Cisco ASR 9000 シリーズ ルータ への RIP の実装

ルーティング情報プロトコル（RIP）は、小規模ネットワークの自律システム（AS）内で情報を交換するために設計された従来のディスタンス ベクトル内部ゲートウェイ プロトコル（IGP）です。

このモジュールでは、基本的なRIPルーティングを実装するための概念とタスクについて説明します。 Cisco IOS XR ソフトウェア は、RFC 2453 に記載されているとおり RIP バージョン 1（RIPv1）との下位互換性をサポートするRIPバージョン2（RIPv2）の標準実装をサポートします。

次の機能に関連する RIP 設定情報については、このモジュールの関連資料，（28 ページ）の項を参照してください。

- マルチプロトコル ラベル スイッチング（MPLS）レイヤ 3 バーチャル プライベート ネットワーク（VPN）
- Site of Origin（SoO）のサポート

（注） Cisco IOS XR ソフトウェアでの RIP の詳細、およびこのモジュールに記載されている RIP コマンドの詳細については、このモジュールの関連資料，（28 ページ）の項を参照してください。設定タスクを実行中に表示される他のコマンドのマニュアルを見つけるには、オンラインで『*Cisco ASR 9000 Series Aggregation Services Router Commands Master List*』を検索してください。

RIP の実装の機能履歴

リリース	変更箇所
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	変更なし。

リリース	変更箇所
リリース 4.0.0	キーチェーンを使用した MD5 認証機能が追加されました。

RIP 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。

RIP の実装に関する情報

RIP 機能の概要

RIP バージョン 1（RIP v1）は、実装が最も容易なルーティング プロトコルと見なされるクラスフル ディスタンスベクトル プロトコルです。 OSPF とは異なり、RIP はユーザ データグラム プロトコル（UDP）データ パケットをブロードキャストして、階層型ではなくフラットなインターネットワークのルーティング情報を交換します。 ネットワークの複雑さとネットワーク管理時間が軽減されます。 一方で、RIP v1 はクラスフル ルーティング プロトコルのため、単一のルートで表されるホスト、サブネットまたはネットワークの連続するブロックのみを許可するために、この有用性を厳しく制限しています。

RIP v2 は、RIP アップデート パケットではより多くの情報の伝送が可能であり、次のサポートが含まれます。

- ルート集約
- クラスレス ドメイン間ルーティング（CIDR）
- 可変長サブネット マスク（VLSM）
- 自律システムと再配布の使用

- RIP アドバタイズメント用のマルチキャスト アドレス 224.0.0.9

RIP が異なるルートの値を評価するときに使用するメトリックは、ホップ カウント です。ホップ カウントは、ルート内で経由されるルータ数です。 直接接続しているネットワークのメトリックはゼロです。到達不能のネットワークのメトリックは 16 です。 RIP はこのようにメトリックの範囲が小さいので、大規模なネットワークに適したルーティング プロトコルではありません。

ルーティング情報のアップデートはデフォルトでは30秒ごとにアドバタイズされ、隣接ルータで検出された新しいアップデートはルーティング テーブルに格納されます。

RFC 2453 に記載のとおり、RIP バージョン 2（RIP v2）のみが Cisco IOS XR ソフトウェアでサポートされていて、デフォルトでは、このソフトウェアは RIP v2 パケットのみを送受信します。 一方で、バージョン 1 パケットとバージョン 2 パケットのバージョン タイプのパケットの両方、またはいずれか一方のみを送受信、または送信と受信のいずれかを実行するようにソフトウェアをインターフェイスごとに設定できます。

RIP を使用する利点は、次のとおりです。

- さまざまなネットワーク デバイスとの互換性
- 使用される帯域幅、設定、および管理時間の観点からして、オーバーヘッドがわずかなため小規模ネットワークに最適
- レガシー ホスト システムのサポート

RIP は使用が容易なため、世界中のネットワークに実装されています。

RIP のスプリット ホライズン

通常、ブロードキャスト型の IP ネットワークに接続し、ディスタンスベクトル ルーティング プロトコルを使用しているルータは、スプリットホライズンメカニズムを使用して、ルーティングがループする可能性を軽減しています。 スプリット ホライズンでは、情報が発生したインターフェイス外部のルータによって、ルートに関する情報がアドバタイズされることが防止されます。通常、この動作は、複数のルータ間の（特にリンクが破損した場合の）通信を最適化します。

セカンダリ IPアドレスを使用してインターフェイスを設定し、スプリットホライズンがイネーブルの場合、すべてのセカンダリ アドレスからアップデートを送信できないことがあります。 スプリットホライズンをディセーブルにしない場合、1つのルーティング アップデートは、1つのネットワーク番号ごとに送信されます。

（注） スプリット ホライズン機能は、デフォルトでイネーブルになっています。 一般的に、適切にルートをアドバタイズするために動作で変更が必要なことが確実である場合を除き、スプリット ホライズンのデフォルト状態を変更しないことを推奨します。

RIP のルート タイマー

RIP では、ルーティング アップデートの頻度、ルートが無効になるまでの時間、および他のパラメータなどの変数を決めるいくつかのタイマーを使用します。 これらのタイマーを調整して、ルーティングプロトコルパフォーマンスがインターネットワークのニーズにより適合するように調整するには、次のタイマー調整を実行します。

- ルーティング アップデートを送信する頻度（アップデートの秒単位の間隔）
- ルートが無効と宣言された後の間隔（秒単位）
- より適切なパスに関するルーティング情報が抑制されている間隔（秒単位）
- RIP トポロジ テーブルからルートが削除される前に経過する必要がある時間（秒単位）
- RIP アップデート パケット間の遅延時間

最初の 4 つのタイマー調整は、**timers basic** コマンドで設定されます。 **output-delay** コマンドは、RIP アップデート パケット間の遅延時間を変更します。 設定の詳細については、RIP のカスタマイズ, （11 ページ）を参照してください。

また、ソフトウェアの IP ルーティングのサポートを調整して、多様な IP ルーティング アルゴリズムのコンバージェンスを高速化でき、必要に応じて冗長ルータへのドロップ バックが迅速にできます。 総体的な結果として、迅速なリカバリが重要な状況で、ネットワークのエンドユーザの作業が中断する問題が最小限に抑えられます。

RIP のルート再配布

再配布とは、異なるルーティング ドメインでルーティング情報を交換できる機能のことです。 異なるルーティング ドメイン間をルーティングするネットワーキングデバイスは、境界ルータと呼ばれています。これらのデバイスが 1 つのルーティング プロトコルから別のルーティング プロトコルにルートを挿入します。 ルーティング ドメイン内のルータは、境界ルータに再配布が実装されていないかぎり、ドメイン内部のルートのみを認識します。

ルーティング ドメインで RIP を実行しているとき、インターネットワーク内で複数のルーティングプロトコルを使用してそれらの間でルートを再配布する必要がある場合があります。 次に、一般的な理由をいくつか示します。

- 他のプロトコルから RIP にルートをアドバタイズするため（スタティック、接続済み、OSPF、および BGP など）。
- RIP から EIGRP などの新しい内部ゲートウェイ プロトコル（IGP）に移行するため。
- ホスト システムをサポートするために、いくつかのルータでルーティング プロトコルを保持する一方で、他の部門グループのルータをアップグレードするため。
- 多様なルータ ベンダー環境間で通信するため。 基本的にはネットワークの一部ではシスコに固有のプロトコルを使用して、シスコ以外のデバイスと通信するために RIP を使用する場合があります。

また、ルート再配布を使用すると、企業は異なるルーティング プロトコルをそれぞれのプロトコルが特に効果的な作業グループまたは領域で実行できます。 Cisco IOS XR ルート再配布は、ユーザに対して単一ルーティング プロトコルのみを使用するように制限を加えないことにより、ダイバーシティによって技術的利点を最大化する一方でコストを最小化する優れた機能です。

インターネットワークへのルート再配布の実装に関しては、非常に単純にもでき、また非常に複雑にもできます。 単純な単一方向再配布の例は、RIP がイネーブルなルータにログインして、**redistribute static** コマンドを使用して、スタティック接続のみをバックボーン ネットワークにアドバタイズして RIP ネットワークをパススルーさせることです。 複雑な例では、ルーティング ループ、互換性のないルーティング情報、および不整合なコンバージェンス時間を考慮する必要があり、複数のルーティング プロトコルが管理上のコストを実行しているときにシスコ ルータが最適なパスを選択する方法を調査して、これらの問題が発生する理由を判断する必要があります。

RIP のデフォルトのアドミニストレーティブ ディスタンス

アドミニストレーティブ ディスタンスは、IP ルーティング情報の送信元の信頼性を示す評価基準として使用されます。 RIP などのダイナミック ルーティング プロトコルが設定されているときに、ルーティング情報の交換に再配布機能を使用する場合、適切なディスタンスの重みを設定できるように他のルート送信元のデフォルトのアドミニストレーティブ ディスタンスを認識していることが大切です。

次の表に、ルーティング プロトコルのデフォルトのアドミニストレーティブ ディスタンスを示します。

表 *1* : **各ルーティング プロトコルのデフォルトのアドミニストレーティブ ディスタンス**

ルーティング プロトコル	アドミニストレーティブ ディスタンス値
接続されているインターフェイス	0
インターフェイス外部のスタティック ルート	0
ネクスト ホップへのスタティック ルート	1
EIGRP サマリー ルート	5
外部 BGP	20
内部 EIGRP	90
OSPF	110
IS-IS	115
RIP バージョン 1 および 2	120
外部 EIGRP	170

ルーティング プロトコル	アドミニストレーティブ ディスタンス値
内部 BGP	200
不明	255

アドミニストレーティブ ディスタンスは、0 ～ 255 の整数です。 通常は、値が大きいほど、信頼性の格付けが下がります。 アドミニストレーティブ ディスタンスが 255 の場合はルーティング情報の送信元をまったく信頼できないため、無視する必要があります。 アドミニストレーティブ ディスタンス値は主観的なため、この値を選択するための定量的方法はありません。

RIP のルーティング ポリシー オプション

ルート ポリシーは、**route-policy** キーワードおよび **end-policy** キーワードで囲まれた一連のステートメントと式によって構成されます。 個別のコマンド（1 行に 1 つのコマンド）の集合ではなく、ルートポリシー内のステートメントには相互に関連するコンテキストがあります。 そのため、個別のコマンドを各行に記すのではなく、各ポリシーまたはセットは独立した設定オブジェクトとして、1 つのユニットとして使用、入力、操作できます。

ポリシー設定の各行は論理サブユニットです。 **then**、**else**、**end-policy** キーワードの後ろには、少なくとも 1 つの新しい行を続ける必要があります。 また、パラメータ リストの閉じ括弧、および AS パス セット、コミュニティ セット、拡張コミュニティ セットまたはプレフィックス セットへの参照にある名前文字列の後ろには新しい行が必要です。 ルート ポリシー、AS パス セット、コミュニティ セット、拡張コミュニティ セット、またはプレフィックス セットの定義の前には、少なくとも新しい行が 1 行必要です。 アクション ステートメントの後ろには 1 行以上の新しい行を続けることができます。 名前付き AS パス セット、コミュニティ セット、拡張コミュニティ セット、プレフィックス セットのカンマ区切りの後ろには 1 行以上の新しい行を続けることができます。 新しい行はポリシー式の論理ユニットの最後に記される必要があります。他の場所に記すことはできません。

RIP でのキーチェーンを使用した認証

Cisco IOS XR ルーティング情報プロトコル（RIP）でのキーチェーンを使用した認証には、キーチェーン認証に基づいて RIP インターフェイスのすべての RIP プロトコル トラフィックを認証するメカニズムが備えられています。 このメカニズムでは、Cisco IOS XR セキュリティ キーチェーン インフラストラクチャを使用して秘密キーを保存および取得し、インターフェイスごとに着信および発信トラフィックを認証するために使用します。

キーチェーン管理は、相互に信頼を確立する前に、キーなどの秘密を交換するすべてのエンティティに対して共有秘密を設定する一般的な認証メソッドです。 Cisco IOS XR ソフトウェアのルーティング プロトコルおよびネットワーク管理アプリケーションでは多くの場合、ピアとの通信中のセキュリティを向上させるために認証を使用します。

ヒント Cisco IOS XR ソフトウェア システム セキュリティ コンポーネントは、キーチェーン管理などのさまざまなシステム セキュリティ機能を実装します。 キーチェーン管理の概念、設定タスク、例、キーチェーン管理の設定に使用するコマンドの詳細については、次のマニュアルを参照してください。

- 『*Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide*』の「*Implementing Keychain Management*」モジュール
- 『*Cisco ASR 9000 Series Aggregation Services Router System Security Command Reference*』の「*Keychain Management Commands*」モジュール

（注） キーチェーン自体には関連性がないため、キーチェーンはピアとキーを（認証のために）使用して通信する必要のあるアプリケーションで使用する必要があります。 キーチェーンには、存続期間に基づいてキーおよびロールオーバーを処理するセキュアなメカニズムが備えられています。 Cisco IOS XR キーチェーン インフラストラクチャは、キーチェーンの秘密キーのヒットレス ロールオーバーを処理します。

IOS XR キーチェーン データベースにキーチェーンを設定してあり、特定の RIP インターフェイスにも同様に設定されている場合、そのインターフェイスのすべての着信および発信 RIP トラフィックの認証にキーチェーンが使用されます。 認証キーチェーンが（デフォルト VRF またはデフォルト以外の VRF の）RIP インターフェイスに設定されていないかぎり、すべての RIP トラフィックが信頼できると見なされて、着信 RIP トラフィックおよび発信 RIP トラフィックに対してセキュリティを保護する認証メカニズムは使用されません。

RIP は、キー付きメッセージ ダイジェスト モードとクリア テキスト モードの 2 つのモードを使用します。 キーチェーン メカニズムを使用する認証を設定するには、**authentication keychain** *keychain-name* **mode** {**md5** | **text**} コマンドを使用します。

キーチェーンが RIP インターフェイスに設定されているが、キーチェーンが実際はキーチェーン データベースに設定されていない場合、またはキーチェーンが MD5 暗号化アルゴリズムを使用して設定されていない場合、そのインターフェイスのすべての着信 RIP パケットはドロップされます。 発信パケットは認証データなしで送信されます。

インターフェイスの着信 RIP トラフィック

次に、インターフェイスがキーチェーンを使用して設定されている場合の、RIP インターフェイスのすべての着信 RIP パケットの検証基準を示します。

条件	結果
RIP インターフェイスに設定されているキーチェーンがキーチェーン データベースに存在していません。	パケットはドロップされます。 コンポーネント レベルの RIP デバッグ メッセージは、認証失敗の特定の詳細を確認できるようにログに記録されます。

条件	結果
キーチェーンが MD5 暗号化アルゴリズムを使用して設定されていません。	パケットはドロップされます。コンポーネントレベルの RIP デバッグ メッセージは、認証失敗の特定の詳細を確認できるようにログに記録されます。
メッセージの最初の（かつ最初だけの）エントリのアドレス ファミリ識別子が 0xFFFF ではない場合、認証は使用されません。	パケットはドロップされます。コンポーネントレベルの RIP デバッグ メッセージは、認証失敗の特定の詳細を確認できるようにログに記録されます。
「認証データ」の MD5 ダイジェストが無効であることが検出されました。	パケットはドロップされます。コンポーネントレベルの RIP デバッグ メッセージは、認証失敗の特定の詳細を確認できるようにログに記録されます。
それ以外の場合、パケットは残りの処理のために転送されます。	

インターフェイスの発信 RIP トラフィック

次に、インターフェイスがキーチェーンを使用して設定されている場合の、RIP インターフェイスのすべての発信 RIP パケットの検証基準を示します。

条件	結果
RIP インターフェイスに設定されているキーチェーンがキーチェーン データベースに存在しています。	同じキーチェーンを使用してパケットを認証するようにもリモート ルータが設定されている場合、RIP パケットはリモート/ピア エンドの認証チェックをパスします。
キーチェーンが MD5 暗号化アルゴリズムを使用して設定されています。	同じキーチェーンを使用してパケットを認証するようにもリモート ルータが設定されている場合、RIP パケットはリモート/ピア エンドの認証チェックをパスします。
それ以外の場合、RIP パケットは認証チェックに失敗します。	

RIP 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。

（注） 設定の変更を保存するには、システムでプロンプトが表示されたら、変更を確定する必要があります。

RIP のイネーブル化

このタスクでは、RIPルーティングをイネーブルにしてRIPルーティング プロセスを確立します。

はじめる前に

IP アドレスの設定前に RIP を設定できますが、少なくとも 1 つの IP アドレスが設定されるまで RIP ルーティングは発生しません。

手順の概要

1. **configure**
2. **router rip**
3. **neighbor** *ip-address*
4. **broadcast-for-v2**
5. **interface** *type interface-path-id*
6. **receive version** { **1** | **2** | **1 2** }
7. **send version** { **1** | **2** | **1 2** }
8. 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	**router rip** 例： `RP/0/RSP0/CPU0:router(config)# router rip`	RIP ルーティング プロセスを設定します。

	コマンドまたはアクション	目的
ステップ 3	**neighbor** *ip-address* 例： `RP/0/RSP0/CPU0:router(config-rip)# neighbor 172.160.1.2`	（任意）RIP プロトコル情報を交換する隣接ルータを定義します。
ステップ 4	**broadcast-for-v2** 例： `RP/0/RSP0/CPU0:router(config-rip)# broadcast-for-v2`	（任意）RIP v2 マルチキャスト アドレス（224.0.0.9）ではなく、ブロードキャスト IP アドレスにバージョン 2 パケットのみを送信するように RIP を設定します。このコマンドは、インターフェイスまたはグローバル コンフィギュレーション レベルで適用できます。
ステップ 5	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-rip)# interface GigabitEthernet 0/1/0/0`	（任意）RIP ルーティング プロトコルを実行するインターフェイスを定義します。
ステップ 6	**receive version** {**1** | **2** | **1 2**} 例： `RP/0/RSP0/CPU0:router(config-rip-if)# receive version 1 2`	（任意）次のパケットを受信するようにインターフェイスを設定します。 • RIP v1 のみ • RIP v2 のみ • RIP v1 および RIP v2 の両方
ステップ 7	**send version** {**1** | **2** | **1 2**} 例： `RP/0/RSP0/CPU0:router(config-rip-if)# send version 1 2`	（任意）次のパケットを送信するようにインターフェイスを設定します。 • RIP v1 のみ • RIP v2 のみ • RIP v1 および RIP v2 の両方

ステップ 8	次のいずれかを実行します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/CPU0:router(config-rip-if)# end`
	または
	`RP/0/RSP0/CPU0:router(config-rip-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

	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。

RIP のカスタマイズ

このタスクでは、ネットワーク タイミングおよびルート エントリの受け入れのために RIP 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 rip**
3. **auto-summary**
4. **timers basic** *update invalid holddown flush*
5. **output-delay** *delay*
6. **nsf**
7. **interface** *type interface-path-id*
8. **metric-zero-accept**
9. **split-horizon disable**
10. **poison-reverse**
11. 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	**router rip** 例： `RP/0/RSP0/CPU0:router(config)# router rip`	RIP ルーティング プロセスを設定します。
ステップ 3	**auto-summary** 例： `RP/0/RSP0/CPU0:router(config-rip)# auto-summary`	（任意）ネットワーク レベルのルートへのサブネット ルートの自動ルート集約をイネーブルにします。 • デフォルトでは、自動集約はディセーブルになっています。 （注） サブネットを切断した場合は **no** キーワードを使用して、自動ルート集約をディセーブルにし、ソフトウェアによるサブネットおよびホスト ルーティング情報をクラスフル ネットワーク境界を越えた送信を許可します。
ステップ 4	**timers basic** *update invalid holddown flush* 例： `RP/0/RSP0/CPU0:router(config-rip)# timers basic 5 15 15 30`	（任意）RIP ネットワーク タイマーを調整します。 （注） 現在およびデフォルトのタイマー値を確認するには、**show rip** コマンドからの出力を参照します。
ステップ 5	**output-delay** *delay* 例： `RP/0/RSP0/CPU0:router(config-rip)# output-delay 10`	（任意）送信された RIP アップデートのパケット間遅延を変更します。 （注） ハイエンドルータが高速で送信しているときに、宛先が低速ルータで、この速い速度で受信できない可能性がある場合は、このコマンドを使用します。
ステップ 6	**nsf** 例： `RP/0/RSP0/CPU0:router(config-rip)# nsf`	（任意）RIP プロセスのシャットダウンまたはリスタート後に RIP ルートに NSF を設定します。

	コマンドまたはアクション	目的
ステップ 7	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-rip)# interface GigabitEthernet 0/1/0/0`	（任意）RIP ルーティング プロトコルを実行するインターフェイスを定義します。
ステップ 8	**metric-zero-accept** 例： `RP/0/RSP0/CPU0:router(config-rip-if)# metro-zero-accept`	（任意）メトリックにゼロ（0）が設定されたアップデートパケットで受信されたルート エントリの受け入れをネットワーキング デバイスで許可します。受信したルート エントリはメトリックに 1 が設定されています。
ステップ 9	**split-horizon disable** 例： `RP/0/RSP0/CPU0:router(config-rip-if)# split-horizon disable`	（任意）スプリット ホライズン メカニズムをディセーブルにします。 • デフォルトでは、スプリット ホライズンはイネーブルです。 • 一般に、アプリケーションで正しくルートをアドバタイズするために変更が必要なことが分かっている場合を除き、**split-horizon** コマンドのデフォルト状態を変更しないことを推奨します。シリアル インターフェイスでスプリット ホライズンがディセーブルで、そのインターフェイスがパケットスイッチド ネットワークに接続されている場合、そのネットワークの関連マルチキャスト グループ内にあるすべてのネットワーキング デバイスに対し、スプリット ホライズンをディセーブルにする必要があります。
ステップ 10	**poison-reverse** 例： `RP/0/RSP0/CPU0:router(config-rip-if)# poison-reverse`	RIP ルータ アップデートのポイズン リバース処理をイネーブルにします。
ステップ 11	次のいずれかを実行します。 • **end** • **commit** 例： `RP/0/RSP0/CPU0:router(config-rip-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]:` ◦ **yes** 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-rip-if)# 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。

ルーティング情報の制御

このタスクでは、ルーティング アップデートの交換および伝搬を制御または防止する方法について説明します。

次に、ルーティング アップデートを制御または防止するいくつかの理由を示します。

- WAN リンクのアップデート トラフィックを遅くするか停止するため。オンデマンド WAN リンクのアップデート トラフィックを制御しないと、リンクは常にアップ状態のままです。デフォルトでは、RIP ルーティング アップデートは 30 秒おきに発生します。
- ルーティング ループを防止するため。冗長パスがある場合、またはルートを別のルーティング ドメインに再配布している場合、いずれかのパスの伝搬をフィルタします。
- アップデートで受信したネットワークをフィルタするため。特定のデバイスの 1 つ以上のルートの解釈を他のルータに学習させない必要がある場合、その情報を抑制できます。
- 他のルータによるダイナミックなルート処理を防止するため。インターフェイスに入るルーティング アップデートを処理したくない場合、その情報を抑制できます。
- 帯域幅を節約するため。必要のないルーティング アップデート トラフィックを削減することによって、データ トラフィックに使用可能な帯域幅を最大化できます。

手順の概要

1. **configure**
2. **router rip**
3. **neighbor** *ip-address*
4. **interface** *type interface-path-id*
5. **passive-interface**
6. exit
7. **interface** *type interface-path-id*
8. **route-policy** { **in** | **out** }
9. 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	**router rip** 例： `RP/0/RSP0/CPU0:router(config)# router rip`	RIP ルーティング プロセスを設定します。
ステップ 3	**neighbor** *ip-address* 例： `RP/0/RSP0/CPU0:router(config-rip)# neighbor 172.160.1.2`	（任意）RIP プロトコル情報を交換する隣接ルータを定義します。
ステップ 4	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-rip)# interface GigabitEthernet 0/1/0/0`	（任意）RIP ルーティング プロトコルを実行するインターフェイスを定義します。

	コマンドまたはアクション	目的
ステップ 5	**passive-interface** 例： `RP/0/RSP0/CPU0:router(config-rip-if)# passive-interface`	（任意）明示的に設定されたネイバー宛てを除き、インターフェイスの RIP アップデートの送信を抑制します。
ステップ 6	exit 例： `RP/0/RSP0 /CPU0:router(config-rip-if)# exit`	（任意）ルータを次に高いコンフィギュレーション モードへ戻します。
ステップ 7	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-rip)# interface GigabitEthernet 0/2/0/0`	（任意）RIP ルーティング プロトコルを実行するインターフェイスを定義します。
ステップ 8	**route-policy** { **in** | **out** } 例： `RP/0/RSP0/CPU0:router(config-rip-if)# route-policy out`	（任意）RIP ネイバーにアドバタイズするアップデートや、RIP ネイバーから受信するアップデートに、ルーティング ポリシーを適用します。

ステップ 9	次のいずれかを実行します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:routerconfig-rip-if)# end`
	または
	`RP/0/RSP0/CPU0:router(config-rip-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。

	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。

RIP のルート ポリシーの作成

このタスクでは、ルート ポリシーを定義して、RIP プロセスのインスタンスに付加する方法を示します。 ルート ポリシーは、次の目的で使用できます。

- 送信および受信されたルートの制御
- 再配信されるルートの制御
- デフォルト ルートの起点の制御

ルート ポリシー定義は、**route-policy** コマンドおよび *name* 引数の後にオプション ポリシー ステートメントのシーケンスが続き、**end-policy** コマンドで終了する構成です。

ルート ポリシーはルーティング プロトコルのルートに適用されてはじめて役に立ちます。

手順の概要

1. **configure**
2. **route-policy** *name*
3. **set rip-metric** *number*
4. **end-policy**
5. 次のいずれかを実行します。
   - **end**
   - **commit**
6. **configure**
7. **router rip**
8. **route-policy** *route-policy-name* { **in** | **out** }
9. 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	**route-policy** *name* 例： `RP/0/RSP0/CPU0:router(config)# route-policy IN-IPv4`	ルート ポリシーを定義して、ルート ポリシー 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	**set rip-metric** *number* 例： `RP/0/RSP0/CPU0:router(config-rpl)# set rip metric 42`	（任意）RIP メトリック属性を設定します。
ステップ 4	**end-policy** 例： `RP/0/RSP0/CPU0:router(config-rpl)# end-policy`	ルートポリシーの定義を終了して、ルートポリシー 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	次のいずれかを実行します。
	• **end**
	• **commit**
	
	例：
	
	`RP/0/RSP0/CPU0:router(config-rpl)# end`
	または
	
	`RP/0/RSP0/CPU0:router(config-rpl)# 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 6	**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 7	**router rip** 例： `RP/0/RSP0/CPU0:router(config)# router rip`	RIP ルーティング プロセスを設定します。
ステップ 8	**route-policy** *route-policy-name* { **in** | **out** } 例： `RP/0/RSP0/CPU0:router(config-rip)# route-policy rp1 in`	RIP ネイバーにアドバタイズするアップデートや、RIP ネイバーから受信するアップデートに、ルーティング ポリシーを適用します。

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-rip)# end`
	または
	`RP/0/RSP0/CPU0:router(config-rip)# 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。

RIP 認証キーチェーンの設定

デフォルト以外の VRF の IPv4 インターフェイスの RIP 認証キーチェーンの設定

RIP 認証キーチェーンをデフォルト以外の VRF の IPv4 インターフェイスに設定するには、次のタスクを実行します。

はじめる前に

キーチェーンを RIP インターフェイス/VRF に適用するためには、『*Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide*』の「*Implementing Keychain Management*」モジュールに記されているコンフィギュレーション コマンドを使用して、Cisco IOS XR キーチェーン データベース内にすべてのキーチェーンを設定する必要があります。

authentication keychain *keychain-name* および **mode md5** コンフィギュレーションは、IOS XR キーチェーン データベースにまだ設定されていないキーチェーン、または MD5 暗号アルゴリズムを使用せずに IOS XR キーチェーン データベースに設定されているキーチェーンの名前を受け入れます。 ただし、両方の場合ですべての着信パケットはインターフェイスでドロップされ、送信パケットは認証データなしで送信されます。

手順の概要

1. **configure**
2. **router rip**
3. **vrf** *vrf_name*
4. **interface** *type interface-path-id*
5. 次のいずれかのコマンドを使用します。
   - **authentication keychain** *keychain-name* **mode md5**
   - **authentication keychain** *keychain-name* **mode text**
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 rip** 例： `RP/0/RSP0/CPU0:router(config)#router rip`	RIP ルーティング プロセスを設定します。
ステップ 3	**vrf** *vrf_name* 例： `RP/0/RSP0/CPU0:router(config-rip)#vrf vrf_rip_auth`	デフォルト以外の VRF を設定します
ステップ 4	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-rip-vrf)#interface POS 0/6/0/0`	RIP ルーティング プロトコルを実行するインターフェイスを定義します。

ステップ 5	次のいずれかのコマンドを使用します。
	• **authentication keychain** *keychain-name* **mode md5**
	• **authentication keychain** *keychain-name* **mode text**
	
	例：
	`RP/0/RSP0/CPU0:router(config-rip-if)#authentication keychain key1 mode md5`
	または
	`RP/0/RSP0/CPU0:router(config-rip-if)#authentication keychain key1 mode text`
		RIP の認証キーチェーン モードを設定します。
		• **md5**：キー メッセージ ダイジェスト（md5）認証モード
		• **text**：クリア テキストの認証モード

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。

デフォルトの VRF の IPv4 インターフェイスの RIP 認証キーチェーンの設定

RIP 認証キーチェーンをデフォルト VRF の IPv4 インターフェイスに設定するには、次のタスクを実行します。

はじめる前に

キーチェーンを RIP インターフェイス/VRF に適用するためには、『*Cisco ASR 9000 Series Aggregation Services Router System Security Configuration Guide*』の「*Implementing Keychain Management*」モジュールに記されているコンフィギュレーション コマンドを使用して、Cisco IOS XR キーチェーン データベース内にすべてのキーチェーンを設定する必要があります。

authentication keychain *keychain-name* および **mode md5** コンフィギュレーションは、IOS XR キーチェーン データベースにまだ設定されていないキーチェーン、または MD5 暗号アルゴリズムを使用せずに IOS XR キーチェーン データベースに設定されているキーチェーンの名前を受け入れます。 ただし、両方の場合ですべての着信パケットはインターフェイスでドロップされ、送信パケットは認証データなしで送信されます。

手順の概要

1. **configure**
2. **router rip**
3. **interface** *type interface-path-id*
4. 次のいずれかのコマンドを使用します。
   - **authentication keychain** *keychain-name* **mode md5**
   - **authentication keychain** *keychain-name* **mode text**
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 rip** 例： `RP/0/RSP0/CPU0:router(config)#router rip`	RIP ルーティング プロセスを設定します。
ステップ 3	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config-rip)#interface POS 0/6/0/0`	RIP ルーティング プロトコルを実行するインターフェイスを定義します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **authentication keychain** *keychain-name* **mode md5**
	• **authentication keychain** *keychain-name* **mode text**
	例：
	`RP/0/RSP0/CPU0:router(config-rip-if)#authentication keychain key1 mode md5`
	または
	`RP/0/RSP0/CPU0:router(config-rip-if)#authentication keychain key1 mode text`
		RIP の認証キーチェーン モードを設定します。
		• **md5**：キー メッセージ ダイジェスト（md5）認証モード
		• **text**：クリア テキストの認証モード

	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。

RIP 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。

基本的な RIP の設定：例

次に、2 つのギガビット イーサネット インターフェイスを RIP を使用して設定する例を示します。

```
interface GigabitEthernet0/6/0/0
 ipv4 address 172.16.0.1 255.255.255.0
!
```

```
interface GigabitEthernet0/6/0/2
 ipv4 address 172.16.2.12 255.255.255.0
!

router rip
 interface GigabitEthernet0/6/0/0
 !
 interface GigabitEthernet0/6/0/2
 !
!
```

プロバイダー エッジでの RIP の設定：例

次に、2 つの VPN ルーティングおよび転送（VRF）インスタンスを使用して PE に基本的な RIP を設定する例を示します。

```
router rip
 interface GigabitEthernet0/6/0/0
 !
 vrf vpn0
  interface GigabitEthernet0/6/0/2
  !
 !
 vrf vpn1
  interface GigabitEthernet0/6/0/3
  !
 !
!
```

各 VRF インスタンスの RIP タイマーの調整：例

次に、各 VPN ルーティングおよび転送（VRF）インスタンスの RIP タイマーを調整する例を示します。

VRF インスタンス vpn0 の場合、**timers basic** コマンドは 10 秒ごとにアップデートをブロードキャストするように設定します。 ルータから 30 秒間送信がないと、そのルートは使用不能と宣言されます。 以降の情報はさらに 30 秒間抑止されます。 フラッシュ期間（45 秒）の終了時に、ルーティング テーブルからルートがフラッシュされます。

VRF インスタンス vpn1 の場合、タイマーは 20、60、60、および 70 秒と異なる調整が行われます。

output-delay コマンドは、vpn1 の RIP アップデートのパケット間遅延を 10 ミリ秒に変更します。デフォルトでは、パケット間遅延はオフになっています。

```
router rip
 interface GigabitEthernet0/6/0/0
 !
 vrf vpn0
  interface GigabitEthernet0/6/0/2
  !
  timers basic 10 30 30 45
 !
 vrf vpn1
  interface GigabitEthernet0/6/0/3
  !
  timers basic 20 60 60 70
```

```
  output-delay 10
 !
!
```

RIP の再配布の設定：例

次に、ボーダー ゲートウェイ プロトコル（BGP）およびスタティック ルートを RIP に再配布する例を示します。

再配布されるルートで使用される RIP メトリックは、ルート ポリシーによって決まります。ルート ポリシーが設定されていないか、ルート ポリシーで RIP メトリックが設定されていない場合は、再配布されるプロトコルに基づいてメトリックが決定されます。BGP によって再配布される VPNv4 ルートの場合、リモート PE ルータで設定された RIP メトリックが有効であれば、それが使用されます。

その他すべての場合（BGP、IS-IS、OSPF、EIGRP、接続済み、スタティック）、**default-metric** コマンドで設定されたメトリックが使用されます。有効なメトリックが決定できない場合、再配布は起こりません。

```
route-policy ripred
  set rip-metric 5
end-policy
!

router rip
 vrf vpn0
  interface GigabitEthernet0/6/0/2
  !
  redistribute connected
  default-metric 3
 !
 vrf vpn1
  interface GigabitEthernet0/6/0/3
  !
  redistribute bgp 100 route-policy ripred
  redistribute static
  default-metric 3
 !
!
```

RIP のルート ポリシーの設定：例

次に、RIP インターフェイスによって受信されるまたは RIP インターフェイスから送信されるルートアップデートを制御するために使用される、着信および発信ルート ポリシーを設定する例を示します。

```
prefix-set pf1
  10.1.0.0/24
end-set
!

prefix-set pf2
  150.10.1.0/24
end-set
!
```

```
route-policy policy_in
  if destination in pf1 then
    pass
  endif
end-policy
!

route-policy pass-all
  pass
end-policy
!

route-policy infil
  if destination in pf2 then
    add rip-metric 2
    pass
  endif
end-policy
!

router rip
 interface GigabitEthernet0/6/0/0
  route-policy policy_in in
 !
 interface GigabitEthernet0/6/0/2
 !
 route-policy infil in
 route-policy pass-all out
```

RIP のパッシブ インターフェイスおよび明示的なネイバーの設定：例

次に、パッシブ インターフェイスおよび明示的なネイバーを設定する例を示します。 インターフェイスがパッシブな場合は、ルーティングアップデートを受信するのみです。つまり、明示的に設定されたネイバー宛てを除き、インターフェイスからアップデートは送信されません。

```
router rip
 interface GigabitEthernet0/6/0/0
  passive-interface
 !
 interface GigabitEthernet0/6/0/2
 !
 neighbor 172.17.0.1
 neighbor 172.18.0.5
!
```

RIP ルートの制御：例

次に、**distance** コマンドを使用して、RIP ルートをルーティング情報ベース（RIB）にインストールする例を示します。 **maximum-paths** コマンドは、RIP ルートごとに許可される最大パス数を制御します。

```
router rip
 interface GigabitEthernet0/6/0/0
  route-policy polin in
 !
 distance 110
 maximum-paths 8
!
```

RIP 認証キーチェーンの設定：例

次に、RIP のデフォルト VRF インターフェイスに認証キーチェーンを適用する例を示します。

```
router rip
 interface POS0/6/0/0
  authentication keychain key1 mode md5
 !
!
end
```

次に、RIP のデフォルト以外のインターフェイスに認証キーチェーンを適用する例を示します。

```
router rip
 vrf rip_keychain_vrf
  interface POS0/6/0/0
   authentication keychain key1 mode md5
  !
 !
!
end
```

その他の参考資料

次の各項では、RIP の実装に関連するその他の資料について説明します。

関連資料

関連項目	参照先
RIP 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、使用に関する注意事項、および例	『*Cisco ASR 9000 Series Aggregation Services Router Routing Command Reference*』
RIP の MPLS VPN サポートの機能情報	『*Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide*』の「*Implementing MPLS Traffic Engineering on* Cisco ASR 9000 シリーズ ルータ 」モジュール
RIP の Site of Origin（SoO）サポートの機能情報	『*Cisco ASR 9000 Series Aggregation Services Router MPLS Configuration Guide*』 の「*Implementing MPLS Traffic Engineering on* Cisco ASR 9000 シリーズ ルータ 」モジュール
Cisco IOS XR スタートアップ ガイド	『*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 on* Cisco ASR 9000 シリーズ ルータ 」モジュール

標準

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

MIB

MIB	MIB のリンク
—	Cisco IOS XR ソフトウェアを使用して MIB の場所を特定してダウンロードするには、次の URL にある Cisco MIB Locator を使用して、[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 2453	『RIP Version 2』

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 サイトには、数千ページに及ぶ検索可能な技術情報があります。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もあります。 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/techsupport