Cisco Nexus 3548 スイッチ NX-OS ユニキャスト ルーティング構成ガイド、リリース 10.3(x)

初版：2022 年 7 月 24 日

最終更新：2023 年 2 月 15 日

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

【注意】シスコ製品をご使用になる前に、安全上の注意（ **www.cisco.com/jp/go/safety_warning/** ）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

はじめに

ここでは、*Cisco Nexus 3548* スイッチ *NX-OS* ユニキャスト ルーティング構成ガイドの対象読者、構成、および表記法について説明します。また、関連マニュアルの入手方法についても説明します。

この章は、次の項で構成されています。

対象読者

このマニュアルを使用するには、IPおよびルーティングのテクノロジーに関する詳しい知識が必要です。

表記法

コマンドの説明では、次の表記法を使用しています。

表記法	説明
太字	コマンドおよびキーワードは太字で示しています。
イタリック体	ユーザーが値を指定する引数は、イタリック体で示しています。
[ ]	角カッコの中の要素は、省略可能です。
[x | y | z]	どれか1つを選択できる省略可能なキーワードは、角カッコで囲み、縦棒で区切って示しています。

表記法	説明
string	引用符を付けない一組の文字。string の前後には引用符を使用しません。引用符を使用すると、その引用符も含めて string とみなされます。

出力例では、次の表記法を使用しています。

screen フォント	スイッチに表示される端末セッションおよび情報は、screen フォントで示しています。
太字の screen フォント	ユーザが入力しなければならない情報は、太字のスクリーン フォントで示しています。
イタリック体の *screen* フォント	ユーザが値を指定する引数は、イタリック体の screen フォントで示しています。
<>	パスワードのように出力されない文字は、山カッコ（<>）で囲んで示しています。
[ ]	システム プロンプトに対するデフォルトの応答は、角カッコで囲んで示しています。
!、#	コードの先頭に感嘆符（!）またはポンド記号（#）がある場合には、コメント行であることを示します。

このマニュアルでは、次の表記法を使用しています。

（注） 「注釈」を意味します。役立つ情報やこのマニュアルに記載されていない参照資料を紹介しています。

注意 「要注意」の意味です。機器の損傷またはデータ損失を予防するための注意事項が記述されています。

ヒント 「問題解決に役立つ情報」です。

Nexus 3548 スイッチ NX-OS ソフトウェアの関連資料

Cisco Nexus 3548 スイッチ ソフトウェア全体のマニュアル セットは、次の URL にあります。

http://www.cisco.com/en/US/products/ps11541/tsd_products_support_series_home.html

リリース ノート

リリース ノートは、次の URL から入手できます。

http://www.cisco.com/en/US/products/ps11541/prod_release_notes_list.html

インストレーション ガイドおよびアップグレード ガイド

インストレーション ガイドおよびアップグレード ガイドは、次の URL から入手できます。

http://www.cisco.com/en/US/products/ps11541/prod_installation_guides_list.html

このカテゴリのマニュアルには、次が含まれます。

- 『Cisco Nexus 5000 Series, Cisco Nexus 3000 Series, and Cisco Nexus 2000 Series Safety Information and Documentation』
- 『Regulatory, Compliance, and Safety Information for the Cisco Nexus 5000 Series, Cisco Nexus 3000 Series, and Cisco Nexus 2000 Series』
- 『Cisco Nexus 3000 Series Hardware Installation Guide』

ライセンス情報

NX-OS の機能ライセンスについては、Cisco NX-OS Licensing Guideを参照してください。次の URL から入手できます：
http://www.cisco.com/en/US/docs/switches/datacenter/sw/nx-os/licensing/guide/b_Cisco_NX-OS_Licensing_Guide.html

コンフィギュレーション ガイド

コンフィギュレーション ガイドは、次の URL から入手できます。

http://www.cisco.com/en/US/products/ps11541/products_installation_and_configuration_guides_list.html

このカテゴリのマニュアルには、次が含まれます。

- 『Fundamentals Configuration Guide』
- 『Interfaces Configuration Guide』
- 『Layer 2 Switching Configuration Guide』
- 『Multicast Configuration Guide』
- 『Quality of Service Configuration Guide』
- 『Security Configuration Guide』
- 『System Management Configuration Guide』
- 『Unicast Routing Configuration Guide』
- 『Verified Scalability Guide for Cisco NX-OS』

コマンド リファレンス

コマンド リファレンスは、次の URL で入手できます。

https://www.cisco.com/c/en/us/support/switches/nexus-3000-series-switches/products-command-reference-list.html

エラー メッセージおよびシステム メッセージ

システム メッセージ リファレンス ガイドは、次の URL で入手できます。

http://www.cisco.com/en/US/products/ps11541/products_system_message_guides_list.html

マニュアルに関するフィードバック

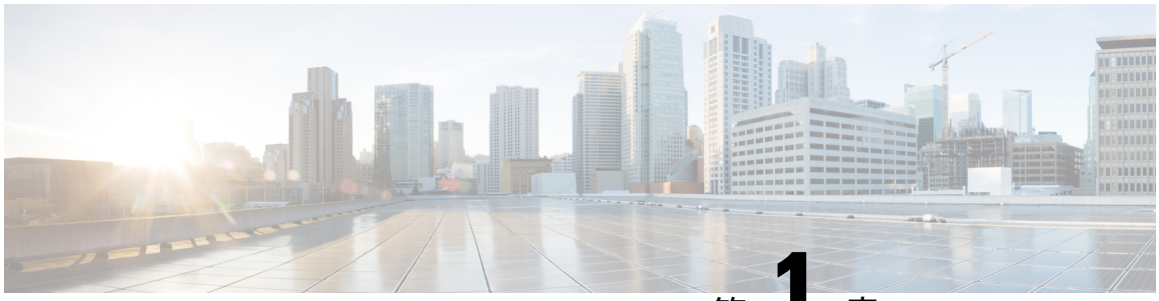
このマニュアルに関する技術的なフィードバック、または誤りや記載もれなどお気づきの点がございましたら、nexus3k-docfeedback@cisco.com までご連絡ください。ご協力をよろしくお願いいたします。

通信、サービス、およびその他の情報

- シスコからタイムリーな関連情報を受け取るには、Cisco Profile Manager でサインアップしてください。
- 重要な技術によりビジネスに必要な影響を与えるには、Cisco Services にアクセスしてください。
- サービス リクエストを送信するには、Cisco Support にアクセスしてください。
- 安全で検証済みのエンタープライズクラスのアプリケーション、製品、ソリューション、およびサービスを探して参照するには、Cisco Marketplace にアクセスしてください。
- 一般的なネットワーキング、トレーニング、認定関連の出版物を入手するには、Cisco Press にアクセスしてください。
- 特定の製品または製品ファミリの保証情報を探すには、Cisco Warranty Finder にアクセスしてください。

Cisco バグ検索ツール

Cisco Bug Search Tool（BST）は、シスコ製品とソフトウェアの障害と脆弱性の包括的なリストを管理する Cisco バグ追跡システムへのゲートウェイとして機能する、Web ベースのツールです。BST は、製品とソフトウェアに関する詳細な障害情報を提供します。



第 1 章

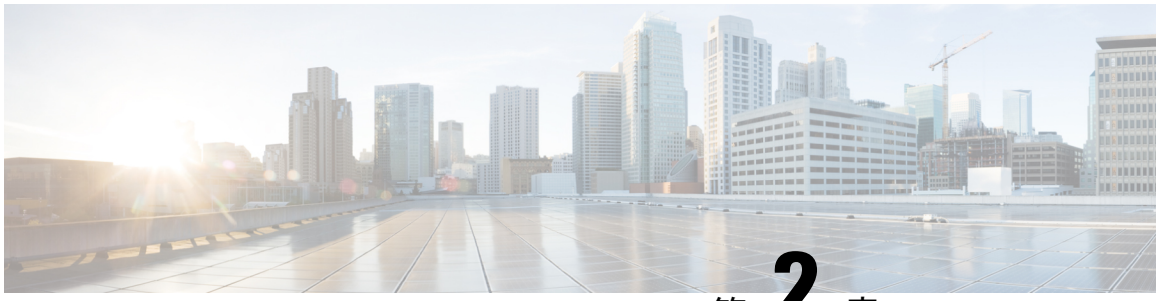
新機能および変更された機能に関する情報

新機能および変更された機能に関する情報

表 *1* : 新機能および変更された機能

特長	説明	変更が行われたリリース	参照先
新機能の更新なし	最初の 10.3(1) リリース	10.3(1)F	N/A

新機能および変更された機能に関する情報



第 2 章

概要

この章では、Cisco NX-OS でのレイヤ 3 ユニキャスト ルーティング プロトコルの基本概念を紹介します。

この章は、次の項で構成されています。

ライセンス要件

Cisco NX-OS ライセンス方式の推奨の詳細と、ライセンスの取得および適用の方法については、『*Cisco NX-OS Licensing Guide*』を参照してください。

レイヤ 3 ユニキャスト ルーティングについて

レイヤ 3 ユニキャスト ルーティングには 2 つの基本的動作（最適なルーティング パスの決定およびパケットの交換）があります。ルーティング アルゴリズムを使用すると、ルータから宛先までの最適なパス（経路）を計算できます。この計算方法は、選択したアルゴリズム、ルー

トメトリック、そしてロードバランシングや代替パスの探索などの考慮事項により異なります。

ルーティングの基礎

ルーティングプロトコルは、メトリックを使用して、宛先までの最適なパスを調べます。メトリックとは、パス帯域幅などの、ルーティングアルゴリズムが宛先までの最適なパスを決定するために使用する測定基準です。パスを決定しやすいように、ルーティングアルゴリズムは、ルート情報（IP 宛先アドレス、および次のルータまたはホップのアドレスなど）を含むルーティング テーブルを初期化して維持します。宛先とネクスト ホップの関連付けにより、ルータは、宛先までの途中にあるネクストホップとなる特定のルータにパケットを送信すると、最適なパスで IP 宛先まで届けられることを判定できます。ルータは、着信パケットを受信すると、宛先アドレスをチェックし、このアドレスをネクスト ホップと関連付けようとします。ルート テーブルの詳細については、「ユニキャスト RIB」の項を参照してください。

ルーティングテーブルには、パスの優先度に関するデータなど、その他の情報が含まれていることもあります。ルータは、メトリックを比較して最適なルートを決定します。これらのメトリックは、使用しているルーティングアルゴリズムの設計によって異なります。「ルーティング メトリック」の項を参照してください。

各ルータは互いに通信し、さまざまなメッセージを送信して、そのルーティングテーブルを維持します。ルーティング更新メッセージはこれらのメッセージのいずれかであり、ルーティング テーブルのすべてまたは一部で構成されます。ルータは、他のすべてのルータからのルーティング更新情報を分析して、ネットワークトポロジの詳細な図を構築できます。ルータ間で送信されるメッセージのもう 1 つの例であるリンクステートアドバタイズメントは、送信ルータのリンク状態を他のルータに通知します。リンク情報を使用して、ルータが、ネットワーク宛先までの最適なルートを決定できるようにすることもできます。詳細については、「ルーティング アルゴリズム」の項を参照してください。

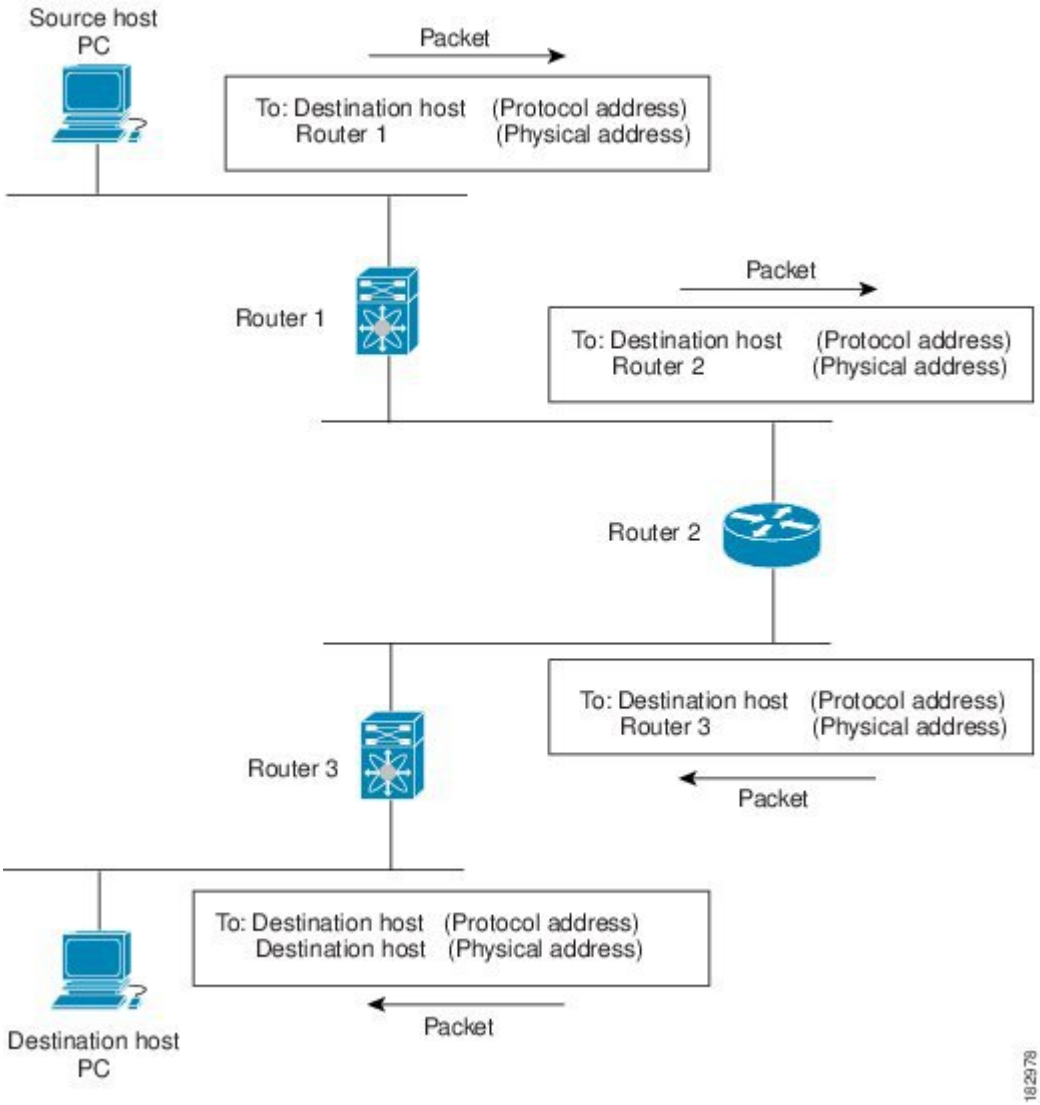
パケット交換

パケット交換では、ホストが、パケットを別のホストに送信する必要があることを決定します。なんらかの方法でルータのアドレスを入手したら、送信元ホストはパケットを明確に、宛先ホストの IP（ネットワーク層）アドレスを含むルータの物理（メディア アクセス コントロール（MAC）レイヤ）アドレス宛に送信します。

ルータは宛先の IP アドレスを調べ、ルーティング テーブルでその IP アドレスを探します。ルータがパケットの転送方法を認識していない場合は、通常はパケットをドロップします。パケットの転送方法がわかった場合、ルータは、宛先の MAC アドレスをネクスト ホップ ルータの MAC アドレスに変更し、パケットを送信します。

ネクストホップが宛先のホストである場合や、同じ交換決定処理を行う別のルータである場合があります。パケットがインターネットワークを介して移動するにつれ、パケットの物理アドレスは変化しますが、プロトコル アドレスは一定のままです（次の図を参照）。

図 *1*: ネットワークを介したパケット ヘッダーの更新



ルーティング メトリック

ルーティング アルゴリズムは、多くの異なるメトリックを使用して最適なルートを決定します。高度なルーティングアルゴリズムは、複数のメトリックに基づいてルートを選択している場合があります。

パス長

パスの長さは、最も一般的なルーティング メトリックです。一部のルーティング プロトコルでは、各ネットワーク リンクに恣意的なコストの割り当てが可能です。この場合、パスの長さは、経由した各リンクに関連付けられたコストの合計となります。それ以外のルーティングプ

ロトコルでは、パケットが送信元から宛先までに経由する必要のある、ルータなどのネットワーク間製品の通過回数を指定するメトリックであるホップ数が定義されます。

Reliability

ルーティングアルゴリズムとの関連における信頼性は、各ネットワークリンクの信頼性（ビット誤り率で示される）です。一部のネットワーク リンクは、他のネットワーク リンクよりダウンする頻度が高い場合があります。ネットワークがダウンした後、特定のネットワークリンクが他のリンクより容易に、または短時間に修復される場合もあります。信頼性のランクを割り当てるときに考慮できる信頼性係数は、一般的にネットワーク リンクに割り当てる任意の数値です。

ルーティング遅延

ルーティング遅延は、送信元から宛先に、インターネットワークを通過してパケットを移動するために必要な時間の長さです。遅延は、中間のネットワーク リンクの帯域幅、経由する各ルータでのポートキュー、中間の全ネットワークリンクでのネットワークの輻輳状況、パケットが移動する物理的な距離など、多くの要素に応じて異なります。ルーティング遅延はいくつかの重要な変数の組み合わせであるため、一般的で便利なメトリックです。

帯域幅

帯域幅は、リンクで使用可能なトラフィック容量です。たとえば、10 ギガビット イーサネット リンクは 1 ギガビット イーサネット リンクより容量が大きく、優れています。帯域幅は、リンクで達成可能な最大スループットですが、帯域幅のより大きいリンクを経由するルートが、帯域幅のより小さいリンクを経由するルートより優れているとは限りません。たとえば、帯域幅の大きいリンクの方が混雑していると、実際には、パケットを宛先に送信するためにさらに長い時間がかかる場合があります。

負荷

負荷は、ルータなどのネットワーク リソースの使用状況の度合いです。負荷は、CPU 使用状況や処理される1秒あたりのパケット数など、さまざまな方法で計算できます。これらのパラメータを継続的にモニタすると、リソースに負担がかかる場合があります。

通信コスト

通信コストは、リンク上でルーティングするための稼働コストの測定単位です。通信コストは重要なメトリックの1つで、特にパフォーマンスより稼働コストの削減が優先される場合に使用されます。たとえば、専用回線での回線遅延が公衆回線より大きくても、使用時間に応じて課金される公衆回線上でなく、自身の専用回線上でパケットを送信できます。

ルータ ID

各ルーティング プロセスには、ルータ ID が関連付けられています。ルータ ID は、システムのあらゆるインターフェイスに設定できます。ルータ ID を構成しなかった場合、Cisco NX-OS は次の基準に基づいてルータ識別子を選択します。

- Cisco NX-OS は、他のあらゆるインターフェイスよりも loopback0 を優先します。loopback0 が存在しなかった場合、Cisco NX-OS は、他のあらゆるインターフェイス タイプよりも、最初のループバック インターフェイスを優先します。
- ループバック インターフェイスを構成しなかった場合、Cisco NX-OS はルータ識別子として構成ファイルの最初のインターフェイスを使用します。Cisco NX-OS がルータ識別子を選択した後にいずれかのループバック インターフェイスを構成した場合は、ループバック インターフェイスがルータ識別子となります。ループバック インターフェイスが loopback0 ではなく、後で loopback0 を IP アドレスで設定した場合は、ルータ ID が loopback0 の IP アドレスに変更されます。
- ルータ ID の元であるインターフェイスが変更されると、新しい IP アドレスがルータ ID となります。他のどのインターフェイスの IP アドレスが変更されても、ルータ ID はまったく変更されません。

自律システム

自律システム（AS）とは、単一の技術的管理エンティティにより制御されるネットワークです。自律システムにより、グローバルな外部ネットワークが個々のルーティング ドメインに分割され、これらのドメインでは、ローカルのルーティング ポリシーが適用されます。この構成により、ルーティング ドメインの管理と一貫したポリシー設定が簡素化されます。

各自律システムは、ルートの再配布により動的にルーティング情報を交換する、複数の内部ルーティング プロトコルをサポートできます。地域インターネット レジストリにより、インターネットに直接接続する各公共自律システムに一意の番号が割り当てられます。この自律システム番号で、ルーティング処理と自律システムの両方が識別されます。

Cisco NX-OS は 4 バイト AS 番号をサポートしています。次の表は、AS 番号の範囲を示します。

表 *2 : AS* 番号

2 バイト番号	AS ドット表記での 4 バイト番号	プレーンテキスト表記での 4 バイト番号	目的
1 ～ 64511	0.1 ～ 0.64511	1 ～ 64511	公共 AS（RIR により割り当てられる）
64512 ～ 65534	0.64512 ～ 0.65534	64512 ～ 65534	専用 AS（ローカルの管理者により割り当てられる）

2 バイト番号	AS ドット表記での 4 バイト番号	プレーンテキスト表記での 4 バイト番号	目的
65535	0.65535	65535	予約済み（Reserved）
なし	1.0 ～ 65535.65535	65536 ～ 4294967295	公共 AS（RIR により割り当てられる）

専用自律システム番号は内部ルーティング ドメインに使用されますが、インターネット上にルーティングされたトラフィック向けに、ルータにより変換される必要があります。ルーティング プロトコルを、専用自律システム番号が外部ネットワークにアドバタイズされるように設定しないでください。デフォルトでは、Cisco NX-OS は専用自律システム番号をルーティング更新情報から削除しません。

（注） 公共ネットワークおよび専用ネットワークの自律システム番号は、インターネット割り当て番号局（IANA）により管理されています。予約済み番号の割り当てを含む自律システム番号の詳細について、または、AS 番号の登録を申請するには、次の URL を参照してください：http://www.iana.org/

コンバージェンス

ルーティング アルゴリズム測定の鍵となる要素の 1 つは、ルータがネットワーク トポロジの変化に対応するために要する時間です。リンク障害など、なんらかの理由でネットワークの一部が変化すると、さまざまなルータのルーティング情報が一致しなくなる場合があります。変化したトポロジに関する情報が更新されているルータと、古い情報が残っているルータがあるためです。コンバージェンスとは、ネットワーク内のすべてのルータが更新され、ルーティング情報が一致するまでにかかる時間の長さです。コンバージェンス時間は、ルーティング アルゴリズムによって異なります。コンバージェンスが速い場合は、不正確なルーティング情報によるパッケージ損失の可能性が小さくなります。

ロード バランシングおよび等コスト マルチパス

ルーティング プロトコルでは、ロード バランシングまたは等コスト マルチパス（ECMP）を使用して、複数のパス上のトラフィックを共有できます。ルータは、特定のネットワークへのルートを複数検出すると、最もアドミニストレーティブ ディスタンスの低いルートをルーティング テーブルにインストールします。ルータが、同じアドミニストレーティブ ディスタンスと宛先までのコストを持つ複数のパスを受信し、インストールすると、ロードバランシングが発生する場合があります。ロードバランシングでは、すべてのパス上にトラフィックが配布され、負荷が共有されます。使用されるパスの数は、ルーティング プロトコルによりルーティング テーブルに配置されるエントリの数に制限されます。Cisco NX-OS は、32 までの宛先パスをサポートします。

Enhanced Interior Gateway Routing Protocol（EIGRP）は、等コストでないロードバランシングもサポートしています。EIGRP の設定方法の詳細については、EIGRP の設定を参照してください。

ルートの再配布

ネットワークに複数のルーティングプロトコルが設定されている場合は、各プロトコルにルートの再配布を設定して、ルーティング情報を共有するように設定できます。たとえば、OSPF（Open Shortest Path First）を設定して、ボーダー ゲートウェイ プロトコル（BGP）で検出したルートをアドバタイズできます。また、スタティック ルートを、どのダイナミック ルーティング プロトコルにも再配布できます。別のプロトコルからのルートを再配布しているルータは、その再配布ルートに対する固定ルート メトリックを設定します。このプロセスにより、異なるルーティング プロトコル間で互換性のないルート メトリックの問題が回避されます。たとえば、EIGRP から OSPF に再配布されたルートには、OSPF が認識できる固定リンク コスト メトリックが割り当てられます。

ルート再配布では、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」セクションを参照）の使用によっても、2 つの異なるルーティング プロトコルで検出されたルートが区別されます。優先ルーティング プロトコルには、より低いアドミニストレーティブディスタンスが与えられており、そのルートが、より高いアドミニストレーティブディスタンスが割り当てられた他のプロトコルからのルートに優先して選択されます。

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、ルーティング情報源の信頼性を示す評価基準です。値が高いほど、信頼性のランクは低くなります。一般的にルートは、複数のプロトコルを通じて検出されます。アドミニストレーティブ ディスタンスは、複数のプロトコルから学習したルートを区別するために使用されます。最もアドミニストレーティブディスタンスが低いルートが IP ルーティング テーブルに組み込まれます。

スタブ ルーティング

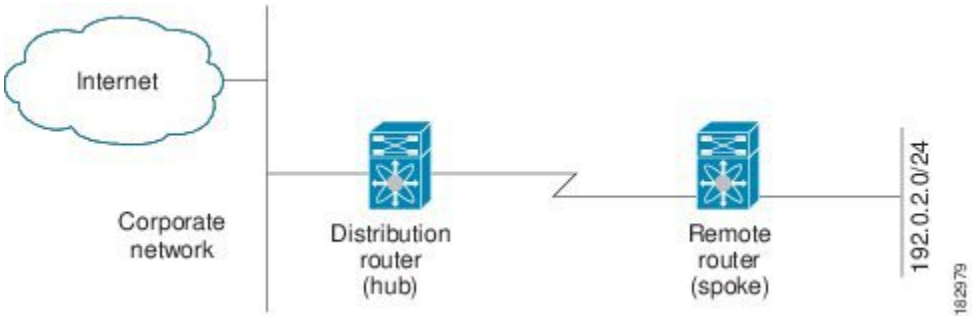
スタブ ルーティングはハブ アンド スポーク型ネットワーク トポロジで使用できます。このトポロジでは、1 つ以上の終端（スタブ）ネットワークが 1 台のリモートルータ（スポーク）に接続され、そのリモート ルータは 1 つ以上のディストリビューション ルータ（ハブ）に接続されています。リモート ルータは、1 つ以上のディストリビューション ルータにのみ隣接しています。リモート ルータへ流れる IP トラフィックのルートは、ディストリビューション ルータ経由のルートのみです。このタイプの設定は、ディストリビューション ルータが直接 WAN に接続されている WAN トポロジで使用されるのが一般的です。ディストリビューション ルータは、さらに多くのリモート ルータに接続できます。ディストリビューション ルータが 100 台以上のリモート ルータに接続されていることも、よくあります。ハブ アンド スポーク型トポロジでは、リモート ルータがすべての非ローカル トラフィックをディストリビューション

ルータに転送する必要があります。これにより、リモート ルータが完全なルーティング テーブルを保持する必要はなくなります。通常、分散ルータは、デフォルトのルートのみをリモート ルータに送信します。

指定されたルートのみが、リモート（スタブ）ルータから伝播されます。スタブ ルータは、サマリー、接続されているルート、再配布されたスタティック ルート、外部ルート、および内部ルートに対するクエリーすべてに、応答として「inaccessible」というメッセージを返します。スタブとして設定されているルータは、自身のスタブ ルータとしてのステータスを報告するために、特殊なピア情報パケットがすべての隣接ルータに送信されます。

スタブ ルータの状態を通知するパケットを受信した隣接ルータは、ルートについてはスタブ ルータに照会しません。また、スタブ ピアを持つルータは、そのピアについては照会しません。スタブ ルータは、ディストリビューション ルータを使用して適切なアップデートをすべてのピアに送信します。次の図は、単純なハブ アンド スポーク型のコンフィギュレーションを示しています。

図 *2:* 単純なハブ アンド スポーク ネットワーク



スタブ ルーティングを使用する場合でも、リモート ルータにルータをアドバタイズできます。図 1-2 は、リモート ルータが、分散ルータのみを使用して企業ネットワークとインターネットにアクセスできることを示しています。この例では、企業ネットワークとインターネットへのパスが常に分散ルータを経由するため、リモート ルータ上の完全なルート テーブルの機能は無意味です。より大規模なルート テーブルを使用しても、リモート ルータに必要なメモリの量が削減されるだけです。使用される帯域幅とメモリは、分散ルータでルートを要約し、フィルタリングすると、削減できます。このネットワーク トポロジでリモート ルータは、他のネットワークから検出されたルートを受信する必要はありません。これは、宛先がどこであっても、リモート ルータは、すべての非ローカル トラフィックを分散ルータに送信する必要があるためです。真のスタブ ネットワークを設定するには、リモート ルータへのデフォルト ルートのみを送信するよう、分散ルータを設定する必要があります。

OSPF はスタブ エリアをサポートしており、EIGRP はスタブ ルータをサポートしています。

（注） EIGRP スタブ ルーティング機能は、スタブ デバイスだけで使用します。スタブ デバイスは、コア中継トラフィックが通過しないネットワーク コアまたはディストリビューション レイヤに接続されたデバイスとして定義されます。リモート ルータへ流れる IP トラフィックのルートは、ディストリビューション ルータ経由のルートのみです。スタブ デバイスがディストリビューション デバイス以外の EIGRP ネイバーを持つことはできません。この制限を無視すると、望ましくない動作が発生します。

ルーティング アルゴリズム

ルーティングアルゴリズムによって、ルータが到達可能性情報を収集して報告する方法、トポロジの変化に対応する方法、宛先までの最適ルートを決定する方法が決まります。ルーティングアルゴリズムにはさまざまなタイプがあり、各アルゴリズムがネットワークやルータ リソースに与える影響もさまざまです。ルーティングアルゴリズムは、最適なルートの計算に影響するさまざまなメトリックを使用します。ルーティングアルゴリズムは、スタティックまたはダイナミック、内部または外部など、タイプで分類できます。

スタティック ルートおよびダイナミック ルーティング プロトコル

スタティック ルートは、手動で設定するルート テーブル エントリです。スタティック ルートは、手動で再設定しない限り、変更されません。スタティック ルートは設計が簡単で、ネットワーク トラフィックが比較的予想しやすい環境や、ネットワーク設計が比較的単純な環境での使用に適しています。

スタティック ルーティング システムはネットワークの変化に対応できないため、絶えず変化する今日の大規模ネットワークには使用すべきではありません。今日のほとんどのルーティング プロトコルは、ダイナミック ルーティング アルゴリズムを使用しています。このアルゴリズムでは、着信ルーティング更新メッセージを分析して、ネットワーク状況の変化に合わせて調整します。メッセージがネットワークが変化したことを示している場合は、ルーティング ソフトウェアはルートを再計算し、新しいルーティング アップデート メッセージを送信します。これらのメッセージがネットワークを通過すると、ルータがそのアルゴリズムを再実行し、それに従ってルーティング テーブルを変更します。

適切であれば、ダイナミック ルーティング アルゴリズムをスタティック ルートで補完することができます。たとえば、各サブネットワークに IP デフォルト ゲートウェイまたは、ラスト リゾート ルータ（ルーティングできないすべてのパケットが送信されるルータ）へのスタティック ルートを設定する必要があります。

内部および外部ゲートウェイ プロトコル

ネットワークを、一意のルーティング ドメインまたは自律システムに分割できます。自律システムは、管理ガイドラインの特定のセットで規制された共通の管理機関の下の内部ネットワークの一部です。自律システム間でのルートを設定するルーティング プロトコルは、外部ゲート

ウェイ プロトコルまたはドメイン間プロトコルと呼ばれます。BGP は、外部ゲートウェイ プロトコルの例です。1 つの自律システム内で使用されるルーティング プロトコルは、内部ゲートウェイ プロトコルまたはドメイン内プロトコルと呼ばれます。EIGRP および OSPF は、内部ゲートウェイ プロトコルの例です。

ディスタンス ベクトル プロトコル

ディスタンス ベクトル プロトコルは、ディスタンス ベクトル アルゴリズム（Bellman-Ford アルゴリズムとも呼ばれます）を使用します。このアルゴリズムにより、各ルータは、そのルーティング テーブルの一部または全部を隣接ルータに送信します。ディスタンス ベクトル アルゴリズムでは、ルートが、ディスタンス（宛先までのホップ数など）および方向（ネクスト ホップ ルータなど）により定義されます。その後、これらのルートは、直接接続されたネイバー ルータにブロードキャストされます。各ルータは、これらの更新情報を使用して、ルーティング テーブルを確認し、更新します。

ルーティング ループを防ぐために、ほとんどのディスタンス ベクトル アルゴリズムはポイズン リバースを指定したスプリット ホライズンを使用します。これは、インターフェイスで検出されたルートを到達不能として設定し、それをそのインターフェイスで、次の定期更新中にアドバタイズするという意味です。この機能により、ルータによるルート更新が、そのルータ自体に返信されなくなります。

ディスタンス ベクトル アルゴリズムは、一定の間隔で更新を送信しますが、ルート メトリックの値の変更に応じて、更新を送信することもできます。このように送信された更新により、ルート コンバージェンス時間の短縮が可能です。Routing Information Protocol（RIP）はディスタンス ベクトル プロトコルの 1 つです。

リンクステート プロトコル

リンク ステート プロトコルは、最短パス優先（SPF）とも呼ばれ、情報を隣接ルータと共有します。各ルータはリンクステート アドバタイズメント（LSA）を構築し、ここに、各リンクおよび直接接続されたネイバー ルータに関する情報が含まれます。

各 LSA にはシーケンス番号があります。ルータが LSA を受信し、そのリンクステート データベースを更新すると、その LSA はすべての隣接ネイバーにフラッディングされます。ルータが同じシーケンス番号の 2 つの LSA（同じルータからの）を受信した場合は、LSA 更新ループを防ぐために、ルータは最後に受信した LSA をネイバー ルータにフラッディングしません。ルータは、受信直後に LSA をフラッディングするため、リンクステート プロトコルのコンバージェンス時間は最小となります。

ネイバー ルータの探索と隣接関係の確立は、リンクステート プロトコルの重要な部分です。ネイバー ルータは、特別な hello パケットを使用して探索されます。このパケットは、各ネイバー ルータのキープアライブ通知としても機能します。隣接関係は、ネイバー ルータ間のリンクステート プロトコルの一般的な動作パラメータ セットで確立されます。

ルータが受信した LSA は、そのリンクステート データベースに追加されます。各エントリは、次のパラメータで構成されます。

- ルータ ID（LSA を構築したルータの）

- ネイバー ID
- リンク コスト
- LSA のシーケンス番号
- LSA エントリの作成時からの経過時間

ルータは、リンクステート データベース上で SPF アルゴリズムを実行し、そのルータの最短パス ツリーを構築します。この SPF ツリーを使用して、ルーティング テーブルにデータが入力されます。

リンクステートアルゴリズムでは、各ルータはネットワークの全体像をそのルーティングテーブルに構築します。リンクステートアルゴリズムが小さな更新を全体的に送信するのに対し、ディスタンスベクトルアルゴリズムは、より大きな更新をネイバールータのみに送信します。

リンクステート アルゴリズムは、より短時間でコンバージェンスするため、ディスタンス ベクトル アルゴリズムより、ルーティング ループがやや発生しにくくなっています。ただし、リンクステートアルゴリズムはディスタンス ベクトル アルゴリズムより、大きな CPU パワーとメモリを必要とします。リンクステートアルゴリズムは、実装とサポートにより多くの費用がかかる場合があります。一般的に、リンクステート プロトコルはディスタンス ベクトル プロトコルよりもスケーラブルです。

OSPF は、リンクステート プロトコルの一例です。

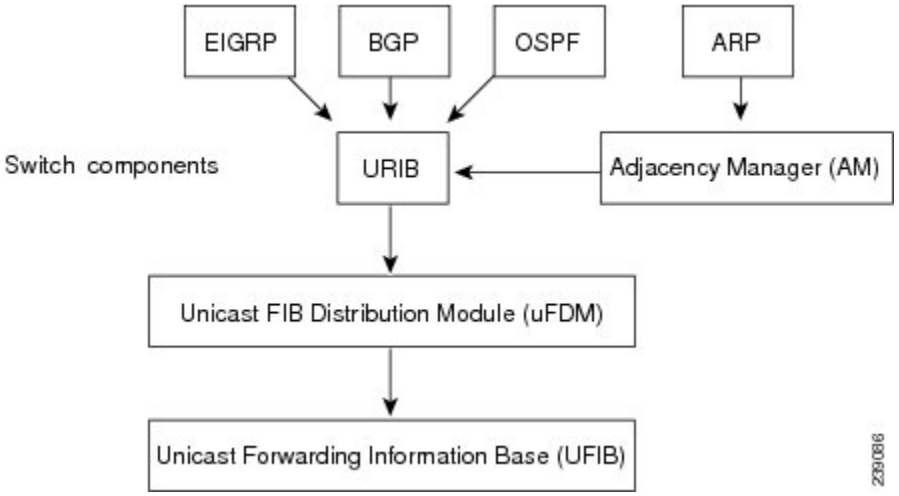
CiscoNX-OS フォワーディング アーキテクチャ

Cisco NX-OS フォワーディング アーキテクチャは、スイッチにおけるすべてのルーティング アップデートの処理および転送情報の入力を担います。

ユニキャスト RIB

Cisco NX-OS の転送アーキテクチャは、次の図に示すように、複数のコンポーネントから構成されています。

図 *3 : CiscoNX-OS* フォワーディング アーキテクチャ



ユニキャスト RIB は、直接接続のルート、スタティック ルート、ダイナミック ユニキャスト ルーティング プロトコルで検出されたルートを含むルーティング テーブルを維持しています。また、アドレス解決プロトコル（ARP）などの送信元から、隣接情報を収集します。ユニキャスト RIB は、ルートに最適なネクストホップを決定し、さらにユニキャスト FIB 分散モジュール（FDM）のサービスを使用して、ユニキャスト転送情報ベース（FIB）にデータを入力します。

各ダイナミック ルーティング プロトコルは、タイム アウトしたあらゆるルートについて、ユニキャスト RIB を更新する必要があります。その後、ユニキャスト RIB はそのルートを削除し、最適なネクスト ホップを再計算します（代わりに使用できるパスがある場合）。

隣接マネージャ

隣接マネージャは、ARP、Open Shortest Path First version 2（OSPFv2）、ネイバー探索プロトコル（NDP）、静的な設定を含む、異なるプロトコルの隣接情報を維持しています。最も基本的な隣接情報は、これらのプロトコルで探索されたレイヤ 3 からレイヤ 2 へのアドレス マッピングです。発信レイヤ 2 パケットは、隣接情報を使用して、レイヤ 2 ヘッダーの作成を終了します。

隣接マネージャは、ARP 要求による、レイヤ 3 からレイヤ 2 への特定のマッピングの探索をトリガーできます。新しいマッピングは、対応する ARP 返信を受信し、処理すると、使用できるようになります。

隣接テーブル

Cisco Nexus 3548 スイッチでは、隣接関係テーブルはレイヤ 2 MAC 転送テーブルと共有されます。

たとえば、隣接エントリの最大数は、レイヤ 2 MAC エントリの最大数と同じで、64,000 エントリです。

MAC または隣接関係テーブルは、ハッシュ テーブルとして実装されます。

ユニキャスト転送分散モジュール

ユニキャスト転送分散モジュールは、ユニキャストRIBおよびその他の送信元からの転送パス情報を配布します。ユニキャストRIBは、ユニキャストFIBがハードウェア転送テーブルにプログラムする転送情報を生成します。また、ユニキャスト転送分散モジュールは、新規挿入されたモジュールへの FIB 情報のダウンロードも行います。

ユニキャスト転送分散モジュールは、隣接情報を収集し、ユニキャストFIBでのルートの更新時に、この情報およびその他のプラットフォーム依存の情報を書き直し（リライトし）ます。隣接情報およびリライト情報には、インターフェイス、ネクストホップ、およびレイヤ3からレイヤ 2 へのマッピング情報が含まれています。インターフェイスとネクストホップの情報は、ユニキャストRIBからのルート更新情報で受信します。レイヤ3からレイヤ2へのマッピングは、隣接マネージャから受信します。

FIB

ユニキャストFIBは、ハードウェア転送エンジンに使用される情報を作成します。ユニキャストFIBは、ユニキャスト転送分散モジュールからルート更新情報を受信し、ハードウェア転送エンジンにプログラミングされるよう、この情報を送信します。ユニキャスト FIB は、ルート、パス、隣接関係の追加、削除、変更を管理します。

ユニキャストFIBは、VRF ごとおよびアドレス ファミリごとに維持されます。ルート更新メッセージに基づいて、ユニキャストFIBは、VRF ごとのプレフィックスとネクストホップ隣接情報データベースを維持します。ネクストホップ隣接データ構造には、ネクストホップの IP アドレスとレイヤ2リライト情報が含まれます。同じネクストホップ隣接情報構造を複数のプレフィックスで使用できます。

またユニキャスト FIB は、インターフェイスごとのユニキャスト リバース パス転送（RPF）チェックをイネーブルまたはディセーブルにします。Cisco Nexus 3548 プラットフォーム スイッチは、各入力側インターフェイスに設定される、次の 2 つの RPF モードをサポートします。

- RPF Strict チェック：ルータ転送テーブルで検証可能な送信元アドレスを持たないパケット、または送信元へのリターン パスに到着しないパケットはドロップされます。
- RPF Looseチェック：パケットはルータ転送テーブルで検証可能な送信元アドレスを持ち、送信元は物理インターフェイスを通じて到達可能です。パケットを受信する入力側インターフェイスは、FIB 内のインターフェイスに一致する必要はありません。

ハードウェア フォワーディング

Cisco NX-OSは、分散パケット転送をサポートしています。入力ポートは、パケット ヘッダーから該当する情報を取得し、その情報をローカル スイッチング エンジンに渡します。ローカル スイッチング エンジンはレイヤ 3 ルックアップを行い、この情報を使って、パケット ヘッ

ダーをリライトします。入力モジュールは、パケットを出力ポートに転送します。出力ポートが別のモジュール上にある場合は、スイッチ ファブリックを使って、パケットが出力モジュールに転送されます。出力モジュールは、レイヤ 3 転送決定には関与しません。

show platform fib または **show platform forwarding** コマンドを使用して、ハードウェア転送の詳細を表示することもできます。

ソフトウェア転送

Cisco NX-OS のソフトウェア転送パスは、主に、ハードウェアでサポートされない機能、またはハードウェア処理中に発生したエラーへの対処に使用されます。通常、IP オプション付きのパケットまたはフラグメンテーションの必要なパケットは CPU に渡されます。ユニキャスト RIB および隣接マネージャは、ソフトウェアでスイッチされるかまたは終了されるパケットに基づいて転送を決定します。

ソフトウェア転送は、コントロール プレーン ポリシーおよびレート リミッタによって管理されます。

レイヤ 3 ユニキャスト ルーティング機能のまとめ

ここでは、Cisco NX-OS でサポートされるレイヤ 3 ユニキャスト機能およびプロトコルを簡単に説明します。

Open Shortest Path First（OSPF）

OSPF プロトコルは、自律システム内のネットワーク到達可能性情報の交換に使用されるリンクステート ルーティング プロトコルです。各 OSPF ルータは、そのアクティブなリンクに関する情報をネイバー ルータにアドバタイズします。リンク情報には、リンク タイプ、リンク メトリック、およびリンクに接続されたネイバー ルータが含まれます。このリンク情報を含むアドバタイズメントは、リンクステートアドバタイズメントと呼ばれます。詳細については、OSPFv2 の設定のセクションを参照してください。

EIGRP

EIGRP プロトコルは、ディスタンス ベクトルとリンクステートの両ルーティング プロトコルの特徴を備えたユニキャスト ルーティング プロトコルです。これは、シスコ専用ルーティング プロトコルである IGRP の改良バージョンです。EIGRP は、典型的なディスタンス ベクトル ルーティング プロトコルのように、ルートを提供するためにネイバー ルータを必要とします。また、リンクステートプロトコルのように、ネイバールータからアドバタイズされたルートからネットワーク トポロジを構築し、この情報を使用して、ループの発生しない、宛先までのパスを選択します。EIGRP の設定方法の詳細については、EIGRP の設定を参照してください。

BGP

BGP は自律システム間ルーティング プロトコルです。BGP ルータは、信頼性の高い転送メカニズムとして伝送制御プロトコル（TCP）を使用し、他の BGP ルータにネットワーク到達可能性情報をアドバタイズします。ネットワーク到達可能性情報には、宛先ネットワーク プレフィックス、宛先に到達するまでに通過する必要のある自律システムのリスト、およびネクストホップ ルータが含まれます。到達可能性情報には、ルートの優先度、ルートの始点、コミュニティなどの詳細なパス属性が含まれます。詳細については、基本的 BGP の設定および高度な BGP の設定のセクションを参照してください。

RIP

RIP は、ホップ数をメトリックとして使用するディスタンス ベクトル プロトコルです。RIP は、世界中のインターネットでトラフィックのルーティングに広く使用されています。また、IGP であるため、単一の自律システム内でルーティングを行います。詳細については、RIP の設定 （227 ページ） を参照してください。

スタティック ルーティング

スタティック ルーティングを使用して、宛先までの一定のルートを入力できます。この機能は、単純なトポロジの小規模ネットワークでは便利です。また、スタティック ルーティングは、他のルーティング プロトコルとともに、デフォルト ルートおよびルート配布の管理に使用されます。詳細については、「スタティック ルーティングの設定」を参照してください。

Route Policy Manager

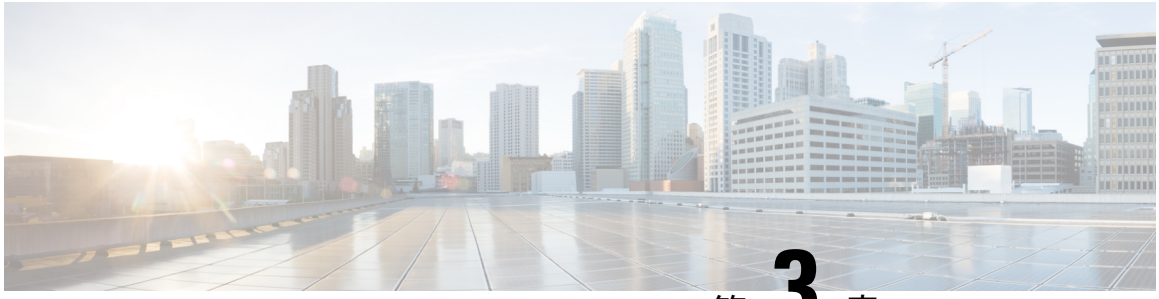
Route Policy Manager は、でルート フィルタリング機能を提供します。Route Policy Manager はルート マップを使用して、さまざまなルーティング プロトコルや、特定のルーティング プロトコル内のさまざまなエンティティ間で配布されたルートをフィルタリングします。フィルタリングは、特定の一致基準に基づいて行われます。これは、アクセス コントロール リストによるパケット フィルタリングに似ています。詳細については、Route Policy Manager の設定のセクションを参照してください。

ファーストホップ冗長プロトコル

ファーストホップ冗長プロトコル（FHRP）は、ホストへの冗長接続を可能にします。アクティブなファーストホップ ルータがダウンした場合は、その機能を引き継ぐスタンバイ ルータが FHRP によって自動的に選択されます。アドレスは仮想のものであり、FHRP グループ内の各ルータ間で共有されているため、ホストを新しい IP アドレスで更新する必要はありません。ホットスタンバイ ルータ プロトコル（HSRP）の詳細については、『Configuring HSRP』を参照してください。仮想ルータ冗長プロトコル（VRRP）の詳細については、VRRP の設定を参照してください。

オブジェクト トラッキング

オブジェクトトラッキングを使用すると、インターフェイス回線プロトコル状態、IPルーティング、ルート到達可能性などの、ネットワーク上の特定のオブジェクトをトラッキングし、トラッキングしたオブジェクトの状態が変化したときに対処することができます。この機能により、ネットワークのアベイラビリティが向上し、オブジェクトがダウンした場合のリカバリ時間が短縮されます。詳細については、 を参照してください。



第 3 章

IPv4 の設定

この章では、Cisco NX-OS スイッチ上でのインターネット プロトコル バージョン 4（IPv4）（アドレス指定を含む）、アドレス解決プロトコル（ARP）および Internet Control Message Protocol（ICMP）の設定方法を説明します。

この章は、次の項で構成されています。

IPv4 の概要

スイッチで IP を設定して、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られます。IP アドレスを割り当てると、インターフェイスがイネーブルになり、そのインターフェイス上のホストと通信できるようになります。

IP アドレスは、スイッチ上でプライマリまたはセカンダリとして設定できます。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。スイッチが生成したパケットは、常にプライマリ IPv4 アドレスを使用するため、インターフェイス上のすべてのネットワーキング スイッチは、同じプライマリ IP アドレスを共有する必要があります。各 IPv4 パケットは、送信元または宛先 IP アドレスからの情報に基づいています。詳細については、複数の IPv4 アドレスのセクションを参照してください。

サブネットを使用して、IP アドレスをマスクできます。マスクは、IP アドレスがどのサブネットに属するかを決定するために使用されます。IP アドレスは、ネットワーク アドレスとホスト アドレスで構成されています。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。サブネット マスクは 32 ビット値で、これにより IP パケットの受信者は、IP アドレスのネットワーク ID 部分とホスト ID 部分を区別できます。

Cisco NX-OS システムの IP 機能には、IPv4 パケットの処理と IPv4 パケットの転送を行う役割があります。これには、IPv4 ユニキャスト ルート検索、リバース パス転送（RPF）チェック、およびソフトウェア アクセス制御リスト（ACL）転送が含まれます。また、IP 機能は、ネットワーク インターフェイス IP アドレス設定、重複アドレス チェック、スタティック ルート、および IP クライアントのパケット送受信インターフェイスも管理します。

複数の IPv4 アドレス

Cisco NX-OS システムは、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。さまざまな状況に備え、いくつでもセカンダリ アドレスを指定できます。最も一般的な状況は次のとおりです。

- 特定のネットワーク インターフェイスのホスト IP アドレスの数が不足している場合。たとえば、サブネットにより、論理サブネットごとに 254 までのホストを使用できるが、物理サブネットの 1 つに 300 のホスト アドレスが必要な場合は、ルータ上またはアクセス サーバ上でセカンダリ IP アドレスを使用して、1 つの物理サブネットで 2 つの論理サブネットを使用できます。
- 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ません。

（注） ネットワーク セグメント上のいずれかのスイッチがセカンダリ IPv4 アドレスを使用している場合は、同じネットワーク インターフェイス上の他のすべてのスイッチも、同じネットワークまたはサブネットからのセカンダリ アドレスを使用する必要があります。ネットワーク セグメント上で、一貫性のない方法でセカンダリ アドレスを使用すると、ただちにルーティング ループが発生する可能性があります。

アドレス解決プロトコル

ネットワーク スイッチおよびレイヤ 3 スイッチは、アドレス解決プロトコル（ARP）を使用して、IP（ネットワーク層）アドレスをメディア アクセス コントロール（MAC）レイヤ アドレスにマップし、IP パケットのネットワーク間の送信を可能にします。スイッチは、別のスイッチにパケットを送信する前に、独自の ARP キャッシュを調べて、宛先スイッチの MAC アドレスおよび対応する IP アドレスがあるかどうかを確認します。エントリがない場合、発信元のスイッチは、ネットワーク上のすべてのスイッチにブロードキャスト メッセージを送信します。

各スイッチは、IP アドレスをそれぞれ自身の IP アドレスと比較します。一致する IP アドレスを持つスイッチだけが、スイッチの MAC アドレスを含むパケットとともにデータを送信したスイッチに返信します。送信元スイッチは、以降の参照用に宛先スイッチの MAC アドレスを自身の ARP テーブルに追加し、データリンク ヘッダーの作成とパケットをカプセル化するトレーラの作成を行った後、データ転送を開始します。次の図は、ARP ブロードキャストと応答プロセスを示しています。

図 *4 : ARP* 処理

宛先スイッチが別のスイッチの背後のリモートネットワークにある場合、データを送信するスイッチがデフォルト ゲートウェイの MAC アドレスに対する ARP 要求を送信する場合を除いてプロセスは同じです。アドレスが解決され、デフォルト ゲートウェイがパケットを受信した後に、デフォルト ゲートウェイは、接続されているネットワーク上で宛先の IP アドレスをブロードキャストします。宛先スイッチのネットワーク上のスイッチは、ARP を使用して宛先スイッチの MAC アドレスを取得し、パケットを配信します。ARP はデフォルトでイネーブルにされています。

デフォルトのシステム定義 CoPP ポリシーは、ARP ブロード キャスト パケットのレート制限を行います。デフォルトのシステム定義 CoPP ポリシーは、ARP ブロードキャスト ストームによるコントロール プレーン トラフィックへの影響を防止し、ブリッジド パケットに影響しません。

ARP キャッシング

ARP キャッシングにより、ブロードキャストが最小になり、ネットワーク リソースの浪費が抑制されます。IP アドレスの MAC アドレスへのマッピングは、インターネットワークを送信される各パケットに対しネットワーク上のホップ（スイッチ）ごとに発生します。そのため、ネットワーク パフォーマンスに影響を与えます。

ARP キャッシングでは、ネットワーク アドレスとそれに関連付けられたデータリンク アドレスが一定の期間、メモリに格納されるため、パケットが送信されるたびに同じアドレスを求めてブロードキャストする場合の、貴重なネットワーク リソースの使用が最小限となります。キャッシュ エントリは、定期的に失効するよう設定されているため、保守が必要です。これは、古い情報が無効となる場合があるためです。ネットワーク上のすべてのスイッチは、アドレスがブロードキャストされるとそれぞれのテーブルを更新します。

ARP キャッシュのスタティックおよびダイナミック エントリ

スタティック ルートの使用時には、各スイッチの各インターフェイスの IP アドレス、サブネットマスク、ゲートウェイ、および対応する MAC アドレスを手動で設定する必要があります。

スタティック ルーティングを使用すると、管理を強化できますが、より多くのルート テーブル保守作業が必要となります。ルートを追加または変更するたびに、テーブルの更新が必要となるためです。

ダイナミック ルーティングは、ネットワーク内のスイッチが相互にルーティング テーブルの情報を交換できるプロトコルを使用します。ダイナミック ルーティングは、キャッシュに制限時間を追加しない限り、ルート テーブルが自動更新されるため、スタティック ルーティングより効率的です。デフォルトの制限時間は 25 分ですが、キャッシュから追加および削除されるルートがネットワークに数多く存在する場合は、制限時間を変更します。

ARP を使用しないデバイス

ネットワークが2つのセグメントに分割されると、ブリッジによりセグメントが結合され、各セグメントへのトラフィックが MAC アドレスに基づいてフィルタリングされます。スイッチとは対照的にMACアドレスだけを使用するブリッジは、独自のアドレス テーブルを作成します。スイッチの場合には、IP アドレスおよび対応する MAC アドレスを含む ARP キャッシュがあります。

パッシブ ハブは、ネットワーク内の他のスイッチを物理的に接続する中央接続スイッチです。これは、そのすべてのポートからスイッチに対してメッセージを送信し、レイヤ1で動作しますが、アドレス テーブルは維持しません。

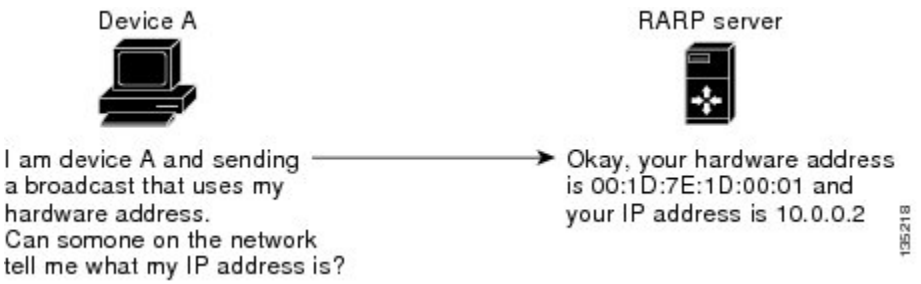
レイヤ2スイッチは、すべてのポートからメッセージを送信するハブとは異なり、メッセージの宛先であるデバイスに接続されるポートを決定し、そのポートにだけ送信します。ただし、レイヤ 3 スイッチは、ARP キャッシュ（テーブル）を作成するスイッチです。

Reverse ARP

RFC 903 で定義された Reverse ARP（RARP）は、ARP と同じように動作しますが、RARP 要求パケットは MAC アドレスではなく IP アドレスを要求する点が異なります。RARP は多くの場合、ディスクレス ワークステーションで使用されます。これは、このタイプのデバイスには、起動時に使用する IP アドレスを格納する手段がないためです。認識できるアドレスは MAC アドレスだけで、これはハードウェアに焼き付けられているためです。

RARP を使用するには、ルータ インターフェイスとして、同じネットワーク セグメント上に RARP サーバが必要です。次の図に、RARP の仕組みを示します。

図 *5 : Reverse ARP*



RARP には、いくつかの制限があります。これらの制限により、ほとんどの企業では、DHCP を使用してダイナミックに IP アドレスを割り当てています。DHCP は、RARP よりコスト効率が高く、必要な保守作業も少ないためです。最も重要な制限は次のとおりです。

- RARP はハードウェア アドレスを使用するため、多くの物理ネットワークを含む大規模なネットワークの場合は、各セグメント上に、冗長性のための追加サーバを備えた RARP サーバが必要です。各セグメントに 2 台のサーバを保持すると、コストがかかります。
- 各サーバは、ハードウェア アドレスと IP アドレスのスタティック マッピングのテーブルで設定する必要があります。IP アドレスの保守は困難です。
- RARP は、ホストの IP アドレスだけを提供し、サブネット マスクもデフォルト ゲートウェイも提供しません。

『Proxy ARP』

プロキシ ARP によって、あるネットワーク上に物理的に存在するスイッチが、同じスイッチまたはファイアウォールに接続された別の物理ネットワークの論理的な一部であることが可能になります。プロキシ ARP によって、ルータの背後のプライベート ネットワーク上のスイッチをパブリック IP アドレスを使用して隠すことができ、さらに、ルータの手前のパブリック ネットワークにあるように見せることができます。ルータはそのアイデンティティを隠すことにより、実際の宛先までパケットをルーティングする役割を担います。プロキシ ARP を使用すると、サブネット上のスイッチは、ルーティングもデフォルト ゲートウェイも設定せずにリモート サブネットまで到達できます。

スイッチが同じデータリンク層ネットワークには存在しないが、同じ IP ネットワークに存在する場合、それらのスイッチはローカル ネットワーク上に存在するものとして、相互にデータ送信を試みます。ただし、これらのスイッチを隔てるルータは、ブロードキャスト メッセージを送信しません。これは、ルータがハードウェア レイヤのブロードキャストを渡さず、アドレスが解決されないためです。

スイッチでプロキシ ARP をイネーブルにし、ARP 要求を受信すると、プロキシ ARP はこれを、ローカル LAN 上にないシステムに対する要求と見なします。スイッチは、ブロードキャストがアドレス指定されたリモートの宛先であるかのように、そのスイッチの MAC アドレスをリモートの宛先の IP アドレスと関連付ける ARP 応答で応答します。ローカル スイッチは、宛先に直接接続されていると確信しますが、実際には、パケットはローカル スイッチによってローカル サブネットワークから宛先サブネットワークへ転送されます。デフォルトでは、プロキシ ARP はディセーブルになっています。

ローカル プロキシ ARP

ローカル Proxy ARP を使用すると、通常ルーティングが必要ないサブネット内の IP アドレスを求める ARP 要求に対し、スイッチが応答するようにできます。ローカル プロキシ ARP をイネーブルにすると、ARP は、サブネット内の IP アドレスを求めるすべての ARP 要求に応答し、サブネット内のホスト間ですべてのトラフィックを転送します。この機能は、接続先ス

イッチ上での設定により、意図的にホスト間の直接的なコミュニケーションが禁止されているサブネットについてだけ使用してください。

Gratuitous ARP

Gratuitous ARP は、送信元 IP アドレスと宛先 IP アドレスが同じである要求を送信し、重複する IP アドレスを検出します。Cisco NX-OS は Gratuitous ARP 要求または ARP キャッシュの更新の有効または無効をサポートします。

収集スロットル

着信 IP パケットがラインカードに転送されたときに、ネクスト ホップのアドレス解決プロトコル（ARP）の要求が解決されない場合、ラインカードはパケットをスーパーバイザに転送します（収集スロットル）。スーパーバイザはネクストホップの MAC アドレスを解決し、ハードウェアをプログラミングします。

ARP 要求が送信されると、ソフトウェアは、同じネクストホップ IP アドレスへのパケットがスーパーバイザに転送されないようにするために、ハードウェア内に /32 ドロップ隣接関係を追加します。ARP が解決されると、そのハードウェア エントリは正しい MAC アドレスで更新されます。タイムアウト期間が経過するまでに ARP エントリが解決されない場合、そのエントリはハードウェアから削除されます。

（注） Glean スロットリングは IPv4 および IPv6 でサポートされますが、IPv6 リンクローカル アドレスはサポートされません。

ICMP

ICMP を使用して、IP 処理に関連するエラーおよびその他の情報を報告するメッセージパケットを提供できます。ICMP は、ICMP 宛先到達不能メッセージ、ICMP エコー要求（2 つのホスト間でパケットを往復送信する）、およびエコー返信メッセージなどのエラー メッセージを生成します。ICMP は多くの診断機能も備えており、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。デフォルトでは、ICMP がイネーブルにされています。

次に示すのは、ICMP メッセージ タイプの一部です。

- ネットワーク エラー メッセージ
- ネットワーク輻輳メッセージ
- トラブルシューティング情報
- タイムアウト告知

（注） ICMP リダイレクトは、ローカル プロキシ ARP 機能がイネーブルになっているインターフェイスではディセーブルになります。

（注） ワープモードでは、IP リダイレクト、出力ルーテッドアクセス コントロール リスト（RACL）、ポート アクセス コントロール リスト（PACL）、および等コスト マルチパス（ECMP）の機能はサポートされません。

仮想化のサポート

IPv4 は、仮想ルーティングおよび転送（VRF）インスタンスをサポートしています。デフォルトでは、特に別の VRF を設定しない限り、Cisco NX-OS はユーザーをデフォルトの VRF に配置します。

IPv4の前提条件

IPv4 には、次の前提条件があります。

- IPv4 はレイヤ 3 インターフェイス上だけで設定可能です。

IPv4 の注意事項および制約事項

IPv4 設定時の注意事項および制約事項は、次のとおりです。

- セカンダリ IP アドレスは、プライマリ IP アドレスの設定後にだけ設定できます。
- Cisco Nexus 3548 スイッチは、VLAN 単位の CAM エージング タイマーをサポートしていません。

IPv4 のデフォルト設定

次の表に、IP パラメータのデフォルト設定値を示します。

表 *3:* デフォルト *IP* パラメータ

パラメータ	デフォルト
ARP タイムアウト	1500 秒
プロキシ ARP	無効

IPv4 の設定

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

IPv4 アドレス指定の設定

ネットワーク インターフェイスにプライマリ IP アドレスを割り当てることができます。

手順の概要

1. **configure terminal**
2. **interface ethernet number**
3. **no switchport**
4. **ip address ip-address/length** [**secondary**]
5. （任意） **show ip interface**
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface ethernet number**
	例：
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		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	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッドインターフェイスとして設定します。
ステップ 4	**ip address ip-address/length** [**secondary**] 例： `switch(config-if)# ip address 192.2.1.1 255.0.0.0`	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。 • 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 5	（任意） **show ip interface** 例： `switch(config-if)# show ip interface`	IPv4 用に設定されたインターフェイスを表示します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

次に、IPv4 アドレスを割り当てる例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 2/3
switch(config-if)# no switchport
switch(config-if)# ip address 192.2.1.1 255.0.0.0
switch(config-if)# copy running-config startup-config
```

複数の IP アドレスの設定

セカンダリ IP アドレスは、プライマリ IP アドレスの設定後にのみ追加できます。

手順の概要

1. **configure terminal**
2. **interface ethernet number**
3. **no switchport**
4. **ip address ip-address/length** [**secondary**]
5. （任意） **show ip interface**
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface ethernet number**
	例：
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		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	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。
ステップ 4	**ip address ip-address/length** [**secondary**] 例： `switch(config-if)# ip address 192.2.1.1 255.0.0.0 secondary`	設定したアドレスをセカンダリ IPv4 アドレスとして指定します。
ステップ 5	（任意） **show ip interface** 例： `switch(config-if)# show ip interface`	IPv4 用に設定されたインターフェイスを表示します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

スタティック ARP エントリの設定

スイッチ上に、IP アドレスを MAC ハードウェア アドレス（スタティック マルチキャスト MAC アドレスを含む）にマップするスタティック ARP エントリを設定できます。

手順の概要

1. **configure terminal**
2. **interface ethernet number**
3. **no switchport**
4. **ip arp ipaddr mac_addr**
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface ethernet number**
	例：
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		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	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。
ステップ 4	**ip arp ipaddr mac_addr** 例： `switch(config-if)# ip arp 192.2.1.1 0019.076c.1a78`	IP アドレスを MAC アドレスにスタティック エントリとして関連付けます。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

次に、スタティック ARP エントリを設定する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 2/3
switch(config-if)# no switchport
switch(config-if)# ip arp 1 92.2.1.1 0019.076c.1a78
switch(config-if)# copy running-config startup-config
```

プロキシ ARP の設定

スイッチで、別のネットワークまたはサブネット上のホストのメディアアドレス定義する Proxy ARP を設定できます。

手順の概要

1. **configure terminal**
2. **interface ethernet number**
3. **no switchport**
4. **ip proxy-arp**
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface ethernet number**
	例：
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		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	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。
ステップ 4	**ip proxy-arp** 例： `switch(config-if)# ip 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 5	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

次に、プロキシ ARP を設定する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 2/3
switch(config-if)# no switchport
```

```
switch(config-if)# ip proxy-arp
switch(config-if)# copy running-config startup-config
```

ローカル プロキシ ARP の設定

スイッチ上でローカル プロキシ ARP を設定できます。

手順の概要

1. **configure terminal**
2. **interface ethernet number**
3. **no switchport**
4. **ip local-proxy-arp**
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface ethernet number**
	例：
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		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	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。
ステップ 4	**ip local-proxy-arp** 例： `switch(config-if)# ip local-proxy-arp`	インターフェイス上でローカル プロキシ ARP をイネーブルにします。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

次に、ローカル プロキシ ARP を設定する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 2/3
switch(config-if)# no switchport
switch(config-if)# ip local-proxy-arp
switch(config-if)# copy running-config startup-config
```

無償 ARP の設定

インターフェイス上で Gratuitous ARP を設定できます。

手順の概要

1. **configure terminal**
2. **interface ethernet number**
3. **no switchport**
4. **ip arp gratuitous** { **request** | **update** }
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface ethernet number**
	例：
	`switch(config)# interface ethernet 2/3`
	`switch(config-if)#`
		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	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。
ステップ 4	**ip arp gratuitous** { **request**	**update** } 例： `switch(config-if)# ip arp gratuitous request`	インターフェイス上で無償 ARP をイネーブルにします。デフォルトは有効です。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

次に、Gratuitous ARP 要求をディセーブルにする例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 2/3
switch(config-if)# no switchport
switch(config-if)# no ip arp gratuitous request
switch(config-if)# copy running-config startup-config
```

IP ダイレクト ブロードキャストの設定

IP ダイレクト ブロードキャストは、宛先アドレスが何らかの IP サブネットの有効なブロードキャストアドレスであるにもかかわらず、その宛先サブネットに含まれないノードから発信される IP パケットです。

宛先サブネットに直接接続されていないスイッチは、ユニキャスト IP パケットをそのサブネット上のホストに転送するのと同じ方法で、IP ダイレクト ブロードキャストを転送します。ダイレクト ブロードキャスト パケットが、宛先サブネットに直接接続されたスイッチに到着すると、宛先サブネット上のブロードキャストとして「展開」されます。パケットの IP ヘッダー内の宛先アドレスはそのサブネットに設定された IP ブロードキャスト アドレスに書き換えられ、パケットはリンク層ブロードキャストとして送信されます。

あるインターフェイスでダイレクトブロードキャストがイネーブルになっている場合、着信した IP パケットが、そのアドレスに基づいて、そのインターフェイスが接続されているサブネットを対象とするダイレクト ブロードキャストとして識別されると、そのパケットはそのサブネット上にブロードキャストとして展開されます。

IP ダイレクトブロードキャス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 モードで次のコマンドを使用します。

コマンド	目的
ip directed-broadcast	ダイレクトブロードキャストの物理ブロードキャストへの変換をイネーブルにします。

IP 収集スロットルの設定

IP 収集スロットルを設定して、到達できないかまたは存在しないネクスト ホップの ARP 解決のためにスーパーバイザに送信される不要な収集パケットをフィルタリングすることを推奨します。IP 収集スロットルは、ソフトウェアのパフォーマンスを向上させ、トラフィックをより効率的に管理します。

（注） Glean スロットリングは IPv4 および IPv6 でサポートされますが、IPv6 リンクローカル アドレスはサポートされません。

手順の概要

1. **configure terminal**
2. [**no**] **hardware ip glean throttle**
3. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	[**no**] **hardware ip glean throttle** 例： `switch(config) # hardware ip glean throttle`	IP 収集スロットルをイネーブルにします。
ステップ 3	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

ハードウェア IP 収集スロットルの最大値の設定

転送情報ベース（FIB）にインストールされている隣接関係の最大ドロップ数を制限できます。

手順の概要

1. **configure terminal**
2. **hardware ip glean throttle maximum count**
3. **no hardware ip glean throttle maximum count**
4. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**hardware ip glean throttle maximum count** 例：	FIB にインストールされるドロップ隣接関係の数を設定します。

	コマンドまたはアクション	目的
	`switch(config)# hardware ip glean throttle maximum 2134`	
ステップ 3	**no hardware ip glean throttle maximum count** 例： `switch(config)# no hardware ip glean throttle maximum 2134`	デフォルトの制限値を適用します。 デフォルト値は 1000 です。範囲は 0 ～ 16,383 エントリです。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、FIB にインストールされている隣接関係の最大ドロップ数を制限する例を示します。

```
switch# configure terminal
switch(config)# hardware ip glean throttle maximum 2134
switch(config-if)# copy running-config startup-config
```

ハードウェア IP 収集スロットルのタイムアウトの設定

インストールされたドロップ隣接関係が FIB 内に残る時間のタイムアウトを設定できます。

手順の概要

1. **configure terminal**
2. **hardware ip glean throttle maximum timeout timeout-in-sec**
3. **no hardware ip glean throttle maximum timeout timeout-in-sec**
4. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**hardware ip glean throttle maximum timeout timeout-in-sec** 例： `switch(config)# hardware ip glean throttle maximum timeout 300`	インストールされたドロップ隣接関係が FIB 内に残る時間のタイムアウトを設定します。

	コマンドまたはアクション	目的
ステップ 3	**no hardware ip glean throttle maximum timeout timeout-in-sec** 例： `switch(config)# no hardware ip glean throttle maximum timeout 300`	デフォルトの制限値を適用します。 タイムアウト値は秒単位です。範囲は300秒（5分）～ 1800 秒（30 分）です。 （注） タイムアウト期間を超えた後、ドロップ隣接関係は FIB から削除されます。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、インストールされているドロップ隣接関係のタイムアウトを設定する例を示します。

```
switch# configure terminal
switch(config)# hardware ip glean throttle maximum timeout 300
switch(config-if)# copy running-config startup-config
```

ハードウェア IP 収集スロットルの syslog の設定

特定のフローでドロップされたパケットの数が設定されているパケット数を超えた場合は、syslog を生成できます。

手順の概要

1. **configure terminal**
2. **hardware ip glean throttle syslog pck-count**
3. **no hardware ip glean throttle syslog pck-count**
4. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

	コマンドまたはアクション	目的
ステップ 2	**hardware ip glean throttle syslog pck-count** 例： `switch(config)# hardware ip glean throttle syslog 1030`	特定のフローでドロップされたパケットの数が設定されているパケット数を超えた場合に、syslog を生成できます。
ステップ 3	**no hardware ip glean throttle syslog pck-count** 例： `switch(config)# no hardware ip glean throttle syslog 1030`	デフォルトの制限値を適用します。 デフォルトは 10000 パケットです。範囲は 0 ～ 65535 パケットです。 （注） タイムアウト期間を超えた後、ドロップ隣接関係は FIB から削除されます。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、あるフローのドロップされたパケット数が、設定されたパケット数を超えた場合に syslog を生成する例を示します。

```
switch# configure terminal
switch(config)# hardware ip glean throttle syslog 1030
switch(config-if)# copy running-config startup-config
```

IPv4 設定の確認

IPv4 の設定を表示するには、次のいずれかの作業を行います。

コマンド	目的
show hardware forwarding ip verify	IP パケット検証の設定を表示します。
show ip adjacency	隣接関係テーブルを表示します。
show ip arp	ARP テーブルを表示します。
show ip interface	IPに関連するインターフェイス情報を表示します。
show ip arp statistics [**vrf** *vrf-name*]	ARP 統計情報を表示します。
show ip adjacency summary	スロットル隣接関係の数のサマリーを表示します。
show ip arp summary	スロットル隣接関係の数のサマリーを表示します。

コマンド	目的
show ip adjacency throttle statistics	スロットリングされた隣接関係のみを表示します。

IPv4 の設定例

次に、IPv4 アドレスを設定する例を示します。

```
configure terminal
interface ethernet 1/2
no switchport
ip address 192.2.1.1/16
```

その他の参考資料

IP の実装に関する詳細情報については、次の各項を参照してください。

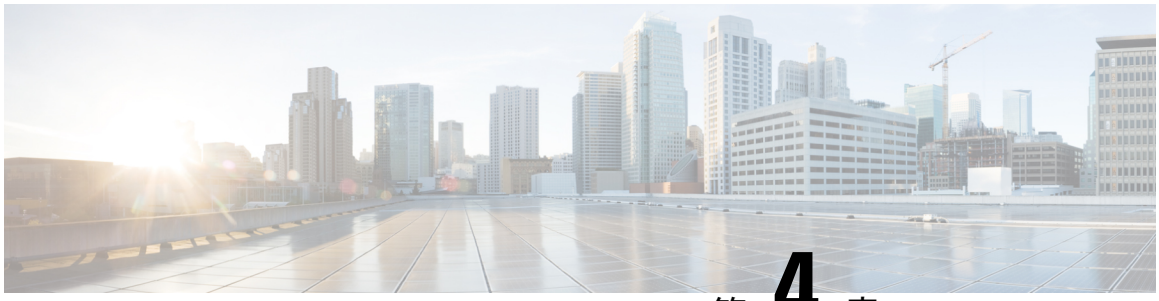
- 関連資料
- 標準

関連資料

関連項目	マニュアル タイトル
IP CLI コマンド	Cisco Nexus 3000 シリーズ ユニキャスト ルーティング コマンド リファレンス

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。	—



第 4 章

OSPFv2 の設定

この章では、Cisco NX-OS スイッチで IPv4 ネットワーク用の Open Shortest Path First version 2（OSPFv2）を設定する方法について説明します。

この章は、次の項で構成されています。

OSPFv2 について

OSPFv2 は、IPv4 ネットワーク用 IETF リンクステート プロトコルです（「リンクステート プロトコル」の項を参照）。OSPFv2 ルータは、hello パケットと呼ばれる特別なメッセージを各 OSPF 対応インターフェイスに送信して、ほかの OSPFv2 隣接ルータを探索します。ネイバー ルータが発見されると、この 2 台のルータは hello パケットの情報を比較して、両者の設定に互換性のあるかどうかを判定します。これらのネイバー ルータは隣接を確立しようとします。つまり、両者のリンクステート データベースを同期させて、確実に同じ OSPFv2 ルーティング情報を持つようにします。隣接ルータは、各リンクの稼働状態に関する情報、リンクのコスト、およびその他のあらゆるネイバー情報を含むリンクステート アドバタイズメント（LSA）を共有します。これらのルータはその後、受信した LSA をすべての OSPF 対応インターフェイスにフラッディングします。これにより、すべての OSPFv2 ルータのリンクステート データベースが最終的に同じになります。すべての OSPFv2 ルータのリンクステート データベースが同じになると、ネットワークは収束します（「コンバージェンス」を参照）。その後、各ルータは、ダイクストラの最短パス優先（SPF）アルゴリズムを使用して、自身のルート テーブルを構築します。

OSPFv2 ネットワークは、複数のエリアに分割できます。ルータは、ほとんどの LSA を 1 つのエリア内だけに送信するため、OSPF 対応ルータの CPU とメモリの要件が緩やかになります。

OSPFv2 は IPv4 をサポートしています。

Hello パケット

OSPFv2 ルータは、すべての OSPF 対応インターフェイスに hello パケットを定期的に送信します。ルータがこの hello パケットを送信する頻度は、インターフェイスごとに設定された hello 間隔により決定されます。OSPFv2 は、hello パケットを使用して、次のタスクを実行します。

- ネイバー探索
- キープアライブ
- 指定ルータの選定（「指定ルータ」セクションを参照してください）

hello パケットには、リンクの OSPFv2 コスト割り当て、hello 間隔、送信元ルータのオプション機能など、送信元の OSPFv2 インターフェイスとルータに関する情報が含まれます。これらの hello パケットを受信する OSPFv2 インターフェイスは、設定に受信インターフェイスの設定との互換性があるかどうかを判定します。互換性のあるインターフェイスはネイバーと見なされ、ネイバー テーブルに追加されます（「Neighbors」の項を参照してください）。

hello パケットには、送信元インターフェイスが通信したルータのルータ ID のリストも含まれます。受信インターフェイスが、このリストで自身の ID を見つけた場合は、2 つのインターフェイス間で双方向通信が確立されます。

OSPFv2 は、hello パケットをキープアライブ メッセージとして使用して、ネイバーが通信を継続中であるかどうかを判定します。ルータが設定されたデッド間隔（通常は hello 間隔の倍数）の間、hello パケットを受信しない場合、そのネイバーはローカル ネイバー テーブルから削除されます。

Neighbors

ネイバーであると見なされるようにするには、リモート インターフェイスと互換性があるように、OSPFv2 インターフェイスを設定しておく必要があります。この 2 つの OSPFv2 インターフェイスで、次の基準が一致している必要があります。

- hello 間隔
- デッド間隔
- エリア ID（「エリア」の項を参照）
- 認証
- オプション機能

一致する場合は、次の情報がネイバー テーブルに入力されます。

- ネイバー ID：ネイバーのルータ ID。

- プライオリティ：ネイバーのプライオリティ。プライオリティは、指定ルータの選定（「指定ルータ」を参照）に使用されます。
- 状態：ネイバーから通信があったか、双方向通信の確立処理中であるか、リンクステート情報を共有しているか、または完全な隣接関係が確立されたかを示します。
- デッド タイム：このネイバーから最後の hello パケットを受信した後に経過した時間を示します。
- IP アドレス：ネイバーの IP アドレス。
- 指定ルータ：ネイバーが指定ルータ、またはバックアップ指定ルータとして宣言されたかどうかを示します（指定ルータを参照）。
- ローカル インターフェイス：このネイバーの hello パケットを受信したローカル インターフェイス。

隣接関係

すべてのネイバーが隣接関係を確立するわけではありません。ネットワーク タイプと確立された指定ルータに応じて、完全な隣接関係を確立して、すべてのネイバーと LSA を共有するものと、そうでないものがあります。詳細については、「指定ルータ」セクションを参照してください。

隣接関係は、OSPF のデータベース説明パケット、リンク状態要求パケット、およびリンク状態更新パケットを使用して確立されます。データベース説明パケットには、ネイバーのリンクステート データベースからの LSA ヘッダーだけが含まれます（リンクステート データベースのセクションを参照）。ローカル ルータは、これらのヘッダーを自身のリンクステート データベースと比較して、新規の LSA か、更新された LSA かを判定します。ローカル ルータは、新規または更新の情報を必要とする各 LSA について、リンク状態要求パケットを送信します。これに対し、ネイバーはリンク状態更新パケットを返信します。このパケット交換は、両方のルータのリンクステート情報が同じになるまで継続します。

指定ルータ

複数のルータを含むネットワークは、OSPF 特有の状況です。すべてのルータがネットワークで LSA をフラッディングした場合は、同じリンクステート情報が複数の送信元から送信されます。ネットワークのタイプによっては、OSPFv2 は指定ルータ（DR）という 1 台のルータを使用して LSA のフラッディングを制御し、OSPFv2 の残りの部分に対してネットワークを代表する役割をさせる場合があります（エリアのセクションを参照）。DR がダウンした場合、OSPFv2 はバックアップ指定ルータ（BDR）を選択します。DR がダウンすると、OSPFv2 はこの BDR を使用します。

ネットワーク タイプは次のとおりです。

- ポイントツーポイント：2 台のルータ間にのみ存在するネットワーク。ポイントツーポイント ネットワーク上の全ネイバーは隣接関係を確立し、DR は存在しません。

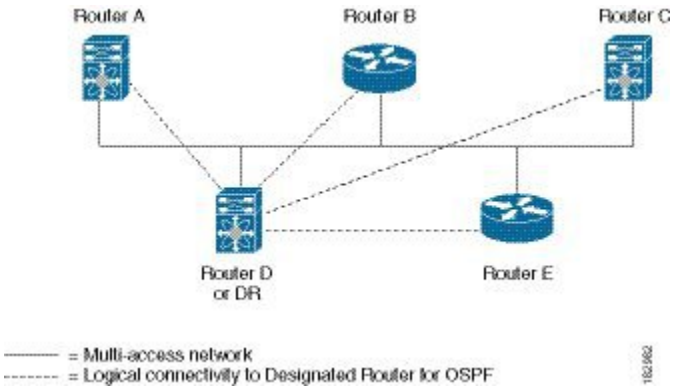
- ブロードキャスト： ブロードキャスト トラフィックが可能なイーサネットなどの共有メディア上で通信できる複数のルータを持つネットワーク。OSPFv2 ルータは DR および BDR を確立し、これらにより、ネットワーク上の LSA フラッディングを制御します。OSPFv2 は、MAC アドレス 0100.5300.0005 を使用して、ネイバーと通信します。

DR と BDR は、hello パケット内の情報に基づいて選択されます。インターフェイスは hello パケットの送信時に、どれが DR および BDR かわかっている場合は、優先フィールドと、DR および BDR フィールドを設定します。ルータは、hello パケットの DR および BDR フィールドで宣言されたルータと優先フィールドに基づいて、選定手順を実行します。最終的に OSPFv2 は、最も大きいルータ ID を DR および BDR として選択します。

他のすべてのルータは、DR との隣接関係を確立します。次の図は、すべてのルータと DR との隣接関係を示しています。

DR は、ルータ インターフェイスに基づいています。1 つのネットワークの DR であるルータは、別のインターフェイス上の他のネットワークの DR となることはできません。

図 *6:* マルチアクセス ネットワークの *DR*



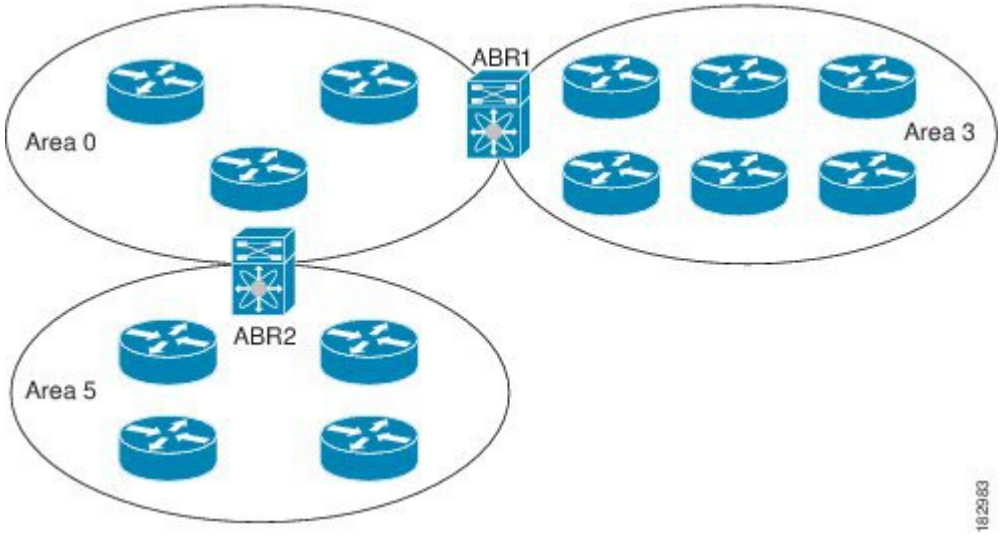
エリア

OSPFv2 ネットワークを複数のエリアに分割すると、ルータに要求される OSPFv2 の CPU とメモリに関する要件を制限できます。エリアとは、ルータの論理的な区分で、OSPFv2 ドメイン内にリンクして別のサブドメインを作成します。LSA フラッディングはエリア内でのみ発生し、リンクステートデータベースはエリア内のリンクにのみ制限されます。定義されたエリア内のインターフェイスには、エリア ID を割り当てることができます。エリア ID は、10.2.3.1 などの、数字またはドット付き 10 進表記で入力できる 32 ビット値です。

Cisco NX-OS はエリアを常にドット付き 10 進表記で表示します。

OSPFv2 ネットワーク内に複数のエリアを定義する場合は、0 という予約されたエリア ID を持つバックボーンエリアも定義する必要があります。エリアが複数ある場合は、1 台以上のルータがエリア境界ルータ（ABR）となります。ABR は、バックボーン エリアと他の 1 つ以上の定義済みエリアの両方に接続します（下図を参照）。

図 *7 : OSPFv2* エリア



ABR には、接続するエリアごとに個別のリンクステート データベースがあります。ABR は、接続したエリアの 1 つからバックボーン エリアにネットワーク集約（タイプ 3）LSA（「ルート集約」セクションを参照）を送信します。バックボーンエリアは、1 つのエリアに関する集約情報を別のエリアに送信します。**OSPFv2 エリア**の図では、エリア 0 が、エリア 5 に関する集約情報をエリア 3 に送信しています。

OSPFv2 では、自律システム境界ルータ（ASBR）という、もう 1 つのルータ タイプも定義されています。このルータは、OSPFv2 エリアを別の自律システムに接続します。自律システムとは、単一の技術的管理エンティティにより制御されるネットワークです。OSPFv2 は、そのルーティング情報を別の自律システムに再配布したり、再配布されたルートを別の自律システムから受信したりできます。詳細については、高度な機能のセクションを参照してください。

リンクステート アドバタイズメント

OSPFv2 はリンクステート アドバタイズメント（LSA）を使用して、固有のルーティング テーブルを構築します。

LSA タイプ

次の表に、Cisco NX-OS でサポートされる LSA タイプを示します。

表 *4: LSA* タイプ

タイプ	名前	説明
1	ルータ LSA	すべてのルータが送信する LSA。この LSA には、すべてのリンクの状態とコスト、およびリンク上のすべての OSPFv2 ネイバーの一覧が含まれます。ルータ LSA は SPF 再計算をトリガーします。ルータ LSA はローカル OSPFv2 エリアにフラッディングされます。
2	ネットワーク LSA	DR が送信する LSA。この LSA には、マルチアクセス ネットワーク内のすべてのルータの一覧が含まれます。ネットワーク LSA は SPF 再計算をトリガーします。「指定ルータ」のセクションを参照してください。
3	ネットワーク集約 LSA	エリア境界ルータが、ローカル エリア内の宛先ごとに外部エリアに送信する LSA。この LSA には、エリア境界ルータからローカルの宛先へのリンク コストが含まれます。「エリア」のセクションを参照してください。
4	ASBR 集約 LSA	エリア境界ルータが外部エリアに送信する LSA。この LSA は、リンク コストを ASBR のみにアドバタイズします。「エリア」の項を参照してください。
5	AS 外部 LSA	ASBR が生成する LSA。この LSA には、外部自律システム宛先へのリンク コストが含まれます。AS 外部 LSA は、自律システム全体にわたってフラッディングされます。「エリア」の項を参照してください。
7	NSSA 外部 LSA	ASBR が Not-So-Stubby Area（NSSA）内で生成する LSA。この LSA には、外部自律システム宛先へのリンク コストが含まれます。NSSA 外部 LSA は、ローカル NSSA 内のみでフラッディングされます。「エリア」のセクションを参照してください。

タイプ	名前	説明
9–11	不透明 LSA	OSPF の拡張に使用される LSA。「不透明 LSA」のセクションを参照してください。

リンク コスト

各 OSPFv2 インターフェイスは、リンク コストを割り当てられています。このコストは任意の数字です。デフォルトでは、Cisco NX-OS が、設定された参照帯域幅をインターフェイス帯域幅で割った値をコストとして割り当てます。デフォルトでは、参照帯域幅は 40 Gbps です。リンク コストは各リンクに対して、LSA 更新情報で伝えられます。

フラッディングと LSA グループ ペーシング

OSPFv2 ルータは LSA を受信すると、その LSA をすべての OSPF 対応インターフェイスに転送し、この情報を使用して OSPFv2 エリアをフラッディングします。この LSA フラッディングにより、ネットワーク内のすべてのルータが同じルーティング情報を持つことが保証されます。LSA フラッディングは、OSPFv2 エリアの設定により異なります（「エリア」を参照）。LSA は、リンクステート リフレッシュ時間に基づいて（デフォルトでは 30 分ごとに）フラッディングされます。各 LSA には、リンクステート リフレッシュ時間が設定されています。

ネットワークの LSA 更新情報のフラッディング レートは、LSA グループ ペーシング機能を使用して制御できます。LSA グループ ペーシングにより、CPU またはバッファの使用率を低下させることができます。この機能により、同様のリンクステート リフレッシュ時間を持つ LSA がグループ化されるため、OSPFv2 で、複数の LSA を 1 つの OSPFv2 更新メッセージにまとめることが可能となります。

デフォルトでは、相互のリンクステート リフレッシュ時間が 4 分以内の LSA が同じグループに入れられます。この値は、大規模なリンクステート データベースでは低く、小規模のデータベースでは高くして、ネットワーク上の OSPFv2 負荷を最適化する必要があります。

リンクステート データベース

各ルータは、OSPFv2 ネットワーク用のリンクステート データベースを保持しています。このデータベースには、収集されたすべての LSA が含まれ、ネットワークを通過するすべてのルートに関する情報が格納されます。OSPFv2 は、この情報を使用して、各宛先への最適パスを計算し、この最適パスをルーティング テーブルに入力します。

MaxAge と呼ばれる設定済みの時間間隔で受信された LSA 更新情報がまったくない場合は、リンクステート データベースから LSA が削除されます。ルータは、LSA を 30 分ごとに繰り返してフラッディングし、正確なリンクステート情報が期限切れで削除されるのを防ぎます。Cisco NX-OS は、すべての LSA が同時にリフレッシュされるのを防ぐために、LSA グループ機能をサポートしています。詳細については、フラッディングと LSA グループ ペーシングのセクションを参照してください。

不透明 LSA

不透明 LSA により、OSPF 機能の拡張が可能となります。不透明 LSA は、標準 LSA ヘッダーと、それに続くアプリケーション固有の情報で構成されます。この情報は、OSPFv2 または他のアプリケーションにより使用される場合があります。次のような 3 種類の不透明 LSA タイプが定義されています。

- LSA タイプ 9：ローカル ネットワークにフラッディングされます。
- LSA タイプ 10：ローカル エリアにフラッディングされます。
- LSA タイプ 11：ローカル自律システムにフラッディングされます。

OSPFv2およびユニキャストRIB

OSPFv2 は、リンクステート データベースでダイクストラの SPF アルゴリズムを実行します。このアルゴリズムにより、パス上の各リンクのリンク コストの合計に基づいて、各宛先への最適なパスが選択されます。そして、選択された各宛先への最短パスがOSPFv2ルートテーブルに入力されます。OSPFv2 ネットワークが収束すると、このルート テーブルはユニキャスト RIB にデータを提供します。OSPFv2 はユニキャスト RIB と通信し、次の動作を行います。

- ルートの追加または削除
- 他のプロトコルからのルートの再配布への対応
- 変更されていない OSPFv2 ルートの削除およびスタブ ルータ アドバタイズメントを行うためのコンバージェンス更新情報の提供（OSPFv2 スタブ ルータ アドバタイズメントのセクションを参照）

さらにOSPFv2は、変更済みダイクストラアルゴリズムを実行して、集約および外部（タイプ 3、4、5、7）LSA の変更の高速再計算を行います。

認証

OSPFv2 メッセージに認証を設定して、ネットワークでの不正な、または無効なルーティング更新を防止できます。Cisco NX-OS は、次の 2 つの認証方式をサポートしています。

- 簡易パスワード認証
- MD5 認証ダイジェスト

OSPFv2 認証は、OSPFv2 エリアに対して、またはインターフェイスごとに設定できます。

簡易パスワード認証

簡易パスワード認証では、OSPFv2 メッセージの一部として送信された単純なクリア テキストのパスワードを使用します。受信 OSPFv2 ルータが OSPFv2 メッセージを有効なルート更新情報として受け入れるには、同じクリア テキスト パスワードで設定されている必要があります。

パスワードがクリアテキストであるため、ネットワーク上のトラフィックをモニタできるあらゆるユーザがパスワードを入手できます。

MD5 認証

OSPFv2 メッセージを認証するには、MD5 認証を使用する必要があります。そのためには、ローカル ルータとすべてのリモート OSPFv2 ネイバーが共有するパスワードを設定します。Cisco NX-OS は各 OSPFv2 メッセージに対して、メッセージと暗号化されたパスワードに基づく MD5 一方向メッセージ ダイジェストを作成します。インターフェイスはこのダイジェストを OSPFv2 メッセージとともに送信します。受信する OSPFv2 ネイバーは、同じ暗号化パスワードを使用して、このダイジェストを確認します。メッセージが変更されていない場合はダイジェストの計算が同一であるため、OSPFv2 メッセージは有効と見なされます。

MD5 認証には、ネットワークでのメッセージの再送を防ぐための、各 OSPFv2 メッセージのシーケンス番号が含まれます。

高度な機能

Cisco NX-OS は、ネットワークでの OSPFv2 の可用性やスケーラビリティを向上させる数多くの高度な OSPFv2 機能をサポートしています。

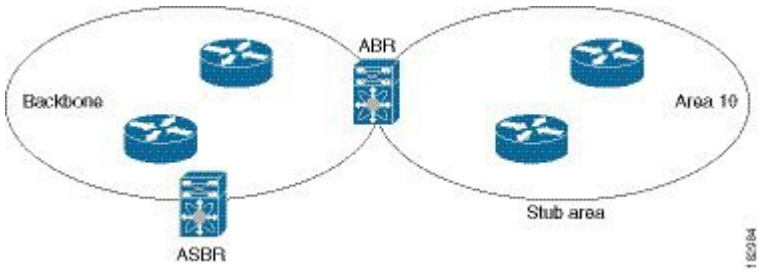
スタブ エリア

エリアをスタブエリアにすると、エリアでフラッディングされる外部ルーティング情報の量を制限できます。スタブ エリアとは、AS 外部（タイプ 5）LSA（リンクステート アドバタイズメントのセクションを参照）が許可されないエリアです。これらの LSA は通常、外部ルーティング情報を伝播するためにローカル自律システム全体でフラッディングされます。スタブエリアには、次の要件があります。

- スタブ エリア内のすべてのルータはスタブ ルータです。「スタブ ルーティング」の項を参照してください。
- スタブ エリアには ASBR ルータは存在しません。
- スタブ エリアには仮想リンクを設定できません。

次の図には、外部 AS に到達するためにエリア 0.0.0.10 内のすべてのルータが ABR を通過する必要のある OSPFv2 AS の例を示します。エリア 0.0.0.10 はスタブ エリアとして設定できます。

図 *8:* スタブ エリア



スタブ エリアは、外部自律システムへのバックボーン エリアを通過する必要のあるすべてのトラフィックにデフォルト ルートを使用します。IPv4 の場合のデフォルト ルートは 0.0.0.0 です。

Not-So-Stubby Area

Not-So-Stubby Area（NSSA）は、スタブ エリアに似ていますが、NSSA では、再配布を使用して NSSA 内で自律システム外部ルートをインポートできる点が異なります。NSSA ASBR はこれらのルートを再配布し、NSSA 外部（タイプ 7）LSA を生成して NSSA 全体でフラッディングします。または、NSSA を他のエリアに接続する ABR を設定することにより、この NSSA 外部 LSA を AS 外部（タイプ 5）LSA に変換することもできます。こうすると、エリア ボーダー ルータ（ABR）は、これらの AS 外部 LSA を OSPFv2 自律システム全体にフラッディングします。変換中は集約とフィルタリングがサポートされます。NSSA 外部 LSA の詳細については、リンクステート アドバタイズメントのセクションを参照してください。

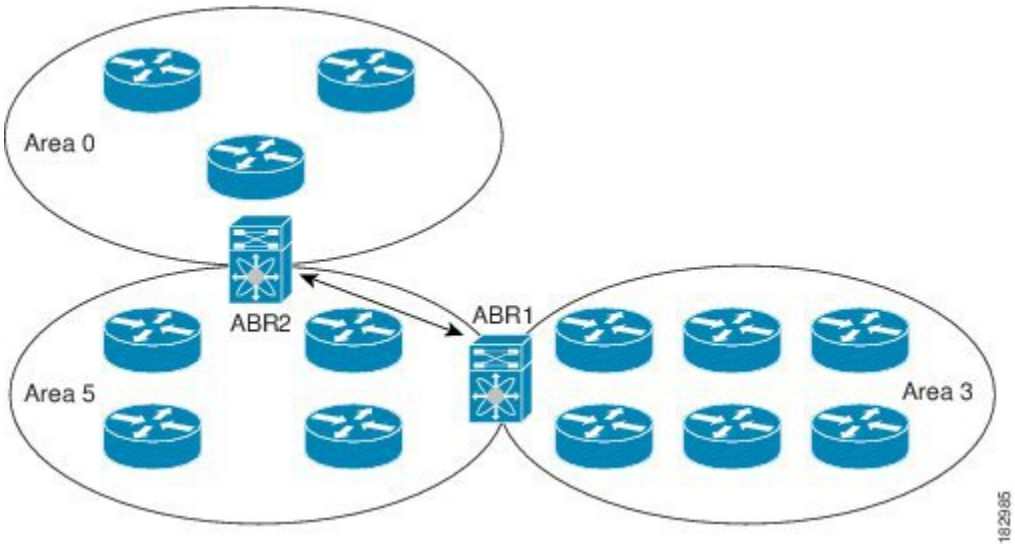
たとえば、OSPFv2 を使用する中央サイトを、異なるルーティング プロトコルを使用するリモート サイトに接続するときに NSSA を使用すると、管理作業を簡素化できます。リモート サイトへのルートはスタブエリア内に再配布できないため、NSSA を使用する前に、企業サイトの境界ルータとリモート ルータの間の接続を OSPFv2 スタブエリアとして実行できません。NSSA を使用すると、企業のルータとリモート ルータ間のエリアを NSSA として定義する（「NSSA の設定」を参照）ことで、OSPFv2 を拡張してリモート接続性をサポートできます。

バックボーン エリア 0 を NSSA にできません。

仮想リンク

仮想リンクを使用すると、物理的に直接接続できない場合に、OSPFv2 エリア ABR をバックボーン エリア ABR に接続できます。次の図には、エリア 3 をエリア 5 経由でバックボーン エリアに接続する仮想リンクを示します。

図 *9*: 仮想リンク



また、仮想リンクを使用して、分割エリアから一時的に回復できます。分割エリアは、エリア内のリンクがダウンしたために隔離された一部のエリアで、ここからはバックボーンエリアへの代表 ABR に到達できません。

ルートの再配布

OSPFv2 は、ルート再配布を使用して、他のルーティング プロトコルからルートを学習できます。ルートの再配布のセクションを参照してください。リンク コストをこれらの再配布されたルートに割り当てるか、またはデフォルト リンク コストを再配布されたすべてのルートに割り当てるように、OSPFv2 を設定します。

ルート再配布では、ルート マップを使用して、再配布する外部ルートを管理します。ルート マップの設定については、Route Policy Manager の設定を参照してください。ルート マップを使用して、これらの外部ルートがローカル OSPFv2 自律システムでアドバタイズされる前に AS 外部（タイプ 5）LSA および NSSA 外部（タイプ 7）LSA のパラメータを変更できます。

ルート集約

OSPFv2 は、学習したすべてのルートを、すべての OSPF 対応ルータと共有するため、ルート集約を使用して、すべての OSPF 対応ルータにフラッディングされる一意のルートの数を削減した方がよい場合があります。ルート集約により、より具体的な複数のアドレスが、すべての具体的なアドレスを表す 1 つのアドレスに置き換えられるため、ルート テーブルが簡素化されます。たとえば、10.1.1.0/24、10.1.2.0/24、および 10.1.3.0/24 というアドレスを 1 つの集約アドレス 10.1.0.0/16 に置き換えることができます。

一般的には、エリア境界ルータ（ABR）の境界ごとに集約します。集約は2つのエリアの間でも設定できますが、バックボーンの方向に集約する方が適切です。こうすると、バックボーンがすべての集約アドレスを受信し、すでに集約されているそれらのアドレスを他のエリアに投入できるためです。集約には、次の 2 タイプがあります。

- エリア間ルート集約
- 外部ルート集約

エリア間ルート集約は ABR 上で設定し、自律システム内のエリア間のルートを集約します。集約の利点を生かすには、これらのアドレスを 1 つの範囲内にまとめることができるように、連続するネットワーク番号をエリア内で割り当てる必要があります。

外部ルート集約は、ルート再配布を使用して OSPFv2 に投入される外部ルートに特有のルート集約です。集約する外部の範囲が連続していることを確認する必要があります。異なる 2 台のルータからの重複範囲を集約すると、誤った宛先にパケットが送信される原因となる場合があります。外部ルート集約は、ルートを OSPF に再配布している ASBR で設定してください。

集約アドレスの設定時に Cisco NX-OS は、ルーティング ブラック ホールおよびルート ループを防ぐために、集約アドレスの廃棄ルートを自動的に設定します。

OSPFv2 スタブ ルータ アドバタイズメント

OSPFv2 スタブ ルータ アドバタイズメント機能を使用して、OSPFv2 インターフェイスをスタブルータとして機能するように構成できます。この機能は、ネットワークに新規ルータを機能制限付きで導入する場合や、過負荷になっているルータの負荷を制限する場合など、このルータ経由の OSPFv2 トラフィックを制限するときに使用します。また、この機能は、さまざまな管理上またはトラフィック エンジニアリング上の理由により使用する場合もあります。

OSPFv2 スタブ ルータ アドバタイズメントは、OSPFv2 ルータをネットワーク トポロジから削除しませんが、他の OSPFv2 ルータがこのルータを使用して、ネットワークの他の部分にトラフィックをルーティングできないようにします。このルータを宛先とするトラフィック、またはこのルータに直接接続されたトラフィックだけが送信されます。

OSPFv2 スタブ ルータ アドバタイズメントは、すべてのスタブ リンク（ローカル ルータに直接接続された）を、ローカル OSPFv2 インターフェイスのコストとしてマークします。すべてのリモート リンクは、最大のコスト（0xFFFF）としてマークされます。

複数の OSPFv2 インスタンス

Cisco NX-OS は、同じノード上で動作する、OSPFv2 プロトコルの複数インスタンスをサポートしています。同一インターフェイスには複数のインスタンスを設定できません。デフォルトでは、すべてのインスタンスが同じシステム ルータ ID を使用します。複数のインスタンスが同じ OSPFv2 自律システムにある場合は、各インスタンスのルータ ID を手動で設定する必要があります。

SPF 最適化

Cisco NX-OS は、次の方法で SPF アルゴリズムを最適化します。

- ネットワーク（タイプ 2）LSA、ネットワーク集約（タイプ 3）LSA、および AS 外部（タイプ 5）LSA 用の部分的 SPF：これらの LSA のいずれかが変更されると、Cisco NX-OS は、全体的な SPF 計算ではなく、高速部分計算を実行します。
- SPF タイマー：さまざまなタイマーを設定して、SPF 計算を制御できます。これらのタイマーには、後続の SPF 計算の幾何バックオフが含まれます。幾何バックオフにより、複数の SPF 計算による CPU 負荷が制限されます。

仮想化のサポート

OSPFv2は、仮想ルーティングおよび転送（VRF）インスタンスをサポートしています。デフォルトでは、特に別の VRF を設定しない限り、Cisco NX-OS はユーザーをデフォルトの VRF に配置します。各 OSPFv2 インスタンスは、システム制限値の範囲で複数の VRF をサポートできます。

OSPFv2 の前提条件

OSPFv2 には、次の前提条件があります。

- OSPF を設定するための、ルーティングの基礎に関する詳しい知識がある。
- スイッチにログインしている。
- リモート OSPFv2 ネイバーと通信可能な IPv4 用インターフェイスが 1 つ以上設定されている。
- LAN Base Services ライセンスがインストールされている。
- OSPFv2 ネットワーク戦略と、ネットワークのプランニングが完成している。たとえば、複数のエリアが必要かどうかを決定します。
- OSPF 機能を有効にしてある（OSPFv2 機能のイネーブル化のセクションを参照）。

OSPFv2 の注意事項および制約事項

OSPFv2 設定時の注意事項および制約事項は、次のとおりです。

- Cisco NX-OS は、ユーザがエリアを 10 進表記で入力するか、ドット付き 10 進表記で入力するかに関係なく、ドット付き 10 進表記でエリアを表示します。

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

OSPFv2のデフォルト設定

次の表に、OSPFv2 パラメータのデフォルト設定値を示します。

表 *5 : OSPFv2* のデフォルト パラメータ

パラメータ	デフォルト
hello 間隔	10 秒
デッド間隔	40 秒
OSPFv2 機能	ディセーブル
スタブ ルータ アドバタイズメントの宣言期間	600 秒
リンク コスト計算の参照帯域幅	40 Gbps
LSA 最小到着時間	1000 ミリ秒

パラメータ	デフォルト
LSA グループ ペーシング	240 秒
SPF 計算初期遅延時間	200 ミリ秒
SPF 計算最小ホールド タイム	1000 ミリ秒
SPF 計算の最大待機時間	5000 ミリ秒

基本的な OSPFv2 の設定

OSPFv2 は、OSPFv2 ネットワークを設計した後に設定します。

OSPFv2 機能のイネーブル化

OSPFv2 を設定するには、その前に OSPFv2 機能を有効にする必要があります。

手順の概要

1. **configure terminal**
2. [**no**] **feature ospf**
3. （任意） **show feature**
4. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	[**no**] **feature ospf** 例： `switch(config)# feature ospf`	OSPFv2 機能を有効にします。 （注） OSPFv2 機能をディセーブルにし、関連付けられた設定をすべて削除するには、このコマンドの **no** バージョンを使用します。
ステップ 3	（任意） **show feature** 例： `switch(config)# show feature`	有効および無効にされた機能を表示します。

	コマンドまたはアクション	目的
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

OSPFv2インスタンスの作成

OSPFv2 を設定する最初のステップは、OSPFv2 インスタンスを作成することです。作成した OSPFv2インスタンスには、一意のインスタンス タグを割り当てます。インスタンス タグは任意の文字列です。

OSPFv2 インスタンス パラメータの詳細については、高度なOSPFv2の設定のセクションを参照してください。

始める前に

OSPF 機能を有効にしてあることを確認します（OSPFv2 機能のイネーブル化のセクションを参照）。

show ip ospf *instance-tag* コマンドを使用して、インスタンス タグが使用されていないことを確認します。

OSPFv2 がルータ ID（設定済みのループバック アドレスなど）を入手可能であるか、またはルータ ID オプションを設定する必要があります。

手順の概要

1. **configure terminal**
2. **router ospf instance-tag**
3. （任意） **router-id** *ip-address*
4. （任意） **show ip ospf** *instance-tag*
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router ospf instance-tag**
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

	コマンドまたはアクション	目的
ステップ 3	（任意） **router-id** *ip-address* 例： `switch(config-router)# router-id 192.0.2.1`	OSPFv2 ルータ ID を設定します。この IP アドレスにより、このOSPFv2インスタンスが識別されます。このアドレスは、システムの設定済みインターフェイス上に存在する必要があります。
ステップ 4	（任意） **show ip ospf** *instance-tag* 例： `switch(config-router)# show ip ospf 201`	OSPF 情報を表示します。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

OSPFv2 インスタンスと、関連付けられた設定をすべて削除するには、**no router ospf** コマンドを使用します。

コマンド	目的
no router ospf *instance-tag* 例： `switch(config)# no router ospf 201`	OSPF インスタンスと、関連付けられた設定を削除します。

（注） このコマンドは、インターフェイス モードでは OSPF 設定を削除しません。インターフェイス モードで設定された OSPFv2 コマンドはいずれも、手動で削除する必要があります。

OSPFv2 インスタンスのオプション パラメータの設定

OSPF のオプション パラメータを設定できます。

OSPFv2 インスタンス パラメータの詳細については、高度なOSPFv2の設定のセクションを参照してください。

始める前に

OSPF 機能がイネーブルになっていることを確認します。OSPFv2 機能のイネーブル化のセクションを参照してください。

OSPFv2 がルータ ID（設定済みのループバック アドレスなど）を入手可能であるか、またはルータ ID オプションを設定する必要があります。

手順の概要

1. **distance** *number*
2. **log-adjacency-changes** [**detail**]
3. **maximum-paths** *path-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	**distance** *number* 例： `switch(config-router)# distance 25`	この OSPFv2 インスタンスのアドミニストレーティブ ディスタンスを設定します。範囲は 1 ～ 255 です。デフォルトは 110 です。
ステップ 2	**log-adjacency-changes** [**detail**] 例： `switch(config-router)# log-adjacency-changes`	ネイバーの状態が変化するたびに、システム メッセージを生成します。
ステップ 3	**maximum-paths** *path-number* 例： `switch(config-router)# maximum-paths 4`	ルート テーブル内の宛先への同じ OSPFv2 パスの最大数を設定します。このコマンドはロードバランシングに使用されます。指定できる範囲は 1 ～ 32 です。デフォルト値は 8 です。

例

次の例は、OSPFv2 インスタンスを作成する方法を示しています。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# copy running-config startup-config
```

OSPFv2でのネットワークの設定

ルータがこのネットワークへの接続に使用するインターフェイスを介して、OSPFv2 へのネットワークを関連付けることで、このネットワークを設定できます（Neighborsのセクションを参照）。すべてのネットワークをデフォルトバックボーンエリア（エリア0）に追加したり、任意の 10 進数または IP アドレスを使用して新規エリアを作成したりできます。

（注） すべてのエリアは、バックボーンエリアに直接、または仮想リンク経由で接続する必要があります。

（注） インターフェイスに有効な IP アドレスを設定するまでは、OSPF はインターフェイス上でイネーブルにされません。

始める前に

OSPF 機能を有効にしてあることを確認します（OSPFv2 機能のイネーブル化のセクションを参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **no switchport**
4. **ip address** *ip-prefix/length*
5. **ip router ospf** *instance-tag* **area** *area-id* [**secondaries none**]
6. （任意） **show ip ospf** *instance-tag* **interface** *interface-type slot/port*
7. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

ステップ 3	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッドインターフェイスとして設定します。
ステップ 4	**ip address** *ip-prefix/length* 例： `switch(config-if)# ip address 192.0.2.1/16`	このインターフェイスに IP アドレスおよびサブネット マスクを割り当てます。
ステップ 5	**ip router ospf** *instance-tag* **area** *area-id* [**secondaries none**] 例： `switch(config-if)# ip router ospf 201 area 0.0.0.15`	OSPFv2 インスタンスおよびエリアにインターフェイスを追加します。

	コマンドまたはアクション	目的
ステップ 6	（任意） **show ip ospf** *instance-tag* **interface** *interface-type slot/port* 例： `switch(config-if)# show ip ospf 201 interface ethernet 1/2`	OSPF 情報を表示します。
ステップ 7	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

インターフェイス コンフィギュレーション モードで、省略可能な次の OSPFv2 パラメータを設定できます。

コマンド	目的
ip ospf cost *number* 例： `switch(config-if)# ip ospf cost 25`	このインターフェイスの OSPFv2 コスト メトリックを設定します。デフォルトでは、参照帯域幅とインターフェイス帯域幅に基づいて、コスト メトリックが計算されます。有効な範囲は 1 ～ 65535 です。
ip ospf dead-interval *seconds* 例： `switch(config-if)# ip ospf dead-interval 50`	OSPFv2 デッド間隔を秒単位で設定します。有効な範囲は 1 ～ 65535 です。デフォルトでは、hello 間隔の秒数の 4 倍です。
ip ospf hello-interval *seconds* 例： `switch(config-if)# ip ospf hello-interval 25`	OSPFv2 hello 間隔を秒単位で設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 10 秒です。
ip ospf mtu-ignore 例： `switch(config-if)# ip ospf mtu-ignore`	OSPFv2 で、ネイバーとのあらゆる IP MTU 不一致が無視されるように設定します。デフォルトでは、ネイバー MTU がローカル インターフェイス MTU が不一致の場合には、隣接関係が確立されません。
ip ospf passive-interface 例： `switch(config-if)# ip ospf passive-interface`	インターフェイス上でルーティングが更新されないようにします。

コマンド	目的
ip ospf priority *number* 例： `switch(config-if)# ip ospf priority 25`	エリアの DR の決定に使用される OSPFv2 プライオリティを設定します。有効な範囲は 0 ～ 255 です。デフォルトは 1 です。「指定ルータ」の項を参照してください。
ip ospf shutdown 例： `switch(config-if)# ip ospf shutdown`	このインターフェイス上の OSPFv2 インスタンスをシャットダウンします。

次に、OSPFv2 インスタンス 201 にネットワーク エリア 0.0.0.10 を追加する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip address 192.0.2.1/16
switch(config-if)# ip router ospf 201 area 0.0.0.10
switch(config-if)# copy running-config startup-config
```

show ip ospf interface コマンドを使用してインターフェイス設定を確認します。**show ip ospf neighbor** コマンドを使用してこのインターフェイスのネイバーを確認します。

エリアの認証の設定

エリア内のすべてのネットワーク、またはエリア内の個々のインターフェイスの認証を設定できます。インターフェイス認証設定を使用すると、エリア認証は無効になります。

始める前に

OSPF 機能を有効にしてあることを確認します（OSPFv2 機能のイネーブル化のセクションを参照）。

インターフェイス上のすべてのネイバーが、共有認証キーを含め、同じ認証設定を共有することを確認します。

この認証設定のためのキー チェーンを作成します。Cisco Nexus 3548 スイッチ NX-OS セキュリティ構成ガイドを参照してください。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **area** *area-id* **authentication** [**message-digest**]
4. **interface** *interface-type slot/port*
5. **no switchport**
6.
7. （任意） **show ip ospf** *instance-tag* **interface** *interface-type slot/port*

8. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

ステップ 3	**area** *area-id* **authentication** [**message-digest**] 例： `switch(config-router)# area 0.0.0.10 authentication`	エリアの認証モードを設定します。

ステップ 4	**interface** *interface-type slot/port*
	例：
	`switch(config-router)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

ステップ 5	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。
ステップ 6		

オプション	説明
コマンド	説明
ip ospf authentication-key [**0** | **3**] *key* 例： `switch(config-if)# ip ospf authentication-key 0 mypass`	このインターフェイスに簡易パスワード認証を設定します。認証が、キーチェーンにもメッセージ ダイジェストにも設定されていない場合は、このコマンドを使用します。0 の場合は、パスワードをクリア テキストで設定します。3 の場合は、パスワードを 3DES 暗号化として設定します。
ip ospf message-digest-key	このインターフェイスにメッセージ ダイジェスト認証を設

	コマンドまたはアクション		目的
	オプション	説明	
	key-id **md5** [**0** | **3**] *key* 例： `switch(config-if)# ip ospf message-digest-key 21 md5 0 mypass`	定します。認証がメッセージダイジェストに設定されている場合は、このコマンドを使用します。key-id の範囲は 1 ～ 255 です。MD5 オプションが 0 の場合はパスワードがクリア テキストで設定され、3 の場合はパス キーが 3DES 暗号化として設定されます。	
ステップ 7	（任意） **show ip ospf** *instance-tag* **interface** *interface-type slot/port* 例： `switch(config-if)# show ip ospf 201 interface ethernet 1/2`		OSPF 情報を表示します。
ステップ 8	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`		この設定変更を保存します。

インターフェイスの認証の設定

エリア内の個々のインターフェイスに認証を設定できます。インターフェイス認証設定を使用すると、エリア認証は無効になります。

始める前に

OSPF 機能を有効にしてあることを確認します（OSPFv2 機能のイネーブル化のセクションを参照）。

インターフェイス上のすべてのネイバーが、共有認証キーを含め、同じ認証設定を共有することを確認します。

この認証設定のキーチェーンを作成します。Cisco Nexus 3548 スイッチ NX-OS セキュリティ構成ガイドを参照してください。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **no switchport**
4. **ip ospf authentication** [**message-digest**]
5. （任意） **ip ospf authentication key-chain** *key-name*

6. （任意） **ip ospf authentication-key** [**0** | **3** | **7**] *key*
7. （任意） **ip ospf message-digest-key** *key-id* **md5** [**0** | **3** | **7**] *key*
8. （任意） **show ip ospf** *instance-tag* **interface** *interface-type slot/port*
9. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

ステップ 3	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。
ステップ 4	**ip ospf authentication** [**message-digest**] 例： `switch(config-if)# ip ospf authentication`	OSPFv2 のインターフェイス認証モードをクリアテキスト タイプとメッセージ ダイジェスト タイプのどちらかでイネーブルにします。これにより、エリアに基づくこのインターフェイスの認証が無効となります。すべてのネイバーが、この認証タイプを共有する必要があります。
ステップ 5	（任意） **ip ospf authentication key-chain** *key-name* 例： `switch(config-if)# ip ospf authentication key-chain Test1`	OSPFv2 のキーチェーンを使用するようにインターフェイス認証を設定します。キーチェーン実装の詳細については、Cisco Nexus 3548 スイッチ NX-OS セキュリティ構成ガイドを参照してください。
ステップ 6	（任意） **ip ospf authentication-key** [**0**	**3**	**7**] *key* 例： `switch(config-if)# ip ospf authentication-key 0 mypass`	このインターフェイスに簡易パスワード認証を設定します。認証が、キーチェーンにもメッセージ ダイジェストにも設定されていない場合は、このコマンドを使用します。 オプションは次のとおりです。 • 0：パスワードをクリアテキストで設定します。 • 3：パス キーを 3DES 暗号化として設定します。 • 7：パス キーを Cisco タイプ 7 暗号化として設定します。

	コマンドまたはアクション	目的
ステップ 7	（任意） **ip ospf message-digest-key** *key-id* **md5** [**0** | **3** | **7**] *key* 例： `switch(config-if)# ip ospf message-digest-key 21 md5 0 mypass`	このインターフェイスにメッセージダイジェスト認証を設定します。認証がメッセージダイジェストに設定されている場合は、このコマンドを使用します。key-id の範囲は 1 ～ 255 です。MD5 オプションは次のとおりです。 • 0：パスワードをクリアテキストで設定します。 • 3：パス キーを 3DES 暗号化として設定します。 • 7：パス キーを Cisco タイプ 7 暗号化として設定します。
ステップ 8	（任意） **show ip ospf** *instance-tag* **interface** *interface-type slot/port* 例： `switch(config-if)# show ip ospf 201 interface ethernet 1/2`	OSPF 情報を表示します。
ステップ 9	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、インターフェイスに暗号化されていない簡単なパスワードを設定し、イーサネット インターフェイス 1/2 のパスワードを設定する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip router ospf 201 area 0.0.0.10
switch(config-if)# ip ospf authentication
switch(config-if)# ip ospf authentication-key 0 mypass
switch(config-if)# copy running-config startup-config
```

高度なOSPFv2の設定

OSPFv2 は、OSPFv2 ネットワークを設計した後に設定します。

境界ルータのフィルタ リストの設定

OSPFv2 ドメインを関連ネットワークを含む一連のエリアに分割できます。すべてのエリアは、エリア境界ルータ（ABR）経由でバックボーンエリアに接続している必要があります。OSPFv2 ドメインは、自律システム境界ルータ（ASBR）を介して、外部ドメインに接続可能です。「エリア」の項を参照してください。

ABR には、省略可能な次の設定パラメータがあります。

- Area range：エリア間のルート集約を設定します。
- Filter list：ABR 上で、外部エリアから受信したネットワーク集約（タイプ 3）LSA をフィルタリングします。

ASBR もフィルタ リストをサポートしています。

始める前に

OSPF 機能を有効にしてあることを確認します（OSPFv2 機能のイネーブル化のセクションを参照）。

フィルタ リストが、着信または発信ネットワーク集約（タイプ 3）LSA の IP プレフィックスのフィルタリングに使用するルート マップを作成します。Route Policy Manager の設定を参照してください。

手順の概要

1. configure terminal
2. **router ospf** *instance-tag*
3. **area** *area-id* **filter-list route-map** *map-name* { **in** | **out** }
4. （任意） **show ip ospf policy statistics area** *id* **filter-list** { **in** | **out** }
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	configure terminal
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

ステップ 3	**area** *area-id* **filter-list route-map** *map-name* { **in**	**out** } 例：	ABR 上で着信または発信ネットワーク集約（タイプ 3）LSA をフィルタリングします。

	コマンドまたはアクション	目的
	`switch(config-router)# area 0.0.0.10 filter-list route-map FilterLSAs in`	
ステップ 4	（任意） **show ip ospf policy statistics area** *id* **filter-list** { **in** | **out** } 例： `switch(config-if)# show ip ospf policy statistics area 0.0.0.10 filter-list in`	OSPF ポリシー情報を表示します。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、エリア 0.0.0.10 でフィルタ リストを設定する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 filter-list route-map FilterLSAs in
switch(config-router)# copy running-config startup-config
```

スタブ エリアの設定

OSPFv2 ドメインの外部トラフィックが不要な個所にスタブエリアを設定できます。スタブエリアはAS外部（タイプ5）LSAをブロックし、不要な、選択したネットワークへの往復のルーティングを制限します。「スタブエリア」の項を参照してください。また、すべての集約ルートがスタブ エリアを経由しないようブロックすることもできます。

始める前に

OSPF 機能を有効にしてあることを確認します（OSPFv2 機能のイネーブル化のセクションを参照）。

設定されるスタブエリア内に、仮想リンクとASBRのいずれも含まれないことを確認します。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **area** *area-id* **stub**
4. （任意） **area** *area-id* **default-cost** *cost*
5. （任意） **show ip ospf** *instance-tag*
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

ステップ 3	**area** *area-id* **stub** 例： `switch(config-router)# area 0.0.0.10 stub`	このエリアをスタブ エリアとして作成します。
ステップ 4	（任意） **area** *area-id* **default-cost** *cost* 例： `switch(config-router)# area 0.0.0.10 default-cost 25`	このスタブ エリアに送信されるデフォルト サマリ ルートのコスト メトリックを設定します。指定できる範囲は 0 ～ 16777215 です。デフォルトは 1 です。
ステップ 5	（任意） **show ip ospf** *instance-tag* 例： `switch(config-if)# show ip ospf 201`	OSPF 情報を表示します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、スタブ エリアを作成する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 stub
switch(config-router)# copy running-config startup-config
```

Totally Stubby エリアの設定

Totally Stubby エリアを作成して、すべての集約ルート更新がスタブ エリアに入るのを防ぐことができます。

Totally Stubby エリアを作成するには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
area *area-id* **stub no-summary** 例： `switch(config-router)# area 20 stub no-summary`	このエリアを Totally Stubby エリアとして作成します。

NSSA の設定

OSPFv2 ドメインの一部で一定限度の外部トラフィックが必要な場合は、その部分に NSSA を設定できます。「Not-So-Stubby エリア」のセクションを参照してください。また、この外部トラフィックを AS 外部（タイプ 5）LSA に変換して、このルーティング情報で OSPFv2 ドメインをフラッディングすることもできます。NSSA は、省略可能な次のパラメータで設定できます。

- No redistribution：再配布されたルートは、NSSA をバイパスして OSPFv2 自律システム内の他のエリアに再配布されます。このオプションは、NSSA ASBR が ABR も兼ねているときに使用します。
- Default information originate：外部自律システムへのデフォルト ルートの NSSA 外部（タイプ 7）LSA を生成します。このオプションは、ASBR のルーティング テーブルにデフォルト ルートが含まれる場合に NSSA ASBR 上で使用します。このオプションは、ASBR のルーティング テーブルにデフォルト ルートが含まれるかどうかに関係なく、NSSA ASBR 上で使用できます。
- Route map：目的のルートだけが NSSA および他のエリア全体でフラッディングされるように、外部ルートをフィルタリングします。
- Translate：NSSA 外のエリア向けに、NSSA 外部 LSA を AS 外部 LSA に変換します。再配布されたルートを OSPFv2 自律システム全体でフラッディングするには、このコマンドを NSSA ABR 上で使用します。また、これらの AS 外部 LSA の転送アドレスを無効にすることもできます。このオプションを選択した場合は、転送アドレスが 0.0.0.0 に設定されます。
- No summary：すべての集約ルートが NSSA でフラッディングされないようにします。このオプションは NSSA ABR 上で使用します。

始める前に

OSPF 機能を有効にしてあることを確認します（OSPFv2 機能のイネーブル化のセクションを参照）。

設定する NSSA 上に仮想リンクがないことと、この NSSA がバックボーン エリアでないことを確認します。

手順の概要

1. **configure terminal**

2. **router ospf** *instance-tag*
3. **area** *area-id* **nssa** [**no-redistribution**] [**default-information-originate**] [**route-map** *map-name*]] [**no-summary**] [**translate type7** { **always** | **never** } [**suppress-fa**]]
4. （任意） **area** *area-id* **default-cost** *cost*
5. （任意） **show ip ospf** *instance-tag*
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

ステップ 3	**area** *area-id* **nssa** [**no-redistribution**] [**default-information-originate**] [**route-map** *map-name*]] [**no-summary**] [**translate type7** { **always** | **never** } [**suppress-fa**]] 例： `switch(config-router)# area 0.0.0.10 nssa`	このエリアを NSSA として作成します。
ステップ 4	（任意） **area** *area-id* **default-cost** *cost* 例： `switch(config-router)# area 0.0.0.10 default-cost 25`	この NSSA に送信されるデフォルト集約ルートのコスト メトリックを設定します。
ステップ 5	（任意） **show ip ospf** *instance-tag* 例： `switch(config-if)# show ip ospf 201`	OSPF 情報を表示します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、すべての集約ルート更新をブロックする NSSA を作成する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
```

```
switch(config-router)# area 0.0.0.10 nssa no-summary
switch(config-router)# copy running-config startup-config
```

次に、デフォルト ルートを生成する NSSA を作成する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 nssa default-info-originate
switch(config-router)# copy running-config startup-config
```

次に、外部ルートをフィルタリングし、すべての集約ルート更新をブロックするNSSA を作成する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 nssa route-map ExternalFilter no-summary
switch(config-router)# copy running-config startup-config
```

次に、常に NSSA 外部（タイプ 5）LSA を AS 外部（タイプ 7）LSA に変換する NSSA を作成する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 nssa translate type 7 always
switch(config-router)# copy running-config startup-config
```

仮想リンクの設定

仮想リンクは、隔離されたエリアを中継エリアを介してバックボーンエリアに接続します。仮想リンク セクションを展開します。仮想リンクには、省略可能な次のパラメータを設定できます。

- Authentication：簡単なパスワード認証または MD5 メッセージ ダイジェスト認証、および関連付けられたキーを設定します。
- Dead interval：ローカル ルータがデッドであることを宣言し、隣接関係を解消する前に、ネイバーが hello パケットを待つ時間を設定します。
- Hello interval：連続する hello パケット間の時間間隔を設定します。
- Retransmit interval：連続する LSA 間の推定時間間隔を設定します。
- Transmit delay：LSA をネイバーに送信する推定時間を設定します。

（注） リンクがアクティブになる前に、関与する両方のルータで仮想リンクを設定する必要があります。

スタブ エリアには仮想リンクを追加できません。

始める前に

OSPF 機能を有効にしてあることを確認します（OSPFv2 機能のイネーブル化のセクションを参照）。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **area** *area-id* **virtual-link** *router-id*
4. （任意） **show ip ospf virtual-link** [**brief**]
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

ステップ 3	**area** *area-id* **virtual-link** *router-id*
	例：
	`switch(config-router)# area 0.0.0.10 virtual-link 10.1.2.3`
	`switch(config-router-vlink)#`
		リモート ルータへの仮想リンクの端を作成します。仮想リンクをリモート ルータ上に作成して、リンクを完成させる必要があります。

ステップ 4	（任意） **show ip ospf virtual-link** [**brief**] 例： `switch(config-router-vlink)# show ip ospf virtual-link`	OSPF 仮想リンク情報を表示します。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config-router-vlink)# copy running-config startup-config`	この設定変更を保存します。

例

仮想リンク コンフィギュレーション モードで、省略可能な次のコマンドを設定できます。

コマンド	目的
authentication [**key-chain** *key-id* | **message-digest** | **null**] 例： `switch(config-router-vlink)# authentication message-digest`	エリアに基づくこの仮想リンクの認証がオーバーライドされます。
authentication-key [**0** | **3**] *key* 例： `switch(config-router-vlink)# authentication-key 0 mypass`	この仮想リンクに簡易パスワードを設定します。認証が、キーチェーンにもメッセージダイジェストにも設定されていない場合は、このコマンドを使用します。0 の場合は、パスワードをクリア テキストで設定します。3 の場合は、パスワードを 3DES 暗号化として設定します。
dead-interval *seconds* 例： `switch(config-router-vlink)# dead-interval 50`	OSPFv2 デッド間隔を秒単位で設定します。有効な範囲は 1 ～ 65535 です。デフォルトでは、hello 間隔の秒数の 4 倍です。
hello-interval *seconds* 例： `switch(config-router-vlink)# hello-interval 25`	OSPFv2 hello 間隔を秒単位で設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 10 秒です。
message-digest-key *key-id* **md5** [**0** | **3**] *key* 例： `switch(config-router-vlink)# message-digest-key 21 md5 0 mypass`	この仮想リンクにメッセージダイジェスト認証を設定します。認証がメッセージダイジェストに設定されている場合は、このコマンドを使用します。0の場合は、パスワードをクリア テキストで設定します。3 の場合は、パス キーを 3DES 暗号化として設定します。
retransmit-interval*seconds* 例： `switch(config-router-vlink)# retransmit-interval 50`	OSPFv2 再送信間隔を秒単位で設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 5 分です。
transmit-delay *seconds* 例： `switch(config-router-vlink)# transmit-delay 2`	OSPFv2 送信遅延を秒単位で設定します。指定できる範囲は 1 ～ 450 です。デフォルトは 1 です。 .

次に、2 つの ABR 間に簡単な仮想リンクを作成する例を示します。

ABR 1（ルータ ID 27.0.0.55）の設定は、次のとおりです。

```
switch# configure terminal
switch(config)# router ospf 201
```

```
switch(config-router)# area 0.0.0.10 virtual-link 10.1.2.3
switch(config-router-vlink)# copy running-config startup-config
```

ABR 2（ルータ ID 10.1.2.3）の設定は、次のとおりです。

```
switch# configure terminal
switch(config)# router ospf 101
switch(config-router)# area 0.0.0.10 virtual-link 27.0.0.55
switch(config-router-vlink)# copy running-config startup-config
```

再配布の設定

他のルーティング プロトコルから学習したルートを、ASBR 経由で OSPFv2 自律システムに再配布できます。

OSPF でのルート再配布には、省略可能な次のパラメータを設定できます。

- Default information originate：外部自律システムへのデフォルト ルートの AS 外部（タイプ 5）LSA を生成します。

（注） Default information originate はオプションのルート マップ内の **match** 文を無視します。

- Default metric：すべての再配布ルートに同じコスト メトリックを設定します。

（注） スタティック ルートを再配布する場合、デフォルトの 7.0(3)I7(6) スタティック ルートを正常に再配布するためには、Cisco NX-OS は **default-information originate** コマンドを必要とします。

始める前に

OSPF 機能を有効にしてあることを確認します（OSPFv2 機能のイネーブル化のセクションを参照）。

再配布で使用する、必要なルート マップを作成します。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **redistribute** { **bgp** *id* | **direct** | **eigrp** *id* | **ospf** *id* | **rip** *id* | **static** } **route-map** *map-name*
4. **default-information originate** [**always**] [**route-map** *map-name*]
5. **default-metric** *cost*
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

ステップ 3	**redistribute** { **bgp** *id* | **direct** | **eigrp** *id* | **ospf** *id* | **rip** *id* | **static** } **route-map** *map-name* 例： `switch(config-router)# redistribute bgp 64496 route-map FilterExternalBGP`	設定したルートマップ経由で、選択したプロトコルを OSPF に再配布します。 （注） スタティック ルートを再配布する場合、デフォルトの 7.0(3)I7(6) スタティック ルートを正常に再配布するためには、Cisco NX-OS は **default-information originate** コマンドを必要とします。
ステップ 4	**default-information originate** [**always**] [**route-map** *map-name*] 例： `switch(config-router)# default-information-originate route-map DefaultRouteFilter`	デフォルト ルートが RIB に存在する場合は、この OSPF ドメインにデフォルト ルートを作成します。次の省略可能なキーワードを使用します。 • **always**：ルートが RIB に存在しない場合でも、常にデフォルト ルート 0.0.0. を生成します。 • **route-map**：ルート マップが true を返す場合にデフォルト ルートを生成します。 このコマンドは、ルートマップの **match** 文を無視します。
ステップ 5	**default-metric** *cost* 例： `switch(config-router)# default-metric 25`	再配布されたルートのコストメトリックを設定します。これは、直接接続されたルートには適用されません。ルート マップを使用して、直接接続されたルートのデフォルトのメトリックを設定します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config-router)# copy running-config startup-config`	この設定変更を保存します。

例

次に、ボーダー ゲートウェイ プロトコル（BGP）を OSPF に再配布する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# redistribute bgp route-map FilterExternalBGP
switch(config-router)# copy running-config startup-config
```

再配布されるルート数の制限

ルートの再配布によって、OSPFv2 ルート テーブルに多くのルートが追加される可能性があります。外部プロトコルから受け取るルートの数の上限を設定できます。OSPFv2 には、再配布ルートの制限を設定するために次のオプションが用意されています。

- 上限固定：設定された最大値に OSPFv2 が達すると、メッセージをログに記録します。OSPFv2は以降の再配布ルートを受け取りません。任意で、最大値のしきい値パーセンテージを設定して、OSPFv2 がこのしきい値を超えたときに警告を記録するようにすることもできます。
- 警告のみ：OSPFv2 が最大値に達したときのみ、警告のログを記録します。OSPFv2 は、再配布されたルートを受け入れ続けます。
- 取り消し：OSPFv2 が最大値に達したときにタイムアウト期間を開始します。このタイムアウト期間後、現在の再配布されたルート数が最大制限より少なければ、OSPFv2 はすべての再配布されたルートを要求します。再配布されたルートの現在数が最大数に達した場合、OSPFv2 はすべての再配布されたルートを取り消します。OSPFv2 が追加の再配布されたルートを受け付ける前に、この状況を解消する必要があります。

  任意で、タイムアウト期間を設定できます。

始める前に

OSPF 機能を有効にしてあることを確認します（OSPFv2 機能のイネーブル化のセクションを参照）。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **redistribute** { **bgp** *id* | **direct** | **eigrp** *id* | **ospf** *id* | **rip** *id* | **static** } **route-map** *map-name*
4. **redistribute maximum-prefix** *max* [*threshold*] [**warning-only** | **withdraw** [*num-retries timeout*]]
5. （任意） **show running-config ospf**
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

ステップ 3	**redistribute** { **bgp** *id* | **direct** | **eigrp** *id* | **ospf** *id* | **rip** *id* | **static** } **route-map** *map-name* 例： `switch(config-router)# redistribute bgp 64496 route-map FilterExternalBGP`	設定したルートマップ経由で、選択したプロトコルを OSPF に再配布します。
ステップ 4	**redistribute maximum-prefix** *max* [*threshold*] [**warning-only** | **withdraw** [*num-retries timeout*]] 例： `switch(config-router)# redistribute maximum-prefix 1000 75 warning-only`	OSPFv2 で配布する最大プレフィックス数を指定します。指定できる範囲は 0 ～ 65536 です。任意で次のオプションを指定します。 • *threshold*：警告メッセージをトリガーする最大プレフィックスの割合。 • **warning-only**：プレフィックスの最大数を超えた場合に警告メッセージを記録します。 • **withdraw**：再配布されたすべてのルートを取り消します。任意で再配布されたルートを取得しようと試みます。*num-retries* の範囲は 1 ～ 12 です。*timeout* は 60 ～ 600 秒です。デフォルトは 300 秒です。**clear ip ospf redistribution** コマンドは、すべてのルートを取り消す場合に使用します。
ステップ 5	（任意） **show running-config ospf** 例： `switch(config-router)# show running-config ospf`	OSPFv2 設定を表示します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config-router)# copy running-config startup-config`	この設定変更を保存します。

例

次に、OSPF に再配布されるルートの数を制限する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# redistribute bgp route-map FilterExternalBGP
switch(config-router)# redistribute maximum-prefix 1000 7
```

ルート集約の設定

集約したアドレス範囲を設定することにより、エリア間ルートのルート集約を設定できます。また、ASBR 上のこれらのルートのサマリアドレスを設定して、外部の再配布されたルートのルート集約を設定することもできます。「ルート集約」の項を参照してください。

始める前に

OSPF 機能を有効にしてあることを確認します（OSPFv2 機能のイネーブル化のセクションを参照）。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **area** *area-id* **range** *ip-prefix/length* [**no-advertise**] [**cost** *cost*]
4. **summary-address** *ip-prefix/length* [**no-advertise** | **tag** *tag-id*]
5. （任意） show ip ospf summary-address
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

ステップ 3	**area** *area-id* **range** *ip-prefix/length* [**no-advertise**] [**cost** *cost*] 例： `switch(config-router)# area 0.0.0.10 range 10.3.0.0/16`	一定の範囲のアドレスの集約アドレスを ABR 上に作成します。この集約アドレスをネットワーク集約（タイプ 3）LSA にアドバタイズしないようにすることもできます。*cost* の範囲は 0 ～ 16777215 です。

	コマンドまたはアクション	目的
ステップ 4	**summary-address** *ip-prefix/length* [**no-advertise** | **tag** *tag-id*] 例： `switch(config-router)# summary-address 10.5.0.0/16 tag 2`	一定の範囲のアドレスのサマリ アドレスを ABR 上に作成します。ルートマップによる再配布で使用できるよう、このサマリ アドレスにタグを割り当てることもできます。
ステップ 5	（任意） show ip ospf summary-address 例： `switch(config-router)# show ip ospf summary-address`	OSPF サマリ アドレスに関する情報を表示します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config-router)# copy running-config startup-config`	この設定変更を保存します。

例

次に、ABR 上のエリア間のサマリ アドレスを作成する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# area 0.0.0.10 range 10.3.0.0/16
switch(config-router)# copy running-config startup-config
```

次に、ASBR 上のサマリ アドレスを作成する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# summary-address 10.5.0.0/16
switch(config-router)# copy running-config startup-config
```

スタブ ルート アドバタイズメントの設定

短期間だけ、このルータ経由の OSPFv2 トラフィックを制限する場合は、スタブ ルート アドバタイズメントを使用します。OSPFv2 スタブルータ アドバタイズメントのセクションを参照してください。

スタブ ルート アドバタイズメントは、省略可能な次のパラメータで設定できます。

- On startup：指定した宣言期間だけ、スタブ ルート アドバタイズメントを送信します。
- Wait for BGP：BGP がコンバージェンスするまで、スタブ ルート アドバタイズメントを送信します。

始める前に

OSPF 機能を有効にしてあることを確認します（OSPFv2 機能のイネーブル化のセクションを参照）。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **max-metric router-lsa** [**on-startup** [*announce-time*] [**wait-for bgp** *tag*]]
4. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

ステップ 3	**max-metric router-lsa** [**on-startup** [*announce-time*] [**wait-for bgp** *tag*]] 例： `switch(config-router)# max-metric router-lsa`	OSPFv2 スタブ ルート アドバタイズメントを設定します。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config-router)# copy running-config startup-config`	この設定変更を保存します。

例

次に、スタブ ルータ アドバタイズメント機能を、起動時に、デフォルトの 600 秒間有効にする例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# max-metric router-lsa on-startup
switch(config-router)# copy running-config startup-config
```

デフォルト タイマーの変更

OSPFv2 には、プロトコル メッセージの動作および最短パス優先（SPF）の計算を制御する多数のタイマーが含まれています。OSPFv2 には、省略可能な次のタイマー パラメータが含まれます。

- LSA arrival time：ネイバーから着信する LSA 間で許容される最小間隔を設定します。この時間より短時間で到着する LSA はドロップされます。
- Pacing LSAs：LSA が集められてグループ化され、リフレッシュされて、チェックサムが計算される間隔、つまり期限切れとなる間隔を設定します。このタイマーは、LSA 更新が実行される頻度を制御し、LSA 更新メッセージで送信される LSA 更新の数を制御します（フラッディングと LSA グループ ペーシングを参照）。
- Throttle LSAs：LSA 生成のレート制限を設定します。このタイマーは、トポロジが変更されない場合に LSA が生成される頻度を制御します。
- Throttle SPF calculation：SPF 計算の実行頻度を制御します。

インターフェイス レベルでは、次のタイマーも制御できます。

- Retransmit interval：連続する LSA 間の推定時間間隔を設定します。
- Transmit delay：LSA をネイバーに送信する推定時間を設定します。

hello 間隔とデッド タイマーに関する情報の詳細については、「OSPFv2でのネットワークの設定」の項を参照してください。

始める前に

OSPF 機能を有効にしてあることを確認します（OSPFv2 機能のイネーブル化のセクションを参照）。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **timers lsa-arrival** *msec*
4. **timers lsa-group-pacing** *seconds*
5. **timers throttle lsa** *start-time hold-interval max-time*
6. **timers throttle spf** *delay-time hold-time max-time*
7. **interface** *type slot/port*
8. **no switchport**
9. **ip ospf hello-interval** *seconds*
10. **ip ospf dead-interval** *seconds*
11. **ip ospf retransmit-interval** *seconds*
12. **ip ospf transmit-delay** *seconds*
13. （任意） **show ip ospf**
14. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router ospf** *instance-tag*
	例：
	`switch(config)# router ospf 201`
	`switch(config-router)#`
		新規OSPFv2インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

ステップ 3	**timers lsa-arrival** *msec* 例： `switch(config-router)# timers lsa-arrival 2000`	LSA 到着時間をミリ秒で設定します。範囲は 10 ～ 600000 です。デフォルトは 1000 ミリ秒です。
ステップ 4	**timers lsa-group-pacing** *seconds* 例： `switch(config-router)# timers lsa-group-pacing 1800`	LSA がグループ化される間隔を秒で設定します。範囲は 1 ～ 1800 です。デフォルトは 240 秒です。
ステップ 5	**timers throttle lsa** *start-time hold-interval max-time* 例： `switch(config-router)# timers throttle lsa 3000 6000 6000`	次のタイマーを使用して、LSA 生成のレート制限をミリ秒で設定します。 *start-time*：指定できる範囲は 50 ～ 5000 ミリ秒です。デフォルト値は 50 ミリ秒です。 *hold-interval*：指定できる範囲は 50 ～ 30,000 ミリ秒です。デフォルト値は 5000 ミリ秒です。 *max-time*：指定できる範囲は 50 ～ 30,000 ミリ秒です。デフォルト値は 5000 ミリ秒です。
ステップ 6	**timers throttle spf** *delay-time hold-time max-time* 例： `switch(config-router)# timers throttle spf 3000 2000 5000`	SPF 最適パス計算間の SPF 最適パス スケジュール初期遅延時間、最小保持時間、および最大待機時間（ミリ秒単位）を設定します。範囲は 1 ～ 600000 ミリ秒です。デフォルト値は、200 ミリ秒の遅延時間、1000 ミリ秒の保持時間、および 5000 ミリ秒の待機時間です。

ステップ 7	**interface** *type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

ステップ 8	**no switchport** 例：	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。

	コマンドまたはアクション	目的
	`switch(config-if)# no switchport`	
ステップ 9	**ip ospf hello-interval** *seconds* 例： `switch(config-if)# ip ospf retransmit-interval 30`	このインターフェイスの hello 間隔を設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 10 です。
ステップ 10	**ip ospf dead-interval** *seconds* 例： `switch(config-if)# ip ospf dead-interval 30`	このインターフェイスのデッド間隔を設定します。有効な範囲は 1 ～ 65535 です。
ステップ 11	**ip ospf retransmit-interval** *seconds* 例： `switch(config-if)# ip ospf retransmit-interval 30`	このインターフェイスから送信される各 LSA 間の推定時間間隔を設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 5 分です。

ステップ 12	**ip ospf transmit-delay** *seconds*
	例：
	`switch(config-if)# ip ospf transmit-delay 450`
	`switch(config-if)#`
		LSA をネイバーに送信する推定時間間隔を秒で設定します。指定できる範囲は1～450です。デフォルトは 1 です。

ステップ 13	（任意） **show ip ospf** 例： `switch(config-if)# show ip ospf`	OSPF に関する情報を表示します。
ステップ 14	（任意） **copy running-config startup-config** 例： `switch(config-router)# copy running-config startup-config`	この設定変更を保存します。

例

次に、lsa-group-pacing オプションで LSA フラッディングを制御する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# timers lsa-group-pacing 300
switch(config-router)# copy running-config startup-config
```

OSPFv2 インスタンスの再起動

OSPv2インスタンスを再起動できます。再起動すると、インスタンスのすべてのネイバーが消去されます。

OSPFv2 インスタンスを再起動して、関連付けられたすべてのネイバーを削除するには、次のコマンドを使用します。

コマンド	目的
restart ospf *instance-tag* 例： `switch(config)# restart ospf 201`	OSPFv2 インスタンスを再起動して、すべてのネイバーを削除します。

仮想化による OSPFv2 の設定

複数の VRF を作成できます。また、各 VRF で同じ OSPFv2 インスタンスを使用することも、複数の OSPFv2 インスタンスを使用することも可能です。VRF には OSPFv2 インターフェイスを割り当てます。

（注） インターフェイスの VRF を設定した後に、インターフェイスの他のすべてのパラメータを設定します。インターフェイスの VRF を設定すると、そのインターフェイスのすべての設定が削除されます。

始める前に

OSPF 機能がイネーブルになっていることを確認します。OSPFv2 機能のイネーブル化のセクションを参照してください。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **router ospf** *instance-tag*
4. **vrf** *vrf-name*
5. （任意） **maximum-paths** *paths*
6. **interface** *interface-typeslot/port*
7. **no switchport**
8. **vrf member** *vrf-name*
9. **ip address** *ip-prefix/length*
10. **ip router ospf instance-tag area** *area-id*
11. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

	コマンドまたはアクション	目的

ステップ 2	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context RemoteOfficeVRF`
	`switch(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。

ステップ 3	**router ospf** *instance-tag*
	例：
	`switch(config-vrf)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

ステップ 4	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf RemoteOfficeVRF`
	`switch(config-router-vrf)#`
		VRF 設定モードを開始します。

ステップ 5	（任意） **maximum-paths** *paths* 例： `switch(config-router-vrf)# maximum-paths 4`	この VRF のルート テーブル内の宛先への、同じ OSPFv2 パスの最大数を設定します。ロードバランシングに使用されます。

ステップ 6	**interface** *interface-typeslot/port*
	例：
	`switch(config-router-vrf)# interface ethernet 1/2`
	`switch(config-if)#`
		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 7	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。
ステップ 8	**vrf member** *vrf-name* 例： `switch(config-if)# vrf member RemoteOfficeVRF`	このインターフェイスを VRF に追加します。
ステップ 9	**ip address** *ip-prefix/length* 例： `switch(config-if)# ip address 192.0.2.1/16`	このインターフェイスの IP アドレスを設定します。このステップは、このインターフェイスを VRF に割り当てたあとに行う必要があります。
ステップ 10	**ip router ospf instance-tag area** *area-id* 例： `switch(config-if)# ip router ospf 201 area 0`	このインターフェイスを OSPFv2 インスタンスおよび設定エリアに割り当てます。
ステップ 11	（任意） **copy running-config startup-config** 例： `switch(config-router)# copy running-config startup-config`	この設定変更を保存します。

例

次に、VRF を作成して、その VRF にインターフェイスを追加する例を示します。

```
switch# configure terminal
switch(config)# vrf context NewVRF
switch(config)# router ospf 201
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# vrf member NewVRF
switch(config-if)# ip address 192.0.2.1/16
switch(config-if)# ip router ospf 201 area 0
switch(config)# copy running-config startup-config
```

OSPFv2 設定の確認

show ip ospf	OSPFv2 設定を表示します。
show ip ospf border-routers [**vrf** { **vrf-name** | **all** | **default** | **management** }]	OSPFv2 境界ルータ設定を表示します。
show ip ospf database [**vrf** { *vrf-name* | **all** | **default** | **management** }]	OSPFv2 リンクステート データベースの要約を表示します。
show ip ospf interface number [**vrf** { *vrf-name* | **all** | **default** | **management** }]	OSPFv2 インターフェイス設定を表示します。
show ip ospf lsa-content-changed-list *interface - type number*	変更された OSPFv2 LSA を表示します。
show ip ospf neighbors [*neighbor-id*] [**detail**] [*interface - type number*] [**vrf** { *vrf-name* | **all** | **default** | **management** }] [**summary**]	OSPFv2 ネイバーの一覧を表示します。
show ip ospf request-list *neighbor-id* [*interface - type number*]	OSPFv2 リンクステート要求の一覧を表示します。
show ip ospf retransmission-list *neighbor-id* [*interface - type number*]	OSPFv2 リンクステート再送の一覧を表示します。
show ip ospf route [*ospf-route*] [**summary**] [**vrf** { *vrf-name* | **all** | **default** | **management** }]	内部 OSPFv2 ルートを表示します。
show ip ospf summary-address [**vrf** { *vrf-name* | **all** | **default** | **management** }]	OSPFv2 サマリ アドレスに関する情報を表示します。
show ip ospf virtual-links [**brief**] [**vrf** { *vrf-name* | **all** | **default** | **management** }]	OSPFv2 仮想リンクに関する情報を表示します。

show ip ospf vrf { *vrf-name* | **all** | **default** | **management** }	VRF ベースの OSPFv2 設定に関する情報を表示します。
show running-configuration ospf	現在実行中の OSPFv2 設定を表示します。

OSPFv2 統計情報の表示

OSPFv2 統計情報を表示するには、次のコマンドを使用します。

コマンド	目的
show ip ospf policy statistics area *area-id* **filter-list** { **in** | **out** } [**vrf** { *vrf-name* | **all** | **default** | **management** }]	エリアの OSPFv2 ルート ポリシー統計情報を表示します。
show ip ospf policy statistics redistribute { **bgp** *id* | **direct** | **eigrp** *id* | **ospf** *id* | **rip** *id* | **static** } **vrf** { *vrf-name* | **all** | **default** | **management** }]	OSPFv2 ルート ポリシー統計情報を表示します。
show ip ospf statistics [**vrf** { *vrf-name* | **all** | **default** | **management** }]	OSPFv2 イベント カウンタを表示します。
show ip ospf traffic [*interface - type number*] [**vrf** { *vrf-name* | **all** | **default** | **management** }]	OSPFv2 パケット カウンタを表示します。

OSPFv2 の設定例

次に、OSPFv2 を設定する例を示します。

```
feature ospf
router ospf 201
router-id 290.0.2.1

interface ethernet 1/2
no switchport
ip router ospf 201 area 0.0.0.10
ip ospf authentication
ip ospf authentication-key 0 mypass
```

その他の参考資料

OSPF の実装に関する詳細情報については、次のページを参照してください。

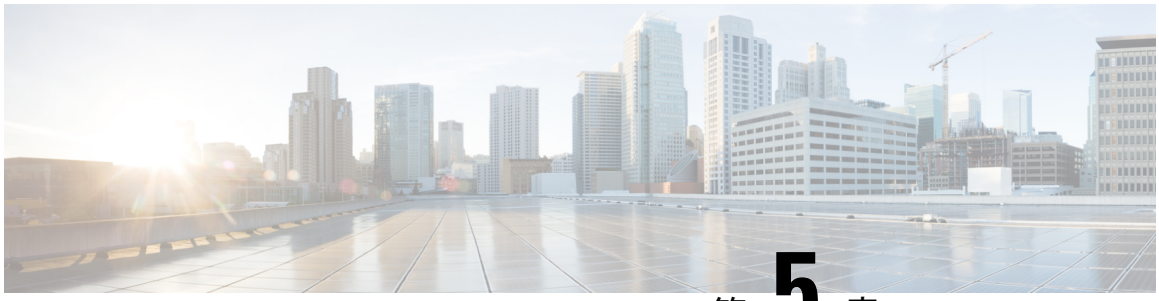
- 関連資料
- MIB

関連資料

関連項目	マニュアル タイトル
OSPFv2 CLI コマンド	『Cisco Nexus 3000 Series Command Reference』
ルート マップ	Route Policy Manager の設定

MIB

MIB	MIB のリンク
• OSPF-MIB • OSPF-IRAP-MIB	MIB を検索してダウンロードするには、次の MIB ロケータに移動します。



第 5 章

EIGRP の設定

この章では、Cisco NX-OS スイッチで Enhanced Interior Gateway Routing Protocol（EIGRP）を設定する方法について説明します。

この章は、次の項で構成されています。

EIGRP に関する情報

EIGRP は、リンクステート プロトコルの機能にディスタンス ベクトル プロトコルの利点を組み合わせたプロトコルです。EIGRP は、定期的に Hello メッセージを送信してネイバーを探索します。EIGRP は、新規ネイバーを検出すると、すべてのローカル EIGRP ルートおよびルート メトリックに対する 1 回限りの更新を送信します。受信側の EIGRP ルータは、受信したメトリックと、その新規ネイバーにローカルで割り当てられたリンクのコストに基づいて、ルート ディスタンスを計算します。この最初の全面的なルート テーブルの更新後は、ルート変更の影響を受けるネイバーにのみ、差分更新が EIGRP により送信されます。この処理により、コンバージェンスにかかる時間が短縮され、帯域幅が最小限になります。

EIGRP コンポーネント

EIGRP には、次の基本コンポーネントがあります。

信頼性の高いトランスポート プロトコル

信頼性の高いトランスポート プロトコルは、すべてのネイバーに EIGRP パケットの順序付けされた配信を保証します。(「ネイバー探索およびネイバー回復」の項を参照してください。) 信頼性の高いトランスポート プロトコルは、ユニキャスト パケットの伝送をサポートしています。この方式により、さまざまな速度のリンクでも短いコンバージェンス時間が維持されるようになります。ユニキャスト パケットの送信を制御するデフォルト タイマーの変更の詳細については、高度な EIGRP の設定 を参照してください。

Reliable Transport Protocol には、次のメッセージ タイプが含まれます。

- Hello：ネイバー探索およびネイバー回復に使用されます。EIGRP はデフォルトでは、定期的なマルチキャスト Hello メッセージをローカル ネットワーク上に、設定された hello 間隔で送信します。デフォルトの hello 間隔は 5 秒です。
- 確認：更新、照会、返信を確実に受信したことを確認します。
- 更新：ルーティング情報が変更されると、その影響を受けるネイバーに送信されます。更新には、ルートの宛先、アドレス マスク、および遅延や帯域幅などのルート メトリックが含まれます。更新情報は EIGRP トポロジ テーブルに格納されます。
- 照会および返信：必要に応じて、EIGRP が使用する DUAL の一部として送信されます。

ネイバー探索およびネイバー回復

EIGRP は、Reliable Transport Protocol からの Hello メッセージを使用して、直接接続されたネットワーク上のネイバー EIGRP ルータを探索します。EIGRP により、ネイバー テーブルにネイバーが追加されます。ネイバーテーブルの情報には、ネイバーアドレス、検出されたインターフェイス、およびネイバー到達不能を宣言する前に EIGRP が待機する時間を示すホールド タイムが含まれています。デフォルトのホールド タイムは、hello 間隔の 3 倍または 15 秒です。

EIGRP は、ローカル EIGRP ルーティング情報を共有するために、一連の更新メッセージを新規ネイバーに送信します。このルート情報は EIGRP トポロジ テーブルに格納されます。このように EIGRP ルート情報全体を最初に送信した後は、ルーティングが変更されたときにのみ、EIGRP により更新メッセージが送信されます。これらの更新メッセージは新情報または更新情報のみを含んでおり、変更の影響を受けるネイバーにのみ送信されます。「EIGRP ルート更新」の項を参照してください。

EIGRP はまた、Hello メッセージをネイバーへのキープアライブのためにも使用します。Hello メッセージを受信している限り、Cisco NX-OS は、ネイバーがダウンせずに機能していると判定します。

拡散更新アルゴリズム

拡散更新アルゴリズム（DUAL）により、トポロジ テーブルの宛先ネットワークに基づいてルーティング情報が計算されます。トポロジ テーブルには、次の情報が含まれます。

- IPv4 アドレス/マスク：この宛先のネットワーク アドレスおよびネットワーク マスク。
- サクセサ：現在のフィジブルディスタンスよりも宛先まで短いディスタンスをアドバタイズする、すべてのフィジブル サクセサまたはネイバーの IP アドレスおよびローカル インターフェイス接続。
- フィージビリティディスタンス（FD）：計算された、宛先までの最短ディスタンス。フィジブル ディスタンスは、ネイバーがアドバタイズした距離に、そのネイバーへのリンク コストを加えた合計です。

DUAL は、ディスタンス メトリックを使用して、ループが発生しない効率的なパスを選択します。DUAL はルートを選択し、フィジブル サクセサに基づいてユニキャスト ルーティング情報ベース（RIB）に挿入します。トポロジが変更されると、DUAL は、トポロジ テーブルでフィジブル サクセサを探します。フィジブル サクセサが見つかった場合、DUAL は、最短のフィジブル ディスタンスを持つフィジブル サクセサを選択して、それをユニキャスト RIB に挿入します。これにより、再計算が不要となります。

フィジブル サクセサが存在しないが、宛先をアドバタイズするネイバーが存在する場合は、DUALがパッシブ状態からアクティブ状態へと移行し、新しいサクセサまたは宛先へのネクスト ホップ ルータを決定する再計算をトリガーします。ルートの再計算に必要な時間は、コンバージェンス時間に影響します。EIGRPは照会メッセージをすべてのネイバーに送信し、フィジブルサクセサを探します。フィジブルサクセサを持つネイバーは、その情報を含む返信メッセージを送信します。フィジブルサクセサを持たないネイバーは、DUALの再計算をトリガーします。

EIGRP ルート更新

トポロジが変更されると、EIGRPは、変更されたルーティング情報のみを含む更新メッセージを影響を受けるネイバーに送信します。更新メッセージには、新規の、または更新されたネットワーク宛先へのディスタンス情報が含まれます。

EIGRP でのディスタンス情報は、帯域幅、遅延、負荷使用状況、リンクの信頼性などの使用可能なルート メトリックの組み合わせとして表現されます。各メトリックには重みが関連付けられており、これにより、メトリックがディスタンスの計算に含まれるかどうかが決定します。このメトリックの重みは設定することができます。特性を微調整して最適なパスを完成することもできますが、設定可能なメトリックの大部分でデフォルト設定を使用することを推奨します。

内部ルート メトリック

内部ルートとは、同じ EIGRP 自律システム内のネイバー間のルートです。これらのルートには、次のメトリックがあります。

- ネクスト ホップ：ネクスト ホップ ルータの IP アドレス。

- 遅延：宛先ネットワークへのルートを形成するインターフェイス上で設定された遅延の合計。10 マイクロ秒単位で設定されます。
- 帯域幅：宛先へのルートの一部であるインターフェイスで設定された最小帯域幅から計算されます。

（注） デフォルト帯域幅の値の使用を推奨します。この帯域幅パラメータは EIGRP でも使用されます。

- MTU：宛先へのルート上の最大伝送単位の最小値。
- ホップ カウント：宛先までにルートが通過するホップまたはルータの数。このメトリックは、DUAL 計算で直接には使用されません。
- 信頼性：宛先までのリンクの信頼性を示します。
- 負荷：宛先までのリンク上のトラフィック量を示します。

デフォルトで EIGRP は、帯域幅と遅延のメトリックを使用して、宛先までのディスタンスを計算します。計算に他のメトリックが含まれるように、メトリックの重みを変更できます。

外部ルート メトリック

外部ルートとは、異なる EIGRP 自律システムにあるネイバー間のルートです。これらのルートには、次のメトリックがあります。

- ネクスト ホップ：ネクスト ホップ ルータの IP アドレス。
- ルータ ID：このルートを EIGRP に再配布したルータのルータ ID。
- AS 番号：宛先の自律システムの番号。
- プロトコル ID：宛先へのルートを学習したルーティング プロトコルを表すコード。
- タグ：ルート マップで使用可能な任意のタグ。
- メトリック：外部ルーティング プロトコルの、このルートのルート メトリック。

EIGRP とユニキャスト RIB

EIGRP は、すべての学習したルートを EIGRP トポロジ テーブルとユニキャスト RIB に追加します。トポロジが変更されると、EIGRP は、これらのルートを使用してフィジブル サクセサを探します。EIGRP は、他のルーティング プロトコルから EIGRP に再配布されたあらゆるルートの変更についてのユニキャスト RIB からの通知も待ち受けます。

高度な EIGRP

EIGRP の高度な機能を使用して、EIGRP の設定を最適化できます。

アドレス ファミリ

EIGRP は、IPv4 アドレス ファミリをサポートします。

アドレス ファミリ コンフィギュレーション モードには、次の EIGRP 機能が含まれます。

- 認証
- AS 番号
- デフォルト ルート
- メトリック
- ディスタンス
- グレースフル リスタート
- ロギング
- ロード バランシング
- 再分配
- ルータ ID
- スタブ ルータ
- タイマー

複数のコンフィギュレーション モードで同じ機能を設定できません。たとえばルータ コンフィギュレーション モードでデフォルト メトリックを設定すると、アドレス ファミリ モードでデフォルト メトリックを設定できません。

認証

EIGRP メッセージに認証を設定することで、ネットワークでの不正なルーティング更新や無効なルーティング更新を防止できます。EIGRP 認証は MD5 認証ダイジェストをサポートしています。

認証キーのキーチェーン管理を使用して、仮想ルーティングおよび転送（VRF）インスタンスごと、またはインターフェイスごとに EIGRP 認証を設定できます。キーチェーン管理を使用すると、MD5 認証ダイジェストが使用する認証キーへの変更を管理できます。キー チェーンの作成の詳細については、Cisco Nexus 3548 スイッチ NX-OS セキュリティ構成ガイドを参照してください。

MD5 認証を行うには、ローカル ルータとすべてのリモート EIGRP ネイバーで同一のパスワードを設定します。EIGRP メッセージが作成されると、Cisco NX-OS は、そのメッセージ自体と暗号化されたパスワードに基づいて MD5 一方向メッセージ ダイジェストを作成し、このダイジェストを EIGRP メッセージとともに送信します。受信する EIGRP ネイバーは、同じ暗号化パスワードを使用して、このダイジェストを確認します。メッセージが変更されていない場合は計算が同一であるため、EIGRP メッセージは有効と見なされます。

MD5認証には各EIGRPメッセージのシーケンス番号も含まれており、これにより、ネットワークでのメッセージの再送が防止されます。

スタブ ルータ

EIGRP スタブ ルーティング機能を使用すると、ネットワークの安定性の向上、リソース使用量の削減、スタブ ルータ設定の簡易化を実現できます。スタブ ルータは、リモート ルータ経由で EIGRP ネットワークに接続します。「スタブ ルーティング」の項を参照してください。

EIGRP スタブ ルーティングを使用すると、EIGRP を使用するように配布とリモート ルータを設定し、リモート ルータのみをスタブとして設定する必要があります。EIGRP スタブ ルーティングで、分散ルータでの集約が自動的にイネーブルになるわけではありません。ほとんどの場合、分散ルータでの集約の設定が必要です。

EIGRP スタブ ルーティングを使用しない場合は、分散ルータからリモート ルータに送信されたルートがフィルタリングまたは集約された後でも、問題が発生することがあります。たとえば、ルートが企業ネットワーク内のどこかで失われた場合に、EIGRP が分散ルータに照会を送信することがあります。分散ルータは、ルートが集約されている場合でも、リモート ルータに照会を送信することがあります。分散ルータとリモート ルータの間の WAN リンク上の通信に問題が発生した場合は、EIGRP がアクティブ状態のままとなり、ネットワークの他の場所が不安定となる場合があります。EIGRP スタブ ルーティングを使用すると、リモート ルータに照会が送信されなくなります。

ルート集約

指定したインターフェイスにサマリー集約アドレスを設定できます。ルート集約を使用すると、固有性の強い一連のアドレスをすべての固有アドレスを代表する 1 つのアドレスに置き換えることによって、ルート テーブルを簡素化できます。たとえば、10.1.1.0/24、10.1.2.0/24、および 10.1.3.0/24 というアドレスを 1 つの集約アドレス 10.1.0.0/16 に置き換えることができます。

より具体的なアドレスがルーティング テーブルにある場合、EIGRP は、より具体的なルートの最小メトリックに等しいメトリックを持つインターフェイスからの集約アドレスをアドバタイズします。

（注） EIGRP は、自動ルート集約をサポートしていません。

ルートの再配布

EIGRP を使用して、ダイレクト ルート、スタティック ルート、他の EIGRP 自律システムから学習したルート、または他のプロトコルからのルートを再配布できます。再配布を含むルート マップを設定して、どのルートが EIGRP に渡されるかを制御します。ルート マップを使用すると、宛先、送信元プロトコル、ルート タイプ、ルート タグなどの属性に基づいて、ルートをフィルタリングできます。「Route Policy Manager の設定」を参照してください。

インポートされた EIGRP へのすべてのルートに使用されるデフォルト メトリックも設定できます。

ロード バランシング

ロードバランシングを使用すると、ルータは、宛先アドレスから等距離内にあるすべてのルータのネットワーク ポートにトラフィックを分散できます。ロード バランシングにより、ネットワーク セグメントの使用率が向上し、それによってネットワーク帯域幅の効率も向上します。

Cisco NX-OS は、等コスト マルチパス（ECMP）機能をサポートします。EIGRP ルート テーブルおよびユニキャスト RIB の等コスト パスは最大 32 です。これらのパスの一部または全部に対してトラフィックのロード バランスを行うよう、EIGRP を設定できます。

（注） Cisco NX-OS の EIGRP は、等コストでないロード バランシングはサポートしていません。

（注） Cisco Nexus 3548 スイッチでは、ECMP はワープ モードでサポートされていません。

Split Horizon

スプリット ホライズンを使用すると、ルートを学習したインターフェイスから EIGRP がルートをアドバタイズしないようにできます。

スプリット ホライズンは、EIGRP 更新パケットおよび EIGRP 照会パケットの送信を制御する方式です。インターフェイスでスプリット ホライズンをイネーブルにすると、Cisco NX-OS は、このインターフェイスから学習された宛先への更新パケットも照会パケットも送信しません。この方法でアップデート パケットとクエリー パケットを制御すると、ルーティング ループが発生する可能性が低くなります。

ポイズン リバースによるスプリット ホライズンにより、EIGRP は、EIGRP がルートを学習したインターフェイス経由で、そのルートを到達不能としてアドバタイズするよう設定されます。

EIGRP は、次のシナリオでスプリット ホライズン、またはポイズン リバースによるスプリット ホライズンを使用します。

- スタートアップ モードで、2 台のルータ間で初めてトポロジ テーブルを交換する。
- トポロジ テーブルの変更をアドバタイズする。
- 照会メッセージを送信する。

デフォルトでは、スプリット ホライズン機能がすべてのインターフェイスでイネーブルになっています。

仮想化のサポート

Cisco NX-OSは、同一システム上で動作する複数のEIGRPプロトコルインスタンスをサポートします。EIGRP は、仮想ルーティングおよび転送（VRF）インスタンスをサポートしています。デフォルトでは、特に別の VRF を設定しない限り、Cisco NX-OS はユーザーをデフォルトの VRF に配置します。

デフォルトでは、すべてのインスタンスが同じシステム ルータ ID を使用します。インスタンスごとに一意のルータ ID を設定することもできます。

EIGRP の前提条件

EIGRP を使用するには、次の前提条件を満たしている必要があります。

- EIGRP 機能をイネーブルにする必要があります（EIGRP 機能の有効化を参照）。

EIGRP の注意事項と制約事項

EIGRP 設定時の注意事項および制約事項は次のとおりです。

- 他のプロトコル、接続されたルータ、またはスタティック ルートからの再配布には、メトリック設定（デフォルト メトリック設定オプションまたはルート マップによる）が必要です（Route Policy Manager の設定を参照）。
- Cisco NX-OS の EIGRP は Cisco IOS ソフトウェアの EIGRP と互換性があります。
- 妥当な理由がない限り、メトリックの重みを変更しないでください。メトリックの重みを変更した場合は、同じ自律システム内のすべての EIGRP ルータに、それを適用する必要があります。
- 大規模ネットワークの場合は、スタブの使用を検討してください。
- EIGRP ベクトル メトリックは維持されないため、異なる EIGRP 自律システム間での再配布は避けてください。
- **no ip next-hop-self** コマンドは、ネクスト ホップの到達可能性を保証しません。
- **ip passive-interface eigrp** コマンドを使用すると、ネイバーが形成されなくなります。
- Cisco NX-OS は IGRP も、IGRP および EIGRP クラウドの接続もサポートしていません。
- 自動集約は、デフォルトではイネーブルにされていません。
- Cisco NX-OS は IP のみをサポートします。

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

EIGRP のデフォルト設定

次の表は、各 EIGRP パラメータに対するデフォルト設定を示しています。

表 *6 :* デフォルト *EIGRP* パラメータ

パラメータ	デフォルト
アドミニストレーティブ ディスタンス	• 内部ルート：90 • 外部ルート：170
帯域幅の割合	50%
再配布されたルートのデフォルトのメトリック	• bandwidth：100000 kbps • delay：100（10 マイクロ秒単位） • reliability：255 • loading：1 • MTU：1500
EIGRP 機能	ディセーブル
hello 間隔	5 秒
Hold time	15 秒
等コスト パス	8
メトリック重み	1 0 1 0 0
アドバタイズされたネクスト ホップ アドレス	ローカル インターフェイスの IP アドレス
再分配	ディセーブル
スプリット ホライズン	有効（Enabled）

基本的な EIGRP の設定

EIGRP 機能の有効化

EIGRP を設定するには、その前に EIGRP 機能をイネーブルにする必要があります。

手順の概要

1. **configure terminal**
2. **feature eigrp**
3. （任意） **show feature**
4. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**feature eigrp** 例： `switch(config)# feature eigrp`	EIGRP 機能を有効にします。
ステップ 3	（任意） **show feature** 例： `switch(config)# show feature`	有効にされた機能に関する情報を表示し。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

no feature eigrp コマンドを使用して、BFD 機能を無効にし、関連する構成をすべて削除します。

コマンド	目的
no feature eigrp 例： `switch(config)# no feature eigrp`	EIGRP 機能をディセーブルにして、関連付けられたコンフィギュレーションをすべて削除します。

EIGRP インスタンスの作成

EIGRPインスタンスを作成して、そのインスタンスにインターフェイスを関連付けることができます。この EIGRP プロセスに一意の自律システム番号を割り当てます（「自律システム」の項を参照）。ルート再配布をイネーブルにしていない限り、他の自律システムからルートがアドバタイズされることも、受信されることもありません。

始める前に

EIGRP機能を有効にしていることを確認します（EIGRP機能の有効化のセクションを参照）。

EIGRPがルータ ID（設定済みのループバック アドレスなど）を入手可能であるか、またはルータ ID オプションを設定する必要があります。

AS 番号であると認められていないインスタンス タグを設定する場合は、AS 番号を明示的に設定する必要があります。そうしないと、この EIGRP インスタンスはシャットダウン状態のままになります。

手順の概要

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. （任意） **autonomous-system** *as-number*
4. （任意） **log-adjacency-changes**
5. **log-neighbor-warnings** [*seconds*]
6. **interface** *interface-type slot/port*
7. **no switchport**
8. **ip router eigrp** *instance-tag*
9. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

	コマンドまたはアクション	目的

ステップ 2	**router eigrp** *instance-tag*
	例：
	`switch(config)# router eigrp Test1`
	`switch(config-router)#`
		インスタンス タグを設定して、新しい EIGRP プロセスを作成します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。
		AS 番号であると認められていない *instance-tag* を設定する場合は、**autonomous-system** コマンドを使用して AS 番号を明示的に設定する必要があります。そうしないと、この EIGRP インスタンスはシャットダウン状態のままになります。

ステップ 3	（任意） **autonomous-system** *as-number* 例： `switch(config-router)# autonomous-system 33`	この EIGRP インスタンスに一意の AS 番号を設定します。有効な範囲は 1 ～ 65535 です。
ステップ 4	（任意） **log-adjacency-changes** 例： `switch(config-router)# log-adjacency-changes`	隣接関係の状態が変化するたびに、システム メッセージを生成します。このコマンドは、デフォルトでイネーブルになっています。
ステップ 5	**log-neighbor-warnings** [*seconds*] 例： `switch(config-router)# log-neighbor-warnings`	ネイバーの警告が発生するたびに、システム メッセージを生成します。警告メッセージの時間間隔を、1 ～ 65535 の秒数で設定できます。デフォルトは10秒です。このコマンドは、デフォルトでイネーブルになっています。

ステップ 6	**interface** *interface-type slot/port*
	例：
	`switch(config-router)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。**?** を使用すると、スロットおよびポートの範囲を確認できます。

ステップ 7	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。
ステップ 8	**ip router eigrp** *instance-tag* 例： `switch(config-if)# ip router eigrp Test1`	このインターフェイスを、設定された EIGRP プロセスに関連付けます。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。
ステップ 9	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

EIGRP プロセスと、関連付けられた設定を削除するには、**no router eigrp** コマンドを使用します。

コマンド	目的
no router eigrp *instance-tag* 例： `switch(config)# no router eigrp Test1`	EIGRP プロセスと、関連付けられたすべての設定を削除します。

（注） EIGRP プロセスを削除する場合は、インターフェイス モードで設定された EIGRP コマンドも削除する必要があります。

次に、EIGRP プロセスを作成し、EIGRP のインターフェイスを設定する例を示します。

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config)# i nterface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip router eigrp Test1
switch(config-if)# no shutdown
switch(config-if)# copy running-config startup-config
```

その他の EIGRP パラメータの詳細については、 高度な EIGRP の設定のセクションを参照してください。

EIGRP インスタンスの再起動

EIGRP インスタンスを再起動できます。再起動すると、インスタンスのすべてのネイバーが消去されます。

EIGRP インスタンスを再起動して、関連付けられたすべてのネイバーを削除するには、次のコマンドを使用します。

コマンド	目的
flush-routes 例： `switch(config)# flush-routes`	この EIGRP インスタンスを再起動するときに、ユニキャスト RIB のすべての EIGRP ルートをフラッシュします。
restart eigrp *instance-tag* 例： `switch(config)# restart eigrp Test1`	EIGRP インスタンスを再起動して、すべてのネイバーを削除します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。

EIGRP インスタンスのシャットダウン

EIGRPインスタンスを正常にシャットダウンできます。これにより、すべてのルートと隣接関係は移動しますが、EIGRP 設定は保持されます。

EIGRP インスタンスをディセーブルにするには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
shutdown 例： `switch(config-router)# shutdown`	この EIGRP インスタンスをディセーブルにします。EIGRP ルータ設定は残ります。

EIGRP のパッシブ インターフェイスの設定

EIGRP のパッシブ インターフェイスを設定できます。パッシブ インターフェイスは、EIGRP 隣接関係に参加しませんが、このインターフェイスのネットワーク アドレスは EIGRP トポロジ テーブルに残ります。

EIGRPのパッシブインターフェイスを設定するには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
ip passive-interface eigrp *instance-tag*	EIGRP hello を抑制します。これにより、EIGRP インターフェイス上でネイバーがルーティング アップデートを形成および送信することを防ぎます。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。

インターフェイスでの EIGRP のシャットダウン

インターフェイスで EIGRP を正常にシャットダウンできます。これにより、すべての隣接関係が削除され、このインターフェイスで EIGRP トラフィックが停止しますが、EIGRP 設定は保持されます。

インターフェイスで EIGRP を無効にするには、インターフェイス設定モードで次のコマンドを使用します。

コマンド	目的
ip eigrp *instance-tag* **shutdown** 例： `switch(config-router)# ip eigrp Test1 shutdown`	このインターフェイスでEIGRPを無効にします。EIGRP インターフェイス設定は残ります。インスタンスタグには最大20文字の英数字を使用できます。大文字と小文字を区別します。

高度な EIGRP の設定

このセクションは、次のトピックで構成されています。

EIGRP での認証の設定

EIGRP のネイバー間に認証を設定できます。「認証」セクションを参照してください。

EIGRP プロセスまたは個々のインターフェイスに対応する EIGRP 認証を設定できます。インターフェイスの EIGRP 認証設定は、EIGRP プロセスレベルの認証設定よりも優先します。

始める前に

EIGRP機能を有効にしていることを確認します（EIGRP機能の有効化のセクションを参照）。

EIGRPプロセスのすべてのネイバーが、共有認証キーを含め、同じ認証設定を共有することを確認します。

この認証設定のキーチェーンを作成します。Cisco Nexus 3548 スイッチ NX-OS セキュリティ構成ガイドを参照してください。

手順の概要

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. **address-family ipv4 unicast**
4. **authentication key-chain** *key-chain*
5. **authentication mode md5**
6. **interface** *interface-type slot/port*
7. **no switchport**
8. **ip router eigrp** *instance-tag*
9. **ip authentication key-chain eigrp** *instance-tag key-chain*
10. **ip authentication mode eigrp** *instance-tag* **md5**
11. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router eigrp** *instance-tag* 例：	インスタンス タグを設定して、新しい EIGRP プロセスを作成します。インスタンス タグには最大 20

	コマンドまたはアクション	目的
	`switch(config)# router eigrp Test1` `switch(config-router)#`	文字の英数字を使用できます。大文字と小文字を区別します。 AS 番号であると認められていない *instance-tag* を設定する場合は、**autonomous-system** コマンドを使用して AS 番号を明示的に設定する必要があります。そうしないと、この EIGRP インスタンスはシャットダウン状態のままになります。

ステップ 3	**address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		アドレス ファミリ コンフィギュレーション モードを開始します。IPv4 の場合、このコマンドはオプションです。

ステップ 4	**authentication key-chain** *key-chain* 例： `switch(config-router-af)# authentication key-chain routeKeys`	この VRF の EIGRP プロセスにキー チェーンを関連付けます。キー チェーン名は、大文字と小文字が区別される 20 文字以下の任意の英数字文字列にできます。
ステップ 5	**authentication mode md5** 例： `switch(config-router-af)# authentication mode md5`	この VRF の MD5 メッセージ ダイジェスト認証モードを設定します。

ステップ 6	**interface** *interface-type slot/port*
	例：
	`switch(config-router-af) interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。**?** を使用すると、サポートされているインターフェイスを調べることができます。

ステップ 7	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。
ステップ 8	**ip router eigrp** *instance-tag* 例： `switch(config-if)# ip router eigrp Test1`	このインターフェイスを、設定された EIGRP プロセスに関連付けます。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。
ステップ 9	**ip authentication key-chain eigrp** *instance-tag key-chain* 例： `switch(config-if)# ip authentication key-chain eigrp Test1 routeKey`	このインターフェイスの EIGRP プロセスにキー チェーンを関連付けます。この設定は、ルータの VRF モードで設定された認証設定よりも優先します。 インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。

	コマンドまたはアクション	目的
ステップ 10	**ip authentication mode eigrp** *instance-tag* **md5** 例： `switch(config-if)# ip authentication mode eigrp Test1 md5`	このインターフェイスのMD5メッセージダイジェスト認証モードを設定します。この設定は、ルータの VRF モードで設定された認証設定よりも優先します。 インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。
ステップ 11	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、EIGRP の MD5 メッセージ ダイジェスト認証をイーサネット インターフェイス 1/2 上で設定する例を示します。

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip router eigrp Test1
switch(config-if)# ip authentication key-chain eigrp Test1 routeKeys
switch(config-if)# ip authentication mode eigrp Test1 md5
switch(config-if)# copy running-config startup-config
```

EIGRP スタブ ルーティングの設定

ルータで EIGRP スタブ ルーティングを設定するには、アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
stub [**direct** | **receive-only** | **redistributed** [**direct**] **leak-map** *map-name*] 例： `switch(config-router-af)# eigrp stub redistributed`	リモート ルータを EIGRP スタブ ルータとして設定します。マップ名には最大 20 文字の英数字を使用できます。大文字と小文字は区別されます。

次に、直接接続され、再配布されるルートをアドバタイズするスタブ ルータを設定する例を示します。

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# address-family ipv4 unicast
```

```
switch(config-router-af)# stub direct redistributed
switch(config-router-af)# copy running-config startup-config
```

ルータがスタブ ルータとして設定されていることを確認するには、show ip eigrp neighbor detail コマンドを使用します。出力の最後の行は、リモート ルータまたはスポーク ルータのスタブ ステータスを示します。次に、**show ip eigrp neighbor detail** コマンドの出力例を示します。

```
Router# show ip eigrp neighbor detail
IP-EIGRP neighbors for process 201
H Address Interface Hold Uptime SRTT RTO Q Seq Type
(sec) (ms) Cnt Num
0 10.1.1.2 Se3/1 11 00:00:59 1 4500 0 7
Version 12.1/1.2, Retrans: 2, Retries: 0
Stub Peer Advertising ( CONNECTED SUMMARY) Routes
```

EIGRP のサマリー アドレスの設定

指定したインターフェイスにサマリー集約アドレスを設定できます。ルーティング テーブルに他にも個別のルートがある場合、EIGRP は、他の個別ルートすべての中で最小のメトリックと等しいメトリックで、サマリー アドレスをインターフェイスからアドバタイズします。ルート集約のセクションを参照してください。

サマリー集約アドレスを設定するには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
ip summary-address eigrp *instance-tag ip-prefix/length* [*distance* | **leak-map** *map-name*] 例： `switch(config-if)# ip summary-address eigrp Test1 192.0.2.0/8`	サマリー集約アドレスを、IP アドレスとネットワーク マスク、または IP プレフィックス/長さとして設定します。インスタンス タグおよびマップ名には最大 20 文字の英数字を使用できます。大文字と小文字は区別されます。 また、この集約アドレスのアドミニストレーティブ ディスタンスを設定することもできます。集約アドレスのデフォルト アドミニストレーティブ ディスタンスは 5 です。

次に、EIGRP によりネットワーク 192.0.2.0 がイーサネット 1/2 のみに集約されるようにする例を示します。

```
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip summary-address eigrp Test1 192.0.2.0 255.255.255.0
```

EIGRP へのルートの再配布

EIGRP 機能を有効にしていることを確認します（EIGRP 機能の有効化のセクションを参照）。

他のプロトコルから再配布されるルートには、メトリック（デフォルト メトリック設定オプションまたはルート マップによる）を設定する必要があります。

ルート マップを作成して、EIGRP に再配布されるルートのタイプを管理する必要があります。「Route Policy Manager の設定」を参照してください。

始める前に

他のルーティング プロトコルから EIGRP にルートを再配布できます。

手順の概要

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. **address-family ipv4 unicast**
4. **redistribute** { **bgp** *as* | { **eigrp** | **ospf** | **ospfv3** | rip } *instance-tag* | **direct** | **static** } **route-map** *name*
5. **default-metric** *bandwidth delay reliability loading mtu*
6. **show ip eigrp route-map statistics redistribute**
7. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router eigrp** *instance-tag*
	例：
	`switch(config)# router eigrp Test1`
	`switch(config-router)#`
		インスタンス タグを設定して、新しい EIGRP プロセスを作成します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。
		AS 番号であると認められていない *instance-tag* を設定する場合は、**autonomous-system** コマンドを使用して AS 番号を明示的に設定する必要があります。そうしないと、このEIGRPインスタンスはシャットダウン状態のままになります。

ステップ 3	**address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		アドレス ファミリ コンフィギュレーション モードを開始します。IPv4 の場合、このコマンドはオプションです。

ステップ 4	**redistribute** { **bgp** *as*	{ **eigrp**	**ospf**	**ospfv3**	rip } *instance-tag*	**direct**	**static** } **route-map** *name* 例： `switch(config-router-af)# redistribute bgp 100 route-map BGPFilter`	1 つのルーティング ドメインから EIGRP にルートを注入します。インスタンス タグおよびマップ名には最大 20 文字の英数字を使用できます。大文字と小文字は区別されます。
ステップ 5	**default-metric** *bandwidth delay reliability loading mtu* 例： `switch(config-router-af)# default-metric 500000 30 200 1 1500`	ルート再配布で学習したルートに割り当てられるメトリックを設定します。デフォルト値は次のとおりです。

	コマンドまたはアクション	目的
		• bandwidth：100000 kbps • delay：100（10 マイクロ秒単位） • reliability：255 • loading：1 • MTU：1492
ステップ 6	**show ip eigrp route-map statistics redistribute** 例： `switch(config-router-af)# show ip eigrp route-map statistics redistribute bgp`	EIGRP ルート マップ統計に関する情報を表示します。
ステップ 7	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、BGP を IPv4 向けの EIGRP に再配布する例を示します。

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# redistribute bgp 100 route-map BGPFilter
switch(config-router)# default-metric 500000 30 200 1 1500
switch(config-router)# copy running-config startup-config
```

再配布されるルート数の制限

ルートの再配布では、多くのルートを EIGRP ルート テーブルに追加できます。外部プロトコルから受け取るルートの数の上限を設定できます。EIGRP では、再配布されるルートの上限を設定するために次のオプションが用意されています。

- 上限固定：EIGRP が設定された最大値に達すると、メッセージをログに記録します。EIGRP は、それ以上の再配布されたルートを受け入れません。しきい値を超えたときに EIGRP が警告をログに記録する、最大値のしきい値に対する割合を設定することもできます。
- 警告のみ：EIGRP が最大値に達したときのみ、警告のログを記録します。EIGRP は、再配布されたルートを受け入れ続けます。
- 取り消し：EIGRP が最大値に達すると、タイムアウト期間が開始します。タイムアウト期間の経過後、再配布されたルートの現在数が最大数よりも少ない場合、EIGRP はすべての再配布されたルートを要求します。再配布されたルートの現在数が最大数に達した場合、EIGRP はすべての再配布されたルートを取り消します。EIGRP が再配布されたルートをさ

らに受け入れられるように、この条件をクリアする必要があります。任意で、タイムアウト期間を設定できます。

始める前に

EIGRP機能を有効にしていることを確認します（EIGRP機能の有効化のセクションを参照）。

手順の概要

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. **redistribute** { **bgp** *id* | **direct** | **eigrp** *id* | **ospf** *id* | **rip** *id* | **static** } **route-map** *map-name*
4. **redistribute maximum-prefix** *max* [*threshold*] [**warning-only** | **withdraw** [*num-retries timeout*]]
5. （任意） **show running-config eigrp**
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router eigrp** *instance-tag*
	例：
	`switch(config)# router eigrp Test1`
	`switch(config-router)#`
		インスタンス タグを設定して、新しい EIGRP プロセスを作成します。

ステップ 3	**redistribute** { **bgp** *id* | **direct** | **eigrp** *id* | **ospf** *id* | **rip** *id* | **static** } **route-map** *map-name* 例： `switch(config-router)# redistribute bgp route-map FilterExternalBGP`	設定したルートマップ経由で、選択したプロトコルを EIGRP に再配布します。
ステップ 4	**redistribute maximum-prefix** *max* [*threshold*] [**warning-only** | **withdraw** [*num-retries timeout*]] 例： `switch(config-router)# redistribute maximum-prefix 1000 75 warning-only`	EIGRP が再配布するプレフィックスの最大数を指定します。指定できる範囲は 0 ～ 65536 です。任意で次のオプションを指定します。 • *threshold*：警告メッセージをトリガーする最大プレフィックスの割合。 • **warning-only**：プレフィックスの最大数を超えた場合に警告メッセージを記録します。 • **withdraw**：再配布されたすべてのルートを取り消します。任意で再配布されたルートを取得し

	コマンドまたはアクション	目的
		ようと試みます。*num-retries* の範囲は 1 ～ 12 です。*timeout* は 60 ～ 600 秒です。デフォルトは 300 秒です。**clear ip eigrp redistribution** コマンドは、すべてのルートを取り消す場合に使用します。
ステップ 5	（任意） **show running-config eigrp** 例： `switch(config-router)# show running-config eigrp`	EIGRP の設定を表示します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、EIGRP に再配布されるルートの数を制限する例を示します。

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# redistribute bgp route-map FilterExternalBGP
switch(config-router)# redistribute maximum-prefix 1000 75
```

EIGRP でのロードバランスの設定

EIGRP でのロードバランスを設定できます。最大パス オプションを使用して、ECMP ルートの数を設定できます。

始める前に

EIGRP 機能がイネーブルにされていることを確認します。EIGRP 機能の有効化のセクションを参照してください。

手順の概要

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. **address-family ipv4 unicast**
4. **maximum-paths** *num-paths*
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router eigrp** *instance-tag*
	例：
	`switch(config)# router eigrp Test1`
	`switch(config-router)#`
		インスタンス タグを設定して、新しい EIGRP プロセスを作成します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。
		自律システム番号であると認められていない *instance-tag* を設定する場合は、**autonomous-system** コマンドを使用して自律システム番号を明示的に設定する必要があります。そうしないと、この EIGRP インスタンスはシャットダウン状態のままになります。

ステップ 3	**address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		アドレス ファミリ コンフィギュレーション モードを開始します。IPv4 の場合、このコマンドはオプションです。

ステップ 4	**maximum-paths** *num-paths* 例： `switch(config-router-af)# maximum-paths 5`	EIGRP がルート テーブルに受け入れる等コスト パスの数を設定します。指定できる範囲は 1 ～ 32 です。デフォルト値は 8 です。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、6 つまでの等コストパスによる、EIGRP の等コスト ロードバランスを IPv4 上で設定する例を示します。

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# maximum-paths 6
switch(config-router)# copy running-config startup-config
```

hello パケット間のインターバルとホールド タイムの調整

Hello メッセージの間隔とホールド タイムは調整できます。

デフォルトでは、5 秒ごとに Hello メッセージが送信されます。ホールド タイムは Hello メッセージでアドバタイズされ、ネイバーに、送信者が有効であると見なすべき時間を示します。デフォルトの保留時間は、hello 間隔の 3 倍（15 秒）です。

hello パケットの間隔を変更するには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
ip hello-interval eigrp *instance-tag seconds* 例： `switch(config-if)# ip hello-interval eigrp Test1 30`	EIGRP ルーティング処理の hello 間隔を設定します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。範囲は 1 ～ 65535 秒です。デフォルトは 5 分です。

非常に輻輳した大規模なネットワークでは、デフォルトの保留時間では、全ルータがネイバーから hello パケットを受信するまでに十分な時間がない場合もあります。この場合は、ホールド タイムを増やすことを推奨します。

ホールド タイムを変更するには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
ip hold-time eigrp *instance-tag seconds* 例： `switch(config-if)# ip hold-time eigrp Test1 30`	EIGRP ルーティング処理のホールド タイムを設定します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。有効な範囲は 1 ～ 65535 です。

タイマー設定を確認するには、**show ip eigrp interface detail** コマンドを使用します。

スプリット ホライズンの無効化

スプリットホライズンを使用すると、ルータによって情報元インターフェイスからルート情報がアドバタイズされないようにできます。通常はスプリットホライズンにより、特にリンクに障害がある場合に、複数のルーティング スイッチ間での通信が最適化されます。

デフォルトでは、スプリットホライズンはすべてのインターフェイスで有効になっています。

スプリット ホライズンを無効にするには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
no ip split-horizon eigrp *instance-tag* 例： `switch(config-if)# no ip split-horizon eigrp Test1`	スプリット ホライズンを無効にします。

EIGRP の調整

オプション パラメータを設定し、ネットワークに合わせて EIGRP を調整できます。

アドレス ファミリ コンフィギュレーション モードでは、次のオプション パラメータを設定できます。

コマンド	目的
default-information originate [**always** | **route-map** *map-name*] 例： `switch(config-router-af)# default-information originate always`	プレフィックス 0.0.0.0/0 を持つデフォルト ルートを発信するか、受け入れます。ルート マップが提供されると、ルート マップが true 状態となっている場合にのみデフォルト ルートが発信されます。マップ名には最大 20 文字の英数字を使用できます。大文字と小文字は区別されます。
distance *internal external* 例： `switch(config-router-af)# distance 25 100`	この EIGRP プロセスのアドミニストレーティブ ディスタンスを設定します。範囲は 1 ～ 255 です。内部の値で、同じ自律システム内で学習したルートのディスタンスが設定されます（デフォルト値は 90 です）。外部の値で、外部自律システムから学習したルートのディスタンスが設定されます（デフォルト値は 170 です）。
metric maximum-hops *hop-count* 例： `switch(config-router-af)# metric maximum-hops 70`	アドバタイズされるルートに許容される最大ホップ数を設定します。ホップ数がこの最大値を超えるルートは、到達不能としてアドバタイズされます。範囲は 1 ～ 255 です。デフォルトは 100 です。
metric weights *tos k1 k2 k3 k4 k5* 例： `switch(config-router-af)# metric weights 0 1 3 2 1 0`	EIGRP メトリックまたは K 値を調整します。EIGRP は次の式を使用して、ネットワークへの合計メトリックを決定します。 metric = [k1 X bandwidth + (k2 X bandwidth)/(256 – load) + k3 X delay] X [k5/(reliability + k4)] デフォルト値と指定できる範囲は、次のとおりです。 • TOS：0。指定できる範囲は 0 ～ 8 です。 • k1：1。有効な範囲は 0 ～ 255 です。 • k2：0。有効な範囲は 0 ～ 255 です。 • k3：1。有効な範囲は 0 ～ 255 です。 • k4：0。有効な範囲は 0 ～ 255 です。 • k5：0。有効な範囲は 0 ～ 255 です。

コマンド	目的
timers active-time { *time-limit* | **disabled** } 例： `switch(config-router-af)# timers active-time 200`	（照会の送信後に）ルートがアクティブ（SIA）状態のままとなっていることを宣言するまでに、ルータが待機する時間を分単位で設定します。有効な範囲は1～65535です。デフォルトは 3 です。

インターフェイス コンフィギュレーション モードで、省略可能な次のパラメータを設定できます。

コマンド	目的
ip bandwidth eigrp *instance-tag bandwidth* 例： `switch(config-if)# ip bandwidth eigrp Test1 30000`	インターフェイス上の EIGRP の帯域幅メトリックを設定します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。帯域幅の範囲は、1 ～ 2,560,000,000 kbps です。
ip bandwidth-percent eigrp *instance-tag percent* 例： `switch(config-if)# ip bandwidth-percent eigrp Test1 30`	EIGRP がインターフェイス上で使用する可能性のある帯域幅の割合を設定します。インスタンス タグには最大20文字の英数字を使用できます。大文字と小文字を区別します。 割合の範囲は 0 ～ 100 です。デフォルトは 50 です。
no ip delay eigrp *instance-tag delay* 例： `switch(config-if)# ip delay eigrp Test1 100`	インターフェイス上の EIGRP の遅延メトリックを設定します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。遅延の範囲は、1 ～ 16777215（10 マイクロ秒単位）です。
ip distribute-list eigrp *instance-tag* { **prefix-list** *name* | **route-map** *name*} { **in** | **out** } 例： `switch(config-if)# ip distribute-list eigrp Test1 route-map EigrpTest in`	このインターフェイス上の EIGRP のルータ フィルタリング ポリシーを設定します。インスタンス タグ、プレフィックス リスト名、およびルート マップ名には最大 20 文字の英数字を使用できます。大文字と小文字は区別されます。
no ip next-hop-self eigrp *instance-tag* 例： `switch(config-if)# ip next-hop-self eigrp Test1`	このインターフェイスのアドレスではなく、受信したネクストホップ アドレスを使用するよう、EIGRP を設定します。デフォルトでは、このインターフェイスの IP アドレスをネクストホップ アドレスに使用します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。

コマンド	目的
ip offset-list eigrp *instance-tag* { **prefix-list** *name* | **route-map** *name*} { **in** | **out** } *offset* 例： `switch(config-if)# ip offfset-list eigrp Test1 prefix-list EigrpList in`	EIGRP が学習したルートに、着信および発信メトリックへのオフセットを追加します。インスタンス タグ、プレフィックス リスト名、およびルート マップ名には最大 20 文字の英数字を使用できます。大文字と小文字は区別されます。
ip passive-interface eigrp *instance-tag* 例： `switch(config-if)# ip passive-interface eigrp Test1`	EIGRP hello を抑制します。これにより、EIGRP インターフェイス上でネイバーがルーティング アップデートを形成および送信することを防ぎます。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。

EIGRP の仮想化の設定

複数の VRF を作成して、各 VRF で同じまたは複数の EIGRP プロセスを使用することもできます。VRF にはインターフェイスを割り当てます。

（注） インターフェイスの VRF を設定した後に、インターフェイスの他のすべてのパラメータを設定します。インターフェイスの VRF を設定すると、そのインターフェイスの他の設定がすべて削除されます。

始める前に

EIGRP 機能がイネーブルにされていることを確認します（EIGRP 機能の有効化 （96 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **router eigrp** *instance-tag*
4. **interface ethernet slot/port**
5. **no switchport**
6. **vrf member** *vrf-name*
7. **ip router eigrp** *instance-tag*
8. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context RemoteOfficeVRF`
	`switch(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。VRN 名には最大 20 文字の英数字を使用できます。大文字と小文字は区別されます。

ステップ 3	**router eigrp** *instance-tag*
	例：
	`switch(config)# router eigrp Test1`
	`switch(config-router)#`
		インスタンス タグを設定して、新しい EIGRP プロセスを作成します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。
		AS 番号であると認められていない *instance-tag* を設定する場合は、**autonomous-system** コマンドを使用して AS 番号を明示的に設定する必要があります。そうしないと、この EIGRP インスタンスはシャットダウン状態のままになります。

ステップ 4	**interface ethernet slot/port**
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		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。**?** を使用すると、スロットおよびポートの範囲を検索できます。

ステップ 5	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。
ステップ 6	**vrf member** *vrf-name* 例： `switch(config-if)# vrf member RemoteOfficeVRF`	このインターフェイスを VRF に追加します。VRF 名には最大 20 文字の英数字を使用できます。大文字と小文字は区別されます。
ステップ 7	**ip router eigrp** *instance-tag* 例： `switch(config-if)# ip router eigrp Test1`	このインターフェイスを EIGRP プロセスに追加します。インスタンス タグには最大 20 文字の英数字を使用できます。大文字と小文字を区別します。
ステップ 8	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、VRF を作成して、その VRF にインターフェイスを追加する例を示します。

```
switch# configure terminal
switch(config)# vrf context NewVRF
switch(config-vrf)# router eigrp Test1
switch(config-router)# i nterface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip router eigrp Test1
switch(config-if)# vrf member NewVRF
switch(config-if)# copy running-config startup-config
```

EIGRP の設定の確認

EIGRP の設定情報を表示するには、次のいずれかの作業を行います。

コマンド	目的
show ip eigrp [*instance-tag*]	設定した EIGRP プロセスの要約を表示します。
show ip eigrp [*instance-tag*] **interfaces** [*type number*] [**brief**] [**detail**]	設定されているすべての EIGRP インターフェイスに関する情報を表示します。
show ip eigrp *instance-tag* **neighbors** [*type number*]	すべての EIGRP ネイバーに関する情報を表示します。EIGRP ネイバーの設定を確認するには、このコマンドを使用します。
show ip eigrp [*instance-tag*] **route** [*ip-prefix/length*] [**active**] [**all-links**] [**detail-links**] [**pending**] [**summary**] [**zero-successors**] [**vrf** *vrf-name*]	すべての EIGRP ルートに関する情報を表示します。
show ip eigrp [*instance-tag*] **topology** [*ip-prefix/length*] [**active**] [**all-links**] [**detail-links**] [**pending**] [**summary**] [**zero-successors**] [**vrf** *vrf-name*]	EIGRP トポロジ テーブルに関する情報を表示します。
show running-configuration eigrp	現在実行中の EIGRP コンフィギュレーションを表示します。

EIGRP 統計情報の表示

EIGRP 統計情報を表示するには、次のコマンドを使用します。

コマンド	目的
show ip eigrp [*instance-tag*] **accounting** [**vrf** *vrf-name*]	EIGRP の課金統計情報を表示します。
show ip eigrp [*instance-tag*] **route-map statistics redistribute**	EIGRP の再配布統計情報を表示します。
show ip eigrp [*instance-tag*] **traffic** [**vrf** *vrf-name*]	EIGRP のトラフィック統計情報を表示します。

EIGRP の設定例

次に、EIGRP を設定する例を示します。

```
feature eigrp
interface ethernet 1/2
no switchport
ip address 192.0.2.55/24
ip router eigrp Test1
no shutdown
router eigrp Test1
router-id 192.0.2.1
```

関連項目

ルートマップの詳細については、Route Policy Manager の設定 （281 ページ）を参照してください。

その他の参考資料

EIGRP の実装に関する詳細情報については、次のページを参照してください。

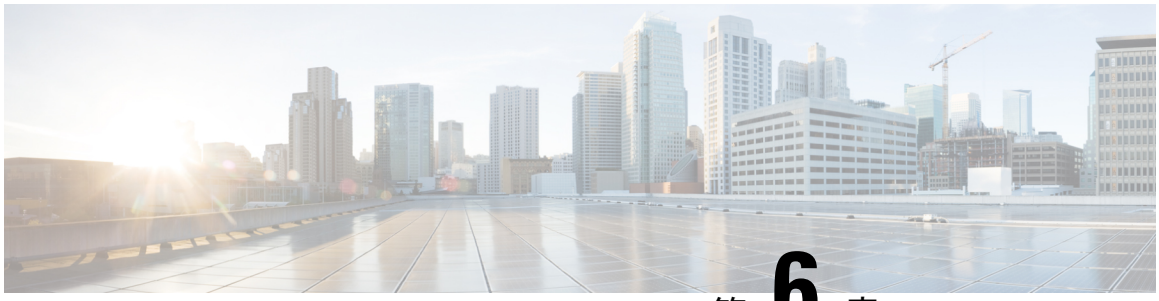
- 関連資料
- MIB

関連資料

関連項目	マニュアルタイトル
EIGRP CLI コマンド	『Cisco Nexus 3000 Series Command Reference』
https://www.cisco.com/c/en/us/support/docs/ip/enhanced-interior-gateway-routing-protocol-eigrp/16406-eigrp-toc.html?dtid=osscdc000283	『Introduction to EIGRP Tech Note』
http://www.cisco.com/en/US/tech/tk365/technologies_q_and_a_item09186a008012dac4.shtml	EIGRP よく寄せられる質問（FAQ）

MIB

MIB	MIB のリンク
CISCO-EIGRP-MIB	MIB を検索してダウンロードするには、次の MIB ロケータに移動します。



第 6 章

基本的 BGP の設定

この章では、Cisco NX-OS スイッチでボーダー ゲートウェイ プロトコル（BGP）を設定する方法について説明します。

この章は、次の項で構成されています。

ベーシック BGP の概要

Cisco NX-OS は BGP バージョン 4 をサポートします。BGP v4 に組み込まれているマルチプロトコル拡張機能を使用すると、IP ルートおよび複数のレイヤ 3 プロトコル アドレス ファミリに関するルーティング情報を BGP に伝送させることができます。BGP では、他の BGP 対応スイッチとの間で TCP セッションを確立するための、信頼できるトランスポート プロトコルとして TCP を使用します。

BGP ではパスベクトル ルーティング アルゴリズムを使用して、BGP 対応ネットワーク スイッチまたは BGP スピーカ間でルーティング情報を交換します。各 BGP スピーカはこの情報を使用して、特定の宛先までのパスを判別し、なおかつルーティング ループを伴うパスを検出して回避します。ルーティング情報には、宛先の実際のルート プレフィックス、宛先に対する自律システムのパス、およびその他のパス属性が含まれます。

BGPはデフォルトで、宛先ホストまたはネットワークへのベストパスとして、1つだけパスを選択します。各パスは、BGP ベストパス分析で使用される well-known mandatory、well-known discretionary、optional transitive の各属性を伝送します。BGP ポリシーを設定し、これらの属性の一部を変更することによって、BGP パス選択を制御できます。詳細については、ルート ポリシーおよび BGP セッションのリセットのセクションを参照してください。

BGP は、ロード バランシングまたは等コスト マルチパス（ECMP）もサポートします。詳細については、「ロード シェアリングおよびマルチパス」の項を参照してください。

BGP 自律システム

自律システム（AS）とは、単一の管理エンティティにより制御されるネットワークです。自律システムは 1 つまたは複数の IGP および整合性のある一連のルーティング ポリシーを使用して、ルーティング ドメインを形成します。BGP は 16 ビットおよび 32 ビットの自律システム番号をサポートします。詳細については、「自律システム」を参照してください。

個々の BGP 自律システムは外部 BGP（eBGP）ピアリング セッションを通じて、ルーティング情報をダイナミックに交換します。同じ自律システム内の BGP スピーカは、内部 BGP（iBGP）を通じて、ルーティング情報を交換できます。

4 バイトの AS 番号のサポート

BGP では、2 バイトまたは 4 バイトの AS 番号をサポートしています。Cisco NX-OS は、プレーンテキスト表記で 4 バイト（つまり 32 ビットの整数）の AS 番号を表示します。4 バイトの AS 番号は、プレーンテキスト表記（たとえば 1 ～ 4294967295）または AS ドット表記（たとえば 1.0）で設定できます。詳細については、「自律システム」を参照してください。

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、ルーティング情報源の信頼性を示す評価基準です。デフォルトでは、BGP は表に示されたアドミニストレーティブ ディスタンスを使用します。

表 *7:* デフォルトの *BGP* アドミニストレーティブ ディスタンス

ディスタンス	デフォルト値	機能
外部	20	eBGP から学習したルートに適用されます。
内部	200	iBGP から学習したルートに適用されます。
ローカル	200	ルータを起点とするルートに適用されます。

（注） アドミニストレーティブ ディスタンスが BGP パス選択アルゴリズムに影響を与えることはありませんが、BGP で学習されたルートが IP ルーティング テーブルに組み込まれるかどうかを左右します。

詳細については、「アドミニストレーティブ ディスタンス」のセクションを参照してください。

BGP ピア

BGP スピーカーは他の BGP スピーカーを自動的に検出しません。ユーザ側で BGP スピーカ間の関係を設定する必要があります。BGP ピアは、別の BGP スピーカへのアクティブな TCP 接続を持つ BGP スピーカです。

BGP セッション

BGP は TCP ポート 179 を使用して、ピアとの TCP セッションを作成します。ピア間で TCP 接続が確立されると、各 BGP ピアは最初に相手と、それぞれのすべてのルートを交換し、BGP ルーティング テーブルを完成させます。初期交換以後、BGP ピアはネットワーク トポロジが変化したとき、またはルーティング ポリシーが変更されたときに、差分アップデートだけを送信します。更新と更新の間の非アクティブ期間には、ピアは「キープアライブ」と呼ばれる特別なメッセージを交換します。ホールド タイムは、は、次の BGP アップデートまたはキープアライブ メッセージを受信するまでに経過することが許容される、最大時間限度です。

Cisco NX-OS では、次のピア設定オプションをサポートしています。

- 個別の IPv4：BGP は、リモート アドレスと AS 番号が一致する BGP スピーカとのセッションを確立します。
- 単一 AS 番号の IPv4 プレフィックス ピア：BGP は、プレフィックスおよび AS 番号が一致する BGP スピーカとのセッションを確立します。
- ダイナミック AS 番号プレフィックス ピア：BGP は、プレフィックスと、設定済み AS 番号のリストに載っている AS 番号と一致する BGP スピーカとのセッションを確立します。

プレフィックス ピアのダイナミック AS 番号

Cisco NX-OS では、BGP セッションを確立する AS 番号の範囲またはリストを受け入れます。たとえば IPv4 プレフィックス 192.0.2.0/8 および AS 番号 33、66、99 を使用するように BGP を設定する場合、BGP は 192.0.2.1 および AS 番号 66 を使用してセッションを確立しますが、192.0.2.2 および AS 番号 50 からのセッションは拒否します。

Cisco NX-OS では、セッションが確立されるまで内部 BGP（iBGP）または外部 BGP（eBGP）セッションとして、プレフィックス ピアをダイナミック AS 番号と関連付けません。iBGP および eBGP の詳細については、を参照してください。

（注） ダイナミック AS 番号プレフィックス ピア設定は、BGP テンプレートから継承した個々の AS 番号の設定よりも優先します。詳細については、 の章を参照してください。

BGP ルータ ID

ピア間で BGP セッションを確立するには、BGP セッションの確立時に、OPEN メッセージで BGP ピアに送信されるルータ ID を BGP に設定する必要があります。BGP ルータ ID は 32 ビット値であり、IPv4 アドレスで表すことがよくあります。ルータ ID はユーザ側で設定できます。デフォルトでは、Cisco NX-OS によって、ルータのループバック インターフェイスの IPv4 アドレスにルータ ID が設定されます。ルータ上でループバック インターフェイスが設定されていない場合は、BGP ルータ ID を表すためにルータ上の物理インターフェイスに設定されている最上位の IPv4 アドレスがソフトウェアによって選択されます。BGP ルータ ID は、ネットワーク内の BGP ピアごとに一意である必要があります。

BGP にルータ ID が設定されていない場合、BGP ピアとのピアリング セッションを確立できません。

BGP パスの選択

BGP は複数の送信元から、同じルートのアドバタイズメントを受信する可能性がありますが、BGP はベスト パスとして、パスを 1 つだけ選択します。BGP は、そのパスを IP ルーティング テーブルに格納し、ピアにパスを伝達します。

所定のネットワークでパスが追加または削除されるたびに、ベストパス アルゴリズムが実行されます。ベストパス アルゴリズムは、ユーザが BGP 設定を変更した場合にも実行されます。BGP は所定のネットワークで使用できる一連の有効パスの中から、最適なパスを選択します。

Cisco NX-OS は次の手順で、BGP ベストパス アルゴリズムを実行します。

1. **ステップ 1**：2 つのパスを比較し、どちらが適切かを判別します（ステップ 1：パス ペアの比較のセクションを参照してください）。
2. **ステップ 2**：すべてのパスについて繰り返し、全体として最適なパスを選択するためにパスを比較する順序を決定します（ステップ 2：比較順序の決定のセクションを参照してください）。
3. **ステップ 3**：.新しいベスト パスを使用するに値するだけの差が新旧のベスト パスにあるかどうかを判別します（ステップ 3：ベスト パス変更の抑制の決定のセクションを参照してください）。

（注） 重要なのは、パート 2 で決定される比較順序です。3 つのパス A、B、C があり、Cisco NX-OS が A と B を比較して A を選択し、Cisco NX-OS が B と C を比較して B を選択したとします。しかし、Cisco NX-OS が A と C を比較したときには、A を選択しないかもしれません。これは一部の BGP メトリックが同じネイバー自律システムからのパスだけに適用され、すべてのパスにわたっては適用されないからです。

パス選択には、BGP AS パス属性が使用されます。AS パス属性には、アドバタイズされたパスでたどる自律システム番号（AS 番号）のリストが含まれます。BGP 自律システムを自律システムの集合または連合に細分化する場合は、AS パスにローカル定義の自律システムを指定した連合セグメントが含まれます。

ステップ 1：パス ペアの比較

BGP ベストパス アルゴリズムの最初のステップでは、より適切なパスを判別するために 2 つのパスを比較します。次に、Cisco NX-OS が 2 つのパスを比較して、より適切なパスを判別する基本的なステップについて説明します。

1. Cisco NX-OS は、比較する有効なパスを選択します（たとえば、到達不能なネクストホップがあるパスは無効です）。
2. Cisco NX-OS は、重み値が最大のパスを選択します。
3. Cisco NX-OS は、ローカル プリファレンスが最大のパスを選択します。
4. パスの一方がローカル起点の場合、Cisco NX-OS はそのパスを選択します。
5. Cisco NX-OS は、AS パスが短い方のパスを選択します。

（注） AS パス長を計算するときに、Cisco NX-OS は連合セグメントを無視し、AS セットを 1 として数えます。詳細については、「AS 連合」の項を参照してください。

6. Cisco NX-OS は、オリジンが低い方のパスを選択します。IGP は EGP よりも低いと見なされます。
7. Cisco NX-OS は、multi exit discriminator（MED）が小さい方のパスを選択します。

   このステップが実行されるされないを左右する、一連のオプションを選択できます。Cisco NX-OS が両方のパスの MED を比較するのは、通常、同じ自律システムのピアからそれらのパスを受け取った場合です。それ以外の場合、Cisco NX-OS は MED の比較を省略します。

   パスのピア自律システムに関係なく、ベストパス アルゴリズムの MED 比較が必ず実行されるように、Cisco NX-OS を設定することもできます。詳細については、「ベストパス アルゴリズムの調整」を参照してください。この設定を行わなかった場合、MED 比較が実行されるかどうかは、次のように比較する 2 つのパスの AS パス属性によって決まります。

1. パスに AS パスがない、 または AS_SET から始まる AS パスがある場合、パスは内部であり、Cisco NX-OS は他の内部パスに対して MED を比較します。
2. AS パスが AS_SEQUENCE から始まる場合、ピア自律システムがシーケンスで最初の AS 番号になり、Cisco NX-OS は同じピア自律システムを持つ他のパスに対して MED を比較します。
3. AS パス に連合セグメントだけが含まれている場合、または連合セグメントで始まり、AS_SET が続いている場合、パスは内部であり、Cisco NX-OS は他の内部パスに対して MED を比較します。
4. AS パスが連合セグメントで始まり、AS_SEQUENCE が続いている場合、ピア自律システムが AS_SEQUENCE で最初の AS 番号になり、Cisco NX-OS は同じピア自律システムを持つ他のパスに対して MED を比較します。

（注） Cisco NX-OS がパスで指定された MED 属性を受信しなかった場合、Cisco NX-OS は欠落 MED が使用可能な最大値になるようにユーザーがベストパス アルゴリズムを設定していない限り、MED を 0 と見なします。詳細については、「ベストパス アルゴリズムの調整」を参照してください。

5. 非決定性の MED 比較機能がイネーブルの場合、ベスト パス アルゴリズムでは Cisco IOS スタイルの MED 比較が使用されます。詳細については、「ベストパス アルゴリズムの調整」を参照してください。

8. 一方のパスが内部ピアから、他方のパスが外部ピアからの場合、Cisco NX-OS は外部ピアからのパスを選択します。
9. ネクストホップ アドレスへの IGP メトリックが異なるパスの場合、Cisco NX-OS は IGP メトリックが小さい方のパスを選択します。
10. Cisco NX-OS は、最後に実行したベストパス アルゴリズムによって選択されたパスを使用します。

    ステップ 1 ～ 9 のすべてのパス パラメータが同じ場合、ルータ ID を比較するようにベストパス アルゴリズムを設定できます。詳細については、「ベストパス アルゴリズムの調整」を参照してください。パスに発信元属性が含まれている場合、Cisco NX-OS はその属性をルータ ID として使用して比較します。発信もと属性が含まれていない場合、Cisco NX-OS はパスを送信したピアのルータ ID を使用します。パス間でルータ ID が異なる場合、Cisco NX-OS はルータ ID が小さい方のパスを選択します。

（注） 属性の送信元をルータ ID として使用する場合は、2 つのパスに同じルータ ID を設定することができます。また、同じピア ルータとの 2 つの BGP セッションが可能です。したがって、同じルータ ID で 2 つのパスを受信できます。

11. Cisco NX-OS は、クラスタ長が短いほうのパスを選択します。クラスタ リスト属性の指定されたパスを受け取らなかった場合、クラスタ長は 0 です。
12. Cisco NX-OS は、IP アドレスが小さい方のピアから受信したパスを選択します。ローカル発生のパス（再配布のパスなど）は、ピア IP アドレスが 0 になります。

（注） ステップ 9 以降が同じパスは、マルチパスを設定している場合、マルチパスに使用できます。詳細については、「ロード シェアリングおよびマルチパス」の項を参照してください。

ステップ 2：比較順序の決定

BGP ベストパス アルゴリズム実装の 2 番めのステップでは、Cisco NX-OS がパスを比較する順序を決定します。

1. Cisco NX-OS は、パスをグループに分けます。各グループ内で、Cisco NX-OS はすべてのパスにわたって MED を比較します。Cisco NX-OSは、ステップ 1：パス ペアの比較と同じルールを使用して、2 つのパス間で MED を比較できるかどうかを判断します。この比較では通常、ネイバー自律システムごとに 1 つずつグループが選択されます。**bgp bestpath med always** コマンドを設定すると、Cisco NX-OS はすべてのパスが含まれた 1 グループだけを選択します。
2. Cisco NX-OS は、常に最適な方を維持しながら、グループのすべてのパスを反復することによって、各グループのベスト パスを決定します。Cisco NX-OS は、各パスをそれまでの一時的なベスト パスと比較します。それまでのベスト パスよりも適切な場合は、そのパスが新しく一時的なベスト パスになり、Cisco NX-OS はグループの次のパスと比較します。
3. Cisco NX-OS は、ステップ 2 の各グループで選択されたベスト パスからなる、パス セットを形成します。Cisco NX-OS は、このパス セットでもステップ 2 と同様にそれぞれの比較を繰り返すことによって、全体としてのベスト パスを選択します。

ステップ 3：ベスト パス変更の抑制の決定

実装の次のパートでは、Cisco NX-OS が新しいベスト パスを使用するのか抑制するのかを決定します。新しいベスト パスが古いパスとまったく同じ場合、ルータは引き続き既存のベスト パスを使用できます（ルータ ID が同じ場合）。Cisco NX-OS では引き続き既存のベスト パスを使用することによって、ネットワークにおけるルート変更を回避できます。

抑制機能をオフにするには、ルータ ID を比較するようにベスト パス アルゴリズムを設定します。詳細については、「ベストパス アルゴリズムの調整」を参照してください。この機能を設定すると、新しいベスト パスが常にに既存のベスト パスよりも優先されます。

次の条件が発生した場合に、ベスト パス変更を抑制できません。

- 既存のベスト パスが無効になった。

- 既存または新しいベストパスを内部（または連合）ピアから受信したか、またはローカルに発生した（再配布などによって）。
- 同じピアからパスを受信した（パスのルータ ID が同じ）。
- パス間で重み値、ローカル プリファレンス、オリジン、またはネクストホップ アドレスに対する IGP メトリックが異なっている。
- パス間で MED が異なっている。

BGP およびユニキャスト RIB

BGP はユニキャスト RIB（ルーティング情報ベース）と通信して、ユニキャスト ルーティング テーブルに IPv4 ルートを格納します。ベストパスの選択後、ベストパスの変更をルーティング テーブルに反映させる必要があると BGP が判別した場合、BGP はユニキャスト RIB にルート アップデートを送信します。

BGP はユニキャスト RIB における BGP ルートの変更に関して、ルート通知を受け取ります。さらに、再配布をサポートする他のプロトコル ルートに関するルート通知を受け取ります。

BGP はネクストホップの変更に関する通知も、ユニキャスト RIB から受け取ります。BGP はこれらの通知を使用して、ネクストホップ アドレスへの到達可能性および IGP メトリックを追跡します。

ユニキャスト RIB でネクストホップ到達可能性または IGP メトリックが変更されるたびに、BGP は影響を受けるルートについて、ベスト パス再計算を開始させます。

BGP の前提条件

BGP を使用するには、次の前提条件を満たしている必要があります。

- BGP機能を有効にする必要があります（BGP機能のイネーブル化のセクションを参照）。
- システムに有効なルータ ID を設定しておく必要があります。
- Regional Internet Registry（RIR）によって割り当てられたか、またはローカル管理の AS 番号を取得しておく必要があります。
- 再帰ネクストホップ解決に対応できる IGP を 1 つ以上設定する必要があります。
- BGP セッションを確立するネイバー環境で、アドレス ファミリを設定する必要があります。

BGP に関する注意事項と制約事項

BGP 設定時の注意事項および制約事項は、次のとおりです。

- ダイナミック AS 番号プレフィックス ピア設定は、BGP テンプレートから継承した個々の AS 番号の設定よりも優先します。
- AS 連合でプレフィックス ピアにダイナミック AS 番号を設定した場合、BGP はローカル連合の AS 番号のみでセッションを確立します。
- ダイナミック AS 番号プレフィックス ピアで作成された BGP セッションは、設定済みの eBGP マルチホップ存続可能時間（TTL）値や直接接続ピアに対するディセーブル済みのチェックを無視します。
- ルータ ID の自動変更およびセッション フラップを避けるために、BGP 用のルータ ID を設定する必要があります。
- ピアごとに最大プレフィックス設定オプションを使用し、受信するルート数および使用するシステム リソース数を制限する必要があります。
- update-source を設定し、BGP/eBGP マルチホップ セッションでセッションを確立する必要があります。
- 再配布を設定する場合、BGP ポリシーを指定する必要があります。
- VRF 内で BGP ルータ ID を定義する必要があります。
- キープアライブおよびホールド タイマーの値を小さくすると、BGP セッション フラップが発生する可能性があります。
- VRF を設定する場合には、望ましい VRF を入力します。

CLI コンフィギュレーション モード

以下の項では、BGP に対応する各 CLI コンフィギュレーション モードの開始方法について説明します。現行のモードで **?** コマンドを入力すると、そのモードで使用可能なコマンドを表示できます。

グローバル コンフィギュレーション モード

グローバル コンフィギュレーション モードは、BGP プロセスを作成したり、AS 連合、ルート ダンプニングなどの拡張機能を設定したりする場合に使用します。詳細については、高度な BGP の設定を参照してください。

次に、ルータ コンフィギュレーション モードを開始する例を示します。

```
switch# configuration
switch(config)# router bgp 64496
switch(config-router)#
```

BGP は仮想ルーティングおよび転送（VRF）をサポートします。ネットワークで VRF を使用する場合は、適切な VRF 内で BGP を設定できます。設定の詳細については、「仮想化の設定」の項を参照してください。

次に、VRF コンフィギュレーション モードを開始する例を示します。

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
switch(config-router-vrf)#
```

アドレス ファミリ設定モード

任意で、BGP がサポートするアドレス ファミリを設定できます。アドレス ファミリ用の機能を設定する場合は、ルータ設定モードで **address-family** コマンドを使用します。ネイバーに対応する特定のアドレス ファミリを設定する場合は、ネイバー設定モードで **address-family** コマンドを使用します。

ルート再配布、アドレス集約、ロードバランシングなどの拡張機能を使用する場合は、アドレス ファミリを設定する必要があります。

次に、ルータ設定モードからアドレス ファミリ設定モードを開始する例を示します。

```
switch(config)# router bgp 64496
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)#
```

次に、VRF を使用している場合に、VRF アドレス ファミリ設定モードを開始する例を示します。

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
switch(config-router-vrf)# address-family ipv4 unicast
switch(config-router-vrf-af)#
```

ネイバー コンフィギュレーション モード

Cisco NX-OS には、BGP ピアを設定するためのネイバー コンフィギュレーション モードがあります。ネイバー コンフィギュレーション モードを使用して、ピアのあらゆるパラメータを設定できます。

次に、ネイバー コンフィギュレーション モードを開始する例を示します。

```
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.1
switch(config-router-neighbor)#
```

次に、VRF ネイバー コンフィギュレーション モードを開始する例を示します。

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
switch(config-router-vrf)# neighbor 192.0.2.1
switch(config-router-vrf-neighbor)#
```

ネイバー アドレス ファミリ コンフィギュレーション モード

アドレス ファミリ固有のネイバー設定を入力し、ネイバーのアドレス ファミリをイネーブルにするには、ネイバー コンフィギュレーション サブモード内のアドレス ファミリ コンフィギュレーション サブモードを使用できます。このモードは、所定のネイバーに認められるプレフィックス数の制限、eBGP のプライベート AS 番号の削除といった拡張機能に使用します。

次に、ネイバー アドレス ファミリ コンフィギュレーション モードを開始する例を示します。

```
switch(config)# router bgp 64496
switch(config-router# neighbor 192.0.2.1
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)#
```

次に、VRF ネイバー アドレス ファミリ コンフィギュレーション モードを開始する例を示します。

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
switch(config-router-vrf)# neighbor 209.165.201.1
switch(config-router-vrf-neighbor)# address-family ipv4 unicast
switch(config-router-vrf-neighbor-af)#
```

BGP のデフォルト設定

次の表に、BGP パラメータのデフォルト設定値を示します。

表 *8:* デフォルトの *BGP* パラメータ

パラメータ	デフォルト
BGP 機能	ディセーブル
キープアライブ インターバル	60 秒
ホールド タイマー	180 秒

基本的 BGP の設定

ベーシック BGP を設定するには、BGP をイネーブルにして、BGP ピアを設定する必要があります。ベーシック BGP ネットワークの設定は、いくつかの必須作業と多数の任意の作業からなります。BGP ルーティング プロセスおよび BGP ピアの設定は必須です。

BGP 機能のイネーブル化

始める前に

BGP を設定する前に、BGP 機能をイネーブルにする必要があります。

手順の概要

1. **configure terminal**
2. **feature bgp**

3. （任意） **show feature**
4. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**feature bgp** 例： `switch(config)# feature bgp`	BGP 機能をイネーブルにします。
ステップ 3	（任意） **show feature** 例： `switch(config)# show feature`	有効および無効にされた機能を表示します。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

no feature bgp コマンドを使用して、BGP機能をディセーブルにし、関連するコンフィギュレーションをすべて削除します。

コマンド	目的
no feature bgp 例： `switch(config)# no feature bgp`	BGP機能をディセーブルにして、関連するすべての設定を削除します。

BGP インスタンスの作成

BGP インスタンスを作成し、BGP インスタンスにルータ ID を割り当てることができます。「BGP ルータ ID」のセクションを参照してください。Cisco NX-OS は、2 バイトまたは 4 バイトのプレーンテキスト表記またはAS ドット表記による自律システム（AS）番号をサポートします。詳細については、4 バイトの AS 番号のサポートのセクションを参照してください。

始める前に

BGP 機能を有効にしていることを確認します（BGP 機能のイネーブル化のセクションを参照してください）。

BGP はルータ ID（設定済みループバック アドレスなど）を取得できなければなりません。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. （任意） **router-id** *ip-address*
4. （任意） **address-family ipv4 unicast**
5. （任意） **network** *ip-prefix* [**route-map** *map-name*]
6. （任意） **show bgp all**
7. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 3	（任意） **router-id** *ip-address* 例： `switch(config-router)# router-id 192.0.2.255`	BGP ルータ ID を設定します。この IP アドレスによって、この BGP スピーカを特定します。このコマンドによって、BGP ネイバーセッションの自動通知およびセッション リセットが開始されます。

ステップ 4	（任意） **address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		指定されたアドレス ファミリに対応するグローバル アドレス ファミリ コンフィギュレーション モードを開始します。このコマンドによって、すべての BGP ネイバーセッションの自動通知およびセッション リセットが開始されます。

ステップ 5	（任意） **network** *ip-prefix* [**route-map** *map-name*] 例： `switch(config-router-af)# network 192.0.2.0`	ネットワークを、この自律システムに対してローカルに設定し、BGP ルーティング テーブルに追加します。 エクステリア プロトコルの場合、network コマンドでアドバタイズするネットワークを制御します。イ

	コマンドまたはアクション	目的
		ンテリア プロトコルでは、network コマンドを使用して、アップデートの送信先を決定します。
ステップ 6	（任意） **show bgp all** 例： `switch(config-router-af)# show bgp all`	すべての BGP アドレス ファミリに関する情報を表示します。
ステップ 7	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

BGP プロセスおよび関連するすべての設定を削除するには、**no router bgp** コマンドを使用します。

コマンド	目的
no router bgp *autonomous-system-number* 例： `switch(config)# no router bgp 201`	BGPプロセスおよび関連する設定を削除します。

次に、IPv4 ユニキャスト アドレス ファミリを指定して BGP をイネーブルに設定し、アドバタイズするネットワークを 1 つ追加する例を示します。

```
switch# configure terminal
switch(config)# router bgp 64496
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# network 192.0.2.0
switch(config-router-af)# copy running-config startup-config
```

BGP インスタンスの再起動

BGP インスタンスを再起動し、そのインスタンスのすべてのピア セッションをクリアできます。

BGPインスタンスを再起動し、関連付けられたすべてのピアを削除するには、次のコマンドを使用します。

コマンド	目的
restart bgp *instance-tag* 例： `switch(config)# restart bgp 201`	BGP インスタンスを再起動し、すべてのピアリング セッションをリセットまたは再確立します。

BGP のシャットダウン

BGP プロトコルをシャットダウンして BGP を正常にディセーブルし、設定を保持できます。

BGP をシャットダウンするには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
shutdown 例： `switch(config-router)# shutdown`	BGPを正常にシャットダウンします。

BGP ピア設定

BGP プロセス内で BGP ピアを設定できます。BGP ピアごとに、関連付けられたキープアライブ タイマーとホールド タイマーがあります。これらのタイマーは、グローバルに設定することも、BGP ピアごとに設定することもできます。ピア設定はグローバル設定を上書きします。

（注） ピアごとに、ネイバー コンフィギュレーション モードでアドレス ファミリを設定する必要があります。

始める前に

BGP 機能を有効にしていることを確認します（BGP 機能のイネーブル化のセクションを参照してください）。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor** *ip-address* { **ipv4** } **remote-as as-number**
4. （任意） **description** *text*
5. （任意） **timers keepalive**-*time* **hold**-*time*
6. （任意） **shutdown**
7. **address-family ipv4 unicast**

8. （任意） **show bgp ipv4 unicast neighbors**
9. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 3	**neighbor** *ip-address* { **ipv4** } **remote-as as-number**
	例：
	`switch(config-router)# neighbor 209.165.201.1 remote-as 64497`
	`switch(config-router-neighbor)#`
		リモート BGP ピアの指定タイプと AS 番号を設定します。The *ip-address* 形式は x.x.x.x です。

ステップ 4	（任意） **description** *text*
	例：
	`switch(config-router-neighbor)# description Peer Router B`
	`switch(config-router-neighbor)#`
		ネイバーの説明を追加します。最大 80 文字の英数字ストリングを使用できます。

ステップ 5	（任意） **timers keepalive**-*time* **hold**-*time* 例： `switch(config-router-neighbor )# timers 30 90`	ネイバーのキープアライブおよびホールドタイムを表す BGP タイマー値を追加します。指定できる範囲は 0 ～ 3600 秒です。デフォルトは、キープアライブ タイムで 60 秒、ホールド タイムで 180 秒です。
ステップ 6	（任意） **shutdown** 例： `switch(config-router-neighbor)# shutdown`	この BGP ネイバーを管理目的でシャットダウンします。このコマンドによって、BGP ネイバー セッションの自動通知およびセッション リセットが開始されます。

ステップ 7	**address-family ipv4 unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		ユニキャスト指定のアドレス ファミリーに対応したネイバー アドレス ファミリー構成モードを開始します。

ステップ 8	（任意） **show bgp ipv4 unicast neighbors** 例：	BGP ピアに関する情報を表示します。

	コマンドまたはアクション	目的
	`switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors`	
ステップ 9	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、BGP ピアを設定する例を示します。

```
switch# configure terminal
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.1 remote-as 64497
switch(config-router-neighbor)# description Peer Router B
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# copy running-config startup-config
```

プレフィックス ピアのダイナミック AS 番号の設定

BGP プロセス内で複数の BGP ピアを設定できます。BGP セッションの確立をルート マップの単一の AS 番号または複数の AS 番号に制限できます。

プレフィックス ピアのダイナミック AS 番号を介して設定された BGP セッションは、**ebgp-multihop** コマンドおよび **disable-connected-check** コマンドを無視します

ルートマップの AS 番号のリストは変更できますが、ルートマップ名を変更するには **no neighbor** コマンドを使用する必要があります。設定されたルート マップの AS 番号に変更を加えた場合、新しいセッションのみに影響します。

始める前に

BGP 機能が有効になっていることを確認します

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor** *prefix* **remote-as route-map** *map-name*
4. **show bgp ipv4 unicast neighbors**
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 3	**neighbor** *prefix* **remote-as route-map** *map-name*
	例：
	`switch(config-router)# neighbor 192.0.2.0/8 remote-as routemap BGPPeers`
	`switch(config-router-neighbor)#`
		IPv4 プレフィックス、およびリモート BGP ピアの受け付けられた AS 番号のリストのルート マップを構成します。
		map-name には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。

ステップ 4	**show bgp ipv4 unicast neighbors** 例： `switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors`	BGP ピアに関する情報を表示します。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-confi`	この設定変更を保存します。

例

次に、プレフィックス ピアのダイナミック AS 番号を設定する例を示します。

```
switch# configure terminal
switch(config)# route-map BGPPeers
switch(config-route-map)# match as-number 64496, 64501-64510
switch(config-route-map)# match as-number as-path-list List1, List2
switch(config-route-map)# exit
switch(config)# router bgp 64496
switch(config-router)# neighbor 192.0.2.0/8 remote-as route-map BGPPeers
switch(config-router-neighbor)# description Peer Router B
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# copy running-config startup-config
```

BGP 情報の消去

BGP 情報を消去するには、次のコマンドを使用します。

コマンド	目的
clear bgp all { *neighbor* | * | *as-number* | **peer-template** *name* | *prefix* } [**vrf** *vrf-name*]	すべてのアドレス ファミリから 1 つ以上のネイバーをクリアします。* を指定すると、すべてのアドレス ファミリのすべてのネイバーが消去されます。引数は次のとおりです。 • *neighbor*：ネイバーの IPv4 アドレス。 • *as-number*：自律システム番号。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。 • *name*：ピア テンプレート名。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。 • prefix：IPv4 プレフィックス。そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。
clear bgp all dampening [**vrf** *vrf-name*]	すべてのアドレス ファミリのルート フラップ ダンプニング ネットワークをクリアします。*vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と小文字は区別されます。
clear bgp all flap-statistics [**vrf** *vrf-name*]	すべてのアドレス ファミリのルート フラップ統計情報をクリアします。*vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と小文字は区別されます。
clear bgp unicast dampening [**vrf** *vrf-name*]	選択したアドレス ファミリのルート フラップ ダンプニング ネットワークをクリアします。*vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と-小文字は区別されます。
clear bgp unicast flap-statistics [**vrf** *vrf-name*]	選択したアドレス ファミリのルート フラップ統計情報をクリアします。*vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と小文字は区別されます。

コマンド	目的
clear bgp { **ipv4** } **unicast** { *neighbor* | * | *as-number* | **peer-template** *name* | *prefix* } [**vrf** *vrf-name*]	すべてのアドレス ファミリから 1 つ以上のネイバーをクリアします。* を指定すると、すべてのアドレス ファミリのすべてのネイバーが消去されます。引数は次のとおりです。 • *neighbor*：ネイバーの IPv4 アドレス。 • *as-number*：自律システム番号。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。 • *name*：ピア テンプレート名。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。 • prefix：IPv4 プレフィックス。そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。
clear bgp { **ip** {**unicast** } { *neighbor* | * | *as-number* | **peer-template** *name* | *prefix* } [**vrf** *vrf-name*]	すべてのアドレス ファミリから 1 つ以上のネイバーをクリアします。* を指定すると、すべてのアドレス ファミリのすべてのネイバーが消去されます。引数は次のとおりです。 • *neighbor*：ネイバーの IPv4 アドレス。 • *as-number*：自律システム番号。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。 • *name*：ピア テンプレート名。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。 • prefix：IPv4 プレフィックス。そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

コマンド	目的
clear ip bgp dampening [*ip-neighbor* | *ip-prefix*] [**vrf** *vrf-name*]	1 つ以上のネットワークのルート フラップ ダンプニングをクリアします。引数は次のとおりです。 • *ip-neighbo*r：ネイバーの IPv4 アドレス。 • *ip-prefix*： IPv4。そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。
clear ip bgp flap-statistics [*ip-neighbor* | *ip-prefix*] [**vrf** *vrf-name*]	1 つ以上のネットワークのルート フラップ統計情報をクリアします。引数は次のとおりです。 • *ip-neighbo*r：ネイバーの IPv4 アドレス。 • *ip-prefix*： IPv4。そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。
clear ip mbgp { **ip** {**unicast** } { *neighbor* | * | *as-number* | **peer-template** *name* | *prefix* } [**vrf** *vrf-name*]	すべてのアドレス ファミリから 1 つ以上のネイバーをクリアします。* を指定すると、すべてのアドレス ファミリのすべてのネイバーが消去されます。引数は次のとおりです。 • *neighbor*：ネイバーの IPv4 アドレス。 • *as-number*：自律システム番号。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。 • *name*：ピア テンプレート名。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。 • prefix：IPv4 プレフィックス。そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

コマンド	目的
clear ip mbgp dampening [*ip-neighbor* | *ip-prefix*] [**vrf** *vrf-name*]	1 つ以上のネットワークのルート フラップ ダンプニングをクリアします。引数は次のとおりです。 • *ip-neighbo*r：ネイバーの IPv4 アドレス。 • *ip-prefix*： IPv4。そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。
clear ip mbgp flap statistics [*ip-neighbor* | *ip-prefix*] [**vrf** *vrf-name*]	1 つ以上のネットワークのルート フラップ統計情報をクリアします。引数は次のとおりです。 • *ip-neighbo*r：ネイバーの IPv4 アドレス。 • *ip-prefix*： IPv4。そのプレフィックス内のすべてのネイバーがクリアされます。 • *vrf-name*：VRF 名。その VRF 内のすべてのネイバーがクリアされます。名称は 64 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

ベーシック BGP の設定の確認

BGP の設定情報を表示するには、次のいずれかの作業を行います。

コマンド	目的
show bgp all [**summary**] [**vrf** *vrf-name*]	すべてのアドレス ファミリについて、BGP 情報を表示します。
show bgp convergence [**vrf** *vrf-name*]	すべてのアドレス ファミリについて、BGP 情報を表示します。
show bgp { **ipv4** { **unicast** } [**ip-address**] **community** { **regexp expression** | [**community**] [**no-advertise**] [**no-export**] [**no-export-subconfed**]} [**vrf** *vrf-name*]	BGP コミュニティと一致する BGP ルートを表示します。
show bgp [**vrf** *vrf-name*] { **ip** } {**unicast**} [**ip-address**] **community-list** *list-name* [**vrf** *vrf-name*]	BGP コミュニティ リストと一致する BGP ルートを表示します。

コマンド	目的
show bgp ip {**unicast**} [**ip-address**] **extcommunity** {**regexp expression** | **generic** [**non-transitive** | **transitive**] **aa4:nn** [**exact-match**]} [**vrf** *vrf-name*]	BGP 拡張コミュニティと一致する BGP ルートを表示します。
show bgp ip { **unicast** } [**ip-address**] **extcommunity-list** *list-name* [**exact-match**] [**vrf** *vrf-name*]	BGP 拡張コミュニティ リストと一致する BGP ルートを表示します。
show bgp ip { **unicast** } [**ip-address**] {**dampening dampened-paths** [**regexp expression**]} [**vrf** *vrf-name*]	BGP ルート ダンプニングの情報を表示します。ルート フラップ ダンプニング情報を消去するには、**clear bgp dampening** コマンドを使用します。
show bgp ip { **unicast**} [**ip-address**] **history-paths** [**regexp expression**] [**vrf** *vrf-name*]	BGP ルート ヒストリ パスを表示します。
show bgp ip { **unicast** } [**ip-address**] **filter-list** *list-name* [**vrf** *vrf-name*]	BGP フィルタ リストの情報を表示します。
show bgp ip { **unicast**} [**ip-address**] **neighbors** [**ip-address**] [**vrf** *vrf-name*]	BGP ピアの情報を表示します。これらのネイバーを消去するには、**clear bgp neighbors** コマンドを使用します。
show bgp ip { **unicast** } [**ip-address**] { **nexthop** | **nexthop-database** } [**vrf** *vrf-name*]	BGP ルート ネクスト ホップの情報を表示します。
show bgp paths	BGP パス情報を表示します。
show bgp ip { **unicast** } [**ip-address**] **policy name** [**vrf** *vrf-name*]	BGP ポリシー情報を表示します。ポリシー情報を消去するには、**clear bgp policy** コマンドを使用します。
show bgp ip { **unicast** } [**ip-address**] **prefix-list** *list-name* [**vrf** *vrf-name*]	プレフィックス リストと一致する BGP ルートを表示します。
show bgp ip { **unicast** } [**ip-address**] received-paths [**vrf** *vrf-name*]	ソフト再構成用に保管されている BGP パスを表示します。
show bgp ip {**unicast**} [**ip-address**] **regexp expression** [**vrf** *vrf-name*]	AS_path 正規表現と一致する BGP ルートを表示します。
show bgp ip {**unicast**} [**ip-address**] **route-map** *map-name* [**vrf** *vrf-name*]	ルート マップと一致する BGP ルートを表示します。
show bgp peer-policy name [**vrf** *vrf-name*]	BGP ピア ポリシー情報を表示します。
show bgp peer-session name [**vrf** *vrf-name*]	BGP ピア セッション情報を表示します。

コマンド	目的
show bgp peer-template name [**vrf** *vrf-name*]	BGP ピア テンプレート情報を表示します。ピア テンプレートのすべてのネイバーを消去するには、**clear bgp peer-template** コマンドを使用します。
show bgp process	BGP プロセス情報を表示します。
show ip bgp *options*	BGP のステータスと構成情報を表示します。このコマンドには複数のオプションがあります。詳細については、Cisco Nexus 3000 シリーズ コマンド リファレンスを参照してください。
show ip mbgp *options*	BGP のステータスと構成情報を表示します。このコマンドには複数のオプションがあります。詳細については、Cisco Nexus 3000 シリーズ コマンド リファレンスを参照してください。
show running-configuration bgp	現在実行中の BGP コンフィギュレーションを表示します。

BGP 統計情報の表示

BGP の統計情報を表示するには、次のコマンドを使用します。

コマンド	目的
show bgp ip {**unicast**} [**ip-address**] **flap-statistics** [**vrf** *vrf-name*]	BGP ルート フラップの統計情報を表示します。これらの統計情報をクリアするには、**clear bgp flap-statistics** コマンドを使用します。
show bgp sessions [**vrf** *vrf-name*]	すべてのピアの BGP セッションを表示します。これらの統計情報をクリアするには、**clear bgp sessions** コマンドを使用します。
show bgp sessions [**vrf** *vrf-name*]	すべてのピアの BGP セッションを表示します。これらの統計情報をクリアするには、**clear bgp sessions** コマンドを使用します。
show bgp statistics	BGP 統計情報を表示します。

ベーシック BGP の設定例

次に、ベーシック BGP 設定の例を示します。

```
feature bgp
router bgp 64496
neighbor 192.0.2.1 remote-as 64496
address-family ipv4 unicast
next-hop-self
```

関連項目

BGP の関連項目は、次のとおりです。

- Route Policy Manager の設定

次の作業

次の機能の詳細については、高度な BGP の設定を参照してください。

- ピア テンプレート
- ルートの再配布
- ルート マップ

その他の参考資料

BGP の実装に関連する詳細情報については、次の項を参照してください。

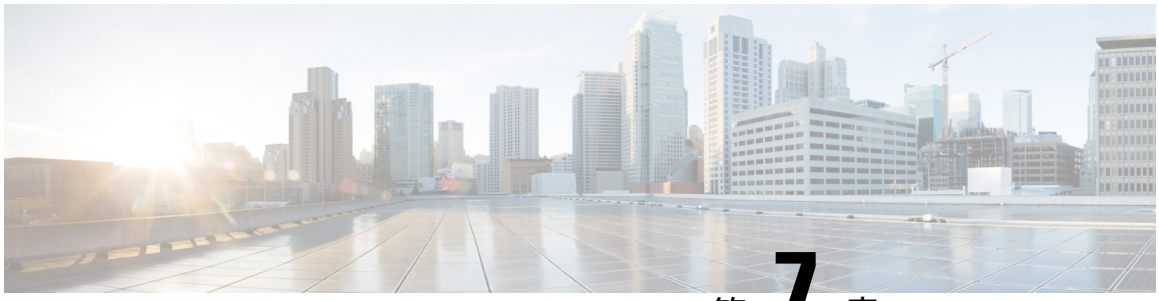
- 関連資料
- MIB

関連資料

関連項目	マニュアル タイトル
BGP CLI コマンド	『Cisco Nexus 3000 Series Command Reference』

MIB

MIB	MIB のリンク
BGP4-MIB CISCO-BGP4-MIB	MIB を検索してダウンロードするには、次の MIB ロケータに移動します。



第 7 章

高度な BGP の設定

この章では、Cisco NX-OS スイッチでボーダー ゲートウェイ プロトコル（BGP）の拡張機能を設定する方法について説明します。

この章は、次の項で構成されています。

拡張 BGP の概要

BGP は、組織または自律システム間のループフリー ルーティングを実現する、インタードメイン ルーティング プロトコルです。Cisco NX-OS は BGP バージョン 4 をサポートします。BGP v4 に組み込まれているマルチプロトコル拡張機能を使用すると、IP ルートおよび複数のレイヤ 3 プロトコル アドレス ファミリに関するルーティング情報を BGP に伝送させることができます。BGP では、他の BGP 対応スイッチ（BGP ピア）との間で TCP セッションを確立するために、信頼できるトランスポート プロトコルとして TCP を使用します。外部組織に接続するときには、ルータが外部 BGP（eBGP）ピアリング セッションを作成します。同じ組織内の BGP ピアは、内部 BGP（iBGP）ピアリング セッションを通じて、ルーティング情報を交換します。

ピア テンプレート

BGP ピア テンプレートを使用すると、類似した BGP ピア間で再利用できる共通のコンフィギュレーションブロックを作成できます。各ブロックでは、ピアに継承させる一連の属性を定義できます。継承した属性の一部を上書きすることもできるので、非常に柔軟性のある方法で、繰り返しの多い BGP の設定を簡素化できます。

Cisco NX-OS は、3 種類のピア テンプレートを実装します。

- **peer-session** テンプレートでは、トランスポートの詳細、ピアのリモート自律システム番号、セッション タイマーなど、BGP セッション属性を定義します。peer-session テンプレートは、別の peer-session テンプレートから属性を継承することもできます（ローカル定義の属性によって、継承した peer-session 属性は上書きされます）。
- **peer-policy** テンプレートでは、着信ポリシー、発信ポリシー、フィルタ リスト、プレフィックス リストを含め、アドレス ファミリに依存する、ピアのポリシー要素を定義します。peer-policy テンプレートは、一連の peer-policy テンプレートからの継承が可能です。Cisco NX-OS は、継承設定のプリファレンス値で指定された順序で、これらの peer-policy テンプレートを評価します。最小値が大きい値よりも優先されます。
- **peer** テンプレートは、peer-session および peer-policy テンプレートからの継承が可能であり、ピアの定義を簡素化できます。peer テンプレートの使用は必須ではありませんが、peer テンプレートによって再利用可能なコンフィギュレーション ブロックが得られるので、BGP の設定を簡素化できます。

認証

BGP ネイバー セッションに認証を設定できます。この認証方式によって、ネイバーに送られる各 TCP セグメントに MD5 認証ダイジェストが追加され、不正なメッセージや TCP セキュリティ アタックから BGP が保護されます。

（注） MD5 パスワードは、BGP ピア間で一致させる必要があります。

ルート ポリシーおよび BGP セッションのリセット

BGP ピアにルート ポリシーを関連付けることができます。ルート ポリシーではルート マップを使用して、BGP が認識するルートを制御または変更します。着信または発信ルート アップデートに関するルート ポリシーを設定できます。ルート ポリシーはプレフィックス、AS_path 属性など、さまざまな条件で一致が必要であり、ルートを選択して受け付けるかまたは拒否します。ルート ポリシーでパス属性を変更することもできます。

BGP ピアに適用するルート ポリシーを変更する場合は、そのピアの BGP セッションをリセットする必要があります。Cisco NX-OS は、BGP ピアリング セッションのリセット方法として、次の 3 種類をサポートします。

- ハード リセット：ハード リセットでは、指定されたピアリング セッションが TCP 接続を含めて切断され、指定のピアからのルートが削除されます。このオプションを使用すると、BGP ネットワーク上のパケット フローが中断します。ハード リセットは、デフォルトでディセーブルです。
- ソフト再構成着信：ソフト再構成着信によって、セッションをリセットすることなく、指定されたピアのルーティング アップデートが開始されます。このオプションを使用できるのは、着信ルート ポリシーを変更する場合です。ソフト再構成着信の場合、ピアから受け取ったすべてのルートのコピーを保存したあとで、着信ルート ポリシーを介してルートが処理されます。着信ルート ポリシーを変更する場合、Cisco NX-OS は変更された着信ルート ポリシーを介して保存ルートを渡し、既存のピアリング セッションを切断することなく、ルート テーブルをアップデートします。ソフト再構成着信の場合、まだフィルタリングされていない BGP ルートの保存に、大量のメモリ リソースを使用する可能性があります。ソフト再構成着信は、デフォルトでディセーブルです。
- ルート リフレッシュ：ルート リフレッシュでは、着信ルート ポリシーの変更時に、サポートするピアにルート リフレッシュ要求を送信することによって、着信ルーティング テーブルがダイナミックにアップデートされます。リモート BGP ピアは新しいルート コピーで応答し、ローカル BGP スピーカが変更されたルート ポリシーでそれを処理します。Cisco NX-OS はピアに、プレフィックスの発信ルート リフレッシュを自動的に送信します。
- BGP ピアは、BGP ピア セッションの確立時に、BGP 機能ネゴシエーションの一部として、ルート リフレッシュ機能をアドバタイズします。ルート リフレッシュは優先オプションであり、デフォルトでイネーブルです。

（注） BGP はさらに、ルート再配布、ルート集約、ルート ダンプニングなどの機能にルート マップを使用します。

eBGP

eBGP を使用すると、異なる AS からの BGP ピアを接続し、ルーティング アップデートを交換できます。外部ネットワークへの接続によって、自分のネットワークから他のネットワークへ、またインターネットを介して、トラフィックを転送できます。

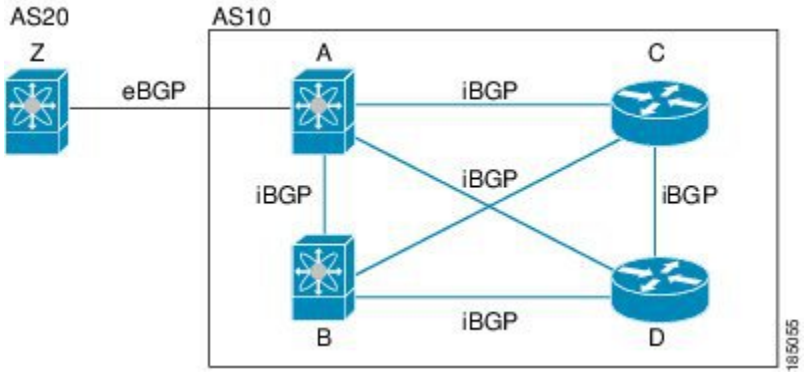
eBGP ピアリング セッションの確立には、ループバック インターフェイスを使用します。ループバック インターフェイスは、インターフェイス フラップが発生する可能性が小さいからです。インターフェイス フラップが発生するのは、障害またはメンテナンスが原因で、インターフェイスが管理上アップまたはダウンになったときです。マルチホップ、高速外部フォールオーバー、AS パス属性のサイズ制限については、eBGP の設定のセクションを参照してください。

iBGP

内部 BGP（iBGP）を使用すると、同じ自律システム内の BGP ピアを接続できます。iBGP はマルチホーム BGP ネットワーク（同じ外部自律システムに対して複数の接続があるネットワーク）に使用できます。

次の図に、より大きな BGP ネットワークの中の iBGP ネットワークを示します。

図 *10 : iBGP* ネットワーク



iBGP ネットワークはフルメッシュです。各 iBGP ピアは、ネットワーク ループを防止するために、他のすべての iBGP ピアに対して直接接続されています。

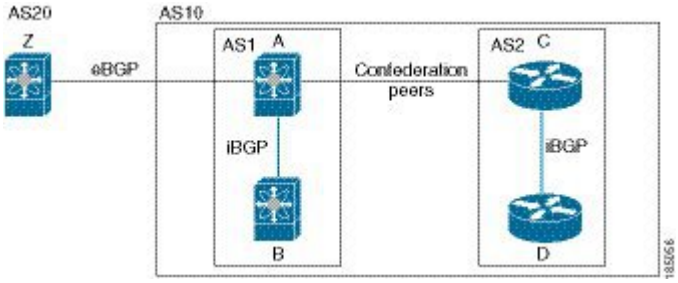
（注） iBGP ネットワークでは別個のインテリア ゲートウェイ プロトコルを設定する必要があります。

AS 連合

フルメッシュの iBGP ネットワークは、iBGP ピア数が増えるにしたがって複雑になります。自律システムを複数のサブ自律システムに分割し、それを 1 つの連合としてまとめることによって、iBGP メッシュを緩和できます。連合は、同じ自律システム番号を使用して外部ネットワークと通信する、iBGP ピアからなるグループです。各サブ AS はその中ではフルメッシュであり、同じ連合内の他のサブ AS に対する少数の接続があります。

図には、BGP ネットワークが 2 つのサブ自律システムと 1 つのコンフェデレーションに分けられて表示されています。

図 *11* : *AS* 連合



この例では、AS10 が 2 つの AS（AS1 および AS2）に分割されています。各サブ AS はフルメッシュですが、サブ AS 間のリンクは 1 つだけです。AS コンフェデレーションを使用することによって、図 1 のフルメッシュ自律システムに比べて、リンク数を少なくできます。

ルート リフレクタ

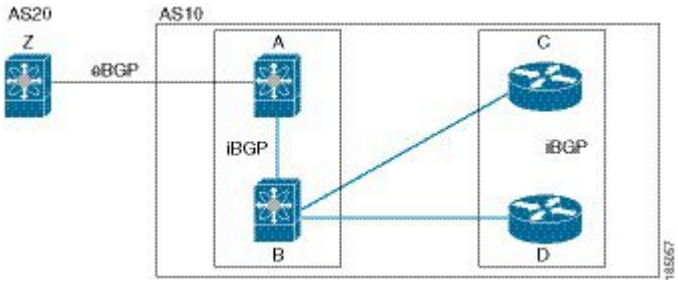
ルート リフレクタ構成を使用することによって、iBGP メッシュを緩和することもできます。ルート リフレクタは学習したルートをネイバーに渡すことで、すべての iBGP ピアをフルメッシュにしなくてもすむようにします。

図 1 に、メッシュの iBGP スピーカを 4 つ使用する（ルータ A、B、C、D）、単純な iBGP 構成を示します。ルート リフレクタを使用しなかった場合、外部ネイバーからルートを受け取ったルータ A は、3 つの iBGP ネイバーのすべてにルートをアドバタイズします。

ある iBGP ピアをルート リフレクタとして設定すると、そのピアが iBGP で学習したルートを一連の iBGP ネイバーに渡す役割を担います。

次の図では、ルータ B がルート リフレクタです。ルート リフレクタは、ルータ A からアドバタイズされたルートを受信すると、ルータ C と D へのルートをアドバタイズ（リフレクト）します。ルータ A は、ルータ C と D の両方にアドバタイズする必要がなくなります。

図 *12* : ルート リフレクタ



ルート リフレクタおよびそのクライアント ピアは、クラスタを形成します。ルート リフレクタのクライアント ピアとして動作するように、すべての iBGP ピアを設定する必要はありません。ただし、完全な BGP アップデートがすべてのピアに届くように、非クライアント ピアはフルメッシュとして設定する必要があります。

機能ネゴシエーション

BGP スピーカは機能ネゴシエーション機能を使用することによって、ピアがサポートする BGP 拡張機能について学習できます。機能ネゴシエーションによって、リンクの両側の BGP ピアがサポートする機能セットだけを BGP に使用させることができます。

BGP ピアが機能ネゴシエーションをサポートしない場合で、なおかつアドレス ファミリが IPv4 として設定されている場合、Cisco NX-OS は機能ネゴシエーションを行わずに、ピアとの新規セッションを試みます。

ルート ダンプニング

ルート ダンプニングは、インターネットワーク上でのフラッピング ルートの伝搬を最小限に抑える BGP 機能です。ルート フラップが発生するのは、使用可能ステートと使用不能ステートが短時間で次々切り替わる場合です。

AS1、AS2、および AS3 という 3 つの BGP 自律システムからなるネットワークの場合について考えてみます。AS1 のルートがフラップした（使用不能になった）とします。ルート ダンプニングを使用しない場合、AS1 は AS2 に回収メッセージを送信します。AS2 は AS3 にその回収メッセージを伝達します。フラッピング ルートが再び発生すると、AS1 から AS2 にアドバタイズメント メッセージを送信し、AS2 は AS3 にそのアドバタイズメントを送信します。ルートの使用不能と使用可能が繰り返されると、AS1 は多数の回収メッセージおよびアドバタイズメント メッセージを送信することになり、それが他の自律システムに伝播します。

ルートダンプニングによって、フラッピングを最小限に抑えることができます。ルート フラップが発生したとします。（ルート ダンプニングがイネーブルの）AS2 がルートにペナルティとして 1000 を割り当てます。AS2 は引き続き、ネイバーにルートの状態をアドバタイズします。ルート フラップが発生するたびに、AS2 がペナルティ値を追加します。ルート フラップが頻繁に発生して、ペナルティが設定可能な抑制限度を超えると、AS2 はフラップ回数に関係なく、ルートのアドバタイズを中止します。その結果、ルートが減衰（ダンプニング）します。

ルートに与えられたペナルティは、再使用限度に達するまで減衰します。その時点で、AS2 は再びルートをアドバタイズします。再使用限度が 50% になると、AS2 はそのルートのダンプニング情報を削除します。

（注） ルート ダンプニングがイネーブルの場合は、ピアのリセットによってルートが回収されても、リセット中の BGP にはペナルティは適用されません。

ロード シェアリングおよびマルチパス

BGP はルーティング テーブルに、同じ宛先プレフィックスに到達する複数の等コスト eBGP または iBGP パスを組み込むことができます。その場合、宛先プレフィックスへのトラフィックは、組み込まれたすべてのパス間で共有されます。

BGP ベストパス アルゴリズムでは、次の属性が同じ場合に、等コスト パスと見なされます。

- 重量
- ローカル プリファレンス
- AS_path
- オリジン コード
- Multi-Exit Discriminator（MED）
- BGP ネクスト ホップまでの IGP コスト

BGP はこれら複数のパスの中から、ベスト パスとして 1 つだけ選択し、そのパスを BGP ピアにアドバタイズします。

（注） 異なる AS 連合から受け取ったパスは、外部 AS_path 値およびその他の属性が同じ場合に、等コスト パスと見なされます。

（注） iBGP マルチパスに関してルート リフレクタを設定すると、ルート リフレクタが、選択されたベスト パスをピアにアドバタイズします。そのパスのネクスト ホップは変更されません。

ルート集約

集約アドレスを設定できます。ルート集約を使用すると、固有性の強い一連のアドレスをすべての固有アドレスを代表する 1 つのアドレスに置き換えることによって、ルート テーブルを簡素化できます。たとえば、10.1.1.0/24、10.1.2.0/24、および 10.1.3.0/24 という固有性の強い 3 つのアドレスを 1 つの集約アドレス 10.1.0.0/16 に置き換えることができます。

アドバタイズされるルートが少なくなるように、BGP ルート テーブル内には集約プレフィックスが存在します。

（注） Cisco NX-OS は、自動ルート集約をサポートしていません。

ルート集約はフォワーディング ループにつながる可能性があります。この問題を回避するために、集約アドレスのアドバタイズメントを生成するときに、BGP はローカル ルーティング テーブルに、その集約アドレスに対応するサマリー廃棄ルートを自動的に組み込みます。BGP はサマリー廃棄のアドミニストレーティブ ディスタンスを 220 に設定し、ルート タイプを廃棄に設定します。BGP はネクストホップ解決に廃棄ルートを使用しません。

BGP 条件付きアドバタイズメント

BGP 条件付きアドバタイズメントを使用すると、プレフィックスが BGP テーブルに存在するかどうかに基づいてルートをアドバタイズまたは撤回するように BGP を設定できます。この機能は、たとえば、BGP でいずれかのプロバイダーにプレフィックスをアドバタイズするようなマルチホーム ネットワーク（他のプロバイダーからの情報が存在しない場合のみ）で便利です。

AS1、AS2、および AS3 という 3 つの BGP 自律システムからなるネットワークの例について考えてみます。この例で、AS1 と AS3 はインターネットと AS2 に接続しています。条件付きアドバタイズメントを使用しない場合、AS2 はすべてのルートを AS1 と AS3 の両方にプロパゲートします。条件付きアドバタイズメントを使用すれば、AS1 からのルートが存在しない場合のみ（たとえば AS1 へのリンクがダウンした場合）、特定のルートを AS3 にアドバタイズするように AS2 を設定できます。

BGP 条件付きアドバタイズメントでは、設定されたルート マップに一致する各ルートに、存在テストまたは非存在テストが追加されます。「BGP 条件付きアドバタイズメントの設定」を参照してください。

BGP ネクスト ホップ アドレス トラッキング

BGP は、インストールされているルートのネクスト ホップ アドレスをモニタして、ネクスト ホップの到達可能性の確認、および BGP ベストパスの選択、インストール、検証を行います。BGP ネクストホップ アドレスのトラッキングを行うと、ネクストホップの到達可能性に影響を及ぼす可能性のあるルート変更が RIB で行われたときに確認プロセスをトリガーすることで、このようなネクストホップ到達可能性テストの速度が向上します。

ネクストホップ情報が変更されると、BGP は RIB から通知を受信します（イベント駆動型の通知）。BGP は、次のいずれかのイベントが発生したときに通知を受けます。

- ネクスト ホップが到達不能になった。
- ネクスト ホップが到達可能になった。
- ネクスト ホップへの完全な繰り返し IGP メトリックが変更される。
- ファースト ホップの IP アドレスまたはファースト ホップのインターフェイスが変更される。
- ネクスト ホップが接続された。
- ネクスト ホップが接続解除された。
- ネクスト ホップがローカル アドレスになった。
- ネクスト ホップが非ローカル アドレスになった。

（注） 到達可能性および再帰メトリック イベントは、最適パスの再計算をトリガーします。

RIB からのイベント通知は、クリティカルおよび非クリティカルとして分類されます。クリティカルおよび非クリティカルイベントの通知は、別々のバッチで送信されます。ただし、非クリティカルイベントが保留中であり、クリティカルイベントを読み込む要求がある場合は、非クリティカル イベントがクリティカル イベントとともに送信されます。

- クリティカル イベントは、ネクスト ホップの到達可能性（到達可能と到達不能）、接続性（接続と非接続）、および局在性（ローカルと非ローカル）に関係があります。これらのイベントの通知は遅延しません。
- 非クリティカル イベントには、IGP メトリックの変更のみが含まれます。

詳細については、「BGP ネクスト ホップ アドレス トラッキングの設定」を参照してください。

ルートの再配布

スタティック ルートまたは他のプロトコルからのルートを再配布するように、BGP を設定できます。再配布を指定してルート ポリシーを設定し、BGP に渡されるルートを制御します。ルート ポリシーを使用すると、宛先、送信元プロトコル、ルート タイプ、ルート タグなどの属性に基づいて、ルートをフィルタリングできます。詳細については、Route Policy Manager の設定のセクションを参照してください。

BGP の調整

BGP タイマーによって、さらにベストパス アルゴリズムの調整によって、BGP のデフォルト動作を変更できます。

BGP タイマー

BGP では、ネイバー セッションおよびグローバル プロトコル イベントにさまざまなタイプのタイマーを使用します。確立されたセッションごとに、最低限2つのタイマーがあります。定期的にキープアライブ メッセージを送信するためのタイマー、さらに想定時間内にピアのキープアライブが届かなかった場合に、セッションをタイムアウトさせるためのタイマーです。また、個々の機能を処理するための、その他のタイマーがあります。これらのタイマーは通常、秒単位で設定します。タイマーには、異なる BGP ピアで同じタイマーが異なるタイミングでスタートするように、ランダム アジャストメントが組み込まれています。

ベストパス アルゴリズムの調整

オプションの設定パラメータによって、ベストパスアルゴリズムのデフォルト動作を変更できます。たとえば、アルゴリズムでの MED 属性およびルータ ID の扱い方を変更できます。

マルチプロトコル BGP

Cisco NX-OS の BGP は、複数のアドレス ファミリをサポートします。マルチプロトコル BGP（MP-BGP）は、アドレス ファミリに応じて異なるルート セットを伝送します。たとえば、

BGP は IPv4 ユニキャスト ルーティングのルート 1 セットを伝送します。IP マルチキャスト ネットワークではリバース パス フォワーディング（RPF）のチェックに MP-BGP を使用できます。

（注） マルチキャスト BGP ではマルチキャスト状態情報をプロパゲートしないため、プロトコル独立マルチキャスト（PIM）などのマルチキャスト プロトコルが必要です。

マルチプロトコル BGP 設定をサポートするには、ルータ アドレスファミリおよびネイバー アドレス ファミリの各コンフィギュレーション モードを使用します。MP-BGP では、設定されたアドレス ファミリごとに別々の RIB が維持されます（ユニキャスト RIB と、BGP のマルチキャスト RIB など）。

マルチプロトコル BGP ネットワークは下位互換性がありますが、マルチプロトコル拡張機能をサポートしない BGP ピアは、アドレス ファミリ ID 情報など、マルチプロトコル拡張機能が伝送するルーティング情報を転送できません。

BGP の前提条件

BGP を使用するには、次の前提条件を満たしている必要があります。

- BGP 機能を有効にする必要があります（BGP 機能のイネーブル化のセクションを参照）。
- システムに有効なルータ ID を設定しておく必要があります。
- Regional Internet Registry（RIR）によって割り当てられたか、またはローカル管理の AS 番号を取得しておく必要があります。
- ネイバー関係を作成しようとするピアに到達可能でなければなりません（Interior Gateway Protocol（IGP）、スタティック ルート、直接接続など）。
- BGP セッションを確立するネイバー環境で、アドレス ファミリを明示的に設定する必要があります。

拡張 BGP に関する注意事項と制限事項

BGP 設定時の注意事項および制約事項は、次のとおりです。

- プレフィックス ピアリングは、パッシブ TCP モードでのみ動作します。ピア アドレスがプレフィックス内にある場合、リモート ピアからの着信接続を受け入れます。
- ダイナミック自律システム番号プレフィックス ピア設定は、BGP テンプレートから継承した個々の自律システム番号の設定よりも優先します。
- 自律システム連合でプレフィックス ピアにダイナミック自律システム番号を設定した場合、BGP はローカル連合の自律システム番号のみでセッションを確立します。

- ダイナミック自律システム番号プレフィックス ピアで作成された BGP セッションは、設定済みの eBGP マルチホップ存続可能時間（TTL）値や直接接続ピアに対するディセーブル済みのチェックを無視します。
- ルータ ID の自動変更およびセッション フラップを避けるために、BGP 用のルータ ID を設定します。
- ピアごとに最大プレフィックス設定オプションを使用し、受信するルート数および使用するシステム リソース数を制限してください。
- update-source を設定し、eBGP マルチホップ セッションでセッションを確立します。
- 再配布を設定する場合は、BGP ルート マップを指定します。
- VRF 内で BGP ルータ ID を設定します。
- キープアライブおよびホールドタイマーの値を小さくすると、ネットワークでセッション フラップが発生する可能性があります。

BGP のデフォルト設定

次の表に、BGP パラメータのデフォルト設定値を示します。

表 *9:* デフォルトの *BGP* パラメータ

パラメータ	デフォルト
BGP 機能	無効
キープアライブ インターバル	60 秒
ホールド タイマー	180 秒

高度な BGP の設定

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

BGP セッション テンプレートの設定

BGP セッション テンプレートを使用すると、類似した設定が必要な複数の BGP ピアで、BGP の設定を簡素化できます。BGPテンプレートによって、共通のコンフィギュレーションブロックを再利用できます。先に BGP テンプレートを設定し、その後で BGP ピアにテンプレートを適用します。

BGP セッション テンプレートでは、継承、パスワード、タイマー、セキュリティなどのセッション属性を設定できます。

peer-session テンプレートは、別の peer-session テンプレートからの継承が可能です。第 3 のテンプレートから継承するように第2テンプレートを設定できます。さらに最初のテンプレートもこの第3のテンプレートから継承させることができます。この間接継承を続けることができる peer-session テンプレートの数は、最大 7 つです。

ネイバーに設定した属性は、ネイバーが BGP テンプレートから継承した属性よりも優先されます。

始める前に

BGP 機能を有効にしていることを確認します（BGP 機能のイネーブル化のセクションを参照してください）。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer-session** *template-name*
4. （任意） **password** *number password*
5. （任意） **timers** *keepalive hold*
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **inherit peer-session** *template-name*
9. （任意） **description** *text*
10. （任意） **show bgp peer-session** *template-name*
11. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *autonomous-system-number* 例：	BGP を有効にして、ローカル BGP スピーカに自律システム番号を割り当てます。

	コマンドまたはアクション	目的
	`switch(config)# router bgp 65536` `switch(config-router)#`	

ステップ 3	**template peer-session** *template-name*
	例：
	`switch(config-router)# template peer-session BaseSession`
	`switch(config-router-stmp)#`
		peer-session テンプレート コンフィギュレーション モードを開始します。

ステップ 4	（任意） **password** *number password* 例： `switch(config-router-stmp)# password 0 test`	ネイバーにクリア テキストのパスワード *tes* を追加します。パスワードは 3DES（タイプ 3 暗号形式）で保存および表示されます。
ステップ 5	（任意） **timers** *keepalive hold* 例： `switch(config-router-stmp)# timers 30 90`	peer-session テンプレートに BGP キープアライブおよびホールドタイマー値を追加します。 デフォルトのキープアライブ インターバルは 60 です。デフォルトのホールド タイムは 180 です。

ステップ 6	**exit**
	例：
	`switch(config-router-stmp)# exit`
	`switch(config-router)#`
		peer-session テンプレート コンフィギュレーション モードを終了します。

ステップ 7	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65536`
	`switch(config-router-neighbor)#`
		BGP ルーティング用のネイバー コンフィギュレーション モードを開始し、ネイバー IP アドレスを設定します。

ステップ 8	**inherit peer-session** *template-name*
	例：
	`switch(config-router-neighbor)# inherit peer-session BaseSession`
	`switch(config-router-neighbor)`
		ピアに peer-session テンプレートを適用します。

ステップ 9	（任意） **description** *text*
	例：
	`switch(config-router-neighbor)# description Peer Router A`
	`switch(config-router-neighbor)`
		ネイバーの説明を追加します。

ステップ 10	（任意） **show bgp peer-session** *template-name* 例： `switch(config-router-neighbor)# show bgp peer-session BaseSession`	peer-policy テンプレートを表示します。

	コマンドまたはアクション	目的
ステップ 11	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor)# copy running-config startup-config`	この設定変更を保存します。

例

show bgp neighbor コマンドを実行して、適用されたテンプレートを確認します。テンプレートで使用できるすべてのコマンドの詳細については、Cisco Nexus 3000 シリーズ コマンド リファレンスを参照してください。

BGP peer-session テンプレートを設定して、BGP ピアに適用する例を示します。

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer-session BaseSession
switch(config-router-stmp)# timers 30 90
switch(config-router-stmp)# exit
switch(config-router)# neighbor 192.168.1.2 remote-as 65536
switch(config-router-neighbor)# inherit peer-session BaseSession
switch(config-router-neighbor)# description Peer Router A
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor)# copy running-config startup-config
```

BGP peer-policy テンプレートの設定

peer-policyテンプレートを設定すると、特定のアドレス ファミリに対応する属性を定義できます。各 peer-policyテンプレートにプリファレンスを割り当て、指定した順序でテンプレートが継承されるようにします。ネイバー アドレス ファミリでは最大 5 つの peer-policy テンプレートを使用できます。

Cisco NX-OS は、プリファレンス値を使用して、アドレス ファミリの複数のピア ポリシーを評価します。プリファレンス値が最小のものが最初に評価されます。ネイバーに設定した属性は、ネイバーが BGP テンプレートから継承した属性よりも優先されます。

peer-policyテンプレートでは、AS-path フィルタ リスト、プレフィックス リスト、ルート リフレクション、ソフト再構成など、アドレス ファミリ固有の属性を設定できます。

始める前に

BGP 機能を有効にしていることを確認します（BGP 機能のイネーブル化のセクションを参照）。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer-policy** *template-name*

4. （任意） **advertise-active-only**
5. （任意） **maximum-prefix** *number*
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **address-family ipv4 unicast**
9. **inherit peer-policy** *template-name preference*
10. （任意） **show bgp peer-policy** *template-name*
11. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 65536`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに自律システム番号を割り当てます。

ステップ 3	**template peer-policy** *template-name*
	例：
	`switch(config-router)# template peer-policy BasePolicy`
	`switch(config-router-ptmp)#`
		peer-policy テンプレートを作成します。

ステップ 4	（任意） **advertise-active-only** 例： `switch(config-router-ptmp)# advertise-active-only`	アクティブ ルートのみをピアにアドバタイズします。
ステップ 5	（任意） **maximum-prefix** *number* 例： `switch(config-router-ptmp)# maximum-prefix 20`	このピアに認めるプレフィックスの最大数を設定します。

ステップ 6	**exit**
	例：
	`switch(config-router-ptmp)# exit`
	`switch(config-router)#`
		peer-policy テンプレート コンフィギュレーション モードを終了します。

ステップ 7	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65536`
	`switch(config-router-neighbor)#`
		BGP ルーティング用のネイバー設定モードを開始し、ネイバー IP アドレスを設定します。

	コマンドまたはアクション	目的

ステップ 8	**address-family ipv4 unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		指定されたアドレス ファミリに対応するグローバル アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 9	**inherit peer-policy** *template-name preference* 例： `switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1`	ピア アドレス ファミリ設定に peer-policy テンプレートを適用し、このピア ポリシーのプリファレンス値を割り当てます。
ステップ 10	（任意） **show bgp peer-policy** *template-name* 例： `switch(config-router-neighbor-af)# show bgp peer-policy BasePolicy`	peer-policy テンプレートを表示します。
ステップ 11	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor)# copy running-config startup-config`	この設定変更を保存します。

例

show bgp neighbor コマンドを実行して、適用されたテンプレートを確認します。テンプレートで使用できるすべてのコマンドの詳細については、Cisco Nexus 3000 シリーズ コマンド リファレンスを参照してください。

BGP peer-session テンプレートを設定して、BGP ピアに適用する例を示します。

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer-session BasePolicy
switch(config-router-ptmp)# maximum-prefix 20
switch(config-router-ptmp)# exit
switch(config-router)# neighbor 192.168.1.1 remote-as 65536
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# inherit peer-policy BasePolicy
switch(config-router-neighbor-af)# copy running-config startup-config
```

BGP peer テンプレートの設定

BGP peer テンプレートを設定すると、1 つの再利用可能なコンフィギュレーション ブロックで、セッション属性とポリシー属性を結合することができます。peer テンプレートも、peer-session または peer-policy テンプレートを継承できます。ネイバーに設定した属性は、ネイバーが BGP テンプレートから継承した属性よりも優先されます。ネイバーに設定できる peer

テンプレートは 1 つだけですが、peer テンプレートは peer-session および peer-policy テンプレートを継承できます。

peer テンプレートは、eBGP マルチホップ TTL、最大プレフィックス数、ネクストホップ セルフ、タイマーなど、セッション属性およびアドレス ファミリ属性をサポートします。

始める前に

BGP 機能を有効にしていることを確認します（BGP 機能のイネーブル化のセクションを参照してください）。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer-session** *template-name*
4. （任意） **inherit peer-session** *template-name*
5. **address-family ipv4 unicast**
6. （任意） **inherit peer** *template-name*
7. **exit**
8. （任意） **timers** *keepalive hold*
9. **exit**
10. **neighbor** *ip-address* **remote-as** *as-number*
11. **inherit peer** *template-name*
12. （任意） **timers** *keepalive hold*
13. （任意） **show bgp peer-template** *template-name*
14. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 65536`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに自律システム番号を割り当てます。

ステップ 3	**template peer-session** *template-name*
	例：
	`switch(config-router)# template peer-session BaseSession`
	`switch(config-router-stmp)#`
		peer-session 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。

	コマンドまたはアクション	目的
ステップ 4	（任意） **inherit peer-session** *template-name* 例： `switch(config-router-neighbor)# inherit peer-session BaseSession`	peer テンプレートで peer-session テンプレートを継承します。

ステップ 5	**address-family ipv4 unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		指定のアドレス ファミリに対しグローバル アドレス ファミリ コンフィギュレーション モードを設定します。

ステップ 6	（任意） **inherit peer** *template-name* 例： `switch(config-router-neighbor-af)# inherit peer BasePolicy`	ネイバー アドレス ファミリ設定に peer-policy テンプレートを適用します。

ステップ 7	**exit**
	例：
	`switch(config-router-neighbor-af)# exit`
	`switch(config-router-neighbor)#`
		BGP ネイバー アドレス ファミリ コンフィギュレーション モードを終了します。

ステップ 8	（任意） **timers** *keepalive hold* 例： `switch(config-router-neighbor)# timers 45 100`	ピアに BGP タイマー値を追加します。 これらの値によって、peer-session テンプレート、BaseSession のタイマー値が上書きされます。

ステップ 9	**exit**
	例：
	`switch(config-router-stmp)# exit`
	`switch(config-router)#`
		BGP peer テンプレート コンフィギュレーション モードを終了します。

ステップ 10	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65536`
	`switch(config-router-neighbor)#`
		BGP ルーティング用のネイバー コンフィギュレーション モードを開始し、ネイバー IP アドレスを設定します。

ステップ 11	**inherit peer** *template-name* 例： `switch(config-router-neighbor)# inherit peer BasePeer`	peer テンプレートを継承します。
ステップ 12	（任意） **timers** *keepalive hold* 例： `switch(config-router-neighbor)# timers 60 120`	このネイバーに BGP タイマー値を追加します。 これらの値によって、peer テンプレートおよび peer-session テンプレートのタイマー値が上書きされます。

	コマンドまたはアクション	目的
ステップ 13	（任意） **show bgp peer-template** *template-name* 例： `switch(config-router-neighbor-af)# show bgp peer-template BasePeer`	peer テンプレートを表示します。
ステップ 14	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor)# copy running-config startup-config`	この設定変更を保存します。

例

show bgp neighbor コマンドを実行して、適用されたテンプレートを確認します。テンプレートで使用できるすべてのコマンドの詳細については、Cisco Nexus 3000 シリーズ コマンド リファレンスを参照してください。

BGP peer テンプレートを設定して、BGP ピアに適用する例を示します。

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# template peer BasePeer
switch(config-router-neighbor)# inherit peer-session BaseSession
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# exit
switch(config-router)# neighbor 192.168.1.2 remote-as 65536
switch(config-router-neighbor)# inherit peer BasePeer
switch(config-router-neighbor)# copy running-config startup-config
```

IPv4 および IPv6 アドレス ファミリ向け IPv6 リンク ローカル経由の BGP インターフェイス ピアリングの設定

アンナンバード インターフェイスを使用した自動 BGP ネイバー探索のために、IPv4 および IPv6 アドレス ファミリの IPv6 リンクローカルを経由して、BGP インターフェイス ピアリングを設定できます。これにより、インターフェイス名を（インターフェイススコープのアドレスではなく）BGP ピアとして使用する BGP セッションを設定できます。この機能は、ICMPv6 ネイバー探索（ND）のルート アドバタイズメント（RA）を使用して自動ネイバー探索を行い、RFC 5549 を使用して IPv6 ネクスト ホップで IPv4 ルートを送信します。

始める前に

BGP を有効にする必要があります。

手順の概要

1. **configure terminal**

2. **router bgp** *autonomous-system-number*
3. **neighbor** *interface-name* **remote-as** {*as-number* | **route-map** *map-name*}
4. **inherit peer** *template-name*
5. （任意） **maximum-peers** *value*
6. **address-family** {**ipv4** | **ipv6**} **unicast**
7. （任意） **show bgp** {**ipv4** | **ipv6**} **unicast neighbors** *interface*
8. （任意） **show ip bgp neighbors** *interface-name*
9. （任意） **show ipv6 routers** [**interface** *interface*]
10. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `switch# configure terminal`	コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *autonomous-system-number*
	例：
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに自律システム番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

ステップ 3	**neighbor** *interface-name* **remote-as** {*as-number* | **route-map** *map-name*}
	例：
	`switch(config-router)# neighbor Ethernet1/1 remote-as route-map Testmap`
	`switch(config-router-neighbor)#`
		BGP ルーティングのためにルータをネイバー設定モードにして、インターフェイスを BGP ピア用に設定します。
		（注） 指定できるのは、イーサネット インターフェイス、ポートチャネル インターフェイス、サブインターフェイス、およびブレークアウト インターフェイスだけです。
		Cisco NX-OS リリース 9.3(6) 以降では、ルート マップを指定でき、AS リストを含められるルート マップを指定できます。ダイナミック AS 番号の使用の詳細については、プレフィックス ピアのダイナミック AS 番号（121 ページ）を参照してください。
		設定を複数のインターフェイスに適用する必要がある場合、*interface-name* は範囲にすることができます。

	コマンドまたはアクション	目的
ステップ 4	**inherit peer** *template-name* 例： `switch(config-router-neighbor)# inherit peer PEER`	peer テンプレートを継承します。
ステップ 5	（任意） **maximum-peers** *value* 例： `switch(config-router-neighbor)# maximum-peers 120`	ネイバー設定モードのこのプレフィックス ピアリングの最大ピア数を設定します。範囲は 1 ～ 1000 です。 （注） 単一のインターフェイス ピアによって起動できるセッションのデフォルト数は 1 です。

ステップ 6	**address-family** {**ipv4** | **ipv6**} **unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		指定のアドレス ファミリに対しグローバル アドレス ファミリ設定モードを開始します。

ステップ 7	（任意） **show bgp** {**ipv4** | **ipv6**} **unicast neighbors** *interface*
	例：
	`switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors e1/25`
	例：
	`switch(config-router-neighbor-af)# show bgp ipv6 unicast neighbors 3FFE:700:20:1::11`
		BGP ピアに関する情報を表示します。

ステップ 8	（任意） **show ip bgp neighbors** *interface-name* 例： `switch(config-router-neighbor-af)# show ip bgp neighbors Ethernet1/1`	BGP ピアとして使用されるインターフェイスを表示します。
ステップ 9	（任意） **show ipv6 routers** [**interface** *interface*] 例： `switch(config-router-neighbor-af)# show ipv6 routers interface Ethernet1/1`	IPv6 ICMP ルータ アドバタイズメントによって学習されたリモート IPv6 ルータのリンク ローカル アドレスを表示します。
ステップ 10	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor-af)# copy running-config startup-config`	この設定変更を保存します。

例

この例は、ルート マップを使用して、IPv4 および IPv6 アドレス ファミリの IPv6 リンクローカル経由で、BGP インターフェイス ピアリングを設定する例を示します。

リーフ 1 の iBGP インターフェイス ピアリング設定：

```
switch# configure terminal
switch(config)# route-map Testmap permit 10
switch(config-route-map)# match as-number 100-200, 300, 400
switch(config-route-map)# exit
switch(config)# router bgp 65000
switch(config-router)# neighbor Ethernet1/1 remote-as route-map Testmap
switch(config-router-neighbor)# inherit peer PEER
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor)# address-family ipv6 unicast
switch(config-router-neighbor-af)# copy running-config startup-config
```

次に、IPv4 および IPv6 アドレス ファミリの IPv6 リンクローカル経由での、BGP インターフェイス ピアリングのサンプル出力例を示します。

```
switch(config-router-neighbor)# show bgp ipv4 unicast neighbors e1/15.1
BGP neighbor is fe80::2, remote AS 100, ibgp link, Peer index 4
Peer is an instance of interface peering Ethernet1/15.1
BGP version 4, remote router ID 5.5.5.5
Neighbor previous state = OpenConfirm
BGP state = Established, up for 2d16h
Neighbor vrf: default
Peer is directly attached, interface Ethernet1/15.1
Last read 00:00:54, hold time = 180, keepalive interval is 60 seconds
Last written 00:00:08, keepalive timer expiry due 00:00:51
Received 3869 messages, 0 notifications, 0 bytes in queue
Sent 3871 messages, 0 notifications, 0(0) bytes in queue
Enhanced error processing: On
0 discarded attributes
Connections established 2, dropped 1
Last reset by peer 2d16h, due to session closed
Last error length received: 0
Reset error value received 0
Reset error received major: 104 minor: 0
Notification data received:
Last reset by us never, due to No error
Last error length sent: 0
Reset error value sent: 0
Reset error sent major: 0 minor: 0
--More--
```

インターフェイス コンフィギュレーション：

次のいずれかのコマンドを使用して、対応するインターフェイスで IPv6 を有効にする必要があります。

- **ipv6 address** *ipv6-address*
- **ipv6 address use-link-local-only**
- **ipv6 link-local** *link-local-address*

```
switch# configure terminal
switch(config)# interface Ethernet1/1
switch(config-if)# ipv6 address use-link-local-only
```

（注） インターフェイスで IPv4 アドレスが設定されていない場合は、**ip forward** コマンドをインターフェイスで設定して IPv4 転送を有効にする必要があります。

（注） IPv6 ND タイマーを調整して、ネイバー探索を高速化し、BGP のルートコンバージェンスを高速化できます。

```
switch(config-if)# ipv6 nd ra-interval 4 min 3
switch(config-if)# ipv6 nd ra-lifetime 10
```

（注） Cisco NX-OS リリース 9.3(6) 以降で、パラレル リンクを使用するカスタマーの導入では、インターフェイス モードで次のコマンドを追加する必要があります。

```
switch(config-if)# ipv6 link-local use-bia
```

このコマンドは、異なるインターフェイス間での IPv6 LLA を一意にします。

BGP 認証の設定

MD5 ダイジェストを使用してピアからのルート更新を認証するように、BGP を設定できます。

MD5 認証を使用するように BGP を設定するには、ネイバー コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
password [**0** | **3** | **7**] *string* 例： `switch(config-router-neighbor)# password BGPpassword`	MGP ネイバー セッションの MD5 パスワードを設定します。

BGP セッションのリセット

BGP のルート ポリシーを変更した場合は、関連付けられた BGP ピア セッションをリセットする必要があります。BGP ピアがルート リフレッシュをサポートしない場合は、着信ポリシー変更に関するソフト再構成を設定できます。Cisco NX-OS は自動的に、セッションのソフト リセットを試みます。

ソフト再構成着信を設定するには、ネイバー アドレス ファミリ設定モードで次のコマンドを使用します。

コマンド	目的
soft-reconfiguration inbound 例： `switch(config-router-neighbor-af)# soft-reconfiguration inbound`	着信 BGP ルート アップデートを格納するために、ソフト再構成をイネーブルにします。このコマンドによって、BGP ネイバー セッションの自動ソフト クリアまたはリフレッシュが開始されます。

BGP ネイバー セッションをリセットするには、任意のモードで次のコマンドを使用します。

コマンド	目的
clear bgp ip { **unicast** } *ip-address* **soft** { **in** | **out** } 例： `switch# clear bgp ip unicast 192.0.2.1 soft in`	TCP セッションを切断しないで、BGP セッションをリセットします。

ネクストホップ アドレスの変更

次の方法で、ルートアドバタイズメントで使用するネクストホップアドレスを変更できます。

- ネクストホップ計算をディセーブルにして、ローカル BGP スピーカ アドレスをネクストホップ アドレスとして使用します。
- ネクストホップ アドレスをサードパーティ アドレスとして設定します。この機能は、元のネクスト ホップ アドレスがルートの送り先のピアと同じサブネット上にある場合に使用します。この機能を使用すると、フォワーディング時に余分なホップを節約できます。

ネクストホップ アドレスを変更するには、コマンドアドレス ファミリ コンフィギュレーション モードで次のパラメータを使用します。

コマンド	目的
next-hop-self 例： `switch(config-router-neighbor-af)# next-hop-self`	ルートアップデートのネクストホップアドレスとして、ローカル BGP スピーカ アドレスを使用します。このコマンドによって、BGP ネイバー セッションの自動ソフト クリアまたはリフレッシュが開始されます。
next-hop-third-party 例： `switch(config-router-neighbor-af)# next-hop-third-party`	ネクストホップ アドレスをサードパーティ アドレスとして設定します。このコマンドは、**next-hop-self** を設定されていないシングルホップ EBGP ピアに使用します。

BGP ネクスト ホップ アドレス トラッキングの設定

BGP ネクスト ホップ アドレス トラッキングはデフォルトで有効であり、無効にすることができません。

BGP ネクスト ホップ トラッキングのパフォーマンスを向上するために、RIB チェック間の遅延インターバルを変更できます。BGP ネクストホップの到達可能性に影響を及ぼすルートのクリティカル タイマーを設定したり、BGP テーブルのその他のルートすべての非クリティカル タイマーを設定したりできます。

BGP ネクストホップ アドレス トラッキングを変更するには、アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
nexthop trigger-delay {**critical** | **non-critical** } *milliseconds* 例： `switch(config-router-af)# nexthop trigger-delay critical 5000`	クリティカルなネクスト ホップの到達可能性ルートおよび非クリティカルなルートについて、ネクストホップ アドレス トラッキングの遅延タイマーを指定します。指定できる範囲は 1 ～ 4294967295 ミリ秒です。クリティカル タイマーのデフォルトは 3000 です。非クリティカル タイマーのデフォルトは 10000 です。
nexthop route-map *name* 例： `switch(config-router-af)# nexthop route-map nextHopLimits`	BGP ネクストホップ アドレスが一致するルート マップを指定します。63 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

ネクスト ホップ フィルタリングの設定

BGP ネクストホップ フィルタリングを使用すると、RIB でネクストホップ アドレスがチェックされるときにそのネクストホップ アドレスの基盤となるルートがルート マップを経由します。ルート マップでそのルートが拒否されると、ネクストホップ アドレスは到達不能として扱われます。

BGP は、ルート ポリシーによって拒否されたすべてのネクストホップを無効であるとマークし、無効なネクストホップ アドレスを使用するルートについてベスト パスを計算しません。

BGP ネクストホップ フィルタリングを設定するには、アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
nexthop route-map *name* 例： `switch(config-router-af)# nexthop route-map nextHopLimits`	BGP ネクストホップ ルートが一致するルート マップを指定します。63 文字以内の英数字のストリング（大文字と小文字を区別）で指定します。

ネクストホップセルフによるリフレクト ルートの制御

NX-OS では、**next-hop-self** [all] 引数を使用して特定のピアに送信する際の iBGP ルートを制御できます。これらの引数を使用すると、ルートのリフレクトが実施されている場合でも、ルートのネクストホップを選択的に変更できます。

コマンド	目的
next-hop-self [all] 例： `switch(config-router-af)# next-hop-self all`	ルートアップデートのネクストホップアドレスとして、ローカルBGPスピーカアドレスを使用します。 all キーワードはオプションです。all を指定すると、すべてのルートが next-hop-self を使用するピアに送信されます。all を指定しなかった場合、リフレクトしたルートのネクストホップは変更されません。

機能ネゴシエーションのディセーブル化

機能ネゴシエーションをディセーブルにすると、機能ネゴシエーションをサポートしない古い BGP ピアとの相互運用が可能です。

機能ネゴシエーションをディセーブルにするには、ネイバー コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
dont-capability-negotiate 例： `switch(config-router-neighbor)# dont-capability-negotiate`	機能ネゴシエーションをディセーブルにします。このコマンドの設定後、BGP セッションを手動でリセットする必要があります。

eBGP の設定

このセクションは、次のトピックで構成されています。

eBGP シングルホップ チェックの無効化

シングルホップ eBGP ピアがローカル ルータに直接接続されているかどうかのチェック機能を無効にするように、eBGP を設定できます。このオプションは、直接接続されたスイッチ間のシングルホップ ループバック eBGP セッションの設定に使用します。

シングルホップ eBGP ピアが直接接続されているかどうかのチェックを無効にするには、ネイバー設定モードで次のコマンドを使用します。

コマンド	目的
disable-connected-check 例： `switch(config-router-neighbor)# disable-connected-check`	シングルホップ eBGP ピアが直接接続されているかどうかのチェックを無効にします。このコマンドの使用後、BGP セッションを手動でリセットする必要があります。

eBGP マルチホップの設定

eBGP マルチホップをサポートする eBGP 存続可能時間（TTL）値を設定できます。eBGP ピアは状況によって、別の eBGP ピアに直接接続されず、リモート eBGP ピアに到達するために複数のホップを必要とします。ネイバーセッションに eBGP TTL 値を設定すると、このようなマルチホップ セッションが可能になります。

eBGP マルチホップを設定するには、ネイバー コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
ebgp-multihop *ttl-value* 例： `switch(config-router-neighbor)# ebgp-multihop 5`	eBGP マルチホップの eBGP TTL を設定します。有効な範囲は 2 ～ 255 です。このコマンドの使用後、BGP セッションを手動でリセットする必要があります。

高速外部フェールオーバーのディセーブル化

通常、BGP ルータと直接接続 eBGP ピア間の接続が失われると、ピアとの eBGP セッションをリセットすることによって、BGP が高速外部フェールオーバーを開始します。この高速外部フェールオーバーをディセーブルにすると、リンク フラップが原因の不安定さを制限できます。

高速外部フェールオーバーをディセーブルにするには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
no fast-external-failover 例： `switch(config-router)# no fast-external-failover`	eBGP ピアの高速外部フェールオーバーをディセーブルにします。このコマンドは、デフォルトでイネーブルになっています。

AS パス属性の制限

AS パス属性で自律システム番号が高いルートを廃棄するように eBGP を設定できます。

AS パス属性で AS 番号の多いルートを廃棄するには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
maxas-limit *number* 例： `switch(config-router)# maxas-limit 50`	AS パス セグメントの番号が指定された上限を超えている eBGP ルートを廃棄します。指定できる範囲は 1 ～ 2000 です。

ローカル AS サポートの設定

ローカル AS 機能では、ルータが実際の AS に加えて、2 番めの自律システム（AS）のメンバーであるように見せることができます。ローカル AS を使用すると、ピアリングの調整を変更せずに 2 つの ISP をマージできます。マージされた ISP 内のルータは、新しい自律システムのメンバになりますが、使用者に対しては古い自律システム番号を使用し続けます。

ローカル AS は正しい eBGP ピアにしか使用できません。別のコンフェデレーションのサブ自律システムのメンバである 2 ピアに対しては、この機能は使用できません。

eBGP ローカル AS のサポートを設定するには、ネイバー コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
local-as *number* [**no-prepend** [**replace-as** [**dual-as**]]] 例： `switch(config-router-neighbor)# local-as 1.1`	AS_PATH 属性の前に **local AS** *number* を付加するよう eBGP を設定します。 **local AS** *number* としては 16 ビット整数または 32 ビット整数が可能です。32 ビットの場合、上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式にします。 **no-prepend** キーワードは、**local AS** *number* とピアリングしているパートナーを除き、**local AS** *number*がダウンストリーム BGP ネイバーの前に付加されないようにします。 **replace-as** キーワードは、ピアリング セッションの **local AS** *number* だけが AS_PATH 属性の前に付加されるようにします。ローカル BGP ルーティング プロセスからの自律システム番号は、プリペンドされません。 **dual-as** キーワードは、eBGP ネイバーを構成し、実際の自律システム番号（ローカルの BGP ルーティング プロセスからのもの）またはローカル AS として構成された自律システム番号を使用して、ピアリング セッションを確立するようにします。

BGP 属性フィルタリングの設定とエラー処理

Cisco NX-OS リリース 9.3(3) 以降では、BGP属性フィルタリングとエラー処理を設定して、セキュリティ レベルを向上させることができます。次の機能を利用でき、次の順序で実装されます。

- **パス属性 treat-as-withdraw:** アップデートに指定した属性タイプが含まれている場合に、指定したネイバーから受け取った BGP アップデートを treat-as-withdraw とすることを許可します。アップデートに含まれるプレフィックスは、ルーティング テーブルから削除されます。
- **パス属性 discard:**BGP アップデートの特定のパス属性を特定のネイバーから削除できます。
- **拡張属性エラー処理:** 形式が誤っているアップデートに起因するピア セッションのフラッピングを防止します。

属性タイプ 1、2、3、4、8、14、15、16 は、パス属性 treat-as-withdraw とパス属性 discard に対して設定できません。属性タイプ 9（Originator）、タイプ 10（Cluster-id）は、eBGP ネイバーでのみ設定できます。

BGP 更新メッセージからのパス属性の取り消しとしての処理

特定のパス属性を含むBGP更新を「扱うように」処理するには、ルータネイバーコンフィギュレーションモードで次のコマンドを使用します。

手順

	コマンドまたはアクション	目的

ステップ 1	[**no**] **path-attribute treat-as-withdraw** [*value* | **range** *start end*] **in**
	例：
	`switch#(config-router)# neighbor 10.20.30.40`
	`switch(config-router-neighbor)# path-attribute treat-as-withdraw 100 in`
	例：
	`switch#(config-router)# neighbor 10.20.30.40`
	`switch(config-router-neighbor)# path-attribute treat-as-withdraw range 21 255 in`
		指定されたパス属性またはパス属性の範囲を含む着信BGP更新メッセージをすべて取り消すものとして扱い、ルーティングテーブルが最新であることを確認するために着信ルートリフレッシュをトリガーします。treat-as-withdraw である BGP 更新のプレフィックスは、BGP ルーティング テーブルから削除されます。
		このコマンドは、BGP テンプレート ピアおよびBGP テンプレート ピア セッションでもサポートされます。

BGP 更新メッセージからのパス属性の破棄

特定のパス属性を含む BGP アップデートを廃棄するには、ルータ ネイバー コンフィギュレーション モードで次のコマンドを使用します。

手順

	コマンドまたはアクション	目的

ステップ 1	[**no**] **path-attribute discard** [*value* | **range** *start end*] **in**
	例：
	`switch#(config-router)# neighbor 10.20.30.40`
	`switch(config-router-neighbor)# path-attribute discard 100 in`
	例：
	`switch#(config-router)# neighbor 10.20.30.40`
	`switch(config-router-neighbor)# path-attribute discard range 100 255 in`
		指定されたネイバーの BGP アップデート メッセージ内の指定されたパス属性をドロップし、ルーティング テーブルが最新であることを確認するために着信ルート リフレッシュをトリガーします。特定の属性または不要な属性の範囲全体を設定できます。
		このコマンドは、BGP テンプレート ピアおよび BGP テンプレート ピア セッションでもサポートされます。
		（注） discard と treat-as-withdaw の両方に同じパス属性が設定されている場合、treat-as-withdaw の優先順位が高くなります。

拡張属性エラー処理のイネーブル化またはディセーブル化

BGP 拡張属性エラー処理はデフォルトで有効になっていますが、無効にすることもできます。この機能は、RFC 7606 に準拠しており、不正な更新によるピア セッションのフラッピングを防止します。デフォルトの動作は、eBGP ピアと iBGP ピアの両方に適用されます。

拡張エラー処理を無効または再度有効にするには、ルータ設定モードで次のコマンドを使用します。

手順

	コマンドまたはアクション	目的

ステップ 1	[**no**] **enhanced-error**
	例：
	`switch(config)# router bgp 1000`
	`switch(config-router)# enhanced-error`
		BGP 拡張属性エラー処理をいネーブルまたはディセーブルにします。

取り消されたパス属性または破棄されたパス属性の表示

廃棄または不明なパス属性に関する情報を表示するには、次のいずれかのタスクを実行します。

コマンド	目的
show bgp {**ipv4** | **ipv6**} **unicast path-attribute discard**]	属性が破棄されたすべてのプレフィックスを表示します。
show bgp {**ipv4** | **ipv6**} **unicast path-attribute unknown**]	不明な属性を持つすべてのプレフィックスを表示します。

コマンド	目的
show bgp {**ipv4** | **ipv6**} **unicast** *ip-address*	プレフィックスに関連付けられている不明な属性および破棄された属性を表示します。

次の例は、属性が廃棄されたプレフィックスを示しています。

```
switch# show bgp ipv4 unicast path-attribute discard
Network            Next Hop
1.1.1.1/32         20.1.1.1
1.1.1.2/32         20.1.1.1
1.1.1.3/32         20.1.1.1
```

次の例は、不明な属性を持つプレフィックスを示しています。

```
switch# show bgp ipv4 unicast path-attribute unknown
Network            Next Hop
2.2.2.2/32         20.1.1.1
2.2.2.3/32         20.1.1.1
```

次の例は、プレフィックスに関連付けられている不明な属性および破棄された属性を表示します。

```
switch# show bgp ipv4 unicast 2.2.2.2
BGP routing table entry for 2.2.2.2/32, version 6241
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  1000
    20.1.1.1 from 20.1.1.1 (20.1.1.1)
      Origin IGP, localpref 100, valid, external, best
      unknown transitive attribute: flag 0xE0 type 0x62 length 0x64
        value 0000 0000 0100 0000 0200 0000 0300 0000
              0400 0000 0500 0000 0600 0000 0700 0000
              0800 0000 0900 0000 0A00 0000 0B00 0000
              0C00 0000 0D00 0000 0E00 0000 0F00 0000
              1000 0000 1100 0000 1200 0000 1300 0000
              1400 0000 1500 0000 1600 0000 1700 0000
              1800 0000
      rx pathid: 0, tx pathid: 0x0
      Updated on Jul 20 2019 07:50:43 PST
```

AS 連合の設定

AS連合を設定するには、連合識別情報を指定する必要があります。AS連合内の自律システムグループは、自律システム番号として連合 ID を持つ、1 つの自律システムとして認識されます。

BGP連合 ID を設定するには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
confederation identifier *as-number* 例： `switch(config-router)# confederation identifier 64512`	AS 連合を表す連合 ID を設定します。 各連合には別のサブ自律システム番号があり、通常は専用番号です（64512 ～ 65534）。 このコマンドによって、BGP ネイバー セッションの自動通知およびセッション リセットが開始されます。

AS 連合に所属する自律システムを設定するには、ルータ コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
bgp confederation peers *as-number* [*as-number2...*] 例： `switch(config-router)# bgp confederation peers 5 33 44`	連合に所属する自律システムのリストを指定します。 このコマンドによって、BGP ネイバーセッションの自動通知およびセッション リセットが開始されます。

独自の自律システムを含む自律システム パスの設定

独自の自律システムを含む自律システム（AS）パスを受け入れる機能を BGP でイネーブルにします。

始める前に

BGP 機能を有効にしていることを確認します（BGP 機能のイネーブル化のセクションを参照してください）。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-number*
4. **address-family ipv4 unicast**
5. [**no** | **default**] **allowas-in** [*allowas-in-cnt*]
6. **end**
7. （任意） **show running-config bgp**
8. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *as-number* 例： `switch(config)# router bgp as-number`	BGP モードを開始し、ローカル BGP スピーカに自律システム番号を割り当てます。*as-number* の値の範囲は 1 ～ 65535 です。

ステップ 3	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65536`
	`switch(config-router-neighbor)#`
		BGP ルーティング用のネイバー設定モードを開始し、ネイバー IP アドレスを設定します。

ステップ 4	**address-family ipv4 unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		指定のアドレス ファミリに対応するルータ アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 5	[**no** | **default**] **allowas-in** [*allowas-in-cnt*] 例： `switch(config-router-neighbor-af)# allowas-in 5`	BGP の allowas-in 機能をイネーブルにし、自律システム番号の発生回数を設定します。 *allowas-in-cnt* には、1 ～ 10 の整数を入力します。デフォルトでは、自律システム番号の発生回数は 3 に設定されます。
ステップ 6	**end** 例： `switch(config-router-neighbor-af)# end`	ルータ アドレス ファミリ コンフィギュレーション モードを終了します。
ステップ 7	（任意） **show running-config bgp** 例： `switch# show running-config bgp`	BGP の設定を表示します。
ステップ 8	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor)# copy running-config startup-config`	この設定変更を保存します。

例

次に、BGP の allowas-in 機能を設定し、ユニキャスト IPv4 アドレス ファミリ用に設定する例を示します。

```
switch# configure terminal
switch(config)# router bgp 77
switch(config-router)# neighbor 6.20.1.1 remote-as 66
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# allowas-in 5
switch(config-router-neighbor-af)# end
```

ルート リフレクタの設定

ルート リフレクタとして動作するローカル BGP スピーカに対するルート リフレクタ クライアントとして、iBGP ピアを設定できます。ルート リフレクタとそのクライアントがともにクラスタを形成します。クライアントからなるクラスタには通常、ルート リフレクタが 1 つ存在します。このような状況では、ルート リフレクタのルータ ID でクラスタを識別します。ネットワークの冗長性を高め、シングル ポイント障害を回避するために、複数のルート リフレクタからなるクラスタを設定できます。クラスタ内のすべてのルート リフレクタは、同じ 4 バイト クラスタ ID で設定する必要があります。これは、ルート リフレクタが同じクラスタ内のルート リフレクタからのアップデートを認識できるようにするためです。

始める前に

BGP 機能を有効にしていることを確認します（BGP 機能のイネーブル化のセクションを参照）。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **cluster-id** *cluster-id*
4. **address-family ipv4 unicast**
5. （任意） **client-to-client reflection**
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **address-family ipv4 unicast**
9. **route-reflector-client**
10. **show bgp ip unicast neighbors**
11. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65536`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに自律システム番号を割り当てます。

ステップ 3	**cluster-id** *cluster-id* 例： `switch(config-router)# cluster-id 192.0.2.1`	クラスタに対応するルート リフレクタの 1 つとして、ローカル ルータを設定します。クラスタを識別するクラスタ ID を指定します。このコマンドによって、BGP ネイバー セッションの自動ソフト クリアまたはリフレッシュが開始されます。

ステップ 4	**address-family ipv4 unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		指定のアドレス ファミリに対応するグローバル アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 5	（任意） **client-to-client reflection** 例： `switch(config-router-af)# client-to-client reflection`	クライアント間のルート リフレクションを設定します。この機能は、デフォルトでイネーブルになっています。このコマンドによって、BGP ネイバー セッションの自動ソフトクリアまたはリフレッシュが開始されます。

ステップ 6	**exit**
	例：
	`switch(config-router-neighbor-af)# exit`
	`switch(config-router-neighbor)#`
		ルータ アドレス コンフィギュレーション モードを終了します。

ステップ 7	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65536`
	`switch(config-router-neighbor)#`
		リモート BGP ピアの IP アドレスおよび AS 番号を設定します。

ステップ 8	**address-family ipv4 unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		指定のアドレス ファミリに対応しネイバー アドレス ファミリ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 9	**route-reflector-client** 例： `switch(config-router-neighbor-af)# route-reflector-client`	BGP ルート リフレクタとしてスイッチを設定し、そのクライアントとしてネイバーを設定します。このコマンドによって、BGP ネイバー セッションの自動通知およびセッション リセットが開始されます。
ステップ 10	**show bgp ip unicast neighbors** 例： `switch(config-router-neighbor-af)# show bgp ip unicast neighbors`	BGP ピアを表示します。
ステップ 11	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor)# copy running-config startup-config`	この設定変更を保存します。

例

次に、ルート リフレクタとしてルータを設定し、クライアントとしてネイバーを 1 つ追加する例を示します。

```
switch(config)# router bgp 65536
switch(config-router)# neighbor 192.0.2.10 remote-as 65536
switch(config-router-neighbor)# address-family ip unicast
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# copy running-config startup-config
```

ルート ダンプニングの設定

iBGP ネットワーク上でのルート フラップの伝播を最小限に抑えるために、ルート ダンプニングを設定できます。

ルート ダンプニングを設定するには、アドレス ファミリまたは VRF アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
dampening [{ *half-life reuse-limit suppress-limit max-suppress-time* | **route-map** *map-name* }] 例： `switch(config-router-af)# dampening route-map bgpDamp`	機能ネゴシエーションをディセーブルにします。パラメータ値は次のとおりです。 • half-life：指定できる範囲は 1 ～ 45 です。 • reuse-limit：指定できる範囲は 1 ～ 20000 です。 • suppress-limit：指定できる範囲は 1 ～ 20000 です。 • max-suppress-time：指定できる範囲は 1 ～ 255 です。

ロード シェアリングおよび ECMP の設定

等コスト マルチパス ロード バランシング用に BGP がルート テーブルに追加するパスの最大数を設定できます。

（注） ECMP はワープ モードではサポートされません。

パスの最大数を設定するには、ルータ アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
maximum-paths [**ibgp**] *maxpaths* 例： `switch(config-router-af)# maximum-paths 12`	ロード シェアリング用の等コスト パスの最大数を設定します。指定できる範囲は 1 ～ 32 です。デフォルトは 1 です。 （注） ECMP 構成を防ぐために、ワープ モードで maximum-path を 1 に設定することをお勧めします。

最大プレフィックス数の設定

BGP が BGP ピアから受け取ることのできるプレフィックスの最大数を設定できます。任意で、プレフィックス数がこの値を超えた場合に、BGP に警告メッセージを生成させる、またはピアとの BGP セッションを切断させることを設定できます。

BPG ピアに認めるプレフィックスの最大数を設定するには、ネイバー アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
maximum-prefix *maximum* [*threshold*] [**restart** *time* | **warming-only**] 例： `switch(config-router-neighbor-af)# maximum-prefix 12`	ピアからのプレフィックスの最大数を設定します。パラメータの範囲は次のとおりです。 • *maximum*：指定できる範囲は 1 ～ 300000 です。 • *threshold*：指定できる範囲は 1 ～ 100 % です。デフォルトは 75% です。 • *time*：指定できる範囲は 1 ～ 65535 分です。 このコマンドによって、プレフィックス限度を超えた場合に、BGP ネイバー セッションの自動通知およびセッション リセットが開始されます。

ダイナミック機能の設定

BGP ピアのダイナミック機能を設定できます。

ダイナミック機能を設定するには、ネイバー コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
dynamic-capability 例： `switch(config-router-neighbor)# dynamic-capability`	ダイナミック機能をイネーブルにします。このコマンドによって、BGP ネイバー セッションの自動通知およびセッション リセットが開始されます。 このコマンドは、デフォルトでディセーブルになっています。

集約アドレスの設定

BGP ルート テーブルの集約アドレス エントリを設定できます。

集約アドレスを設定するには、ルータ アドレス ファミリ コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
aggregate-address *ip-prefix/length* [**as-set**] [**summary-only**] [**advertise-map** *map-name*] [**attribute-map** *map-name*] [**suppress-map** *map-name*] 例： `switch(config-router-af)# aggregate-address 192.0.2.0/8 as-set`	集約アドレスを作成します。このルートに関してアドバタイズされるパスは、集約されているすべてのパスに含まれるすべての要素からなる、自律システム セットです。 • **as-set** キーワードは、関係するパスから自律システム セット パス情報およびコミュニティ情報を生成します。 • **summary-only** キーワードは、アップデートから具体的なルートをすべてフィルタリングします。 • **advertise-map** キーワードおよび引数では、選択されたルートから属性情報を選択するためのルート マップを指定します。 • **attribute-map** キーワードおよび引数では、集約から属性情報を選択するためのルート マップを指定します。 • **suppress-map** キーワードおよび引数では、固有性の強いルートを条件付きでフィルタ処理します。

BGP 条件付きアドバタイズメントの設定

BGP がプロパゲートするルートを制限するように BGP 条件付きアドバタイズメントを設定できます。次の 2 つのルート マップを定義します。

- アドバタイズ マップ：BGP が条件付きアドバタイズメントを考慮する前にルートが一致する必要のある条件を指定します。このルートマップには、適切な match 文を含めることができます。
- 存在マップまたは非存在マップ：BGP がアドバタイズ マップに一致するルートをプロパゲートする前に BGP テーブルに存在する必要のあるプレフィックスを定義します。非存在マップは、BGP がアドバタイズ マップに一致するルートをプロパゲートする前に BGP テーブルに存在してはならないプレフィックスを定義します。BGP は、これらのルートマップでプレフィックス リストの match 文内にある permit 文のみを処理します。

ルートが条件を渡さない場合、そのルートが BGP テーブルにあれば BGP によってルートが取り消されます。

始める前に

BGP 機能を有効にしていることを確認します（BGP 機能のイネーブル化のセクションを参照してください）。

手順の概要

1. **configure terminal**

2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-number*
4. **address-family ipv4 unicast**
5. **advertise-map** *adv-map* { **exist-map** *exist-rmap* | **non-exist-map** *nonexist-rmap* }
6. （任意） **show ip bgp neighbor**
7. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65536`
	`switch(config-router)#`
		BGP モードを開始し、ローカル BGP スピーカに自律システム番号を割り当てます。

ステップ 3	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65537`
	`switch(config-router-neighbor)#`
		BGP ルーティング用のネイバー設定モードを開始し、ネイバー IP アドレスを設定します。

ステップ 4	**address-family ipv4 unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		アドレス ファミリ設定モードを開始します。

ステップ 5	**advertise-map** *adv-map* { **exist-map** *exist-rmap*	**non-exist-map** *nonexist-rmap* } 例： `switch(config-router-neighbor-af)# advertise-map advertise exist-map exist`	2 つの設定済みルート マップに従い、ルートを条件付きでアドバタイズするように BGP を設定します。 • *adv-map*：BGP がルートを次のルート マップに渡す前に、そのルートが渡す必要のある match 文を使用してルート マップを指定します。adv-map には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。 • *exist-rmap*：プレフィックス リストの match ステートメントを使用してルートマップを指定します。BGP テーブル内のプレフィックスは、BGP がルートをアドバタイズする前に、プレフィックスリスト内のプレフィックスと一致する必要があります。exist-rmap には最大 63 文字

	コマンドまたはアクション	目的
		の英数字を使用できます。大文字と小文字は区別されます。 • *nonexist-rmap*：プレフィックス リストの match ステートメントを使用してルートマップを指定します。BGPテーブル内のプレフィックスは、BGP がルートをアドバタイズする前に、プレフィックスリスト内のプレフィックスと一致してはいけません。nonexist-rmap には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。
ステップ 6	（任意） **show ip bgp neighbor** 例： `switch(config-router-neighbor-af)# show ip bgp neighbor`	BGPに関する情報、および設定した条件付きアドバタイズメントのルートマップに関する情報を表示します。
ステップ 7	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor)# copy running-config startup-config`	この設定変更を保存します。

例

次に、BGP 条件付きアドバタイズメントを設定する例を示します。

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# neighbor 192.0.2.2 remote-as 65537
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# advertise-map advertise exist-map exist
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# exit
switch(config-router)# exit
switch(config)# route-map advertise
switch(config-route-map)# match as-path pathList
switch(config-route-map)# exit
switch(config)# route-map exit
switch(config-route-map)# match ip address prefix-list plist
switch(config-route-map)# exit
switch(config)# ip prefix-list plist permit 209.165.201.0/27
```

ルートの再配布の設定

別のルーティング プロトコルからのルーティング情報を受け入れて、BGP ネットワークを通じてその情報を再配布するように、BGP を設定できます。任意で、再配布ルートのためのデフォルト ルートを割り当てることができます。

始める前に

BGP 機能を有効にしていることを確認します（BGP 機能のイネーブル化のセクションを参照してください）。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **address-family ipv4 unicast**
4. **redistribute** { **direct** | { **eigrp** | **ospf** | **ospfv3** | **rip** } *instance-tag* | static } **route-map** *map-name*
5. （任意） **default-metric** *value*
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65536`
	`switch(config-router)#`
		BGP モードを開始し、ローカル BGP スピーカに自律システム番号を割り当てます。

ステップ 3	**address-family ipv4 unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		アドレス ファミリ設定モードを開始します。

ステップ 4	**redistribute** { **direct** | { **eigrp** | **ospf** | **ospfv3** | **rip** } *instance-tag* | static } **route-map** *map-name* 例： `switch(config-router-af)# redistribute eigrp 201 route-map Eigrpmap`	他のプロトコルからのルートを BGP に再配布します。ルートマップの設定の詳細については、ルートマップの設定のセクションを参照してください。
ステップ 5	（任意） **default-metric** *value* 例： `switch(config-router-af)# default-metric 33`	BGP へのデフォルト ルートを生成します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor)# copy running-config startup-config`	この設定変更を保存します。

例

次に、EIGRP を BGP に再配布する例を示します。

```
switch# configure terminal
switch(config)# router bgp 65536
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# redistribute eigrp 201 route-map Eigrpmap
switch(config-router-af)# copy running-config startup-config
```

マルチプロトコル BGP の設定

複数のアドレスファミリ（IPv4 のユニキャストルートを含む）をサポートするように MP-BGP を設定できます。

始める前に

BGP 機能を有効にしていることを確認します（BGP 機能のイネーブル化のセクションを参照してください）。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-number*
4. **address-family ipv4 unicast**
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65536`
	`switch(config-router)#`
		BGP モードを開始し、ローカル BGP スピーカに自律システム番号を割り当てます。

ステップ 3	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65537`
	`switch(config-router-neighbor)#`
		BGP ルーティング用のネイバー設定モードを開始し、ネイバー IP アドレスを設定します。

	コマンドまたはアクション	目的

ステップ 4	**address-family ipv4 unicast**
	例：
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		アドレス ファミリ コンフィギュレーション モードを開始します。

ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor)# copy running-config startup-config`	この設定変更を保存します。

BGP の調整

一連のオプション パラメータを使用することによって、BGP 特性を調整できます。

BGP を調整するには、ルータ コンフィギュレーション モードで次のオプション コマンドを使用します。

コマンド	目的

bestpath [**always-compare-med** | **compare-routerid** | **med** { **missing-as-worst** | **non-deterministic** } | **as-path multipath-relax**]
例：
`switch(config-router)# bestpath always-compare-med`
`switch(config-router)# bestpath as-path multipath-relax`
	ベストパス アルゴリズムを変更します。オプション パラメータは次のとおりです。
	• **always-compare-med**：異なる自律システム（AS）からのパスの MED を比較します。
	• **compare-routerid**：同一の eBGP パスのルータ ID を比較します。
	• **med missing-as-worst**：消失した MED を最高の MED と見なします。
	• **med non-deterministic**：同じ自律システムからのパス間で、必ずしも最適な MED パスを選択しません。
	• **as-path multipath-relax**：AS パスの長さが同じで、他のマルチパスの条件を満たしている場合、別の自律システムから受け取ったパスをマルチパスとして扱えるようにします。

enforce-first-as 例： `switch(config-router)# enforce-first-as`	ネイバー自律システムを eBGP の AS_path 属性で指定する最初の AS 番号にします。

コマンド	目的
log-neighbor-changes 例： `switch(config-router)# log-neighbor-changes`	ネイバーでステートが変化したときに、システム メッセージを生成します。
router-id *id* 例： `switch(config-router)# router-id 209.165.20.1`	この BGP スピーカのルータ ID を手動で設定します。
timers [**bestpath-delay** *delay* | **bgp** *keepalive holdtime* | **prefix-peer-timeout** *timeout*] 例： `switch(config-router)# timers bgp 90 270`	BGP タイマー値を設定します。オプション パラメータは次のとおりです。 このコマンドの設定後、BGP セッションを手動でリセットする必要があります。

BGP を調整するには、ルータ アドレス ファミリ コンフィギュレーション モードで次のオプション コマンドを使用します。

コマンド	目的
distance *ebgp-distance ibgp distance local-distance* 例： `switch(config-router-af)# distance 20 100 200`	BGP のアドミニストレーティブ ディスタンスを設定します。範囲は 1 ～ 255 です。デフォルトの設定は次のとおりです。

BGP を調整するには、ネイバー コンフィギュレーション モードで次のオプション コマンドを使用します。

コマンド	目的
description *string* 例： `switch(config-router-neighbor)# description main site`	この BGP ピアを説明するストリングを設定します。ストリングには最大 80 の英数字を使用できます。
low-memory exempt 例： `switch(config-router-neighbor)# low-memory exempt`	メモリ不足状態によるシャットダウンからこの BGP ネイバーを除外します。

コマンド	目的
transport connection-mode passive 例： `switch(config-router-neighbor)# transport connection-mode passive`	受動接続の確立だけが可能です。この BGP スピーカは BGP ピアへの TCP 接続を開始しません。このコマンドの設定後、BGP セッションを手動でリセットする必要があります。
remove-private-as 例： `switch(config-router-neighbor)# remove-private-as`	eBGP ピアへの発信ルートアップデートからプライベート AS 番号を削除します。このコマンドによって、BGP ネイバーセッションの自動ソフトクリアまたはリフレッシュが開始されます。
update-source *interface-type number* 例： `switch(config-router-neighbor)# update-source ethernet 2/1`	ピアとの BGP セッション用に設定されたインターフェイスの送信元 IP アドレスを使用するように、BGP スピーカを設定します。このコマンドによって、BGP ネイバー セッションの自動通知およびセッション リセットが開始されます。

BGP を調整するには、ネイバー アドレス ファミリ コンフィギュレーション モードで次のオプション コマンドを使用します。

コマンド	目的
suppress-inactive 例： `switch(config-router-neighbor-af)# suppress-inactive`	ベスト（アクティブ）ルートだけを BGP ピアにアドバタイズします。このコマンドによって、BGP ネイバーセッションの自動ソフト クリアまたはリフレッシュが開始されます。
default-originate [**route-map** *map-name*] 例： switch(config-router-neighbor-af)# default-originate	BGP ピアへのデフォルト ルートを作成します。
filter-list *list-name* { **in** | **out** } 例： `switch(config-router-neighbor-af)# filter-list BGPFilter in`	着信または発信ルートアップデートに関して、この BGP ピアに AS_path フィルタ リストを適用します。このコマンドによって、BGP ネイバー セッションの自動ソフト クリアまたはリフレッシュが開始されます。
prefix-list *list-name* { **in** | **out** } 例： `switch(config-router-neighbor-af)# prefix-list PrefixFilter in`	着信または発信ルートアップデートに関して、この BGP ピアにプレフィックス リストを適用します。このコマンドによって、BGP ネイバー セッションの自動ソフトクリアまたはリフレッシュが開始されます。

コマンド	目的
send-community 例： `switch(config-router-neighbor-af)# send-community`	この BGP ピアにコミュニティ属性を送信します。このコマンドによって、BGP ネイバー セッションの自動ソフト クリアまたはリフレッシュが開始されます。
send-extcommunity 例： `switch(config-router-neighbor-af)# send-extcommunity`	この BGP ピアに拡張コミュニティ属性を送信します。このコマンドによって、BGP ネイバー セッションの自動ソフト クリアまたはリフレッシュが開始されます。

仮想化の設定

始める前に

BGP 機能を有効にしていることを確認します（BGP 機能の有効化のセクションを参照）。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **exit**
4. **router bgp** *as-number*
5. **vrf** *vrf-name*
6. **neighbor** *ip-address* **remote-as** *as-number*
7. （任意） **bestpath as-path multipath-relax**
8. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context RemoteOfficeVRF`
	`switch(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。

ステップ 3	**exit** 例：	VRF設定モードを終了します。

	コマンドまたはアクション	目的
	`switch(config-vrf)# exit` `switch(config)#`	

ステップ 4	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65536`
	`switch(config-router)#`
		自律システム番号を設定して、新しいBGPプロセスを作成します。

ステップ 5	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf RemoteOfficeVRF`
	`switch(config-router-vrf)#`
		ルータ VRF設定モードを開始し、この BGP インスタンスと VRF を関連付けます。

ステップ 6	**neighbor** *ip-address* **remote-as** *as-number*
	例：
	`switch(config-router-vrf)# neighbor 209.165.201.1 remote-as 65536`
	`switch(config-router--vrf-neighbor)#`
		リモート BGP ピアの IP アドレスおよび AS 番号を設定します。

ステップ 7	（任意） **bestpath as-path multipath-relax** 例： `switch(config-router-vrf)# bestpath as-path multipath-relax`	自律パスの長さが同じで、他のマルチパスの条件を満たしている場合、別の自律システムから受け取ったパスをマルチパスとして扱えるようにします。
ステップ 8	（任意） **copy running-config startup-config** 例： `switch(config-router-neighbor)# copy running-config startup-config`	この設定変更を保存します。

例

次に、VRF を作成し、VRF でルータ ID を設定する例を示します。

```
switch# configure terminal
switch(config)# vrf context NewVRF
switch(config-vrf)# exit
switch(config)# router bgp 65536
switch(config-router)# vrf NewVRF
switch(config-router-vrf)# neighbor 209.165.201.1 remote-as 65536
switch(config-router-vrf-neighbor)# copy running-config startup-config
```

BGP グレースフル シャットダウン

BGP グレース フル シャットダウンに関する情報

リリース 9.3(1) 以降、BGP はグレースフル シャットダウン機能をサポートしています。この BGP 機能は、BGP **shutdown** コマンドと連携します。名前にかかわらず、BGP グレースフル シャットダウンは実際にはシャットダウンを引き起こしません。代わりに、ルータまたはリンクが間もなくダウンすることを、接続されているルータに通知します。

- ルータまたはリンクがオフラインになったときのネットワーク コンバージェンス時間を大幅に短縮します。
- ルータまたはリンクがオフラインになったときに、転送中のドロップされたパケットを削減または排除します。

グレースフル シャットダウン機能は、GRACEFUL_SHUTDOWN ウェルノウン コミュニティ（0xFFFF0000 または 65535:0）を使用します。これは、IANA および IETF によって RFC 8326 によって識別されます。この既知のコミュニティは任意のルートにアタッチでき、ルートの他の属性と同様に処理されます。

この機能は、ルータまたはリンクがダウンすることを通知するため、メンテナンス時間帯または計画停止の準備に役立ちます。トラフィックへの影響を制限するには、BGP をシャットダウンする前にこの機能を使用します。

グレースフル シャットダウンの認識とアクティブ化

BGPルータは、すべてのルートの優先事項を、GRACEFUL SHUTDOWN 対応というコンセプトを通し、GRACEFUL_SHUTDOWN コミュニティによって制御できます。グレースフル シャットダウン対応は、デフォルトでイネーブルになっています。これにより、受信側ピアは、GRACEFUL_SHUTDOWN コミュニティを伝える着信ルートを優先しなくなります。一般的な使用例ではありませんが、**graceful-shutdown aware** コマンドを使用して、グレースフル シャットダウン対応を無効にしてから再度有効にすることもできます。

グレースフル シャットダウン対応は、BGP グローバル コンテキストでのみ適用されます。コンテキストの詳細については、グレースフル シャットダウンのコンテキスト（194 ページ）を参照してください。対応のためのオプションは、**activate** という別のオプションと一緒に動作します。このオプションをルートマップに割り当てると、グレースフル シャットダウンのルートをより詳細に制御できます。

グレースフル シャットダウン対応オプションとアクティブ化オプションの協同作用

グレースフル シャットダウンがアクティブな場合、**activate** キーワードを指定した場合にのみ、GRACEFUL_SHUTDOWN コミュニティがルート更新に追加されます。この時点で、コミュニティを含む新しいルート更新が生成され、送信されます。**graceful-shutdown aware** コマンドが設定されると、コミュニティを受信するすべてのルータは、アップデート内のルートの優先

を解除します（そのルート優先度を下げます）。**graceful-shutdown aware** コマンドを使用しなかった場合、BGPはGRACEFUL_SHUTDOWN コミュニティの設定されたルートの優先度を下げません。

この機能がアクティブになり、ルータがグレースフルシャットダウンの対応状態になった場合でも、BGP は引き続き、GRACEFUL_SHUTDOWN コミュニティが有効だとしてルートを考慮します。ただし、これらのルートには、最適パスの計算で最低の優先度が与えられます。代替パスが使用可能な場合は、新しい最適パスが選択され、まもなくダウンするルータまたはリンクに対応するためのコンバージェンスが行われます。

グレースフル シャットダウンのコンテキスト

BGPのグレースフルシャットダウン機能には、機能の影響と使用可能な機能を決定する2つのコンテキストがあります。

コンテキスト	影響	コマンド
グローバル	スイッチ全体と、スイッチによって処理されるすべてのルート。たとえば、GRACEFUL_SHUTDOWN コミュニティを持つすべてのルートを再アドバタイズします。	**graceful-shutdown activate [route-map** ルート マップ**]** **graceful-shutdown aware**
Peer	BGP ピアまたはネイバー間のリンク。たとえば、ピア間のリンクを 1 つだけ GRACEFUL_SHUTDOWNコミュニティでアドバタイズします。	**graceful-shutdown activate [route-map** ルート マップ**]**

ルート マップによるグレースフル シャットダウン

グレースフル シャットダウンは、ルート ポリシー マネージャ（RPM）機能と連携して、スイッチの BGP ルータが GRACEFUL_SHUTDOWN コミュニティを使用してルートを送受信する方法を制御します。ルート マップは、インバウンドおよびアウトバウンド方向でコミュニティとのルート更新を処理できます。通常、ルートマップは必要ありません。ただし、必要に応じて、グレースフルシャットダウンルートの制御をカスタマイズするために使用できます。

通常のインバウンド ルート マップ

通常のインバウンド ルート マップは、BGP ルータに着信するルートに影響します。ルータはデフォルトでグレースフル シャットダウンを認識するため、通常のインバウンド ルート マップはグレースフル シャットダウン機能では一般的に使用されません。

Cisco NX-OS リリース 9.3 (1) 以降を実行している Cisco Nexus スイッチでは、グレースフル シャットダウン機能のインバウンドルートマップは必要ありません。Cisco NX-OSリリース9.3（1）以降には、BGPルータがグレースフルシャットダウン対応である場合にGRACEFUL_SHUTDOWNコミュニティを持つすべてのルートを自動的に非優先にする、暗黙のインバウンドルート マップがあります。

通常のインバウンド ルート マップは、既知の GRACEFUL_SHUTDOWN コミュニティと一致するように設定できます。これらの着信ルートマップは一般的ではありませんが、使用される場合があります。

- スイッチが 9.3 (1) よりも前の Cisco NX-OS リリースを実行している場合、NX-OS 9.3 (1) には暗黙的なインバウンド ルート マップがありません。これらのスイッチでグレースフル シャットダウン機能を使用するには、グレースフル シャットダウン インバウンド ルート マップを作成する必要があります。ルートマップは、既知のGRACEFUL_SHUTDOWN コミュニティを持つインバウンドルートと一致し、それらを許可し、それらを非優先にする必要があります。着信ルート マップが必要な場合は、9.3 (1) より前のバージョンの NX-OS を実行し、グレースフル シャットダウン ルートを受信しているBGPピアで作成します。
- グレースフル シャットダウン認識をディセーブルにし、一部の BGP ネイバーからの GRACEFUL_SHUTDOWN コミュニティを持つ着信ルートでルータを動作させる場合は、それぞれのピアでインバウンド ルート マップを設定できます。

通常のアウトバウンド ルート マップ

通常のアウトバウンド ルート マップは、BGP ルータが送信するルートの転送を制御します。通常のアウトバウンド ルート マップは、グレースフル シャットダウン機能に影響を与える可能性があります。たとえば、GRACEFUL_SHUTDOWN コミュニティで一致するようにアウトバウンド ルート マップを設定し、属性を設定できます。これは、グレースフル シャットダウン アウトバウンド ルート マップよりも優先されます。

グレースフル シャットダウン アウトバウンド ルート マップ

アウトバウンドグレースフルシャットダウンルートマップは、グレースフルシャットダウン機能のアウトバウンド ルート マップの特定のタイプです。これらはオプションですが、ルート マップに関連付けられているコミュニティ リストがすでにある場合に役立ちます。通常のグレースフル シャットダウン アウトバウンド ルート マップには、特定の属性を設定または変更するための `set` 句のみが含まれています。

アウトバウンド ルート マップは、次の方法で使用できます。

- 既存のアウトバウンド ルート マップをすでに持っている顧客の場合は、より大きいシーケンス番号を持つ新しいエントリを追加し、GRACEFUL_SHUTDOWN ウェルノウンコミュニティで照合し、必要な属性を追加できます。
- **graceful-shutdown activate route-map** *name* オプションを使用してグレースフル シャットダウン アウトバウンドルート マップを使用することもできます。これが一般的な使用例です。

このルート マップには match 句が必要ないため、ルート マップはネイバーに送信されるすべてのルートで一致します。

ルート マップの優先順位

同じルータ上に複数のルートマップが存在する場合は、次の優先順位が適用されて、コミュニティとのルートの処理方法が決定されます。次の例を考慮してください。60のローカル設定を設定する標準の発信ルートマップ名 Redがあるとします。また、Blue という名前のピア グレースフル シャットダウン ルートマップがあり、local-prefが 30 に設定されているとします。ルート更新が処理されると、Red は Blue を上書きするため、ローカル プリファレンスは 60 に設定されます。

- 通常の発信ルート マップは、ピア グレースフル シャットダウン マップよりも優先されます。
- ピア グレースフル シャットダウン マップは、グローバル グレースフル シャットダウン マップよりも優先されます。

注意事項と制約事項

BGP グローバル シャットダウンの制限事項と注意事項は、次のとおりです。

- グレースフル シャットダウン機能は、影響を受けるルータの代替ルートがネットワークに存在する場合にのみ、トラフィック損失を回避するのに役立ちます。ルータに代替ルートがない場合は、GRACEFUL_SHUTDOWN コミュニティを伝送するルートが使用可能な唯一のルートであるため、最適パスの計算に使用されます。この状況では、機能の目的が失われます。
- GRACEFUL_SHUTDOWN コミュニティを送信するには、BGP 送信コミュニティの設定が必要です。
- ルート マップの場合:
  - グローバル ルート マップとネイバー ルート マップが設定されている場合、ネイバー単位のルート マップが優先されます。
  - 発信ルート マップは、グレースフル シャットダウン用に設定されたグローバル ルート マップよりも優先されます。
  - 発信ルート マップは、グレースフル シャットダウン用に設定されたピア ルート マップよりも優先されます。
  - レガシー（既存の）インバウンド ルート マップにグレースフル シャットダウン機能を追加するには、次の手順を実行します。
    1. graceful shutdown match 句をルート マップの先頭に追加します。これには、句に低いシーケンス番号（たとえば、シーケンス番号 0）を設定します。

2. graceful shutdown 句の後に continue ステートメントを追加します。continue ステートメントを省略すると、graceful shutdown 句と一致するルートマップ処理が停止します。シーケンス番号が大きい他の句（たとえば、1以上）は処理されません。

グレースフル シャットダウン タスクの概要

グレースフル シャットダウン機能を使用するには、通常、すべての Cisco Nexus スイッチでグレースフル シャットダウン対応をイネーブルにし、機能をイネーブルのままにします。BGP ルータをオフラインにする必要がある場合は、graceful-shutdown activate を設定します。

次の詳細に、グレースフル シャットダウン機能を使用するためのベスト プラクティスを示します。

ルータまたはリンクをダウンさせるには、次の手順を実行します。

1. グレースフル シャットダウン機能を設定します。
2. ネイバーでベスト パスを確認します。
3. 最適パスが再計算されたら、BGP を無効にする **shutdown** コマンドを発行します。
4. ルータまたはリンクをシャットダウンする必要がある作業を実行します。

ルータまたはリンクをオンラインに戻すには、次の手順を実行します。

1. シャットダウンが必要な作業が完了したら、BGP を再度イネーブルにします（**no shutdown**）。
2. グレースフル シャットダウン機能を無効にします（config モードの **no graceful-shutdown activate**）。

リンクのグレースフル シャットダウンの設定

この作業では、2 つの BGP ルータ間の特定のリンクでグレースフル シャットダウンを設定できます。

始める前に

BGP をまだ有効にしていない場合は、ここで有効にします（**feature bgp**）。

手順の概要

1. **config terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor {** *ipv4-address*|*ipv6-address* **} remote-as** *as-number*
4. **graceful-shutdown activate [route-map** *map-name***]**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**config terminal**
	例：
	`switch-1# configure terminal`
	`switch-1(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**router bgp** *autonomous-system-number*
	例：
	`switch-1(config)# router bgp 110`
	`switch-1(config-router)#`
		ルータ コンフィギュレーションモードを開始して、BGP ルーティング プロセスを作成または設定します。

ステップ 3	**neighbor {** *ipv4-address*|*ipv6-address* **} remote-as** *as-number*
	例：
	`switch-1(config-router)# neighbor 10.0.0.3 remote-as 200`
	`switch-1(config-router-neighbor)#`
		ネイバーが属する自律システム（AS）を設定します。

ステップ 4	**graceful-shutdown activate [route-map** *map-name***]**
	例：
	`switch-1(config-router-neighbor)# graceful-shutdown activate route-map gshutPeer`
	`switch-1(config-router-neighbor)#`
		ネイバーへのリンクでグレースフル シャットダウンを設定します。また、既知の GRACEFUL_SHUTDOWN コミュニティを使用してルートをアドバタイズし、アウトバウンドルート更新にルートマップを適用します。
		ルートは、デフォルトでグレースフル シャットダウン コミュニティでアドバタイズされます。この例では、ルートは gshutPeer という名前のルート マップを使用して、グレースフル シャットダウン コミュニティを持つネイバーにアドバタイズされます。
		gshut コミュニティを受信したデバイスは、ルートのコミュニティを確認し、オプションでコミュニティを使用してルーティング ポリシーを適用します。

GRACEFUL_SHUTDOWN コミュニティに基づく BGP ルートのフィルタリングとローカルプリファレンスの設定

まだ 9.3(1) を実行していないスイッチには、GRACEFUL_SHUTDOWN コミュニティ名と一致するインバウンドルートマップがありません。したがって、正しいルートを識別して先送りする方法はありません。

9.3(1) よりも前のリリースの NX-OS を実行しているスイッチでは、グレースフル シャットダウン（65535:0）のコミュニティ値と一致するインバウンド ルート マップを設定し、ルートを非優先にする必要があります。

スイッチが 9.3(1) 以降を実行している場合、着信ルートマップを設定する必要はありません。

手順の概要

1. **configure terminal**
2. **ip community list standard** *community-list-name* **seq** *sequence-number* **{ permit | deny }** *value*
3. **route map** *map-tag* **{deny | permit}** *sequence-number*
4. **match community** *community-list-name*
5. **set local-preference** *local-pref-value*
6. **exit**
7. **router bgp** *community-list-name*
8. **neighbor {** *ipv4-address|ipv6-address* **}**
9. **address-family {** *address-family sub family* **}**
10. **send community**
11. **route map** *map-tag* **in**

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 terminal**
	例：
	`switch-1# configure terminal`
	`switch-1<config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**ip community list standard** *community-list-name* **seq** *sequence-number* **{ permit | deny }** *value*
	例：
	`switch-1(config)# ip community-list standard GSHUT seq 10 permit 65535:0`
	`switch-1(config)#`
		コミュニティ リストを設定し、よく知られたグレースフル シャットダウン コミュニティ値を持つルートを許可または拒否します。

ステップ 3	**route map** *map-tag* **{deny | permit}** *sequence-number*
	例：
	`switch-1(config)# route-map RM_GSHUT permit 10`
	`switch-1(config-route-map)#`
		ルート マップをシーケンス 10 として設定し、GRACEFUL_SHUTDOWN コミュニティを持つルートを許可します。

ステップ 4	**match community** *community-list-name*
	例：
	`switch-1(config-route-map)# match community GSHUT`
	`switch-1(config-route-map)#`
		IP コミュニティ リスト GSHUT に一致するルートがルート ポリシー マネージャ（RPM）により処理されるように設定します。

ステップ 5	**set local-preference** *local-pref-value*
	例：
	`switch-1(config-route-map)# set local-preference 10`
	`switch-1(config-route-map)#`
		IP コミュニティ リスト GSHUT に一致するルートに、指定されたローカル プリファレンスが与えられるように設定します。

	コマンドまたはアクション	目的

ステップ 6	**exit**
	例：
	`switch-1(config-route-map)# exit`
	`switch-1(config)#`
		ルート マップ設定モードを終了し、グローバル設定モードに戻ります。

ステップ 7	**router bgp** *community-list-name*
	例：
	`switch-1(config)# router bgp 100`
	`switch-1(config-router)#`
		ルータ設定モードを開始し、BGP インスタンスを作成します。

ステップ 8	**neighbor {** *ipv4-address*|*ipv6-address* **}**
	例：
	`switch-1(config-router)# neighbor 10.0.0.3`
	`switch-1(config-router-neighbor)#`
		指定したネイバーのルート BGP ネイバー モードを開始します。

ステップ 9	**address-family {** *address-family sub family* **}**
	例：
	`nxosv2(config-router-neighbor)# address-family ipv4 unicast`
	`nxosv2(config-router-neighbor-af)#`
		ネイバーをアドレス ファミリ（AF）設定モードにします。

ステップ 10	**send community**
	例：
	`nxosv2(config-router-neighbor-af)# send-community`
	`nxosv2(config-router-neighbor-af)#`
		ネイバーとの BGP コミュニティ交換を可能にします。

ステップ 11	**route map** *map-tag* **in**
	例：
	`nxosv2(config-router-neighbor-af)# route-map RM_GSHUT in`
	`nxosv2(config-router-neighbor-af)#`
		ネイバーからの着信ルートにルート マップを適用します。この例では、RM_GSHUT という名前のルート マップは、ネイバーからの GRACEFUL_SHUTDOWN コミュニティを持つルートを許可します。

すべての BGP ネイバーのグレースフル シャットダウンの設定

グレースフル シャットダウン イニシエータのすべてのネイバーに GRACEFUL_SHUTDOWN ウェルノウン コミュニティを手動で適用できます。

すべての BGP ネイバーに対して、グローバル レベルでグレースフル シャットダウンを設定できます。

始める前に

BGP をまだ有効にしていない場合は、ここで有効にします（**feature bgp**）。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **graceful-shutdown activate [route-map** *map-name***]**

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 terminal**
	例：
	`switch-1# configure terminal`
	`switch-1(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**router bgp** *autonomous-system-number*
	例：
	`switch-1(config)# router bgp 110`
	`switch-1(config-router)#`
		ルータ コンフィギュレーション モードを開始して、BGP ルーティング プロセスを作成または設定します。

ステップ 3	**graceful-shutdown activate [route-map** *map-name***]**
	例：
	`switch-1(config-router-neighbor)#`
	`graceful-shutdown activate route-map gshutPeer`
	`switch-1(config-router-neighbor)#`
		すべてのネイバーへのリンクのグレースフル シャットダウン ルート マップを設定します。また、既知の GRACEFUL_SHUTDOWN コミュニティを持つすべてのルートをアドバタイズし、ルートマップをアウトバウンド ルート アップデートに適用します。
		ルートはデフォルトで GRACEFUL_SHUTDOWN コミュニティでアドバタイズされます。この例では、ルートが gshutPeer という名前のルートマップを持つコミュニティを持つすべてのネイバーにアドバタイズされます。ルートマップには set 句のみを含める必要があります。
		GRACEFUL_SHUTDOWN コミュニティを受信したデバイスは、ルートのコミュニティを確認し、オプションでコミュニティを使用してルーティング ポリシーを適用します。

GRACEFUL_SHUTDOWNコミュニティを使用したすべてのルートのプリファレンスの制御

Cisco NX-OS では、GRACEFUL_SHUTDOWN コミュニティを持つ着信ルートの優先順位を下げることができます。**graceful shutdown aware** が有効になっている場合、最適パス計算時に、BGPはコミュニティを伝送するルートを最も低い優先順位と見なします。デフォルトでは、プレファレンスの引き下げが有効になっていますが、このオプションを選択的に無効にすることもできます。

このオプションをイネーブルまたはディセーブルにするたびに、BGPのベストパス計算がトリガーされます。このオプションを使用すると、グレースフルシャットダウンのウェルノウンコミュニティにおける BGP のベストパス計算の動作を柔軟に制御できます。

始める前に

BGPを有効にしていない場合は、ここで有効にします（**feature bgp**）。

手順の概要

1. **configure terminal**
2. **router bgp** *autonoums-system*
3. （任意） **no graceful-shutdown aware**

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 terminal**
	例：
	`switch-1(config)# config terminal`
	`switch-1(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**router bgp** *autonoums-system*
	例：
	`switch-1(config)# router bgp 100`
	`switch-1(config-router)#`
		ルータ コンフィギュレーション モードを開始し、BGP ルーティング プロセスを設定します。

ステップ 3	（任意） **no graceful-shutdown aware**
	例：
	`switch-1(config-router)# no graceful-shutdown aware`
	`switch-1(config-router)#`
		このBGPルータでは、GRACEFUL_SHUTDOWN コミュニティを持つすべてのルートに低い優先順位を指定しないという意味です。グレースフルシャットダウン認識機能がディセーブルになっている場合、デフォルトアクションはルートを非優先にします。そのため、コマンドには **no** 形式というオプションが存在しており、これを使用すると、グレースフルシャットダウン ルートは非優先になりません。

GRACEFUL_SHUTDOWN コミュニティのピアへの送信の防止

発信ルート更新にルート属性として追加されたGRACEFUL_SHUTDOWNコミュニティが不要になった場合は、コミュニティを削除して、指定されたネイバーに送信しなくなります。1 つの使用例は、ルータが自律システム境界にあり、グレースフルシャットダウン機能が自律システム境界の外部に伝播しないようにする場合です。

GRACEFUL_SHUTDOWN がピアに送信されないようにするには、send community オプションを無効にするか、コミュニティを発信ルート マップから削除します。

次の方法の中から 1 つを選択してください。

- 実行コンフィギュレーションで send-community を無効にします。

  例：

  ```
  nxosv2(config-router-neighbor-af)# no send-community standard
  nxosv2(config-router-neighbor-af)#
  ```

  このオプションを使用すると、スイッチはGRACEFUL_SHUTDOWNコミュニティを受信しますが、発信ルート マップを介してダウンストリーム ネイバーに送信されません。すべての標準コミュニティも送信されません。

- 次の手順に従って、発信ルートマップを介してGRACEFUL_SHUTDOWNコミュニティを削除します。

  1. GRACEFUL_SHUTDOWNコミュニティと一致するIPコミュニティ リストを作成します。
  2. GRACEFUL_SHUTDOWN コミュニティと照合する発信ルート マップを作成します。
  3. **set community-list delete** 句を使用して GRACEFUL_SHUTDOWN コミュニティを削除します。

  このオプションを使用すると、コミュニティ リストはGRACEFUL_SHUTDOWNコミュニティと一致し、許可されます。その後、発信ルートマップはコミュニティと照合され、発信ルート マップから削除されます。他のすべてのコミュニティは、問題なく発信ルートマップを通過します。

グレースフル シャットダウン情報の表示

グレースフル シャットダウン機能に関する情報は、次の **show** コマンドで確認できます。

コマンド	アクション
show ip bgp community-list graceful-shutdown	GRACEFUL_SHUTDOWNコミュニティを持つBGPルーティング テーブル内のすべてのエントリを表示します。
show running-config bgp	実行中のBGPのデフォルト設定を示します。
show running-config bgp all	グレースフル シャットダウン機能に関する情報など、実行中のBGP設定のすべての情報を表示します。

コマンド	アクション
show bgp *address-family* **neighbors** *neighbor-address*	機能がピアに設定されている場合、次のように表示されます。 • 指定されたネイバーの graceful-shutdown-activate 機能の状態 • 指定されたネイバーに設定されたグレースフル シャットダウン ルート マップの名前
show bgp process	コンテキストに応じて異なる情報を表示します。 graceful-shutdown-activate オプションがピア コンテキストで設定されている場合、`graceful-shutdown-active` を介して機能の有効または無効状態を示します。 graceful-shutdown-activate オプションがグローバル コンテキストで設定され、graceful-shutdown ルート マップがある場合は、次のように機能の有効状態が表示されます。 • `graceful-shutdown-active` • `graceful-shutdown-aware` • `graceful-shutdown route-map`
show ip bgp *address*	指定されたアドレスについて、次を含む BGP ルーティング テーブル情報を表示します。 • 最適パスとして指定されたアドレスの状態 • 指定されたアドレスが GRACEFUL_SHUTDOWN コミュニティの一部であるかどうか

グレースフル シャットダウンの設定例

次に、グレースフル シャットダウン機能を使用するための設定例を示します。

BGP リンクのグレースフル シャットダウンの設定

次に、ローカル プリファレンスとコミュニティを設定しながらグレースフル シャットダウンを設定する例を示します。

- 指定されたネイバーへのリンクのグレースフル シャットダウン アクティブ化の設定
- ルートへの GRACEFUL_SHUTDOWN コミュニティの追加
- コミュニティとのアウトバウンド ルートに対して set 句のみを使用して gshutPeer という名前のルートマップを設定します。

```
router bgp 100
    neighbor 20.0.0.3 remote-as 200
         graceful-shutdown activate route-map gshutPeer
         address-family ipv4 unicast
              send-community

route-map gshutPeer permit 10
    set local-preference 0
    set community 200:30
```

All-Neighbor BGP リンクのグレースフル シャットダウンの設定

次に例を示します。

- ローカル ルータとそのすべてのネイバーを接続するすべてのリンクに対してグレース フルシャットダウン アクティブ化を設定します。
- GRACEFUL_SHUTDOWN コミュニティをルートに追加しています。
- すべての発信ルートに対して set 句のみを使用して gshutAall という名前のルートマップを設定します。

```
router bgp 200
   graceful-shutdown activate route-map gshutAll

route-map gshutAll permit 10
   set as-path prepend 10 100 110
   set community 100:80

route-map Red permit 10
   set local-pref 20

router bgp 100
   graceful-shutdown activate route-map gshutAll
      router-id 2.2.2.2
         address-family ipv4 unicast
         network 2.2.2.2/32
         neighbor 1.1.1.1 remote-as 100
         update-source loopback0
         address-family ipv4 unicast
            send-community
         neighbor 20.0.0.3 remote-as 200
         address-family ipv4 unicast
            send-community
               route-map Red out
```

この例では、ネイバー 1.1.1.1 に対して `gshutAll` ルート マップが有効になりますが、ネイバー 20.0.0.3 で設定された発信ルートマップ `Red` が優先されるため、ネイバー 20.0.0.3 に対しては有効になりません。

ピアテンプレートでのグレースフル シャットダウンの設定

この例では、ピアセッションテンプレートでグレースフルシャットダウン機能を設定します。これはネイバーによって継承されます。

```
router bgp 200
   template peer-session p1
      graceful-shutdown activate route-map gshut_out
   neighbor 1.1.1.1 remote-as 100
      inherit peer-session p1
      address-family ipv4 unicast
         send-community
```

GRACEFUL_SHUTDOWN コミュニティの使用およびインバウンドルートマップに基づく BGP ルートのフィルタリングとローカル プリファレンスの設定

次に、コミュニティ リストを使用して、GRACEFUL_SHUTDOWN コミュニティを持つ着信ルートをフィルタリングする例を示します。この設定は、Cisco NX-OS 9.3(1) を最小バージョンとして実行していないレガシー スイッチに役立ちます。

次に例を示します。

- GRACEFUL_SHUTDOWN コミュニティを持つルートを許可する IP コミュニティ リスト。
- RM_GSHUT という名前のルート マップは、GSHUT という名前の標準コミュニティ リストに基づいてルートを許可します。
- また、ルートマップは、処理するルートの優先順位を 0 に設定します。これにより、ルータがオフラインになったときに、それらのルートに最適パス計算の優先順位が低くなります。ネイバー (20.0.0.2) からの着信 IPv4 ルートにルート マップが適用されます。

```
ip community-list standard GSHUT permit 65535:0

route-map RM_GSHUT permit 10
   match community GSHUT
   set local-preference 0

router bgp 200
   neighbor 20.0.0.2 remote-as 100
      address-family ipv4 unicast
         send-community
         route-map RM_GSHUT in
```

拡張 BGP の設定の確認

BGP の設定情報を表示するには、次のいずれかの作業を行います。

コマンド	目的
show bgp all [**summary**] [**vrf** *vrf-name*]	すべてのアドレス ファミリについて、BGP 情報を表示します。
show bgp convergence [**vrf** *vrf-name*]	すべてのアドレス ファミリについて、BGP 情報を表示します。

コマンド	目的
show bgp ip {**unicast**} [*ip-address*] **community** {**regexp** *expression* | [**community**] [**no-advertise**] [**no-export**] [**no-export-subconfed**]} [**vrf** *vrf-name*]	BGP コミュニティと一致する BGP ルートを表示します。
show bgp [**vrf** *vrf-name*] **ip** {**unicast**} [*ip-address*] **community-list** list-name [**vrf** *vrf-name*]	BGP コミュニティ リストと一致する BGP ルートを表示します。
show bgp ip {**unicast**} [*ip-address*] **extcommunity** {**regexp** *expression* | **generic** [**non-transitive** | **transitive**] *aa4:nn* [**exact-match**]} [**vrf** *vrf-name*]	BGP 拡張コミュニティと一致する BGP ルートを表示します。
show bgp ip {**unicast**} [ip-address] **extcommunity-list** *list-name* [**exact-match**] [**vrf** *vrf-name*]	BGP拡張コミュニティ リストと一致するBGPルートを表示します。
show bgp ip {**unicast**} [*ip-address*] {**dampening** *dampened-paths* [**regexp** *expression*]} [**vrf** *vrf-name*]	BGP ルート ダンプニングの情報を表示します。ルート フラップ ダンプニング情報を消去するには、**clear bgp dampening** コマンドを使用します。
show bgp ip {**unicast**} [*ip-address*] **history-paths** [**regexp** *expression*] [**vrf** *vrf-name*]	BGP ルート ヒストリ パスを表示します。
show bgp ip {**unicast**} [*ip-address*] **filter-list** *list-name* [**vrf** *vrf-name*]	BGP フィルタ リストの情報を表示します。
show bgp ip {**unicast**} [*ip-address*] **neighbors** [*ip-address*] [**vrf** *vrf-name*]	BGPピアの情報を表示します。これらのネイバーを消去するには、**clear bgp neighbors** コマンドを使用します。
show bgp ip {**unicast**} [*ip-address*] {**nexthop** | **nexthop-database**} [**vrf** *vrf-name*]	BGPルート ネクスト ホップの情報を表示します。
show bgp paths	BGP パス情報を表示します。
show bgp ip {**unicast**} [*ip-address*] **policy name** [**vrf** *vrf-name*]	BGPポリシー情報を表示します。ポリシー情報を消去するには、**clear bgp policy** コマンドを使用します。
show bgp ip {**unicast**} [*ip-address*] **prefix-list** *list-name* [**vrf** *vrf-name*]	プレフィックス リストと一致する BGP ルートを表示します。
show bgp ip {**unicast**} [*ip-address*] **received-paths** [**vrf** *vrf-name*]	ソフト再構成用に保管されているBGPパスを表示します。

コマンド	目的
show bgp ip {**unicast**} [*ip-address*] **regexp** *expression* [**vrf** *vrf-name*]	AS_path 正規表現と一致する BGP ルートを表示します。
show bgp ip {**unicast**} [*ip-address*] **route-map** *map-name* [**vrf** *vrf-name*]	ルート マップと一致する BGP ルートを表示します。
show bgp peer-policy *name* [**vrf** *vrf-name*]	BGP ピア ポリシー情報を表示します。
show bgp peer-session *name* [**vrf** *vrf-name*]	BGP ピア セッション情報を表示します。
show bgp peer-template *name* [**vrf** *vrf-name*]	BGP ピア テンプレート情報を表示します。ピア テンプレートのすべてのネイバーを消去するには、**clear bgp peer-template** コマンドを使用します。
show bgp process	BGP プロセス情報を表示します。
show ip bgp *options*	BGP のステータスと構成情報を表示します。このコマンドには複数のオプションがあります。詳細については、Cisco Nexus 3000 シリーズ コマンド リファレンスを参照してください。
show ip mbgp *options*	BGP のステータスと構成情報を表示します。このコマンドには複数のオプションがあります。詳細については、Cisco Nexus 3000 シリーズ コマンド リファレンスを参照してください。
show running-configuration bgp	現在実行中の BGP コンフィギュレーションを表示します。

BGP 統計情報の表示

BGP の統計情報を表示するには、次のコマンドを使用します。

コマンド	目的
show bgp ip {**unicast**} [*ip-address*] **flap-statistics** [**vrf** *vrf-name*]	BGP ルート フラップの統計情報を表示します。これらの統計情報をクリアするには、**clear bgp flap-statistics** コマンドを使用します。
show bgp sessions [**vrf** *vrf-name*]	すべてのピアの BGP セッションを表示します。これらの統計情報をクリアするには、**clear bgp sessions** コマンドを使用します。

コマンド	目的
show bgp sessions [**vrf** *vrf-name*]	すべてのピアの BGP セッションを表示します。これらの統計情報をクリアするには、**clear bgp sessions** コマンドを使用します。
show bgp statistics	BGP 統計情報を表示します。

関連項目

BGP の詳細については、次の項目を参照してください。

- 基本的 BGP の設定
- Route Policy Manager の設定

その他の参考資料

BGP の実装に関連する詳細情報については、次の項を参照してください。

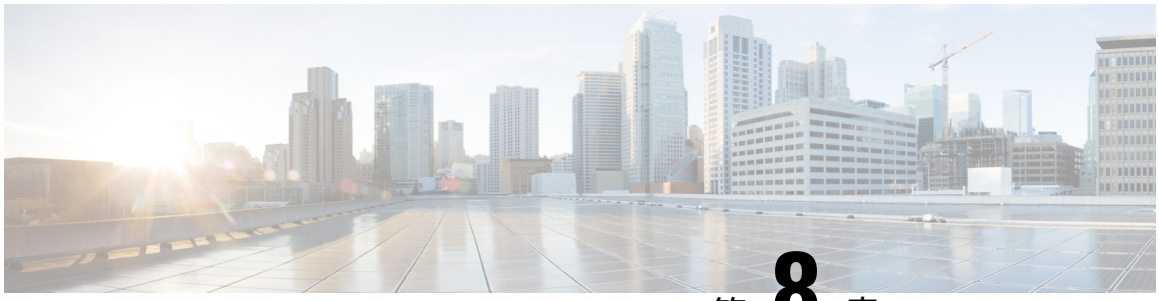
- 関連資料
- MIB

関連資料

関連項目	マニュアル タイトル
BGP CLI コマンド	『Cisco Nexus 3000 Series Command Reference』

MIB

MIB	MIB のリンク
BGP4-MIB CISCO-BGP4-MIB	MIB を検索してダウンロードするには、次の MIB ロケータに移動します。



第 8 章

BGP 追加パスの設定

この章では、以前のパスを新しいパスで暗黙的に置き換えずに、同じプレフィックスの同じピアリング セッションを介したマルチパスのアドバタイズメントを可能にする BGP 追加パスの設定方法について説明します。この動作により、パス ダイバーシティが向上し、Multi-Exit Discriminator（MED）の変動が減少します。

この章は、次の項で構成されています。

BGP 追加パスについて

このセクションは、次のトピックで構成されています。

追加パスで解決できる問題

BGP ルータおよびルート リフレクタ（RR）は、セッションにおけるベスト パスにのみ伝播します。プレフィックス アドバタイズメントで、以前アナウンスされたプレフィックスを置き換えます（この動作は暗黙の取り消しとして知られています）。暗黙の取り消しはスケーリングには適していますが、パス ダイバーシティに影響があります。

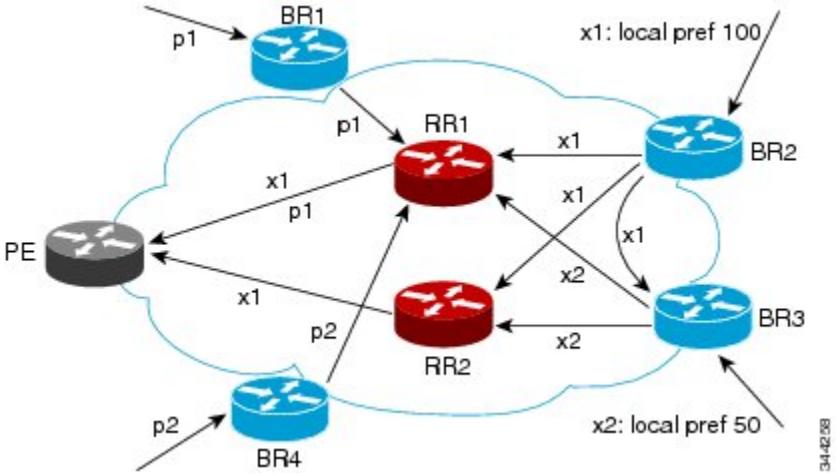
パスの隠蔽は BGP マルチパスの効率的な使用や、スムーズな定期メンテナンスを妨げ、MED の変動や最適でないホットポテト ルーティングが発生する可能性があります。ネクスト ホップが失敗した場合も、ネットワークは BGP コントロール プレーンのコンバージェンスによりトラフィックが復旧するのを待たなければならないので、パスの隠蔽は迅速かつローカルの復旧の妨げになります。BGP 追加パス機能では、パス ダイバーシティを一般的な方法で提供します。Best External または Best Internal 機能は、限られた場合にのみパス ダイバーシティを提供します。

BGP追加パス機能は、同じプレフィックスのマルチパスに対して、新しいパスで以前のパスを暗黙的に置き換えることなく、アドバタイズする手段を提供します。したがって、パスを隠蔽しないでパス ダイバーシティが実現されます。

パスの隠蔽の例

ここでは、パスの隠蔽が発生する過程の詳細を説明します。次の図では、BR1 および BR4 から RR1 にアドバタイズされるプレフィックス p を持つパス p1 および p2 があります。RR1 は 2 つのうちベスト パスを選択し、PE に p1 のみアドバタイズします。

図 *13 : RR* で追加パスを非表示にする

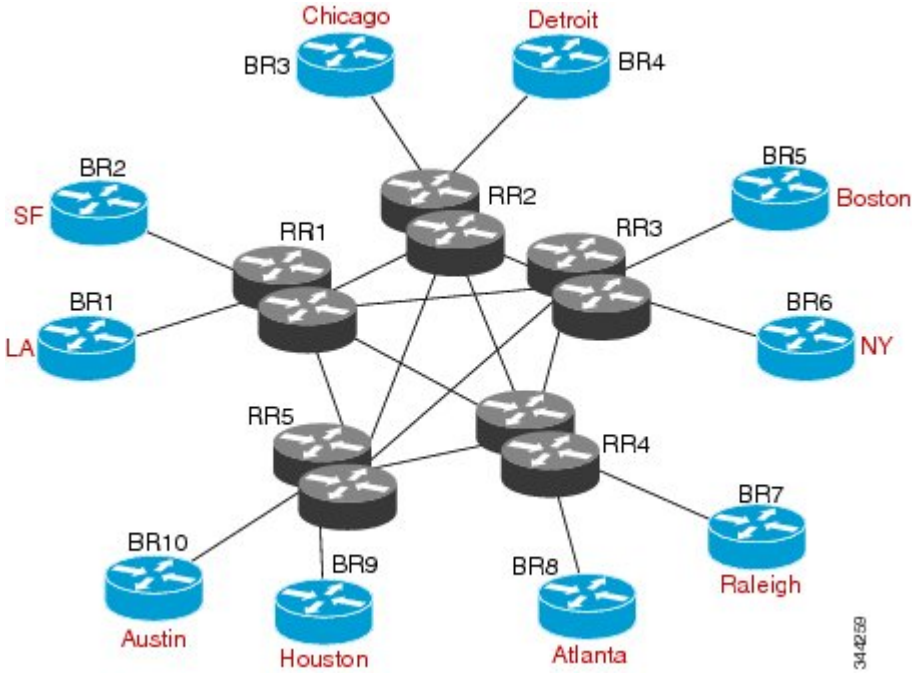


また**追加のパスを隠している RR** の図では、BR2 から（パス x2 がある）BR3 にローカル プリファレンス 100 でアドバタイズされる、プレフィックス x を持つパス x1 が表示されています。BR3 にはパス x2 もありますが、ルーティング ポリシーにより、x2 ではなく RR の x1（表示されていません）をアドバタイズし、x2 のアドバタイズは抑制されます。ユーザーは BR3 で最良外部のアドバタイズメントを有効にして RR に x2 をアドバタイズできますが、この場合も RR はベスト パスのみをアドバタイズします。

最適ではないのホット ポテト ルーティングの例

内部転送コストを最小化するために、中継する ISP は（IGP コストに基づいて）最も近い出口ポイントにパケットを転送しようとします。この動作は、ホット ポテト ルーティングと呼ばれます。次の図の分散 RR クラスタ モデルでは、ロサンゼルスから発信されたトラフィックがメキシコに進む必要があることを想定しています。すべてのリンクで、IGP コストは同じです。メキシコへの出口ポイントは 2 つあり、1 つがオースティンに向かい、もう 1 つがアトランタに向かう場合、ロサンゼルスからは、アトランタよりオースティンに向かう方が IGP コストが低いため、オースティンに向けてトラフィックを送信します。RR3 がある（および RR1、RR2、RR4 および RR5 がない）場所に中央 RR が存在する集中型 RR モデルでは、RR3 から見てメキシコへの最も近い出口ポイントはアトランタとなります。ロサンゼルスからアトランタの BR に向けてトラフィックを送信し、それによって最適ではないホット ポテト ルーティングが生じることは望ましくありません。

図 *14* : 分散 *RR* クラスタ



BGP 追加パスの利点

BGP ルータおよびルート リフレクタ（RR）は、セッションにおけるベスト パスにのみ伝播します。プレフィックス アドバタイズメントで、以前アナウンスされたプレフィックスを置き換えます（この動作は暗黙の取り消しとして知られています）。

この動作は、スケーリングには適していますが、パス ダイバーシティを妨げる可能性があります（これによって脆弱になるまたは完全に無くなるおそれがあります）。同様にこの動作は、BGP マルチパスの効率的な使用や、スムーズな定期メンテナンスを妨げ、MED の変動や最適でないホットポテト ルーティングが発生する可能性があります。ネクスト ホップが失敗した場合も、ネットワークは BGP コントロール プレーンのコンバージェンスによりトラフィックが復旧するのを待たなければならないので、迅速かつローカルの復旧の妨げになります。

BGP 追加パス機能は、暗黙的に以前のパスに代わる新しいパスなしで、同じプレフィックスのマルチパスをアドバタイズする BGP の拡張機能です。これにより、パス ダイバーシティが向上し、MED の変動が減少します。

BGP 追加パスの機能

BGP 追加パス機能は、NLRI で各パスにパス ID を追加することによって実現します。パス ID は VPN のルート識別子（RD）のようなものです。ただし、パス ID はすべてのアドレス ファミリに適用できます。パス ID はピアリング セッション内で一意で、各ネットワークに生成されます。ルート アナウンスが暗黙的に以前のパスを取り消すことを防ぐために、パス ID が使用されます。追加パス機能は、ベスト パスに加えその他のパスのアドバタイズメントが可能で

す。追加パスは、暗黙的に以前のパスから新しいパスに代わることなく、同じプレフィックスのマルチパスをアドバタイズする機能を備えています。

BGP 追加パス機能を使用する場合は、次の 3 つの一般的な手順を実行する必要があります。

1. デバイスが追加パスを送信、受信、または送受信するかどうかを指定します。これらはアドレス ファミリ レベルまたはネイバー レベルで行われます。セッションの確立中に、2 つの BGP ネイバーが追加パス機能（送信または受信のどちらか一方、あるいは両方を実行できるか）についてネゴシエートします。
2. 選択基準を指定して、アドバタイズメントする候補パスのセット選択します。
3. 示された候補パスから追加パスのセットをネイバーに対してアドバタイズします。

追加パスを送受信するには、追加パス機能をネゴシエートする必要があります。ネゴシエートしない場合、選択基準によりベストパス以上のパスが指定され、ネイバーが指定されたパスをアドバタイズするように設定されていても、ネゴシエートできないために選択パスは利用されず、ベストパスのみ送信されます。

追加パスの送受信を BGP に設定すると、デバイスのピアに対して追加パス機能のネゴシエーションが開始されます。この機能についてネゴシエートしたネイバーは、（他のアップデート グループ ポリシーが許可する場合）アップデート グループに追加され、この機能についてネゴシエートされていないピアとは別のアップデート グループに分類されます。したがって、追加パス機能によってネイバーのアップデート グループ メンバーシップが再計算されます。

追加パスの選択

受信機能がイネーブルの場合、追加パスとしてすべての BGP パスをアドバタイズする **set path-selection all advertise** コマンドを設定しない限り、最適パスのみピアにアドバタイズされます。

選択したパスの一部をアドバタイズ

パスのセットを選択する際に、別のパスのセットをアドバタイズしたい場合は注意してください。アドバタイズするパスのセットが、選択されたパスのサブセットではない場合、意図したパスがアドバタイズされません。

注意事項と制約事項

BGP の追加パスの設定には次のガイドラインと制約事項があります。

- BGP 追加パスはダイナミックな機能としてはサポートされていません。これは OPEN に含まれますが、CAPABILITY メッセージには含まれません。設定は次のセッション確立時に有効となります。確立されたセッションが中断されることはありません。

BGP 追加パスの設定方法

このセクションは、次のトピックで構成されています。

アドレス ファミリごとの追加パスの設定

デバイスがアドレス ファミリ内のすべてのネイバーとの間で追加パスを送受信をできるかどうか指定するには、次の手順を実行します。

始める前に

BGP 機能が有効になっていることを確認します

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **address-family ipv4 unicast**
4. （任意） **additional-paths receive**
5. （任意） **additional-paths send**
6. （任意） **additional-paths selection route-map**
7. （任意） **end**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65000`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに自律システム番号を割り当てます。

ステップ 3	**address-family ipv4 unicast** 例： `switch(config-router)# address family ipv4 unicast`	アドレス ファミリ設定モードを開始します。
ステップ 4	（任意） **additional-paths receive** 例：	使用可能なピアから受信するプレフィックスの BGP 追加パスをイネーブルにします。

	コマンドまたはアクション	目的
	`switch(config-router-af)# additional-paths receive`	（注） この機能は、ネイバーで additional-paths receive disable コマンドによって明示的に無効にされない限り、指定されたアドレス ファミリのすべてのネイバーに適用され、そしてアドレス ファミリの設定が上書きされます。
ステップ 5	（任意） **additional-paths send** 例： `switch(config-router-af)# additional-paths send`	使用可能なピアに送信するプレフィックスの BGP 追加パスをイネーブルにします。 （注） この機能は、ネイバーで additional-paths send disable コマンドによって明示的に無効にされない限り、指定されたアドレス ファミリのすべてのネイバーに適用され、そしてアドレス ファミリの設定が上書きされます。

ステップ 6	（任意） **additional-paths selection route-map**
	例：
	`switch(config-router-stmp)# exit`
	`switch(config-router)#`
		プレフィックスの追加パス選択機能を設定します。

ステップ 7	（任意） **end** 例： `switch(config-router-af)# end`	特権 EXEC モードに戻ります。

ネイバーごとの追加パスの設定

特定のネイバーが追加のパスを送受信できるかどうかを設定するには、次の手順を実行します。

始める前に

BGP 機能を有効にしていることを確認します（BGP 機能のイネーブル化のセクションを参照してください）。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** { *ipv4-address* | *ipv4-prefix/length* } [**remote-as** { *as-num* } [. *as-num*]]
4. **address-family ipv4 unicast**
5. （任意） **capability additional-paths receive [disable]**
6. （任意） **capability additional-paths send [disable]**

7. （任意） **end**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65000`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに自律システム番号を割り当てます。

ステップ 3	**neighbor** { *ipv4-address* | *ipv4-prefix/length* } [**remote-as** { *as-num* } [. *as-num*]]	BGP ネイバー（ルータ、VRF）を設定し、ネイバー コンフィギュレーション モードを開始します。
ステップ 4	**address-family ipv4 unicast** 例： `switch(config-router)# address family ipv4 unicast`	アドレス ファミリ設定モードを開始します。
ステップ 5	（任意） **capability additional-paths receive [disable]** 例： `switch(config-router-af)# capability additional-paths receive`	指定されたネイバーの追加パス受信機能を設定します。 （注） このコマンドは、アドレス ファミリのレベルで設定されたすべての送受信機能を上書きします。
ステップ 6	（任意） **capability additional-paths send [disable]** 例： `switch(config-router-af)# capability additional-paths send`	指定されたネイバーの追加パス送信機能を設定します。 （注） このコマンドは、アドレス ファミリのレベルで設定されたすべての送受信機能を上書きします。
ステップ 7	（任意） **end** 例： `switch(config-router-af)# end`	特権 EXEC モードに戻ります。

ピア ポリシー テンプレートを使用した追加パスの設定

この設定作業では、追加パスを送受信する機能および選択基準をアドレス ファミリに設定してから、テンプレートを設定します。

始める前に

BGP 機能を有効にしていることを確認します（BGP 機能のイネーブル化のセクションを参照してください）。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **template peer-policy** *template-name*
4. （任意） **capability additional-paths receive [disable]**
5. （任意） **capability additional-paths send [disable]**
6. **exit**
7. **neighbor** { *ipv4-address* | *ipv4-prefix/length* } [**remote-as** { *as-num* } [. *as-num*]]
8. （任意） **address-family ipv4 unicast**
9. **inherit peer-policy** *template-name sequence-number*
10. （任意） **end**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 65000`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに自律システム番号を割り当てます。

ステップ 3	**template peer-policy** *template-name* 例： `switch(config-router)# template peer-policy rr-client-ptl #`	ポリシー テンプレート コンフィギュレーション モードを開始し、ピア ポリシー テンプレートを作成します。
ステップ 4	（任意） **capability additional-paths receive [disable]** 例： `switch(config-router-af)# capability additional-paths receive`	指定されたネイバーの追加パス受信機能を設定します。 （注） このコマンドは、アドレス ファミリのレベルで設定されたすべての送受信機能を上書きします。
ステップ 5	（任意） **capability additional-paths send [disable]** 例：	指定されたネイバーの追加パス送信機能を設定します。

	コマンドまたはアクション	目的
	`switch(config-router-af)# capability additional-paths send`	（注） このコマンドは、アドレス ファミリのレベルで設定されたすべての送受信機能を上書きします。
ステップ 6	**exit** 例： `switch(config-router-ptmp)# exit`	ポリシー テンプレート コンフィギュレーション モードを終了し、ルータ コンフィギュレーション モードに戻ります。
ステップ 7	**neighbor** { *ipv4-address* | *ipv4-prefix/length* } [**remote-as** { *as-num* } [. *as-num*]]	BGP ネイバー（ルータ、VRF）を設定し、ネイバー コンフィギュレーション モードを開始します。
ステップ 8	（任意） **address-family ipv4 unicast** 例： `switch(config-router)# address family ipv4 unicast`	アドレス ファミリ設定モードを開始します。
ステップ 9	**inherit peer-policy** *template-name sequence-number* 例： `switch(config-router-neighbor-af)# inherit peer-policy rr-client-ptl 10`	ネイバーが設定を継承できるように、ピアポリシー テンプレートをこのネイバーに送信します。
ステップ 10	（任意） **end** 例： `switch(config-router-af)# end`	特権 EXEC モードに戻ります。

追加パスのフィルタリングおよび設定操作

必要に応じて、アドバタイズされる候補である追加パスのプレフィックスを照合することで、アドバタイズされるパスをフィルタ処理するためにルートマップを使用できます（これらのプレフィックスは、**additional-paths selection** コマンドを使用して設定します）。

また、必要に応じて、ルートマップを通過したこれらのパスに対して実行するアクションを設定することもできます。このタスクでは **set metric** コマンドを使用していますが、このタスクには記載されていない他の **set** コマンドも使用できます。

手順の概要

1. **configure terminal**
2. **route-map** *map-name* [**deny** | **permit**] [*sequence-number*]
3. **set path-selection all advertise**
4. **set metric** *metric-value*
5. （任意） **end**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**route-map** *map-name* [**deny** | **permit**] [*sequence-number*] 例： `switch(config)# route-map add_path4 permit 10`	ある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再配布するルートマップと条件を定義します。
ステップ 3	**set path-selection all advertise** 例： `switch(config-route-map)# set path-selection all advertise`	受信機能がイネーブルの場合、ピアに追加パスとしてすべての BGP パスをアドバタイズします。
ステップ 4	**set metric** *metric-value* 例： `switch(config-route-map)# set metric 500`	一致基準を満たす追加パスのメトリックを設定します。 • 他の設定コマンドを使用して、ルートマップを通過したパスに対してアクションを実行することもできます。
ステップ 5	（任意） **end** 例： `switch(config-router-af)# end`	特権 EXEC モードに戻ります。

BGP 追加パスの設定の確認

BGP 追加パスの設定に関する情報を表示するには、次のコマンドを使用します。

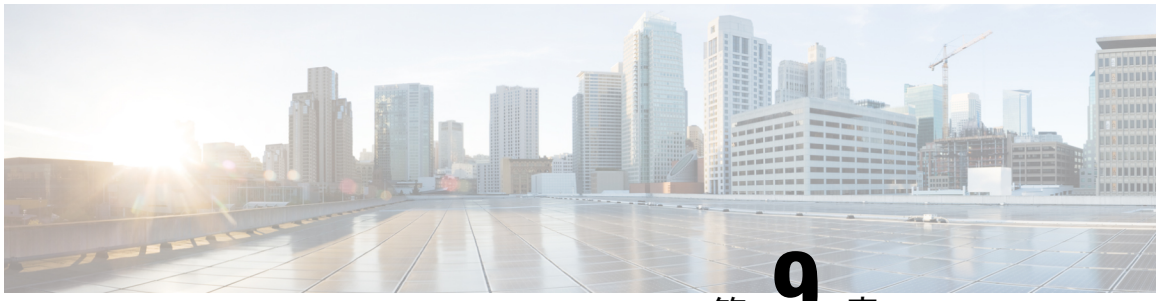
コマンド	目的
show ip bgp [*ip-address*]	BGP テーブル内のエントリを表示します。
show ip bgp neighbors [*ip-address* [**advertise-routes**]]	設定されたネイバーおよび各ネイバーに固有の他の情報を表示します。

BGP 追加パスの機能の履歴

次の表に、この機能のリリースの履歴を示します。

表 *10 : BGP* の各機能の履歴

機能名	リリース	機能情報
BGP の追加パス	6.0(2)U1(1)	BGP 追加パスは、暗黙的に以前のパスから新しいパスに代わることなく、同じプレフィックスのマルチパスをアドバタイズする機能を備えています。 次のコマンドが導入されました。 • **additional-paths receive** • **additional-paths selection** • **additional-paths send** • **capability additional-paths receive** • **capability additional-paths send** • **set path-selection all advertise** 次のコマンドが変更されました。 • **show ip bgp** • **show ip bgp neighbors**



第 9 章

RIP の設定

この章では、Cisco NX-OS スイッチでの Routing Information Protocol（RIP）の設定方法について説明します。

この章は、次の項で構成されています。

RIP 情報

このセクションは、次のトピックで構成されています。

RIP の概要

RIP はユーザデータグラムプロトコル（UDP）データパケットを使用して、小規模なインターネットワークでルーティング情報を交換します。RIPv2 は IPv4 をサポートしています。RIPv2 は RIPv2 プロトコルがサポートするオプションの認証機能を使用します（「RIPv2 認証」の項を参照）。

RIP では次の 2 種類のメッセージを使用します。

- 要求：他の RIP 対応ルータからのルート アップデートを要求するためにマルチキャスト アドレス 224.0.0.9 に送信されます。

- 応答：デフォルトでは 30 秒間隔で送信されます（「RIP の設定の確認」の項を参照）。ルータも、要求メッセージの受信後に応答メッセージを送信します。応答メッセージには、RIP ルート テーブル全体が含まれます。RIP ルーティング テーブルが 1 つの応答パケットに収まらない場合、RIP は 1 つの要求に対して複数の応答パケットを送信します。

RIP はルーティング メトリックとして、ホップ カウントを使用します。ホップ カウントは、パケットが宛先に到達するまでに、通過できるルータの数です。直接接続されたネットワークのメトリックは 1 です。到達不能な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。

RIPv2 認証

RIP メッセージに認証を設定して、ネットワークでの不正な、または無効なルーティング更新を防止できます。Cisco NX-OS は簡易パスワードまたは MD5 認証ダイジェストをサポートしています。

認証キーのキーチェーン管理を使用することによって、インターフェイスごとにRIP認証を設定できます。キーチェーン管理によって、MD5認証ダイジェストまたは単純テキストパスワード認証で使用される認証キーの変更を制御できます。キーチェーンの作成の詳細については、Cisco Nexus 3548 スイッチ NX-OS セキュリティ構成ガイドを参照してください。

MD5 認証ダイジェストを使用するには、ローカル ルータとすべてのリモート RIP ネイバーが共有するパスワードを設定します。Cisco NX-OS は、そのメッセージ自体と暗号化されたパスワードに基づいて MD5 一方向メッセージ ダイジェストを作成し、このダイジェストを RIP メッセージ（要求または応答）とともに送信します。受信側のRIPネイバーは、同じ暗号パスワードを使用して、ダイジェストを検証します。メッセージが変更されていない場合は、計算が一致し、RIP メッセージは有効と見なされます。

MD5 認証ダイジェストの場合はさらに、ネットワークでメッセージが再送されないように、各 RIP メッセージにシーケンス番号が組み込まれます。

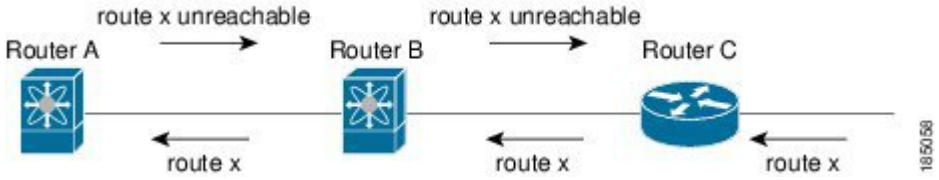
Split Horizon

スプリット ホライズンを使用すると、ルートを学習したインターフェイスから RIP がルートをアドバタイズしないようにできます。

スプリット ホライズンは、RIP アップデートおよびクエリー パケットの送信を制御する方法です。インターフェイスでスプリット ホライズンをイネーブルにすると、Cisco NX-OS は、このインターフェイスから学習された宛先への更新パケットを送信しません。この方法でアップデート パケットを制御すると、ルーティング ループの発生する可能性が小さくなります。

ポイズン リバースを指定してスプリット ホライズンを使用すると、ルートを学習したインターフェイス経由では到達不能であると RIP が学習したルートをアドバタイズするように、インターフェイスを設定できます。下の図に、ポイズン リバースをイネーブルにしてスプリット ホライズンを指定した、RIP ネットワークの例を示します。

図 *15* : スプリット ホライズン ポイズン リバースを指定した *RIP*



ルータ C はルート X について学習し、そのルートをルータ B にアドバタイズします。ルータ B はルート X をルータ A にアドバタイズしますが、ルート X の到達不能アップデートをルータ C に送り返します。

デフォルトでは、スプリット ホライズンはすべてのインターフェイスでイネーブルになっています。

ルートのフィルタリング

RIP 対応インターフェイス上でルート ポリシーを設定すると、RIP アップデートをフィルタリングできます。Cisco NX-OS は、ルート ポリシーで許可されたルートだけを使用して、ルート テーブルをアップデートします。

ルート集約

指定したインターフェイスに複数のサマリー集約アドレスを設定できます。ルート集約を使用すると、固有性の強い一連のアドレスをすべての固有アドレスを代表する 1 つのアドレスに置き換えることによって、ルート テーブルを簡素化できます。たとえば、10.1.1.0/24、10.1.2.0/24、および 10.1.3.0/24 というアドレスを 1 つの集約アドレス 10.1.0.0/16 に置き換えることができます。

RIP はルーティング テーブルに含まれている固有性の強いルートが多いほど、固有性の強いルートの最大メトリックと同じメトリックのインターフェイスからのサマリー アドレスをアドバタイズします。

（注） Cisco NX-OS は、自動ルート集約をサポートしていません。

ルートの再配布

RIP を使用すると、スタティック ルートや他のプロトコルからのルートを再配布できます。再配布を設定するには、ルート ポリシーを使用して、RIP に渡すルートを制御します。ルート ポリシーを使用すると、宛先、送信元プロトコル、ルート タイプ、ルート タグなどの属性に基づいて、ルートをフィルタリングできます。詳細については、Route Policy Manager の設定のセクションを参照してください。

RIP ルーティング ドメインにルートを再配布しても、デフォルトでは Cisco NX-OS がそのつど、RIP ルーティング ドメインにデフォルト ルートを再配布することはありません。RIP にデフォルト ルートを生成し、ルート ポリシーでそのルートを制御できます。

RIP にインポートされたすべてのルートに使用する、デフォルトのメトリックも設定できます。

ロード バランシング

ロードバランシングを使用すると、ルータは、宛先アドレスから等距離内にあるすべてのルータのネットワーク ポートにトラフィックを分散できます。ロードバランシングは、ネットワーク セグメントの使用率を向上させ、有効ネットワーク帯域幅を増加させます。

Cisco NX-OS は、等コスト マルチパス（ECMP）機能をサポートします。RIP ルート テーブルおよびユニキャスト RIB の等コスト パスは最大 32 です。これらのパスの一部または全部でトラフィックのロード バランシングが行われるように、RIP を設定できます。

仮想化のサポート

Cisco NX-OSは、同一システム上で動作する複数の RIP プロトコル インスタンスをサポートします。RIP は仮想ルーティングおよび転送（VRF）インスタンスをサポートします。

デフォルトでは、特に別の VRF を設定しない限り、Cisco NX-OS はユーザーをデフォルトの VRF に配置します。

RIP の前提条件

RIP を使用するには、次の前提条件を満たしている必要があります。

- RIP機能がイネーブルになっていることを確認します（RIP機能のイネーブル化のセクションを参照）。

RIP に関する注意事項と制約事項

RIP には、次の注意事項および制限事項があります。

- Cisco NX-OS では、RIPv1 はサポートされていません。Cisco NX-OS は、RIPv1 パケットを受信すると、メッセージを記録してパケットをドロップします。
- Cisco NX-OS は、RIPv1 ルータとの隣接関係を確立しません。

RIP のデフォルト設定

次の表に、RIP パラメータのデフォルト設定を示します。

表 *11* : デフォルトの *RIP* パラメータ

パラメータ	デフォルト
ロード バランシングを行う最大パス数	16
RIP 機能	ディセーブル
スプリット ホライズン	有効（Enabled）

RIP の設定

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

RIP 機能のイネーブル化

RIP を設定する前に、RIP 機能をイネーブルにする必要があります。

手順の概要

1. **configure terminal**
2. **feature rip**
3. （任意） **show feature**
4. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

	コマンドまたはアクション	目的
ステップ 2	**feature rip** 例： `switch(config)# feature rip`	RIP 機能を有効にします。
ステップ 3	（任意） **show feature** 例： `switch(config)# show feature`	有効および無効にされた機能を表示します。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

no feature rip コマンドを使用して、RIP 機能をディセーブルにし、関連するコンフィギュレーションをすべて削除します。

コマンド	目的
no feature rip 例： `switch(config)# no feature rip`	RIP 機能をディセーブルにして、関連するすべての設定を削除します。

RIP インスタンスの作成

RIP インスタンスを作成し、そのインスタンスのアドレス ファミリを設定できます。

始める前に

DHCP 機能が有効になっていることを確認します。RIP 機能のイネーブル化のセクションを参照してください。

手順の概要

1. **configure terminal**
2. **router rip** *instance-tag*
3. **address-family ipv4 unicast**
4. （任意） **show ip rip** [**instance** *instance-tag*] [**vrf** *vrf-name*]
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router rip** *instance-tag*
	例：
	`switch(config)# router RIP Enterprise`
	`switch(config-router)#`
		instance-tag 値 を設定して、新しい RIP インスタンスを作成します。

ステップ 3	**address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		この RIP インスタンスのアドレス ファミリを設定し、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します。

ステップ 4	（任意） **show ip rip** [**instance** *instance-tag*] [**vrf** *vrf-name*] 例： `switch(config-router-af)# show ip rip`	すべての RIP インスタンスの RIP 要約情報を表示します。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

RIPインスタンスおよび関連するすべての設定を削除する場合は、**no router rip** コマンドを使用します。

コマンド	目的
no router rip *instance-tag* 例： `switch(config)# no router rip Enterprise`	RIP インスタンスおよび関連するすべての設定を削除します。

（注） インターフェイス モードで設定した RIP コマンドを削除することも必要です。

アドレス ファミリ コンフィギュレーション モードでは、RIP に次のオプション パラメータを設定できます。

コマンド	目的
distance *value* 例： `switch(config-router-af)# distance 30`	RIP のアドミニストレーティブ ディスタンスを設定します。範囲は 1 ～ 255 です。デフォルトは 120 です。「アドミニストレーティブ ディスタンス」のセクションを参照してください。
maximum-paths *number* 例： `switch(config-router-af)# maximum-paths 6`	RIP がルート テーブルで維持する等コスト パスの最大数を設定します。指定できる範囲は 1 ～ 32 です。デフォルトは 16 です。

次に、IPv4 に対応する RIP インスタンスを作成し、ロード バランシングのための等コスト パス数を設定する例を示します。

```
switch# configure terminal
switch(config)# router rip Enterprise
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# max-paths 10
switch(config-router-af)# copy running-config startup-config
```

RIP インスタンスの再起動

RIP インスタンスの再起動が可能です。再起動すると、インスタンスのすべてのネイバーが消去されます。

RIP インスタンスを再起動し、関連付けられたすべてのネイバーを削除するには、次のコマンドを使用します。

コマンド	目的
restart rip *instance-tag* 例： `switch(config)# restart rip Enterprise`	RIP インスタンスを再起動し、すべてのネイバーを削除します。

インターフェイスでの RIP の設定

RIP インスタンスにインターフェイスを追加できます。

始める前に

RIP 機能がイネーブルになっていることを確認します（RIP 機能のイネーブル化を参照してください）。

手順の概要

1. **configure terminal**

2. **interface** *interface-type slot/port*
3. **no switchport**
4. **ip router rip** *instance-tag*
5. （任意） **show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot/port*] [**vrf** *vrf-name*] [**detail**]
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

ステップ 3	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。
ステップ 4	**ip router rip** *instance-tag* 例： `switch(config-if)# ip router rip Enterprise`	このインターフェイスを RIP インスタンスに関連付けます。
ステップ 5	（任意） **show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot/port*] [**vrf** *vrf-name*] [**detail**] 例： `switch(config-if)# show ip rip Enterprise tethernet 1/2`	インターフェイスの RIP 情報を表示します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、RIP インスタンスにインターフェイス ethernet 1/2 を追加する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip router rip Enterprise
switch(config)# copy running-config startup-config
```

RIP インスタンスの再起動

RIP インスタンスの再起動が可能です。再起動すると、インスタンスのすべてのネイバーが消去されます。

RIP インスタンスを再起動し、関連付けられたすべてのネイバーを削除するには、次のコマンドを使用します。

コマンド	目的
restart rip *instance-tag* 例： `switch(config)# restart rip Enterprise`	RIP インスタンスを再起動し、すべてのネイバーを削除します。

インターフェイスでの RIP の設定

RIP インスタンスにインターフェイスを追加できます。

始める前に

RIP 機能がイネーブルになっていることを確認します（RIP 機能のイネーブル化を参照してください）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **no switchport**
4. **ip router rip** *instance-tag*
5. （任意） **show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot/port*] [**vrf** *vrf-name*] [**detail**]
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定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	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッドインターフェイスとして設定します。
ステップ 4	**ip router rip** *instance-tag* 例： `switch(config-if)# ip router rip Enterprise`	このインターフェイスを RIP インスタンスに関連付けます。
ステップ 5	（任意） **show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot/port*] [**vrf** *vrf-name*] [**detail**] 例： `switch(config-if)# show ip rip Enterprise tethernet 1/2`	インターフェイスの RIP 情報を表示します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、RIP インスタンスにインターフェイス ethernet 1/2 を追加する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip router rip Enterprise
switch(config)# copy running-config startup-config
```

RIP 認証の設定

始める前に

RIP 機能がイネーブルになっていることを確認します（RIP 機能のイネーブル化を参照）。

認証をイネーブルにする前に、必要に応じてキーチェーンを設定します。キーチェーン実装の詳細については、Cisco Nexus 3548 スイッチ NX-OS セキュリティ構成ガイドを参照してください。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **no switchport**
4. **ip rip authentication mode** { **text** | **md5** }
5. **ip rip authentication key-chain** *key*

6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

ステップ 3	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッドインターフェイスとして設定します。
ステップ 4	**ip rip authentication mode** { **text** | **md5** } 例： `switch(config-if)# ip rip authentication mode md5`	クリアテキストまたは MD5 認証ダイジェストとして、このインターフェイスにおける RIP 認証タイプを設定します。
ステップ 5	**ip rip authentication key-chain** *key* 例： `switch(config-if)# ip rip authentication keychain RIPKey`	このインターフェイス上で RIP に使用する認証キーを設定します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、キーチェーンを作成し、RIP インターフェイス上で MD5 認証を設定する例を示します。

```
switch# configure terminal
switch(config)# key chain RIPKey
switch(config)# key-string myrip
switch(config)# accept-lifetime 00:00:00 Jan 01 2000 infinite
switch(config)# send-lifetime 00:00:00 Jan 01 2000 infinite
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip rip authentication mode md5
switch(config-if)# ip rip authentication keychain RIPKey
switch(config-if)# copy running-config startup-config
```

パッシブ インターフェイスの設定

インターフェイスを受動モードに設定することによって、ルートを受信するが、ルートアップデートの送信は行わないように RIP インターフェイスを設定できます。

受動モードでRIPインターフェイスを設定するには、インターフェイス設定モードで次のコマンドを使用します。

コマンド	目的
ip rip passive-interface **例：** `switch(config-if)# ip rip passive-interface`	インターフェイスを受動モードに設定します。

ポイズン リバースを指定したスプリット ホライズンの設定

インターフェイスの設定でポイズンリバースをイネーブルにすると、RIPが学習したルートについて、ルートを学習したインターフェイス経由では到達不能であることをアドバタイズできます。

インターフェイス上で、ポイズンリバースを指定してスプリットホライズンを設定するには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
ip rip poison-reverse **例：** `switch(config-if)# ip rip poison-reverse`	ポイズン リバースを指定してスプリット ホライズンをイネーブルにします。ポイズンリバースを指定したスプリット ホライズンは、デフォルトでディセーブルです。

ルート集約の設定

ルーティング テーブルでサマリー アドレスによって表される集約アドレスを作成できます。Cisco NX-OS は、固有性の強いすべてのルートの中でメトリックが最小のサマリー アドレス メトリックをアドバタイズします。

インターフェイス上でサマリーアドレスを設定するには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
ip rip summary-address *ip-prefix/mask-len* **例：** `switch(config-if)# ip router rip summary-address 192.0.2.0/24`	IPv4アドレスに対応する、RIP用のサマリーアドレスを設定します。

ルートの再配布の設定

別のルーティングプロトコルからのルーティング情報を受け入れて、RIP ネットワークを通じてその情報を再配布するように、RIP を設定できます。再配布されたルートを任意で、デフォルト ルートとして割り当てることができます。

始める前に

RIP 機能がイネーブルになっていることを確認します（RIP 機能のイネーブル化を参照）。

再配布を設定する前に、ルートマップを設定します。ルートマップの設定の詳細については、「ルート マップの設定 」セクションを参照してください。

手順の概要

1. **configure terminal**
2. **router rip** *instance-tag*
3. **address-family ipv4 unicast**
4. **bgp** {**direct** *as* | **eigrp** | {**ospf** | **ospfv3** | **rip** | **static** | *route-map*} **instance-tag** | } redistributemap-name
5. （任意） **default-information originate** [**always**] [**route-map** *map-name*]
6. （任意） **default-metric** *value*
7. （任意） **show ip rip route** [*ip-prefix* [**longer-prefixes** | **shorter-prefixes**] [**vrf** *vrf-name*] [*summary*]
8. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router rip** *instance-tag*
	例：
	`switch(config)# router RIP Enterprise`
	`switch(config-router)#`
		instance-tag 値 を設定して、新しい RIP インスタンスを作成します。

ステップ 3	**address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		この RIP インスタンスのアドレス ファミリを設定し、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します。

ステップ 4	**bgp** {**direct** *as*	**eigrp**	{**ospf**	**ospfv3**	**rip**	**static**	*route-map*} **instance-tag**	} redistributemap-name 例：	他のプロトコルからのルートを RIP に再配布します。ルートマップの設定の詳細については、ルートマップの設定のセクションを参照してください。

	コマンドまたはアクション	目的
	`switch(config-router-af)# redistribute eigrp 201 route-map RIPmap`	
ステップ 5	（任意） **default-information originate** [**always**] [**route-map** *map-name*] 例： `switch(config-router-af)# default-information originate always`	RIP にデフォルト ルートを生成し、必要に応じてルート マップにより制御します。
ステップ 6	（任意） **default-metric** *value* 例： `switch(config-router-af)# default-metric 10`	再配布されたすべてのルートにデフォルト メトリックを設定します。有効な範囲は 1 ～ 15 です。デフォルトは 1 です。
ステップ 7	（任意） **show ip rip route** [*ip-prefix* [**longer-prefixes** | **shorter-prefixes**] [**vrf** *vrf-name*] [*summary*] 例： `switch(config-router-af)# show ip rip route`	RIP のルートを表示します。
ステップ 8	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、EIGRP を RIP に再配布する例を示します。

```
switch# configure terminal
switch(config)# router rip Enterprise
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# redistribute eigrp 201 route-map RIPmap
switch(config-router-af)# copy running-config startup-config
```

仮想化の設定

複数の VRF を作成できます。また、各 VRF で同じ RIP インスタンスを使用することも、複数の RIP インスタンスを使用することも可能です。VRF に RIP インターフェイスを割り当てます。

（注） インターフェイスの VRF を設定した後に、インターフェイスの他のすべてのパラメータを設定します。インターフェイスの VRF を設定すると、そのインターフェイスのすべての設定が削除されます。

始める前に

RIP 機能がイネーブルになっていることを確認します（RIP 機能のイネーブル化を参照）。

手順の概要

1. **configure terminal**
2. **vrf** *vrf-name*
3. **exit**
4. **router rip** *instance-tag*
5. **vrf context** *vrf-name*
6. （任意） **address-family ipv4 unicast**
7. （任意） **bgp** {**direct** *as* | **eigrp** | {**ospf** | **ospfv3** | **rip** | **static** | *route-map*} **instance-tag** | } redistributemap-name
8. **interface** *ethernet slot/port*
9. **no switchport**
10. **vrf member** *vrf-name*
11. **ip address** *ip-prefix/length*
12. **ip router rip** *instance-tag*
13. （任意） **show ip rip route** [*ip-prefix* [**longer-prefixes** | **shorter-prefixes**] [**vrf** *vrf-name*] [*summary*]
14. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**vrf** *vrf-name*
	例：
	`switch(config)# vrf RemoteOfficeVRF`
	`switch(config-vrf)#`
		新しい VRF を作成します。

ステップ 3	**exit**
	例：
	`switch(config-vrf)# exit`
	`switch(config)#`
		VRF設定モードを終了します。

ステップ 4	**router rip** *instance-tag*
	例：
	`switch(config)# router RIP Enterprise`
	`switch(config-router)#`
		instance-tag 値 を設定して、新しい RIP インスタンスを作成します。

	コマンドまたはアクション	目的

ステップ 5	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context RemoteOfficeVRF`
	`switch(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。

ステップ 6	（任意） **address-family ipv4 unicast**
	例：
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		この RIP インスタンスの VRF アドレス ファミリを設定します。

ステップ 7	（任意） **bgp** {**direct** *as* | **eigrp** | {**ospf** | **ospfv3** | **rip** | **static** | *route-map*} **instance-tag** | } redistributemap-name 例： `switch(config-router-af)# redistribute eigrp 201 route-map RIPmap`	他のプロトコルからのルートを RIP に再配布します。ルート マップの設定の詳細については、ルートマップの設定のセクションを参照してください。

ステップ 8	**interface** *ethernet slot/port*
	例：
	`switch(config-router-vrf-af)# interface ethernet 1/2`
	`switch(config-if)#`
		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 9	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。
ステップ 10	**vrf member** *vrf-name* 例： `switch(config-if)# vrf member RemoteOfficeVRF`	このインターフェイスを VRF に追加します。
ステップ 11	**ip address** *ip-prefix/length* 例： `switch(config-if)# ip address 192.0.2.1/16`	このインターフェイスの IP アドレスを設定します。このステップは、このインターフェイスを VRF に割り当てたあとに行う必要があります。
ステップ 12	**ip router rip** *instance-tag* 例： `switch(config-if)# ip router rip Enterprise`	このインターフェイスを RIP インスタンスに関連付けます。
ステップ 13	（任意） **show ip rip route** [*ip-prefix* [**longer-prefixes** | **shorter-prefixes**] [**vrf** *vrf-name*] [*summary*] 例： `switch(config-router-af)# show ip rip route`	RIP のルートを表示します。

	コマンドまたはアクション	目的
ステップ 14	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、VRF を作成して、その VRF にインターフェイスを追加する例を示します。

```
switch# configure terminal
switch(config)# vrf context RemoteOfficeVRF
switch(config-vrf)# exit
switch(config)# router rip Enterprise
switch(config-router)# vrf RemoteOfficeVRF
switch(config-router-vrf)# address-family ipv4 unicast
switch(config-router-vrf-af)# redistribute eigrp 201 route-map RIPmap
switch(config-router-vrf-af)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# vrf member RemoteOfficeVRF
switch(config-if)# ip address 192.0.2.1/16
switch(config-if)# ip router rip Enterprise
switch(config-if)# copy running-config startup-config
```

RIP の調整

ネットワーク要件に適合するように RIP を調整できます。RIP では複数のタイマーを使用して、ルーティングアップデート間隔、ルートが無効になるまでの時間の長さ、およびその他のパラメータを決定します。これらのタイマーを調整すると、インターネットワークのニーズに適合するように、ルーティング プロトコルのパフォーマンスを調整できます。

（注） ネットワーク上のすべての RIP 対応ルータで、RIP タイマーに同じ値を設定する必要があります。

RIP を調整するには、アドレス ファミリ コンフィギュレーション モードで次のオプション コマンドを使用します。

コマンド	目的
timers basic *update timeout holddown garbage-collection* 例： `switch(config-router-af)# timers basic 40 120 120 100`	RIP タイマーを秒数で設定します。パラメータは次のとおりです。 • update：指定できる範囲は 5 ～任意の正の整数。デフォルトは 30 です。 • timeout：ルートの無効を宣言するまでに、 Cisco NX-OS が待機する時間。タイムアウト インターバルが終了するまでに、このルートのアップデート情報を Cisco NX-OS が受信しなかった場合、Cisco NX-OS はルートの無効を宣言します。指定できる範囲は 1 ～任意の正の整数です。デフォルトは 180 です。 • holddown：無効ルートに関するよりよいルート情報を Cisco NX-OS が無視する時間。指定できる範囲は 0 ～任意の正の整数です。デフォルトは 180 です。 • garbage-collection：Cisco NX-OS がルートを無効として表示してから、Cisco NX-OS がそのルートをルーティング テーブルから削除するまでの時間。指定できる範囲は 1 ～任意の正の整数です。デフォルトは 120 です。

RIP を調整するには、インターフェイス コンフィギュレーション モードで次のオプション コマンドを使用します。

コマンド	目的
ip rip metric-offset *value* 例： `switch(config-if)# ip rip metric-offset 10`	このインターフェイスで受信する各ルータのメトリックに値を追加します。有効な範囲は 1 ～ 15 です。デフォルトは 1 です。
ip rip route-filter {**prefix-list** *list-name* | **route-map** *map-name* | [**in** | **out**]} 例： `switch(config-if)# ip rip route-filter route-map InputMap in`	着信または発信 RIP アップデートをフィルタリングするための、ルート マップを指定します。

RIP の設定の確認

RIP の設定情報を表示するには、次のいずれかの作業を行います。

コマンド	目的
show ip rip instance [*instance-tag*] [**vrf** *vrf-name*]	RIP インスタンスの状態を表示します。

コマンド	目的
show ip rip [**instance** *instance-tag*] *interface slot/port* **detail** [**vrf** *vrf-name*]	インターフェイスの RIP ステータスを表示します。
show ip rip [**instance** *instance-tag*] **neighbor** [*interface-type number*] [**vrf** *vrf-name*]	RIP ネイバー テーブルを表示します。
show ip rip [**instance** *instance-tag*] **route** [*ip-prefix/lengh* [**longer-prefixes** | **shorter--prefixes**]] [*summary*] [**vrf** *vrf-name*]	RIP ルート テーブルを表示します。
show running-configuration rip	現在実行中の RIP コンフィギュレーションを表示します。

RIP 統計情報の表示

RIP 統計情報設定表示するには、次のコマンドを使用します。

コマンド	目的
show ip rip [**instance** *instance-tag*] **policy statistics redistribute** { **bgp** *as* | **direct** | { **eigrp** | **ospf** | **ospfv3** | **rip** } *instance-tag* | **static**} [**vrf** *vrf-name*]	RIP ポリシー ステータスを表示します。
show ip rip [**instance** *instance-tag*] **statistics** *interface-type number*] [**vrf** *vrf-name*]	RIP の統計情報を表示します。

clear ip rip policy コマンドを使用して、ポリシー統計情報をクリアします。

clear ip rip statistics コマンドを使用して、RIP 統計情報をクリアします。

RIP の設定例

VRF で Enterprise RIP インスタンスを作成し、その RIP インスタンスにイーサネット インターフェイス 1/2 を追加する例を示します。さらに、enthernet interface 1/2 の認証を設定し、この RIP ドメインに EIGRP を再配布します。

```
vrf context NewVRF

!

feature rip

router rip Enterprise

vrf NewVRF

address-family ip unicast
```

```
redistribute eigrp 201 route-map RIPmap
max-paths 10
!
interface ethernet 1/2
no switchport
vrf NewVRF
ip address 192.0.2.1/16
ip router rip Enterprise
ip rip authentication mode md5
ip rip authentication keychain RIPKey
```

関連項目

ルート マップの詳細については、Route Policy Manager の設定を参照してください。

その他の参考資料

RIP の実装に関連する詳細情報については、次の項を参照してください。

- 関連資料
- 標準

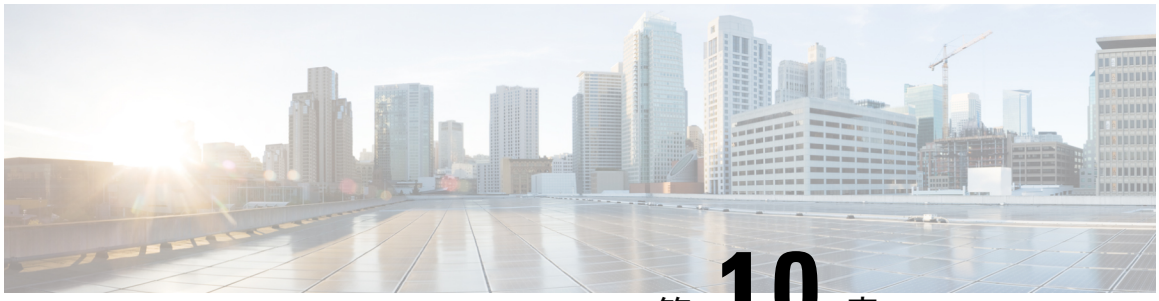
関連資料

関連項目	マニュアル タイトル
RIP CLI コマンド	『Cisco Nexus 3000 Series Command Reference』

標準

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

標準



第 10 章

スタティック ルーティングの設定

この章では、Cisco NX-OS スイッチ上でスタティック ルーティングを設定する方法について説明します。

この章は、次の項で構成されています。

スタティック ルーティングについての情報

ルータは、ユーザーが手動で設定したルート テーブル エントリのルート情報を使用するか、またはダイナミック ルーティング アルゴリズムで計算されたルート情報を使用して、パケットを転送します。

スタティック ルートは、2 つのルータ間の明示パスを定義するものであり、自動的にはアップデートされません。ネットワークに変更があった場合は、ユーザーが手動でスタティック ルートを再設定する必要があります。スタティック ルートは、ダイナミック ルートに比べて使用する帯域幅が少なくなります。ルーティング アップデートの計算や分析に CPU サイクルを使用しません。

必要に応じて、スタティック ルートでダイナミック ルートを補うことができます。スタティック ルートをダイナミック ルーティング アルゴリズムに再配布することはできますが、ダイナミック ルーティング アルゴリズムで計算されたルーティング情報をスタティック ルーティング テーブルに再配布することはできません。

スタティック ルートは、ネットワーク トラフィックが予測可能で、ネットワーク設計が単純な環境で使用します。スタティック ルートはネットワークの変化に対応できないので、大規模でたえず変化しているネットワークでは、スタティック ルートを使用すべきではありません。

大部分のネットワークは、ルータ間の通信にダイナミック ルートを使用しますが、特殊な状況でスタティック ルートを 1 つか 2 つ設定する場合があります。スタティック ルートは、最終手段としてのゲートウェイ（ルーティング不能なすべてのパケットの送信先となるデフォルト ルータ）を指定する場合にも便利です。

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、2つの異なるルーティング プロトコルから同じ宛先に、2 つ以上のルートが存在する場合に、最適なパスを選択するために、ルータが使用するメトリックです。複数のプロトコルがユニキャスト ルーティング テーブルに同じルートを追加した場合に、アドミニストレーティブ ディスタンスを手がかりに、他のルーティング プロトコル（またはスタティック ルート）ではなく、特定のルーティング プロトコル（またはスタティック ルート）が選択されます。各ルーティング プロトコルは、アドミニストレーティブ ディスタンス値を使用して、信頼性の高い順にプライオリティが与えられます。

スタティック ルートのデフォルトのアドミニストレーティブ ディスタンスは 1 です。ルータは値の小さいルートが最短であると見なすので、スタティック ルートがダイナミック ルートより優先されます。ダイナミック ルートでスタティック ルートを上書きする場合は、スタティック ルートにアドミニストレーティブ ディスタンスを指定します。たとえば、アドミニストレーティブ ディスタンスが 120 のダイナミック ルートが 2 つある場合に、ダイナミック ルートでスタティック ルートを上書きするには、スタティック ルートに 120 より大きいアドミニストレーティブ ディスタンスを指定します。

直接接続のスタティック ルート

直接接続のスタティック ルートで指定しなければならないのは、出力インターフェイス（あらゆるパケットを宛先ネットワークに送り出すインターフェイス）だけです。ルータは宛先が出力インターフェイスに直接接続されているものと見なし、パケットの宛先をネクスト ホップ アドレスとして使用します。ネクストホップは、ポイントツーポイントインターフェイスの場合に限り、インターフェイスにできます。ブロードキャストインターフェイスの場合は、ネクストホップを IPv4 アドレスにする必要があります。

フローティング スタティック ルート

フローティング スタティック ルートは、ダイナミック ルートをバックアップするためにルータが使用するスタティック ルートです。フローティング スタティック ルートには、バックアップするダイナミック ルートより大きいアドミニストレーティブ ディスタンスを設定する必要があります。この場合、ルータはフローティング スタティック ルートよりダイナミック ルートを優先させます。フローティング スタティック ルートは、ダイナミック ルートが失われた場合の代用として使用できます。

（注） デフォルトでは、ルータはダイナミック ルートよりスタティック ルートを優先させます。スタティック ルートの方がダイナミック ルートより、アドミニストレーティブ ディスタンスが小さいからです。

完全指定のスタティック ルート

完全指定のスタティック ルートでは、出力インターフェイス（あらゆるパケットを宛先ネットワークに送り出すインターフェイス）またはネクスト ホップ アドレスのどちらかを指定する必要があります。完全指定のスタティック ルートを使用できるのは、出力インターフェイスがマルチアクセス インターフェイスで、ネクストホップ アドレスを特定する必要がある場合です。ネクストホップアドレスは、指定された出力インターフェイスに直接接続する必要があります。

スタティック ルートのリモート ネクスト ホップ

リモート（非直接接続）ネクスト ホップを指定したスタティック ルートの場合、ルータに直接接続されていない隣接ルータのネクストホップアドレスを指定できます。データ転送時に、スタティック ルートにリモート ネクストホップがあると、そのネクスト ホップがユニキャスト ルーティング テーブルで繰り返し使用され、リモート ネクストホップに到達可能な、対応する直接接続のネクストホップ（複数可）が特定されます。

仮想化のサポート

スタティック ルートは仮想ルーティングおよび転送（VRF）インスタンスをサポートします。デフォルトでは、特に別の VRF を設定しない限り、Cisco NX-OS はユーザーをデフォルトの VRF に配置します。

スタティック ルーティングの前提条件

スタティック ルーティングの前提条件は、次のとおりです。

- スタティック ルートのネクストホップ アドレスは到達可能である必要があります。そうでないと、そのスタティック ルートはユニキャスト ルーティング テーブルに追加されません。

スタティック ルーティングの注意事項および制約事項

スタティック ルーティング設定時の注意事項および制約事項は、次のとおりです。

- スタティック ルートのネクストホップ アドレスとしてインターフェイスを指定できるのは、GRE トンネルなどのポイントツーポイント インターフェイスの場合に限られます。

静的ルーティングのデフォルト設定

表にスタティック ルーティング パラメータのデフォルト設定を示します。

表 *12*: デフォルトのスタティック ルーティング パラメータ

パラメータ	デフォルト
アドミニストレーティブ ディスタンス	1
RIP 機能	無効

スタティック ルーティングの設定

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

スタティック ルーティングの設定

ルータ上でスタティック ルートを設定できます。

手順の概要

1. **configure terminal**
2. **ip route** { *ip-prefix* | *ip-addr ip-mask* } {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**tag** *tag-value* [*pref*]
3. （任意） **show ip static-route**
4. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例：	コンフィギュレーション モードに入ります。

	コマンドまたはアクション	目的
	`switch# configure terminal` `switch(config)#`	
ステップ 2	**ip route** { *ip-prefix* | *ip-addr ip-mask* } {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**tag** *tag-value* [*pref*] 例： `switch(config)# ip route 192.0.2.0/8 ethernet 1/2 192.0.2.4`	スタティック ルートおよびこのスタティック ルート用のインターフェイスを設定します。任意でネクストホップアドレスを設定できます。*pref* 値で管理上の距離を設定します。範囲は 1 ～ 255 です。デフォルトは 1 です。
ステップ 3	（任意） **show ip static-route** 例： `switch(config)# show ip static-route`	スタティック ルート情報を表示します。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

スタティック ルートの設定例を示します。

```
switch# configure terminal
switch(config)# ip route 192.0.2.0/8 192.0.2.10
switch(config)# copy running-config startup-config
```

no ip static-route コマンドを使用すれば、スタティック ルートを削除できます。

仮想化の設定

VRF でスタティック ルートを設定できます。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **ip route** { *ip-prefix* | *ip-addr ip-mask* } {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**tag** *tag-value* [*pref*]
4. （任意） **show ip static-route vrf** *vrf-name*
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**vrf context** *vrf-name* 例： `switch(config)# vrf context StaticVrf`	VRF を作成し、VRF設定モードを開始します。
ステップ 3	**ip route** { *ip-prefix* | *ip-addr ip-mask* } {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**tag** *tag-value* [*pref*]] 例： `switch(config)# ip route 192.0.2.0/8 ethernet 1/2 192.0.2.4`	スタティック ルートおよびこのスタティック ルート用のインターフェイスを設定します。任意でネクストホップ アドレスを設定できます。*pref* 値で管理上の距離を設定します。範囲は 1 ～ 255 です。デフォルトは 1 です。
ステップ 4	（任意） **show ip static-route vrf** *vrf-name* 例： `switch(config)# show ip static-route`	スタティック ルート情報を表示します。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

スタティック ルートの設定例を示します。

```
switch# configure terminal
switch(config)# vrf context StaticVrf
switch(config-vrf)# ip route 192.0.2.0/8 192.0.2.10
switch(config-vrf)# copy running-config startup-config
```

スタティック ルーティングの設定確認

スタティック ルーティングの設定情報を表示するには、次のコマンドを使用します。

コマンド	目的
show ip static-route	設定されているスタティック ルートを表示します。

設定：スタティック スタティック ルーティングの例

次に、スタティック ルーティングの設定例を示します。

```
configure terminal
ip route 192.0.2.0/8 192.0.2.10
copy running-config startup-config
```

その他の参考資料

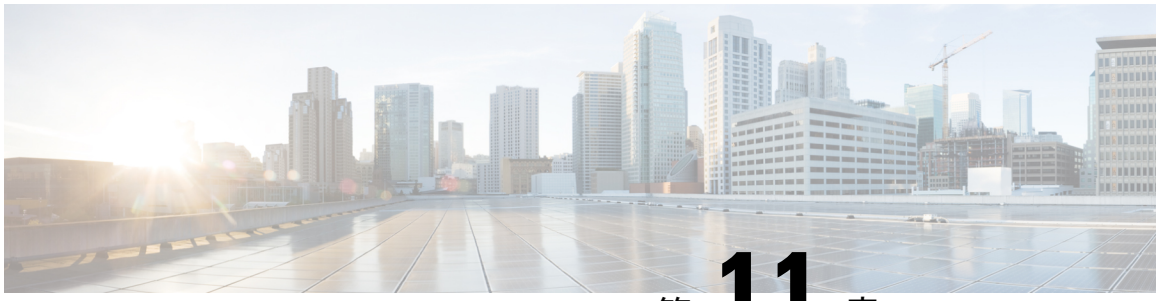
スタティック ルーティングの実装に関連する詳細情報については、次の項を参照してください。

- 関連資料

関連資料

関連項目	マニュアル タイトル
スタティック ルーティング CLI	『Cisco Nexus 3000 Series Command Reference』

関連資料



第 11 章

レイヤ 3 仮想化の設定

この章では、レイヤ 3 仮想化の設定手順について説明します。

この章は、次の項で構成されています。

レイヤ 3 仮想化

このセクションは、次のトピックで構成されています。

レイヤ 3 仮想化の概要

Cisco NX-OS は、仮想ルーティングおよび転送（VRF）インスタンスをサポートしています。各 VRF には、IPv4 に対応するユニキャスト ルート テーブルを備えた、独立したアドレス空間が 1 つずつあり、他の VRF と無関係にルーティングを決定できます。

ルータごとに、デフォルト VRF および管理 VRF があります。すべてのレイヤ 3 インターフェイスおよびルーティング プロトコルは、ユーザが別の VRF に割り当てない限り、デフォルト VRF に存在します。mgmt0 インターフェイスは、管理 VRF 内に存在します。スイッチは、VRF-Lite 機能を使用して、カスタマー エッジ（CE）スイッチで複数の VRF をサポートします。VRF-Lite によって、サービス プロバイダーは 1 つのインターフェイスを使用して、重複

する IP アドレスを持つ複数のバーチャル プライベート ネットワーク（VPN）をサポートできます。

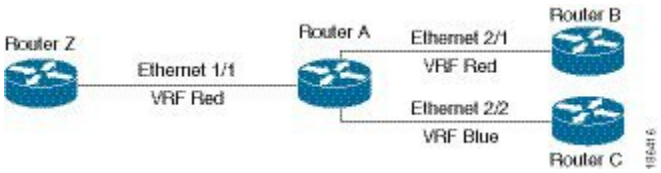
（注） スイッチでは、VPN のサポートのためにマルチプロトコル ラベル スイッチング（MPLS）が使用されません。

VRF およびルーティング

すべてのユニキャストおよびマルチキャスト ルーティング プロトコルは VRF をサポートします。VRF でルーティング プロトコルを設定する場合は、同じルーティング プロトコル インスタンスの別の VRF のルーティング パラメータに依存しないルーティング パラメータをその VRF に設定します。

VRF にインターフェイスおよびルーティング プロトコルを割り当てることによって、仮想レイヤ 3 ネットワークを作成できます。インターフェイスが存在する VRF は 1 つだけです。次の図は、1 つの物理ネットワークが 2 つの VRF からなる 2 つの仮想ネットワークに分割されている例を示しています。ルータ Z、A、および B は、VRF Red にあり、1 つのアドレス ドメインを形成しています。これらのルータは、ルータ C が含まれないルート アップデートを共有します。ルータ C は別の VRF で設定されているからです。

図 *16 :* ネットワーク内の *VRF*



デフォルトで、着信インターフェイスの VRF を使用して、ルート検索に使用するルーティング テーブルを選択します。ルート ポリシーを設定すると、この動作を変更し、Cisco NX-OS が着信パケットに使用する VRF を設定できます。

VRF は VRF 間のルート リーク（インポートまたはエクスポート）をサポートします。いくつかの制限が、VRF-Lite のルート リークに適用されます。詳細については、VRF ルート リークの注意事項と制約事項のセクションを参照してください。

VRF-Lite

VRF-Lite の機能によって、サービス プロバイダーは、VPN 間で重複した IP アドレスを使用できる複数の VPN をサポートできます。VRF-Lite は入力インターフェイスを使用して異なる VPN のルートを区別し、各 VRF に 1 つまたは複数のレイヤ 3 インターフェイスを対応付けて仮想パケット転送テーブルを形成します。VRF のインターフェイスは、イーサネット ポートなどの物理インターフェイス、または VLAN SVI などの論理インターフェイスにすることができますが、レイヤ 3 インターフェイスは、一度に複数の VRF に属することはできません。

（注） VRF-Lite の実装では、マルチプロトコル ラベル スイッチング（MPLS）および MPLS コントロール プレーンはサポートされません。

（注） VRF-Lite インターフェイスは、レイヤ 3 インターフェイスである必要があります。

VRF 認識サービス

Cisco NX-OS アーキテクチャの基本的な特徴として、すべての IP ベースの機能が VRF を認識することがあげられます。

次の VRF 認識サービスは、特定の VRF を選択することにより、リモート サーバへの接続や、選択した VRF に基づいた情報のフィルタリングを可能にします。

- AAA：詳細については、『Cisco Nexus 3548 スイッチ NX-OS セキュリティ構成ガイド』を参照してください。
- Call Home：詳細については、『Cisco Nexus 3548 スイッチ NX-OS システム管理構成ガイド』を参照してください。
- HSRP：詳細については、「HSRP の構成」の章を参照してください。
- HTTP：詳細については、『Cisco Nexus 3548 Series NX-OS 基本構成ガイド』を参照してください。
- ライセンス：詳細については、Cisco NX-OS ライセンス ガイドを参照してください。
- NTP：詳細については、『Cisco Nexus 3548 スイッチ NX-OS システム管理構成ガイド』を参照してください。
- RADIUS：詳細については、『Cisco Nexus 3548 スイッチ NX-OS セキュリティ構成ガイド』を参照してください。
- Ping と Traceroute ：詳細については、『Cisco Nexus 3548 スイッチ NX-OS 基本構成ガイド』を参照してください。
- SSH：詳細については、『Cisco Nexus 3548 スイッチ基本構成ガイド』を参照してください。
- SNMP：詳細については、『Cisco Nexus 3548 スイッチ NX-OS システム管理構成ガイド』を参照してください。
- Syslog：詳細については、『Cisco Nexus 3548 スイッチ NX-OS システム管理構成ガイド』を参照してください。
- TACACS+：詳細については、『Cisco Nexus 3548 スイッチ NX-OS セキュリティ構成ガイド』を参照してください。

- TFTP：詳細については、『Cisco Nexus 3548 スイッチ NX-OS 基本構成ガイド』を参照してください。
- VRRP：詳細については、「VRRP の構成」の章を参照してください。

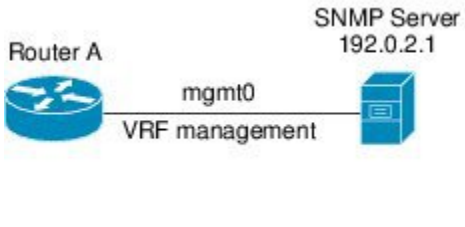
各サービスで VRF サポートを設定する方法の詳細については、各サービスに対応したCisco Nexus 3548 スイッチ構成ガイドを参照してください。

Reachability

到達可能性は、サービスを提供するサーバに到達するために必要なルーティング情報がどの VRF にあるかを示します。たとえば、管理 VRF で到達可能な SNMP サーバを設定できます。ルータにサーバ アドレスを設定する場合は、サーバに到達するために Cisco NX-OS が使用すべき VRF も設定します。

次の図は、管理 VRF を介して到達可能な SNMP サーバを示しています。SNMP サーバ ホスト 192.0.2.1 には管理 VRF を使用するように、ルータ A を設定します。

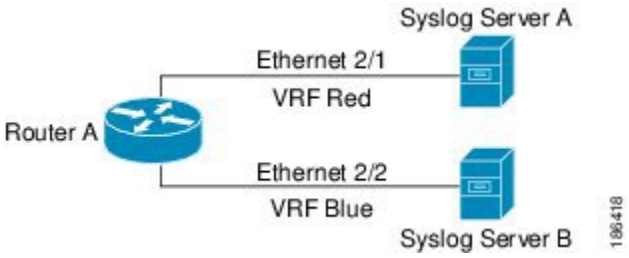
図 *17*：サービス *VRF* の到達可能性



フィルタリング

フィルタリングにより、VRF に基づいて VRF 認識サービスに渡される情報のタイプを制限できます。たとえば、Syslog サーバが特定の VRF をサポートするように設定できます。次の図は、それぞれが 1 つの VRF をサポートしている 2 つの syslog サーバーを示しています。syslog サーバ A は VRF Red で設定されているので、Cisco NX-OS は VRF Red で生成されたシステム メッセージだけを syslog サーバ A に送信します。

図 *18*：サービス *VRF* のフィルタリング



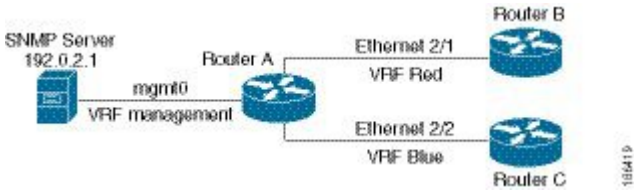
到達可能性とフィルタリングの組み合わせ

VRF 認識サービスの到達可能性とフィルタリングを組み合わせることができます。サービスに接続するために Cisco NX-OS が使用する VRF とともに、そのサービスがサポートする VRF も

設定できます。デフォルト VRF でサービスを設定する場合は、任意で、すべての VRF をサポートするようにサービスを設定できます。

次の図は、管理 VRF を介して到達可能な SNMP サーバを示しています。たとえば、SNMP サーバが VRF Red からの SNMP 通知だけをサポートするように設定できます。

図 *19*：サービス *VRF* の到達可能性とフィルタリング



VRF の注意事項と制約事項

VRF には VRF Lite のシナリオにおいて次の設定の注意事項と制約事項があります:

- インターフェイスを既存の VRF のメンバにすると、Cisco NX-OS はあらゆるレイヤ 3 設定を削除します。VRF にインターフェイスを追加したあとで、すべてのレイヤ 3 パラメータを設定する必要があります。
- 管理 VRF に mgmt0 インターフェイスを追加し、そのあとで mgmt0 の IP アドレスおよびその他のパラメータを設定します。
- VRF が存在しないうちに VRF のインターフェイスを設定した場合は、VRF を作成するまで、そのインターフェイスは運用上のダウンになります。
- Cisco NX-OS はデフォルトで、デフォルトと管理 VRF を作成します。mgmt0 は管理 VRF のメンバにする必要があります。
- **write erase boot** コマンドを実行しても、管理 VRF の設定は削除されません。まず **write erase command** コマンド、それから **write erase boot** コマンドを使用する必要があります。

VRF-Lite の注意事項と制限事項

VRF-lite には、次の注意事項と制限事項があります。

- VRF-lite を備えたスイッチは、各 VRF に対してそれぞれ、グローバル ルーティング テーブルとは異なる IP ルーティング テーブルを持ちます。
- VRF-lite が異なる VRF テーブルを使用するため、同じ IP アドレスを再利用できます。別々の VPN では IP アドレスの重複が許可されます。
- VRF-Lite では、一部の MPLS-VRF 機能（ラベル交換、LDP の隣接関係、またはラベル付きパケット）がサポートされていません。
- 複数の仮想レイヤ 3 インターフェイスを VRF-lite スイッチに接続できます。

- スイッチでは、物理ポートか VLAN SVI、またはその両方の組み合わせを使用して、VRF を設定できます。SVI は、アクセス ポートまたはトランク ポートで接続できます。
- レイヤ 3 TCAM リソースは、すべての VRF 間で共有されます。各 VRF が十分な CAM 領域を持つようにするには、maximum routes コマンドを使用します。
- すべての VRF でサポートされるルートの総数は、TCAM の容量によって制限されます。
- VRF-lite は、BGP、RIP、スタティック ルーティングをサポートします。
- VRF-lite では、EIGRP はサポートされません。
- VRF-Lite は、パケット スイッチング レートに影響しません。
- マルチキャストを同時に同一のレイヤ 3 インターフェイス上に設定することはできません。

VRF ルート リークの注意事項と制約事項

VRF ルート リークには次の注意事項と制約事項があります。

- ルート リークはデフォルト以外の 2 つの VRF 間でサポートされます。また、デフォルト VRF と任意の他の VRF 間でもサポートされます。
- デフォルト VRF へのルート リークは、グローバル VRF であるため使用できません。
- 指定した IP アドレスにマッチするルート マップのフィルタを使用して、特定のルートに対してルート リークを制限できます。
- デフォルトでは、リークできる IP プレフィックスの最大数は 1000 ルートに設定されています。この数値は 0 から 1000 までの任意の値に設定できます。
- VRF ルート リークには Enterprise ライセンスが必要で、BGP をイネーブルにする必要があります。

デフォルト設定

次の表に、VRF パラメータのデフォルト設定値を示します。

表 *13*：デフォルトの *VRF* パラメータ

パラメータ	デフォルト
設定されている VRF	デフォルト、管理
ルーティング コンテキスト	デフォルト VRF

VRF の設定

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

VRF の作成

スイッチに VRF を作成できます。

手順の概要

1. **configure terminal**
2. **vrf context** *name*
3. **ip route** { *ip-prefix* | *ip-addr ip-mask* } {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**tag** *tag-value* [*pref*]
4. （任意） **show vrf** [*vrf-name*]
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**vrf context** *name*
	例：
	`switch(config)# vrf definition Enterprise`
	`switch(config-vrf)#`
		新しい VRF を作成し、VRF 設定モードを開始します。*name* には最大 32 文字の英数字を使用できます。大文字と小文字は区別されます。

ステップ 3	**ip route** { *ip-prefix* | *ip-addr ip-mask* } {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**tag** *tag-value* [*pref*] 例： `switch(config-vrf)# ip route 192.0.2.0/8 ethernet 1/2 192.0.2.4`	スタティック ルートおよびこのスタティック ルート用のインターフェイスを設定します。任意でネクスト ホップ アドレスを設定できます。*preference* 値で管理上の距離を設定します。範囲は 1 ～ 255 です。デフォルトは 1 です。
ステップ 4	（任意） **show vrf** [*vrf-name*] 例： `switch(config-vrf)# show vrf Enterprise`	VRF 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 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

VRF および関連する設定を削除するには、**no vrf context** コマンドを使用します。

コマンド	目的
no vrf context *name* 例： `switch(config)# no vrf context Enterprise`	VRFおよび関連するすべての設定を削除します。

グローバル設定モードで使用できるコマンドはすべて、VRF 設定モードでも使用できます。

次に、VRF を作成し、VRF にスタティック ルートを追加する例を示します。

```
switch# configure terminal
switch(config)# vrf context Enterprise
switch(config-vrf)# ip route 192.0.2.0/8 ethernet 1/2
switch(config-vrf)# exit
switch(config)# copy running-config startup-config
```

インターフェイスへの VRF メンバーシップの割当て

インターフェイスを VRF のメンバにできます。

始める前に

VRF 用のインターフェイスを設定したあとで、インターフェイスに IP アドレスを割り当てます。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrf member** *vrf-name*
4. **ip address** *ip-prefix/length*
5. **show vrf** *vrf-name* **interface** *interface-type number*
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface** *interface-type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

ステップ 3	**vrf member** *vrf-name* 例： `switch(config-if)# vrf member RemoteOfficeVRF`	このインターフェイスを VRF に追加します。
ステップ 4	**ip address** *ip-prefix/length* 例： `switch(config-if)# ip address 192.0.2.1/16`	このインターフェイスの IP アドレスを設定します。このステップは、このインターフェイスを VRF に割り当てたあとに行う必要があります。
ステップ 5	**show vrf** *vrf-name* **interface** *interface-type number* 例： `switch(config-vrf)# show vrf Enterprise interface ethernet 1/2`	VRF 情報を表示します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、VRF にインターフェイスを追加する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# vrf member RemoteOfficeVRF
switch(config-if)# ip address 192.0.2.1/16
switch(config-if)# copy running-config startup-config
```

ルーティング プロトコル用の VRF パラメータの設定

1 つまたは複数の VRF にルーティング プロトコルを関連付けることができます。ルーティング プロトコルに関する VRF の設定については、該当する章を参照してください。ここでは、詳細な設定手順の例として、OSPFv2 プロトコルを使用します。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **vrf** *vrf-name*
4. （任意） **maximum-paths** *paths*
5. **interface** *interface-typeslot/port*
6. **vrf member** *vrf-name*
7. **ip address** *ip-prefix/length*
8. **ip router ospf** *instance-tag* **area** *area-id*
9. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**router ospf** *instance-tag*
	例：
	`switch(config-vrf)# router ospf 201`
	`switch(config-router)#`
		新規 OSPFv2 インスタンスを作成して、設定済みのインスタンス タグを割り当てます。

ステップ 3	**vrf** *vrf-name*
	例：
	`switch(config-router)# vrf RemoteOfficeVRF`
	`switch(config-router-vrf)#`
		VRF 設定モードを開始します。

ステップ 4	（任意） **maximum-paths** *paths* 例： `switch(config-router-vrf)# maximum-paths 4`	この VRF のルート テーブル内の宛先への、同じ OSPFv2 パスの最大数を設定します。ロード バランシングに使用されます。

ステップ 5	**interface** *interface-typeslot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。

ステップ 6	**vrf member** *vrf-name* 例： `switch(config-if)# vrf member RemoteOfficeVRF`	このインターフェイスを VRF に追加します。
ステップ 7	**ip address** *ip-prefix/length* 例： `switch(config-if)# ip address 192.0.2.1/16`	このインターフェイスの IP アドレスを設定します。このステップは、このインターフェイスを VRF に割り当てたあとに行う必要があります。

	コマンドまたはアクション	目的
ステップ 8	**ip router ospf** *instance-tag* **area** *area-id* 例： `switch(config-if)# ip router ospf 201 area 0`	このインターフェイスを OSPFv2 インスタンスおよび設定エリアに割り当てます。
ステップ 9	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、VRF を作成して、その VRF にインターフェイスを追加する例を示します。

```
switch# configure terminal
switch(config)# vrf context RemoteOfficeVRF
switch(config-vrf)# exit
switch(config)# router ospf 201
switch(config-router)# vrf RemoteOfficeVRF
switch(config-router-vrf)# maximum-paths 4
switch(config-router-vrf)# interface ethernet 1/2
switch(config-if)# vrf member RemoteOfficeVRF
switch(config-if)# ip address 192.0.2.1/16
switch(config-if)# ip router ospf 201 area 0
switch(config-if)# exit
switch(config)# copy running-config startup-config
```

VRF 認識サービスの設定

VRF 認識サービスの到達可能性とフィルタリングを設定できます。VRF 用サービスの設定手順を扱っている、該当する章またはコンフィギュレーション ガイドへのリンクについては、VRF 認識サービスのセクションを参照してください。ここでは、サービスの詳細な設定手順の例として、SNMP および IP ドメイン リストを使用します。

手順の概要

1. **configure terminal**
2. **snmp-server host** *ip-address* [**filter_vrf** *vrf-name*] [**use-vrf** *vrf-name*]
3. **vrf context** *vrf-name*
4. **ip domain-list** *domain-name* [**all-vrfs**][**use-vrf** *vrf-name*]
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例：	コンフィギュレーション モードに入ります。

	コマンドまたはアクション	目的
	`switch# configure terminal` `switch(config)#`	

ステップ 2	**snmp-server host** *ip-address* [**filter_vrf** *vrf-name*] [**use-vrf** *vrf-name*]
	例：
	`switch(config)# snmp-server host 192.0.2.1 use-vrf Red`
	`switch(config-vrf)#`
		グローバル SNMP サーバを設定し、サービスに到達するために Cisco NX-OS が使用する VRF を設定します。選択した VRF からこのサーバへの情報をフィルタリングするには、**filter-vrf** キーワードを使用します。

ステップ 3	**vrf context** *vrf-name*
	例：
	`switch(config)# vrf context Blue`
	`switch(config-vrf)#`
		新しい VRF を作成します。

ステップ 4	**ip domain-list** *domain-name* [**all-vrfs**][**use-vrf** *vrf-name*]
	例：
	`switch(config-vrf)# ip domain-list List all-vrfs use-vrf Blue`
	`switch(config-vrf)#`
		VRF でドメイン リストを設定し、必要に応じて、リスト内のドメイン名に到達するために Cisco NX-OS が使用する VRF を設定します。

ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次の例は、VRF Red 上の到達可能な SNMP ホスト 192.0.2.1 に、すべての VRF の SNMP 情報を送信する方法を示しています。

```
switch# configure terminal
switch(config)# snmp-server host 192.0.2.1 for-all-vrfs use-vrf Red
switch(config)# copy running-config startup-config
```

次に、VRF Red で到達可能な SNMP ホスト 192.0.2.12 に対して、VRF Blue の SNMP 情報をフィルタリングする例を示します。

```
switch# configure terminal
switch(config)# vrf definition Blue
switch(config-vrf)# snmp-server host 192.0.2.12 use-vrf Red
switch(config)# copy running-config startup-config
```

VRF スコープの設定

すべての EXEC コマンド（**show** コマンドなど）には、対応する VRF スコープを設定できます。VRF スコープを設定すると、EXEC コマンド出力のスコープが設定された VRF に自動的

に限定されます。このスコープは、一部の EXEC コマンドで使用できる VRF キーワードによって上書きできます。

VRF スコープを設定するには、EXEC モードで次のコマンドを使用します。

コマンド	目的

routing-context vrf *vrf-name*
例：
`switch# routing-context vrf red`
`switch%red#`
	すべての EXEC コマンドに対応するルーティング コンテキストを設定します。デフォルトのルーティング コンテキストはデフォルト VRF です。

デフォルトの VRF スコープに戻すには、EXEC モードで次のコマンドを使用します。

コマンド	目的

routing-context vrf default
例：
`switch%red# routing-context vrf default`
`switch#`
	デフォルトのルーティング コンテキストを設定します。

VRF の設定の確認

VRF の設定情報を表示するには、次のいずれかの作業を行います。

コマンド	目的
show vrf [*vrf-name*]	すべてまたは 1 つの VRF の情報を表示します。
show vrf [*vrf-name*] **detail**	すべてまたは 1 つの VRF の詳細情報を表示します。
show vrf [*vrf-name*] [**interface** *interface-type slot/port*]	インターフェイスの VRF ステータスを表示します。

VRF の設定例

次に、VRF Red を設定して、その VRF に SNMP サーバを追加し、VRF Red に OSPF インスタンスを追加する例を示します。

```
vrf context Red
    snmp-server host 192.0.2.12 use-vrf Red
    router ospf 201
    interface ethernet 1/2
    vrf member Red
```

```
    ip address 192.0.2.1/16
    ip router ospf 201 area 0
```

次に、VRF Red および Blue を設定し、各 VRF に OSPF インスタンスを追加して、各 OSPF インスタンスの SNMP コンテキストを作成する例を示します。

```
vrf context Red
vrf context Blue

feature ospf
    router ospf Lab
    vrf Red
    router ospf Production
    vrf Blue

interface ethernet 1/2
    vrf member Red
    ip address 192.0.2.1/16
    ip router ospf Lab area 0
    no shutdown

interface ethernet 10/2
    vrf member Blue
    ip address 192.0.2.1/16
    ip router ospf Production area 0
    no shutdown

snmp-server user admin network-admin auth md5 nbv-12345
    snmp-server community public ro

snmp-server context lab instance Lab vrf Red
    snmp-server context production instance Production vrf Blue
    Use the SNMP context lab to access the OSPF-MIB values for the OSPF instance Lab in
 VRF Red in the previous example.
```

次に、デフォルト以外の 2 つの VRF 間、およびデフォルト VRF からデフォルト以外の VRF にルート リークを設定する例を示します。

```
feature bgp
    vrf context Green
    ip route 33.33.33.33/32 35.35.1.254
    address-family ipv4 unicast
    route-target import 3:3
    route-target export 2:2
    export map test
    import map test
    import vrf default map test
    interface Ethernet1/7
    vrf member Green
    ip address 35.35.1.2/24
    vrf context Shared

ip route 44.44.44.44/32 45.45.1.254
    address-family ipv4 unicast
    route-target import 1:1
    route-target import 2:2
    route-target export 3:3
    export map test
    import map test
    import vrf default map test
    interface Ethernet1/11
    vrf member Shared
    ip address 45.45.1.2/24
```

```
    router bgp 100
    address-family ipv4 unicast
    redistribute static route-map test
    vrf Green
    address-family ipv4 unicast
    redistribute static route-map test
    vrf Shared
    address-family ipv4 unicast
    redistribute static route-map test

ip prefix-list test seq 5 permit 0.0.0.0/0 le 32
    route-map test permit 10
    match ip address prefix-list test
    ip route 100.100.100.100/32 55.55.55.1

nexus# show ip route vrf all
    IP Route Table for VRF "default"
    '*' denotes best ucast next-hop
    '**' denotes best mcast next-hop
    '[x/y]' denotes [preference/metric]
    '%<string>' in via output denotes VRF <string>
    55.55.55.0/24, ubest/mbest: 1/0, attached
    *via 55.55.55.5, Lo0, [0/0], 00:07:59, direct
    55.55.55.5/32, ubest/mbest: 1/0, attached
    *via 55.55.55.5, Lo0, [0/0], 00:07:59, local
    100.100.100.100/32, ubest/mbest: 1/0
    *via 55.55.55.1, [1/0], 00:07:42, static

IP Route Table for VRF "management"
    '*' denotes best ucast next-hop
    '**' denotes best mcast next-hop
    '[x/y]' denotes [preference/metric]
    '%<string>' in via output denotes VRF <string>
    0.0.0.0/0, ubest/mbest: 1/0
    *via 10.29.176.1, [1/0], 12:53:54, static
    10.29.176.0/24, ubest/mbest: 1/0, attached
    *via 10.29.176.233, mgmt0, [0/0], 13:11:57, direct
    10.29.176.233/32, ubest/mbest: 1/0, attached
    *via 10.29.176.233, mgmt0, [0/0], 13:11:57, local

IP Route Table for VRF "Green"
    '*' denotes best ucast next-hop
    '**' denotes best mcast next-hop
    '[x/y]' denotes [preference/metric]
    '%<string>' in via output denotes VRF <string>
    33.33.33.33/32, ubest/mbest: 1/0
    *via 35.35.1.254, [1/0], 00:23:44, static
    35.35.1.0/24, ubest/mbest: 1/0, attached
    *via 35.35.1.2, Eth1/7, [0/0], 00:26:46, direct
    35.35.1.2/32, ubest/mbest: 1/0, attached
    *via 35.35.1.2, Eth1/7, [0/0], 00:26:46, local
    44.44.44.44/32, ubest/mbest: 1/0
    *via 45.45.1.254%Shared, [20/0], 00:12:08, bgp-100, external, tag 100
    100.100.100.100/32, ubest/mbest: 1/0
    *via 55.55.55.1%default, [20/0], 00:07:41, bgp-100, external, tag 100

IP Route Table for VRF "Shared"
    '*' denotes best ucast next-hop
    '**' denotes best mcast next-hop
    '[x/y]' denotes [preference/metric]
    '%<string>' in via output denotes VRF <string>
    33.33.33.33/32, ubest/mbest: 1/0
    *via 35.35.1.254%Green, [20/0], 00:12:34, bgp-100, external, tag 100
    44.44.44.44/32, ubest/mbest: 1/0
```

```
    *via 45.45.1.254, [1/0], 00:23:16, static
    45.45.1.0/24, ubest/mbest: 1/0, attached
    *via 45.45.1.2, Eth1/11, [0/0], 00:25:53, direct
    45.45.1.2/32, ubest/mbest: 1/0, attached
    *via 45.45.1.2, Eth1/11, [0/0], 00:25:53, local
    100.100.100.100/32, ubest/mbest: 1/0
    *via 55.55.55.1%default, [20/0], 00:07:41, bgp-100, external, tag 100
nexus(config)#
```

関連項目

VRF の詳細については、次の項目を参照してください。

- Cisco Nexus 3548 スイッチ NX-OS 基礎構成ガイド
- Cisco Nexus 3548 スイッチ NX-OS システム管理構成ガイド

その他の参考資料

仮想化の実装に関連する詳細情報については、次の項を参照してください。

- 関連資料
- 標準

関連資料

関連項目	マニュアル タイトル
VRF CLI	『Cisco Nexus 3000 Series Command Reference』

標準

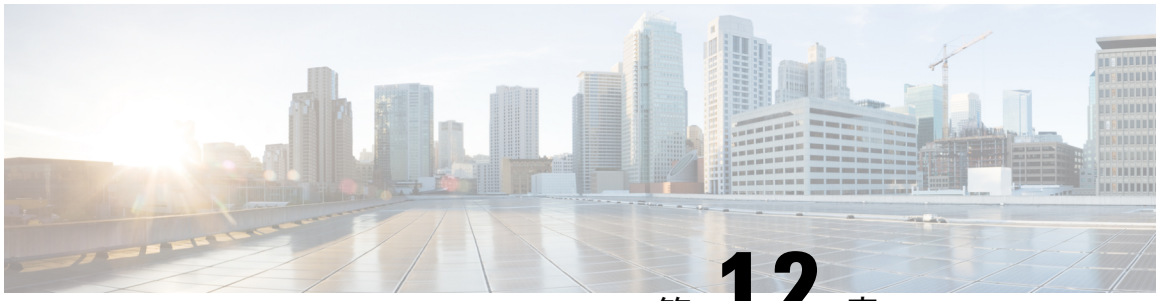
標準	タイトル
この機能でサポートされる新規の標準または変更された標準はありません。また、既存の標準のサポートは変更されていません。	—

VRF 機能の履歴

次の表に、この機能のリリースの履歴を示します。

表 *14 : VRF* 機能の履歴

機能名	リリース	機能情報
VRF	5.0(3)A1(1)	この機能が導入されました。
VRF ルート リーク	6.0(2)A1(1)	この機能が導入されました。



第 12 章

ユニキャスト RIB および FIB の設定

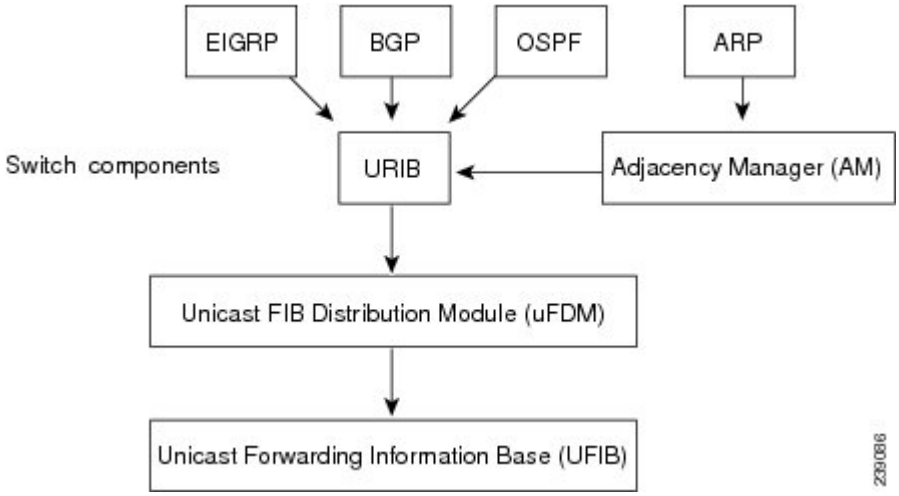
この章では、Cisco Nexus スイッチのユニキャスト ルーティング情報ベース（RIB）および転送情報ベース（FIB）のルートを設定し、管理する方法について説明します。

この章は、次の項で構成されています。

ユニキャスト RIB および FIB についての情報

ユニキャスト RIB（IPv4 RIB）および FIB は、下に示すように、 Cisco NX-OS の転送アーキテクチャの一部です。

図 *20 : CiscoNX-OS* フォワーディング アーキテクチャ



ユニキャスト RIB は、直接接続のルート、スタティック ルート、ダイナミック ユニキャスト ルーティング プロトコルで検出されたルートを含むルーティング テーブルを維持しています。

また、アドレス解決プロトコル（ARP）などの送信元から、隣接情報を収集します。ユニキャストRIBは、ルートに最適なネクストホップを決定し、さらにユニキャストFIB分散モジュール（FDM）のサービスを使用して、ユニキャスト転送情報ベース（FIB）にデータを入力します。

各ダイナミック ルーティング プロトコルは、タイム アウトしたあらゆるルートについて、ユニキャスト RIB を更新する必要があります。その後、ユニキャスト RIB はそのルートを削除し、そのルートに最適なネクスト ホップを再計算します（代わりに使用できるパスがある場合）。

レイヤ 3 整合性チェッカー

まれな事例として、各モジュールのユニキャストRIBとFIBの間に不整合が発生することがあります。Cisco NX-OS は、レイヤ 3 整合性チェッカーをサポートします。この機能は、各インターフェイス モジュールのユニキャスト IPv4 RIB と FIB の間の不整合を検出します。不整合には次のようなものがあります。

- 欠落したプレフィックス
- 余分なプレフィックス
- ネクストホップ アドレスの誤り
- ARP またはネイバー探索（ND）キャッシュ内の不正なレイヤ 2 リライト文字列

レイヤ 3 整合性チェッカーは、FIB のエントリと隣接マネージャ（AM）から取得した最新の隣接情報を比較し、不整合があれば記録します。次に整合性チェッカーは、ユニキャストRIBのプレフィックスをモジュールのFIBと比較し、不整合があればログに記録します。「レイヤ 3 整合性チェッカーのトリガー」の項を参照してください。

不整合は手動で解消できます。「FIB 内の転送情報の消去」の項を参照してください。

整合性が失われる前に整合性チェッカーを実行すれば、整合性の点では合格します。しかし、4K のハードウェア制限を超えて多くのルートが学習され、**show consistency-checker forwarding ipv4** コマンドを実行した場合も、整合性の点で合格します。整合性のない状態から整合性のある状態に移行する場合も同様です。障害ルートは引き続き表示されます。**test forwarding ipv4 inconsistency route** コマンドが再実行されるまで、この状態は終了しません。これは予期された動作です。

FIB テーブル

次に、スイッチが通常の転送モードで設定されている場合の Cisco Nexus 3548 スイッチのユニキャスト ルーティング テーブルの容量を示します。

- ユニキャスト ルーティング ホスト テーブル = 64,000 ハッシュ テーブル エントリ
- ユニキャスト ルーティング LPM テーブル = 16,000 TCAM エントリ
- ECMP メンバー テーブル サイズ：16,000 エントリ

次に、スイッチがワープ モードで設定されている場合の Cisco Nexus 3548 スイッチのユニキャスト ルーティング テーブルの容量を示します。

- L3 ユニキャスト ホスト テーブル = 8000 TCAM エントリ
- L3 ユニキャスト LPM テーブル = 4000 TCAM エントリ

（注） ECMP はワープ モードではサポートされません。

（注） ワープ モードでは、2 つの等コスト パスが RIB で受信されると、パスの 1 つがハードウェアにインストールされます。ルーティング プロトコル構成で、最大パスを 1 に構成することをお勧めします。

仮想化のサポート

ユニキャスト RIB および FIB は、仮想ルーティングおよび転送（VRF）インスタンスをサポートします。

ユニキャスト RIB および FIB の管理

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

モジュールの FIB 情報の表示

スイッチの FIB 情報を表示できます。

手順の詳細

スイッチの FIB 情報を表示するには、任意のモードで次のコマンドを使用します。

コマンド	目的
show ip fib adjacency 例： `switch# show ip fib adjacency`	FIB の隣接情報を表示します。

コマンド	目的
show forwarding ipv4 adjacency 例： `switch# show forwarding ipv4 adjacency`	IPv4 の隣接情報を表示します。
show ip fib interfaces 例： `switch# show ip fib interfaces`	IPv4 の FIB インターフェイス情報を表示します。
show ip fib route 例： `switch# show ip fib route`	IPv4 のルート テーブルを表示します。
show forwarding ipv4 route 例： `switch# show forwarding ipv4 route`	IPv4 のルート テーブルを表示します。

次に、スイッチの FIB の内容を表示する例を示します。

```
switch# show ip fib route

IPv4 routes for table default/base

------------------+------------------+----------------------
Prefix | Next-hop | Interface
------------------+------------------+----------------------
0.0.0.0/32 Drop Null0
255.255.255.255/32 Receive sup-eth1
```

ユニキャスト FIB でのロード シェアリングの設定

OSPF（Open Shortest Path First）などのダイナミック ルーティング プロトコルは、等コスト マルチパス（ECMP）によるロードシェアリングをサポートしています。ルーティング プロトコルは、そのプロトコルに設定されたメトリックに基づいて最適なルートを決定し、そのプロトコルに設定された最大数までのパスをユニキャスト RIB に組み込みます。ユニキャスト RIB は、RIB に含まれるすべてのルーティング プロトコル パスのアドミニストレーティブ ディスタンスを比較し、ルーティング プロトコルによって組み込まれたすべてのパス セットから最適なパス セットを選択します。ユニキャスト RIB は、この最適なパス セットを FIB に組み込み、フォワーディング プレーンで使用できるようにします。

フォワーディング プレーンは、ロード シェアリングのアルゴリズムを使用して、FIB に組み込まれたパスのいずれかを選択し、それを特定のデータ パケットに使用します。

ロード シェアリングの次の設定項目をグローバルに設定できます。

- ロード シェアリング モード：宛先のアドレスとポート、または送信元と宛先のアドレスとポートに基づいて、最適なパスを選択します。

- 汎用 ID：ハッシュ アルゴリズムのランダム シードを設定します。汎用 ID を設定する必要はありません。ユーザが設定しなかった場合は、Cisco NX-OS が汎用 ID を選択します。

ロードシェアリングでは、特定のフローに含まれるすべてのパケットに対して同じパスが使用されます。フローは、ユーザが設定したロードシェアリング方式によって定義されます。たとえば、送信元/宛先のロード シェアリングを設定すると、送信元 IP アドレスと宛先 IP アドレスのペアが同じであるすべてのパケットが同じパスをたどります。

ユニキャスト FIB のロード シェアリング アルゴリズムを設定するには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
ip load-sharing address { **destination port** *destination* | **source-destination** [**port source-destination**]} [**universal-id** *seed*] 例： `switch(config)# ip load-sharing address source-destination`	データ トラフィックに対するユニキャスト FIB のロードシェアリング アルゴリズムを設定します。*universal-id* の範囲は 1 ～ 4294967295 です。

ユニキャスト FIB のロード シェアリング アルゴリズムを表示するには、任意のモードで次のコマンドを使用します。

コマンド	目的
show ip load-sharing 例： `switch(config)# show ip load-sharing`	データ トラフィックに対するユニキャスト FIB のロード シェアリング アルゴリズムを表示します。

ユニキャスト RIB および FIB が特定の送信元アドレス/宛先アドレスに使用するルートを表示するには、任意のモードで次のコマンドを使用します。

コマンド	目的
show routing hash source-addr dest-addr [**source-port dest-port**] [**vrf** *vrf-name*] 例： `witch# show routing hash 192.0.2.1 10.0.0.1`	ユニキャスト RIB および FIB が特定の送信元/宛先アドレス ペアに使用するルートを表示します。送信元アドレスと宛先アドレスの形式は x.x.x.x です。送信元ポートと宛先ポートの範囲は 1 ～ 65535 です。VRF 名には最大 64 文字の英数字文字列を指定します。大文字と小文字は区別されます。

次に、特定の送信元/宛先ペアのために選択されたルートを表示する例を示します。

```
switch# show routing hash 10.0.0.5 30.0.0.2
Load-share parameters used for software forwarding:
load-share mode: address source-destination port source-destination
Universal-id seed: 0xe05e2e85
Hash for VRF "default"
Hashing to path *20.0.0.2 (hash: 0x0e), for route:
```

ルーティング情報と隣接情報の表示

ユーザーは、ルーティング情報と隣接情報を表示できます。

ルーティング情報と隣接情報を表示するには、任意のモードで次のコマンドを使用します。

コマンド	目的
show ip route [**route-type** | **interface int-type number** | **next-hop**] 例： `switch# show ip route`	ユニキャスト ルート テーブルを表示します。*route-type* 引数には、1 つのルート プレフィックス、直接、静的、またはダイナミック ルーティング プロトコルを指定します。**?** キーワードを使用して、サポートされるインターフェイスを表示します。
show ip adjacency [*prefix* | *interface number* [**summary**] | **non-best**] [**detail**] [**vrf** *vrf-id*] 例： `switch# show ip adjacency`	隣接関係テーブルを表示します。引数の範囲は次のとおりです。 • *prefix*：任意の IPv4 プレフィックス アドレス。 • *interface-type number*：**?** コマンドを使用して、サポートされるインターフェイスを表示します。 • *vrf-id*：最大 32 文字の英数字文字列。大文字と小文字は区別されます。
show ip routing [*route-type* | **interface** *int-type number* | **next-hop** | **recursive-next-hop** | **summary** | **updated** { **since** | **until** } *time*] 例： `switch# show routing summary`	ユニキャスト ルート テーブルを表示します。*route-type* 引数には、1 つのルート プレフィックス、直接、静的、またはダイナミック ルーティング プロトコルを指定します。**?** キーワードを使用して、サポートされるインターフェイスを表示します。

次に、ユニキャスト ルート テーブルを表示する例を示します。

```
switch# show ip route
IP Route Table for VRF "default"
'*' denotes best ucast next-hop
'**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]

192.168.0.2/24, ubest/mbest: 1/0, attached
*via 192.168.0.32, Eth1/5, [0/0], 22:34:09, direct
192.168.0.32/32, ubest/mbest: 1/0, attached
*via 192.168.0.32, Eth1/5, [0/0], 22:34:09, local
```

次に、隣接情報を表示する例を示します。

```
switch# show ip adjacency

IP Adjacency Table for VRF default
Total number of entries: 2
Address Age MAC Address Pref Source Interface Best
10.1.1.1 02:20:54 00e0.b06a.71eb 50 arp mgmt0 Yes
10.1.1.253 00:06:27 0014.5e0b.81d1 50 arp mgmt0 Yes
```

レイヤ 3 整合性チェッカーのトリガー

レイヤ 3 整合性チェッカーを手動でトリガーできます。

レイヤ 3 整合性チェッカーを手動でトリガーにするには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
test [**ipv4**] [**unicast**] **forwarding inconsistency** [**vrf** *vrf-name*] [**module** { *slot* | **all** }] 例： `switch(config)# test forwarding inconsistency`	レイヤ 3 整合性チェックを開始します。*vrf-name* には最大 32 文字の英数字文字列を指定します。大文字と-小文字は区別されます。*slot* の範囲は 1 ～ 10 です。

レイヤ 3 整合性チェッカーを停止するには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
test forwarding [**ipv4**] [**unicast**] **inconsistency** [**vrf** *vrf-name*] [**module** { *slot* | **all** }] **stop** 例： `switch(config)# test forwarding inconsistency stop`	レイヤ 3 整合性チェックを停止します。*vrf-name* には最大 64 文字の英数字文字列を指定します。大文字と-小文字は区別されます。*slot* の範囲は 1 ～ 10 です。

レイヤ 3 の不整合を表示するには、任意のモードで次のコマンドを使用します。

コマンド	目的
show forwarding [**ipv4**] **inconsistency** [**vrf** *vrf-name*] [**module** { *slot* | **all** }] 例： `switch(config)# show forwarding inconsistency`	レイヤ 3 整合性チェックの結果を表示します。*vrf-name* には最大 32 文字の英数字文字列を指定します。大文字と-小文字は区別されます。*slot* の範囲は 1 ～ 10 です。

FIB 内の転送情報の消去

FIB 内の 1 つまたは複数のエントリを消去できます。

（注） **clear forwarding** コマンドを実行すると、スイッチ上の転送は中断されます。

FIB 内のエントリ（レイヤ 3 の不整合を含む）を消去するには、任意のモードで次のコマンドを使用します。

コマンド	目的
clear forwarding { **ipv4** } **route** {* | *prefix* } [**vrf** *vrf-name*] [**module** { *slot* | **all** }] 例： `switch(config)# clear forwarding ipv4 route *`	FIB から 1 つまたは複数のエントリを消去します。ルートのオプションは次のとおりです。 • *：すべてのルート。 • *prefix*：任意の IP プレフィックス。 *vrf-name* には最大 32 文字の英数字文字列を指定します。大文字と-小文字は区別されます。**slot** の範囲は 1 ～ 10 です。

（注） FIB エントリをクリアした後は、RIB エントリをクリアしてください。

ルートのメモリ要件の見積もり

一連のルートおよびネクストホップ アドレスが使用するメモリを見積もることができます。

ルートのメモリ要件を見積もるには、任意のモードで次のコマンドを使用します。

コマンド	目的
show routing memory estimate routes *num-routes* **next-hops** *num-nexthops* 例： `switch# show routing memory estimate routes 1000 next-hops 1`	ルートのメモリ要件を表示します。*num-routes* の範囲は 1000 ～ 1000000 です。*num-nexthops* の範囲は 1 ～ 16 です。

ユニキャスト RIB 内のルートの消去

ユニキャスト RIB から 1 つまたは複数のルートを消去できます。

注意 * キーワードを使用すると、ルーティングが著しく妨害されます。

ユニキャスト RIB 内の 1 つまたは複数のエントリを消去するには、任意のモードで次のコマンドを使用します。

コマンド	目的
clear iproute {***** | { *route* | *prefix/length* }[*next-hop interface*]} [**vrf** *vrf-name*] 例： `switch(config)# clear ip route 10.2.2.2`	ユニキャスト RIB とすべてのモジュール FIB から 1 つまたは複数のルートを消去します。ルートのオプションは次のとおりです。 • *****：すべてのルート。 • *route*：個々の IP ルート。 • *prefix/length*：任意の IP プレフィックス。 • *next-hop*：ネクストホップ アドレス。 • *interface*：ネクストホップ アドレスに到達するためのインターフェイス *vrf-name* には最大 32 文字の英数字文字列を指定します。大文字と-小文字は区別されます。
clear routing unicast [**ip** | **ipv4**] {***** | { *route* | *prefix/length* }[*next-hop interface*]} [**vrf** *vrf-name*] 例： `switch(config)# clear routing ip 10.2.2.2`	ユニキャスト RIB から 1 つまたは複数のルートを消去します。ルートのオプションは次のとおりです。 • *****：すべてのルート。 • *route*：個々の IP ルート。 • *prefix/length*：任意の IP プレフィックス。 • *next-hop*：ネクストホップ アドレス。 • *interface*：ネクストホップ アドレスに到達するためのインターフェイス *vrf-name* には最大 32 文字の英数字文字列を指定します。大文字と-小文字は区別されます。

ユニキャスト RIB および FIB の確認

ユニキャスト RIB および FIB の設定情報を表示するには、次のいずれかの作業を行います。

コマンド	目的
show forwarding adjacency	モジュールの隣接関係テーブルを表示します。
show forwarding distribution { **clients** | **fib-state** }	FIB の分散情報を表示します。
show forwarding interfaces module *slot*	モジュールの FIB 情報を表示します。

コマンド	目的
show forwarding ipv4route	FIB 内のルートを表示します。
show ip adjacency	隣接関係テーブルを表示します。
show ip route	ユニキャスト RIB から受け取った IPv4 ルートを表示します。
show routing	ユニキャスト RIB から受け取ったルートを表示します。

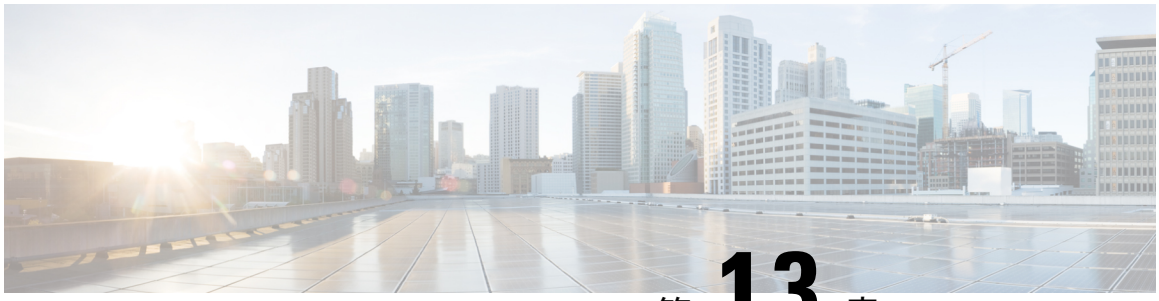
その他の参考資料

ユニキャスト RIB および FIB の管理に関連する詳細情報については、次の項を参照してください。

- 関連資料

関連資料

関連項目	マニュアル タイトル
ユニキャスト RIB および FIB の CLI コマンド	『Cisco Nexus 3000 Series Command Reference』



第 13 章

Route Policy Manager の設定

この章では、Cisco NX-OS スイッチで Route Policy Manager を設定する方法について説明します。

この章は、次の項で構成されています。

Route Policy Manager の概要

Route Policy Manager は、ルート マップおよび IP プレフィックス リストをサポートしています。この機能は、ルート再配布に使用されます。プレフィックス リストには、1 つまたは複数の IPv4 ネットワーク プレフィックスおよび関連付けられたプレフィックス長の値を指定します。プレフィックス リストは、ボーダー ゲートウェイ プロトコル（BGP）テンプレート、ルート フィルタリング、またはルーティング ドメイン間で交換されるルートの再配布などの機能で、単独で使用できます。

ルート マップは、ルートおよび IP パケットの両方に適用できます。ルート フィルタリングおよび再配布は、ルート マップを使用してルートを渡します。

プレフィックス リスト

プレフィックス リストを使用すると、アドレスまたはアドレス範囲を許可または拒否することができます。プレフィックス リストによるフィルタリングでは、ルートまたはパケットのプレフィックスと、プレフィックス リストに指定されているプレフィックスの照合が行われます。

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、実質的に拒否されたものと見なされます。

プレフィックスリストに複数のエントリを設定し、エントリと一致したプレフィックスを許可または拒否できます。各エントリにはシーケンス番号が関連付けられています。この番号はユーザが設定できます。シーケンス番号が設定されていない場合は、Cisco NX-OS によって自動的にシーケンス番号が設定されます。Cisco NX-OS はシーケンス番号が最も小さいエントリから順番にプレフィックス リストを評価します。Cisco NX-OS は、所定のプレフィックスと最初に一致したエントリを処理します。一致すると、Cisco NX-OS は permit 文または deny 文を処理し、プレフィックス リストの残りのエントリは評価しません。

（注） プレフィックス リストが空の場合は、すべてのルートが許可されます。

MAC リスト

MAC リストを使用すると、MAC アドレスまたはアドレス範囲を許可または拒否できます。MAC リストは MAC アドレスとオプションの MAC マスクのリストです。MAC マスクはワイルドカード マスクで、ルート マップが MAC リストのエントリと一致すると論理的に MAC アドレスと AND 結合されます。MAC リストによるフィルタリングでは、パケットの MAC アドレスと MAC リスト内の MAC リストが照合されます。特定の MAC アドレスが MAC リストのどのエントリとも一致しなかった場合、実質的に拒否されたものと見なされます。

MAC リストに複数のエントリを設定し、エントリと一致した MAC アドレスを許可または拒否できます。各エントリにはシーケンス番号が関連付けられています。この番号はユーザが設定できます。シーケンス番号が設定されていない場合は、Cisco NX-OS によって自動的にシーケンス番号が設定されます。Cisco NX-OS はシーケンス番号が最も小さいエントリから順番に MAC リストを評価します。Cisco NX-OS は指定された MAC アドレスと最初に一致するエントリを処理します。一致すると、Cisco NX-OS は permit 文または deny 文を処理し、MAC リストの残りのエントリは評価しません。

ルート マップ

ルート マップは、ルートの再配布に使用できます。ルート マップ エントリは、一致基準および設定基準のリストからなります。一致基準では、着信ルートまたはパケットの一致条件を指定します。設定基準では、一致基準を満たした場合のアクションを指定します。

同じルートマップに複数のエントリを設定できます。これらのエントリには、同じルートマップ名を指定し、シーケンス番号で区別します。

一意のルートマップ名の下に1つまたは複数のルートマップエントリをシーケンス番号に従って並べ、ルート マップを作成します。ルート マップ エントリのパラメータは、次のとおりです。

- シーケンス番号
- アクセス権：許可または拒否

- 一致基準
- 設定変更

ルート マップではデフォルトで、最小のシーケンス番号から順にルートまたは IP パケットが処理されます。**continue** 文を使用すると、次に処理するルート マップ エントリを決定できるので、別の順序で処理するようにルート マップを設定できます。

一致基準

さまざまな基準を使用して、ルート マップでルートや IP パケットを照合できます。BGP コミュニティ リストのように、特定のルーティング プロトコルだけに適用できる基準もありますが、IP 送信元または宛先アドレスなど、その他の基準はあらゆるルートまたは IP パケットに使用できます。

ルート マップに従ってルートまたはパケットを処理する場合、Cisco NX-OS は設定されている個々の match 文とルートまたはパケットを比較します。ルートまたはパケットが設定されている基準と一致した場合、Cisco NX-OS はルート マップ内で一致するエントリに対する許可または拒否設定、および設定されている設定基準に基づいて、このルートやパケットを処理します。

一致のカテゴリおよびパラメータは、次のとおりです。

- BGP パラメータ：AS 番号、AS パス、コミュニティ属性、または拡張コミュニティ属性に基づく一致
- プレフィックス リスト：アドレスまたはアドレス範囲に基づく一致
- マルチキャスト パラメータ：ランデブー ポイント、グループ、または送信元に基づく一致
- その他のパラメータ：IP ネクストホップ アドレスまたはパケット長に基づく一致

設定変更

ルートまたはパケットがルート マップのエントリと一致したら、設定済みの 1 つ以上の set 文に基づいて、そのルートまたはパケットを変更できます。

設定変更は次のとおりです。

- BGP パラメータ：AS パス、タグ、コミュニティ、拡張コミュニティ、ダンプニング、ローカル プリファレンス、オリジン、または重み値属性の変更
- メトリック：ルート メトリック、ルート タグ、またはルート タイプの変更
- その他のパラメータ：フォワーディング アドレスまたは IP ネクストホップ アドレスの変更

アクセス リスト

IP アクセス リストでは、次のような IP パケット フィールドとパケットを照合できます。

- 送信元または宛先 IPv4 アドレス
- プロトコル
- Precedence
- ToS

詳細については、Cisco Nexus 3548 スイッチ NX-OS セキュリティ構成ガイドを参照してください。

BGP の AS 番号

BGP ピアとの照合に使用する AS 番号のリストを設定できます。BGP ピアがリスト内の AS 番号と一致し、さらに他の BGP ピア設定と一致する場合、BGP はセッションを作成します。BGP ピアがリスト内の AS 番号と一致しない場合は、BGP はピアを無視します。AS 番号は AS 番号の範囲のリストとして設定できます。また、AS パス リストを使用して AS 番号を正規表現と比較することもできます。

BGP の AS パス リスト

AS パス リストを設定すると、着信または発信 BGP ルートのアップデートをフィルタリングできます。ルート アップデートに AS パス リストのエントリと一致する AS パス属性が含まれている場合、ルータは設定されている許可または拒否条件に基づいてルートを処理します。ルート マップの中で AS パス リストを設定できます。

同じ AS パス リスト名を使用することによって、AS パス リストで複数の AS パス エントリを設定できます。ルータは最初に一致したエントリを処理します。

BGP のコミュニティ リスト

ルート マップのコミュニティ リストを使用すると、BGP コミュニティに基づいて BGP ルート アップデートをフィルタリングできます。コミュニティ属性はコミュニティ リストに基づいて照合できます。また、コミュニティ属性はルート マップを使用して設定できます。

コミュニティ リストには、1 つまたは複数のコミュニティ属性を指定します。同じコミュニティ リスト エントリに複数のコミュニティ属性を設定した場合、BGP ルートが一致と見なされるには、指定されたすべてのコミュニティ属性と一致しなければなりません。

同じコミュニティ リスト名を使用することによって、コミュニティ リストのそれぞれ個別のエントリとして、複数のコミュニティ属性を設定することもできます。この場合、ルータは最初に BGP ルートと一致したコミュニティ属性を、そのエントリの許可または拒否設定に基づいて処理します。

コミュニティ リストのコミュニティ属性は、次の形式のいずれか 1 つで設定できます。

- 名前付きコミュニティ属性（**internet**、**no-export** など）。
- **aa:nn** 形式（前の 2 バイトは 2 バイトの自律システム番号、後の 2 バイトはユーザーが定義するネットワーク番号を表します）。
- 正規表現。

BGP の拡張コミュニティ リスト

拡張コミュニティ リストでは 4 バイトの AS 番号がサポートされています。拡張コミュニティ リストのコミュニティ属性は、次のいずれかの形式で設定できます。

- *aa4:nn* 形式（最初の 4 バイトは 4 バイトの AS 番号、最後の 2 バイトはユーザが定義するネットワーク番号を表します）。
- 正規表現。

Cisco NX-OS は汎用の特定拡張コミュニティ リストをサポートしています。このリストを使用すると、4 バイトの AS 番号に対して通常のコミュニティ リストと同様の機能を使用できます。汎用の特定拡張コミュニティ リストには次のプロパティを設定できます。

- Transitive：BGP はコミュニティ属性を自律システム間に伝達します。
- Nontransitive：BGP はコミュニティ属性を削除してからルートを他の自律システムに伝達します。

ルートの再配布およびルート マップ

ルート マップを使用すると、ルーティング ドメイン間でのルートの再配布を制御できます。ルートマップではルートの属性を照合し、一致基準を満たすルートだけを再配布します。設定変更を使用することによって、再配布時に、ルートマップでルート属性を変更することもできます。

ルータは再配布されたルートを各ルートマップエントリと照合します。match 文が複数ある場合は、ルートがすべての一致基準を満たしている必要があります。ルートがルート マップ エントリで定義されている一致基準を満たす場合は、エントリで定義されているアクションが実行されます。ルートが基準と一致しなかった場合、ルータは後続のルート マップ エントリとルートを比較します。ルートの処理は、ルートがルートマップのいずれかのエントリと一致するか、どのエントリとも一致せずすべてのエントリによる処理が完了するまで継続します。ルータがルートマップの全エントリとルートを比較しても一致しなかった場合、ルータはそのルートを受け付けるか（着信ルート マップ）またはルートを転送します（発信ルート マップ）。

Route Policy Manager の注意事項と制約事項

Route Policy Manager 設定時の注意事項および制約事項は、次のとおりです。

- CLI は **route-tag**では **set** または **match**が有効になっていますが、サポートされておらず、その特定のルートマップシーケンスに対して意図しない動作が発生します。
- プレフィックス リスト内の名前は、大文字と小文字が区別されません。一意の名前を使用することを推奨します。大文字と小文字を変更しただけの名前は使用しないでください。たとえば、CTCPrimaryNetworks と CtcPrimaryNetworks は 2 つの異なるエントリではありません。

- ルートマップが存在しない場合、すべてのルートが拒否されます。
- プレフィックス リストが存在しない場合は、すべてのルートが許可されます。
- ルート マップ エントリに match 文がない場合、ルート マップ エントリのアクセス権（許可または拒否）によって、すべてのルートまたはパケットの処理結果が決まります。
- ルート マップ エントリの match 文の中で参照されたポリシー（プレフィックス リストなど）から no-match または deny-match が戻った場合、 は match 文を Cisco NX-OS 失敗として、次のルート マップ エントリを処理します。
- ルート マップを変更しても、ルート マップ コンフィギュレーション サブモードを終了するまでは、Cisco NX-OS によりすべての変更が保留されます。その後、Cisco NX-OS がすべての変更をプロトコル クライアントに送信すると、変更が有効になります。
- 同じルートマップシーケンスに IPv4 と IPv6 の両方の match ステートメントを含めないことを推奨します。両方が必要な場合は、同じルートマップの異なるシーケンスで指定する必要があります。
- ルートマップは定義する前に使用できるので、設定変更を終えるときには、すべてのルート マップが存在していることを確認してください。
- 再配布およびフィルタリングを行う場合、ルート マップの使用状況を確認できます。各ルーティング プロトコルには、これらの統計情報を表示する機能があります。
- BGP をIGP に再配布するとき、iBGP も再配布されます。この動作を無効にするには、ルート マップに追加 deny 文を挿入します。
- Route Policy Manager は MAC リストをサポートしていません。
- ip access-list name コマンドの ACL 名の最大文字数は 64 です。ただし、RPM コマンドに関連付けられている ACL 名（ip prefix-list や match ip address など）は、最大 63 文字しか使用できません。
- BGP は特定の **match** コマンドのみをサポートします。詳細については、ルート マップの設定 （297 ページ） セクションの **match** コマンドの表を参照してください。
- 「prefix-list」という名前の ACL を作成する場合、match ip address コマンドを使用して作成されたルート マップに関連付けることはできません。RPM コマンドの match ip address prefix-list は、前のコマンド（「prefix-list」ACL 名）をあいまいにします。
- match ip address コマンドを使用する場合、設定できる ACL は 1 つだけです。

Route Policy Manager のデフォルト設定

下の表に、Route Policy Manager のデフォルト設定を示します。

表 *15* : デフォルトの *Route Policy Manager* パラメータ

パラメータ	デフォルト
Route Policy Manager	有効（Enabled）

Route Policy Manager の設定

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

IP プレフィックス リストの設定

IP プレフィックス リストでは、プレフィックスおよびプレフィックス長のリストに対して IP パケットまたはルートを照合します。IPv4 の IP プレフィックス リストを作成できます。

指定したプレフィックス長と完全に一致するプレフィックス リスト エントリのみを対象とするよう設定できます。また、指定したプレフィックス長の範囲に該当するすべてのプレフィックスを対象とすることもできます。

ge キーワードと **lt** キーワードを使用すると、プレフィックス長の範囲を指定できます。着信パケットまたはルートがプレフィックス リストと一致すると判定されるのは、プレフィックスが一致し、プレフィックス長が **ge** キーワードの値（設定されている場合）以上かつ **lt** キーワードの値（設定されている場合）以下の場合です。キーワード **eq** を使用する場合、設定する値はプレフィックスのマスク長より大きくする必要があります。

手順の概要

1. **configure terminal**
2. （任意） **ip prefix-list** *name* **description** *string*
3. **ip prefix-list** *name* [**seq** *number*] [{ **permit** | **deny** } *prefix* {[**eq** *prefix-length*] | [**ge** *prefix-length*] [**le** *prefix-length*]}]
4. （任意） **show ip prefix-list** *name*
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例：	コンフィギュレーション モードに入ります。

	コマンドまたはアクション	目的
	`switch# configure terminal` `switch(config)#`	
ステップ 2	（任意） **ip prefix-list** *name* **description** *string* 例： `switch(config)# ip prefix-list AllowPrefix description allows engineering server`	プレフィックス リストについての情報ストリングを追加します。
ステップ 3	**ip prefix-list** *name* [**seq** *number*] [{ **permit** | **deny** } *prefix* {[**eq** *prefix-length*] | [**ge** *prefix-length*] [**le** *prefix-length*]}] 例： `switch(config)# ip prefix-list AllowPrefix seq 10 permit 192.0.2.0 eq 24`	IPv4 プレフィックス リストを作成するか、または既存のプレフィックス リストにプレフィックスを追加します。プレフィックス長の照合は次のように行われます。 • eq：*prefix length* の値と完全に一致するものが対象。この値は、マスク長より大きくする必要があります。 • ge：設定された *prefix length* 以上のプレフィックス長が対象。 • le：設定された *prefix length* 以下のプレフィックス長が対象。
ステップ 4	（任意） **show ip prefix-list** *name* 例： `switch(config)# show ip prefix-list AllowPrefix`	プレフィックス リストについての情報を表示します。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch# copy running-config startup-config`	この設定変更を保存します。

例

次に、2 つのエントリからなる IPv4 プレフィックス リストを作成し、BGP ネイバーにプレフィックス リストを適用する例を示します。

```
switch# configure terminal
switch(config)# ip prefix-list allowprefix seq 10 permit 192.0.2.0/23 eq 24
switch(config)# ip prefix-list allowprefix seq 20 permit 209.165.201.0/27 eq 28
switch(config)# router bgp 65536:20
switch(config-router)# neighbor 192.0.2.1/16 remote-as 65535:20
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# prefix-list allowprefix in
```

MAC リストの設定

MAC リストを設定すると、特定の範囲の MAC アドレスを許可または拒否できます。

手順の概要

1. **configure terminal**
2. **mac-list** *name* [**seq** *number*] { **permit** | **deny** } *mac-address* {*mac-mask*]}
3. （任意） **show mac-list** *name*
4. （任意） **show ip prefix-list** *name*

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 terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**mac-list** *name* [**seq** *number*] { **permit** | **deny** } *mac-address* {*mac-mask*]} 例： `switch(config)# mac-list AllowMac seq 1 permit 0022.5579.a4c1 ffff.ffff.0000`	MAC リストを作成するか、既存の MAC リストに MAC アドレスを追加します。*seq* の範囲は 1 ～ 4294967294 です。*mac-mask* は照合する MAC アドレスの部分を表します。MAC アドレス形式である必要があります。
ステップ 3	（任意） **show mac-list** *name* 例： `switch(config)# show mac-list AllowMac`	MAC リストに関する情報を表示します。
ステップ 4	（任意） **show ip prefix-list** *name* 例： `switch(config)# show ip prefix-list AllowPrefix`	プレフィックス リストについての情報を表示します。

AS パス リストの設定

発信と着信の両方の BGP ルートに AS パス リスト フィルタを指定できます。各フィルタは、正規表現を使用するアクセス リストです。正規表現が ASCII ストリングとして表されたルートの AS パス属性と一致した場合は、許可または拒否条件が適用されます。

手順の概要

1. **configure terminal**
2. **ip as-path access-list** *name* { **deny** | **permit** } *expression*
3. （任意） **show ip as-path-access-list** *name*
4. （任意） **show ip prefix-list** *name*

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 terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**ip as-path access-list** *name* { **deny** | **permit** } *expression* 例： `switch(config)# ip as-path access-list Allow40 permit 40`	
ステップ 3	（任意） **show ip as-path-access-list** *name* 例： `switch(config)# show ip as-path-access-list Allow40`	as-path アクセス リストの情報を表示します。
ステップ 4	（任意） **show ip prefix-list** *name* 例： `switch(config)# show ip prefix-list AllowPrefix`	プレフィックス リストについての情報を表示します。

例

次に、2 つのエントリからなる AS パス リストを作成し、BGP ネイバーに AS パス リストを適用する例を示します。

```
switch# configure terminal
switch(config)# ip as-path access-list AllowAS permit 64510
switch(config)# ip as-path access-list AllowAS permit 64496
switch(config)# copy running-config startup-config
switch(config)# router bgp 65536:20
switch(config-router)# neighbor 192.0.2.1/16 remote-as 65535:20
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# filter-list AllowAS in
```

BGP AS-path 属性の置き換え

次の手順では、着信および発信ルート マップの BGP as-path 属性を変更することにより、BGP ルーティング ポリシーを操作できます。

BGP as-path 属性を置き換えるときは、次のガイドラインを考慮してください。

- この機能は、アドレス ファミリ識別子 (AFI) ごとに eBGP ネイバーにのみ適用されます。iBGP ネイバーで機能を設定しようとしても、構成は無視されます。
- この機能を備えたルート マップは、BGP ネイバーのインバウンド側とアウトバウンド側の両方に適用できます。

- この機能は、AS_SET、AS_SEQUENCE、CONFED_SET、およびCONFED_SEQUENCE の任意の組み合わせをサポートします。
- 2 バイト AS のみをサポートする BGP スピーカーと対話する場合、4 バイト AS 番号は予約済みの 2 バイト AS 番号 23456 に置き換えられます。
- コンフェデレーション識別子が設定されている場合は、コンフェデレーションの外部にあるピアと対話するときに、CLI でローカル ASN としてコンフェデレーション識別子を使用することを検討してください。同じコンフェデレーションに属するピアと対話する場合は、**router bgp** *asn* コマンドでプロセス ASN を使用することを検討してください。
- BGP **local-as** 機能が設定されている場合、設定された local-as は CLI でローカル ASN と見なされます。
- アウトバウンド ルート マップの場合、ローカル ASN は常に CLI からの結果の as_path に付加されます。
- **set as-path** または **set as-path replace** コマンドでは、最大 32 個の AS 番号を設定できます。
- 1 つのルート マップ シーケンスの下では、**set as-path**、**set as-path prepend**、および **set as-path replace** のオプションのうち 1 つだけを設定できます。
- **remove-private-as** が設定されている場合、アウトバウンド側で新しいルート マップ コマンドを適用する前に適用されます。
- **as-override** が設定されている場合、アウトバウンド側で新しいルート マップ コマンドを適用した後に適用されます。
- AS_PATH ループ チェックは、新しいルート マップ コマンドが着信側と発信側の両方に適用される前に、元の AS_PATH で実行されます。これらのチェックは、インバウンド側で **allow-as in** とアウトバウンド側で**disable-peer-as-check** を使用することで緩和できます。

完全な AS パスの置き換え

この手順を使用して、着信または発信 BGP アップデートの AS パスをカスタム AS パスに変更します。AS パスを完全に削除することもできます。

手順

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

ステップ 2	**route-map** *map-name* [**permit** | **deny**] [*seq*]
	例：
	`switch(config)# route-map Testmap permit 10`
	`switch(config-route-map)#`
		ルート マップを作成するか、または既存のルート マップに対応するルートマップ設定モードを開始します。ルートマップのエントリを順序付けるには、*seq* を使用します。

	コマンドまたはアクション	目的
ステップ 3	[**no**] **set as-path** { **none** | {*as-number* | **remote-as** | **local-as**}+] } 例： `switch(config-route-map)# set as-path 11 local-as remote-as 13`	AS_PATH をカスタム ASN のリストに置き換えるか、AS_PATH をクリアします。コマンド オプションは次のとおりです。 • *as-number*: 指定された AS 番号。 • **remote-as**: BGP ピアの AS 番号。 • **local-as**: ローカル AS 番号。 **none** キーワードは、AS パスを完全に削除します。

例

次の例では、これらの値が想定されています。

- 元の AS_PATH は **10 20 30 40 50 60** です。
- local-as は **100** です。
- remote-as は **200** です。

この例は、カスタム AS パスを指定する方法を示しています。このコマンドは、AS パスを **11 100 200 13 200 10.10 65535** に変更します。

```
switch# configure terminal
switch(config)# route-map Testmap permit 10
switch(config-route-map)# set as-path 11 local-as remote-as 13 remote-as 10.10 65535
```

この例は、AS パスをクリアする方法を示しています。このコマンドにより、AS パスが空になります。

```
switch# configure terminal
switch(config)# route-map Testmap permit 10
switch(config-route-map)# set as-path none
```

AS パスでの選択した AS 番号の置き換え

この手順を使用して、AS パス内の特定の AS 番号を置き換え、着信または発信 BGP 更新でそれらをカスタム AS 番号に置き換えます。**private-as** をマッチ キーワードとして指定することもできます。この場合、private-as の任意のインスタンスが一致し、置換または削除できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例：	グローバル設定モードを開始します。

	コマンドまたはアクション	目的
	`switch# configure terminal` `switch(config)#`	

ステップ 2	**route-map** *map-name* [**permit** | **deny**] [*seq*]
	例：
	`switch(config)# route-map Testmap permit 10`
	`switch(config-route-map)#`
		ルート マップを作成するか、または既存のルート マップに対応するルートマップ設定モードを開始します。ルートマップのエントリを順序付けるには、*seq* を使用します。

ステップ 3	[**no**] **set as-path replace** {*asn_list* | **private-as**} [**with** {*as-number* | **remote-as** | **none**}] 例： `switch(config-route-map)# set as-path replace 1, 2, private-as with remote-as`	**with** キーワードが指定されていない場合は、コンマで区切られた *asn_list* で示されている ASN のインスタンスを local-as に置き換えます。**private-as** キーワードが指定されている場合は、private-as を置き換えます。 **with** キーワードが指定されている場合は、一致した ASN の **with** キーワードの後の値、または **private-as** キーワードが指定されている場合は private-as を置き換えます。 **with** キーワードに続くコマンドオプションは次のとおりです。 • *as-number* : 一致した値は、指定された AS 番号に置き換えられます。 • **remote-as**: 一致した値は、BGP ピアの AS 番号に置き換えられます。 • **none**: 一致した値は AS-path から削除されます。

例

次の例では、これらの値が想定されます。

- 元の AS_PATH は **1 5 2 10.10 65534 20** です。
- local-as は **100** です。
- remote-as は **200** です。

この例は、2 つの特定の ASN と、private-as を local-as に置き換える方法を示しています。このコマンドは、AS パスを **100 5 100 10.10 100 20** に変更します。

```
switch# configure terminal
switch(config)# route-map Testmap permit 10
switch(config-route-map)# set as-path replace 1, 2, private-as
```

この例は、2 つの特定の ASN と、private-as をネイバーの ASN (remote-as) に置き換える方法を示しています。このコマンドは、AS パスを **200 5 200 10.10 200 20** に変更します。

```
switch# configure terminal
switch(config)# route-map Testmap permit 10
switch(config-route-map)# set as-path replace 1, 2, private-as with remote-as
```

この例は、2 つの特定の ASN と private-as を削除する方法を示しています。このコマンドは、AS パスを **5 10.10 20** に変更します。

```
switch# configure terminal
switch(config)# route-map Testmap permit 10
switch(config-route-map)# set as-path replace 1, 2, private-as with none
```

コミュニティ リストの設定

コミュニティ リストを使用すると、コミュニティ属性に基づいて BGP ルートをフィルタリングできます。コミュニティ番号は *aa:nn* 形式の 4 バイト値です。最初の 2 バイトは自律システム番号を表し、最後の 2 バイトはユーザ定義のネットワーク番号です。

同じコミュニティ リスト文で複数の値を設定した場合、コミュニティ リスト フィルタを満足させるには、すべてのコミュニティ値が一致しなければなりません。複数の値をそれぞれ個別のコミュニティ リスト文で設定した場合は、最初に条件が一致したリストが処理されます。

コミュニティ リストを match 文で使用すると、コミュニティ属性に基づいて BGP ルートをフィルタリングできます。

手順の概要

1. **configure terminal**
2.
3. （任意） show ip community-list *name*
4. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2		

オプション	説明
コマンド	目的

	コマンドまたはアクション		目的
	オプション	説明	
	ip community-list standard *list-name* { **deny** | **permit** } [*community-list*] [**internet**] [**local-AS**] [**no-advertise**] [**no-export**] 例： `switch(config)# ip community-list standard BGPCommunity permit no-advertise 65536:20`	標準 BGP コミュニティ リストを作成します。*list-*名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。*community-list* には、1 つ以上のコミュニティを *aa:nn* 形式で指定できます。	
	ip community-list expanded *list-name* { **deny** | **permit** } *expression* 例： `switch(config)# ip community-list expanded BGPComplex deny 50000:[0-9][0-9]_`	正規表現を使用して拡張 BGP AS コミュニティ リストを作成します。	
ステップ 3	（任意） show ip community-list *name* **例**： `switch(config)# show ip community-list BGPCommunity`		コミュニティ リストの情報を表示します。
ステップ 4	（任意） **copy running-config startup-config** **例**： `switch# copy running-config startup-config`		この設定変更を保存します。

例

次に、2 つのエントリからなるコミュニティ リストの作成例を示します。

```
switch# configure terminal
switch(config)# ip community-list standard BGPCommunity permit no-advertise 65536:20
switch(config)# ip community-list standard BGPCommunity permit local-AS no-export
switch(config)# copy running-config startup-confi
```

拡張コミュニティ リストの設定

拡張コミュニティ リストを使用すると、コミュニティ属性に基づいて BGP ルートをフィルタリングできます。コミュニティ番号は *aa4:nn* 形式の 6 バイト値です。最初の 4 バイトは自律システム番号を表し、最後の 2 バイトはユーザ定義のネットワーク番号です。

同じ拡張コミュニティ リスト文で複数の値を設定した場合、拡張コミュニティ リスト フィルタの条件を満たすには、すべての拡張コミュニティ値が一致しなければなりません。複数の値をそれぞれ個別の拡張コミュニティ リスト文で設定した場合は、最初に条件が一致したリストが処理されます。

拡張コミュニティ リストを match 文で使用すると、拡張コミュニティ属性に基づいて BGP ルートをフィルタリングできます。

手順の概要

1. **configure terminal**
2.
3. （任意） show ip community-list *name*
4. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2		

オプション	説明
コマンド	目的
ip community-list standard *list-name* { **deny** | **permit** } **4bytegeneric** { **transitive** | **non-transitive** } **community1** [**community2**] 例： `switch(config)# ip extcommunity-list standard BGPExtCommunity permit 4bytegeneric transitive 65536:20`	標準 BGP コミュニティ リストを作成します。*community-list* には、1 つ以上の拡張コミュニティを *aa:nn* 形式で指定できます。
ip extcommunity-list expanded *list-name* { **deny** | **permit** } *expression* 例： `switch(config)# ip community-list expanded BGPComplex deny 50000:[0-9][0-9]_`	正規表現を使用して拡張 BGP AS コミュニティ リストを作成します。

	コマンドまたはアクション	目的
ステップ 3	（任意） show ip community-list *name* 例： `switch(config)# show ip community-list BGPCommunity`	コミュニティ リストの情報を表示します。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch# copy running-config startup-config`	この設定変更を保存します。

例

次に、汎用の特定拡張コミュニティ リストを作成する例を示します。

```
switch# configure terminal
switch(config)# ip extcommunity-list standard test1 permit 4bytegeneric transitive
65536:40 65536:60
switch(config)# copy running-config startup-config
```

ルート マップの設定

始める前に

ルート マップを使用して、ルートの再配布やルート フィルタリングを行うことができます。ルート マップには、複数の一致基準と複数の設定基準を含めることができます。

BGP にルート マップを設定すると、BGP ネイバー セッションの自動ソフト クリアまたはリフレッシュのトリガーになります。

手順の概要

1. **configure terminal**
2. configure terminal
3. （任意） **continue** *seq*
4. （任意） **exit**
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

	コマンドまたはアクション	目的

ステップ 2	configure terminal
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 3	（任意） **continue** *seq* 例： `switch(config-route-map)# continue 10`	ルートマップで次を処理するシーケンス文を決定します。使用するのは、フィルタリングおよび再配布の場合だけです。
ステップ 4	（任意） **exit** 例： `switch(config-route-map)# continue 10`	ルートマップで次を処理するシーケンス文を決定します。使用するのは、フィルタリングおよび再配布の場合だけです。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch# copy running-config startup-config`	この設定変更を保存します。

例

ルートマップ コンフィギュレーション モードで、ルートマップに対して次のオプションの match パラメータを設定できます。

（注） **default-information originate** コマンドでは、オプションのルート マップの **match** 文は無視されます。

コマンド	目的
match as-path *name* [*name...*] 例： `switch(config-route-map)# match as-path Allow40`	1 つまたは複数の AS パス リストと照合。AS パス リストは、**ip as-path access-list** コマンドで作成します。
match as-number { *number* [,*number...*] | **as-path-list** *name* [*name...*]} 例： `switch(config-route-map)# match as-number 33,50-60`	1 つまたは複数の AS 番号または AS パス リストと照合。AS パス リストは、**ip as-path access-list** コマンドで作成します。指定できる範囲は 1 ～ 65535 です。AS パス リスト名には最大 63 文字の英数字を使用できます。大文字と小文字は区別されます。

コマンド	目的
match community name [*name...*] [**exact-match**] 例： `switch(config-route-map)# match community BGPCommunity`	1 つまたは複数のコミュニティ リストと照合。コミュニティ リストは、**ip community-list** コマンドで作成します。
match extcommunity *name* [*name...*] [**exact-match**] 例： `switch(config-route-map)# match extcommunity BGPextCommunity`	1 つまたは複数の拡張コミュニティ リストと照合。コミュニティリストは、**ip extcommunity-list** コマンドで作成します。
match interface *interface-type number* [*interface-type number...*] 例： `switch(config-route-map)# match interface e 1/2`	設定済みのインターフェイスのいずれかからのネクスト ホップと照合。**?** を使用すると、サポートされているインターフェイス タイプのリストを検索できます。
match ip address prefix-list *name* [*name...*] 例： `switch(config-route-map)# match ip address prefix-list AllowPrefix`	1 つまたは複数の IPv4 プレフィックス リストと照合。プレフィックス リストは **ip prefix-list** コマンドを使用して作成します。
match ip next-hop prefix-list *name* [*name ...*] 例： `switch(config-route-map)# match ip next-hop prefix-list AllowPrefix`	1 つまたは複数の IP プレフィックス リストに対して、ルートの IPv4 ネクストホップ アドレスを照合。プレフィックス リストは *ip prefix-list* コマンドを使用して作成します。
match ip route-source prefix-list *name* [*name ...*] 例： `switch(config-route-map)# match ip route-source prefix-list AllowPrefix`	1 つまたは複数の IP プレフィックス リストに対して、ルートの IPv4 ルート送信元アドレスを照合。プレフィックス リストは **ip prefix-list** コマンドを使用して作成します。
match mac-list *name* [*name...*] 例： `switch(config-route-map)# match mac-list AllowMAC`	1 つまたは複数の MAC リストと照合。MAC リストは **mac-list** コマンドを使用して作成します。

コマンド	目的
match metric *value* [*+-deviation*] [*value..*] 例： `switch(config-route-map)# match mac-list AllowMAC`	ルート メトリック値を 1 つまたは複数のメトリック値または値の範囲と照合。メトリック範囲は **+-deviation** 引数を使用して設定します。ルート マップは次の範囲に該当するすべてのルート メトリックと照合されます。 *value - deviation* to *value + deviation*.
match route-type *route-type* 例： `switch(config-route-map)# match route-type level 1 level 2`	ルート タイプと照合。*route-type* は、次のうちの 1 つまたは複数にできます。 • external • internal • level-1 • level-2 • ローカル • nssa-external • type-1 • type-2
match tag *tagid* [*tagid.. .]* 例： `switch(config-route-map)# match tag 2`	フィルタリングまたは再配布に関する 1 つまたは複数のタグとルートを照合。
match vlan *vlan-id* [*vlan-range*] 例： `switch(config-route-map)# match vlan 3, 5-10`	VLAN と照合。

ルート マップ設定モードで、オプションとして、ルート マップに次の set パラメータを設定できます。

コマンド	目的
set as-path { **tag** | **prepend** { **last-as** *number* | *as-1* [*as-2.. .]*}} 例： `switch(config-route-map)# set as-path prepend 10 100 110`	BGP ルートの AS パス属性を変更します。最後の AS 番号として設定された *number* または特定の AS パス値としてのストリング（*as-1 as-2...as-n*）を前に付加できます。

コマンド	目的
set comm-list *name* **delete** 例： `switch(config-route-map)# set comm-list BGPCommunity delete`	着信または発信 BGP ルート アップデートのコミュニティ属性から、コミュニティを削除します。コミュニティ リストは **ip community-list** コマンドを使用して作成します。
set community { **none** | **additive** | **local-AS** | **no-advertise** | **no-export** | *community-1* [*community-2. ..*]} 例： `switch(config-route-map)# set community local-AS`	BGP ルート アップデートのコミュニティ属性を設定します。 （注） ルート マップ属性の同じシーケンスで、**set community** コマンドと **set comm-list delete** コマンドを両方使用すると、設定処理より先に削除処理が実行されます。 （注） **send-community** コマンドを BGP ネイバー アドレス ファミリ コンフィギュレーション モードで使用して、BGP コミュニティ属性を BGP ピアにプロパゲートします。
set dampening *halflife reuse suppress duration* 例： `switch(config-route-map)# set dampening 30 1500 10000 120`	BGP ルート ダンプニング パラメータを設定します。 • *halflife*：指定できる範囲は 1 ～ 45 分です。デフォルトは 15 です。 • *reuse*：指定できる範囲は 1 ～ 20000 秒です。デフォルトは 750 です。 • *suppress*：指定できる範囲は 1 ～ 20000 です。デフォルトは 2000 です。 • *duration*：指定できる範囲は 1 ～ 255 分です。デフォルトは 60 です。
set extcomm-list *name* **delete** 例： `switch(config-route-map)# set extcomm-list BGPextCommunity delete`	着信または発信 BGP ルート アップデートの拡張コミュニティ属性から、コミュニティを削除します。拡張コミュニティ リストは **ip extcommunity-list** コマンドを使用して作成します。

コマンド	目的
set extcommunity generic { **transitive** | **nontransitive** }{ **none** | **additive**] *community-1* [*community-2...*]} 例： `switch(config-route-map)# set extcommunity generic transitive 1.0:30`	BGP ルート アップデートの拡張コミュニティ属性を設定します。 （注） ルート マップ属性の同じシーケンスで、**set extcommunity** コマンドと **set extcomm-list delete** コマンドを両方使用すると、設定処理より先に削除処理が実行されます。 （注） **send-community** コマンドを BGP ネイバー アドレス ファミリ コンフィギュレーション モードで使用して、BGP コミュニティ属性を BGP ピアにプロパゲートします。
set forwarding-address 例： `switch(config-route-map)# set forwarding-address`	OSPF のフォワーディング アドレスを設定します。
set level { **backbone** | **level-1** | **level-1-2** | **level-2** } 例： `switch(config-route-map)# set level backbone`	IS-IS 用にルートをインポートするエリアを設定します。IS-IS のオプションは level-1、level-1-2、または level-2 です。デフォルトは level-1 です。
set local-preference *value* 例： `switch(config-route-map)# set local-preference 4000`	BGP ローカル プリファレンス値を設定します。範囲は 0 ～ 4294967295 です。
set metric [+ | -] *bandwidth-metric* 例 ： `switch(config-route-map)# set metric +100`	既存のメトリック値を増減します。メトリックは Kb/s 単位です。範囲は 0 ～ 4294967295 です。

コマンド	目的
set metric *bandwidth* [*delay reliability load mtu*] 例 ： `switch(config-route-map)# set metric 33 44 100 200 1500`	ルート メトリック値を設定します。 メトリックは次のとおりです。 • *metric0*：帯域幅（Kb/s）。範囲は 0 ～ 4294967295 です。 • *metric1*：遅延（10 マイクロ秒単位）。 • *metric2*：信頼性。指定できる範囲は 0 ～ 255（100% の信頼性）です。 • *metric3*：ロード中。指定できる範囲は 1 ～ 200（100% のロード）です。 • *metric4*：パスの MTU。値の範囲は 1 ～ 4294967295 です。
set metric-type { **external** | **internal** | **type-1** | **type-2** } 例： `switch(config-route-map)# set metric-type internal`	宛先ルーティング プロトコルのメトリック タイプを設定します。オプションは次のとおりです。 external：IS-IS 外部メトリック internal：BGP の MED として IGP メトリックを使用 type-1：OSPF 外部タイプ 1 メトリック type-2：OSPF 外部タイプ 2 メトリック
set origin { **egp** *as-number* | **igp** | **incomplete**} 例： `switch(config-route-map)# set origin incomplete`	BGP オリジン属性を設定します。EGP *as-number* の範囲は 0 ～ 65535 です。
set tag *name* 例： `switch(config-route-map)# set tag 33`	宛先ルーティング プロトコルのタグ値を設定します。*name* パラメータは符号なし整数です。
set weight *count* 例： `switch(config-route-map)# set weight 33`	BGP ルートの重み値を設定します。範囲は 0 ～ 65535 です。

set metric-type internal コマンドは、発信ポリシーと eBGP ネイバーにのみ作用します。同じ BGP ピア発信ポリシーに **metric** コマンドと **metric-type internal** コマンドを両方設定した場合、Cisco NX-OS は **metric-type internal** コマンドを無視します。

Route Policy Manager の設定の確認

Route Policy Manager の設定情報を表示するには、次の作業のいずれかを行います。

コマンド	目的
show ip community-list [*name*]	コミュニティ リストの情報を表示します。
show ip ext community-list [*name*]	拡張コミュニティ リストの情報を表示します。
show [ip] prefix-list [*name*]	IPv4 プレフィックス リストの情報を表示します。
show route-map [*name*]	ルート マップの情報を表示します。

Route Policy Manager の設定例

次の例では、アドレス ファミリを使用して Route Policy Manager を設定し、ネイバー 209.0.2.1 からのユニキャスト ルートやマルチキャスト ルートが AllowPrefix プレフィックス リストと一致した場合に、それらのルートが承認されるようにします。

```
router bgp 64496

neighbor 172.16.0.1 remote-as 64497
  address-family ipv4 unicast
    route-map filterBGP in

route-map filterBGP
 match ip address prefix-list AllowPrefix

ip prefix-list AllowPrefix 10 permit 192.0.2.0/24
ip prefix-list AllowPrefix 20 permit 172.16.201.0/27
```

関連項目

Route Policy Manager の詳細については、次の項目を参照してください。

- 基本的 BGP の設定

その他の参考資料

IP の実装に関する詳細情報については、次の各項を参照してください。

- 関連資料
- 標準

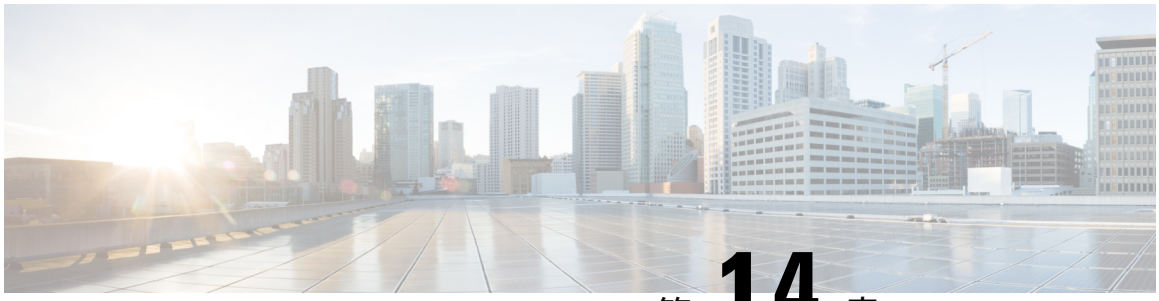
関連資料

関連項目	マニュアル タイトル
Route Policy Manager CLI コマンド	『Cisco Nexus 3000 Series Command Reference』

標準

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

標準



第 14 章

双方向フォワーディング検出の設定

この章では、双方向フォワーディング検出（BFD）を設定する方法について説明します。

この章は、次の項で構成されています。

BFD に関する情報

BFD は、メディア タイプ、カプセル化、トポロジ、およびルーティング プロトコルの転送パス障害を高速で検出するように設計された検出プロトコルです。BFDを使用することで、さまざまなプロトコルの Hello メカニズムにより、変動速度ではなく一定速度で転送パス障害を検出できます。BFDはプロファイリングおよびプランニングを簡単にし、再コンバージェンス時間の一貫性を保ち、予測可能にします。

BFD では、2 台の隣接デバイス間のサブセカンド障害を検出します。

非同期モード

Cisco NX-OS は、BFD 非同期モードをサポートします。BFD 非同期モードでは、2 個の隣接するデバイス間で BFD 制御パケットが送信され、デバイス間の BFD ネイバー セッションがアクティベートされ、維持されます。両方のデバイス（または BFD ネイバー）で BFD を設定できます。適切なプロトコルで一度 BFD がイネーブルになると、Cisco NX-OS は BFD セッションを作成し、BFD セッション パラメータをネゴシエートし、BFD 制御パケットをネゴシエートされた間隔で各 BFD ネイバーに送信し始めます。BFD セッション パラメータは、次のとおりです。

- 目的の最小送信間隔：このデバイスが BFD Hello メッセージを送信する間隔。

- 必要最小受信間隔：このデバイスが別の BFD デバイスからの BFD Hello メッセージを受け付ける最小間隔。
- 検出乗数：転送パスの障害を検出するまでに喪失した、別の BFD デバイスからの BFD Hello メッセージの数。

BFD の障害検出

一度 BFD セッションが確立され、タイマー ネゴシエーションが終了すると、BFD ネイバーは、より速い速度の場合を除き IGP Hello プロトコルと同じ動作をする BFD 制御パケットを送信し、活性度を検出します。BFDは障害を検出しますが、プロトコルが障害の発生したピアをバイパスするための処置を行う必要があります。

BFD は転送パスに障害を検出したとき、障害検出通知を BFD 対応プロトコルに送信します。ローカルデバイスは、プロトコル再計算プロセスを開始してネットワーク全体の収束時間を削減できます。

ネットワークで障害が発生すると、次のことが発生します。

1. 1. BFD 隣接ルータでの BFD ネイバー セッションが停止します。
2. 2. BFD はローカル BFD プロセスに BFD ネイバーに接続できなくなったことを通知します。
3. 3. ローカル BFD プロセスは BFD ネイバー関係を解除します。
4. 4. 代替パスが使用可能な場合、ルータはただちにそのパスでコンバージェンスを開始します。

（注） BFD 障害検出は 1 秒未満で行われます。

BFD エコー機能

BFD エコー機能は、転送エンジンからリモート BFD ネイバーにエコー パケットを送信します。BFD ネイバーは検出を実行するために同じパスに沿ってエコー パケットを返送します。BFD ネイバーは、エコー パケットの実際の転送に参加しません。エコー機能および転送エンジンが検出の処理を行います。BFDはエコー機能がイネーブルになっている場合に非同期セッションの速度を低下させ、2 台の BFD ネイバー間で送信される BFD 制御パケット数を減らすために、slow timer を使用できます。また、転送エンジンは、リモート システムを含めないでリモート（ネイバー）システムの転送パスをテストするので、パケット間遅延の変動が少なくなり、障害検出時間が短縮されます。

BFD ネイバーの両方がエコー機能を実行している場合、エコー機能は非対称になります。

セキュリティ

Cisco NX-OS は BFD パケットを隣接する BFD ピアから受信したことを確認するためにパケットの存続可能時間（TTL）値を使用します。すべての非同期およびエコー要求パケットの場合、BFD ネイバーは TTL 値を 255 に設定し、ローカル BFD プロセスは着信パケットを処理する前に TTL 値を 255 として確認します。エコー応答パケットの場合、BFD は TTL 値を 254 に設定します。

仮想化のサポート

BFD は、仮想ルーティングおよび転送（VRF）インスタンスをサポートしています。

BFD の前提条件

BFD には、次の前提条件があります。

- BFD機能を有効にする必要があります（BFD機能のイネーブル化のセクションを参照）。
- クライアント プロトコル上でBFD を有効にする場合は、そのクライアント プロトコルの BFD を有効にします。
- BFD 対応インターフェイスでインターネット制御メッセージ プロトコル（ICMP）リダイレクト メッセージをディセーブルにします。
- 設定作業とともに一覧表示されているその他の詳細な前提条件を参照してください。
- Cisco NX-OS リリース 6.0(2)A4(1) 以降、BGP および PIM の BFD がサポートされています。

注意事項と制約事項

BFD 設定時のガイドラインと制約事項は次のとおりです。

- BFD は BFD バージョン 1 をサポートします。
- BFD は、IPv4、BGPv4、PIM、およびスタティック ルートをサポートします。
- BFD は、シングルホップ BFD をサポートします。
- BGP の BFD は、送信元更新でシングル ホップ eBGP および iBGP をサポートします。
- BFDは、レイヤ3インターフェイスとして、物理インターフェイス、ポートチャネル、サブインターフェイス、および VLAN インターフェイス（SVI）をサポートします。
- BFD は、ポート チャネル上の BFD の認証またはリンクごとの BFD セッションはサポートしません。

- BFD はレイヤ 3 隣接情報に応じて、レイヤ 2 のトポロジ変更を含むトポロジ変更を検出します。レイヤ 3 隣接情報が使用できない場合、VLAN インターフェイス（SVI）の BFD セッションはレイヤ 2 トポロジのコンバージェンス後に稼働しない可能性があります。
- ポート チャネル設定の制限事項

BFD で使用されるレイヤ 3 ポート チャネルでは、ポート チャネルの Link Aggregation Control Protocol（LACP）を有効にする必要があります。

SVI セッションで使用されるレイヤ 2 ポートチャネルでは、ポートチャネルの LACP を有効にする必要があります。

- SVI の制限事項

トポロジを変更すると（たとえば、VLANへのリンクの追加または削除、レイヤ2ポートチャネルからのメンバの削除など）、SVIセッションが影響を受ける場合があります。SVIセッションはダウンした後、トポロジ ディスカバリの終了後に起動する場合があります。

ヒント SVIのセッションがフラップしないようにし、トポロジを変更する必要がある場合は、変更を加える前にBFD機能をディセーブルにして、変更後、BFDを再度イネーブルにできます。また、大きな値（たとえば、5秒）になるようにBFDタイマーを設定し、上記のイベントの完了後に高速なタイマーに戻すこともできます。

- Cisco NX-OS は、BFD パケット処理の CPU 負荷軽減のための、互換モジュールへの BFD 動作の分散は行いません。
- BFDはステートレス リスタートおよびインサービス ソフトウェア アップグレード（ISSU）をサポートしません。
- ポート チャネルを介して到達可能なピアの BFD を有効にする場合は、ポート チャネルで LACP を構成する必要があります。
- Cisco Nexus 3548 は、BFD エコー パケットを転送する中間スイッチになることはできません。

デフォルト設定

次の表に、BFD パラメータのデフォルト設定値を示します。

表 *16*：デフォルトの *BFD* パラメータ

パラメータ	デフォルト
BFD 機能	ディセーブル
必要最小受信間隔	250 ミリ秒

パラメータ	デフォルト
目的の最小送信間隔	250 ミリ秒
検出乗数	3
エコー機能	イネーブル
モード	非同期
ポート チャネル	論理モード（送信元/宛先ペアのアドレスごとに 1 セッション）
slow timer	2000 ミリ秒
サブインターフェイスの最適化	ディセーブル

BFD の設定

このセクションは、次のトピックで構成されています。

設定階層

BFD は、グローバル レベル、VRF のレベル、インターフェイスまたはポート チャネル レベル、またはサブインターフェイス レベルで設定できます（物理インターフェイスとポート チャネルの場合）。VRF の設定はグローバル設定よりも優先されます。インターフェイスまたはポート チャネルの設定は、VRF またはグローバル設定よりも優先されます。サポートされているインターフェイス上での、サブインターフェイス レベルの設定は、サブインターフェイスの最適化がイネーブルになっていない限りインターフェイスまたはポート チャネル設定よりも優先されます。詳細については、サブインターフェイスの BFD の最適化のセクションを参照してください。

ポート チャネルのメンバである物理ポートについては、メンバ ポートはプライマリ ポート チャネルの BFD 設定を継承します。メンバー ポート サブインターフェイスは、サブインターフェイスの最適化がイネーブルになっていない限り、マスター ポート チャネルの BFD 設定より優先させることができます。

BFD 設定のタスク フロー

BFD の設定には、次の作業を行います。

ステップ 1 BFD 機能のイネーブル化

ステップ 2 グローバルな BFD パラメータの設定またはインターフェイスでの BFD の設定

ステップ 3 BGP での BFD の設定

BFD 機能のイネーブル化

インターフェイスとプロトコルの BFD を設定する前に、BFD 機能をイネーブルにする必要があります。

手順の概要

1. **configure terminal**
2. **feature bfd**
3. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**feature bfd** 例： `switch(config)# feature bfd`	BFD 機能をイネーブルにします。
ステップ 3	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

グローバルな BFD パラメータの設定

デバイスのすべての BFD セッションの BFD セッション パラメータを設定できます。BFD セッション パラメータは、スリーウェイ ハンドシェイクの BFD ピア間でネゴシエートされます。

インターフェイスでこれらのグローバルなセッション パラメータを上書きするには、インターフェイスでの BFD の設定のセクションを参照してください。

始める前に

BFD機能をイネーブルにします。BFD機能のイネーブル化のセクションを参照してください。

手順の概要

1. **configure terminal**

2. **bfd interval** *mintx* **min_rx** *msec* **multiplier** *value*
3. **bfd slow-timer** *interval*
4. **exit**
5. （任意） **show running-config bfd**
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**bfd interval** *mintx* **min_rx** *msec* **multiplier** *value* 例： `switch(config)# bfd interval 250 min_rx 250 multiplier 3`	デバイスのすべてのBFDセッションのBFDセッション パラメータを設定します。インターフェイスでBFDセッションパラメータを設定することにより、これらの値を上書きすることができます。*mintx* および *msec* の範囲は 250 ～ 999 ミリ秒で、デフォルトは 250 です。乗数の範囲は 3 ～ 50 です。乗数のデフォルトは 3 です。 デフォルト設定に戻すには、**no bfd interval** コマンドを使用します。
ステップ 3	**bfd slow-timer** *interval* 例： `switch(config)# bfd slow-timer 2000`	スロー タイマーを設定します。この値は BFD が新しいセッションを開始する速度を決定し、BFD エコー機能がイネーブルの場合に非同期セッションの速度を低下させるために使用されます。指定できる範囲は1000～30000ミリ秒です。デフォルトは2000です。 デフォルト設定に戻すには、no bfd slow-timer コマンドを使用します。

ステップ 4	**exit**
	例：
	`switch(config)# exit`
	`switch#`
		EXEC モードに戻ります。

ステップ 5	（任意） **show running-config bfd** 例： `switch# show running-config bfd`	BFD の実行構成を表示します。
ステップ 6	（任意） **copy running-config startup-config** 例：	この設定変更を保存します。

	コマンドまたはアクション	目的
	`switch# copy running-config startup-config`	

インターフェイスでの BFD の設定

BFD機能をイネーブルにします。BFD機能のイネーブル化のセクションを参照してください。

始める前に

インターフェイスのすべてのBFDセッションのBFDセッション パラメータを設定できます。BFDセッション パラメータは、スリーウェイ ハンドシェイクのBFDピア間でネゴシエートされます。

この設定は、設定されたインターフェイスのグローバル セッション パラメータより優先されます。

手順の概要

1. **configure terminal**
2. **interface** *int-if*
3. **bfd interval** *mintx* **min_rx** *msec* **multiplier** *value*
4. **exit**
5. **exit**
6. （任意） **show running-config bfd**
7. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**interface** *int-if*
	例：
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		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。**?** キーワードを使用して、サポートされるインターフェイスを表示します。

ステップ 3	**bfd interval** *mintx* **min_rx** *msec* **multiplier** *value* 例： `switch(config-if)# bfd interval 250 min_rx 250 multiplier 3`	インターフェイスのすべてのBFDセッションのBFDセッションパラメータを設定します。このコマンドはグローバルなBFD セッション パラメータより優先されます。*mintx* および *msec* の範囲は 250 ～ 999 ミリ秒で、デフォルトは 250 です。乗数の範囲は 3 ～ 50 です。乗数のデフォルトは 3 です。

	コマンドまたはアクション	目的
		デフォルト設定に戻すには、**no bfd interval** コマンドを使用します。

ステップ 4	**exit**
	例：
	`switch(config-if)# exit`
	`switch (config)#`
		インターフェイス コンフィギュレーション モードを終了します。

ステップ 5	**exit**
	例：
	`switch (config)# exit`
	`switch#`
		コンフィギュレーション モードを終了し、EXEC モードに戻ります。

ステップ 6	（任意） **show running-config bfd** 例： `switch# show running-config bfd`	BFD の実行構成を表示します。
ステップ 7	（任意） **copy running-config startup-config** 例： `switch# copy running-config startup-config`	この設定変更を保存します。

ポート チャネルの BFD の設定

ポート チャネルのすべての BFD セッションの BFD セッション パラメータを設定できます。たとえば、ポート チャネルの 1 つのリンクの BFD セッションが稼働している場合、BGP などのクライアント プロトコルにポート チャネルが稼働していることが通知されます。BFD セッション パラメータは、スリーウェイ ハンドシェイクの BFD ピア間でネゴシエートされます。

この設定は、設定されたポート チャネルのグローバル セッション パラメータより優先されます。ポートチャネルのメンバー ポートは、メンバー ポートのサブインターフェイス レベルで BFD パラメータを設定しない限り、ポートチャネルの BFD セッション パラメータを継承します。その場合、サブインターフェイス最適化がイネーブルにされていなければ、メンバポート サブインターフェイスはサブインターフェイス BFD コンフィギュレーションを使用します。詳細については、サブインターフェイスの BFD の最適化のセクションを参照してください。

始める前に

BFD をイネーブルにする前に、ポート チャネルの Link Aggregation Control Protocol（LACP）がイネーブルにされていることを確認します。

BFD機能をイネーブルにします。BFD機能のイネーブル化のセクションを参照してください。

手順の概要

1. **configure terminal**

2. **interface port-channel** *number*
3. （任意） **bfd interval** *mintx* **min_rx** *msec* **multiplier** *value*
4. **exit**
5. **exit**
6. （任意） **show running-config bfd**
7. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**interface port-channel** *number*
	例：
	`switch(config)# interface port-channel 2`
	`switch(config-if)#`
		ポート チャネル コンフィギュレーション モードを開始します。**?** キーワードを使用して、サポートされる数値の範囲を表示します。

ステップ 3	（任意） **bfd interval** *mintx* **min_rx** *msec* **multiplier** *value* 例： `switch(config-if)# bfd interval 250 min_rx 250 multiplier 3`	インターフェイスのすべてのBFDセッションのBFDセッションパラメータを設定します。このコマンドはグローバルな BFD セッション パラメータより優先されます。*mintx* および *msec* の範囲は 250 ～ 999 ミリ秒で、デフォルトは 250 です。乗数の範囲は 3 ～ 50 です。乗数のデフォルトは 3 です。 デフォルト設定に戻すには、**no bfd interval** コマンドを使用します。

ステップ 4	**exit**
	例：
	`switch(config-if)# exit`
	`switch (config)#`
		インターフェイス コンフィギュレーション モードを終了します。

ステップ 5	**exit**
	例：
	`switch (config)# exit`
	`switch#`
		コンフィギュレーション モードを終了し、EXEC モードに戻ります。

ステップ 6	（任意） **show running-config bfd** 例： `switch# show running-config bfd`	BFD の実行構成を表示します。
ステップ 7	（任意） **copy running-config startup-config** 例：	この設定変更を保存します。

	コマンドまたはアクション	目的
	`switch# copy running-config startup-config`	

BFD エコー機能の設定

BFD モニタ対象リンクの一端または両端で BFD エコー機能を設定できます。エコー機能は設定された slow timer に基づいて必要最小受信間隔を遅くします。[RequiredMinEchoRx] BFD セッション パラメータは、エコー機能がディセーブルの場合、ゼロに設定されます。slow timer は、エコー機能がイネーブルの場合、必要最小受信間隔になります。

始める前に

BFD機能をイネーブルにします。BFD機能のイネーブル化のセクションを参照してください。

BFD セッション パラメータを設定します。グローバルな BFD パラメータの設定のセクション、またはインターフェイスでの BFD の設定のセクションを参照してください。

インターネット制御メッセージ プロトコル（ICMP）のリダイレクト メッセージが BFD 対応インターフェイスでディセーブルであることを確認します。インターフェイスで **no ip redirects** コマンドを使用します。

手順の概要

1. **configure terminal**
2. **bfd slow-timer** *echo-interval*
3. **interface** *int-if*
4. **bfd echo**
5. **exit**
6. **exit**
7. （任意） **show running-config bfd**
8. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**bfd slow-timer** *echo-interval* 例： `switch(config)# bfd slow-timer 2000`	エコー機能で使用される slow timer を設定します。この値はBFDが新しいセッションを開始する速度を決定し、BFDエコー機能がイネーブルの場合に非同期セッションの速度を低下させるために使用されます。この値は、エコー機能がイネーブルの場合、必要最小受信間隔より優先されます。指定できる範囲

	コマンドまたはアクション	目的
		は 1000 ～ 30000 ミリ秒です。デフォルトは 2000 です。 デフォルト設定に戻すには、no bfd slow-timer コマンドを使用します。

ステップ 3	**interface** *int-if*
	例：
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		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。**?** キーワードを使用して、サポートされるインターフェイスを表示します。

ステップ 4	**bfd echo** 例： `switch(config-if)# bfd echo`	エコー機能をイネーブルにします。デフォルトではイネーブルになっています。 エコー機能をディセーブルにするには、no bfd echo コマンドを使用します。

ステップ 5	**exit**
	例：
	`switch(config-if)# exit`
	`switch (config)#`
		インターフェイス コンフィギュレーション モードを終了します。

ステップ 6	**exit**
	例：
	`switch (config)# exit`
	`switch#`
		コンフィギュレーション モードを終了し、EXEC モードに戻ります。

ステップ 7	（任意） **show running-config bfd** 例： `switch# show running-config bfd`	BFD の実行構成を表示します。
ステップ 8	（任意） **copy running-config startup-config** 例： `switch# copy running-config startup-config`	この設定変更を保存します。

サブインターフェイスの BFD の最適化

サブインターフェイスの BFD は最適化できます。BFD により、設定されているすべてのサブインターフェイスのセッションが作成されます。BFD により、設定されている最小の VLAN ID を持つサブインターフェイスがプライマリ サブインターフェイスとして設定され、そのサブインターフェイスは親インターフェイスの BFD セッション パラメータを使用します。残りのサブインターフェイスは slow timer を使用します。最適化サブインターフェイス セッションでエラーが検出されると、BFD により、その物理インターフェイスのすべてのサブインターフェイスがダウンとマークされます。

始める前に

BFD機能をイネーブルにします。BFD機能のイネーブル化のセクションを参照してください。

BFD セッション パラメータを設定します。グローバルな BFD パラメータの設定のセクション、またはインターフェイスでの BFD の設定のセクションを参照してください。

これらのサブインターフェイスが別のCisco NX-OSデバイスに接続されていることを確認します。この機能は、Cisco NX-OS でのみサポートされます。

手順の概要

1. **configure terminal**
2. **interface** *int-if*
3. **bfd optimize subinterface**
4. **exit**
5. **exit**
6. （任意） **show running-config bfd**
7. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**interface** *int-if*
	例：
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		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。**?** キーワードを使用して、サポートされるインターフェイスを表示します。

ステップ 3	**bfd optimize subinterface** 例： `switch(config-if)# bfd optimize subinterface`	BFD対応インターフェイスのサブインターフェイスを最適化します。デフォルトではディセーブルになっています。 最適化されたサブインターフェイスを無効にするには、 **no bfd optimize subinterfaces** コマンドを使用します。

ステップ 4	**exit**
	例：
	`switch(config-if)# exit`
	`switch (config)#`
		インターフェイス コンフィギュレーション モードを終了します。

ステップ 5	**exit** 例：	コンフィギュレーション モードを終了し、EXEC モードに戻ります。

	コマンドまたはアクション	目的
	`switch (config)# exit` `switch#`	
ステップ 6	（任意） **show running-config bfd** 例： `switch# show running-config bfd`	BFD の実行構成を表示します。
ステップ 7	（任意） **copy running-config startup-config** 例： `switch# copy running-config startup-config`	この設定変更を保存します。

BGP での BFD の設定

ボーダー ゲートウェイ プロトコル（BGP）の BFD を設定できます。

始める前に

BFD機能をイネーブルにします。BFD機能のイネーブル化のセクションを参照してください。

BFD セッション パラメータを設定します。グローバルな BFD パラメータの設定のセクション、またはインターフェイスでの BFD の設定のセクションを参照してください。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** { *ip-address* } **remote-as** *as-number*
4. **bfd**
5. （任意） **show running-config bfd**
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**router bgp** *as-number*
	例：
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		BGP を有効にして、ローカル BGP スピーカに AS 番号を割り当てます。AS 番号は 16 ビット整数または 32 ビット整数にできます。上位 16 ビット 10 進数と下位 16 ビット 10 進数による xx.xx という形式です。

	コマンドまたはアクション	目的

ステップ 3	**neighbor** { *ip-address* } **remote-as** *as-number*
	例：
	`switch(config-router)# neighbor 209.165.201.1 remote-as 64497`
	`switch(config-router-neighbor)#`
		リモート BGP ピアの IPv4 アドレスおよび AS 番号を設定します。The *ip-address* 形式は x.x.x.x です。

ステップ 4	**bfd** 例： `switch(config-router-neighbor)# bfd`	この BGP ピアの BFD をイネーブルにします。
ステップ 5	（任意） **show running-config bfd** 例： `switch# show running-config bfd`	BFD の実行構成を表示します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch# copy running-config startup-config`	この設定変更を保存します。

PIM での BFD の設定

PIM（Protocol Independent Multicast）プロトコルの BFD を設定できます。

始める前に

BFD機能をイネーブルにします。BFD機能のイネーブル化のセクションを参照してください。

PIM 機能をイネーブルにします。詳細については、『Cisco Nexus 3548 スイッチ NX-OS マルチキャスト ルーティング構成ガイド』を参照してください。

手順の概要

1. **configure terminal**
2. **ip pim bfd**
3. **interface** *type slot/port*
4. （任意） **ip pim bfd-instance** [**disable**]
5. （任意） **show running-config pim**
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例：	グローバル 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
	`switch# configure terminal` `switch(config)#`	
ステップ 2	**ip pim bfd** 例： `switch(config)# ip pim bfd`	PIM の BFD をイネーブルにします。

ステップ 3	**interface** *type slot/port*
	例：
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		インターフェイス設定モードを開始します。**?** キーワードを使用して、サポートされるインターフェイスを表示します。

ステップ 4	（任意） **ip pim bfd-instance** [**disable**] 例： `switch(config-if)# ip pim bfd-instance`	PIM インターフェイスの BFD をイネーブルまたはディセーブルにします。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	（任意） **show running-config pim** 例： `switch(config)# show running-config pim`	PIM の実行コンフィギュレーションを表示します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch# copy running-config startup-config`	この設定変更を保存します。

OSPFv2 での BFD の設定

Open Shortest Path First Protocol（OSPFv2）で BFD を設定できます。

始める前に

BFD 機能をイネーブルにします。

グローバルに、または特定のインターフェイスに対して、BFD セッション パラメータを設定します。

OSPFv2 機能を有効にします。

手順の概要

1. **configure terminal**
2. **router ospf** *process-id*
3. **bfd**
4. **interface** *int-if*
5. （任意） **[no] ip ospf bfd disable**
6. （任意） **show running-config ospf**

7. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**router ospf** *process-id*
	例：
	`switch(config)# router ospf 64496`
	`switch(config-router)#`
		設定された ID で新しい OSPFv2 プロセスを作成します。

ステップ 3	**bfd** 例： `switch(config-router)# bfd`	この OSPFv2 ピアの BFD を有効にします。デフォルト値は [無効（Disabled）] です。

ステップ 4	**interface** *int-if*
	例：
	`switch(config-router)# interface ethernet 1/2`
	`switch (config-if)#`
		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。? キーワードを使用して、サポートされるインターフェイスを表示します。

ステップ 5	（任意） **[no] ip ospf bfd disable** 例： `switch(config-if)# ip ospf bfd disable`	OSPFv2 インターフェイスで BFD をディセーブルにします。デフォルト値はイネーブルです。
ステップ 6	（任意） **show running-config ospf** 例： `switch(config)# show running-config ospf`	OSPFv2 実行設定を表示します。
ステップ 7	（任意） **copy running-config startup-config** 例： `switch# copy running-config startup-config`	この設定変更を保存します。

スタティック ルートの BFD の設定

インターフェイスのスタティック ルータの BFD を設定できます。Virtual Routing and Forwarding（VRF）インスタンス内のスタティック ルートでの BFD を任意で設定できます。

始める前に

BFD機能をイネーブルにします。BFD機能のイネーブル化のセクションを参照してください。

手順の概要

1. **configure terminal**
2. （任意） **vrf context** *vrf-name*
3. **ip route** *route interface* { *nh-address* | *nh-prefix* }
4. **ip route static bfd** *interface* {*nh-address* | *nh-prefix*}
5. （任意） **show ip route static** [**vrf** *vrf-name*]
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル設定モードを開始します。

ステップ 2	（任意） **vrf context** *vrf-name*
	例：
	`switch(config)# vrf context Red`
	`switch(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	**ip route** *route interface* { *nh-address*	*nh-prefix* } 例： `switch(config-vrf)# ip route 192.0.2.1 ethernet 2/1 192.0.2.4`	スタティック ルートを作成します。**?** キーワードを使用して、サポートされているインターフェイスを表示します。
ステップ 4	**ip route static bfd** *interface* {*nh-address*	*nh-prefix*} 例： `switch(config-vrf)# ip route static bfd ethernet 2/1 192.0.2.4`	インターフェイスのすべてのスタティック ルートの BFD をイネーブルにします。**?** キーワードを使用して、サポートされるインターフェイスを表示します。
ステップ 5	（任意） **show ip route static** [**vrf** *vrf-name*] 例： `switch(config-vrf)# show ip route static vrf Red`	スタティック ルートを表示します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch# copy running-config startup-config`	この設定変更を保存します。

BFD 設定の確認

BFD 設定情報を表示するには、次の作業のいずれかを行います。

コマンド	目的
show running-config bfd	実行 BFD コンフィギュレーションを表示します。
show startup-config bfd	次のシステム起動時に適用される BFD コンフィギュレーションを表示します。

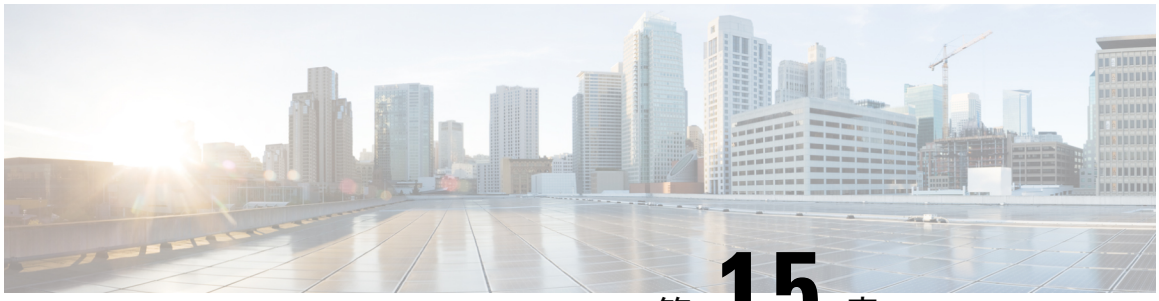
これらのコマンドの出力フィールドの詳細については、Cisco Nexus 3548 スイッチ NX-OX インターフェイス コマンド リファレンス リリース 6.xを参照してください。

BFD のモニタリング

BFD を表示するには、次のコマンドを使用します。

コマンド	目的
show bfd neighbors [**application** *name*] [**details**]	BGP などのサポートされるアプリケーションの BFD に関する情報を表示します。
show bfd neighbors [**interface** *int-if*] [**details**]	インターフェイスの BGP セッションに関する情報を表示します。
show bfd neighbors [**dest-ip** *ip-address*] [**src-ip** *ip-address*][*details*]	インターフェイス上の指定された BGP セッションに関する情報を表示します。
show bfd neighbors [**vrf** *vrf-name*] [**details**]	VRF の BFD に関する情報を表示します。

これらのコマンドの出力フィールドの詳細については、Cisco Nexus 3548 スイッチ コマンド リファレンスを参照してください。



第 15 章

ポリシーベース ルーティングの設定

この章では、Cisco NX-OS デバイスでポリシー ベース ルーティングを設定する方法について説明します。

この章は、次の項で構成されています。

ポリシーベース ルーティングの概要

ポリシーベース ルーティングを使用すると、IPv4 トラフィック フローに定義済みのポリシーを設定し、ルーティング プロトコルから派生したルートへの依存度を弱めることができます。ポリシーベース ルーティングが有効のインターフェイスで受信したすべてのパケットは、拡張パケット フィルタまたはルート マップを経由して渡されます。ルート マップでは、パケットの転送先を決定するポリシーを記述します。

ルート マップは match 文および set 文からなり、許可または拒否を指定できます。文の解釈は次のとおりです。

- パケットがいずれかの route map 文と一致した場合、すべての set 文が適用されます。アクションには、ネクスト ホップの選択が含まれます。

- 文が permit とマークされており、パケットがいずれの route-map 文とも一致しない場合、そのパケットは通常の転送チャネルを介して返送され、接続先ベースのルーティングが実行されます。

詳細については、ルート マップを参照してください。

ポリシーベース ルーティングには、次の機能が含まれます。

- 送信元ベース ルーティング：異なるユーザ セットを起点とするトラフィックをポリシー ルータ上のそれぞれ異なる接続を使用してルーティングします。
- ロードシェアリング：トラフィックの特性に基づいて、複数のパスにトラフィックを分散します。

ポリシー ルート マップ

ルート マップは、さまざまなルーティング プロトコルや、特定のルーティング プロトコル内のさまざまなエンティティ間で配布されたルートのフィルタリングで使用されます。ルート マップのエントリごとに、match 文と set 文の組み合わせが 1 つずつ含まれています。match 文では、該当するパケットが特定のポリシーを満たす基準（つまり、満たすべき条件）を定義します。set 文節で、match 基準を満たしたパケットをどのようにルーティングするかを説明します。

ルートマップ文を許可または拒否として指定できます。文に拒否が指定されている場合、一致基準を満たすパケットは標準のフォワーディング チャネルを通じて送り返されます（宛先ベース ルーティングが実行されます）。文に許可が指定されていて、なおかつパケットが一致基準を満たしている場合は、すべての set 文節が適用されます。文に許可が指定されていて、なおかつパケットが一致基準を満たしていない場合は、それらのパケットも標準のルーティング チャネルを通じて転送されます。

（注） ポリシールーティングは、パケットの送信元となるインターフェイスではなく、パケットを受信するインターフェイス上で指定します。

ポリシーベース ルーティングの set 基準

ルート マップの set 基準は、ルート マップに指定された順番で評価されます。ポリシーベース ルーティング用のルート マップに固有の set 基準は、次のとおりです。

1. 指定 IP アドレスのリスト：IP アドレスでは、パケットの転送先である宛先へのパス上の隣接ネクストホップ ルータを指定できます。その時点でアップの接続インターフェイスに関連付けられた最初の IP アドレスがパケットのルーティングに使用されます。

（注） 任意で、最大 16 の IP アドレスにロード バランシングを行うように、ネクストホップ アドレスの set 基準を設定できます。この場合、Cisco NX-OS は各 IP フローのすべてのトラフィックを特定の IP ネクスト ホップ アドレスに送信します。

2. NULL インターフェイス：set null インターフェイスを使用すると、match ステートメントに一致するトラフィックがドロップされます。

   パケットが定義された一致基準のいずれにも一致しない場合、そのパケットは標準の宛先ベース ルーティング プロセスを使用してルーティングされます。

ポリシーベース ルーティングの前提条件

ポリシーベース ルーティングの前提条件は、次のとおりです。

- 有効なライセンスをインストールします。
- ポリシーベース ルーティングをイネーブルにする必要があります（ポリシーベース ルーティング機能のイネーブル化の項を参照）。
- インターフェイスに IP アドレスを割り当て、インターフェイスをアップにしてから、ポリシーベース ルーティング用のルート マップをインターフェイス上で適用します。

ポリシーベース ルーティングの注意事項と制約事項

ポリシーベース ルーティングに関する注意事項および制約事項は、次のとおりです。

- match コマンドで、ポリシーベース ルーティング用ルート マップの複数の ACL を参照できません。
- ポリシーベース ルーティングのルート マップで使用する ACL には、deny 文を含めることができません。
- インターフェイスが同じ仮想ルーティング/転送（VRF）インスタンスに所属している場合は、ポリシーベース ルーティング対応のさまざまなインターフェイス間で、同じルート マップを共有できます。

デフォルト設定

下の表に、ポリシーベース ルーティング パラメータのデフォルト設定を示します。

ポリシーベース ルーティングの設定

表 *17:* デフォルトのポリシーベース ルーティング パラメータ

パラメータ	デフォルト
ポリシーベース ルーティング	ディセーブル

ポリシーベース ルーティングの設定

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

ポリシーベース ルーティング機能のイネーブル化

ルート ポリシーを設定する前に、ポリシーベース ルーティング機能をイネーブルにしておく必要があります。

手順の概要

1. **configure terminal**
2. **feature pbr**
3. （任意） **show feature**
4. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**feature pbr** 例： `switch(config)# feature pbr`	ポリシーベース ルーティング機能をイネーブルにします。
ステップ 3	（任意） **show feature** 例： `switch(config)# show feature`	有効および無効にされた機能を表示します。

	コマンドまたはアクション	目的
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

no feature pbr コマンドを使用して、ポリシーベースのルーティング機能をディセーブルにし、関連するコンフィギュレーションをすべて削除します。

コマンド	目的
no feature pbr 例： `switch(config)# no feature pbr`	ポリシーベース ルーティングをディセーブルにして、関連するすべての設定を削除します。

ルート ポリシーの設定

ポリシーベース ルーティングでルートマップを使用すると、着信インターフェイスにルーティング ポリシーを割り当てることができます。ルート マップの設定のセクションを参照してください。

手順の概要

1. **configure terminal**
2. **interface** *type slot/port*
3. **ip policy route-map** *map-name*
4. （任意） **exit**
5. （任意） **exit**
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		グローバル コンフィギュレーション モードを開始します。

ステップ 2	**interface** *type slot/port* 例：	インターフェイス設定モードを開始します。

	コマンドまたはアクション	目的
	`switch(config)# interface ethernet 1/2` `switch(config-if)#`	
ステップ 3	**ip policy route-map** *map-name* 例： `switch(config-if)# ip policy route-map Testmap`	IPv4 ポリシーベース ルーティング用のルート マップをインターフェイスに割り当てます。
ステップ 4	（任意） **exit** 例： `switch(config-route-map)# exit`	ルート マップ設定モードを終了します。
ステップ 5	（任意） **exit** 例： `switch(config)# exit`	グローバル コンフィギュレーション モードを終了します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config)# copy running-config startup-config`	この設定変更を保存します。

例

次に、インターフェイスにルート マップを追加する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip policy route-map Testmap
switch(config)# exit
switch(config)# copy running-config startup-config
```

ルート マップ設定モードで、オプションとして、ルート マップに次の match パラメータを設定できます。

コマンド	目的
match ip address access-list-name *name* [*name...*] 例： `switch(config-route-map)# match ip address access-list-name ACL1`	1 つまたは複数の IP アクセス コントロール リスト（ACL）に対して IPv4 アドレスを照合します。このコマンドはポリシーベース ルーティング用であり、ルート フィルタリングまたは再配布では無視されます。

ルート マップ設定モードで、オプションとして、ルート マップに次の set パラメータを設定できます。

コマンド	目的
set ip next-hop *address1* [*address2.. .*] { **load-share** } 例： `switch(config-route-map)# set ip next-hop 192.0.2.1`	ポリシーベース ルーティング用の IPv4 ネクストホップ アドレスを設定します。このコマンドでは、複数のアドレスが設定されている場合に、最初の有効なネクストホップアドレスが使用されます。 任意の **load-share** キーワードを使用して、最大 16 のネクスト ホップ アドレスにトラフィックのロード バランシングを行います。
set ip default next-hop *address1* [*address2...*] { **load-share** } 例： `switch(config-route-map)# set ip default next-hop 192.0.2.2`	宛先への明示的ルートがない場合に使用する、ポリシーベース ルーティング用の IPv4 ネクスト ホップ アドレスを設定します。このコマンドでは、複数のアドレスが設定されている場合に、最初の有効なネクスト ホップ アドレスが使用されます。 任意の **load-share** キーワードを使用して、最大 16 のネクスト ホップ アドレスにトラフィックのロード バランシングを行います。

Cisco NX-OS はネクストホップおよびインターフェイスを検出すると、ただちにパケットをルーティングします。

ポリシーベース ルーティングの設定の確認

ポリシーベース ルーティングの設定情報を表示するには、次のいずれかのタスクを実行します。

コマンド	目的
show ip policy [*name*]	IPv4 ポリシーに関する情報を表示します。
show route-map [*name*] *pbr-statistics*	ポリシー統計情報を表示します。

route-map *map-name* **pbr-statistics** を使用してポリシー統計情報を有効にします。**clear route-map** *map-name* **pbr-statistics** を使用してこれらのポリシー統計情報をクリアします。

ポリシーベース ルーティング統計情報の表示

ポリシーベース ルーティングの統計情報を表示するには、**show route-map** *rmap-name* **pbr-statistics** コマンドを使用します。統計情報は、ルート マップ シーケンスごとに維持されます。これは、特定のルート マップ シーケンスの一致条件に基づいてポリシー ルーティングされるパケット数を示します。デフォルトのルーティング テーブルを使用してルーティングした他のパケット

（set コマンドでは到達不能なネクスト ホップが原因の場合がある）もすべて表示されます。統計情報を表示する前に、PBR 統計情報の収集をオンにする必要があります。

次に、PBR 統計情報を表示する例を示します。

```
switch(config)# show route-map pbr-sample pbr-statistics
```

ポリシーベース ルーティング統計情報の消去

ルートマップの PBR 統計のために保持されているカウンタをクリアするには、**clear route-map** *rmap-name* **pbr-statistics** コマンドを使用します。

次の例では、すべての PBR 統計情報をクリアする方法を示します。

```
switch(config)# clear route-map pbr-sample pbr-statistics
```

ポリシーベース ルーティングの設定例

インターフェイス上で単純なルート ポリシーを設定する例を示します。

```
feature pbr
ip access-list pbr-sample
permit tcp host 10.1.1.1 host 192.168.2.1 eq 80
!
route-map pbr-sample
match ip address pbr-sample
set ip next-hop 192.168.1.1
!
route-map pbr-sample pbr-statistics

interface ethernet 1/2
ip policy route-map pbr-sample
```

次の出力で、この設定を確認します。

```
n3000# show route-map pbr-sample

route-map pbr-sample, permit, sequence 10
Match clauses:
ip address (access-lists): pbr-sample
Set clauses:
ip next-hop 192.168.1.1

n3000# show route-map pbr-sample pbr-statistics

route-map pbr-sample, permit, sequence 10
Policy routing matches: 84 packets
```

関連項目

ポリシーベース ルーティングの詳細については、次の項目を参照してください。

- Route Policy Manager の設定

その他の参考資料

IP の実装に関する詳細情報については、次の各項を参照してください。

- 関連資料
- 標準

関連資料

関連項目	マニュアル タイトル
ポリシーベース ルーティング CLI コマンド	Cisco Nexus 3000 シリーズ NX-OS ユニキャスト ルーティング コマンド リファレンス

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。	—

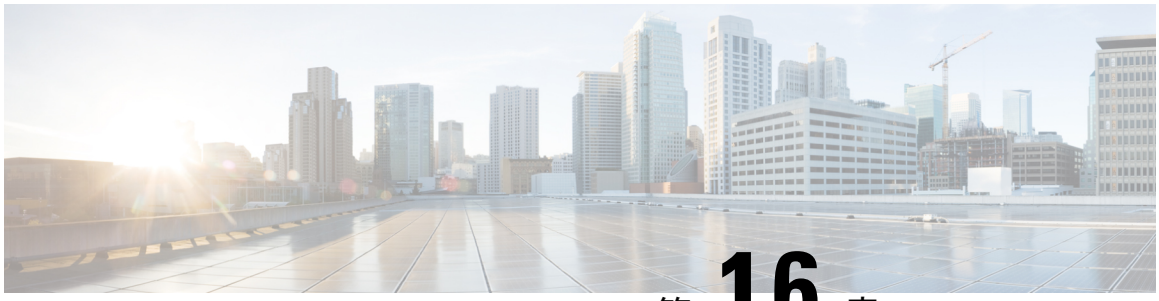
ポリシーベース ルーティングの機能の履歴.

次の表に、この機能のリリースの履歴を示します。

表 *18*：ポリシーベース ルーティングの機能の履歴.

機能名	リリース	機能情報
ポリシーベース ルーティング	6.0(2)A7(1)	この機能が導入されました。

ポリシーベース ルーティングの機能の履歴.



第 16 章

『Configuring HSRP』

この章では、Cisco NX-OS スイッチでホットスタンバイ ルータ プロトコル（HSRP）を設定する方法について説明します。

この章は、次の項で構成されています。

HSRP に関する情報

HSRP はファーストホップ冗長プロトコル（FHRP）であり、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します。HSRP は、デフォルト ルータの IP アドレスを指定して設定された、イーサネット ネットワーク上の IP ホストにファーストホップ ルーティングの冗長性を提供します。ルータ グループでは HSRP を使用して、アクティブ ルータおよびスタンバイ ルータを選択します。ルータ グループでは、アクティブ ルータはパケットをルーティングするルータです。スタンバイ ルータは、アクティブ ルータで障害が発生した場合、または事前に設定された条件が満たされた場合に、引き継ぐルータです。

大部分のホストの実装では、ダイナミックなルータ ディスカバリ メカニズムをサポートしていませんが、デフォルトのルータを設定することはできます。すべてのホスト上でダイナミックなルータ ディスカバリ メカニズムを実行するのは、管理上のオーバーヘッド、処理上のオーバーヘッド、セキュリティ上の問題など、さまざまな理由で適切ではありません。HSRP は、そうしたホスト上にフェールオーバー サービスを提供します。

HSRP の概要

HSRP を使用する場合、HSRP の仮想 IP アドレスを（実際のルータの IP アドレスではなく）ホストのデフォルト ルータとして設定します。仮想 IP アドレスは、HSRP が動作するルータのグループで共有される IPv4 アドレスです。

ネットワーク セグメントに HSRP を設定する場合は、HSRP グループ用の仮想 MAC アドレスと仮想 IP アドレスを設定します。グループの各 HSRP 対応インターフェイス上で、同じ仮想アドレスを指定します。各インターフェイス上で、実アドレスとして機能する固有の IP アドレスおよび MAC アドレスも設定します。HSRP はこれらのインターフェイスのうちの 1 つをアクティブ ルータにするために選択します。アクティブ ルータは、グループの仮想 MAC アドレス宛てのパケットを受信してルーティングします。

指定されたアクティブ ルータで障害が発生すると、HSRP によって検出されます。その時点で、選択されたスタンバイ ルータが HSRP グループの MAC アドレスおよび IP アドレスの制御を行うことになります。HSRPはこの時点で、新しいスタンバイ ルータの選択も行います。

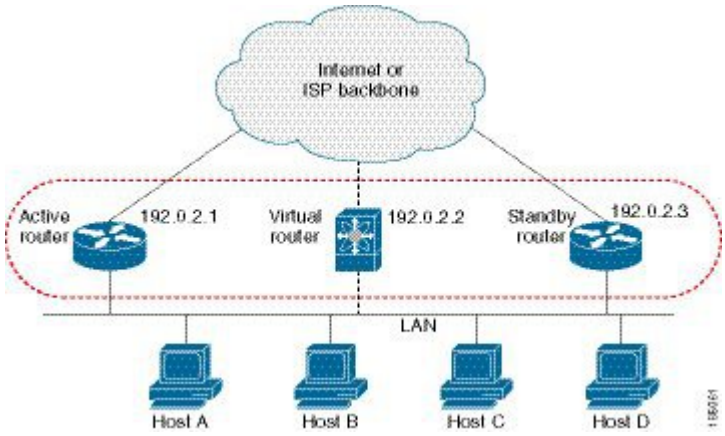
HSRPではプライオリティ メカニズムを使用して、デフォルトのアクティブルータにするHSRP設定インターフェイスを決定します。アクティブルータとしてインターフェイスを設定するには、グループ内の他のすべてのHSRP設定インターフェイスよりも高いプライオリティを与えます。デフォルトのプライオリティは100なので、それよりもプライオリティが高いインターフェイスを1つ設定すると、そのインターフェイスがデフォルトのアクティブルータになります。

HSRPが動作するインターフェイスは、マルチキャストユーザデータグラムプロトコル（UDP）ベースの hello メッセージを送受信して、障害を検出し、アクティブおよびスタンバイ ルータを指定します。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。アクティブ ルータとスタンバイ ルータ間のパケット フォワーディング機能の移動は、ネットワーク上のすべてのホストに対して完全に透過的です。

1 つのインターフェイス上で複数の HSRP グループを設定できます。

次の図に、HSRP 用に設定されたネットワークのセグメントを示します。仮想 MAC アドレスおよび仮想IPアドレスの共有によって、2つ以上のインターフェイスが単一の仮想ルータのように動作できます。

図 *21 : 2* 台の対応ルータを含む *HSRP* トポロジ



仮想ルータは物理的には存在しませんが、相互にバックアップするように設定されたインターフェイスにとって、共通のデフォルト ルータになります。アクティブ ルータの IP アドレスを使用して、LAN 上でホストを設定する必要はありません。代わりに、デフォルト ルータとして仮想ルータの IP アドレス（仮想 IP アドレス）を使用して、ホストを設定します。アクティブ ルータが設定時間内に hello メッセージを送信できなかった場合は、スタンバイ ルータが引き継いで仮想アドレスに応答し、アクティブ ルータになってアクティブ ルータの役割を引き受けます。ホストの観点からは、仮想ルータは同じままです。

（注） ルーテッド ポートで受信した HSRP 仮想 IP アドレス宛のパケットは、ローカル ルータ上で終端します。そのルータがアクティブ HSRP ルータであるのかスタンバイ HSRP ルータであるのかは関係ありません。これには ping トラフィックと Telnet トラフィックが含まれます。レイヤ 2（VLAN）インターフェイスで受信した HSRP 仮想 IP アドレス宛てのパケットは、アクティブ ルータ上で終端します。

HSRP for IPv4

HSRP ルータは、HSRP hello パケットを交換することによって相互に通信します。これらのパケットは、UDP ポート 1985 上の宛先 IP マルチキャスト アドレス 224.0.0.2（すべてのルータと通信するための予約済みマルチキャスト アドレス）に送信されます。アクティブ ルータは設定 IP アドレスおよび HSRP 仮想 MAC アドレスから hello パケットを得るのに対して、スタンバイ ルータは設定 IP アドレスおよびインターフェイス MAC アドレスから hello パケットを取得します。インターフェイス MAC アドレスは、バーンドイン アドレス（BIA）のこともあれば、そうではないこともあります。BIA は、MAC アドレスの下位 6 バイトで、ネットワーク カード（NIC）の製造元によって割り当てられます。

ホストはデフォルト ルータが HSRP 仮想 IP アドレスとして設定されているので、HSRP 仮想 IP アドレスに関連付けられた MAC アドレスと通信する必要があります。この MAC アドレスは、仮想 MAC アドレス 0000.0C07.ACxy です。この場合、xy はそれぞれのインターフェイスに基づく、16 進数の HSRP グループ番号です。たとえば、HSRP グループ 1 は 0000.0C07.AC01

という HSRP 仮想 MAC アドレスを使用します。隣接 LAN セグメント上のホストは、標準のアドレス解決プロトコル（ARP）プロセスを使用して、関連付けられた MAC アドレスを解決します。

HSRP バージョン 2 では新しい IP マルチキャスト アドレス 224.0.0.102 を使用して hello パケットを送信します。バージョン 1 では、このマルチキャスト アドレスが 224.0.0.2 です。バージョン 2 では、拡張グループ番号範囲 0 ～ 4095 を使用できます。また、新しい MAC アドレス範囲 0000.0C9F.F000 ～ 0000.0C9F.FFFF を使用します。

HSRP のバージョン

Cisco NX-OS は、デフォルトでは HSRP バージョン 1 をサポートしています。HSRP バージョン 2 を使用するようにインターフェイスを設定できます。

HSRP バージョン 2 では、HSRP バージョン 1 から次のように拡張されています。

- グループ番号の範囲が拡大されました。HSRP バージョン 1 がサポートするグループ番号は 0 ～ 255 です。HSRP バージョン 2 がサポートするグループ番号は 0 ～ 4095 です。
- IPv4 では IPv4 マルチキャスト アドレス 224.0.0.102 を使用して hello パケットを送信します。HSRP バージョン 1 では、このマルチキャスト アドレスが 224.0.0.2 です。
- MAC アドレス範囲 0000.0C9F.F000 ～ 0000.0C9F.FFFF を使用します。HSRP バージョン 1 で使用する MAC アドレス範囲は、0000.0C07.AC00 ～ 0000.0C07.ACFF です。
- MD 5 認証のサポートが追加されました。

HSRP のバージョンを変更すると、Cisco NX-OS がグループを再初期化します。新しい仮想 MAC アドレスがグループに与えられるからです。

HSRP バージョン 2 では HSRP バージョン 1 とは異なるパケット フォーマットを使用します。パケット フォーマットは Type-Length-Value（TLV）です。HSRP バージョン 1 ルータは、HSRP バージョン 2 パケットを受信しても無視します。

HSRP 認証

HSRP のメッセージ ダイジェスト 5（MD5）アルゴリズム認証は、HSRP スプーフィング ソフトウェアから保護し、業界標準の MD5 アルゴリズムを使用して信頼性とセキュリティを向上させています。HSRP は IPv4 アドレスを認証 TLV に含めます。

HSRP メッセージ

HSRP が設定されたルータは、次の 3 種類のマルチキャスト メッセージを交換できます。

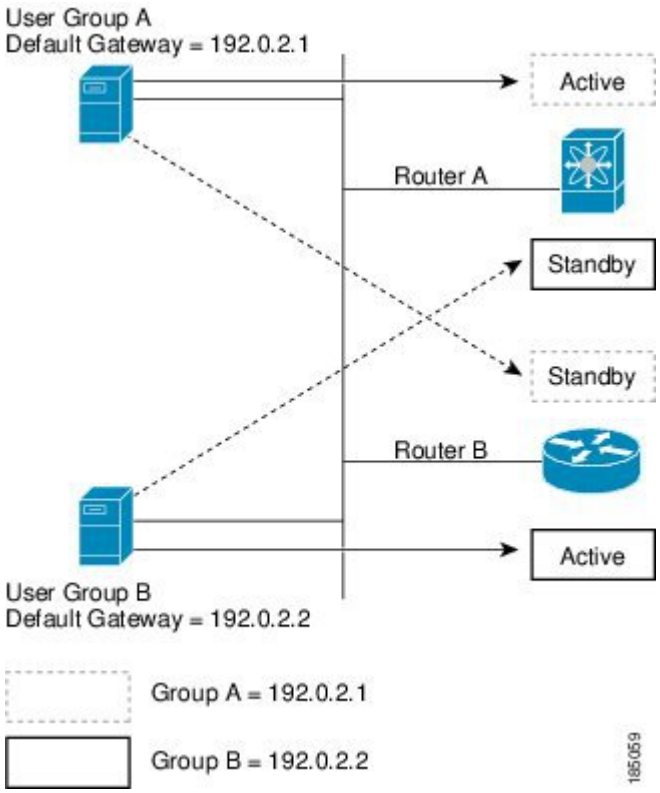
- hello：hello メッセージは、ルータの HSRP プライオリティおよびステート情報を他の HSRP ルータに伝えます。
- coup：スタンバイ ルータがアクティブ ルータの機能を引き受けるときに、coup メッセージを送信します。

• resign：アクティブ ルータは、アクティブ ルータとして機能する必要がなくなったときに、このメッセージを送信します。

HSRP ロード シェアリング

HSRP では、1 つのインターフェイスに複数のグループを設定できます。オーバーラップする 2 つの IPv4 HSRP グループを設定すると、期待されるデフォルト ルータの冗長性を HSRP から提供しながら、接続ホストからのトラフィックのロードシェアリングが可能です。下の図に、ロード シェアリングが行われる HSRP IPv4 構成の例を示します。

図 *22 : HSRP* ロード シェアリング



HSRP ロード シェアリングの図には、2 台のルータ（A および B）と 2 つの HSRP グループが示されています。ルータAはグループAのアクティブルータですが、グループBのスタンバイルータです。同様に、ルータBはグループBのアクティブルータであり、グループAのスタンバイルータです。両方のルータがアクティブのままの場合、HSRPは両方のルータにまたがるホスト。どちらかのルータで障害が発生すると、残りのルータが引き続き、両方のホストのトラフィックを処理します。

オブジェクト トラッキングおよび HSRP

オブジェクト トラッキングを使用すると、別のインターフェイスの動作状態に基づいて、HSRP インターフェイスのプライオリティを変更できます。オブジェクト トラッキングによって、メ

イン ネットワークへのインターフェイスで障害が発生した場合に、スタンバイ ルータにルーティングできます。

トラッキング可能なオブジェクトは、インターフェイスのライン プロトコル ステートまたはIPルートの到達可能性の2種類です。指定したオブジェクトがダウンすると、設定された値だけ Cisco NX-OS が HSRP プライオリティを引き下げます。詳細については、「HSRP オブジェクト トラッキングの設定」の項を参照してください。

HSRP の前提条件

HSRP の前提条件は、次のとおりです。

- HSRP グループを設定してイネーブルにするには、その前にHSRP機能をスイッチでイネーブルにする必要があります。

HSRP の注意事項と制約事項

HSRP 設定時の注意事項および制約事項は、次のとおりです。

- 最小 hello タイマー値は 250 ミリ秒です。
- 最小ホールド タイマー値は 750 ミリ秒です。
- HSRP を設定するインターフェイスに IP アドレスを設定し、そのインターフェイスをイネーブルにしてからでなければ、HSRP はアクティブになりません。
- IPv4 では、仮想 IP アドレスは、インターフェイス IP アドレスと同じサブネットになければなりません。
- 同一インターフェイス上では、複数のファーストホップ冗長プロトコルを設定しないことを推奨します。
- HSRP バージョン 2 は HSRP バージョン 1 と相互運用できません。どちらのバージョンも相互に排他的なので、インターフェイスはバージョン 1 およびバージョン 2 の両方を運用できません。しかし、同一ルータの異なる物理インターフェイス上であれば、異なるバージョンを実行できます。
- バージョン 1 で認められるグループ番号範囲（0 ～ 255）を超えるグループを設定している場合は、バージョン 2 からバージョン 1 への変更はできません。
- Cisco NX-OS では、VDC、インターフェイス VRF メンバーシップ、ポート チャネル メンバーシップを変更したり、ポート モードをレイヤ2に変更した場合は、インターフェイス上のすべてのレイヤ 3 設定が削除されます。
- PACL フィルタリングによる HSRP ローカリゼーションおよび HSRP の 4 ウェイ設定はサポートされていません。

HSRP のデフォルト設定

次の表に、HSRP パラメータのデフォルト設定値を示します。

表 *19* : デフォルトの *HSRP* パラメータ

パラメータ	デフォルト
HSRP	ディセーブル
認証	バージョン 1 の場合はテキストとしてイネーブル、パスワードは cisco
HSRP バージョン	バージョン 1
プリエンプション	無効
プライオリティ	100
仮想 MAC アドレス	HSRP グループ番号から生成

『Configuring HSRP』

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

HSRP 機能のイネーブル化

HSRP グループを設定してイネーブルにするには、その前に HSRP 機能をグローバルでイネーブルにする必要があります。

手順の詳細

HSRP機能をイネーブルにするには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
feature hsrp 例： `switch(config)# feature hsrp`	HSRPをイネーブルにします。

HSRP 機能をディセーブルにして、関連付けられている設定をすべて削除するには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
no feature hsrp 例： `switch(config)# no feature hsrp`	HSRP をディセーブルにします。

HSRP バージョン設定

HSRP のバージョンを設定できます。既存グループのバージョンを変更すると、仮想 MAC アドレスが変更されるので、Cisco NX-OS がそれらのグループの HSRP を再初期化します。HSRP のバージョンは、インターフェイス上のすべてのグループに適用されます。

HSRP のバージョンを設定するには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
hsrp version { **1** | **2** } 例： `switch(config-if)# hsrp version 2`	HSRP バージョンを設定します。デフォルトはバージョン 1 です。

IPv4 の HSRP グループの設定

IPv4 インターフェイスに HSRP グループを設定し、その HSRP グループに仮想 IP アドレスと仮想 MAC アドレスを設定できます。

始める前に

HSRP 機能が有効になっていることを確認します（HSRP 機能のイネーブル化のセクションを参照）。

グループのいずれかのメンバインターフェイス上で仮想 IP アドレスを設定すると、Cisco NX-OS によって HSRP がイネーブルになります。HSRP グループをイネーブルにする前に、認証、タイマー、プライオリティなどの HSRP 属性を設定する必要があります。

手順の概要

1. **configure terminal**
2. **interface** *type number*
3. **no switchport**
4. **ip address** *ip-address/length*
5. **hsrp** *group-number* [**ipv4**]

6. **ip** [*ip-address* [**secondary**]]
7. **exit**
8. **no shutdown**
9. （任意） **show hsrp** [**group** *group-number*] [**ipv4**]
10. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface** *type number*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス構成モードを開始します。

ステップ 3	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。
ステップ 4	**ip address** *ip-address/length* 例： `switch(config-if)# ip address 192.0.2.2/8`	インターフェイスの IPv4 アドレスを設定します。

ステップ 5	**hsrp** *group-number* [**ipv4**]
	例：
	`switch(config-if)# hsrp 2`
	`switch(config-if-hsrp)`
		HSRP グループを作成し、HSRP コンフィギュレーション モードを開始します。HSRP バージョン 1 で指定できる範囲は 0 ～ 255 です。HSRP バージョン 2 で指定できる範囲は 0 ～ 4095 です。デフォルト値は 0 です

ステップ 6	**ip** [*ip-address* [**secondary**]] 例： `switch(config-if-hsrp)# ip 192.0.2.1`	HSRP グループの仮想 IP アドレスを設定し、グループを有効にします。このアドレスは、インターフェイスの IPv4 アドレスと同じサブネットになければなりません。
ステップ 7	**exit** 例： `switch(config-if-hsrp)# exit`	HSRP設定モードを終了します。
ステップ 8	**no shutdown** 例： `switch(config-if)# no shutdown`	インターフェイス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 9	（任意） **show hsrp** [**group** *group-number*] [**ipv4**] 例： `switch(config-if)# show hsrp group 2`	HSRP 情報を表示します。
ステップ 10	（任意） **copy running-config startup-config** 例： `switch(config-if)# copy running-config startup-config`	この設定変更を保存します。

例

（注） 設定完了後にインターフェイスを有効にするには、**no shutdown** コマンドを使用する必要があります。

次に Ethernet 1/2 上で HSRP グループを設定する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# ip 192.0.2.2/8
switch(config-if)# hsrp 2
switch(config-if-hsrp)# ip 192.0.2.1
switch(config-if-hsrp)# exit
switch(config-if)# no shutdown
switch(config-if)# copy running-config startup-config
```

HSRP 仮想 MAC アドレスの設定

設定されているグループ番号から HSRP が導き出したデフォルトの仮想 MAC アドレスを変更できます。

HSRP グループの仮想 MAC アドレスを手動で設定するには、HSRP コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
mac-address *string* 例： `switch(config-if-hsrp)# mac-address 5000.1000.1060`	HSRP グループの仮想 MAC アドレスを設定します。ストリングには標準の MAC アドレス フォーマット（xxxx.xxxx.xxxx）を使用します。

仮想 MAC アドレスに BIA（バーンドイン MAC アドレス）を使用するように HSRP を設定するには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
hsrp use-bia [**scope interface**] 例： `switch(config-if)# hsrp use-bia`	HSRP 仮想 MAC アドレスにインターフェイスの BIA を使用するように、HSRP を設定します。**scope interface** キーワードを使用すると、このインターフェイス上のすべてのグループに焼き込み MAC アドレスを使用するように HSRP を設定できます。

HSRP の認証

クリアテキストまたはMD5ダイジェスト認証を使用してプロトコルを認証するように、HSRPを設定できます。MD5 認証はキー チェーンを使用します（Cisco Nexus 3548 スイッチ NX-OS セキュリティ構成ガイドを参照)。

始める前に

HSRP 機能が有効になっていることを確認します（HSRP 機能のイネーブル化のセクションを参照）。

HSRP グループのすべてのメンバに同じ認証およびキーを設定する必要があります。

MD5 認証を使用する場合は、キーチェーンが作成してあることを確認します。

手順の概要

1. **configure terminal**
2. **interface** *interface type slot/port*
3. **no switchport**
4. **hsrp** *group-number* [**ipv4**]
5.
6. （任意） **show hsrp** [**group** *group-number*]
7. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface** *interface type slot/port*
	例：
	`switch(config)# interface ethernet 1/2`
	`switch(config-if)#`
		インターフェイス構成モードを開始します。

<table>
<tr><th></th><th>コマンドまたはアクション</th><th>目的</th></tr>
<tr><td>ステップ 3</td><td>no switchport
例：
<pre>switch(config-if)# no switchport</pre></td><td>そのインターフェイスを、レイヤ3ルーテッドインターフェイスとして設定します。</td></tr>
<tr><td>ステップ 4</td><td>hsrp group-number [ipv4]
例：
<pre>switch(config-if)# hsrp 2
switch(config-if-hsrp)</pre></td><td>HSRP グループを作成し、HSRP設定モードを開始します。</td></tr>
<tr><td>ステップ 5</td><td><table>
<tr><th>オプション</th><th>説明</th></tr>
<tr><td>コマンド</td><td>目的</td></tr>
<tr><td>authentication text string
例：
<pre>switch(config-if-hsrp)#
 authentication text
mypassword</pre></td><td>このインターフェイス上で、HSRPのクリアテキスト認証を設定します。</td></tr>
<tr><td>authentication md5 { key-chain key-chain | key-string { 0 | 7 } text [timeout seconds]}
例：
<pre>switch(config-if-hsrp)#
 authentication md5
key-chain hsrp-keys</pre></td><td>このインターフェイス上で、HSRP の MD5 認証を設定します。キーチェーンまたはキー ストリングを使用できます。キー ストリングを使用する場合は、HSRPが新しいキーだけを受け付けるように、任意でタイムアウトを設定できます。指定できる範囲は 0 ～ 32767 秒です。</td></tr>
</table></td><td></td></tr>
<tr><td>ステップ 6</td><td>（任意） show hsrp [group group-number]
例：
<pre>switch(config-if-hsrp)# show hsrp group 2</pre></td><td>HSRP 情報を表示します。</td></tr>
<tr><td>ステップ 7</td><td>（任意） copy running-config startup-config
例：
<pre>switch(config-if-hsrp)# copy running-config
startup-config</pre></td><td>この設定変更を保存します。</td></tr>
</table>

例

次に、キーチェーン作成後に HSRP の MD5 認証を Ethernet 1/2 上で設定する例を示します。

```
switch# configure terminal
switch(config)# key chain hsrp-keys
switch(config-keychain)# key 0
switch(config-keychain-key)# key-string 7 zqdest
switch(config-keychain-key) accept-lifetime 00:00:00 Jun 01 2008 23:59:59 Sep 12 2008
switch(config-keychain-key) send-lifetime 00:00:00 Jun 01 2008 23:59:59 Aug 12 2008
switch(config-keychain-key) key 1
switch(config-keychain-key) key-string 7 uaeqdyito
switch(config-keychain-key) accept-lifetime 00:00:00 Aug 12 2008 23:59:59 Dec 12 2008
switch(config-keychain-key) send-lifetime 00:00:00 Sep 12 2008 23:59:59 Nov 12 2008
switch(config-keychain-key)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# hsrp 2
switch(config-if-hsrp)# authenticate md5 key-chain hsrp-keys
switch(config-if-hsrp)# copy running-config startup-config
```

HSRP オブジェクト トラッキングの設定

他のインターフェイスまたはルータの可用性に基づいて、プライオリティが調整されるように HSRP グループを設定できます。スイッチがオブジェクト トラッキング対応として設定されていて、なおかつトラッキング対象のオブジェクトがダウンした場合、スイッチのプライオリティはダイナミックに変更されます。トラッキング プロセスはトラッキング対象オブジェクトに定期的にポーリングを実行し、値の変化をすべて記録します。値が変化すると、HSRP がプライオリティを再計算します。HSRP インターフェイスにプリエンプションを設定している場合は、プライオリティの高い HSRP インターフェイスがアクティブ ルータになります。

HSRP では、トラッキング対象のオブジェクトおよびトラック リストをサポートします。トラック リストの詳細については、オブジェクト トラッキングの設定を参照してください。

始める前に

HSRP 機能が有効になっていることを確認します（HSRP 機能のイネーブル化のセクションを参照）。

手順の概要

1. **configure terminal**
2.
3. **interface** *interface-type slot/port*
4. no switchport
5. **hsrp** *group-number* [**ipv4**]
6. **priority** [*value*]
7. **track** *object-number* [**decrement** *value*]
8. **preempt** [**delay** [**minimum** *seconds*] [**reload** *seconds*] [**sync** *seconds*]]
9. （任意） **show hsrp interface** *interface-type number*

10. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2		

オプション	説明
コマンド	目的

track *object-id* **interface** *interface-type number* { **ip routing** | **line-protocol** }
例：
`switch(config)# track 1 interface ethernet 2/2 line-protocol`
`switch(config-track#`
	この HSRP インターフェイスが追跡するインターフェイスを設定します。インターフェイスのステート変化は次のように、このHSRPのプライオリティを左右します。
	• HSRP コンフィギュレーション モードで、**track** コマンドで使用するインターフェイスおよび対応するオブジェクト番号を設定します。
	• **line-protocol** キーワードを指定すると、インターフェイスがアップ状態かどうかが追跡されます。**ip** キーワードを指定すると、インターフェイス上で IP ルーティングがイネーブルであり、IP アドレスが設定されているかどうかもチェックされます。

track *object-id* **ip route** *ip-prefix/length* **reachability** 例：	ルートのトラッキング対象オブジェクトを作成し、トラッキング コンフィギュレーション モードを開始し

	コマンドまたはアクション	目的
	オプション / 説明: `switch(config)# track 2 ip route 192.0.2.0/8 reachability switch(config-track#` / ます。**object-id** の範囲は 1 ～ 500 です。	
ステップ 3	**interface** *interface-type slot/port* 例： `switch(config)# interface ethernet 1/2 switch(config-if)#`	インターフェイス設定モードを開始します。
ステップ 4	no switchport 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。
ステップ 5	**hsrp** *group-number* [**ipv4**] 例： `switch(config-if)# hsrp 2 switch(config-if-hsrp)#`	HSRP グループを作成し、HSRP 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	**priority** [*value*] 例： `switch(config-if-hsrp)# priority 254`	HSRP グループでのアクティブ ルータ選択に使用するプライオリティ レベルを設定します。有効な範囲は 0 ～ 255 です。デフォルトは 100 です。
ステップ 7	**track** *object-number* [**decrement** *value*] 例： `switch(config-if-hsrp)# track 1 decrement 20`	HSRP インターフェイスの重み付けを左右する、トラッキング対象のオブジェクトを指定します。 *value* 引数には、トラッキング対象のオブジェクトで障害が発生した場合に、HSRP インターフェイスのプライオリティから差し引く値を指定します。範囲は 1 ～ 255 です。デフォルトは 10 です。
ステップ 8	**preempt** [**delay** [**minimum** *seconds*] [**reload** *seconds*] [**sync** *seconds*]] 例： `switch(config-if-hsrp)# preempt delay minimum 60`	現在のアクティブ ルータよりプライオリティが高い場合に、HSRP グループのアクティブ ルータとして引き継ぐようにルータを設定します。このコマンドは、デフォルトでディセーブルになっています。指定できる範囲は 0 ～ 3600 秒です。
ステップ 9	（任意） **show hsrp interface** *interface-type number* 例： `switch(config-if-hsrp)# show hsrp interface ethernet 1/2`	インターフェイスの HSRP 情報を表示します。
ステップ 10	（任意） **copy running-config startup-config** 例：	この設定変更を保存します。

	コマンドまたはアクション	目的
	`switch(config-if)# copy running-config startup-config`	

例

次に、Ethernet 1/2 上で HSRP オブジェクト トラッキングを設定する例を示します。

```
switch# configure terminal
switch(config)# track 1 interface ethernet 2/2 line-protocol
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# hsrp 2
switch(config-if-hsrp)# track 1 decrement 20
switch(config-if-hsrp)# copy running-config startup-config
```

HSRP プライオリティの設定

インターフェイス上で HSRP プライオリティを設定できます。HSRP では、プライオリティを使用して、アクティブ ルータとして動作する HSRP グループ メンバを決定します。

HSRPプライオリティを設定するには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
priority *level* [**forwarding-threshold lower** *lower-value* **upper** *upper-value*] 例： `switch(config-if-hsrp)# priority 60 forwarding-threshold lower 40 upper 50`	HSRP グループでのアクティブ ルータ選択に使用するプライオリティ レベルを設定します。*level* の範囲は 0 ～ 255 です。デフォルトは 100 です。

HSRP のカスタマイズ

必要に応じて、HSRPの動作をカスタマイズできます。仮想IPアドレスを設定することによって、HSRP グループをイネーブルにすると、そのグループがただちに動作可能になることに注意してください。HSRP をカスタマイズする前に HSRP 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、ルータがグループの制御を引き継いでアクティブ ルータになる可能性があります。HSRP のカスタマイズを予定している場合は、HSRP グループをイネーブルにする前に行ってください。

HSRP をカスタマイズするには、HSRP コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
name *string* 例： `switch(config-if-hsrp)# name HSRP-1`	HSRP グループの IP 冗長名を指定します。*string* は 1 ～ 255 文字です。デフォルト ストリングのフォーマットは **hsrp-interface-short-name**-*group-id* です。 次の例を参考にしてください。 `hsrp-Eth2/1-1`
preempt [**delay** [**minimum** *seconds*] [**reload** *seconds*] [sync *seconds*]] のようになります。 例： `switch(config-if-hsrp)# preempt delay minimum 60`	現在のアクティブ ルータよりもプライオリティが高い場合に、HSRP グループのアクティブ ルータとして引き継ぐようにルータを設定します。このコマンドは、デフォルトでディセーブルになっています。指定できる範囲は 0 ～ 3600 秒です。
timers [**msec**] *hellotime* [**msec**] *holdtime* 例： `switch(config-if-hsrp)# timers 5 18`	次のように、この HSRP メンバーの hello タイムおよびホールド タイムを設定します。 オプションの *msec* キーワードは、引数がデフォルトの秒単位ではなく、ミリ秒単位で表されることを指定します。タイマーの範囲（ミリ秒）は次のとおりです。 • *hellotime*：hello パケットを送信してから、次の hello パケットを送信するまでのインターバル。指定できる範囲は 255 ～ 999 ミリ秒です。 • *holdtime*：hello パケットの情報が無効と見なされるまでのインターバル。指定できる範囲は 750 ～ 3000 ミリ秒です。

HSRP をカスタマイズするには、インターフェイス コンフィギュレーション モードで次のコマンドを使用します。

コマンドまたはアクション	目的
hsrp delay minimum *seconds* 例： `switch(config-if)# hsrp delay minimum 30`	グループがイネーブルになってから、グループに参加するまでに HSRP が待機する最小時間を指定します。指定できる範囲は 0 ～ 10000 秒です。デフォルトは 0 です。
hsrp delay reload *seconds* 例： `switch(config-if)# hsrp delay reload 30`	リロード後、グループに参加するまでに HSRP が待機する最小時間を指定します。指定できる範囲は 0 ～ 10000 秒です。デフォルトは 0 です。

HSRP 設定の確認

HSRP の設定情報を表示するには、次のいずれかの作業を行います。

コマンド	目的
show hsrp [**group** *group-number*]	すべてのグループまたは特定のグループの HSRP ステータスを表示します。
show hsrp delay [**interface** *interface-type slot/port*]	すべてのインターフェイスまたは特定のインターフェイスの HSRP 遅延値を表示します。
show hsrp [**interface** *interface-type slot/port*]	インターフェイスの HSRP ステータスを表示します。
show hsrp [**group** *group-number*] [**interface** *interface-type slot/port*] [**active**] [**all**] [**init**] [**learn**] [**listen**] [**speak**] [**standby**]	ステートが active、init、listen、または standby の仮想フォワーダについて、グループまたはインターフェイスの HSRP ステータスを表示します。disabled を含めてすべてのステートを表示する場合は、**all** キーワードを使用します。
show hsrp [**group** *group-number*] [**interface** *interface-type slot/port*] **active**] [**all**] [**init**] [**learn**] [**listen**] [**speak**] [**standby**] **brief**	ステートが active、init、listen、または standby の仮想フォワーダについて、グループまたはインターフェイスの HSRP ステータスの要約を表示します。disabled を含めてすべてのステートを表示する場合は、**all** キーワードを使用します。

HSRP の設定例

次に、MD5 認証およびインターフェイス トラッキングを指定して、インターフェイス上で HSRP をイネーブルにする例を示します。

```
key chain hsrp-keys
key 0
key-string 7 zqdest
accept-lifetime 00:00:00 Jun 01 2008 23:59:59 Sep 12 2008
send-lifetime 00:00:00 Jun 01 2008 23:59:59 Aug 12 2008
key 1
key-string 7 uaeqdyito
accept-lifetime 00:00:00 Aug 12 2008 23:59:59 Dec 12 2008
send-lifetime 00:00:00 Sep 12 2008 23:59:59 Nov 12 2008
feature hsrp
track 2 interface ethernet 2/2 ip
interface ethernet 1/2
no switchport
ip address 192.0.2.2/8
hsrp 1
authenticate md5 key-chain hsrp-keys
priority 90
track 2 decrement 20
```

```
ip-address 192.0.2.10
no shutdown
```

その他の参考資料

HSRP の実装に関する詳細は、次の各項を参照してください。

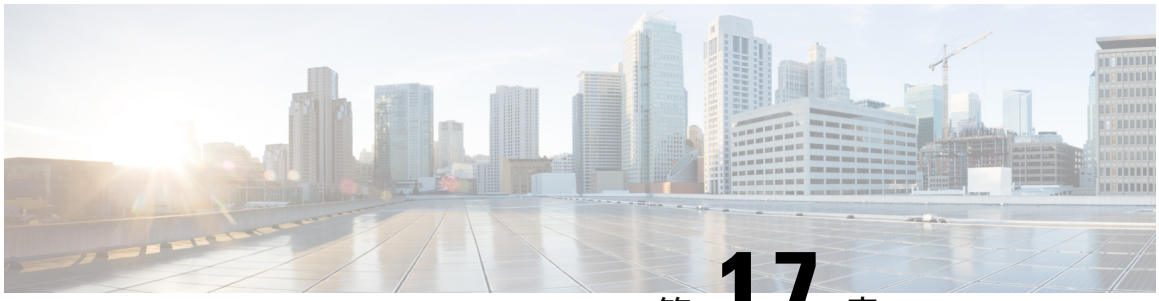
- 関連資料
- MIB

関連資料

関連項目	マニュアル タイトル
VRRP の設定	VRRP の設定
HSRP CLI コマンド	『Cisco Nexus 3000 Series Command Reference』

MIB

MIB	MIB のリンク
CISCO-HSRP-MIB	MIB を検索してダウンロードするには、次の MIB ロケータに移動します。



第 17 章

VRRP の設定

この章では、Cisco NX-OS スイッチ上で仮想ルータ冗長プロトコル（VRRP）を設定する方法について説明します。

この章は、次の項で構成されています。

VRRP の概要

VRRP を使用すると、仮想 IP アドレスを共有するルータ グループを設定することによって、ファーストホップ IP ルータで透過的フェールオーバーが可能になります。VRRP はそのグループのプライマリ ルータを選択して、仮想 IP アドレスへのすべてのパケットが処理できるようにします。残りのルータはスタンバイになり、プライマリ ルータで障害が発生した場合に処理を引き継ぎます。

VRRP の動作

LAN クライアントは、ダイナミック プロセスまたはスタティック設定を使用することによって、特定のリモート宛先へのファーストホップにするルータを決定できます。ダイナミック ルータ ディスカバリの例を示します。

- 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。
- ICMP Router Discovery Protocol（IRDP）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。

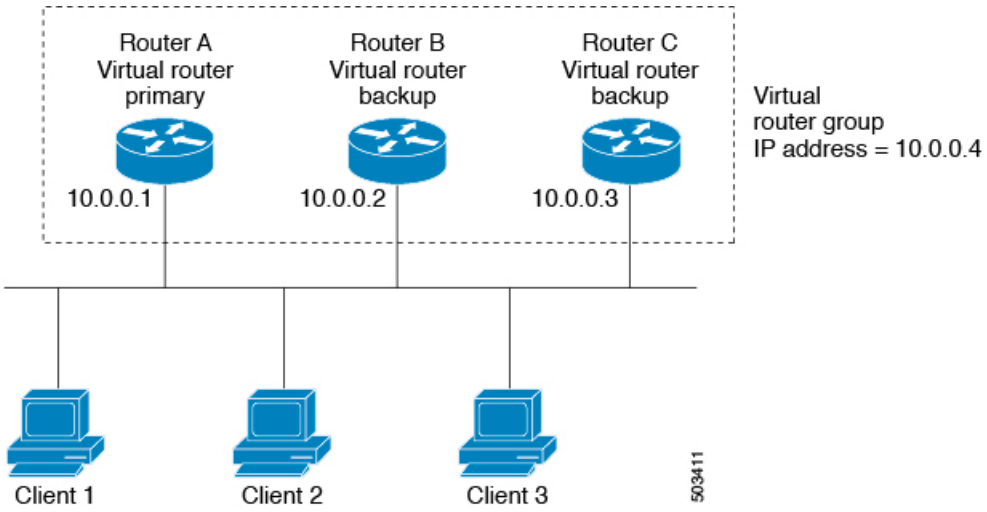
ダイナミック ディスカバリ プロトコルのデメリットは、LAN クライアントにある程度、設定および処理のオーバーヘッドが発生することです。また、ルータが故障した場合、他のルータに切り替えるプロセスも遅くなる場合があります。

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および処理が簡素化されますが、シングルポイント障害が生じます。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。

VRRP では、ルータ グループ（VRRP グループ）が単一の仮想 IP アドレスを共有できるようにすることによって、スタティック設定に伴う問題を解決できます。さらに、デフォルト ゲートウェイとして仮想 IP アドレスを指定して、LAN クライアントを設定できます。

次の図は、基本的な VLANトポロジです。この例では、ルータ A、B、および C が VRRP グループを形成します。グループの IP アドレスは、ルータ A のインターフェイス インターフェイスに設定されているアドレス（10.0.0.1）と同じです。

図 *23*：基本的な *VRRP* トポロジ



仮想 IP アドレスにルータ A の物理イーサネット インターフェイスの IP アドレスが使用されるので、ルータ A がプライマリ（「IP アドレス オーナー」）になります。ルータ A はプライマリとして、VRRP グループ ルータの仮想 IP アドレスを所有し、送信されたパケットをこの IP アドレスに転送します。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 の役割はバックアップです。プライマリで障害が発生すると、プライオリティが最も高いバックアップルータがプライマリになり、仮想IPアドレスを引き継いで、LANホストへのサービスが途切れないようにします。ルータ A が回復すると、これが再びプライマリ ルータになります。詳細については、「VRRP ルータのプライオリティおよびプリエンプション」のセクションを参照してください。

（注） ルーテッド ポートで受信した VRRP 仮想 IP アドレス宛のパケットは、ローカル ルータ上で終端します。そのルータがプライマリ VRRP ルータであるのかバックアップ VRRP ルータであるのかは関係ありません。これには ping トラフィックと Telnet トラフィックが含まれます。レイヤ 2（VLAN）インターフェイスで受信した、VRRP 仮想 IP アドレス宛のパケットは、プライマリ ルータに届きます。

VRRP の利点

VRRP の利点は、次のとおりです。

- 冗長性：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ので、ネットワークにシングル ポイント障害が発生する確率が下がります。
- ロード シェアリング：複数のルータで LAN クライアントとの間のトラフィックを分担できます。トラフィックの負荷が使用可能なルータ間でより公平に分担されます。
- マルチ VRRP グループ：プラットフォームがマルチ MAC アドレスをサポートする場合、ルータの物理インターフェイス上で、最大 255 の VRRP グループをサポートします。マルチ VRRP グループによって、LAN トポロジで冗長性およびロード シェアリングを実現できます。
- マルチ IP アドレス：セカンダリ IP アドレスを含めて、複数の IP アドレスを管理できます。イーサネット インターフェイス上で複数のサブネットを設定している場合は、各サブネットで VRRP を設定できます。
- プリエンプト：障害プライマリを引き継いでいたバックアップ ルータより、さらにプライオリティが高いバックアップ ルータが使用可能になったときに、プライオリティが高い方を優先させることができます。
- アドバタイズメント プロトコル：VRRP アドバタイズメントに、専用の Internet Assigned Numbers Authority（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 は VRRP に IP プロトコル番号 112 を割り当てています。
- VRRP トラッキング：インターフェイスのステートに基づいて VRRP プライオリティを変更することによって、最適な VRRP ルータがグループのプライマリになることが保証されます。

複数の VRRP グループ

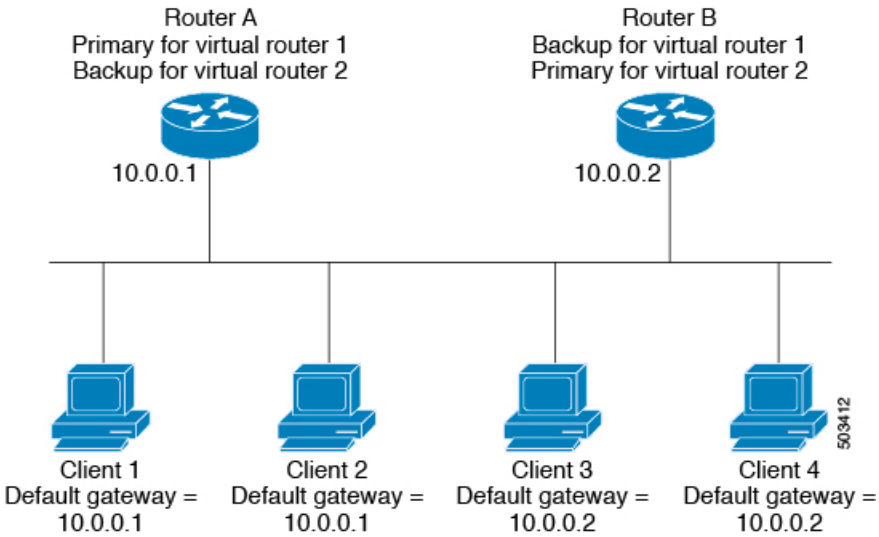
物理インターフェイス上で、最大255のVRRPグループを設定できます。ルータインターフェイスがサポートできるVRRPグループの実際の数は、次の要因によって決まります。

- ルータの処理能力
- ルータのメモリの能力

ルータインターフェイス上で複数のVRRPグループが設定されたトポロジでは、インターフェイスはあるVRRPグループのプライマリ、および他の1つまたは複数のVRRPグループのバックアップとして動作可能です。

次の図のLANトポロジでは、ルータAとBがクライアント1～4のトラフィックを共有するように、VRRPが設定されています。ルータAとBの一方で障害が発生した場合、もう一方がバックアップとして機能します。

図 *24 :* ロード シェアリングおよび冗長構成の *VRRP* トポロジ



このトポロジには、オーバーラップする2つのVRRPグループに対応する2つの仮想IPアドレスが含まれています。VRRPグループ1では、ルータAがIPアドレス10.0.0.1のオーナーであり、プライマリです。ルータBはルータAのバックアップです。クライアント1と2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。

VRRPグループ2では、ルータBがIPアドレス10.0.0.2のオーナーであり、プライマリです。ルータAはルータBをバックアップします。クライアント3と4には、デフォルト ゲートウェイのIPアドレス10.0.0.2が設定されています。

VRRP ルータのプライオリティおよびプリエンプション

VRRP 冗長構成の重要な側面は、VRRP ルータのプライオリティです。各 VRRP ルータが果たす役割やプライマリ ルータで障害が発生した場合のアクションは、プライオリティによって決まるからです。

VRRP ルータが仮想 IP アドレスおよび物理インターフェイスの IP アドレスを所有する場合、そのルータはプライマリとして機能します。プライマリのプライオリティは 255 です。

プライオリティによって、VRRP ルータがバックアップ ルータとして動作するかどうかが決まり、さらに、プライマリで障害が発生した場合にプライマリになる順序も決まります。

たとえば、ルータ A が LAN トポロジにおけるプライマリであり、そのルータ A で障害が発生した場合、VRRP はバックアップ B が引き継ぐのか、バックアップ C が引き継ぐのかを判断する必要があります。ルータ B にプライオリティ 101 が設定されていて、ルータ C がデフォルトのプライオリティ 100 の場合、VRRP はルータ B をプライマリになるべきルータとして選択します。ルータ B の方がプライオリティが高いからです。ルータ B および C にデフォルトのプライオリティ 100 が設定されている場合は、VRRP は IP アドレスが大きい方のバックアップをプライマリになるべきルータとして選択します。

VRRP ではプリエンプションを使用して、VRRP バックアップ ルータがプライマリになってからのアクションを決定します。プリエンプションはデフォルトでイネーブルなので、VRRP は新しいプライマリよりプライオリティの高いバックアップがオンラインになると、バックアップに切り替えます。たとえば、ルータ A がプライマリであり、そのルータ A で障害が発生した場合、VRRP は（プライオリティの順位が次である）ルータ B を選択します。ルータ C がルータ B より高いプライオリティでオンラインになると、ルータ B で障害が発生していなくても、VRRP はルータ C を新しいプライマリとして選択します。

プリエンプションを無効にした場合、VRRP が切り替わるのは、元のプライマリが回復した場合、または新しいプライマリで障害が発生した場合に限られます。

VRRP のアドバタイズメント

VRRP プライマリは、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、プライマリの優先順位と状態が伝達されます。Cisco NX-OS は VRRP アドバタイズメントを IP パケットにカプセル化して、VRRP グループに割り当てられた IP マルチキャスト アドレスに送信します。Cisco NX-OS がアドバタイズメントを送信する間隔はデフォルトでは 1 秒ですが、ユーザ側で別のアドバタイズインターバルを設定できます。

VRRP 認証

VRRP は、次の認証方式をサポートします。

- 認証なし
- プレーン テキスト認証

VRRP は次の場合に、パケットを拒否します。

- 認証方式がルータと着信パケットで異なる。
- テキスト認証文字列がルータと着信パケットで異なる。

VRRP トラッキング

VRRP は次の 2 つのトラッキング オプションをサポートしています。

- ネイティブ インターフェイス トラッキング：インターフェイスのステートを追跡し、そのステートを使用して VRRP グループの VRRP ルータのプライオリティを判別します。インターフェイスがダウンしている場合、またはインターフェイスにプライマリ IP アドレスがない場合、トラッキング対象ステートはダウンとなります。
- オブジェクト トラッキング：設定されたオブジェクトのステートを追跡し、そのステートを使用して VRRP グループの VRRP ルータのプライオリティを判別します。オブジェクト トラッキングの詳細については、「オブジェクト トラッキングの設定」を参照してください。

トラッキング対象ステート（インターフェイスまたはオブジェクト）がダウンになると、VRRP はユーザがトラッキング対象ステートに対して新しいプライオリティをどのように設定するかに基づいて、プライオリティをアップデートします。トラッキング対象ステートがオンラインになると、VRRP は仮想ルータ グループの元のプライオリティを復元します。

たとえば、ネットワークへのアップリンクがダウンした場合、別のグループ メンバーが VRRP グループのプライマリとして引き継げるように、VRRP グループ メンバーのプライオリティを引き下げなければならないことがあります。詳細については、「VRRP インターフェイス ステート トラッキングの設定」の項を参照してください。

（注） VRRP はレイヤ 2 インターフェイスのトラッキングをサポートしていません。

仮想化のサポート

VRRP は仮想ルーティングおよび転送（VRF）インスタンスをサポートします。デフォルトでは、特に別の VRF を設定しない限り、Cisco NX-OS はユーザーをデフォルトの VRF に配置します。

インターフェイスの VRF メンバーシップを変更すると、Cisco NX-OS によって VRRP を含め、すべてのレイヤ 3 設定が削除されます。

VRRP の注意事項と制約事項

VRRP には、次の注意事項および制限事項があります。

- 管理インターフェイス上で VRRP を設定できません。
- VRRP がイネーブルの場合は、ネットワーク上のスイッチ全体で VRRP 設定を複製する必要があります。
- 同一インターフェイス上では、複数のファーストホップ冗長プロトコルを設定しないことを推奨します。
- VRRP を設定するインターフェイスに IP アドレスを設定し、そのインターフェイスをイネーブルにしてからでなければ、VRRP はアクティブになりません。
- Cisco NX-OS では、VDC、インターフェイス VRF メンバーシップ、ポート チャネル メンバーシップを変更したり、ポート モードをレイヤ 2 に変更した場合は、インターフェイス上のすべてのレイヤ 3 設定が削除されます。
- VRRP でレイヤ 2 インターフェイスを追跡するよう設定した場合、レイヤ 2 をシャットダウンしてからインターフェイスを再度イネーブル化することにより、VRRP プライオリティを更新してレイヤ 2 インターフェイスのステートを反映させる必要があります。

VRRP のデフォルト設定

次の表に、VRRP パラメータのデフォルト設定値を示します。

表 *20 :* デフォルトの *VRRP* パラメータ

パラメータ	デフォルト
advertisement interval	1 秒
認証	認証なし
プリエンプション	有効
priority	100
VRRP 機能	無効

VRRP の設定

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

VRRP 機能のイネーブル化

VRRP グループを設定してイネーブルにするには、その前に VRRP 機能をグローバルでイネーブルにする必要があります。

VRRP 機能をイネーブルにするには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
feature vrrp 例： `switch(config)# feature vrrp`	VRRP をイネーブルにします。

VRRP機能をディセーブルにして、関連付けられている設定をすべて削除するには、グローバル コンフィギュレーション モードで次のコマンドを使用します。

コマンド	目的
no feature vrrp 例： `switch(config)# no feature vrrp`	VRRP機能をディセーブルにします。

VRRP グループの設定

VRRP グループを作成し、仮想 IP アドレスを割り当て、グループを有効にすることができます。

VRRP グループに設定できる仮想 IPv4 アドレスは 1 つです。プライマリ VRRP ルータはデフォルトで、仮想 IP アドレスを直接の宛先とするパケットをドロップします。これは、VRRP プライマリがパケットを転送するネクストホップ ルータとしてのみ想定されているからです。アプリケーションによっては、Cisco NX-OS が仮想ルータ IP 宛のパケットを受け付けるようにする必要があります。仮想 IP アドレスに secondary オプションを使用すると、ローカル ルータが VRRP マスターの場合、これらのパケットを受け付けるようになります。

VRRP グループを設定した場合は、そのグループをアクティブにするために、グループを明示的に有効にする必要があります。

始める前に

インターフェイスに IP アドレスを設定していることを確認します（IPv4 アドレス指定の設定 のセクションを参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface -type slot/port*

3. **no switchport**
4. **vrrp** *number*
5. **address** *ip-address* [**secondary**]
6. **no shutdown**
7. （任意） **show vrrp**
8. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface** *interface -type slot/port*
	例：
	`switch(config)#`
	`switch(config-if)# interface ethernet 2/1`
		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	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。

ステップ 4	**vrrp** *number*
	例：
	`switch(config-if)# vrrp 250`
	`switch(config-if-vrrp)#`
		仮想ルータ グループを作成します。範囲は 1 ～ 255 です。

ステップ 5	**address** *ip-address* [**secondary**] 例： `switch(config-if-vrrp)# address 192.0.2.8`	指定の VRRP グループに仮想 IPv4 アドレスを設定します。このアドレスは、インターフェイスの IPv4 アドレスと同じサブネットになければなりません。 **secondary** オプションは、VRRP ルータが仮想ルータの IP アドレスに送信されたパケットを受け付けて、アプリケーションに配信することをアプリケーションが要求する場合に限られます。

ステップ 6	**no shutdown**
	例：
	`switch(config-if-vrrp)# no shutdown`
	`switch(config-if-vrrp)#`
		VRRP グループを有効にします。デフォルトでは無効です。

ステップ 7	（任意） **show vrrp** 例： `switch(config-if-vrrp)# show vrrp`	VRRP 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 8	（任意） **copy running-config startup-config** 例： `switch(config-if-vrrp)# copy running-config startup-config`	この設定変更を保存します。

VRRP プライオリティの設定

仮想ルータの有効なプライオリティ範囲は 1 ～ 254 です（1 が最下位、254 が最上位のプライオリティ）。バックアップのデフォルトのプライオリティ値は 100 です。インターフェイス アドレスがプライマリ仮想 IP アドレスと同じスイッチ（プライマリ）の場合、デフォルト値は 255 です。

始める前に

VRRP 機能が有効になっていることを確認します（VRRP の設定のセクションを参照）。

インターフェイス上で IP アドレスを設定していることを確認します（IPv4 アドレス指定の設定のセクションを参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface -type slot/port*
3. **no switchport**
4. **vrrp** *number*
5. **shutdown**
6. **priority** *level* [**forwarding-threshold lower** *lower-value* **upper** *upper-value*]
7. **no shutdown**
8. （任意） **show vrrp**
9. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface** *interface -type slot/port*
	例：
	`switch(config)#`
	`switch(config-if)# interface ethernet 2/1`
		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	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッドインターフェイスとして設定します。

ステップ 4	**vrrp** *number*
	例：
	`switch(config-if)# vrrp 250`
	`switch(config-if-vrrp)#`
		仮想ルータ グループを作成します。範囲は 1 ～ 255 です。

ステップ 5	**shutdown**
	例：
	`switch(config-if-vrrp)# shutdown`
	`switch(config-if-vrrp)#`
		VRRP グループを無効にします。デフォルトでは無効です。

ステップ 6	**priority** *level* [**forwarding-threshold lower** *lower-value* **upper** *upper-value*] 例： `switch(config-if-vrrp)# priority 60 forwarding-threshold lower 40 upper 50`	VRRP グループでのアクティブ ルータ選択に使用するプライオリティ レベルを設定します。レベルの範囲は 1 ー 254 です。バックアップの場合、デフォルトは 100 です。インターフェイス IP アドレスが仮想 IP アドレスと等しいプライマリの場合は 255 です。

ステップ 7	**no shutdown**
	例：
	`switch(config-if-vrrp)# no shutdown`
	`switch(config-if-vrrp)#`
		VRRP グループを有効にします。デフォルトでは無効です。

ステップ 8	（任意） **show vrrp** 例： `switch(config-if-vrrp)# show vrrp`	VRRP 情報を表示します。
ステップ 9	（任意） **copy running-config startup-config** 例： `switch(config-if-vrrp)# copy running-config startup-config`	この設定変更を保存します。

VRRP 認証の設定

VRRP グループに単純なテキスト認証を設定できます。

始める前に

ネットワークのすべての VRRP スイッチで認証設定が同じであることを確認します。

VRRP 機能が有効になっていることを確認します（VRRP の設定のセクションを参照）。

インターフェイス上で IP アドレスを設定していることを確認します（IPv4 アドレス指定の設定のセクションを参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface -type slot/port*
3. **no switchport**
4. **vrrp** *number*
5. **shutdown**
6. **authentication text** *password*
7. **no shutdown**
8. （任意） **show vrrp**
9. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface** *interface -type slot/port*
	例：
	`switch(config)#`
	`switch(config-if)# interface ethernet 2/1`
		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	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。

ステップ 4	**vrrp** *number*
	例：
	`switch(config-if)# vrrp 250`
	`switch(config-if-vrrp)#`
		仮想ルータ グループを作成します。範囲は 1 ～ 255 です。

ステップ 5	**shutdown**
	例：
	`switch(config-if-vrrp)# shutdown`
	`switch(config-if-vrrp)#`
		VRRP グループを無効にします。デフォルトでは無効です。

ステップ 6	**authentication text** *password* 例： `switch(config-if-vrrp)# authentication text cisco123`	単純なテキスト認証オプションを指定し、キーネーム パスワードを指定します。キーネームの範囲は 1 ～ 255 文字です。16 文字以上を推奨します。テキスト パスワードは、英数字で最大 8 文字です。

	コマンドまたはアクション	目的

ステップ 7	**no shutdown**
	例：
	`switch(config-if-vrrp)# no shutdown`
	`switch(config-if-vrrp)#`
		VRRP グループを有効にします。デフォルトでは無効です。

ステップ 8	（任意） **show vrrp** 例： `switch(config-if-vrrp)# show vrrp`	VRRP 情報を表示します。
ステップ 9	（任意） **copy running-config startup-config** 例： `switch(config-if-vrrp)# copy running-config startup-config`	この設定変更を保存します。

例

アドバタイズメント パケットのタイム インターバルの設定

アドバタイズメント パケットのタイム インターバルを設定できます。

始める前に

VRRP 機能が有効になっていることを確認します（VRRP の設定のセクションを参照）。

インターフェイス上で IP アドレスを設定していることを確認します（IPv4 アドレス指定の設定のセクションを参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface -type slot/port*
3. **no switchport**
4. **vrrp** *number*
5. **shutdown**
6. **advertisement-interval** *seconds*
7. **no shutdown**
8. （任意） **show vrrp**
9. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface** *interface -type slot/port*
	例：
	`switch(config)#`
	`switch(config-if)# interface ethernet 2/1`
		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	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。

ステップ 4	**vrrp** *number*
	例：
	`switch(config-if)# vrrp 250`
	`switch(config-if-vrrp)#`
		仮想ルータ グループを作成します。範囲は 1 ～ 255 です。

ステップ 5	**shutdown**
	例：
	`switch(config-if-vrrp)# shutdown`
	`switch(config-if-vrrp)#`
		VRRP グループを無効にします。デフォルトでは無効です。

ステップ 6	**advertisement-interval** *seconds* 例： `switch(config-if-vrrp)# advertisement-interval 15`	アドバタイズメント フレームの送信間隔を秒数で設定します。有効な範囲は 1 ～ 254 です。デフォルト値は 1 秒です。

ステップ 7	**no shutdown**
	例：
	`switch(config-if-vrrp)# no shutdown`
	`switch(config-if-vrrp)#`
		VRRP グループを有効にします。デフォルトでは無効です。

ステップ 8	（任意） **show vrrp** 例： `switch(config-if-vrrp)# show vrrp`	VRRP 情報を表示します。
ステップ 9	（任意） **copy running-config startup-config** 例： `switch(config-if-vrrp)# copy running-config startup-config`	この設定変更を保存します。

例

プリエンプションのディセーブル化

VRRP グループ メンバーのプリエンプションをディセーブルにできます。プリエンプションをディセーブルにした場合は、プライオリティのより高いバックアップ ルータが、プライオリティのより低いプライマリ ルータを引き継ぐことはありません。プリエンプションはデフォルトで有効です。

始める前に

VRRP 機能が有効になっていることを確認します（VRRP の設定のセクションを参照）。

インターフェイス上で IP アドレスを設定していることを確認します（IPv4 アドレス指定の設定のセクションを参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface -type slot/port*
3. **no switchport**
4. **vrrp** *number*
5. **shutdown**
6. **no preempt**
7. **no shutdown**
8. （任意） **show vrrp**
9. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface** *interface -type slot/port*
	例：
	`switch(config)#`
	`switch(config-if)# interface ethernet 2/1`
		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	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。

	コマンドまたはアクション	目的

ステップ 4	**vrrp** *number*
	例：
	`switch(config-if)# vrrp 250`
	`switch(config-if-vrrp)#`
		仮想ルータ グループを作成します。範囲は 1 ～ 255 です。

ステップ 5	**shutdown**
	例：
	`switch(config-if-vrrp)# shutdown`
	`switch(config-if-vrrp)#`
		VRRP グループを無効にします。デフォルトでは無効です。

ステップ 6	**no preempt** 例： `switch(config-if-vrrp)# no preempt`	preempt オプションをディセーブルにして、プライオリティが上位のバックアップが使用されてもプライマリが変わらないようにします。

ステップ 7	**no shutdown**
	例：
	`switch(config-if-vrrp)# no shutdown`
	`switch(config-if-vrrp)#`
		VRRP グループを有効にします。デフォルトでは無効です。

ステップ 8	（任意） **show vrrp** 例： `switch(config-if-vrrp)# show vrrp`	VRRP 情報を表示します。
ステップ 9	（任意） **copy running-config startup-config** 例： `switch(config-if-vrrp)# copy running-config startup-config`	この設定変更を保存します。

VRRP インターフェイス ステート トラッキングの設定

インターフェイスのステート追跡機能では、スイッチ内の他のインターフェイスのステートに基づいて、仮想ルータのプライオリティが変更されます。トラッキング対象のインターフェイスがダウンしたり、IP アドレスが削除されると、Cisco NX-OS はトラッキングのプライオリティ値を仮想ルータに割り当てます。トラッキング対象のインターフェイスがオンライン状態になり、IP アドレスがこのインターフェイスに設定されると、Cisco NX-OS は仮想ルータに設定されていたプライオリティを復元します（VRRP プライオリティの設定を参照）。

（注） インターフェイス ステート トラッキングを動作させるには、インターフェイス上でプリエンプションをイネーブルにする必要があります。

（注） VRRP はレイヤ 2 インターフェイスのトラッキングをサポートしていません。

始める前に

VRRP 機能が有効になっていることを確認します（VRRP の設定のセクションを参照）。

インターフェイス上で IP アドレスを設定していることを確認します（IPv4 アドレス指定の設定のセクションを参照）。

仮想ルータが有効になっていることを確認します（VRRP グループの設定のセクションを参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface -type slot/port*
3. **no switchport**
4. **vrrp** *number*
5. **shutdown**
6. **track interface type** *number* **priority** *value*
7. **no shutdown**
8. （任意） **show vrrp**
9. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**interface** *interface -type slot/port*
	例：
	`switch(config)#`
	`switch(config-if)# interface ethernet 2/1`
		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	**no switchport** 例： `switch(config-if)# no switchport`	そのインターフェイスを、レイヤ 3 ルーテッド インターフェイスとして設定します。

ステップ 4	**vrrp** *number*
	例：
	`switch(config-if)# vrrp 250`
	`switch(config-if-vrrp)#`
		仮想ルータ グループを作成します。範囲は 1 ～ 255 です。

	コマンドまたはアクション	目的

ステップ 5	**shutdown**
	例：
	`switch(config-if-vrrp)# shutdown`
	`switch(config-if-vrrp)#`
		VRRP グループを無効にします。デフォルトでは無効です。

ステップ 6	**track interface type** *number* **priority** *value* 例： `switch(config-if-vrrp)# track interface ethernet 2/10 priority 254`	VRRP グループのインターフェイス プライオリティ トラッキングをイネーブルにします。プライオリティの範囲は 1 ～ 254 です。

ステップ 7	**no shutdown**
	例：
	`switch(config-if-vrrp)# no shutdown`
	`switch(config-if-vrrp)#`
		VRRP グループを有効にします。デフォルトでは無効です。

ステップ 8	（任意） **show vrrp** 例： `switch(config-if-vrrp)# show vrrp`	VRRP 情報を表示します。
ステップ 9	（任意） **copy running-config startup-config** 例： `switch(config-if-vrrp)# copy running-config startup-config`	この設定変更を保存します。

VRRP の設定の確認

VRRP の設定情報を表示するには、次のいずれかの作業を行います。

コマンド	目的
show vrrp	すべてのグループについて、VRRP ステータスを表示します。
show vrrp vr group-number	1 つの VRRP グループについて、VRRP ステータスを表示します。
show vrrp interface *interface-type port vr number*	インターフェイスの仮想ルータ設定を表示します。

VRRP 統計情報の表示

VRRP の統計情報を表示するには、次のコマンドを使用します。

コマンド	目的
show vrrp statistics interface *interface-type port vr number*	仮想ルータ情報を表示します。
show vrrp statistics	VRRPの統計情報を表示します。

特定のインターフェイスについて、IPv4 VRRP 統計情報を消去するには、**clear vrrp vr** コマンドを使用します。

VRRP の設定例

この例では、ルータ A とルータ B はそれぞれ 3 つの VRRP グループに属しています。コンフィギュレーションにおいて、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 は優先順位 120 で、このグループのプライマリになります。
  - 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 は優先順位 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：
  - ルータ A は、IP アドレスが上位（10.1.0.2）なので、このグループのプライマリになります。
  - 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。

ルータ A

```
interface ethernet 1/0
no switchport

ip address 10.1.0.2/16
no shutdown
vrrp 1
priority 120
authentication text cisco
advertisement-interval 3
```

```
address 10.1.0.10
no shutdown
vrrp 5
priority 100
advertisement-interval 30
address 10.1.0.50
no shutdown
vrrp 100
no preempt
address 10.1.0.100
no shutdown
```

ルータ B

```
interface ethernet 1/0
no switchport

ip address 10.2.0.1/2
no shutdown
vrrp 1
priority 100
authentication text cisco
advertisement-interval 3
address 10.2.0.10
no shutdown

vrrp 5
priority 200
advertisement-interval 30
address 10.2.0.50
no shutdown
vrrp 100
no preempt
address 10.2.0.100
no shutdown
```

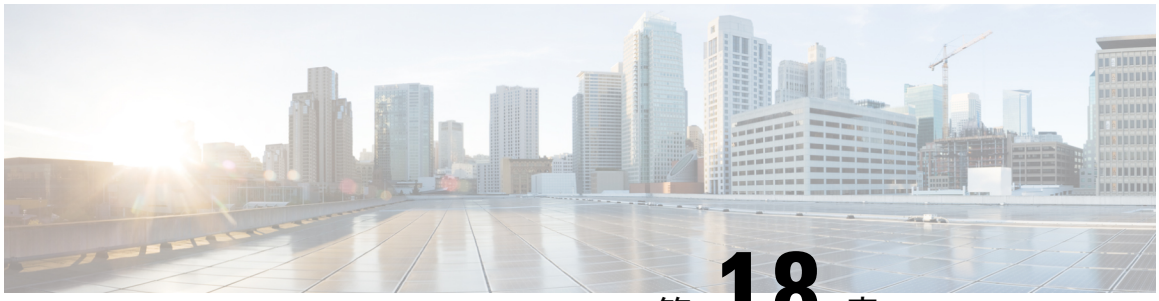
その他の参考資料

VRRP の実装に関連する詳細情報については、次の項を参照してください。

- 関連資料

関連資料

関連項目	マニュアル タイトル
Hot Standby Router Protocol の設定	『Configuring HSRP』
VRRP CLI コマンド	『Cisco Nexus 3000 Series Command Reference』



第 18 章

オブジェクト トラッキングの設定

この章では、Cisco NX-OS スイッチ上でオブジェクト トラッキングを設定する方法について説明します。

この章は、次の項で構成されています。

オブジェクト トラッキングについて

オブジェクト トラッキングを使用すると、インターフェイス ライン プロトコル ステート、IP ルーティング、ルート到達可能性などの、スイッチ上の特定のオブジェクトをトラッキングし、トラッキング対象オブジェクトのステートが変化したときに対処できます。この機能により、ネットワークのアベイラビリティが向上し、オブジェクトがダウンした場合のリカバリ時間が短縮されます。

オブジェクト トラッキングの概要

オブジェクト トラッキング機能を使用すると、トラッキング対象オブジェクトを作成できます。複数のクライアントでこのオブジェクトを使用し、トラッキング対象オブジェクトが変化したときのクライアント動作を変更できます。複数のクライアントがそれぞれの関心をトラッキング プロセスに登録し、同じオブジェクトをトラッキングし、オブジェクトのステートが変化したときに異なるアクションを実行します。

クライアントには次の機能が含まれます。

- ホットスタンバイ冗長プロトコル（HSRP）

- 仮想ルータ冗長プロトコル（VRRP）
- Embedded Event Manager（EEM）

オブジェクト トラッキングは、トラッキング対象オブジェクトのステータスをモニタし、変更があった場合は関係クライアントに伝えます。各トラッキング対象オブジェクトは、一意の番号で識別します。クライアントはこの番号を使用して、トラッキング対象オブジェクトのステートが変化したときに実行するアクションを設定できます。

Cisco NX-OS がトラッキングするオブジェクト タイプは、次のとおりです。

- インターフェイス ラインプロトコル ステート：ラインプロトコル ステートがアップまたはダウンかどうかをトラッキングします。
- インターフェイス IP ルーティング ステート：インターフェイスに IPv4 アドレスが設定されていて、IPv4 ルーティングがイネーブルでアクティブかどうかをトラッキングします。
- IP ルート到達可能性：IPv4 ルートが存在していて、ローカル スイッチから到達可能かどうかをトラッキングします。

たとえば、HSRP を設定すると、冗長ルータの 1 つをネットワークの他の部分に接続するインターフェイスのラインプロトコルをトラッキングできます。そのリンクがダウンした場合、影響のある HSRP ルータのプライオリティを変更できます。

オブジェクト トラッキング リスト

オブジェクト トラッキング リストを使用すると、複数のオブジェクトのステートをまとめてトラッキングできます。オブジェクト トラッキング リストは次の機能をサポートします。

- ブール「and」機能：トラッキング リスト オブジェクトがアップになるには、トラッキング リスト内に定義された各オブジェクトがアップ状態である必要があります。
- ブール「or」機能：トラッキング対象オブジェクトがアップになるには、トラッキング リスト内に定義された少なくとも 1 つのオブジェクトがアップ状態である必要があります。
- しきい値パーセンテージ：トラッキング対象リストに含まれるアップ オブジェクトのパーセンテージが、アップ状態になるトラッキング リストの設定されたアップしきい値を上回っている必要があります。トラッキング対象リストに含まれるダウン オブジェクトのパーセンテージが設定されたトラッキング リストのダウンしきい値を上回っている場合、トラッキング対象リストはダウンとしてマークされます。
- しきい値の重み：トラッキング対象リスト内の各オブジェクトに重み値を割り当て、トラッキング リストに重みしきい値を割り当てます。すべてのアップ オブジェクトの重み値の合計がトラッキング リストの重みアップしきい値を超えている場合、トラッキング リストはアップ状態になります。すべてのダウン オブジェクトの重み値の合計がトラッキング リストの重みダウンしきい値を超えている場合、トラッキング リストはダウン状態になります。

トラックリストの詳細については、「ブール式を含むオブジェクト トラッキング リストの設定」を参照してください。

仮想化のサポート

オブジェクト トラッキングは仮想ルーティングおよび転送（VRF）インスタンスをサポートします。デフォルトでは、特に別の VRF を設定しない限り、Cisco NX-OS はユーザーをデフォルトの VRF に配置します。Cisco NX-OS はデフォルトで、デフォルト VRF のオブジェクトのルート到達可能ステートをトラッキングします。別の VRF のオブジェクトをトラッキングする場合は、オブジェクトをその VRF のメンバとして設定する必要があります（非デフォルト VRF に対するオブジェクト トラッキングの設定」の項を参照）。

オブジェクト トラッキングに関する注意事項と制約事項

オブジェクト トラッキング設定時の注意事項および制約事項は、次のとおりです。

- 最大 500 のトラッキング対象オブジェクトをサポートします。
- イーサネット、サブインターフェイス、トンネル、ポート チャネル、ループバック インターフェイス、および VLAN インターフェイスをサポートします。
- HSRP グループごとに 1 つのトラッキング対象オブジェクトをサポートします。

オブジェクト トラッキングのデフォルト設定

下の表に、オブジェクト トラッキング パラメータのデフォルト設定を示します。

表 *21*：デフォルトのオブジェクト トラッキング パラメータ

パラメータ	デフォルト
Tracked Object VRF	デフォルト VRF のメンバ

オブジェクト トラッキングの設定

（注） Cisco IOS の CLI に慣れている場合、この機能に対応する Cisco NX-OS コマンドは通常使用する Cisco IOS コマンドと異なる場合があるので注意してください。

インターフェイスに対するオブジェクト トラッキングの設定

インターフェイスのライン プロトコルまたは IPv4 ルーティングの状態をトラッキングするように Cisco NX-OS を設定できます。

手順の概要

1. **configure terminal**
2. **track** *object-id* **interface** *interface-type number* { **ip routing** | **line-protocol** }
3. （任意） **show track** [*object-id*]
4. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**track** *object-id* **interface** *interface-type number* { **ip routing** | **line-protocol** }
	例：
	`switch(config)# track 1 interface ethernet 1/2 line-protocol`
	`switch(config-track#`
		インターフェイスのトラッキング対象オブジェクトを作成し、トラッキング コンフィギュレーション モードを開始します。*object-id* の範囲は 1 ～ 500 です。

ステップ 3	（任意） **show track** [*object-id*] 例： `switch(config-track)# show track 1`	オブジェクトのトラッキング情報を表示します。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config-track)# copy running-config startup-config`	この設定変更を保存します。

例

Ethernet 1/2 上でライン プロトコル ステートのオブジェクト トラッキングを設定する例を示します。

```
switch# configure terminal
switch(config)# track 1 interface ethernet 1/2 line-protocol
switch(config-track)# copy running-config startup-config
```

Ethernet 1/2 上で IPv4 ルーティング ステートのオブジェクト トラッキングを設定する例を示します。

```
switch# configure terminal
switch(config)# track 2 interface ethernet 1/2 ip routing
switch(config-track)# copy running-config startup-config
```

ルート到達可能性に対するオブジェクト トラッキングの設定

IP ルートの存在および到達可能性をトラッキングするように Cisco NX-OS を設定できます。

手順の概要

1. **configure terminal**
2. **track** *object-id* **ip route** *prefix/length* **reachability**
3. （任意） **show track** [*object-id*]
4. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**track** *object-id* **ip route** *prefix/length* **reachability**
	例：
	`switch(config)# track 2 ip route 192.0.2.0/8 reachability`
	`switch(config-track)#`
		ルートのトラッキング対象オブジェクトを作成し、トラッキング コンフィギュレーション モードを開始します。*object-id* の範囲は 1 ～ 500 です。IP のプレフィックス フォーマットは A.B.C.D/length です。length の範囲は 1 ～ 32 です。

ステップ 3	（任意） **show track** [*object-id*] 例： `switch(config-track)# show track 1`	オブジェクトのトラッキング情報を表示します。
ステップ 4	（任意） **copy running-config startup-config** 例： `switch(config-track)# copy running-config startup-config`	この設定変更を保存します。

例

次に、デフォルト VRF で IPv4 ルートのオブジェクト トラッキングを設定する例を示します。

```
switch# configure terminal
switch(config)# track 4 ip route 192.0.2.0/8 reachability
switch(config-track)# copy running-config startup-config
```

ブール式を含むオブジェクト トラッキング リストの設定

複数のトラッキング対象オブジェクトを含むオブジェクト トラッキング リストを設定できます。トラッキング対象リストには 1 つまたは複数のオブジェクトが含まれます。ブール式では、「and」または「or」演算子を使用して 2 種類の演算を実行できます。たとえば、「and」演算子を使用して2つのインターフェイスをトラッキングする場合、「アップ」は両方のインターフェイスがアップであることを意味し、「ダウン」はどちらかのインターフェイスがダウンであることを意味します。

手順の概要

1. **configure terminal**
2. **track** *track-number* **list boolean** { **and** | **or** }
3. **object** *object-id* [**not**]
4. （任意） **show track**
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**track** *track-number* **list boolean** { **and**	**or** }
		例：
		`switch(config)# track 1 list boolean and`
		`switch(config-track#`
			トラッキング対象リスト オブジェクトを設定し、トラッキング設定モードを開始します。トラッキング対象リストのステートがブール式に基づいて決まることを指定します。キーワードは次のとおりです。
			• and：すべてのオブジェクトがアップである場合にリストがアップになり、1 つ以上のオブジェクトがダウンの場合にリストがダウンになることを指定します。たとえば 2 つのインターフェイスをトラッキングする場合、アップは両方のインターフェイスがアップ状態であることを表し、ダウンはいずれかのインターフェイスがダウン状態であることを表します。
			• **or**：少なくとも 1 つのオブジェクトがアップであればリストがアップになるように指定します。たとえば 2 つのインターフェイスをトラッキングする場合、アップはいずれか一方のインターフェイスがアップ状態であることを意味し、ダウンは両方のインターフェイスがダウン状態であることを意味します。

	コマンドまたはアクション	目的
		track-number の範囲は 1 ～ 500 です。
ステップ 3	**object** *object-id* [**not**] 例： `switch(config-track)# object 10`	トラッキング リストにトラッキング対象オブジェクトを追加します。*object-id* の範囲は 1 ～ 500 です。オプションの **not** キーワードを指定すると、トラッキング対象オブジェクトのステートが否定されます。 （注） 例では、オブジェクト 10 がアップのときに、トラッキング対象リストがオブジェクト 10 をダウンとして検出します。
ステップ 4	（任意） **show track** 例： `switch(config-track)# show track`	オブジェクトのトラッキング情報を表示します。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config-track)# copy running-config startup-config`	この設定変更を保存します。

例

次に、複数のオブジェクトを含むトラッキング リストをブール「and」で設定する例を示します。

```
switch# configure terminal
switch(config)# track 1 list boolean and
switch(config-track)# object 10
switch(config-track)# object 20 not
```

パーセンテージしきい値を含むオブジェクト トラッキング リストの設定

パーセンテージしきい値を含むオブジェクト トラッキング リストを設定できます。トラッキング対象リストには1つまたは複数のオブジェクトが含まれます。トラッキング リストがアップ状態になるには、アップ オブジェクトのパーセンテージがトラッキング リストに設定されたパーセントしきい値を超えている必要があります。たとえば、追跡対象リストに3つのオブジェクトが含まれており、アップしきい値を60%に設定した場合は、2つのオブジェクト（全オブジェクトの 66 %）がアップ状態になるまで、追跡リストがアップ状態になりません。

手順の概要

1. **configure terminal**

2. **track** *track-number* **list threshold percentage**
3. **threshold percentage up** *up-value* **down** *down-value*
4. （任意） **object** [*object-id*]
5. （任意） **show track**
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**track** *track-number* **list threshold percentage**
	例：
	`switch(config)# track 1 list threshold percentage`
	`switch(config-track#`
		トラッキング対象リスト オブジェクトを設定し、トラッキング設定モードを開始します。トラッキング対象リストのステートが設定されたしきい値パーセントに基づいて決まることを指定します。
		track-number の範囲は 1 ～ 500 です。

ステップ 3	**threshold percentage up** *up-value* **down** *down-value* 例： `switch(config-track)# threshold percentage up 70 down 30`	トラッキング対象リストのしきい値パーセントを設定します。指定できる範囲は 0 ～ 100% です。
ステップ 4	（任意） **object** [*object-id*] 例： `switch(config-track)# object 10`	トラッキング リストにトラッキング対象オブジェクトを追加します。*object-id* の範囲は 1 ～ 500 です。
ステップ 5	（任意） **show track** 例： `switch(config-track)# show track`	オブジェクトのトラッキング情報を表示します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config-track)# copy running-config startup-config`	この設定変更を保存します。

例

次に、アップしきい値が 70 % でダウンしきい値が 30 % の追跡リストを設定する例を示します。

```
switch# configure terminal
switch(config)# track 1 list threshold percentage
```

```
switch(config-track)# threshold percentage up 70 down 30
switch(config-track)# object 10
switch(config-track)# object 20
switch(config-track)# object 30
```

重みしきい値を含むオブジェクト トラッキング リストの設定

重みしきい値を含むオブジェクト トラッキング リストを設定できます。トラッキング対象リストには 1 つまたは複数のオブジェクトが含まれます。トラッキング リストがアップ ステートになるには、アップ オブジェクトの重み値の合計がトラッキング リストに設定されたアップ重みしきい値を超えている必要があります。たとえば、トラッキング対象リストに重み値がデフォルトの 10 である 3 つのオブジェクトがあり、アップしきい値を 15 に設定した場合、トラッキング リストがアップ状態になるには、2 つのオブジェクトがアップ状態になる（重み値の合計が 20 になる）必要があります。

手順の概要

1. **configure terminal**
2. **track** *track-number* **list threshold weight**
3. **threshold weight up** *up-value* **down** *down-value*
4. **object** *object-id* **weight** *value*
5. （任意） **show track**
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**track** *track-number* **list threshold weight**
	例：
	`switch(config)# track 1 list threshold weight`
	`switch(config-track#`
		トラッキング対象リスト オブジェクトを設定し、トラッキング設定モードを開始します。トラッキング対象リストのステートが設定されたしきい値重みに基づいて決まることを指定します。
		track-number の範囲は 1 ～ 500 です。

ステップ 3	**threshold weight up** *up-value* **down** *down-value* 例： `switch(config-track)# threshold weight up 30 down 10`	トラッキング対象リストのしきい値重みを設定します。指定できる範囲は 1 ～ 255 です。
ステップ 4	**object** *object-id* **weight** *value* 例：	トラッキング リストにトラッキング対象オブジェクトを追加します。*object-id* の範囲は 1 ～ 500 です。

	コマンドまたはアクション	目的
	`switch(config-track)# object 10 weight 15`	*value* の範囲は 1 ～ 255 です。デフォルトの重み値は 10 です。
ステップ 5	（任意） **show track** **例**： `switch(config-track)# show track`	オブジェクトのトラッキング情報を表示します。
ステップ 6	（任意） **copy running-config startup-config** **例**： `switch(config-track)# copy running-config startup-config`	この設定変更を保存します。

例

次に、トラッキング リストのアップ重みしきい値を 30、ダウンしきい値を 10 にそれぞれ設定する例を示します。

```
switch# configure terminal
switch(config)# track 1 list threshold weight
switch(config-track)# threshold weight up 30 down 10
switch(config-track)# object 10 weight 15
switch(config-track)# object 20 weight 15
switch(config-track)# object 30
```

この例では、オブジェクト 10 とオブジェクト 20 がアップの場合にトラッキング リストがアップになり、3 つのオブジェクトがすべてダウンの場合にトラッキング リストがダウンになります。

オブジェクト トラッキングの遅延の設定

トラッキング対象オブジェクトまたはオブジェクト トラッキング リストに対して、オブジェクトまたはリストがステートの変化を開始したときに適用する遅延を設定できます。トラッキング対象オブジェクトまたはトラッキング リストは、ステートの変化が発生したときに遅延タイマーを開始しますが、遅延タイマーが切れるまでステートの変化を認識しません。遅延タイマーが切れると、Cisco NX-OS は再びオブジェクトのステートを確認し、オブジェクトまたはリストが現在も変更されたステートのままだった場合にだけステートの変化を記録します。オブジェクト トラッキングは遅延タイマーが切れる前の中間的なステートの変化を無視します。

たとえば、インターフェイス ライン プロトコルのトラッキング対象オブジェクトがアップ ステートであり、ダウン遅延が 20 秒に設定されている場合は、ライン プロトコルがダウンになると遅延タイマーが開始します。20 秒後にライン プロトコルがダウンになっていなければ、このオブジェクトはダウン ステートになりません。

トラッキング対象オブジェクトまたはトラッキング リストには、独立したアップ遅延とダウン遅延を設定できます。遅延を削除すると、オブジェクト トラッキングからアップ遅延とダウン遅延の両方が削除されます。

遅延は任意の時点で変更できます。オブジェクトまたはリストがトリガーされたイベントから遅延タイマーをすでにカウントしている場合は、次のようにして新しい遅延が計算されます。

- 新しい設定値が古い設定値より小さい場合は、新しい値でタイマーが開始します。

  .

- 新しい設定値が古い設定値より大きい場合は、新しい設定値から現在のタイマーのカウントダウンを引き、古い設定値を引いたものがタイマーになります。

手順の概要

1. **configure terminal**
2. track *object- id* { *parameters* }
3. **track** *track-number* **list**{ *parameters* }
4. **delay** { **up** *up-time* [**down** *down-time*] | **down** *down-time* [**up** *up-time*]}
5. （任意） **show track**
6. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	track *object- id* { *parameters* }
	例：
	`switch(config)# track 2 ip route 192.0.2.0/8 reachability`
	`switch(config-track)#`
		ルートのトラッキング対象オブジェクトを作成し、トラッキング コンフィギュレーション モードを開始します。*object-id* の範囲は 1 ～ 500 です。IP のプレフィックス フォーマットは A.B.C.D/length です。length の範囲は 1 ～ 32 です。

ステップ 3	**track** *track-number* **list**{ *parameters* }
	例：
	`switch(config)# track 1 list threshold weight`
	`switch(config-track#`
		トラッキング対象リスト オブジェクトを設定し、トラッキング設定モードを開始します。トラッキング対象リストのステートが設定されたしきい値重みに基づいて決まることを指定します。
		track-number の範囲は 1 ～ 500 です。

ステップ 4	**delay** { **up** *up-time* [**down** *down-time*]	**down** *down-time* [**up** *up-time*]} 例： `switch(config-track)# delay up 20 down 30`	オブジェクトの遅延タイマーを設定します。指定できる範囲は 0 ～ 180 秒です。

	コマンドまたはアクション	目的
ステップ 5	（任意） **show track** 例： `switch(config-track)# show track`	オブジェクトのトラッキング情報を表示します。
ステップ 6	（任意） **copy running-config startup-config** 例： `switch(config-track)# copy running-config startup-config`	この設定変更を保存します。

例

次に、ルートのオブジェクト トラッキングを設定し、遅延タイマーを使用する例を示します。

```
switch# configure terminal
switch(config)# track 2 ip route 209.165.201.0/8 reachability
switch(config-track)# delay up 20 down 30
switch(config-track)# copy running-config startup-config
```

次に、トラッキング リストのアップ重みしきい値を 30、ダウンしきい値を 10 にそれぞれ設定し、遅延タイマーを使用する例を示します。

```
switch# configure terminal
switch(config)# track 1 list threshold weight
switch(config-track)# threshold weight up 30 down 10
switch(config-track)# object 10 weight 15
switch(config-track)# object 20 weight 15
switch(config-track)# object 30
switch(config-track)# delay up 20 down 30
```

次に、インターフェイスがシャットダウンする前後の show track コマンドの出力に表示された遅延タイマーの例を示します。

```
switch(config-track)# show track
Track 1
Interface loopback1 Line Protocol
Line Protocol is UP
1 changes, last change 00:00:13
Delay down 10 secs

switch(config-track)# interface loopback 1
switch(config-if)# shutdown
switch(config-if)# show track
Track 1
Interface loopback1 Line Protocol
Line Protocol is delayed DOWN (8 secs remaining)<------- delay timer counting down
1 changes, last change 00:00:22
Delay down 10 secs
```

非デフォルト VRF に対するオブジェクト トラッキングの設定

特定の VRF でオブジェクトをトラッキングするように Cisco NX-OS を設定できます。

手順の概要

1. **configure terminal**
2. **track object-id ip route** *prefix/length* **reachability**
3. **vrf member** *vrf-name*
4. （任意） **show track** [*object-id*]
5. （任意） **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的

ステップ 1	**configure terminal**
	例：
	`switch# configure terminal`
	`switch(config)#`
		コンフィギュレーション モードに入ります。

ステップ 2	**track object-id ip route** *prefix/length* **reachability**
	例：
	`switch(config)# track 2 ip route 192.0.2.0/8 reachability`
	`switch(config-track)#`
		ルートのトラッキング対象オブジェクトを作成し、トラッキング コンフィギュレーション モードを開始します。*object-id* の範囲は 1 ～ 500 です。IP のプレフィックス フォーマットは A.B.C.D/length です。length の範囲は 1 ～ 32 です。

ステップ 3	**vrf member** *vrf-name* 例： `switch(config-track)# vrf member Red`	設定されたオブジェクトのトラッキングに使用する VRF を設定します。
ステップ 4	（任意） **show track** [*object-id*] 例： `switch(config-track)# show track 3`	オブジェクトのトラッキング情報を表示します。
ステップ 5	（任意） **copy running-config startup-config** 例： `switch(config-track)# copy running-config startup-config`	この設定変更を保存します。

例

ルートのオブジェクト トラッキングを設定し、VRF Red を使用して、そのオブジェクトの到達可能性情報を調べる例を示します。

```
switch# configure terminal
switch(config)# track 2 ip route 209.165.201.0/8 reachability
switch(config-track)# vrf member Red
switch(config-track)# copy running-config startup-config
```

次に、トラッキング対象オブジェクト 2 を変更して、VRF Red の代わりに VRF Blue を使用してこのオブジェクトの到達可能性情報を調べるようにする例を示します。

```
switch# configure terminal
switch(config)# track 2
switch(config-track)# vrf member Blue
switch(config-track)# copy running-config startup-config
```

オブジェクト トラッキングの設定の確認

オブジェクト トラッキングの設定情報を表示するには、次のいずれかの作業を行います。

コマンド	目的
show track [*object-id*] [**brief**]	1つまたは複数のオブジェクトについて、オブジェクト トラッキング情報を表示します。
show track [*object-id*] **interface** [**brief**]	インターフェイスベースのオブジェクト トラッキング情報を表示します。
show track [*object-id*] **ip-route** [**brief**]	IPv4 ルートベースのオブジェクト トラッキング情報を表示します。

オブジェクト トラッキングの設定例

次の例は、ルート到達可能性に対してオブジェクト トラッキングを設定し、VRF Red を使用してルートの到達可能性情報を調べる方法を示しています。

```
switch# configure terminal
switch(config)# track 2 ip route 209.165.201.0/8 reachability
switch(config-track)# vrf member Red
switch(config-track)# copy running-config startup-config
```

その他の参考資料

オブジェクト トラッキングの実装に関連する詳細情報については、次の項を参照してください。

- 関連資料
- 標準

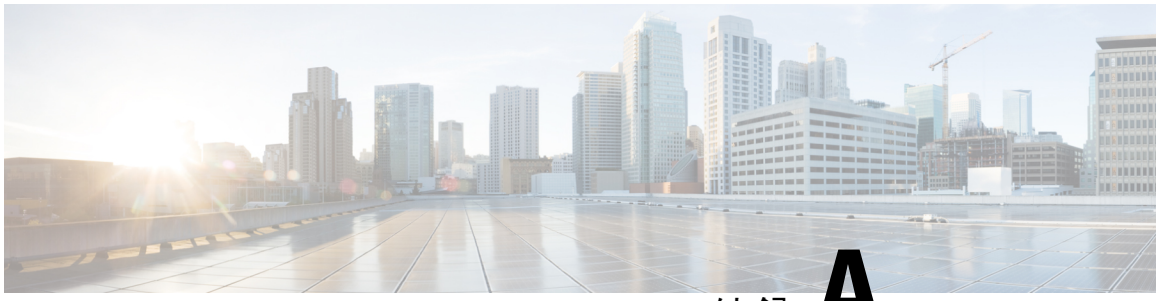
関連資料

関連項目	マニュアル タイトル
オブジェクト トラッキング CLI コマンド	『Cisco Nexus 3000 Series Command Reference』
Embedded Event Manager の設定	Cisco Nexus 3548 スイッチ NX-OS システム管理構成ガイド

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。	—

標準



付録 A

IETF RFC

この付録は、サポートされている IETF RFC の一覧です。

IETF RFC

BGP の RFC

RFC	タイトル
RFC 1997	*BGP* コミュニティの属性
RFC 2385	*TCP MD5* シグネチャ オプションを使用した *BGP* セッションの保護
RFC 2439	*BGP* ルート フラップ ダンピング
RFC 2519	ドメイン ルート間集約のフレームワーク
RFC 2858	*BGP-4* のマルチプロトコル拡張
RFC 3065	*BGP* の自律システム連合
RFC 3392	*BGP-4* による機能のアドバタイズメント
RFC 4271	ボーダー ゲートウェイ プロトコル *4 (BGP-4)*
RFC 4273	*BGP-4* の管理対象オブジェクトの定義
RFC 4456	*BGP* ルート リフレクション*:* フルメッシュ内部 *BGP (IBGP)*の代替
RFC 4486	*BGP Cease* 通知メッセージのサブコード
RFC 4893	*4* オクテット *AS* 番号スペースの *BGP* サポート
RFC 5004	*1* つの外部から別の外部への *BGP* 最良パス移行の回避

RFC	タイトル
draft-ietf-idr-bgp4-mib-15.txt	*BGP4-MIB*

ファーストホップ冗長プロトコルの **RFC**

RFC	タイトル
RFC 2281	『*Hot Standby Redundancy Protocol*』
RFC 3768	『*Virtual Router Redundancy Protocol*』

IP サービスに関する **RFC** の参考資料

RFC	タイトル
RFC 786	*UDP*
RFC 791	*IP*
RFC 792	*ICMP*
RFC 793	*[TCP]*
RFC 826	『*ARP*』
RFC 1027	『*Proxy ARP*』
RFC 1591	『*DNS Client*』
RFC 1812	『*IPv4 routers*』

OSPF の **RFC**

RFC	タイトル
RFC 2328	『*OSPF Version 2*』
RFC 3101	『*The OSPF Not-So-Stubby Area (NSSA) Option*』
RFC 2370	『*The OSPF Opaque LSA Option*』
RFC 3137	『*OSPF Stub Router Advertisement*』

IETF RFC

RIP の RFC

RFC	タイトル
RFC 2453	*RIP* バージョン *2*
RFC 2082	*RIP-2 MD5* 認証

IETF RFC

翻訳について
このドキュメントは、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。