理解しておく必要があります。

VRRP の概要

LAN クライアントは、動的プロセスまたは静的設定を使用して、特定のリモート宛先への最初のホップとなるルータを決定します。 次に、ダイナミック ルータ ディスカバリのクライアント例を示します。

- プロキシ ARP：クライアントはアドレス解決プロトコル（ARP）を使用して到達すべき宛先を取得します。ルータは独自の MAC アドレスで ARP 要求に応答します。
- ルーティング プロトコル：クライアントはダイナミック ルーティング プロトコルのアップデートを（ルーティング情報プロトコル（RIP）などから）受信し、独自のルーティング テーブルを形成します。
- IRDP（ICMP Router Discovery Protocol）クライアント：クライアントはインターネット制御メッセージ プロトコル（ICMP）ルータ ディスカバリ クライアントを実行します。

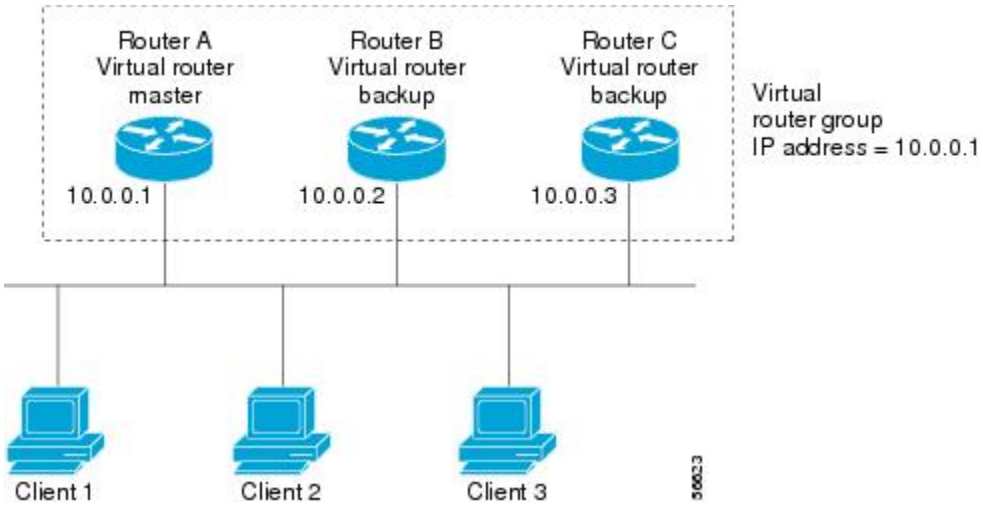
ダイナミック ディスカバリ プロトコルには、LAN クライアントにおいて、設定および処理のオーバーヘッドが発生するという短所があります。 また、ルータが機能を停止したときに、別のルータへの切り替え処理が遅くなる可能性があります。

ダイナミック ディスカバリ プロトコルの代わりに、クライアント上でデフォルト ルータをスタティックに設定することもできます。 このアプローチでは、クライアントの設定と処理は簡略化されますが、単一障害点が生じます。 デフォルト ゲートウェイで障害が発生した場合、LAN クライアントの通信はローカルIPネットワーク セグメントに限定され、ネットワークの他の部分から切り離されます。

仮想ルータ冗長プロトコル（VRRP）機能により、この静的設定の問題を解決できます。 VRRP は、ファーストホップ IP ルータの透過的なフェールオーバーを可能にするように設計された IP ルーティング冗長プロトコルです。 VRRP を使用すると、ルータのグループを 1 つの仮想ルータにすることができます。 これにより、仮想ルータをデフォルトゲートウェイとして使用するように、LAN クライアントを設定できます。 ルータのグループを表す仮想ルータは、*VRRP* グループとも呼ばれます。

例として、図 17：基本的な VRRP トポロジ, （269 ページ）に、VRRP が設定された LAN トポロジを示します。 この例では、ルータ A、B、および C は仮想ルータで構成される *VRRP* ルータ（VRRP を実行するルータ）です。 仮想ルータの IP アドレスは、ルータ A のインターフェイスに設定されたアドレス（10.0.0.1）と同じです。

図 *17*：基本的な *VRRP* トポロジ



仮想ルータはルータ A の物理インターフェイスの IP アドレスを使用するため、ルータ A はマスター仮想ルータのロールを担い、*IP* アドレス所有者 とも呼ばれます。ルータ A は、マスター仮想ルータとして、仮想ルータの IP アドレスを管理し、この IP アドレスに送信されたパケットの転送を行います。 クライアント 1 ～ 3 には、デフォルト ゲートウェイの IP アドレス 10.0.0.1 が設定されています。

ルータ B および C は、バックアップ仮想ルータとして機能します。 マスター仮想ルータに障害が発生すると、高いプライオリティが設定されているルータがマスター仮想ルータになり、LAN ホストに対して中断なくサービスが提供されます。 ルータ A は、回復すると、再びマスター仮想ルータになります。

複数の仮想ルータのサポート

ルータのインターフェイスには、最大 100 の仮想ルータを設定できます。 ルータ インターフェイスがサポートできる実際の仮想ルータの数は、次の要因によって異なります。

- ルータの処理能力
- ルータのメモリの能力
- 複数の MAC アドレスのルータ インターフェイス サポート

1 つのルータ インターフェイス上に複数の仮想ルータが設定されているトポロジでは、そのインターフェイスは 1 つ以上の仮想ルータのマスター、および 1 つ以上の仮想ルータのバックアップとして動作することができます。

VRRP ルータ プライオリティ

VRRP 冗長性スキームの重要な一面に、VRRP ルータ プライオリティがあります。 プライオリティにより、各 VRRP ルータが果たすロールと、マスター仮想ルータが機能を停止したときにどのようなことが起こるかが決定されます。

VRRP ルータが仮想ルータの IP アドレスと物理インターフェイスの IP アドレスのオーナーである場合には、このルータがマスター仮想ルータとして機能します。

IP アドレスのオーナーである VRRP ルータが存在しない場合は、VRRP ルータのプライオリティおよびプリエンプション設定の組み合わせにより、VRRP ルータがマスターとして機能するか、またはバックアップ仮想ルータとして機能するかが決まります。 デフォルトでは、最高のプライオリティを持つ VRRP ルータがマスターとして機能し、その他のすべてがバックアップとして機能します。 プライオリティにより、マスター仮想ルータが機能を停止した場合にマスター仮想ルータになる優先順位も決まります。 **vrrp priority** コマンドを使用して 1 ～ 254 の値を設定し、各バックアップ仮想ルータのプライオリティを設定できます。

たとえば、LAN トポロジのマスター仮想ルータであるルータ A が機能を停止した場合、選択プロセスが実行されて、バックアップ仮想ルータ B または C が引き継ぐかどうかが決定されます。 ルータ B とルータ C がそれぞれプライオリティ 101 と 100 に設定されている場合、プライオリティの高いルータ B がマスター仮想ルータになります。 ルータ B とルータ C が両方ともプライオリティ 100 に設定されている場合、IP アドレスがより高いバックアップ仮想ルータが選択されてマスター仮想ルータになります。

デフォルトでは、プリエンプティブ スキームがイネーブルになっており、使用可能になった高いプライオリティのバックアップ仮想ルータが、現在のマスター仮想ルータから引き継ぎます。 このプリエンプティブ スキームをディセーブルにするには、 **vrrp preempt disable** コマンドを使用します。 プリエンプションがディセーブルの場合、元のプライオリティがより高いマスターの障害時に、マスターになるように選択されたバックアップ仮想ルータは、元のマスター仮想ルータが回復し、再び使用可能になっても、マスターのままとなります。

VRRP のアドバタイズメント

マスター仮想ルータは、同じグループ内の他の VRRP ルータに VRRP アドバタイズメントを送信します。 アドバタイズメントでは、マスター仮想ルータのプライオリティと状態を伝えます。 VRRP アドバタイズメントは IP パケットにカプセル化され、VRRP グループに割り当てられた IP バージョン 4 マルチキャスト アドレスに送信されます。 アドバタイズメントは、デフォルトで 1 秒に 1 回送信されますが、この間隔は設定可能です。

VRRP の利点

VRRP の利点は、次のとおりです。

- 冗長性：VRRP により、複数のルータをデフォルト ゲートウェイ ルータとして設定できるようになるため、ネットワークに単一障害点が生じる可能性を低減できます。
- ロード シェアリング：LAN クライアントとの間のトラフィックを複数のルータで共有するように VRRP を設定できるため、利用可能なルータ間でより均等にトラフィックの負荷を分散できます。
- 複数の仮想ルータ：プラットフォームが複数の MAC アドレスをサポートする場合、VRRP は、ルータのインターフェイス上で最大 100 の仮想ルータ（VRRP グループ）をサポートします。 デフォルト タイマーについてはシステムごとの上限は 100 です。 複数の仮想ルータをサポートすることで、LAN トポロジ内で冗長化とロード シェアリングを実装できます。
- 複数の IP アドレス：仮想ルータは、セカンダリ IP アドレスを含む、複数の IP アドレスを管理できます。 そのため、イーサネット インターフェイスに複数のサブネットを設定した場合、サブネットごとに VRRP を設定できます。
- プリエンプション：VRRP の冗長性スキームにより、障害が発生したマスター仮想ルータを引き継いだバックアップ仮想ルータを、使用可能になった高いプライオリティのバックアップ仮想ルータに切り替えることができます。
- テキスト認証：簡易テキスト パスワードを設定して、仮想ルータを構成している VRRP ルータから受信した VRRP メッセージが認証されたことを確認できます。
- アドバタイズメント プロトコル：VRRP では、VRRP アドバタイズメントに、専用のインターネット割り当て番号局（IANA）規格マルチキャスト アドレス（224.0.0.18）を使用します。 このアドレッシング方式によって、マルチキャストを提供するルータ数が最小限になり、テスト機器でセグメント上の VRRP パケットを正確に識別できるようになります。 IANA では VRRP に IP プロトコル番号 112 を割り当てています。

VRRP の実装方法： Cisco IOS XR ソフトウェア

ここでは、次のタスクの手順を示します。

VRRP のカスタマイズ

VRRP の動作のカスタマイズはオプションです。 VRRP グループをイネーブルにするとすぐに、そのグループは動作を開始することに注意してください。 VRRP をカスタマイズする前に、VRRP グループをイネーブルにした場合、機能のカスタマイズが完了しないうちに、ルータがそのグループの制御をテイクオーバーし、マスター仮想ルータになる可能性があります。 このため、VRRP をカスタマイズする場合には、カスタマイズを行ってから VRRP をイネーブルにすることを推奨します。

以降の項では、VRRP 設定をカスタマイズする方法について説明します。

手順の概要

1. **configure**
2. **router vrrp**
3. **interface** *type interface-path-id*
4. **address-family** {**ipv4** | **ipv6**}
5. **vrrp** *vrid* **version** { *2* | *3* }
6. **text-authentication**
7. **accept-mode**{**disable**}
8. **priority** *priority*
9. **preempt** [**delay** *seconds*] [**disable**]
10. **timer** [**msec**] *interval* [**force**]
11. **track interface** *type instance interface-path-id* [*priority-decrement*]
12. **delay** [**minimum** *seconds*] [**reload** *seconds*]
13. 次のいずれかのコマンドを使用します。
    - **end**
    - **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	**router vrrp** 例：	VRRP コンフィギュレーション モードをイネーブルにします。

	コマンドまたはアクション	目的
	`RP/0/RSP0/CPU0:router(config)# router vrrp`	
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-vrrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスで VRRP インターフェイス コンフィギュレーション モードをイネーブルにします。
ステップ 4	**address-family** {**ipv4** | **ipv6**} 例： `RP/0/RSP0/CPU0:routerconfig-vrrp-if)# address-family ipv6`	IPv4 または IPv6 アドレス ファミリ サブモードを開始します。

ステップ 5	**vrrp** *vrid* **version** { *2* | *3* }
	例：
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)# vrrp 3 version 3`
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)#`
		仮想ルータ 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 6	**text-authentication** 例： `RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)# text-authentication`	（任意）VRRP を実行する他のルータから受信した仮想ルータ冗長プロトコル（VRRP）パケットに使用する簡易テキスト認証を設定します。 • VRRP パケットが別のルータから到着すると、その認証ストリングが、ローカル システムに設定されたストリングと比較されます。 ストリングが一致する場合、そのメッセージが受け入れられます。 一致しない場合、そのパケットは廃棄されます。 • グループ内のすべてのルータは、同じ認証ストリングで設定される必要があります。 • VRRP 認証をディセーブルにするには、**no text-authentication** コマンドを使用します。 （注） プレーンテキスト認証は、セキュリティに使用されることになっているのではありません。 それは、設定ミスのルータが VRRP に参加しないようにする方法を提供しているに過ぎません。

	コマンドまたはアクション	目的
ステップ 7	**accept-mode**{**disable**} 例： `RP/0/RSP0/CPU0:router# (config-vrrp-virtual-router)# accept-mode disable`	IPv4 または IPv6 アドレス ファミリ サブモードを開始します。
ステップ 8	**priority** *priority* 例： `RP/0/RSP0/CPU0:router# (config-vrrp-virtual-router)# priority 254`	（任意）仮想ルータのプライオリティを設定します。 • マスター ルータになるルータを制御するには、**priority** コマンドを使用します。 • **priority** コマンドは、ルータが仮想 IP アドレスのオーナーである間は無視されます。 • 仮想ルータのプライオリティを削除するには、**no priority** コマンドを使用します。
ステップ 9	**preempt** [**delay** *seconds*] [**disable**] 例： `RP/0/RSP0/CPU0:router# (config-vrrp-virtual-router)# preempt delay 15`	（任意）マスター仮想ルータ、および任意で、マスター ルータになる仮想 IP アドレス所有権をルータがアドバタイズするまでの時間（秒単位）を設定します。 • **preempt** コマンドを使用して、マスター ルータになるルータを制御します。 • **preempt** コマンドは、ルータが仮想 IP アドレスのオーナーである間は無視されます。 • （任意）**disable** キーワードを使用して、プリエンプションをディセーブルにします。 デフォルトを再設定（イネーブル）にするには、**no preempt** コマンドを使用します。
ステップ 10	**timer** [**msec**] *interval* [**force**] 例： `RP/0/RSP0/CPU0:router# (config-vrrp-virtual-router)# timer 4`	（任意）マスター ルータが連続してアドバタイズを発行する時間間隔を VRRP 仮想ルータで設定します。 • デフォルト値に戻すには、**no timer** コマンドを使用します。 （注） 他のベンダーとの相互運用時には、同じ VRRPv3 タイマーをすべての VRRP ルータに設定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 11	**track interface** *type instance interface-path-id* [*priority-decrement*] 例： `RP/0/RSP0/CPU0:router# (config-vrrp-virtual-router)# track interface TenGigE 0/0/CPU0/1 30`	（任意）インターフェイスをトラッキングするように VRRP を設定します。 • **no track interface** *type instance interface-path-id* [*priority-decrement*] コマンドを入力して、トラッキングをディセーブルにします。 • IP インターフェイスだけがトラッキングされます。 • トラッキングされるインターフェイスは、そのインターフェイス上の IP が立ち上がると起動します。 IP が立ち上がらないと、トランキングされるインターフェイスはダウンします。 • VRRP 仮想ルータの仮想ルータのプライオリティレベルを VRRP が変更できるように設定できます。インターフェイスの IP プロトコル状態がダウンした場合、またはインターフェイスがルータから削除された場合、バックアップ仮想ルータのプライオリティは、priority-decrement 引数内に指定された値により減少します。 インターフェイスの IP プロトコル状態が起動状態になると、プライオリティが元に戻ります。
ステップ 12	**delay** [**minimum** *seconds*] [**reload** *seconds*] 例： `RP/0/RSP0/CPU0:router# (config-vrrp-virtual-router) # delay minimum 2 reload 10`	（任意）ネットワークが安定する時間を確保し、リンクの起動後すぐに不要な状態変更がないように、インターフェイス起動時にステート マシンの起動を遅らせます。 リロード遅延は、最初のインターフェイス起動イベント後に適用される遅延です。 最小遅延は、後続の（インターフェイスがフラップする場合の）インターフェイス起動イベントに適用される遅延です。

ステップ 13	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		◦ **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。

VRRP のイネーブル化

以降の項で説明しているように、**address** コマンドを使用して、VRRP をインターフェイス上でイネーブルにします。

手順の概要

1. **configure**
2. router vrrp
3. **interface** type interface-path-id
4. **address-family ipv4**
5. **vrrp** *vrid* **version** { *2* | *3* }
6. **address** *address*
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	router vrrp 例： `RP/0/RSP0/CPU0:router(config)# router vrrp`	VRRP コンフィギュレーション モードをイネーブルにします。

ステップ 3	**interface** type interface-path-id
	例：
	`RP/0/RSP0/CPU0:router(config-vrrp)# interface TenGigE 0/2/0/1`
	`RP/0/RSP0/CPU0:router(config-vrrp-if)#`
		特定のインターフェイスで VRRP インターフェイス コンフィギュレーション モードをイネーブルにします。

ステップ 4	**address-family ipv4**
	例：
	`RP/0/RSP0/CPU0:routerconfig-vrrp-if)#`
	`address-family ipv4`
		IPv4 または IPv6 アドレス ファミリ サブモードを開始します。

ステップ 5	**vrrp** *vrid* **version** { *2* | *3* }
	例：
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)#`
	`vrrp 3 version 3`
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)#`
		仮想ルータ 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 6	**address** *address*
	例：
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)#`
	`address 2001:db8::/32`
		仮想ルータ冗長プロトコル（VRRP）をインターフェイスでイネーブルにし、仮想ルータの IP アドレスを指定します。
		• VRRP 設定を IP アドレス オーナーから削除してインターフェイスの IP アドレスをアクティブなままにしないことを推奨します。これは、LAN 上に重複する IP アドレスが作成されるためです。

	コマンドまたはアクション	目的
		• インターフェイス上の VRRP をディセーブルにして、仮想ルータの IP アドレスを削除するには、**no address** *address* コマンドを使用します。

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

VRRP の確認

show vrrp コマンドを使用して、1 つまたはすべての VRRP 仮想ルータの要約ステータスまたは詳細ステータスを表示します。

手順の概要

1. **show vrrp** [**ipv4** | **ipv6**] [**interface** *type instance interface-path-id* [*vrid*]] [**brief** | **detail** | **statistics** [**all**]]

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 vrrp** [**ipv4** | **ipv6**] [**interface** *type instance interface-path-id* [*vrid*]] [**brief** | **detail** | **statistics** [**all**]] 例： `RP/0/RSP0/CPU0:router # show vrrp`	1 つまたはすべての仮想ルータ冗長プロトコル（VRRP）仮想ルータの要約ステータスまたは詳細ステータスを表示します。 • インターフェイスが指定されない場合、すべての仮想ルータが表示されます。

VRRP 統計情報のクリア

clear vrrp statistics コマンドを使用して、指定の仮想ルータの全ソフトウェア カウンタをクリアします。

手順の概要

1. **clear vrrp statistics** [**ipv4** | **ipv6**] [**interface***type interface-path-id* [*vrid*]]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**clear vrrp statistics** [**ipv4** | **ipv6**] [**interface***type interface-path-id* [*vrid*]] 例： `RP/0/RSP0/CPU0:router# clear vrrp statistics`	指定の仮想ルータの全ソフトウェア カウンタをクリアします。 • インターフェイスが指定されない場合、すべての仮想ルータの統計情報が削除されます。

accept-mode の設定

次のタスクを実行して、VRRP 仮想アドレスのルートのインストールをディセーブルにします。

手順の概要

1. **configure**
2. **router vrrp**
3. **interface** *type interface-path-id*
4. **address-family** {**ipv4** | **ipv6**}
5. **vrrp** *vrid* **version** { *2* | *3* }
6. **accept-mode disable**
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router vrrp** 例： `RP/0/RSP0/CPU0:router(config)# router vrrp`	VRRP コンフィギュレーション モードをイネーブルにします。

ステップ 3	**interface** *type interface-path-id*
	例：
	`RP/0/RSP0/CPU0:router(config-vrrp)# interface TenGigE 0/2/0/1`
	`RP/0/RSP0/CPU0:router`
		特定のインターフェイスで VRRP インターフェイス コンフィギュレーション モードをイネーブルにします。

ステップ 4	**address-family** {**ipv4**	**ipv6**}
		例：
		`RP/0/RSP0/CPU0:routerconfig-vrrp-if)# address-family ipv6`
		`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)#`
			IPv4 または IPv6 アドレス ファミリ 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 5	**vrrp** *vrid* **version** { *2* | *3* }
	
	例：
	
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)# vrrp 3 version 3`
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)#`
		仮想ルータ 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 6	**accept-mode disable** 例： `RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)# accept-mode disable`	VRRP仮想アドレスのルートのインストールをディセーブルにします。

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

グローバル仮想 IPv6 アドレスの設定

次のタスクを実行して、仮想ルータのグローバル仮想 IPv6 アドレスを設定します。

手順の概要

1. **configure**
2. **router vrrp**
3. **interface** *type interface-path-id*
4. **address-family ipv4**
5. **vrrp** *vrid* **version 3**
6. **address global** *address*
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router vrrp** 例： `RP/0/RSP0/CPU0:router(config)# router vrrp`	VRRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-vrrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスで VRRP インターフェイス コンフィギュレーション モード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	**address-family ipv4** 例： `RP/0/RSP0/CPU0:routerconfig-vrrp-if)# address-family ipv4`	IPv4 または IPv6 アドレス ファミリ サブモードを開始します。
ステップ 5	**vrrp** *vrid* **version 3** 例： `RP/0/RSP0/CPU0:router(config-vrrp-address-family)# vrrp 3 version 3`	仮想ルータ 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 6	**address global** *address* 例： `RP/0/RSP0/CPU0:routerconfig-vrrp-virtual-router)# address global 2001:db8::/32`	仮想ルータのグローバル仮想 IPv6 アドレスを設定します。 （注） VRRP のパケット サイズは、対応するインターフェイスの最大伝送ユニット（MTU）により制限されます。 これにより、単一の VRRP セッションでサポートできる、グローバル仮想 IPv6 アドレスの最大数が制限されます。 たとえば、ギガビット イーサネット インターフェイス上のデフォルト MTU では、単一セッションで最大 90 の VRRP グローバル仮想 IPv6 アドレスを許可します。 このようなアドレスをより多く使用するには、インターフェイスの MTU をそれに応じて増やす必要があります。

ステップ 7	次のいずれかのコマンドを使用します。
	• **end**
	• **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

	コマンドまたはアクション	目的
		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。

プライマリ仮想 IPv4 アドレスの設定

次のタスクを実行して、仮想ルータのプライマリ仮想 IPv4 アドレスを設定します。

手順の概要

1. **configure**
2. **router vrrp**
3. **interface** *type interface-path-id*
4. **address-family ipv4**
5. **vrrp** *vrid* **version** { *2* | *3* }
6. **address** *address*
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **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。

	コマンドまたはアクション	目的
ステップ 2	**router vrrp** 例： `RP/0/RSP0/CPU0:router(config)# router vrrp`	VRRP コンフィギュレーション モードをイネーブルにします。

ステップ 3	**interface** *type interface-path-id*
	例：
	`RP/0/RSP0/CPU0:router(config-vrrp)# interface TenGigE 0/2/0/1`
	`RP/0/RSP0/CPU0:router`
		特定のインターフェイスで VRRP インターフェイス コンフィギュレーション モードをイネーブルにします。

ステップ 4	**address-family ipv4**
	例：
	`RP/0/RSP0/CPU0:routerconfig-vrrp-if)# address-family ipv4`
	`RP/0/RSP0/CPU0:router(config-vrrp-address-family)#`
		IPv4 アドレス ファミリ サブモードを開始します。

ステップ 5	**vrrp** *vrid* **version** { *2* | *3* }
	例：
	`RP/0/RSP0/CPU0:router(config-vrrp-address-family)# vrrp 3 version 2`
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)`
		仮想ルータ 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 6	**address** *address* 例： `RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)# address 10.20.30.1`	仮想ルータのプライマリ仮想 IPv4 アドレスを設定します。
ステップ 7	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	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、

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	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。

セカンダリ仮想 IPv4 アドレスの設定

次のタスクを実行して、仮想ルータのセカンダリ仮想 IPv4 アドレスを設定します。

手順の概要

1. **configure**
2. **router vrrp**
3. **interface** *type interface-path-id*
4. **address-family ipv4**
5. **vrrp** *vrid* **version** { *2* | *3* }
6. **address** *address* **secondary**
7. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router vrrp** 例： `RP/0/RSP0/CPU0:router(config)# router vrrp`	VRRP コンフィギュレーション モードをイネーブルにします。

ステップ 3	**interface** *type interface-path-id*
	例：
	`RP/0/RSP0/CPU0:router(config-vrrp)# interface TenGigE 0/2/0/1`
	`RP/0/RSP0/CPU0:router`
		特定のインターフェイスで VRRP インターフェイス コンフィギュレーション モードをイネーブルにします。

ステップ 4	**address-family ipv4**
	例：
	`RP/0/RSP0/CPU0:routerconfig-vrrp-if)#`
	`address-family ipv4`
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)#`
		IPv4 アドレス ファミリ サブモードを開始します。

ステップ 5	**vrrp** *vrid* **version** { *2* | *3* }
	例：
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)#`
	`vrrp 3 version 3`
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)#`
		仮想ルータ 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 6	**address** *address* **secondary**
	例：
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)#`
	`address 10.20.30.1 secondary`
		仮想ルータのセカンダリ仮想 IPv4 アドレスを設定します。

ステップ 7	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **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。

仮想リンクローカル IPv6 アドレスの設定

次のタスクを実行して、仮想ルータの仮想リンクローカル IPv6 アドレスを設定するか、または仮想リンクローカル IPv6 アドレスがイネーブルで、仮想ルータの仮想メディア アクセス コントロール（MAC）アドレスから自動的に計算される必要があることを指定します。

IPv6 アドレス空間は、IPv4 に比べて異なる構造になっています。 リンクローカル アドレスは、ローカルネットワーク上の各インターフェイスを識別するために使用します。 これらのアドレスは、インターフェイスのリンクローカル（ハードウェア）アドレス（イーサネット インターフェイスの MAC アドレス）を使用して、標準の方法で設定または決定されます。 リンクローカル アドレスは、標準の形式を持ち、ローカル ネットワークでのみ有効です（複数ホップ先とのルーティングは実行できません）。

グローバル ユニキャスト IPv6 アドレスは、IPv6 アドレス空間で、リンクローカル アドレスから分離したサブセットを占有します。 これらは、複数ホップ先と相互にルーティングでき、関連付けられたプレフィックス長（0 ～ 128 ビット）を持ちます。

各 VRRP 仮想ルータには、関連付けられた仮想リンクローカル アドレスがあります。 これは、仮想ルータの仮想 MAC アドレスから自動的に設定および決定されます。 仮想 MAC アドレスは、ローカル ネットワークで一意である必要があります。 仮想リンクローカル アドレスは、スコープがローカルのアドレスでは重複アドレス検出が不要であるため、その仮想IP（VIP）状態がアップであることが常に考慮される点を除き、IPv4 仮想ルータのプライマリ仮想 IPv4 アドレスに似ています。

手順の概要

1. **configure**
2. **router vrrp**
3. **interface** *type interface-path-id*
4. **address-family ipv4**
5. **vrrp** *vrid* **version 3 address linklocal** {*address* | **autoconfigure**}
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router vrrp** 例： `RP/0/RSP0/CPU0:router(config)# router vrrp`	VRRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-vrrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスで VRRP インターフェイス コンフィギュレーション モード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	**address-family ipv4** 例： `RP/0/RSP0/CPU0:routerconfig-vrrp-if)# address-family ipv4`	IPv6 アドレス ファミリ サブモードを開始します。

ステップ 5	**vrrp** *vrid* **version 3 address linklocal** {*address* | **autoconfigure**}
	
	例：
	
	`RP/0/RSP0/CPU0:routerconfig-vrrp-address-family)# vrrp 1 version 3 address linklocal FE80::260:3EFF:FE11:6770`
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)#`
	
	`RP/0/RSP0/CPU0:router(config-vrrp-address-family)# vrrp 1 version 3 address linklocal autoconfigure`
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)#`
		• 仮想ルータの仮想リンクローカル IPv6 アドレスを設定します。
		• 仮想リンクローカル IPv6 アドレスが、イネーブルで、仮想ルータの仮想 MAC アドレスから自動的に計算されるように指定します。
		
		（注） VRRP ルータの仮想リンクローカルアドレスが、インターフェイスのリンクローカルアドレスと同じである場合は、インターフェイス上で IPv6 重複アドレス検出（DAD）をディセーブルにする必要があります。 DAD がディセーブルになると、重複パケットには重複のフラグが付けられなくなります。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		• 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。

状態変更ロギングのディセーブル化

次のタスクを実行して、syslogを介してVRRP状態変更イベントをロギングするタスクをディセーブルにします。

手順の概要

1. **configure**
2. **router vrrp**
3. **message state disable**
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router vrrp** 例： `RP/0/RSP0/CPU0:router(config)# router vrrp`	VRRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**message state disable** 例：	syslog を介して VRRP 状態変更イベントをロギングするタスク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
	RP/0/RSP0/CPU0:router(config-vrrp)# message state disable RP/0/RSP0/CPU0:router(config-vrrp)#	
ステップ 4	次のいずれかのコマンドを使用します。 • **end** • **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。

VRRP 用 BFD

双方向フォワーディング検出（BFD）は、2 つのフォワーディング エンジン間の障害の検出に使用されるネットワーク プロトコルです。 BFD セッションは、非同期モードまたはデマンド モードという 2 つのモードのいずれかで動作できます。 非同期モードでは、両方のエンドポイントが互いに hello パケットを定期的に送信します。 これらのパケットを複数回受信しない場合は、セッションがダウンしていると見なされます。 デマンド モードでは、hello パケットの交換は必須ではなく、必要に応じてそれぞれのホストが hello メッセージを送信できます。 シスコでは、BFD 非同期モードをサポートしています。

BFD の利点

- BFD は、1 秒未満で障害を検出します。
- BFD では、すべてのタイプのカプセル化をサポートしています。
- BFDは、特定のルーティングプロトコルに限定されることなく、ほとんどすべてのルーティング プロトコルをサポートします。

BFD プロセス

VRRP では BFD を使用して、リンク障害を検出し、過剰な制御パケットオーバーヘッドなしでフェールオーバーを高速化します。

VRRP プロセスでは、必要に応じて BFD セッションを作成します。 BFD セッションがダウンすると、セッションをモニタリングしている各バックアップ グループがマスター状態に移行します。

VRRPは、BFDセッションのダウンによりトリガーされたマスター状態への移行後10秒間は状態選択に参加しません。

BFD の設定

VRRP の場合、設定は、既存の VRRP インターフェイス サブモードで、VRRP 仮想ルータごとに設定できるBFD高速障害検出およびインターフェイスごとに設定できるタイマー（最小インターフェイスと乗数）を使用して適用されます。 BFD高速障害検出は、デフォルトでディセーブルになっています。

双方向フォワーディング検出のイネーブル化

手順の概要

1. **configure**
2. **router vrrp**
3. **interface** *type interface-path-id*
4. **address-family** {**ipv4** | **ipv6**}
5. **vrrp** *vrid* **version** { *2* | *3* }**bfd fast-detect peer ipv4** *address*
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router vrrp** 例： `RP/0/RSP0/CPU0:router(config)# router vrrp`	VRRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-vrrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスで VRRP インターフェイス コンフィギュレーション モードをイネーブルにします。
ステップ 4	**address-family** {**ipv4** | **ipv6**} 例： `RP/0/RSP0/CPU0:router(config-vrrp-if)# address-family ipv4`	（任意）特定のインターフェイスでアドレス ファミリ コンフィギュレーション モードをイネーブルにします。
ステップ 5	**vrrp** *vrid* **version** { *2* | *3* }**bfd fast-detect peer ipv4** *address* 例： `RP/0/RSP0/CPU0:router(config-vrrp-if)# vrrp 100 version 3 bfd fast-detect peer ipv4 2001:db8::/32`	BFD 高速検出を VRRP インターフェイスでイネーブルにします。 （注） BFD は、2 台のルータを使用する冗長システムにのみ適しています。
ステップ 6	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	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

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	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。

BFD タイマー（最小間隔）の変更

最小間隔により、BFD ピアへの BFD パケットの送信頻度（ミリ秒単位）が決まります。 デフォルトの最小間隔は 15 ミリ秒です。

手順の概要

1. **configure**
2. **router vrrp**
3. **interface** *type interface-path-id*
4. **vrrp** *vrid* **version** { *2* | *3* }
5. **bfd minimum-interval** *interval*
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router vrrp** 例： `RP/0/RSP0/CPU0:router(config)# router vrrp`	VRRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-vrrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスで VRRP インターフェイス コンフィギュレーション モードをイネーブルにします。

ステップ 4	**vrrp** *vrid* **version** { *2* | *3* }
	例：
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)# vrrp 3 version 3`
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)#`
		仮想ルータ 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 5	**bfd minimum-interval** *interval* 例： `RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)# bfd minimum-interval`	最小間隔を指定の間隔に設定します。 間隔はミリ秒で、範囲は 15 ～ 30000 ミリ秒です。
ステップ 6	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	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

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	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。

BFD タイマー（乗数）の変更

乗数は、ピアが利用不可であると宣言するまでに許容される、BFD ピアから連続して紛失される BFD パケットの数です。 デフォルトの乗数は 3 です。

手順の概要

1. **configure**
2. **router vrrp**
3. **interface** *type interface-path-id*
4. **vrrp** *vrid* **version** { *2* | *3* }
5. **bfd multiplier** *multiplier*
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**router vrrp** 例： `RP/0/RSP0/CPU0:router(config)# router vrrp`	VRRP コンフィギュレーション モードをイネーブルにします。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-vrrp)# interface TenGigE 0/2/0/1`	特定のインターフェイスで VRRP インターフェイス コンフィギュレーション モードをイネーブルにします。

ステップ 4	**vrrp** *vrid* **version** { *2* | *3* }
	例：
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)# vrrp 3 version 3`
	`RP/0/RSP0/CPU0:router(config-vrrp-virtual-router)#`
		仮想ルータ 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 5	**bfd multiplier** *multiplier* 例： `RP/0/RSP0/CPU0:router(config-vrrp-if)# bfd multiplier`	値に乗数を設定します。 範囲は 2 ～ 50 です。
ステップ 6	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	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

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	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。

MIB の VRRP サポート

VRRP を使用すると、障害が発生したとき、ルータが 1 つ以上の IP アドレスを引き継ぐことができます。たとえば、障害の発生したルータがデフォルト ゲートウェイであったために、ホストからの IP トラフィックがそのルータに到達した場合、そのトラフィックは制御を引き継いだ VRRP ルータによって透過的に転送されます。 VRRP を使用する場合、ダイナミック ルーティングやルータ ディスカバリ プロトコルの設定を各エンド ホストで行う必要はありません。 仮想ルータに割り当てる IP アドレスを制御する VRRP ルータはマスターと呼ばれ、送信されたパケットをそれらの IP アドレスに転送します。 この選択プロセスにより、マスターが使用不可になった場合の転送責任のダイナミック フェールオーバー（スタンバイ）が提供されます。 これにより、エンド ホストでは、LAN 上のすべての仮想ルータ IP アドレスを最初のデフォルト ホップ ルータとして使用できるようになります。VRRP を使用する利点として、デフォルト パスのアベイラビリティが向上し、各エンド ホストでダイナミック ルーティングやルータ ディスカバリ プロトコルを設定する必要がないことを挙げることができます。 SNMP トラップは、仮想ルータ（スタンバイ）がマスター状態に移行した場合、またはスタンバイ ルータがマスターになった場合に、状態変更に関する情報を提供します。

VRRP イベントに関する SNMP サーバ通知の設定

snmp-server traps vrrp events コマンドは、VRRP に関する簡易ネットワーク管理プロトコル（SNMP）サーバ通知をイネーブルにします。

手順の概要

1. **configure**
2. snmp-server traps vrrp events
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	snmp-server traps vrrp events 例： `RP/0/RSP0/CPU0:router(config)snmp-server traps vrrp events`	VRRP に関する SNMP サーバ通知をイネーブルにします。

ステップ 3	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		◦ **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。

VRRP のホット リスタート

1 つのグループで VRRP プロセスの障害が発生した場合には、ピア VRRP マスター ルータ グループで強制的にフェールオーバーが行われないようにする必要があります。 ホット リスタートはウォーム RP フェールオーバーをサポートしており、ピア VRRP ルータへの強制的なフェールオーバーは発生しません。

VRRP 実装の設定例： Cisco IOS XR ソフトウェア

ここでは、次の VRRP 設定例について説明します。

VRRP グループの設定：例

ここでは、それぞれが 3 つの VRRP グループに含まれている、ルータ A およびルータ B の設定例を示します。

ルータ A：

```
config
interface tenGigE 0/4/0/4
ipv4 address 10.1.0.1/24
exit
router vrrp
interface tenGigE 0/4/0/4
address-family ipv4
vrrp 1 version 2
priority 120
text-authentication cisco
timer 3
address 10.1.0.10
vrrp 5 version 2
timer 30
address 10.1.0.50
vrrp 100 version 2
```

```
preempt disable
address 10.1.0.100
commit
```

ルータ B：

```
config
interface tenGigE 0/4/0/4
ipv4 address 10.1.0.2/24
exit
router vrrp
interface tenGigE 0/4/0/4
address-family ipv4
vrrp 1 version 2
priority 100
text-authentication cisco
timer 3
address 10.1.0.10
vrrp 5 version 2
priority 200
timer 30
address 10.1.0.50
vrrp 100 version 2
preempt disable
address 10.1.0.100
commit
```

設定例では、各グループのプロパティは次のとおりです。

- グループ 1：
  - 仮想 IP アドレスは 10.1.0.10 です。
  - ルータ A はプライオリティ 120 で、このグループのマスターになります。
  - アドバタイズ インターバルは 3 秒です。
  - アドバタイズ インターバルは 3 秒です。
  - プリエンプションはイネーブルです。
- グループ 5：
  - ルータ B はプライオリティが 200 で、このグループのマスターになります。
  - アドバタイズ インターバルは 30 秒です。
  - プリエンプションはイネーブルです。
- グループ 100：
  - プリエンプションがディセーブルであるため、最初に設定したルータが、最初にグループのマスターになります。
  - アドバタイズ インターバルはデフォルトの 1 秒です。
  - プリエンプションはディセーブルです。
  - プリエンプションはディセーブルです。

VRRP 統計情報のクリア：例

clear vrrp statistics コマンドは、独自の出力は生成しません。 このコマンドは、**show vrrp statistics** コマンドにより提供された統計情報を変更するため、すべての統計情報がゼロにリセットされます。

次の項では、**show vrrp statistics** コマンドの出力例に続いて **clear vrrp statistics** コマンドの出力例を示します。

```
RP/0/RSP0/CPU0:router# show vrrp statistics
show vrrp statistics
Invalid packets:
 Invalid checksum:              0
 Unknown/unsupported versions:  0
 Invalid vrID:                  10
 Too short:                     0
Protocol:
 Transitions to Master          6
Packets:
 Total received:                155
 Bad TTL:                       0
 Failed authentication:         0
 Unknown authentication:        0
 Conflicting authentication:    0
 Unknown Type field:            0
 Conflicting Advertise time:    0
 Conflicting Addresses:         0
 Received with zero priority:   3
 Sent with zero priority:       3


RP/0/RSP0/CPU0:router# clear vrrp statistics
RP/0/RSP0/CPU0:router# show vrrp statistics
Invalid packets:
 Invalid checksum:              0
 Unknown/unsupported versions:  0
 Invalid vrID:                  0
 Too short:                     0
Protocol:
 Transitions to Master          0
Packets:
 Total received:                0
 Bad TTL:                       0
 Failed authentication:         0
 Unknown authentication:        0
 Conflicting authentication:    0
 Unknown Type field:            0
 Conflicting Advertise time:    0
 Conflicting Addresses:         0
 Received with zero priority:   0
 Sent with zero priority:       0
```

その他の参考資料

ここでは、VRRP の関連資料について説明します。

関連資料

関連項目	参照先
QoS コマンド：コマンド構文の詳細、コマンド モード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『*Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Command Reference*』の「*Quality of Service Commands*」
クラスベースのトラフィック シェーピング、トラフィック ポリシング、低遅延キューイング、および Modified Deficit Round Robin（MDRR）	『*Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Configuration Guide*』の「*Configuring Modular Quality of Service Congestion Management*」
WRED、RED、およびテール ドロップ	『*Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Configuration Guide*』の「*Configuring Modular QoS Congestion Avoidance*」
VRRP コマンド	『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*VRRP Commands*」
マスター コマンド リファレンス	『*Cisco ASR 9000 Series Aggregation Services Router Commands Master List*』
スタートアップ資料	『*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 ASR 9000 Series Aggregation Services Router System Security Configuration Guide*』の「*Configuring AAA Services*」

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
—	MIB を特定してダウンロードするには、次の URL にある Cisco MIB Locator を使用し、[Cisco Access Products] メニューからプラットフォームを選択します。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

第 11 章

ビデオ モニタリングの実装

ビデオ モニタリングの設定は、関連するクラスマップとポリシーマップの設定、インターフェイスへのビデオ モニタリング ポリシーのバインドなどを含む 4 ステップの手順です。

ビデオ モニタリングの実装の前提条件

- 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。
- 高度なビデオサービスのためのパッケージをインストールしてアクティブ化する必要があります。 オプション パッケージのインストールの詳細については、『*Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*』を参照してください。
- マルチキャスト ルーティング ソフトウェアのパッケージをインストールしてアクティブ化し、システムでマルチキャスト ルーティングをイネーブルにする必要があります。 ビデオ モニタリングは、マルチキャストがイネーブルになっているインターフェイスでサポートされます。 マルチキャスト ルーティングの詳細については、「Implementing Layer 3 Multicast Routing on Cisco ASR 9000 Series Routers」の章を参照してください。

ビデオ モニタリングの実装に関する情報

ビデオ モニタリングの概要

ビデオ環境の低下は、サービス プロバイダーにとってサービス コストや収益の損失の面で大きな懸念要因となります。 ヘルプ デスク時間、NOC（ネットワーク オペレーション センター）トラブルシューティング リソース、およびトラック ロールのサービス コストを回避するには、ビデオ トラフィックをモニタする機能が不可欠です。 Cisco ASR9000 ルータでは、ビデオ モニタリングによってビデオ フローの問題を簡単に診断できます。

パケット損失は、ビデオ品質低下の一般的な原因の 1 つです。 その影響は、圧縮されたビデオ フローでさらに大きくなります。 サービス プロバイダーの IP ネットワークで送信されるビデオ トラフィックは、ほとんどが圧縮されたビデオ（MPEG または同様の符号化）です。 圧縮の方法により、トラフィックは非常に損失の影響を受けやすくなります。 ビデオは数秒ごとに独立したフレーム（I-frame）で符号化され、後続のフレームは I-frame からのデルタになります。 I-frame で損失が発生すると、3 ミリ秒のトラフィック（約 1 個の IP パケット）損失により、最大 1.2 秒間表示が低下する可能性があります。

ジッターは重要なフロー特性で、エンドデバイスでバッファ プロビジョニングを慎重に行う必要があります。画面にメディアを表示するセット トップ ボックス（STB）でビデオをリアルタイムにデコードする必要があります。 着信ビデオ ストリームをバッファに格納して、イメージをスムーズにデコードおよび表示できるようにします。 ネットワーク ジッターが大きい場合は、STB でバッファ アンダーランやバッファ オーバーランが発生する可能性があります。 ジッターの大きさに応じて、ディスプレイで視覚的なアーティファクトやブラック スクリーンが発生します。

ブロードキャスト専用のアプリケーションでは、転送におけるエンドツーエンドの遅延は重要ではありません。ただし、ビデオアプリケーションはよりインタラクティブになっているため、エンドツーエンドの遅延が重要な Quality of Experience（QoE）コンポーネントになります。 データ損失は QoE 低下の主な原因です。

QoE 低下の主な原因は次の 3 つにまとめられます。

- パケット損失
- ジッター
- 遅延

ビデオモニタリングは、ビデオ品質の向上およびQoEの拡張において大きな役割を果たします。ビデオモニタリングはルータに実装され、ネットワーク オペレータはフローごとにビデオ転送パフォーマンスを測定および追跡できます。 ビデオパケットはルータを通過します。パケットヘッダーを使用して、ビデオ品質に影響を及ぼすネットワークのパフォーマンスの尺度を示すメトリックを計算できます。 同じフローについて複数のルータから取得したこの情報を比較して、ネットワークにおけるビデオ問題および影響を受けるフローをエンドツーエンドで明確に把握できます。

ビデオ モニタリングによってビデオ フロー（一般的にはストリーミング フロー）の問題を診断できます。 ビデオモニタリングの目的は、QoE低下の原因となるネットワークによる混乱や異常

を検出することです。つまり、ストリーミング（ビデオ）トラフィックの転送パフォーマンスを測定します。符号化エラー、オーディオとビデオ間のラグ、およびその他のエラーもQoE低下の原因となります。ただし、これらはネットワークではなく符号化デバイスで発生します。これらの後者のエラーはモニタされません。

ビデオ モニタリングでサポートされる主要機能

データ プレーンからの直接測定

ビデオモニタリングは、ビデオ品質の向上およびQoEの拡張において大きな役割を果たします。Cisco ASR 9000 シリーズ ルータに実装されたビデオ モニタリングを使用することで、ネットワーク オペレータはリアルタイムにフローごとのビデオ転送パフォーマンスを測定および追跡できます。従来のトラフィック モニタリング ソリューション（サンプリング対象のフローをコントロール プレーン、またはルータ上の専用ブレードなどの他のハードウェアに送信する必要があります）とは対照的に、Cisco ASR 9000 シリーズ ルータのビデオ モニタリングでは、データ プレーン自体でモニタリング操作を実行します。これにより、転送されたパケットをリアルタイムに分析し、ビデオ品質に影響を及ぼすネットワークのパフォーマンスの尺度を示すメトリックを計算できます。

ローカル ストレージおよびリモート アクセス

ビデオ モニタリングでは、有線と同じ速度でパケット損失およびジッターを測定し、収集した情報をルータに保存して、ネットワーク オペレータがユーザインターフェイスを介してその情報にアクセスできるようにします。さらに、測定されて複数のルータに保存されたパフォーマンス メトリックにリモート オペレーション センターから標準の SNMP を介してアクセスできます。これらのメトリックにより、構成および分析できるビデオ フローをエンドツーエンドで明確に把握できます。

プロアクティブおよびリアクティブな用途

Cisco ASR 9000 シリーズ ルータのビデオ モニタリングには、サービス プロバイダーのためのリアクティブな用途とプロアクティブな用途があります。ビデオ モニタリングは、サービス カバレッジを新しいカスタマーに拡大する前に、ビデオ サービスの品質を確認する目的で使用できます。また、強力な分析ツールであり、カスタマー コールのトラブルシューティングに使用できます。ネットワーク オペレータは、パケット損失、ジッター、フロー レート、フロー数などの変動など、各種イベントに対してアラームを発生させるようにビデオ モニタリングを設定できます。このようなアラームは、有効な値または範囲でトリガーされるように設定できます。

ビデオ モニタリング上のフロー

ビデオモニタリングでは、4つのパケット ヘッダー フィールドを使用して一意のフローを識別します。それらのフィールドは、送信元 IP アドレス、宛先 IP アドレス、送信元 UDP ポート、および宛先 UDP ポート（これはプロトコル ID が常に UDP であることを示します）です。

ユニキャストおよびマルチキャスト

ビデオ モニタリングでは、IP ヘッダーに IPv4 マルチキャスト宛先アドレスを含むフローのモニタリングだけでなく、ユニキャスト宛先アドレスを含むフローのモニタリングもサポートされます。 ユニキャスト フローのビデオ モニタリング機能のサポートは、Trident LC との下位互換性を提供し、Typhoon LC でも使用できます。

フロー レート タイプおよびプロトコル レイヤ

ビデオ モニタリングでは、IP レイヤで CBR（固定ビット レート）フローをモニタします。 つまり、IPv4 パケット内の UDP データグラムにカプセル化され CBR で符号化されたメディア ストリーム（たとえば MPEG-2）をモニタできます。 ビデオ モニタリングを使用すると、（メディア パケットの数およびサイズとともに）IP レイヤのパケット レートまたはメディア レイヤのビット レートを設定できます。

メトリック

ビデオ モニタリングでは、IP-UDP レベルの MDI（Media Delivery Index、RFC 4445）定義に従ったパケット損失とジッターの両方のメトリックがサポートされます。 MDI メトリックは、MLR（メディア損失レート）と DF（遅延係数）です。 ビデオ モニタリングでは、MDI MLR の拡張である MRV（メディア レート変動）を使用します。つまり、MLR は損失のみをキャプチャし、MRV は損失と超過の両方をキャプチャします。 ビデオ モニタリングの DF は MDI 定義と同じです。DF はモニタ対象の MDI ジッターに加えて 1 つの公称パケット到着間隔時間を表します。 ビデオ モニタリングでは、2 つの主要メトリックとともに、パケット数、バイト数、パケット レート、ビット レート、パケット サイズ、IP ヘッダー内の TTL（存続可能時間）フィールド、フロー数、発生したアラーム、および各種イベントのタイム スタンプがサポートされます。

（注） MDI ジッターという用語は、ビデオ モニタリングで測定された DF メトリックの正当性を示すために使用されます。 MDI ジッターは、実際のパケット到着時間を公称到着参照と比較することによって測定され、簡単なパケット間ジッターは、2 つの連続するパケット到着時間の差で測定されます。 前者は後者よりも正確に CBR フローのパフォーマンスをキャプチャします。

フロー数

現在のリリースでは、Cisco ASR 9000 シリーズ ルータのビデオ モニタリングは、ユニキャスト トラフィックとマルチキャスト トラフィックの組み合わせについて、Trident LC では NP（ネットワーク プロセッサ）あたり 1024 フローをサポートし、Typhoon LC では NP あたり最大 4096 フローをサポートします。 各ライン カードまたは各システムの最大フローの数は、ライン カード上の NP の数およびシステム上のライン カードの数によって異なります。 シャーシごとのフロースケールは、シャーシ上の NP の数によって異なります。

たとえば、4 個の Trident LC を搭載した Cisco ASR 9000 シリーズ ルータ ボックスがあり、各 LC に 8 個の NP が搭載されている場合、シャーシごとのフロー スケールは最大 1K*8 = 8K フローになります。

ハイ アベイラビリティ機能

Cisco ASR 9000 シリーズ ルータのビデオ モニタリングでは、各レベルでハイ アベイラビリティがサポートされます。 プロセスの OIR（活性抜粋）、ライン カードの OIR、RSP（ルート スイッチプロセッサ）のフェールオーバーおよびルータのリロードがサポートされます。 設定はすべてのハイ アベイラビリティ シナリオで永続的です。 モニタされた統計データは、プロセスの OIR および RSP FO 時に保持されます。

インターフェイスのタイプおよび方向

ビデオ モニタリングをアクティブ化するには、インターフェイスに対してビデオ モニタリング サービス ポリシーを設定する必要があります。 ビデオ モニタリング ポリシーを関連付けることができるインターフェイスには 4 つのタイプがあります。これらは、メイン インターフェイス、サブインターフェイス、イーサネット バンドル インターフェイス、およびイーサネット バンドル サブインターフェイスです。 ビデオモニタリングでは、レイヤ 3 インターフェイスのみサポートされ、レイヤ 2 インターフェイスはサポートされません。 ビデオ モニタリングは、インターフェイスの入力方向にのみ設定できます。

フロー レートと DF の精度

Cisco ASR 9000 シリーズ ルータのビデオ モニタリングでは、1 ミリ秒の精度の DF メトリック パフォーマンスが提供されます。 さらに、最大 100 Mbps フロー レートの標準画質（SD）ビデオ トラフィック（ほとんどが圧縮されます）がサポートされます。圧縮されていないビデオストリームの場合、最大 3 Gbps のフロー レートがサポートされます。

入力のユーザ インターフェイス

ビデオモニタリングでは、設定について MQC（モジュラ QoS 設定）構文に従った従来の CLI（コマンドライン インターフェイス）入力がサポートされます。 アクセス コントロール リスト（ACL）、クラスマップ、およびポリシーマップを設定してビデオ モニタリングを設定できます。ビデオモニタリングは、サービスポリシーをインターフェイスに関連付けることによってアクティブ化できます。 インプレイス ポリシー変更はサポートされません。 設定済みのサービス ポリシーをインターフェイスに関連付けた後で変更するには、インターフェイスとの関連付けを解除する必要があります。

出力のユーザ インターフェイス

ビデオ モニタリングでは、モニタされた統計を取得するための各種 show コマンドと clear コマンドが提供されます。 ビデオ モニタリング コマンドの詳細については、『Cisco ASR 9000 Series Aggregation Services Router Multicast Command Reference』の「Video Monitoring Commands on Cisco ASR 9000 Series Routers」を参照してください。

TCA（しきい値超過アラート）をポリシーマップの一部として設定して、ビデオ モニタリングによるさまざまな状況に関する syslog メッセージの生成を可能にすることができます。 **show** コマンドまたは SNMP pull を使用してスタンディング アラームを取得することもできます。 XML はビデオ モニタリングではサポートされません。

クラスマップとポリシーマップの数

ビデオモニタリングを使用するには、データプレーンでモニタするフローを決定するフィルタとして機能するクラスマップとポリシーマップを設定する必要があります。 ビデオモニタリングでは、ポリシーマップあたり最大 1024 のクラスマップとシステムあたり最大 1024 のクラスマップがサポートされます。 システム全体で最大 256 のポリシーマップがサポートされます。

ビデオ PIE のインストール

ビデオ モニタリングを使用するには、ビデオ PIE をインストールする必要があります。 RSP タイプに応じて、ビデオ PIE の名前に 2 つのバージョンがあります。

- asr9k-video-p.pie（RSP2 バージョン）
- asr9k-video-px.pie（RSP3 バージョン）

ビデオ モニタリングのトラップおよびクローン

トラップおよびクローンは、基本的なパフォーマンス モニタリング サービス機能の拡張です。選択した数のフローからのパケットをフィルタ（トラップ）、複製（クローン）、およびネットワーク上のリモートデバイスに送信して、ビデオ品質をより詳細に分析できます。 クローンされたパケットは、マルチキャスト転送プロセスによって、パフォーマンス トラフィック クローン プロファイルで指定されたインターフェイスにレプリケートされます。 リモート デバイスは、MPEG レイヤ レベルでデータを詳細に分析できます。 このデバイスは、デバッグ ツールとしてもモニタリング ツールとしても使用できます。 同じルータ上のサービス エンジン ブレードとしても使用できます。 マルチキャスト フローの場合、トラップおよびクローン機能は完全に下位互換性があります。ただし、ユニキャスト フローの場合、この機能は Typhoon LC 上のレイヤ 3 スイッチド ポート アナライザ（SPAN）でのみサポートされます。

（注） L3 SPAN では SNMP はサポートされません。 L3 SPAN の詳細については、「Configuring SPAN」を参照してください。

ビデオ モニタリングの用語

Cisco ASR 9000 シリーズ ルータにビデオ モニタリング サービスを実装して設定するには、まずビデオ モニタリングの用語と概念を理解する必要があります。

インターバル間隔およびインターバル アップデート

ビデオ モニタリングでは、ユーザによって設定されたインターバル期間と呼ばれる時間、データプレーン上のすべてのパケットを継続的に分析します。 統計情報は、各インターバル期間の最後に定期的にエクスポートされます。 これらのエクスポートされた統計情報はインターバル アップデートと呼ばれます。 ビデオモニタリングのフローおよびその遷移のステータスは、これらのインターバル アップデートに関してのみ説明されます。 また、これらのインターバル アップデートに関して、エクスポートされたすべてのビデオモニタリング フロー統計情報が格納されます。

インターバル期間は、重要なビデオ モニタリング パラメータです。 ビデオ モニタリング設定では、エクスポート頻度、保存するエクスポート数、非アクティブなフローを削除する時間などの機能についてインターバル期間を決定します。（フローの停止およびパフォーマンスの低下を伴うフローに対する）アラームの発生などのすべてのビデオ モニタリング機能は、インターバル アップデートの内容に基づきます。

ビデオ モニタリング フロー

ビデオ モニタリング フローは、ヘッダー フィールドが設定済みのクラスマップ（およびそれに関連付けられたアクセス コントロール リスト）に一致するパケット ストリームのインスタンスです。 一意のフローは、ビデオ モニタリング サービス ポリシーが関連付けられているインターフェイスに対してローカルです。ビデオモニタリングフローは一連の保存済みインターバルアップデートで構成されます。 モニタリング インターバル後にビデオ モニタリングで作成された一意のフローは新規フローと呼ばれます。 そのため、存続期間が 1 回のモニタリング インターバルよりも短いパケット ストリームは、ビデオ モニタリング フローとしてエクスポートされず、保存されません。

フローの停止

ルータが 1 回のインターバル アップデート以上の期間、モニタ対象フローでのパケットの受信を停止した場合、そのモニタ対象フローは停止していると見なされます。

フローの再開

停止されたビデオ モニタリング フローでパケットの受信が再開されると、通常のインターバル アップデートが次のモニタリング インターバルでエクスポートされます。 再開されたフローには、1 回以上のゼロ インターバルがあり、その後に通常のインターバル アップデートが続きます。

フローのスイッチオーバー

イーサネット バンドル インターフェイスまたはイーサネット バンドル サブインターフェイス上のビデオ モニタリング フローは、ある物理メンバ インターフェイスから別のインターフェイスに移動する場合があります。つまり、パケット ストリームがあるインターフェイスでフローを停止し、別のインターフェイスでフローを再開します。 これはフローのスイッチオーバーと定義されています。 この場合、両方のインターフェイスが同じライン カード上にあれば、ビデオ モニタリングはスイッチオーバー前のフローとスイッチオーバー後のフローを同じフローとして処理します。 それ以外の場合、2 つの異なるフローとして処理します。

フローの削除

停止されたビデオ モニタリング フローが（モニタリング インターバルの数に関して）設定されたタイムアウトの間ゼロ インターバルをエクスポートし続ける場合、フローはデッドと見なされ、削除対象としてマークされます。ユーザが非アクティブ フローを制御できる期間は、タイムアウトパラメータを使用して指定されます。マークされたすべてのフローの実際の削除は、Trident LC では 150 秒ごと、Typhoon LC では 60 秒ごとに実行される定期的なスイープ機能により、少し遅れて実行されます。 フローが削除されると、すべてのエクスポート済み統計情報（ゼロ インターバルを含む一連のインターバル アップデート）は完全にストレージから削除されます。

ビデオ モニタリングの実装

ビデオ モニタリングの設定は、関連するクラスマップとポリシーマップの設定、インターフェイスへのビデオ モニタリング ポリシーのバインドなどを含む 4 ステップの手順です。

IPv4 アクセス リストの作成

この手順は、一般的な IPv4 アクセス リストの作成および設定の手順に似ています。 ここでは、クイック リファレンスとしてビデオ モニタリングの ACL の設定例を示します。 詳細については、『*Cisco ASR 9000* シリーズ アグリゲーション サービス ルータ *IP* アドレスおよびサービス コンフィギュレーションガイド』の「アクセスリストおよびプレフィックスリストの実装」の章を参照してください。

このタスクでは、標準 IPv4 アクセス リストを設定します。

標準アクセス リストでは、照合操作に送信元アドレスを使用します。

（注） ビデオ モニタリング ポリシーでは、ACL 設定で明示的な **deny** ステートメントを使用できません。 また、log または log-input は ACL 設定ではサポートされません。

手順の概要

1. **configure**
2. **ipv4 access-list** *name*
3. [*sequence-number*] **remark** *remark*
4. [*sequence-number*] **permit udp** *source* [*source-port*] **destination** [*destination-port*]
5. 必要に応じてステップ 4 を繰り返し、シーケンス番号でステートメントを追加します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **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。

	コマンドまたはアクション	目的
ステップ 2	**ipv4 access-list** *name* 例： `RP/0/RSP0/CPU0:router# ipv4 access-list acl_1`	IPv4 アクセス リスト コンフィギュレーション モードを開始し、アクセス リスト acl_1 を設定します。
ステップ 3	[*sequence-number*] **remark** *remark* 例： `RP/0/RSP0/CPU0:router(config-ipv4-acl)# 10 remark Do not allow user1 to telnet out`	（任意）名前付きアクセス リストで後続の **permit** ステートメントに関するコメントを記述できます。 • 注釈は最大 255 文字まで可能で、これより長い文字は切り捨てられます。 • 注釈は **permit** ステートメントの前後に設定できますが、一貫性のある場所にする必要があります。
ステップ 4	[*sequence-number*] **permit udp** *source* [*source-port*] **destination** [*destination-port*] 例： `RP/0/RSP0/CPU0:router(config-ipv4-acl)# 20 permit udp 172.16.0.0/24 eq 5000 host 225.0.0.1 eq 5000`	次の条件を指定して送信元ポートと宛先ポートを指定できます。 • ビデオ モニタリングでは udp のみサポートされます。 • パケットの送信元のネットワークまたはホスト番号を指定するには、*source* キーワードを使用します。 • 送信元に適用するワイルドカードビットを指定するには、オプションの *source-wildcard* 引数を使用します。 • パケットの送信先のネットワークまたはホスト番号を指定するには、*destination* キーワードを使用します。 • 宛先に適用するワイルドカード ビットを指定するには、オプションの *destination-wildcard* 引数を使用します。
ステップ 5	必要に応じてステップ 4 を繰り返し、シーケンス番号でステートメントを追加します。 エントリを削除するには、**no** *sequence-number* コマンドを使用します。	アクセス リストは変更できます。
ステップ 6	次のいずれかのコマンドを使用します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config)# end`	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

	コマンドまたはアクション	目的
	または `RP/0/RSP0/CPU0:router(config)# commit`	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。

クラスマップの設定

ここでは、フロー分類子を設定します。 これは個々のフローに一致するか、いくつかのフローに一致する集約フィルタである場合があります。

手順の概要

1. **configure**
2. **class-map type traffic** *class-map-name*
3. **match access-group ipv4** *acl-name*
4. **end-class-map**
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**class-map type traffic** *class-map-name* 例： `RP/0/RSP0/CPU0:router(config)# class-map type traffic class1`	クラスマップ モードを開始します。 クラスマップ タイプは常に traffic として入力する必要があります。
ステップ 3	**match access-group ipv4** *acl-name* 例： `RP/0/RSP0/CPU0:router(config-cmap)# match access-group ipv4 acl1`	このクラスに一致させる ACL を入力します。 クラスあたり 1 つの ACL のみを一致させることができます。
ステップ 4	**end-class-map** 例： `RP/0/RSP0/CPU0:router(config-cmap)# end-class-map`	クラスマップの設定を完了します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		• 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。

ポリシーマップの設定

ビデオ モニタリングのポリシーマップは、performance-traffic タイプです。 ビデオ モニタリングのポリシーマップでは、階層の 1 レベルのみがサポートされます。 つまり、階層型ポリシーマップ設定はビデオ モニタリングではサポートされません。

ビデオ モニタリングのポリシーマップ設定は次の 3 つで構成されます。

- フローパラメータ設定：インターバル期間、必須履歴インターバル、タイムアウトなど、モニタするフローの各種プロパティを指定します。
- メトリック パラメータ設定：モニタするフローについて計算する必要があるメトリックを指定します。
- 反応パラメータ設定：フローについて生成するアラートのベースとなるパラメータを指定します。

設定階層は、*policy*、*class*、*flow* の順です。 つまり、上で指定されたすべてのパラメータは、ポリシーマップ内の特定のクラスに一致するすべてのフローに適用されます。 特定のクラスに一致するフローに対するフローパラメータと反応パラメータの指定はオプションですが、メトリック パラメータは必須です。

メトリック パラメータを使用したポリシーマップの設定

ポリシーマップのメトリック パラメータは次のとおりです。

- レイヤ 3 パケット レート
- メディア ビット レート（指定された UDP ペイロードにおけるメディア パケット カウントの数およびサイズによる）

（注） レイヤ 3 パケット レートおよびメディア レートには、相互に排他的なコンフィギュレーション コマンドがあります。

ここでは、各メトリック パラメータの設定について説明します。

レイヤ 3 パケット レート

手順の概要

1. **configure**
2. **policy-map type** *performance-traffic policy-map-name*
3. **class type** *traffic class-name*
4. **monitor metric ip-cbr**
5. **rate layer3 packet** *packet-rate* **pps**
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**policy-map type** *performance-traffic policy-map-name* 例： `RP/0/RSP0/CPU0:router(config)# policy-map type performance-traffic policy1`	ポリシーマップ モードを開始します。ポリシーマップ タイプは常に performance traffic として入力する必要があります。
ステップ 3	**class type** *traffic class-name* 例： `RP/0/RSP0/CPU0:router(config-pmap)# class type traffic class-name`	このポリシーに一致させるクラスマップを入力します。1 つのポリシーに対して複数のクラスを指定できます。
ステップ 4	**monitor metric ip-cbr** 例： `RP/0/RSP0/CPU0:router(config- pmap-c)# monitor metric ip-cbr`	IP-CBR メトリック モニタ サブモードを開始します。 （注） 現在は IP-CBR メトリック モニタリングのみがビデオモニタリングでサポートされています。
ステップ 5	**rate layer3 packet** *packet-rate* **pps** 例： `RP/0/RSP0/CPU0:router(config-pmap-c-ipcbr)#`	IP レイヤ 3 パケット レートをパケット/秒（pps）単位で指定します。

	コマンドまたはアクション	目的
	`rate layer3 packet packet-rate pps`	

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

メディア ビット レート

メディア ビット レートのメトリック パラメータは、メディア ビット レート、メディア パケット カウント、およびパケット サイズで構成されます。 レート メディア オプションを使用すると、1 つの UDP パケットに存在するメディア ペイロード パケット（つまり MPEG-2 データグラム）の数および各メディア ペイロードのサイズを指定できます。 メディア ビット レートの指定は必須です。Cisco IOS XR ソフトウェア リリース 3.9.1 では、パケット カウントおよびパケット サイズのデフォルトはありません。 これらの値は設定する必要があります。

（注） メディア ビット レートを 1052800 bps、メディア パケット カウントを 7、メディア パケット サイズを 188 バイトに設定すると、メディア パケット レートはレイヤ 3 で 100 pps になります。 計算は、1052800 / (7 *188*x 8) = 100 pps です。

手順の概要

1. **configure**
2. **policy-map type** *performance-traffic policy-map-name*
3. **class type** *traffic class-name*
4. **monitor metric ip-cbr**
5. **rate media** *bit -rate* **{bps|kbps|mbps|gbps}**
6. **media packet count in-layer3** *packet-count*
7. **media packet size packet-size**
8. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**policy-map type** *performance-traffic policy-map-name* 例： `RP/0/RSP0/CPU0:router(config)# policy-map type performance-traffic policy1`	ポリシーマップ モードを開始します。 ポリシーマップ タイプは常に performance traffic として入力する必要があります。
ステップ 3	**class type** *traffic class-name* 例： `RP/0/RSP0/CPU0:router(config-pmap)# class type traffic class-name`	このポリシーに一致させるクラスマップを入力します。 1 つのポリシーに対して複数のクラスを指定できます。
ステップ 4	**monitor metric ip-cbr** 例： `RP/0/RSP0/CPU0:router(config- pmap-c)# monitor metric ip-cbr`	IP-CBR メトリック モニタ サブモードを開始します。 （注） 現在は IP-CBR メトリック モニタリングのみがビデオ モニタリングでサポートされています。
ステップ 5	**rate media** *bit -rate* **{bps|kbps|mbps|gbps}** 例： `RP/0/RSP0/CPU0:router(config-pmap-c-ipcbr)# rate media 100 mbps`	フローのメディア ビット レートを bps、kbps、mbps、または gbps で指定します。 ここで設定をコミットできます。 オプション パラメータを指定することもできます。 （注） メディア ビット レートのデフォルトの単位は kbps です。

	コマンドまたはアクション	目的
ステップ 6	**media packet count in-layer3** *packet-count* 例： `RP/0/RSP0/CPU0:router(config-pmap-c-ipbr)# media packet count in-layer3 10`	各 IP ペイロードのメディア パケット数を指定します。
ステップ 7	**media packet size packet-size** 例： `RP/0/RSP0/CPU0:router(config-pmap-c-ipcbr)# media packet size 188`	IP ペイロード内の各メディア パケットのサイズをバイト単位で指定します。

ステップ 8	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

フロー パラメータを使用したポリシーマップの設定

ポリシーマップのフロー パラメータは次のとおりです。

ビデオモニタリングでは、データプレーンによってフローおよび各インターバルの最後にエクスポートされるメトリックが継続的にモニタされます。 このインターバルの期間およびフロー（履

歴）ごとに保存する必要があるインターバルの数をオプションで指定することもできます。 これらのフロー パラメータはフローごとに指定できます。

- インターバル期間：このインターバル期間の最後にメトリックがエクスポートされます。 5 の倍数（10 ～ 300 秒の任意の値）で指定します。 デフォルト値は 30 です。
- 履歴：フローごとに保存する必要があるフロー情報（フローID、メトリックなど）を含むインターバル数。 1 ～ 60 の任意の値を指定できます。 デフォルト値は 10 です。
- タイムアウト：インターバル期間の倍数で指定し、この時間が経過すると、非アクティブなフローが削除対象としてマークされます。 2 ～ 60 の任意の値を指定できます。 デフォルト値は 0 です （注：タイムアウト値 0 には特別な意味があります。フローはタイムアウトせず、スタティック フローになります）。
- クラスあたりの最大フロー：ポリシーの各クラスでモニタする必要があるフローの最大数。 1 ～ 1024 の任意の値を指定できます。 デフォルト値は 1024 です。

手順の概要

1. **configure**
2. **policy-map type** *performance-traffic policy-map-name*
3. **class type** *traffic class-name*
4. monitor parameters
5. {**interval duration** *duration* | **flows** *number of flows* | **history** *intervals* | **timeout** *duration*}
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**policy-map type** *performance-traffic policy-map-name* 例： `RP/0/RSP0/CPU0:router(config)# policy-map type performance-traffic policy1`	ポリシーマップ モードを開始します。 ポリシーマップ タイプは常に performance traffic として入力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 3	**class type** *traffic class-name* 例： `RP/0/RSP0/CPU0:router(config-pmap)# class type traffic class-name`	このポリシーに一致させるクラスマップを入力します。 1 つのポリシーに対して複数のクラスを指定できます。
ステップ 4	monitor parameters 例： `RP/0/RSP0/CPU0:router(config-pmap-c)# monitor parameters`	フロー モニタ サブモードを開始します。
ステップ 5	{**interval duration** *duration* | **flows** *number of flows* | **history** *intervals* | **timeout** *duration*} 例： `RP/0/RSP0/CPU0:router(config-pmap-c-fparm)# interval duration 10`	• フローごとにインターバル期間を指定するには、**interval duration** オプションを選択します。範囲は 10 ～ 300（5 の倍数）です。 デフォルト値は 30 です。 • フローごとに保存するインターバルデータの最大数を指定するには、**history** オプションを選択します。 1 ～ 60 の任意の値を指定できます。 デフォルト値は 10 です。 • インターバル期間の倍数でタイムアウト値を指定するには、**timeout** オプションを選択します。この時間が経過すると、非アクティブなフローは削除対象としてマークされます。 範囲は 2 ～ 60 です。 デフォルト値は 0 で、スタティック フローを示します。 • クラスごとにモニタできるフローの最大数を指定するには、**flows** オプションを選択します。 範囲は 1 ～ 1024 です。 デフォルト値は 1024 です。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		◦ **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。

反応パラメータを使用したポリシーマップの設定

ポリシーマップの反応パラメータはオプションです。

反応パラメータは、ユーザがフロー品質を指定するための直接参照です。 フローは継続的にモニタされ、インターバル期間の最後に、ユーザによって特定のパラメータに指定されたしきい値を超えたかどうかを確認するために統計情報が調べられます。 しきい値を超えた場合は、コンソールにsyslogアラームが生成されます。 アラームが設定されると、その条件に対してこれ以上syslog通知は発行されなくなります。

ポリシーマップを設定するには次の反応パラメータが使用されます。

- メディア レート変動（MRV）：フローのMRV統計情報がユーザによって指定されたしきい値を超えると、ビデオ モニタリングが反応してアラームを生成します。
- 遅延係数：フローの遅延係数統計情報がユーザによって指定されたしきい値を超えると、ビデオ モニタリングが反応してアラームを生成します。
- メディア停止：フローが停止すると、ビデオ モニタリングが反応してアラームを生成します。これは、1回の完全なモニタリング インターバルの間にフローのパケットを受信しなかったことを示します。
- パケット レート：フローのパケット レートがユーザによって指定されたしきい値を超えると、ビデオ モニタリングが反応してアラームを生成します。
- フロー カウント：各クラスのフロー カウントがユーザによって指定されたしきい値を超えると、ビデオ モニタリングが反応してアラームを生成します。

手順の概要

1. **configure**
2. **policy-map type** *performance-traffic policy-map-name*
3. **class type** *traffic class-name*
4. **react react-id** {**mrv** | **delay-factor** | **packet-rate** | **flow-count** | **media-stop**}
5. **threshold type immediate**
6. **threshold value** {**ge** | **gt** | **le** | **lt** | **range**} *limit*
7. **action syslog**
8. **alarm severity** {**error** | **critical** | **alert** | **emergency**}
9. **alarm type** {**discrete** | **grouped**}
10. 次のいずれかのコマンドを使用します。
    - **end**
    - **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	**policy-map type** *performance-traffic policy-map-name* 例： `RP/0/RSP0/CPU0:router(config)# policy-map type performance-traffic policy1`	ポリシーマップ モードを開始します。 ポリシーマップ タイプは常に performance traffic として入力する必要があります。
ステップ 3	**class type** *traffic class-name* 例： `RP/0/RSP0/CPU0:router(config-pmap)# class type traffic class-name`	このポリシーに一致させるクラスマップを入力します。 1 つのポリシーに対して複数のクラスを指定できます。
ステップ 4	**react react-id** {**mrv**	**delay-factor**	**packet-rate**	**flow-count**	**media-stop**} 例： `RP/0/RSP0/CPU0:router(config- pmap-c)#`	反応パラメータ コンフィギュレーション サブモードを開始します。 ここで指定する反応 ID は、クラスごとに一意である必要があります。

	コマンドまたはアクション	目的
	`react 1 mrv`	（注） media-stop反応パラメータでは、threshold-typeおよびthreshold-valueオプションは適用されません。flow-count反応パラメータでは、alarm-typeオプションは適用されません。
ステップ5	**threshold type immediate** 例： `RP/0/RSP0/CPU0:router(config-pmap-c-react)# threshold type immediate`	しきい値のトリガー タイプを指定します。 現在使用可能なしきい値タイプはimmediateです。
ステップ6	**threshold value** {**ge** | **gt** | **le** | **lt** | **range**} *limit* 例： `RP/0/RSP0/CPU0:router(config-pmap-c-react)# threshold value ge 50`	しきい値のトリガー値範囲を指定します。
ステップ7	**action syslog** 例： `RP/0/RSP0/CPU0:router(config-pmap-c-react)# action syslog`	**action**キーワードでは、しきい値制限を超えたときに実行するアクションを指定します。 現在、syslogアクションが唯一使用可能なオプションです。
ステップ8	**alarm severity** {**error** | **critical** | **alert** | **emergency**} 例： `RP/0/RSP0/CPU0:router(config-pmap-c-react)# alarm severity critical`	syslogのアラーム重大度を指定します。
ステップ9	**alarm type** {**discrete** | **grouped**} 例： `RP/0/RSP0/CPU0:router(config-pmap-c-react)# alarm type discrete`	アラーム タイプを指定します。 しきい値を超えたすべてのフローに対して個別アラームが発生します。 特定の数または割合のフローがしきい値を超えた場合は、グループ化されたアラームが発生します。
ステップ10	次のいずれかのコマンドを使用します。 • **end** • **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]:`

	コマンドまたはアクション	目的

	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		◦ **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。

インターフェイスのサービス ポリシーの設定

ビデオモニタリングサービスをイネーブルにするには、設定したポリシーマップを入力方向のインターフェイスに関連付ける必要があります。

イーサネットバンドルインターフェイスの場合、サービスポリシーは、物理メンバインターフェイスではなくバンドル親インターフェイスにのみ関連付けることができます。イーサネットバンドル サブインターフェイスの場合は、サブインターフェイスにのみ関連付けることができます。VLANサブインターフェイスの場合は、サービス ポリシーをメインインターフェイスに関連付けることはできません。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **service-policy type performance-traffic input** *policy-name*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **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** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface type interface-path-id`	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。 • type 引数でインターフェイス タイプを指定します。 インターフェイス タイプの詳細については、疑問符（?）オンライン ヘルプ機能を使用してください。 • instance 引数で物理インターフェイス インスタンスまたは仮想インスタンスを指定します。 • 物理インターフェイス インスタンスの表記方法は rack/slot/module/port です。 値を区切るスラッシュ（/）は、表記の一部として必要です。 • 仮想インターフェイスインスタンスの番号範囲は、インターフェイス タイプによって異なります。
ステップ 3	**service-policy type performance-traffic input** *policy-name* 例： `RP/0/RSP0/CPU0:router(config-if)# service-policy type performance-traffic input policy1`	ポリシーを入力方向のインターフェイスに関連付けます。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **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。

	コマンドまたはアクション	目的
		◦**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トラップおよびクローンの設定

トラップおよびクローンは既存のビデオモニタリングサービスの拡張であり、現在のコントロールプレーンインフラストラクチャをトラップおよびクローンの設定に対応できるように拡張できます。

フローのタプル情報（送信元および宛先 IP アドレス）を使用してトラップをインストールできます。これにより、最終的にリモートデバイスまたはローカルプローブによって一致するパケットがさらに分析されます。

ここでは、一般的なビデオ モニタリング シナリオにおけるトラップおよびクローン プロセスの動作方法を示します。

- 適切なパッケージ（マルチキャストおよびビデオ PIE）をインストールしてビデオ モニタリングをイネーブルにし、ACL、クラスマップ、ポリシーマップを設定してポリシーマップをインターフェイスにバインドする必要があります。
- フローの送信元と宛先を指定してクローンするフローを指定することで、トラップおよびクローンを設定する必要があります。
- トラップが VidMon コントロール プレーンによってデータ プレーンにインストールされると、VidMon データ プレーンは指定されたフローのパケットのクローンを開始します。
- クローンされたパケットは、リモート モニタリング デバイスに転送されてさらに分析されます。

（注） **show performance traffic clone profile** コマンドを使用すると、インストールされているトラップを確認できます。 ビデオ モニタリングのトラップおよびクローン機能は、マルチキャスト トラフィックに対してのみサポートされ、ユニキャスト フローについては、ユーザが SPAN を設定する必要があります。 マルチキャストでは、ビデオ モニタリングのトラップおよびクローン機能は、クローン インターフェイスのスタティック IGMP グループを使用して実装されます。 クローン インターフェイスは、ローカル プローブに接続された専用ポートに設定できます。

手順の概要

1. **configure**
2. **performance traffic clone profile**
3. **performance traffic clone profile** *profile_name* **description**
4. **interface** type interface-path-id
5. **clone flow ipv4 source** <*source-ip*> **destination** <*destination-ip*>
6. 次のいずれかのコマンドを使用します。
   - **end**
   - **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	**performance traffic clone profile** 例： `RP/0/RSP0/CPU0:router(config)# performance traffic clone profile`	パフォーマンス トラフィック クローン プロファイル モードを開始します。
ステップ 3	**performance traffic clone profile** *profile_name* **description** 例： `RP/0/RSP0/CPU0:router(config-perf-traf-clone-profile)# performance traffic clone profile profile1 description`	クローン プロファイルに対して説明を設定します。

	コマンドまたはアクション	目的
ステップ 4	**interface** type interface-path-id 例： `RP/0/RSP0/CPU0:router(config-perf-traf-clone-profile)# interface GigabitEthernet 0/0/0/1`	クローン プロファイルに対して出力インターフェイスを設定します。
ステップ 5	**clone flow ipv4 source** <*source-ip*> **destination** <*destination-ip*> 例： `RP/0/RSP0/CPU0:router(config-perf-traf-clone-profile)# clone flow ipv4 23.1.1.1 224.2.2.2`	クローンプロファイルに対してクローンが必要なトラフィック フローを設定します。 （注） 複数のフローを 1 つのクローン プロファイルに関連付けることができます。 同様に、1 つのフローを複数のクローン プロファイルに関連付けることができます。

ステップ 6	次のいずれかのコマンドを使用します。
	• **end**
	• **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

ビデオ モニタリングの実装の設定例

シナリオ 1

イーサネット バンドル インターフェイスに 3 つの物理メンバがあり、マルチキャスト ビデオ トラフィックはそのメンバ上をフローあたり 300 pps で移動しています。

ビデオ モニタリングを使用して、このイーサネット バンドル上のすべてのフローをモニタします。フロー単位のトラフィック負荷が予想レートの 10 % を超えた場合に、クリティカル レベルのアラームを発生させます。 遅延係数が 4 ミリ秒を超えた場合はエラー レベルのアラームを発生させます。 収集した統計情報を 10 秒ごとに報告します。 フローがアクティブであるかぎり、報告した統計情報を 10 分間保管します。 パケットを 30 秒間受信しなかった場合はフロー統計情報を削除します。

例

```
ipv4 access-list sample-acl
 10 permit udp any any
!
class-map type traffic match-any sample-class
 match access-group ipv4 sample-acl
 end-class-map
!
policy-map type performance-traffic sample-policy
 class type traffic sample-class
  monitor parameters
   interval duration 10
   history 60
   timeout 3
  !
  monitor metric ip-cbr
   rate layer3 packet 300 pps
  !
  react 100 mrv
   threshold type immediate
   threshold value gt 10.00
   action syslog
   alarm severity error
   alarm type discrete
  !
  react 101  delay-factor
   threshold type immediate
   threshold value gt 4.00
   action syslog
   alarm severity error
   alarm type discrete
  !
 !
 end-policy-map
!
interface Bundle-Ether10
 ipv4 address 172.192.1.1 255.255.255.0
 service-policy type performance-traffic input sample-policy
!
```

```
interface TenGigE0/6/0/0
 bundle id 10 mode on
!
interface TenGigE0/6/0/1
 bundle id 10 mode on
!
interface TenGigE0/6/0/2
 bundle id 10 mode on
!
```

シナリオ 2

VLAN サブインターフェイスは、共通のマルチキャスト グループ アドレス 225.0.0.1 とさまざまな UDP ポート番号を持つ 100 個のビデオ ストリームを伝送しています。 IP レイヤの予想パケット レートは不明ですが、メディア ビット レートは 1052800 bps であることがわかっています。 メディア ペイロードには MPEG-2 で符号化された CBR フローが含まれ、デフォルトのパケット化が使用されます（つまり、1 つの UDP ペイロードに 7 つの MPEG パケットがあり、各パケットの長さは 188 バイトです）。

100 を超えるフローはモニタしません。 フローが停止してもフローのタイムアウトと削除を実行しませんが、停止したフローの割合が 90 % を超えた場合はエラー レベルのアラームを発生させます。

例

```
ipv4 access-list sample-acl
 10 permit udp any host 225.0.0.1
!
class-map type traffic match-any sample-class
 match access-group ipv4 sample-acl
 end-class-map
!
policy-map type performance-traffic sample-policy
 class type traffic sample-class
  monitor parameters
   flows 100
!
  monitor metric ip-cbr
   rate media 1052800 bps
  !
  react 100 media-stop
   action syslog
   alarm severity error
   alarm type grouped percent 90
  !
 end-policy-map
!
interface GigabitEthernet0/0/0/0
 no shutdown
!
interface GigabitEthernet0/0/0/0.1
 encapsulation dot1q 500
 ipv4 address 172.192.1.1 255.255.255.0
 service-policy type performance-traffic input sample-policy
!
```

monitor metric ip-cbr で、次の 2 行はデフォルトであるため設定する必要はありません。

- media packet count in-layer3 7
- media packet size 188

ただし、これらのパラメータがデフォルト値と異なる場合は設定する必要があります。

シナリオ 3

メイン インターフェイスに、マルチキャスト ストリームのグループが 3 つあり、最初のグループでは UDP 宛先ポートが 1000、2 番めのグループでは 2000、3 番めのグループでは 3000 と 4000 です。 これらの 3 つのストリーム グループは、それぞれ 100 pps、200 pps、300 pps で移動します。

各グループのフローの最大数を 300 フローに制限し、フローがプロビジョニングされたフロー容量の 90 % に達した場合にエラー レベルのアラームを発生させます。

例

```
ipv4 access-list sample-acl-1
 10 permit udp any any eq 1000
!
ipv4 access-list sample-acl-2
 10 permit udp any any eq 2000
!
ipv4 access-list sample-acl-3
 10 permit udp any any eq 3000
 20 permit udp any any eq 4000
!
class-map type traffic match-any sample-class-1
 match access-group ipv4 sample-acl-1
 end-class-map
!
class-map type traffic match-any sample-class-2
 match access-group ipv4 sample-acl-2
 end-class-map
!
class-map type traffic match-any sample-class-3
 match access-group ipv4 sample-acl-3
 end-class-map
!
policy-map type performance-traffic sample-policy
 class type traffic sample-class-1
  monitor parameters
   interval duration 10
   history 60
   timeout 3
   flows 300
  !
  monitor metric ip-cbr
   rate layer3 packet 100 pps
  !
  react 100 flow-count
   threshold type immediate
   threshold value gt 270
   action syslog
   alarm severity error
  !
class type traffic sample-class-2
  monitor parameters
   interval duration 10
   history 60
   timeout 3
   flows 300
  !
  monitor metric ip-cbr
   rate layer3 packet 200 pps
  !
  react 100 flow-count
   threshold type immediate
   threshold value gt 270
   action syslog
   alarm severity error
```

```
  !
class type traffic sample-class-1
  monitor parameters
   interval duration 10
   history 60
   timeout 3
   flows 300
  !
  monitor metric ip-cbr
   rate layer3 packet 300 pps
  !
  react 100 flow-count
   threshold type immediate
   threshold value gt 270
   action syslog
   alarm severity error
  !
 !
 end-policy-map
!
interface GigabitEthernet0/0/0/0
 ipv4 address 172.192.1.1 255.255.255.0
 service-policy type performance-traffic input sample-policy
!
```

シナリオ 4

10GE メイン インターフェイスは、スポーツ スタジアムの 6 台の高精度（HD）カメラに直接接続されたデジタル コンテンツ マネージャ（DCM）から 6 つの HD ビデオ ストリームを受信します。各 HD ビデオ ストリームは圧縮されず、帯域幅はレイヤ 2 で 1.611 Gbps であり、これは 140625 pps に相当します。 これらの 6 つの受信ストリームはマルチキャスト グループが 225.0.0.1 ～ 225.0.0.6、UDP ポート番号は 5000 です。

フローの遅延係数が 2 ミリ秒を超えた場合、またはメディア損失率が 5% を超えた場合にクリティカル レベルのアラームを発生させます。 10 秒のインターバルを使用し、最大の履歴を保管します。 このインターフェイスでは 6 つを超えるフローはモニタしません。 非アクティブなフローをタイムアウトにしません。

例

```
ipv4 access-list sample-acl
 10 permit udp any eq 5000 225.0.0.0/24 eq 5000
!
class-map type traffic match-any sample-class
 match access-group ipv4 sample-acl
 end-class-map
!
policy-map type performance-traffic sample-policy
 class type traffic sample-class
  monitor parameters
   interval duration 10
   history 60
   flows 6
  !
  monitor metric ip-cbr
   rate layer3 packet 140625 pps
  !
  react 100 mrv
   threshold type immediate
   threshold value gt 5.00
   action syslog
   alarm severity critical
   alarm type discrete
  !
```

```
  react 200 delay-factor
   threshold type immediate
   threshold value gt 2.00
   action syslog
   alarm severity critical
   alarm type discrete
  !
 end-policy-map
!
interface TenGigE0/2/0/0
 ipv4 address 172.192.1.1 255.255.255.0
 service-policy type performance-traffic input sample-policy
!
```

シナリオ 5

イーサネット インターフェイスが Cisco ASR 9000 シリーズ ルータに設定され、そこをマルチキャスト ビデオ トラフィックが移動しています。 ビデオ モニタリングを使用して、このイーサネット インターフェイス上のすべてのビデオ フローのパフォーマンスをモニタします。 ビデオ モニタリングのトラップおよびクローン機能を使用して、これらのフロー パケットをトラップし、指定された出力インターフェイスにクローン（または複製）します。

指定された出力インターフェイスにクローンするフローを含むトラップおよびクローン プロファイルを設定します。 プロファイルに説明を追加します。

例

```
Performance traffic clone profile profile1
   Description video flows monitored by vidmon
   Interface GigE 0/1/1/1
   flow ipv4 source 23.1.1.1 destination 231.2.2.2
```

シナリオ 6

100GE メイン インターフェイスは、ユニキャスト トラフィックの 5 つの高精度（HD）ビデオ ストリームを受信しています。 各 HD ビデオ ストリームは圧縮されず、そのビット レートは 3 Gbps です。 各ストリームは CBR フローで、パケット レートが 284954 pps であることがわかっています。 これらのストリームの送信元は 192.1.1.2 で、宛先は 10.1.1.1 ～ 10.1.1.5 です。 送信元と宛先の両方に UDP ポート 7700 が使用されています。

フローの遅延係数が 5 ミリ秒を超えた場合、または CBR フロー レートが予想公称レートの 10 % 以上低下した場合、クリティカル レベルのアラームを発生させます。 30 秒のインターバルを使用し、10 インターバルを履歴として保管します。 このポートはまもなく低レートの VoD フローを受信することがわかっているため、最大フロー カウントとして 4000 を許可します。 10.1.1.0/24 サブネット宛てのストリームのみをモニタします。 品質低下が検出された場合は、アラームを syslog 出力以外に NMS システムに報告します。

例

```
ipv4 access-list sample-acl
 10 permit udp 192.1.1.2/32 eq 7700 10.1.1.0/24 eq 7700
!
class-map type traffic match-any sample-class  match access-group ipv4 sample-acl
end-class-map
!
```

```
policy-map type performance-traffic sample-policy  class type traffic sample-class
  monitor parameters
   interval duration 30
   history 10
   flows 4000
  !
  monitor metric ip-cbr
   rate layer3 packet 284954 pps
  !
  react 100 mrv
   threshold type immediate
   threshold value lt 10.00
   action syslog
   action snmp
   alarm severity critical
   alarm type discrete
  !
  react 200 delay-factor
   threshold type immediate
   threshold value gt 5.00
   action syslog
   action snmp
   alarm severity critical
   alarm type discrete
  !
 end-policy-map
!
interface HundredGigE0/1/0/1
 ipv4 address 172.192.1.1 255.255.255.0
 service-policy type performance-traffic input sample-policy
!
```

その他の参考資料

関連資料

関連項目	参照先
マルチキャスト コマンド リファレンス	『*Cisco ASR 9000 Series Aggregation Services Router Multicast Command Reference*』
スタートアップ資料	『*Cisco ASR 9000 Series Aggregation Services Router Getting Started Guide*』
モジュラ Quality of Service コマンド リファレンス	『*Cisco ASR 9000 Series Aggregation Services Router Modular Quality of Service Command Reference*』

MIB

MIB	MIB のリンク
—	Cisco IOS XR ソフトウェアを使用して MIB を特定およびダウンロードするには、次の URL 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。http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFC

RFC	タイトル
RFC4445	『Proposed Media Delivery Index (MDI)』

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

索 引

数字

A

B

C

D

F

G

H

I

L

M

N

O

P

R

S

T

U

V

あ

い

う

お

か

き

く

け

こ

さ

し

す

せ

そ

た

ち

て

と

ね

の

は

ひ

ふ

へ

ほ

ま

も

ゆ

よ

り

る