Cisco Nexus 9000 シリーズ NX-OS ユニキャスト ルーティング コンフィギュレーション ガイド、リリース 7.x

初版:2015 年 2 月
最終更新日:2016 年 5 月

Cisco Systems, Inc.
www.cisco.com

シスコは世界各国 200 箇所にオフィスを開設しています。
各オフィスの住所、電話番号、FAX 番号は
当社の Web サイトをご覧ください。
www.cisco.com/go/offices

【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、米国シスコシステムズ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。

Cisco Nexus 9000 シリーズ NX-OS ユニキャスト ルーティング コンフィギュレーション ガイド、リリース 7.x

新機能および変更された機能に関する情報

この章では、*Cisco Nexus 9000 シリーズ NX-OS ユニキャスト ルーティング コンフィギュレーション ガイド、リリース 7.x* の新機能および変更された機能に関するリリース固有の情報を示します。このマニュアルの最新バージョンは、次のシスコ Web サイトから入手できます。
http://www.cisco.com/en/US/docs/switches/datacenter/nexus9000/sw/7.x/unicast/configuration/guide/l3_cli_nxos.html

Cisco NX-OS Release 7.x に関するその他の情報については、『*Cisco Nexus 9000 Series NX-OS Release Notes*』を参照してください。このドキュメントは、次のシスコ Web サイトから入手できます。
http://www.cisco.com/en/US/docs/switches/datacenter/nexus9000/sw/7.x/release/notes/61-nxos-rn.html

表 1 では、*Cisco Nexus 9000 シリーズ NX-OS ユニキャスト ルーティング コンフィギュレーション ガイド、リリース 7.x* における新機能および変更された機能を要約し、その参照先を示しています。

表 1　　リリース 7.x の新機能および機能変更

機能	説明	変更されたリリース	参照先
Route Policy Manager	IP プレフィックス リストと **match ospf-area** コマンドでのマスク サポートが追加されました。	7.0(3)I4(1)	第 15 章「Route Policy Manager の設定」
OSPFv2	HMAC-SHA 認証および RFC 5709 のサポートを追加。	7.0(3)I3(1)	第 5 章「OSPFv2 の設定」 付録 A「Cisco NX-OS ユニキャスト機能でサポートされている IETF RFC」
OSPFv3	IPSec 認証のサポートと RFC 4552 の部分的なサポートが追加されました。	7.0(3)I3(1)	第 6 章「OSPFv3 の設定」 付録 A「Cisco NX-OS ユニキャスト機能でサポートされている IETF RFC」
BGP	RFC 5549 のサポートが追加されました。	7.0(3)I2(1)	第 9 章「ベーシック BGP の設定」 第 10 章「Configuring Advanced BGP」 付録 A「Cisco NX-OS ユニキャスト機能でサポートされている IETF RFC」

表 1　　リリース 7.x の新機能および機能変更（続き）

機能	説明	変更されたリリース	参照先
HSRP	ダブルサイド vPC のすべてのノードで同じ HSRP グループを設定できる機能のサポートが追加されました。	7.0(3)I2(1)	第 17 章「HSRP の設定」
VRF ルート リーク	デフォルト以外の VRF からデフォルト VRF へのルート リークのサポートが追加されました。	7.0(3)I2(1)	第 13 章「レイヤ 3 仮想化の設定」
BFD	BGP、EIGRP、IS-IS、OSPFv3 用の Bidirectional Forwarding Detection（BFD）での IPv6 サポートが追加されました。	7.0(3)I1(1)	第 6 章「OSPFv3 の設定」 第 7 章「EIGRP の設定」 第 8 章「IS-IS の設定」 第 10 章「Configuring Advanced BGP」
EIGRP	ノンストップ フォワーディング（NSF）中に EIGRP が再配布されたプロトコルのコンバージェンスを待機してからルーティング情報ベース（RIB）に固有のルートをインストールするよう指定するコマンドが追加されました。	7.0(3)I1(1)	第 7 章「EIGRP の設定」
IPv4 and IPv6	Cisco Nexus 9300 シリーズ スイッチ用の ALPM ルーティング モードが追加されました。	7.0(3)I1(1)	第 2 章「IPv4 の設定」および第 3 章「IPv6 の設定」
VRRP	VRRPv3 および VRRS のサポートが追加されました。	7.0(3)I1(1)	第 18 章「VRRP の設定」

はじめに

ここでは、『*Cisco Nexus 9000 シリーズ NX-OS ユニキャスト ルーティング コンフィギュレーション 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。関連情報の取得方法も紹介します。

この前書きは、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。
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 screen 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。

関連資料

Cisco NX-OS には、次の資料が含まれます。

リリース ノート

『Cisco Nexus 9000 Series NX-OS Release Notes』

『Cisco Nexus 9000 Series FPGA/EPLD Upgrade Release Notes』

Cisco NX-OS コンフィギュレーション ガイド

『Cisco Nexus 2000 Series NX-OS Fabric Extender Software Configuration Guide for Cisco Nexus 9000 シリーズ スイッチ』

『Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide』

『Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide』

『Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide』

『Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide』

『Cisco Nexus 9000 Series NX-OS Multicast Routing Configuration Guide』

『Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide』

『Cisco Nexus 9000 Series NX-OS Security Configuration Guide』

『*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*』

『*Cisco Nexus 9000 シリーズ NX-OS ユニキャスト ルーティング コンフィギュレーション ガイド*』

『*Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*』

『*Cisco Nexus 9000 Series Virtual Machine Tracker Configuration Guide*』

『*Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide*』

その他のソフトウェアのマニュアル

『*Cisco Nexus 7000 Series and 9000 Series NX-OS MIB Quick Reference*』

『*Cisco Nexus 9000 Series NX-OS Programmability Guide*』

『*Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide*』

『*Cisco Nexus 9000 Series NX-OS System Messages Reference*』

『*Cisco Nexus 9000 Series NX-OS Troubleshooting Guide*』

『*Cisco NX-OS Licensing Guide*』

『*Cisco NX-OS XML Interface User Guide*』

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、HTML ドキュメント内のフィードバックフォームよりご連絡ください。ご協力をよろしくお願いいたします。

マニュアルの入手方法およびテクニカル サポート

マニュアルの入手方法、テクニカル サポート、その他の有用な情報について、次の URL で、毎月更新される『*What's New in Cisco Product Documentation (Cisco 製品資料の更新情報)*』を参照してください。シスコの新規および改訂版の技術マニュアルの一覧も示されています。
http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html

『*What's New in Cisco Product Documentation*』では、シスコの新規および改訂版の技術マニュアルの一覧を、RSS フィードとして購読できます。また、リーダー アプリケーションを使用して、コンテンツをデスクトップに配信することもできます。RSS フィードは無料のサービスです。

概要

この章では でのレイヤ 3 ユニキャスト ルーティング プロトコルの基盤となる概念を紹介しますCisco NX-OS.

この章は、次の項で構成されています。

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、最適なルーティング パスの決定とパケットの交換という、2 つの基本的動作があります。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。

この項では、次のトピックについて取り上げます。

ルーティングの基本

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、次のルータまたはネクスト 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」セクション（1-11 ページ）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のその他の情報も含まれる場合があります。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」セクション（1-3 ページ）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、ルーティング テーブルの全部または一部で構成されるメッセージです。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」セクション（1-8 ページ）を参照してください。

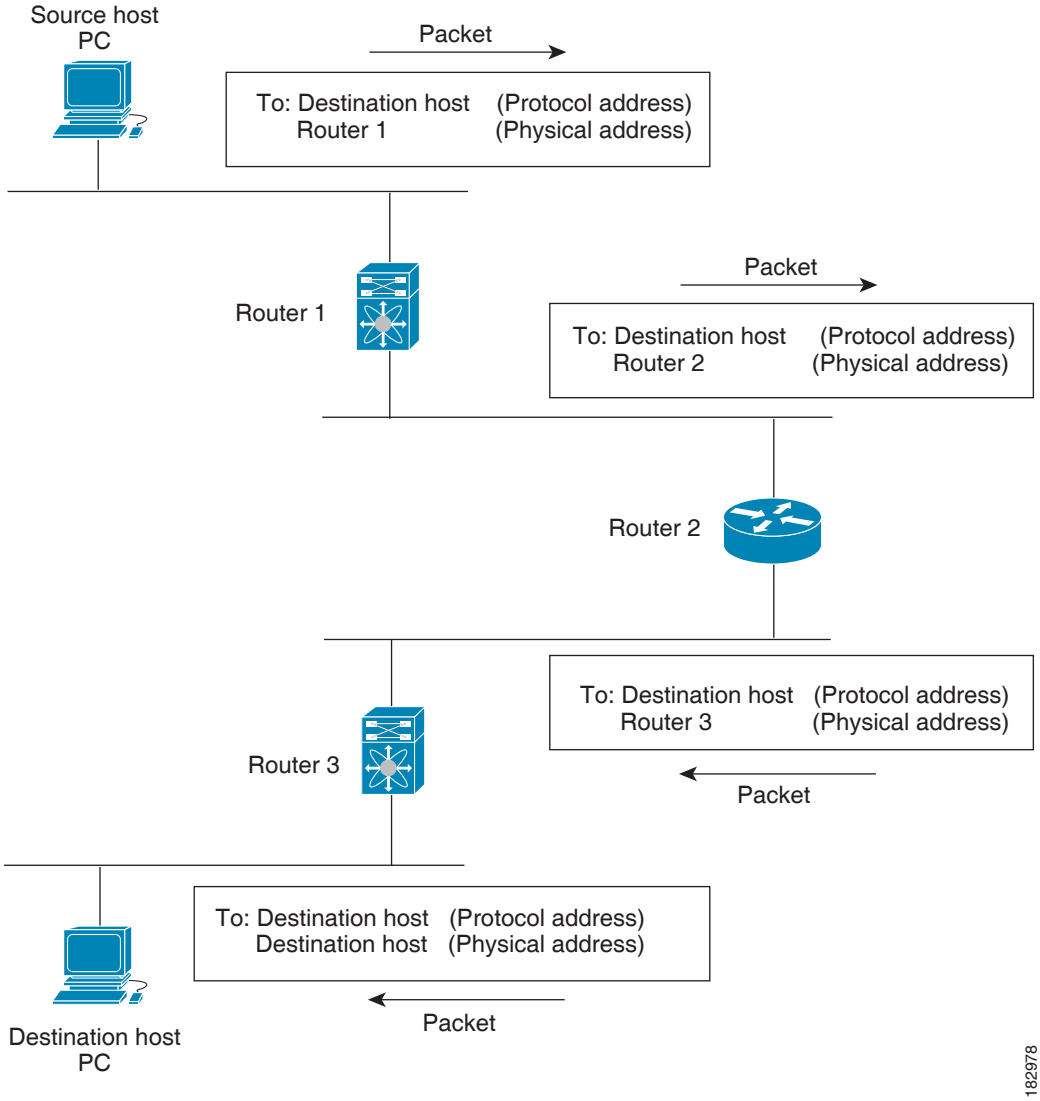
パケット交換

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。何らかの手段でルータ アドレスを取得したら、送信元ホストは、明確にルータの物理（メディア アクセス コントロール（MAC）レイヤ）アドレスにアドレス指定されているが、宛先ホストの IP（ネットワーク層）アドレスを含むパケットを送信します。

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。パケットがネットワーク間を移動するにつれ、その物理アドレスは変更されますが、そのプロトコル アドレスは変わりません（図 1-1を参照）。

図 1-1　ネットワーク上でのパケット 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。

ここでは、次のメトリックについて説明します。

- パス長（1-4 ページ）
- 信頼性（1-4 ページ）
- ルーティング遅延（1-4 ページ）
- 帯域幅（1-4 ページ）
- 負荷（1-4 ページ）
- 通信コスト（1-4 ページ）

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。

信頼性

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 リソースが使用状況の程度です。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 プロセスに関連付けられているルータ ID があります。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 を設定しないと、Cisco NX-OS が次の基準に基づいて、ルータ ID を選択します。

- Cisco NX-OS は、他のあらゆるインターフェイス上で loopback0 を優先します。loopback0 が存在しない場合、Cisco NX-OS は、他のあらゆるインターフェイス タイプ上で最初のループバックを優先します。
- ループバック インターフェイスを設定しなかった場合、Cisco NX-OS はルータ ID としてコンフィギュレーション 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 ID 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設定した場合は、ループバック インターフェイスがルータ ID 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、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（RIR）により、インターネットに直接接続する各公共 AS 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。

ボーダー ゲートウェイ プロトコル（BGP）は、asplain と asdot 表記で表示できる 4 バイトの AS 番号をサポートします。

- asplain：10 進表記方式。2 バイトおよび 4 バイト AS 番号をその 10 進数値で表します。たとえば、65526 は 2 バイト AS 番号、234567 は 4 バイト AS 番号になります。
- asdot：AS ドット付き表記方式。2 バイト AS 番号をその 10 進数値で表し、4 バイトの AS 番号をドット付き表記で表します。たとえば、2 バイト AS 番号 65526 は 65526 として表され、4 バイトの AS 番号 65546 は 1.10 として表されます。

BGP の 4 バイト AS 番号機能は、4 バイト AS 番号をサポートしていない BGP スピーカーをまたがって、4 バイトをベースとする AS パス情報を伝播するために使用されます。

（注）　RFC 5396 は部分的にサポートされます。asplain と asdot 表記はサポートされますが、asdot+ 表記はサポートされません。

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、自律システム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 プロトコルは、ロード バランシングまたは当コスト マルチパス（ECMP）を使用して複数のパス間で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複数のルートを学習するときに、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を IP 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。各ルーティング プロトコルによってサポートされる ECMP パスの数については、『*Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*』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、各プロトコルでルート再配布を設定して、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）プロトコルを設定して、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。他のプロトコルからのルートを再配布するルータは、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 メトリックを防ぐ再配布されたルータの固定ルートを設定します。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。

（注）　ルーティング情報の再配布を設定する場合にルート マップを使用する必要があります。

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（1-7 ページ）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評価は低くなります。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

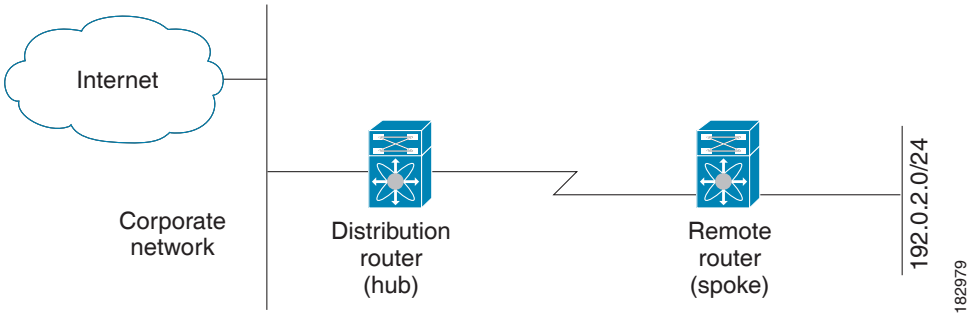
スタブ ルーティングはハブ アンド スポーク型ネットワーク トポロジで使用できます。このトポロジでは、1 つ以上の終端（スタブ）ネットワークが、1 つ以上の分散ルータ（ハブ）に接続されたリモート ルータ（スポーク）に接続されています。リモート ルータは、1 つ以上のディストリビューション ルータにのみ隣接しています。リモート ルータへ流れる IP トラフィックのルートは、ディストリビューション ルータ経由のルートのみです。このタイプの設定は、ディストリビューション ルータが直接 WAN に接続されている WAN トポロジで使用されるのが一般的です。ディストリビューション ルータは、さらに多くのリモート 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。

図 1-2 は、単純なハブ アンド スポーク型設定を示します。

図 1-2　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、Enhanced Interior Gateway Routing Protocol（EIGRP）はスタブ ルータをサポートします。

ルーティング アルゴリズム

ルーティング アルゴリズムは、ルータが到達可能性の情報を収集し、報告する方法、トポロジの変化に対応する方法、および宛先までの最適なルートを決定する方法を決定します。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。

この項では、次のトピックについて取り上げます。

- 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（1-8 ページ）
- 内部および外部ゲートウェイ プロトコル（1-9 ページ）
- ディスタンス ベクトル プロトコル（1-9 ページ）
- リンクステート プロトコル（1-9 ページ）

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、絶えず変化する大規模ネットワークには使用しないでください。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。ボーダー ゲートウェイ プロトコル（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。このプロセスにより、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、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を含むリンクステート アドバタイズメント（LSA）を構築します。

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（同じルータから）同じシーケンス番号の 2 つの LSA を受信した場合、ルータは LSA アップデートのループを回避するため、ネイバーによって受信された最後の LSA を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、そのルータのリンクステート 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、各ルータがそのルーティング テーブル内に、ネットワーク全体の図を構築します。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、より多くの CPU パワーとメモリを必要とし、実行とサポートをするにはよりコストが高くなります。リンクステート プロトコルは通常、ディスタンス ベクトル プロトコルよりスケーラブルです。

OSPF は、リンクステート プロトコルの一例です。

レイヤ 3 仮想化

Cisco NX-OS は、複数の仮想ルーティングおよび転送（VRF）インスタンスおよび複数のルーティング情報ベース（RIB）をサポートしているため、複数のアドレス ドメインがサポートされます。各 VRF は RIB に関連付けられており、この情報が転送情報ベース（FIB）によって収集されます。VRF は、レイヤ 3 アドレス指定ドメインを表します。各レイヤ 3 インターフェイス（論理または物理）は、1 つの VRF に属します。詳細については、第 13 章「レイヤ 3 仮想化の設定」を参照してください。

Cisco NX-OS では、仮想デバイスをエミュレートするバーチャル デバイス コンテキスト（VDC）に、オペレーティング システムおよびハードウェア リソースを分割できます。Cisco Nexus 9000 シリーズ スイッチは、現在複数の VDC をサポートしていません。すべてのスイッチ リソースはデフォルト VDC で管理されます。

Cisco NX-OS転送アーキテクチャ

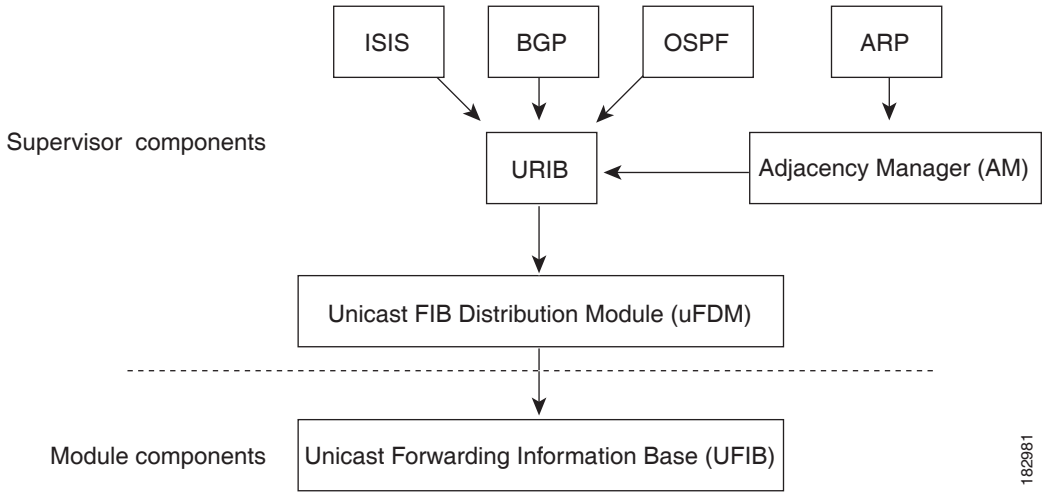
Cisco NX-OS 転送アーキテクチャにより、すべてのルーティングの更新処理と、シャーシ内のすべてのモジュールへの転送情報の入力が行われます。

この項では、次のトピックについて取り上げます。

ユニキャスト RIB

Cisco NX-OS 転送アーキテクチャは、図 1-3 に示すように、複数のコンポーネントで構成されます。

図 1-3　Cisco NX-OS 転送アーキテクチャ



ユニキャスト RIB は、アクティブなスーパーバイザ上にあります。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、特定のルートのための最適なネクスト ホップを決定し、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、モジュール上のユニキャスト 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）。

隣接マネージャ

隣接マネージャは、アクティブなスーパーバイザ上にあり、ARP、ネイバー探索プロトコル（NDP）、スタティック設定などのさまざまなプロトコルの隣接情報を維持します。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。IPv6 の場合は、隣接マネージャが NDP からの、レイヤ 3 からレイヤ 2 へのマッピング情報を探索します。詳細については、第 3 章「IPv6 の設定」を参照してください。

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（FDM）はアクティブなスーパーバイザ上に存在し、ユニキャスト RIB や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 によってスタンバイ スーパーバイザおよびモジュール上のハードウェア転送テーブルにプログラミングされる転送情報を生成します。また、ユニキャスト FDM 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 FDM は隣接関係情報を収集し、ユニキャスト FIB でのルート更新時に、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、スーパーバイザ モジュールとスイッチング モジュール上にあり、ハードウェア転送エンジンが使用する情報を構築します。ユニキャスト FIB は、ユニキャスト FDM 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 address-family ごとに維持されます。つまり、設定された各 VRF について、IPv4 用に 1 つ、IPv6 用に 1 つ維持されます。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。

ハードウェア転送

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。

また、**show platform fib** または **show platform forwarding** コマンドを使用すると、ハードウェア転送の詳細が表示されます。

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、アクティブなスーパーバイザ上の CPU に渡されます。ソフトウェアでの切り替えが必要なパケットや終端される必要のあるパケットはすべて、スーパーバイザに渡されます。スーパーバイザは、ユニキャスト RIB および隣接マネージャから提供された情報を使用して、転送の決定を下します。モジュールは、ソフトウェア転送パスには関与しません。

ソフトウェア転送はコントロール プレーン ポリシーによって制御されます。詳細については、『*Cisco Nexus 9000 Series NX-OS Security Configuration 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。

この項では、次のトピックについて取り上げます。

IPv4 および IPv6

レイヤ 3 は、IPv4 プロトコルまたは IPv6 プロトコルを使用します。IPv6 では、ネットワーク アドレス ビット数が 32 ビット（IPv4 の場合）から 128 ビットに増やされています。詳細については、第 2 章「IPv4 の設定」または第 3 章「IPv6 の設定」を参照してください。

IP サービス

IP サービスには、DHCP クライアントおよびドメイン ネーム システム（DNS）クライアントがあります。詳細については、第 4 章「DNS の設定」を参照してください。

OSPF

Open Shortest Path First（OSPF）プロトコルは、AS 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隣接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、第 5 章「OSPFv2 の設定」を参照してください。

EIGRP

nhanced Interior Gateway Routing Protocol（EIGRP）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 はネイバーに依存し、ルートを提供します。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。詳細については、第 7 章「EIGRP の設定」を参照してください。

IS-IS

Intermediate System-to-Intermediate System（IS-IS）プロトコルは、国際標準化機構（ISO）10589 で指定されたドメイン内開放型システム間相互接続（Open System Interconnection）ダイナミック ルーティング プロトコルです。IS-IS ルーティング プロトコルはリンクステート プロトコルです。IS-IS 機能は次のとおりです。

- 階層型ルーティング
- クラスレス動作
- 新情報の高速フラッディング
- 短時間でのコンバージェンス
- 高いスケーラビリティ

詳細については、第 8 章「IS-IS 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、第 9 章「ベーシック BGP の設定」および第 10 章「Configuring Advanced BGP」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、第 11 章「RIP の設定」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、第 12 章「スタティック ルーティングの設定」を参照してください。

レイヤ 3 仮想化

仮想化を使用すると、複数の管理ドメインにわたる物理リソースを共有できます。Cisco NX-OS は、仮想ルーティングおよび転送（VRF）を含むレイヤ 3 仮想化をサポートしています。VRF では、レイヤ 3 ルーティング プロトコルを設定するための別のアドレス ドメインが提供されます。詳細については、第 13 章「レイヤ 3 仮想化の設定」を参照してください。

Route Policy Manager

Route Policy Manager は、Cisco NX-OS 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、第 15 章「Route Policy Manager の設定」を参照してください。

ポリシーベース ルーティング

ポリシーベース ルーティングは、Route Policy Manager を使用してポリシー ルート フィルタを作成します。これらのポリシー ルート フィルタでは、パケットの送信元またはパケット ヘッダーのその他フィールドに基づいて、指定されたネクスト ホップにパケットを転送できます。プロトコル タイプやポート番号に基づいてルーティングできるように、ポリシー ルートを拡張 IP アクセス リストにリンクすることができます。詳細については、第 16 章「ポリシーベース ルーティングの設定」を参照してください。

ファーストホップ冗長プロトコル（FHRP）

ホットスタンバイ ルータ プロトコル（HSRP）、仮想ルータ冗長プロトコル（VRRP）などのファースト ホップ冗長プロトコル（FHRP）を使用すると、ホストで接続の冗長性を実現できます。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。HSRP の詳細については、第 17 章「HSRP の設定」を参照してください。VRRP の詳細については、第 18 章「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、第 19 章「オブジェクト トラッキングの設定」を参照してください。

関連項目

次のシスコ マニュアルは、レイヤ 3 機能に関連するものです。

- *『Cisco Nexus 9000 Series NX-OS Multicast Routing Configuration Guide』*
- *『Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide』*
- 自律システム番号の詳細については、次のページを参照してください。
  http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_9-1/autonomous_system_numbers.html

IPv4 の設定

この章では、Cisco NX-OS デバイス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 について

デバイス上で IP を設定し、ネットワーク インターフェイスに IP アドレスを割り当てることができます。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、デバイス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。デバイス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 デバイスは、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」セクション（2-2 ページ）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。

IP 機能には、スーパーバイザ モジュールで終端する IPv4 パケットを取り扱い、また同様に、IPv4 ユニキャスト/マルチキャスト ルート ルックアップとソフトウェア アクセス コントロール リスト（ACL）の転送を含む IPv4 パケットの転送を行う役割があります。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信/受信インターフェイスも管理します。

この項では、次のトピックについて取り上げます。

複数の IPv4 アドレス

Cisco NX-OS 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、サブネット化により、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デバイスがセカンダリ 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デバイスも、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。

LPM ルーティング モード

デフォルトでは、Cisco NX-OS プログラムは階層方式でルートをプログラミングし、デバイス上での最長プレフィクス照合（LPM）が可能になります。ただし、より大量の LPM ルート エントリをサポートするために、異なるルーティング モード用にデバイスを設定できます。

次の表に、Cisco Nexus 9300 シリーズおよび 9500 シリーズ スイッチでサポートされている LPM ルーティング モードを示します。

表 2-1　Cisco Nexus 9300 シリーズ スイッチ用の LPM ルーティング モード

LPM ルーティング モード	Broadcom T2 モード	CLI コマンド
デフォルト システム ルーティング モード	3	
ALPM ルーティング モード	4	**system routing max-mode l3**

表 2-2　Cisco Nexus 9500 シリーズ スイッチ用の LPM ルーティング モード

LPM ルーティング モード	Broadcom T2 モード	CLI コマンド
デフォルト システム ルーティング モード	3（ライン カード用） 4（ファブリック モジュール用）	
最大ホスト ルーティング モード	2（ライン カード用） 3（ファブリック モジュール用）	**system routing max-mode host**
非階層ルーティング モード	3（ライン カード用） 4 および **max-l3-mode** オプション（ライン カード用）	**system routing non-hierarchical-routing** [**max-l3-mode**]
64 ビット ALPM ルーティング モード	モード 4 のサブモード（ファブリック モジュール用）	**system routing mode hierarchical 64b-alpm**

設定の詳細については、「IPv4 の設定」セクション（2-8 ページ）を参照してください。

サポートされるルーティング モードの詳細については、『Layer 3 Forwarding Modes on Cisco Nexus 9500, 9300, 3164, and 3200 Platform Switches』を参照してください。

アドレス解決プロトコル

ネットワーキング デバイスおよびレイヤ 3 スイッチは ARP を使用して、IP（ネットワーク層）アドレスを物理（Media Access Control（MAC）レイヤ）アドレスにマッピングし、IP パケットがネットワーク上に送信されるようにします。デバイスは、他のデバイスにパケットを送信する前に自身の ARP キャッシュを調べて、MAC アドレスまたは対応する宛先デバイスの IP アドレスがないかを確認します。エントリがまったくない場合、送信元のデバイスは、ネットワーク上の全デバイスにブロードキャスト メッセージを送信します。

各デバイスは、問い合わせられた IP アドレスを自身のアドレスと比較します。一致する IP アドレスを持つデバイスだけが、デバイスの MAC アドレスを含むパケットとともにデータを送信したデバイスに返信します。送信元デバイスは、あとで参照できるよう、宛先デバイスの MAC アドレスをその ARP テーブルに追加し、データリンク ヘッダーおよびトレーラを作成してパケットをカプセル化し、データの転送へと進みます。図 2-1 ARP ブロードキャストと応答処理を示します。

図 2-1　ARP 処理

宛先デバイスが、別のデバイスを挟んだリモート ネットワーク上にあるときは、同じ処理が行われますが、データを送信するデバイスが、デフォルト ゲートウェイの MAC アドレスを求める ARP 要求を送信する点が異なります。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。宛先デバイスのネットワーク上のデバイスは、ARP を使用して宛先デバイスの 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。

デフォルトでシステム定義された CoPP ポリシー レートは、スーパーバイザ モジュールにバインドされた ARP ブロードキャスト パケットを制限します。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、無駄に使用されるネットワーク リソースが制限されます。IP アドレスの MAC アドレスへのマッピングは、ネットワーク間でパケットが送信されるたびに、ネットワーク上の各ホップ（デバイス）で行われるため、ネットワークのパフォーマンスに影響する場合があります。

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内に保存されるため、パケットが送信されるたびに同じアドレスにブロードキャストするための貴重なネットワーク リソースの使用が最小限に抑えられます。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デバイスは、アドレスのブロードキャストに従ってアドレス テーブルを更新します。

ARP キャッシュのスタティック エントリおよびダイナミック エントリ

スタティック ルーティングは、手動で各デバイスの各インターフェイスに対応する 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 アドレスを設定する必要があります。スタティック ルーティングでは、ルート テーブルを維持するために、より多くの処理が必要です。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、ネットワーク上のデバイスが相互にルーティング テーブル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。ブリッジは MAC アドレスだけを使用する独自のアドレス テーブルを作成します。デバイスが IP アドレスおよび対応する MAC アドレスの両方を含む ARP キャッシュを持っています。

パッシブ ハブは、ネットワーク内の他のデバイスを物理的に接続する集中接続デバイスです。パッシブ ハブはそのすべてのポートでデバイスにメッセージを送信し、レイヤ 1 で動作しますが、アドレス テーブルを保持しません。

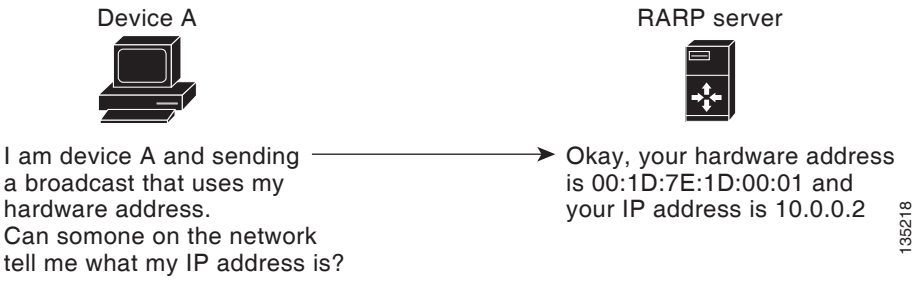
レイヤ 2 スイッチは、デバイス上のどのポートがそのポートだけに送信されたメッセージを受信するかを決定します。ただし、レイヤ 3 スイッチは、ARP キャッシュ（テーブル）を作成するデバイスです。

Reverse ARP

RFC 903 で規定された Reverse ARP（RARP）は ARP と同様に機能しますが、RARP 要求パケットが 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。図 2-2 に、RARP のしくみを示します。

図 2-2　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、多数の物理ネットワークを含む大規模なインターネットワークの場合は、すべてのセグメント上に冗長性のための追加サーバとともに 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。

プロキシ ARP

プロキシ ARP を使用すると、物理的に 1 つのネットワーク上に存在するデバイスが、論理的に、同じデバイス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一部として表示されます。プロキシ ARP で、プライベート ネットワーク上のパブリック IP アドレスを持つデバイスをルータの背後に隠すと同時に、このデバイスを、ルータの前のパブリック ネットワーク上に表示できます。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、サブネット上のデバイスは、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。

複数のデバイスが同じデータリンク層のネットワークでなく、同じ IP ネットワーク内にある場合、これらのデバイスは相互に、ローカル ネットワーク上にあるかのようにデータを送信しようとします。ただし、これらのデバイスを隔てるルータは、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。

デバイスでプロキシ 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。デバイスは、ブロードキャストの宛先であるリモートの宛先であるかのように、自身の MAC アドレスをリモートの宛先の IP アドレスに関連付ける ARP 応答で応答します。ローカル デバイスは、自身が宛先に直接、接続されていると認識していますが、実際には、そのパケットは、ローカル デバイスによりローカル サブネットワークから宛先のサブネットワークへと転送されています。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 プロキシ ARP を使用して、通常はルーティングが不要なサブネット内の IP アドレスを求める ARP 要求に対して、デバイスが応答できるようにすることができます。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、ホストが接続されているデバイスの設定により意図的に、ホストの直接通信が禁止されているサブネットだけで使用してください。

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。

パス MTU ディスカバリ

パス最大伝送ユニット（MTU）ディスカバリは、TCP 接続のエンドポイント間のネットワーク内で使用可能な帯域幅の使用を最大化するための方法です。これは RFC 1191 で規定されています。この機能を有効または無効にしても、既存の接続に影響しません。

ICMP

Internet Control Message Protocol（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 機能がイネーブルであるインターフェイス上ではディセーブルにされています。

仮想化のサポート

IPv4 は、仮想ルーティング/転送（VRF）インスタンスをサポートします。

IPv4 のライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	IP にはライセンスは不要です。ライセンス パッケージに含まれていない機能は nx-os イメージにバンドルされており、無料で提供されます。Cisco NX-OS のライセンス スキームの詳細は、*『Cisco NX-OS Licensing Guide』*を参照してください。

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。

デフォルト設定値

表 2-3 に、IP パラメータのデフォルト設定を示します。

表 2-3　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
『Proxy ARP』	ディセーブル

IPv4 の設定

この項では、次のトピックについて取り上げます。

- IPv4 アドレス指定の設定（2-9 ページ）
- 複数の IP アドレスの設定（2-10 ページ）
- 最大ホスト ルーティング モードの設定（Cisco Nexus 9500 シリーズ スイッチのみ）（2-10 ページ）
- 非階層ルーティング モードの設定（Cisco Nexus 9500 シリーズ スイッチのみ）（2-12 ページ）
- 64 ビット ALPM ルーティング モードの設定（Cisco Nexus 9500 シリーズ スイッチのみ）（2-13 ページ）
- ALPM ルーティング モードの設定（Cisco Nexus 9300 シリーズ スイッチのみ）（2-14 ページ）
- スタティック ARP エントリの設定（2-15 ページ）
- プロキシ ARP の設定（2-16 ページ）
- ローカル プロキシ ARP の設定（2-17 ページ）
- Gratuitous ARP の設定（2-18 ページ）
- パス MTU ディスカバリの設定（2-18 ページ）
- IP ダイレクト ブロードキャストの設定（2-19 ページ）
- IP 収集スロットルの設定（2-20 ページ）
- ハードウェア IP 収集スロットルの最大数の設定（2-20 ページ）
- ハードウェア IP 収集スロットルのタイムアウトの設定（2-21 ページ）
- ICMP 送信元 IP フィールドのインターフェイス IP アドレスの設定（2-22 ページ）

（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. **ip address** *ip-address/length*
4. （任意）**show ip interface**
5. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 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	**ip address** *ip-address/length* [secondary] **例**: `switch(config-if)# ip address 192.168.1.1 255.0.0.0`	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 4	**show ip interface** **例**: `switch(config-if)# show ip interface`	（任意）IPv4 に設定されたインターフェイスを表示します。
ステップ 5	**copy running-config startup-config** **例**: `switch(config-if)# copy running-config startup-config`	（任意）実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

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だけ追加できます。

手順の概要

1. **configure terminal**
2. **interface ethernet** *number*
3. **ip address** *ip-address/length*
4. （任意）**show ip interface**
5. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 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	**ip address** *ip-address/length* [secondary] **例**: `switch(config-if)# ip address 192.168.1.1 255.0.0.0 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 4	**show ip interface** **例**: `switch(config-if)# show ip interface`	(任意)IPv4 に設定されたインターフェイスを表示します。
ステップ 5	**copy running-config startup-config** **例**: `switch(config-if)# copy running-config startup-config`	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

最大ホスト ルーティング モードの設定（Cisco Nexus 9500 シリーズ スイッチのみ）

デフォルトでは、Cisco NX-OS は階層方式で（モード 4 になるように設定されたファブリック モジュールとモード 3 になるように設定されたラインカード モジュールで）ルートをプログラミングし、デバイス上での最長プレフィクス照合（LPM）とホスト スケールが可能になります。

デフォルトの LPM およびホスト スケールを変更してシステム内のホストをさらにプログラミングできます。これは、ノードをレイヤ 2 ～ レイヤ 3 の境界ノードとして位置付けるときに必要になる場合があります。

（注）　LPM テーブルのエントリをさらに拡大したい場合は、「非階層ルーティング モードの設定（Cisco Nexus 9500 シリーズ スイッチのみ）」セクション（2-12 ページ）を参照して、ラインカード上のレイヤ 3 IPv4 および IPv6 ルートすべてをプログラミングしてファブリック モジュール上のルートはそのままにするようデバイスを設定します。

（注）　この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注）　最大ホスト ルーティング モードのスケール数については、『*Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*』を参照してください。

手順の概要

1. **configure terminal**
2. **[no] system routing max-mode host**
3. （任意）**show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

	コマンド	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	`[no] system routing max-mode host`
	
	例:
	`switch(config)# system routing max-mode host`
		ラインカードを Broadcom T2 モード 2 に、ファブリック モジュールを Broadcom T2 モード 3 にして、サポートされるホスト数を増やします。

ステップ 3	`show forwarding route summary`
	
	例:
	`switch(config)# show forwarding route summary`
		（任意）LPM ルーティング モードを表示します。

ステップ 4	`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。

ステップ 5	`reload`
	
	例:
	`switch(config)# reload`
		デバイス全体がリブートします。

非階層ルーティング モードの設定（Cisco Nexus 9500 シリーズ スイッチのみ）

ホストの規模が小さい場合（純粋なレイヤ 3 配置の場合など）、コンバージェンス パフォーマンスを向上させるために、ラインカードの最長プレフィクス照合（LPM）のルートをプログラミングすることを推奨します。そうすることによって、ラインカードのルートおよびホストがプログラミングされ、ファブリック モジュールのルートはプログラミングされません。

（注）　この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

手順の概要

1. **configure terminal**
2. **[no] system routing non-hierarchical-routing [max-l3-mode]**
3. （任意）**show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

	コマンド	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	**[no] system routing non-hierarchical-routing [max-l3-mode]** **例：** `switch(config)# system routing non-hierarchical-routing max-l3-mode`	ラインカードを Broadcom T2 モード 3（または **max-l3-mode** オプションを使用している場合は Broadcom T2 モード 4）にし、より大きな LPM スケールをサポートします。その結果、IPv4 および IPv6 ルートのすべてが、ファブリック モジュールではなくラインカードでプログラミングされます。

ステップ 3	**show forwarding route summary**
	
	例：
	`switch(config)# show forwarding route summary`
	`Mode 3:`
	`120K IPv4 Host table`
	`16k LPM table (> 65 < 127 1k entry reserved)`
	`Mode 4:`
	`16k V4 host/4k V6 host`
	`128k v4 LPM/20K V6 LPM`
		（任意）LPM モードを表示します。

	コマンド	目的

ステップ 4	`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。

ステップ 5	`reload`
	
	例:
	`switch(config)# reload`
		デバイス全体がリブートします。

64 ビット ALPM ルーティング モードの設定（Cisco Nexus 9500 シリーズ スイッチのみ）

64 ビットのアルゴリズム最長プレフィクス照合（ALPM）を使用して IPv4 および IPv6 ルート テーブル エントリを管理できます。64 ビット ALPM ルーティング モードでは、デバイスに保存できるルート エントリの数が大幅に増加します。このモードでは、次のいずれかをプログラミングできます。

- 80,000 個の IPv6 エントリ、IPv4 エントリなし
- IPv6 エントリなし、128,000 個の IPv4 エントリ
- *x* 個の IPv6 エントリと *y* 個の IPv4 エントリ（ただし $2x + y <= 128,000$）

（注） この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注） 64 ビット ALPM ルーティング モードのスケール数については、『*Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*』を参照してください。

手順の概要

1. **configure terminal**
2. **[no] system routing mode hierarchical 64b-alpm**
3. （任意）**show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

	コマンド	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	**[no] system routing mode hierarchical 64b-alpm** **例:** `switch(config)# system routing mode hierarchical 64b-alpm`	マスク長が 64 以下のすべての IPv4 および IPv6 LPM ルートはファブリック モジュールにプログラミングされます。IPv4 および IPv6 のすべてのホスト ルート、およびマスク長が 65 ～ 127 のすべての LPM ルートはライン カードにプログラミングされます。
ステップ 3	**show forwarding route summary** **例:** `switch(config)# show forwarding route summary`	（任意）LPM モードを表示します。
ステップ 4	**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。
ステップ 5	**reload** **例:** `switch(config)# reload`	デバイス全体がリブートします。

ALPM ルーティング モードの設定（Cisco Nexus 9300 シリーズ スイッチのみ）

Cisco Nexus 9300シリーズ スイッチは、より大量の LPM ルート エントリをサポートするように設定できます。

（注） この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注） ALPM ルーティング モードのスケール数については、『*Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*』を参照してください。

手順の概要

1. **configure terminal**
2. **[no] system routing max-mode l3**
3. （任意）**show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

	コマンド	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	**[no] system routing max-mode l3** **例**: `switch(config)# system routing max-mode l3`	デバイスを Broadcom T2 モード 4 にして、より大きな LPM スケールをサポートします。
ステップ 3	**show forwarding route summary** **例**: `switch(config)# show forwarding route summary`	(任意)LPM モードを表示します。
ステップ 4	**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。
ステップ 5	**reload** **例**: `switch(config)# reload`	デバイス全体がリブートします。

スタティック ARP エントリの設定

デバイス上でスタティック ARP エントリを設定して、IP アドレスをスタティック マルチキャスト MAC アドレスを含む MAC ハードウェア アドレスにマッピングできます。

手順の概要

1. **configure terminal**
2. **interface ethernet** *number*
3. **ip arp** *ipaddr mac_addr*
4. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 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	**ip arp** *ipaddr mac_addr* **例:** switch(config-if)# ip arp 192.168.1.1 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 4	**copy running-config startup-config** **例:** switch(config-if)# copy running-config startup-config	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

プロキシ ARP の設定

デバイス上でプロキシ ARP を設定して、他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 アドレスを決定できます。

手順の概要

1. **configure terminal**
2. **interface ethernet** *number*
3. **ip proxy-arp**
4. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 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	目的
ステップ 3	**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 4	**copy running-config startup-config** **例**: `switch(config-if)# copy running-config startup-config`	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

ローカル プロキシ ARP の設定

デバイス上でローカル 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. **ip local-proxy-arp**
4. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 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	**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 4	**copy running-config startup-config** **例**: `switch(config-if)# copy running-config startup-config`	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

Gratuitous 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. **ip arp gratuitous** {**request** | **update**}
4. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 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	**ip arp gratuitous** {**request** | **update**} **例:** `switch(config-if)# ip arp gratuitous request`	インターフェイス上で Gratuitous ARP 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。
ステップ 4	**copy running-config startup-config** **例:** `switch(config-if)# copy running-config startup-config`	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

パス MTU ディスカバリの設定

パス MTU ディスカバリを設定できます。

手順の概要

1. **configure terminal**
2. **ip tcp path-mtu-discovery**
3. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	**ip tcp path-mtu-discovery** **例**: `switch(config)# ip tcp path-mtu-discovery`	パス MTU ディスカバリをイネーブルにします。
ステップ 3	**copy running-config startup-config** **例**: `switch(config)# copy running-config startup-config`	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

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デバイスは、そのサブネット上のホストを宛先とするユニキャスト IP パケットを転送する場合と同じ方法で 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デバイスに到着すると、そのパケットはその宛先サブネット上でブロードキャストされます。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でブロードキャストされます。アクセス リストを通じて渡すこれらパケットのみがサブネット上でブロードキャストされるように、IP アクセス リストを通じてこれらブロードキャストを任意でフィルタリングすることができます。

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 [*acl*]	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 アクセス リスト上のこれらのブロードキャストを任意でフィルタリングできます。

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。

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	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	**[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	**copy running-config startup-config** **例:** `switch(config)# copy running-config startup-config`	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

ハードウェア IP 収集スロットルの最大数の設定

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. **[no] hardware ip glean throttle maximum** *count*
3. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	[**no**] **hardware ip glean throttle maximum** *count* **例:** `switch(config)# hardware ip glean throttle maximum 2134`	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。
ステップ 3	**copy running-config startup-config** **例:** `switch(config)# copy running-config startup-config`	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

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. [**no**] **hardware ip glean throttle maximum timeout** *timeout*
3. (任意)**copy running-config startup-config**

手順の詳細

	コマンド	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	[**no**] **hardware ip glean throttle maximum timeout** *timeout* **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。 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 3	**copy running-config startup-config** **例:** `switch(config)# copy running-config startup-config`	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

ICMP 送信元 IP フィールドのインターフェイス IP アドレスの設定

ICMP エラー メッセージを処理するように ICMP ソース IP フィールドのインターフェイス IP アドレスを設定できます。

手順の概要

1. **configure terminal**
2. [**no**] **ip source** {**ethernet** *slot*/*port* | **loopback** *number* | **port-channel** *number*} **icmp-errors**
3. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	[**no**] **ip source** {**ethernet** *slot*/*port* | **loopback** *number* | **port-channel** *number*} **icmp-errors** **例**: `switch(config)# ip source loopback 0 icmp-errors`	ICMP 送信元 IP フィールドのインターフェイス IP アドレスを設定し、ICMP エラー メッセージをルーティングします。
ステップ 3	**copy running-config startup-config** **例**: `switch(config)# copy running-config startup-config`	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

IPv4 設定の確認

IPv4 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 adjacency	隣接関係テーブルを表示します。
show ip adjacency summary	スロットル隣接のサマリーを表示します。
show ip arp	ARP テーブルを表示します。
show ip arp summary	スロットル隣接数のサマリーを表示します。
show ip interface	IP 関連の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。

IPv6 の設定

この章では、Cisco NX-OS デバイス上でのインターネット プロトコル バージョン 6（IPv6）（アドレス指定を含む）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。

About IPv6

IPv6 は、IPv4 の後継として設計されており、ネットワーク アドレス ビット数が 32 ビット（IPv4 の場合）から 128 ビットに増やされています。IPv6 は IPv4 に基づいていますが、アドレス空間が大幅に拡大されており、メイン ヘッダーと拡張ヘッダーの簡素化など、その他の機能強化が含まれています。

拡大された IPv6 アドレス空間により、ネットワークのスケーラビリティが可能となり、グローバルな到達可能性が提供されます。簡素化された IPv6 パケット ヘッダー形式により、パケットの処理効率が向上しています。柔軟性の高い IPv6 アドレス空間により、プライベート アドレスの必要性と、プライベート（グローバルに一意ではない）アドレスを限られた数のパブリック アドレスに変換するネットワーク アドレス変換（NAT）の使用が削減されます。IPv6 を使用すると、ネットワークの境界にある境界ルータによる特別な処理を必要としない新しいアプリケーション プロトコルがイネーブルになります。

プレフィックス集約、簡易ネットワーク再番号割り当て、IPv6 サイト マルチホーミング機能などの IPv6 機能により、さらに効率的にルーティングが行われます。IPv6 は、Routing Information Protocol（RIP）、Integrated Intermediate System-to-Intermediate System（IS-IS）、IPv6 向け Open Shortest Path First（OSPF）、マルチプロトコル Border Gateway Protocol（BGP）をサポートしています。

この項では、次のトピックについて取り上げます。

IPv6 アドレス フォーマット

IPv6 アドレスは 128 ビットつまり 16 バイトです。このアドレスは、x:x:x:x:x:x:x:x のように、コロン（:）で区切られた 16 ビット 16 進数のブロック 8 つに分かれています。次に、IPv6 アドレスの例を 2 つ示します。

```
2001:0DB8:7654:3210:FEDC:BA98:7654:3210
2001:0DB8:0:0:8:800:200C:417A
```

IPv6 アドレスの中には、連続するゼロが含まれます。IPv6 アドレスの先頭、中間、または末尾で、この連続するゼロの代わりに 2 つのコロン（::）を使用できます。表 3-1 圧縮された IPv6 アドレス フォーマットの一覧です。

（注）　IPv6 アドレスでは、アドレス中で最も長く連続するゼロの代わりに、2 つのコロン（::）を 1 度だけ使用できます。

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

IPv6 アドレス中の 16 進数の文字の大文字と小文字は区別されません。

表 3-1　圧縮された IPv6 アドレス フォーマット

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

ノードは、表 3-1にあるループバック アドレスを使用して、IPv6 パケットを自分宛てに送信できます。IPv6 のループバック アドレスは、IPv4 のループバック アドレスと同じです。詳細については、第 1 章「概要」を参照してください。

（注）　IPv6 ループバック アドレスは、物理インターフェイスには割り当てられません。送信元または宛先のアドレスとして IPv6 ループバック アドレスを含むパケットは、そのパケットを作成したノードの外には転送できません。IPv6 ルータは、IPv6 ループバック アドレスを送信元アドレスまたは宛先アドレスとするパケットを転送しません。

（注）　IPv6 未指定アドレスは、インターフェイスには割り当てられません。未指定 IPv6 アドレスは、IPv6 パケット内の宛先アドレスまたは IPv6 ルーティング ヘッダーとして使用しないでください。

IPv6 プレフィックスは、RFC 2373 で規定された形式です。この形式では、IPv6 アドレスが、コロンに囲まれた 16 ビット値を使用した 16 進数で指定されています。プレフィックス長は、アドレスの高次の連続ビットのうち、何個がプレフィックス（アドレスのネットワーク部分）を構成しているかを指定する 10 進数値です。たとえば、2001:0DB8:8086:6502::/32 は有効な IPv6 プレフィックスです。

IPv6 ユニキャスト アドレス

IPv6 ユニキャスト アドレスは、1 つのノード上の 1 つのインターフェイスの ID です。ユニキャスト アドレスに送信されたパケットは、そのアドレスが示すインターフェイスに配信されます。この項では、次のトピックについて取り上げます。

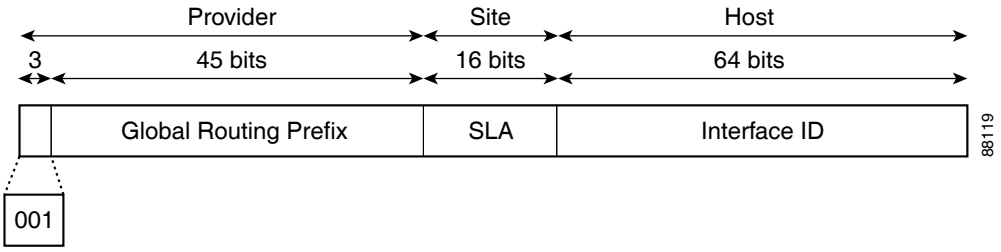
- 集約可能グローバル アドレス（3-3 ページ）
- リンクローカル アドレス（3-5 ページ）
- IPv4 互換 IPv6 アドレス（3-5 ページ）
- 一意のローカル アドレス（3-6 ページ）
- サイトローカル アドレス（3-7 ページ）

集約可能グローバル アドレス

集約可能グローバル アドレスは、集約可能なグローバル ユニキャスト プレフィックスによる IPv6 アドレスです。集約可能グローバル ユニキャスト アドレスの構造により、グローバル ルーティング テーブル内のルーティング テーブル エントリ数を制限するルーティング プレフィックスの厳密な集約が可能になります。集約可能グローバル アドレスは、組織を上に向かって、最終的にインターネット サービス プロバイダー（ISP）まで集約されるリンク上で使用されます。

集約可能なグローバル IPv6 アドレスは、グローバル ルーティング プレフィックス、サブネット ID、およびインターフェイス ID により定義されます。バイナリ 000 で始まるアドレスを除き、グローバル ユニキャスト アドレスはすべて 64 ビット インターフェイス ID を持ちます。IPv6 グローバル ユニキャスト アドレスの割り当てには、バイナリ値 001（2000::/3）から始まるアドレスの範囲が使用されます。図 3-1 集約可能グローバル アドレスの構造を示します。

図 3-1　集約可能グローバル アドレスのフォーマット



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

集約可能なグローバル アドレスは、48 ビット グローバル ルーティング プレフィックスと、16 ビット サブネット ID または Site-Level Aggregator（SLA）で構成されます。IPv6 集約可能なグローバル ユニキャスト アドレス フォーマット文書（RFC 2374）では、グローバル ルーティング プレフィックスには、Top-Level Aggregator（TLA）および Next-Level Aggregator（NLA）という他の 2 つの階層構造のフィールドが含まれるとされていました。TLS フィールドおよび NLA フィールドはポリシーベースであるため、IETF は、これらのフィールドを RFC から削除することを決定しました。この変更以前に展開された既存の IPv6 ネットワークの中には、依然として、古いアーキテクチャ上のネットワークを使用しているものもあります。

個々の組織では、16 ビット サブネット フィールドであるサブネット ID を使用して、ローカル アドレス指定階層構造を作成したり、サブネットを識別したりできます。サブネット ID は IPv4 でのサブネットに似ていますが、IPv6 サブネット ID を持つ組織では最大 65,535 個のサブネットをサポートできるという点が異なります。

インターフェイス ID で、リンク上のインターフェイスが識別されます。インターフェイス ID は、リンク上では一意です。多くの場合、インターフェイス ID は、インターフェイスのリンク層アドレスと同じか、リンク層アドレスに基づいています。集約可能なグローバル ユニキャストやその他の IPv6 アドレス タイプで使用されるインターフェイス ID は 64 ビットであり、形式は変更済み EUI-64 フォーマットです。

インターフェイス ID は、次のいずれかに該当する変更済みの EUI-64 フォーマットです。

- すべての IEEE 802 インターフェイス タイプ（イーサネット、およびファイバ分散データ インターフェイスなど）の場合は、最初の 3 オクテット（24 ビット）がそのインターフェイスの 48 ビット リンク層アドレス（MAC アドレス）の Organizationally Unique Identifier（OUI）、4 番めと 5 番めのオクテット（16 ビット）が FFFE の固定 16 進数値、そして、最後の 3 オクテット（24 ビット）が MAC アドレスの最後の 3 オクテットです。最初のオクテットの 7 番めのビットである Universal/Local（U/L）ビットの値は 0 または 1 です。ゼロはローカルに管理されている ID を表し、1 はグローバルに一意の IPv6 インターフェイス ID を表します。
- その他のすべてのインターフェイス タイプ（シリアル、ループバック、ATM、フレーム リレー種別など）の場合、インターフェイス ID は IEEE 802 インターフェイス タイプのインターフェイス ID に似ていますが、ルータの MAC アドレス プールからの最初の MAC アドレスが ID として使用される点が異なります（インターフェイスが MAC アドレスを持たないため）。

（注） PPP（ポイントツーポイント プロトコル）を使用するインターフェイスの場合は、接続の両端のインターフェイスが同じ MAC アドレスを持つため、接続の両端のインターフェイス ID が、両方の ID が一意となるまでネゴシエートされます（ランダムに選択され、必要に応じて再構築されます）。ルータの最初の MAC アドレスが、PPP を使用するインターフェイスの ID として使用されます。

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

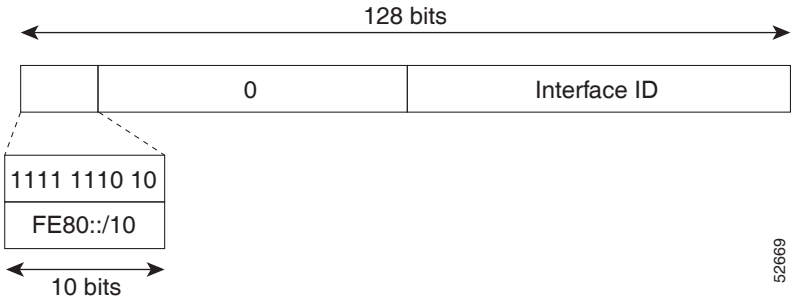
1. ルータに MAC アドレスが（ルータの MAC アドレス プールから）照会されます。
2. 使用可能な MAC アドレスがルータにない場合は、ルータのシリアル番号を使用してリンクローカル アドレスが作成されます。
3. リンクローカル アドレスの作成にルータのシリアル番号を使用できない場合、ルータは MD5 ハッシュを使用して、ルータのホスト名からルータの MAC アドレスを決定します。

リンクローカル アドレス

リンクローカル アドレスは、リンクローカル プレフィックス FE80::/10（1111 1110 10）と変更された EUI-64 形式のインターフェイス識別子を使用するすべてのインターフェイスを自動的に設定できる IPv6 ユニキャスト アドレスです。ネイバー探索プロトコル（NDP）およびステートレス自動設定プロセスでは、リンクローカル アドレスが使用されます。ローカル リンク上のノードは、リンクローカル アドレスを使用して通信できます。ノードの通信にグローバルに一意のアドレスは不要です。図 3-2 リンクローカル アドレスの構造を示します。

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

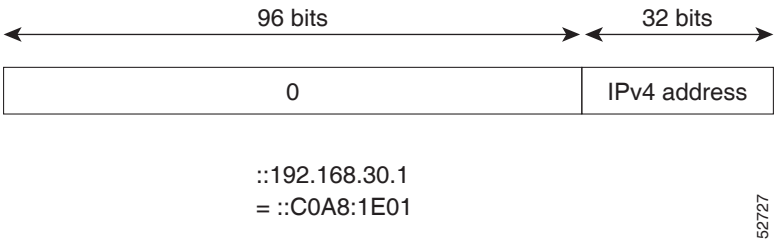
図 3-2　　リンクローカル アドレス フォーマット



IPv4 互換 IPv6 アドレス

IPv4 互換 IPv6 アドレスは、アドレスの上位 96 ビットがゼロであり、アドレスの下位 32 ビットが IPv4 アドレスである IPv6 ユニキャスト アドレスです。IPv4 互換 IPv6 アドレスの形式は、0:0:0:0:0:0:A.B.C.D または ::A.B.C.D です。IPv4 互換 IPv6 アドレスの 128 ビット全体はノードの IPv6 アドレスとして使用され、下位 32 ビットに埋め込まれた IPv4 アドレスはノードの IPv4 アドレスとして使用されます。IPv4 互換 IPv6 アドレスは、IPv4 と IPv6 の両方のプロトコル スタックをサポートするノードに割り当てられ、自動トンネルで使用されます。図 3-3 IPv4 互換 IPv6 アドレスの構造と、許容されるいくつかのアドレス形式を示します。

図 3-3　IPv4 互換 IPv6 アドレスのフォーマット



一意のローカル アドレス

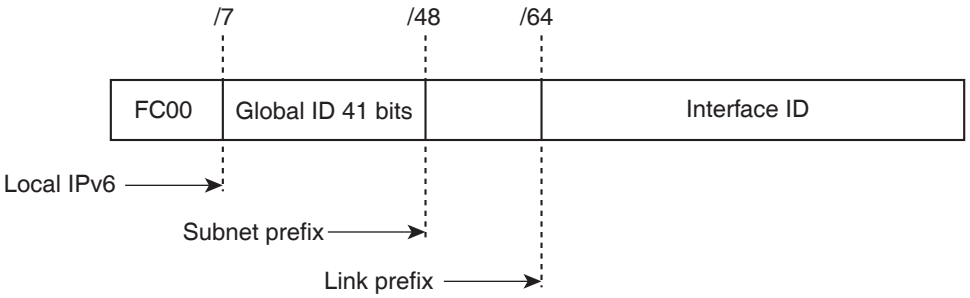
一意のローカル アドレスは、グローバルに一意であり、ローカル通信を目的とした IPv6 ユニキャスト アドレスです。グローバルなインターネット上でのルーティングには対応しておらず、サイトなどの限られたエリア内だけでルーティング可能です。限られた複数のサイト間もルーティングできる場合もあります。アプリケーションは、一意のローカル アドレスをグローバル スコープのアドレスのように扱うことができます。

一意のローカル アドレスには、次の特性があります。

- グローバルに一意のプレフィックスを持っている（一意である可能性が大）。
- 既知のプレフィックスがあるため、サイト境界で簡単にフィルタリングできる。
- アドレス競合を発生させたり、これらのプレフィックスを使用するインターフェイスのリナンバリングを必要としたりすることなく、サイトを結合またはプライベートに相互接続できる。
- ISP に依存せず、永続的または断続的なインターネット接続がなくてもサイト内での通信に使用できる。
- ルーティングやドメイン ネーム サーバ（DNS）を通して誤ってサイト外に漏れても、他のどのアドレスとも競合しない。

図 3-4 に、一意のローカル アドレスの構造を示します。

図 3-4　一意のローカル アドレスの構造



- Prefix — FC00::/7 prefix to identify local IPv6 unicast addresses.
- Global ID — 41-bit global identifier used to create a globally unique prefix.
- Subnet ID — 16-bit subnet ID is an identifier of a subnet within the site.
- Interface ID — 64-bit ID

サイトローカル アドレス

RFC 3879 によりサイトローカル アドレスの使用が廃止されたため、プライベート IPv6 アドレスの設定時には、RFC 4193 で推奨されるユニーク ローカル アドレス（UCA）を使用する必要があります。

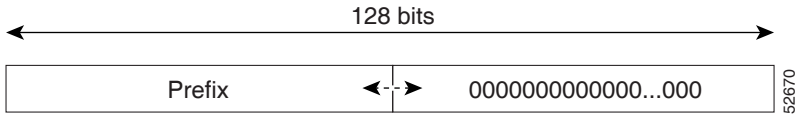
IPv6 エニーキャスト アドレス

エニーキャスト アドレスとは、異なるノードに属するインターフェイス一式に割り当てられたアドレスです。エニーキャスト アドレスに送信されたパケットは、使用しているルーティング プロトコルの定義に従って、そのエニーキャスト アドレスが示す最も近いインターフェイスに送信されます。エニーキャスト アドレスは、ユニキャスト アドレス空間から割り当てられるため、その構文ではユニキャスト アドレスと区別できません。ユニキャスト アドレスを複数のインターフェイスに割り当てると、ユニキャスト アドレスがエニーキャスト アドレスとなります。エニーキャスト アドレスが割り当てられたノードは、アドレスがエニーキャスト アドレスであることを認識できるよう、設定する必要があります。

（注）　エニーキャスト アドレスを使用できるのは、ルータだけです。ホストはエニーキャスト アドレスを使用できません。エニーキャスト アドレスは、IPv6 パケットの送信元アドレスには使用できません。

図 3-5 に、サブネット ルータ エニーキャスト アドレスの形式を示します。アドレスには、連続するゼロで連結されたプレフィックス（インターフェイス ID）があります。サブネット ルータ エニーキャスト アドレスを使用すると、サブネット ルータ エニーキャスト アドレスのプレフィックスが示すリンク上のルータに到達できます。

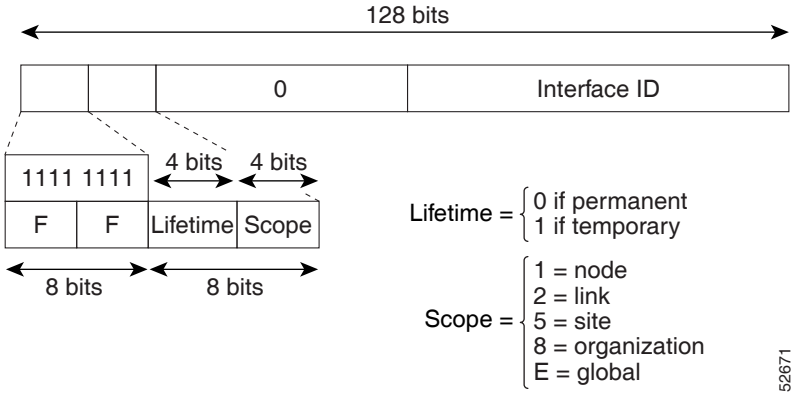
図 3-5　サブネット ルータ エニーキャスト アドレスのフォーマット



IPv6 マルチキャスト アドレス

IPv6 マルチキャスト アドレスは、FF00::/8（1111 1111）というプレフィックスを持つ IPv6 アドレスです。IPv6 マルチキャスト アドレスは、異なるノードに属するインターフェイス一式の ID です。マルチキャスト アドレスに送信されたパケットは、マルチキャスト アドレスが示すすべてのインターフェイスに配信されます。プレフィックスに続く 2 番めのオクテットで、マルチキャスト アドレスのライフタイムとスコープが定義されます。永久マルチキャスト アドレスはライフタイム パラメータが 0 に等しく、一時マルチキャスト アドレスのライフタイム パラメータは 1 に等しくなっています。ノード、リンク、サイト、または組織のスコープ、またはグローバル スコープを持つマルチキャスト アドレスのスコープ パラメータはそれぞれ、1、2、5、8、または E です。たとえば、プレフィックスが FF02::/16 のマルチキャスト アドレスは、リンク スコープを持つ永続マルチキャスト アドレスです。図 3-6 に、IPv6 マルチキャスト アドレスの形式を示します。

図 3-6　IPv6 マルチキャスト アドレス形式



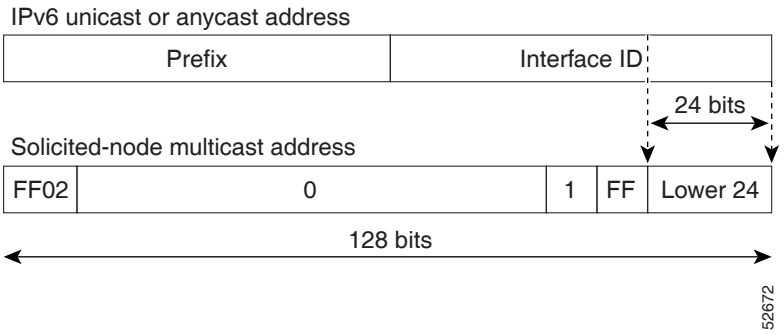
IPv6 ノード（ホストとルータ）は、（受信パケットの宛先となる）次のマルチキャスト グループに加入する必要があります。

- 全ノード マルチキャスト グループ FF02:0:0:0:0:0:0:1（スコープはリンクローカル）
- 割り当てられたユニキャスト アドレスおよびエニーキャスト アドレスごとの送信要求ノード マルチキャスト グループ FF02:0:0:0:0:1:FF00:0000/104

IPv6 ルータは、全ルータ マルチキャスト グループ FF02:0:0:0:0:0:0:2（スコープはリンクローカル）にも加入する必要があります。

送信要求ノード マルチキャスト アドレスは、IPv6 ユニキャスト アドレスまたはエニーキャスト アドレスに対応するマルチキャスト グループです。IPv6 ノードは、割り当てられているユニキャスト アドレスおよびエニーキャスト アドレスごとに、関連付けられた送信要求ノード マルチキャスト グループに加入する必要があります。IPv6 送信要求ノード マルチキャスト アドレスには、対応する IPv6 ユニキャスト アドレスまたは IPv6 エニーキャスト アドレスの下位 24 ビットに連結されたプレフィックス FF02:0:0:0:0:1:FF00:0000/104 があります（図 3-7 を参照）。たとえば、IPv6 アドレス 2037::01:800:200E:8C6C に対応する送信要求ノード マルチキャスト アドレスは FF02::1:FF0E:8C6C です。送信要求ノード アドレスは、ネイバー送信要求メッセージで使用されます。

図 3-7　IPv6 送信要求ノード マルチキャスト アドレスのフォーマット

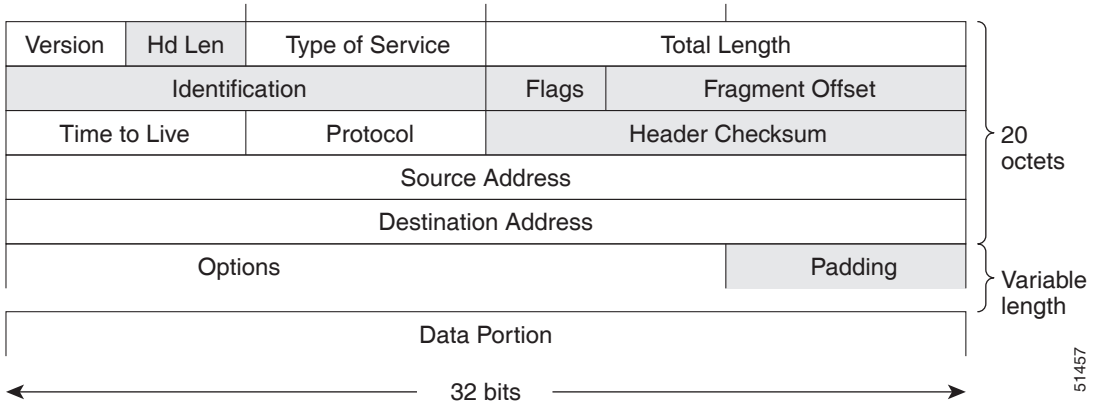


（注）　IPv6 にはブロードキャスト アドレスはありません。ブロードキャスト アドレスの代わりに IPv6 マルチキャスト アドレスが使用されます。

IPv4 パケット ヘッダー

基本 IPv4 パケット ヘッダーには、合計サイズが 20 オクテット（160 ビット）の 12 のフィールドがあります（図 3-8 を参照）。この 12 個のフィールドのあとにはオプション フィールドが、さらにそのあとに、通常はトランスポート レイヤ パケットであるデータ部分が続く場合があります。可変長のオプション フィールドは、IPv4 パケット ヘッダーの合計サイズに加算されます。IPv4 パケット ヘッダーのグレーの部分のフィールドは、IPv6 パケット ヘッダーに含まれません

図 3-8　IPv4 パケット ヘッダーのフォーマット



簡易 IPv6 パケット ヘッダー

基本 IPv6 パケット ヘッダーには、合計サイズが 40 オクテット（320 ビット）の 8 つのフィールドがあります（図 3-9 を参照）。フラグメンテーションはパケットの送信元により処理され、データリンク層のチェックサムとトランスポート層が使用されます。ユーザ データグラム プロトコル（UDP）チェックサムにより、内部パケットと基本 IPv6 パケット ヘッダーの整合性がチェックされ、オプション フィールドが 64 ビットに揃えられるため、IPv6 パケットの処理が容易になります。

表 3-2 は、基本 IPv6 パケット ヘッダー内のフィールドの一覧です。

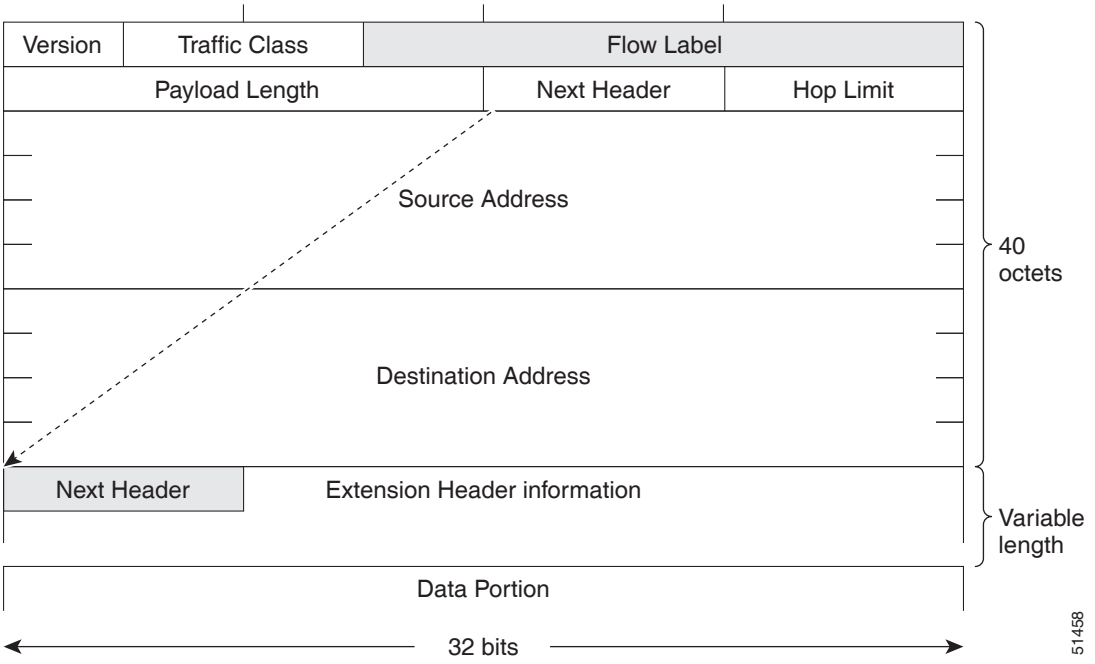
表 3-2　基本 IPv6 パケット ヘッダー フィールド

フィールド	説明
Version	IPv4 パケット ヘッダーのバージョン フィールドに該当しますが、IPv4 で示される数字 4 の代わりに、IPv6 では数字 6 が示されます。
トラフィック クラス	IPv4 パケット ヘッダーのタイプ オブ サービス フィールドと同様です。トラフィック クラス フィールドは、差別化されたサービスで使用されるトラフィック クラスのタグをパケットに付けます。
フロー ラベル	IPv6 パケット ヘッダーの新規フィールドです。フロー ラベル フィールドは、ネットワーク層でパケットを差別化する特定のフローのタグをパケットに付けます。
ペイロード長	IPv4 パケット ヘッダーの合計長フィールドと同様です。ペイロード長フィールドは、パケットのデータ部分の合計長を示します。

表 3-2　基本 IPv6 パケット ヘッダー フィールド（続き）

フィールド	説明
次ヘッダー	IPv4 パケット ヘッダーのプロトコル フィールドと同様です。次ヘッダー フィールドの値により、基本 IPv6 ヘッダーに続く情報のタイプが決まります。基本 IPv6 ヘッダーに続く情報のタイプは、図 3-9 に示すように、TCP パケット、UDP パケット、または拡張ヘッダーなどのトランスポート層パケットです。
ホップ リミット	IPv4 パケット ヘッダーの存続可能時間フィールドと同様です。ホップ リミット フィールドの値は、IPv6 パケットが無効と見なされる前に通過できるルータの最大数です。各ルータを通過するたびに、この値が 1 つずつ減少します。IPv6 ヘッダーにはチェックサムがないため、ルータは値を減らすたびにチェックサムを再計算する必要がなく、処理リソースが節約されます。
送信元アドレス	IPv4 パケット ヘッダーの送信元アドレス フィールドと同様ですが、IPv4 の 32 ビット送信元アドレスの代わりに、IPv6 では 128 ビットの送信元アドレスが含まれます。
Destination Address	IPv4 パケット ヘッダーの宛先アドレス フィールドと同様ですが、IPv4 の 32 ビット宛先アドレスの代わりに、IPv6 では 128 ビットの宛先アドレスが含まれます。

図 3-9　IPv6 パケット ヘッダーの形式



任意に使用できる拡張ヘッダーおよびパケットのデータ部分は、基本 IPv6 パケット ヘッダーの 8 つのフィールドのあとに続きます。存在する場合は、各拡張ヘッダーが 64 ビットに揃えられます。IPv6 パケットの拡張ヘッダーの数は固定されていません。各拡張ヘッダーは、前のヘッダーの次ヘッダー フィールドによって識別されます。通常は、最後の拡張ヘッダーに、TCP や UDP などのトランスポートレイヤ プロトコルの次ヘッダー フィールドがあります。図 3-10 IPv6 拡張ヘッダー形式を示します。

図 3-10　IPv6 拡張ヘッダーのフォーマット

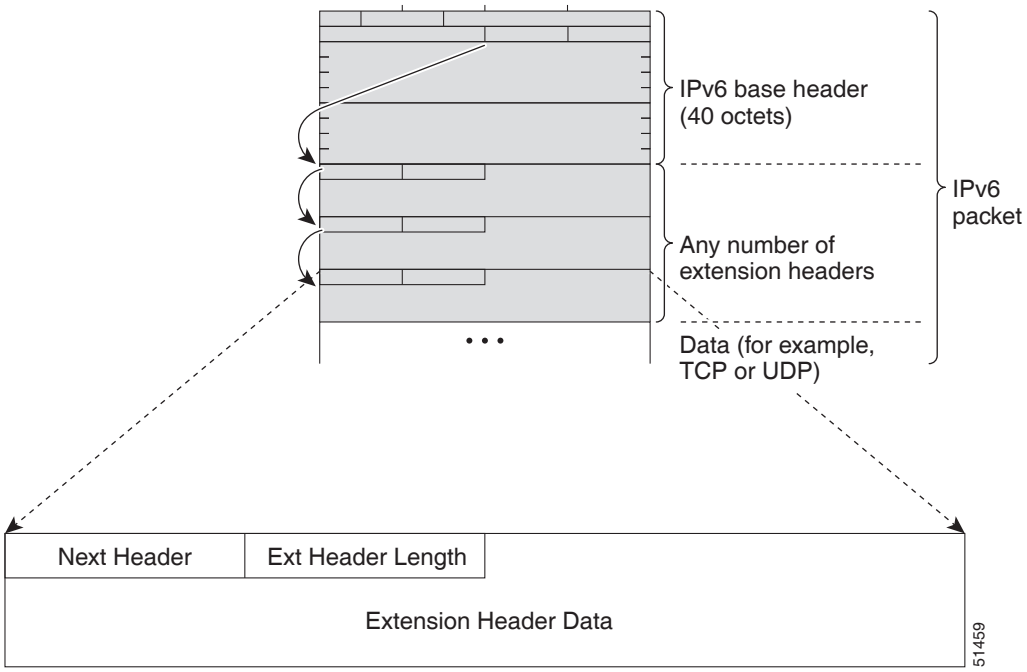


表 3-3 に、拡張ヘッダー タイプとその次ヘッダー フィールド値をリストします。

表 3-3　IPv6 拡張ヘッダーのタイプ

ヘッダー タイプ	次ヘッダーの値	説明
ホップバイホップ オプション ヘッダー	0	パケットのパス上のすべてのホップで処理されるヘッダー。存在する場合、ホップバイホップ オプション ヘッダーは、常に基本 IPv6 パケット ヘッダーの直後に続きます。
宛先オプション ヘッダー	60	任意のホップバイホップ オプション ヘッダーのあとに続くことのあるヘッダー。このヘッダーは、最終の宛先、およびルーティング ヘッダーで指定された各通過アドレスで処理されます。
ルーティング ヘッダー	43	送信元のルーティングに使用されるヘッダー。
フラグメント ヘッダー	44	送信元が、送信元と宛先の間のパスの最大伝送単位（MTU）より大きいパケットをフラグメント化するときに使用されるヘッダー。フラグメント ヘッダーは、フラグメント化された各パケットで使用されます。
上位層ヘッダー	6（TCP） 17（UDP）	データ転送のためにパケット内で使用されるヘッダー。2 つの主要なトランスポート プロトコルは TCP と UDP です。

IPv6 の DNS

IPv6 では、DNS の名前からアドレスおよびアドレスから名前のルックアップ プロセスでサポートされる DNS レコード タイプがサポートされます。DNS レコード タイプは IPv6 アドレスをサポートしています（表 3-4を参照）。

（注）　IPv6 では、IPv6 アドレスから DNS 名への逆マッピングもサポートされます。

表 3-4　IPv6 DNS レコード タイプ

レコード タイプ	説明	書式
AAAA	ホスト名を IPv6 アドレスにマッピングします（IPv4 の A レコードと同等）。	www.abc.test AAAA 3FFE:YYYY:C18:1::2
PTR	IPv6 アドレスをホスト名にマッピングします（IPv4 の PTR レコードと同等）。	2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.8.1.c.0.y.y.y.y.e.f.f.3.ip6.int PTR www.abc.test

IPv6 のパス MTU 探索

IPv4 の場合と同様に、ホストがダイナミックに、データ パス上のすべてのリンクの MTU サイズの差を検出し、それに合わせて調整できるよう、IPv6 でパス MTU ディスカバリを使用できます。ただし、IPv6 では、特定のデータ パス上の 1 つのリンクのパス MTU がパケットのサイズに十分に対応できる大きさでない場合に、フラグメンテーションはパケットの送信元によって処理されます。IPv6 ホストでパケット フラグメンテーションを処理すると、IPv6 ルータの処理リソースが節約され、IPv6 ネットワークの効率が向上します。ICMP の Too Big メッセージの到着によってパス MTU が削減されると、Cisco NX-OS はその低い値を保持します。この接続では、スループットを測定するためにセグメント サイズが増加することはありません。

（注）　IPv6 では、最小リンク MTU は 1280 オクテットです。IPv6 リンクには、1500 オクテットの MTU 値の使用を推奨します。

CDP IPv6 アドレスのサポート

ネイバー情報機能向けの Cisco Discovery Protocol（CDP）IPv6 アドレスのサポートを使用して、2 台のシスコ デバイス間で IPv6 アドレス指定情報を転送できます。IPv6 アドレス向け Cisco Discovery Protocol サポートは、ネットワーク管理製品およびトラブルシューティング ツールに IPv6 情報を提供します。

LPM ルーティング モード

デフォルトでは、Cisco NX-OS プログラムは階層方式でルートをプログラミングし、デバイス上での最長プレフィクス照合（LPM）が可能になります。ただし、より大量の LPM ルート エントリをサポートするために、異なるルーティング モード用にデバイスを設定できます。

次の表に、Cisco Nexus 9300 シリーズおよび 9500 シリーズ スイッチでサポートされている LPM ルーティング モードを示します。

表 3-5　Cisco Nexus 9300 シリーズ スイッチ用の LPM ルーティング モード

LPM ルーティング モード	Broadcom T2モード	CLI コマンド
デフォルト システム ルーティング モード	3	
ALPM ルーティング モード	4	**system routing max-mode l3**

表 3-6　Cisco Nexus 9500 シリーズ スイッチ用の LPM ルーティング モード

LPM ルーティング モード	Broadcom T2モード	CLI コマンド
デフォルト システム ルーティング モード	3（ライン カード用） 4（ファブリック モジュール用）	
最大ホスト ルーティング モード	2（ライン カード用） 3（ファブリック モジュール用）	**system routing max-mode host**
非階層ルーティング モード	3（ライン カード用） 4 および **max-l3-mode** オプション（ライン カード用）	**system routing non-hierarchical-routing** [**max-l3-mode**]
64 ビット ALPM ルーティング モード	モード 4 のサブモード（ファブリック モジュール用）	**system routing mode hierarchical 64b-alpm**

設定の詳細については、「IPv6 の設定」セクション（3-14 ページ）を参照してください。

バーチャライゼーションのサポート

IPv6 は、仮想ルーティング/転送（VRF）インスタンスをサポートします。

IPv6 のライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	IPv6 にはライセンスは不要です。ライセンス パッケージに含まれていない機能は nx-os イメージにバンドルされており、無料で提供されます。Cisco NX-OS のライセンス スキームの詳細は、『*Cisco NX-OS Licensing Guide*』を参照してください。

IPv6 の前提条件

IPv6 には、次の前提条件があります。

- IPv6 アドレッシングおよび IPv6 ヘッダー情報などの IPv6 の基本に関する詳しい知識が必要です。
- デバイスをデュアルスタック デバイス（IPv4/IPv6）にする場合は、必ずメモリ/処理の注意事項に従ってください。

IPv6 の注意事項および制約事項

IPv6 の設定時の注意事項および制約事項は、次のとおりです。

- スイッチは、IPv6 フレームを転送する前にレイヤ 3 パケット情報を確認しないため、IPv6 パケットは、レイヤ 2 LAN スイッチに対して透過的です。IPv6 ホストは、レイヤ 2 LAN スイッチに直接接続できます。
- インターフェイスの同じプレフィックス内に複数の IPv6 グローバル アドレスを設定できます。ただし、1 つのインターフェイス上での複数の IPv6 リンクローカル アドレスはサポートされません。
- RFC 3879 によりサイトローカル アドレスの使用が廃止されたため、RFC 4193 のユニークローカル アドレス（UCA）の推奨に従って、プライベート IPv6 アドレスを設定する必要があります。

IPv6 の設定

この項では、次のトピックについて取り上げます。

（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。

IPv6 アドレッシングの設定

インターフェイスで IPv6 トラフィックを転送できるように、インターフェイス上で IPv6 アドレスを設定する必要があります。インターフェイスでグローバル IPv6 アドレスを設定すると、リンクローカル アドレスが自動的に設定され、そのインターフェイスで IPv6 が有効となります。

手順の概要

1. **configure terminal**
2. **interface ethernet** *number*
3. **ipv6 address** {*addr* [**eui64**] [**route-preference** *preference*] [**secondary**] **tag** *tag-id*]]

   または

   ipv6 address *ipv6-address* **use-link-local-only**
4. （任意）**show ipv6 interface**
5. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 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	**ipv6 address** {*addr* [**eui64**] [**route-preference** *preference*] [**secondary**] **tag** *tag-id*] または **ipv6 address** *ipv6-address* **use-link-local-only** **例**: switch(config-if)# ipv6 address 2001:0DB8::1/10 または switch(config-if)# ipv6 address use-link-local-only	インターフェイスに割り当てられている IPv6 アドレスを指定し、そのインターフェイスで IPv6 処理をイネーブルにします。 **ipv6 address** コマンドを入力すると、IPv6 アドレスの下位 64 ビットにインターフェイス ID を含むグローバル IPv6 アドレスが設定されます。指定する必要があるのはアドレスの 64 ビット ネットワーク プレフィックスだけです。最後の 64 ビットはインターフェイス ID から自動的に計算されます。 **ipv6 address use-link-local-only** コマンドを入力すると、インターフェイス上で IPv6 がイネーブルになったときに自動的に設定されるリンクローカル アドレスの代わりに使用されるリンクローカル アドレスがインターフェイス上に設定されます。 このコマンドは、IPv6 アドレスを設定せずに、インターフェイス上で IPv6 処理をイネーブルにします。
ステップ 4	**show ipv6 interface** **例**: switch(config-if)# show ipv6 interface	（任意）IPv6 に設定されたインターフェイスを表示します。
ステップ 5	**copy running-config startup-config** **例**: switch(config-if)# copy running-config startup-config	（任意）この設定の変更を保存します。

次に、IPv6 アドレスを設定する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# ipv6 address ?
A:B::C:D/LEN IPv6 prefix format: xxxx:xxxx/ml, xxxx:xxxx::/ml,
xxxx::xx/128
use-link-local-only  Enable IPv6 on interface using only a single link-local
address
switch(config-if)# ipv6 address 2001:db8::/64 eui64
```

次に、IPv6 インターフェイスを表示する例を示します。

```
switch(config-if)# show ipv6 interface ethernet 3/1
Ethernet3/1, Interface status: protocol-down/link-down/admin-down, iod: 36
    IPv6 address: 2001:db8:0000:0000:0218:baff:fed8:239d
    IPv6 subnet: 2001:db8::/64
    IPv6 link-local address: fe80::0218:baff:fed8:239d (default)
    IPv6 multicast routing: disabled
    IPv6 multicast groups locally joined:
        ff02::0001:ffd8:239d  ff02::0002  ff02::0001  ff02::0001:ffd8:239d
    IPv6 multicast (S,G) entries joined: none
    IPv6 MTU: 1500 (using link MTU)
    IPv6 RP inbound packet-filtering policy: none
    IPv6 RP outbound packet-filtering policy: none
    IPv6 inbound packet-filtering policy: none
    IPv6 outbound packet-filtering policy: none
    IPv6 interface statistics last reset: never
    IPv6 interface RP-traffic statistics: (forwarded/originated/consumed)
        Unicast packets: 0/0/0
        Unicast bytes: 0/0/0
        Multicast packets: 0/0/0
        Multicast bytes: 0/0/0
```

最大ホスト ルーティング モードの設定（Cisco Nexus 9500 シリーズ スイッチのみ）

デフォルトでは、デバイスは階層方式で（モード 4 になるように設定されたファブリック モジュールとモード 3 になるように設定されたラインカード モジュールで）ルートをプログラミングし、デバイス上での最長プレフィクス照合（LPM）とホスト スケールが可能になります。

デフォルトの LPM およびホスト スケールを変更してシステム内のホストをさらにプログラミングできます。これは、ノードをレイヤ 2 ～ レイヤ 3 の境界ノードとして位置付けるときに必要になる場合があります。

（注）　LPM テーブルのエントリをさらに拡大したい場合は、「非階層ルーティング モードの設定（Cisco Nexus 9500 シリーズ スイッチのみ）」セクション（3-17 ページ）を参照して、ラインカード上のレイヤ 3 IPv4 および IPv6 ルートすべてをプログラミングしてファブリック モジュール上のルートはそのままにするようデバイスを設定します。

（注）　この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注）　最大ホスト ルーティング モードのスケール数については、『*Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*』を参照してください。

手順の概要

1. **configure terminal**
2. **[no] system routing max-mode host**
3. （任意）**show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

	コマンド	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	**[no] system routing max-mode host** **例:** `switch(config)# system routing max-mode host`	ラインカードを Broadcom T2 モード 2 に、ファブリック モジュールを Broadcom T2 モード 3 にして、サポートされるホスト数を増やします。
ステップ 3	**show forwarding route summary** **例:** `switch(config)# show forwarding route summary`	（任意）LPM モードを表示します。
ステップ 4	**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。
ステップ 5	**reload** **例:** `switch(config)# reload`	デバイス全体がリブートします。

非階層ルーティング モードの設定（Cisco Nexus 9500 シリーズ スイッチのみ）

ホストの規模が小さい場合（純粋なレイヤ 3 配置の場合など）、コンバージェンス パフォーマンスを向上させるために、ラインカードの最長プレフィクス照合（LPM）のルートをプログラミングすることを推奨します。そうすることによって、ラインカードのルートおよびホストがプログラミングされ、ファブリック モジュールのルートはプログラミングされません。

（注）　この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

手順の概要

1. **configure terminal**
2. **[no] system routing non-hierarchical-routing [max-l3-mode]**
3. （任意）**show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

	コマンド	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	**[no] system routing non-hierarchical-routing [max-l3-mode]** **例:** switch(config)# system routing non-hierarchical-routing max-l3-mode	ラインカードを Broadcom T2モード 3（または **max-l3-mode** オプションを使用している場合は Broadcom T2 モード 4）にし、より大きな LPM スケールをサポートします。その結果、IPv4 および IPv6 ルートのすべてが、ファブリック モジュールではなくラインカードでプログラミングされます。
ステップ 3	**show forwarding route summary** **例:** switch(config)# show forwarding route summary Mode 3: 120K IPv4 Host table 16k LPM table (> 65 < 127 1k entry reserved) Mode 4: 16k V4 host/4k V6 host 128k v4 LPM/20K V6 LPM	（任意）LPM モードを表示します。
ステップ 4	**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。
ステップ 5	**reload** **例:** switch(config)# reload	デバイス全体がリブートします。

64 ビット ALPM ルーティング モードの設定（Cisco Nexus 9500シリーズ スイッチのみ）

64 ビットのアルゴリズム最長プレフィクス照合（ALPM）を使用して IPv4 および IPv6 ルート テーブル エントリを管理できます。64 ビット ALPM ルーティング モードでは、デバイスに保存できるルート エントリの数が大幅に増加します。このモードでは、次のいずれかをプログラミングできます。

- 80,000 個の IPv6 エントリ、IPv4 エントリなし
- IPv6 エントリなし、128,000 個の IPv4 エントリ
- *x* 個の IPv6 エントリと *y* 個の IPv4 エントリ（ただし 2*x* + *y* <= 128,000）

（注） この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注） 64 ビット ALPM ルーティング モードのスケール数については、『*Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*』を参照してください。

手順の概要

1. **configure terminal**
2. **[no] system routing mode hierarchical 64b-alpm**
3. （任意）**show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

	コマンド	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	**[no] system routing mode hierarchical 64b-alpm** **例:** `switch(config)# system routing mode hierarchical 64b-alpm`	マスク長が 64 以下であるすべての IPv4 および IPv6 LPM ルートがファブリック モジュールでプログラミングされます。IPv4 および IPv6 のすべてのホスト ルート、およびマスク長が 65 ～ 127 であるすべての LPM ルートがライン カードでプログラミングされます。
ステップ 3	**show forwarding route summary** **例:** `switch(config)# show forwarding route summary`	（任意）LPM モードを表示します。

	コマンド	目的
ステップ 4	**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。
ステップ 5	**reload** **例:** `switch(config)# reload`	デバイス全体がリブートします。

ALPM ルーティング モードの設定（Cisco Nexus 9300 シリーズ スイッチのみ）

Cisco Nexus 9300シリーズ スイッチは、非常に多くの LPM ルート エントリをサポートするように設定できます。

（注）　この設定は、IPv4 および IPv6 両方のアドレス ファミリに影響を及ぼします。

（注）　ALPM ルーティング モードのスケール数については、『*Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*』を参照してください。

手順の概要

1. **configure terminal**
2. **[no] system routing max-mode l3**
3. （任意）**show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

手順の詳細

	コマンド	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	**[no] system routing max-mode l3** **例:** `switch(config)# system routing max-mode l3`	デバイスを Broadcom T2モードにして、より大きな LPM スケールをサポートします。

	コマンド	目的
ステップ 3	**show forwarding route summary** **例**: `switch(config)# show forwarding route summary`	(任意)LPM モードを表示します。
ステップ 4	**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。
ステップ 5	**reload** **例**: `switch(config)# reload`	デバイス全体がリブートします。

IPv6 コンフィギュレーションの確認

IPv6 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 ipv6 interface	IPv6 関連のインターフェイス情報を表示します。
show ipv6 adjacency	隣接関係テーブルを表示します。

IPv6 の設定例

次に、IPv6 を設定する例を示します。

```
configure terminal
 interface ethernet 3/1
  ipv6 address 2001:db8::/64 eui64
   ipv6 nd reachable-time 10
```

DNS の設定

この章では、Cisco NX-OS デバイスのドメイン ネーム サーバ（DNS）クライアントを設定する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。

DNS クライアントについて

この項では、次のトピックについて取り上げます。

DNS クライアントの概要

自分で名前の割り当てを管理していないネットワーク内のデバイスとの接続を、ネットワーク デバイスが必要とする場合は、DNS を使用して、ネットワーク間でデバイスを特定する一意のデバイス名を割り当てることができます。DNS は、階層方式を使用して、ネットワーク ノードのホスト名を確立します。これにより、クライアントサーバ方式によるネットワークのセグメントのローカル制御が可能となります。DNS システムは、デバイスのホスト名をその関連する IP アドレスに変換することで、ネットワーク デバイスを検出できます。

インターネット上のドメインは、組織のタイプや場所に基づく一般的なネットワークのグループを表す命名階層ツリーの一部です。ドメイン名は、ピリオド（.）を区切り文字として使用して構成されています。たとえば、シスコは、インターネットでは *com* ドメインで表される営利団体であるため、そのドメイン名は *cisco.com* です。このドメイン内の特定のホスト名、たとえばファイル転送プロトコル（FTP）システムは *ftp.cisco.com* で識別されます。

ネーム サーバ

ネーム サーバはドメイン名の動向を把握し、自身が完全な情報を持っているドメイン ツリーの部分を認識しています。ネーム サーバは、ドメイン ツリーの他の部分の情報を格納している場合もあります。Cisco NX-OS 内の IP アドレスにドメイン名をマッピングするには、ホスト名を示し、ネーム サーバを指定して、DNS サービスをイネーブルにする必要があります。

Cisco NX-OS では、スタティックに IP アドレスをドメイン名にマッピングできます。また、1 つ以上のドメイン ネーム サーバを使用してホスト名の IP アドレスを見つけるよう、Cisco NX-OS を設定することもできます。

DNS の動作

ネーム サーバは、クライアントが DNS サーバに発行した、特定のゾーン内でローカルに定義されたホストの照会を次のように処理します。

- 権限ネーム サーバは、その権限ゾーン内のドメイン名を求める DNS ユーザ照会に、自身のホスト テーブル内にキャッシュされた永久的なエントリを使用して応答します。照会で求められているのが、自身の権限ゾーン内であるが、設定情報が登録されていないドメイン名の場合、権限ネーム サーバはその情報が存在しないと応答します。
- 権限ネーム サーバとして設定されていないネーム サーバは、以前に受信した照会への返信からキャッシュした情報を使用して、DNS ユーザ照会に応答します。ゾーンの権限ネーム サーバとして設定されたルータがない場合は、ローカルに定義されたホストを求める DNS サーバへの照会には、正規の応答は送信されません。

ネーム サーバは、特定のドメインに設定された転送パラメータおよびルックアップ パラメータに従って、DNS 照会に応答します（着信 DNS 照会を転送するか、内部的に生成された DNS 照会を解決します）。

ハイ アベイラビリティ

Cisco NX-OS は、DNS クライアントのステートレス再起動をサポートしています。リブートまたはスーパーバイザ スイッチオーバーのあとに、Cisco NX-OS は実行コンフィギュレーションを適用します。

仮想化のサポート

Cisco NX-OS は、同じシステム上で動作する、DNS クライアントの複数インスタンスをサポートしています。DNS クライアントを設定できます。任意で、各仮想ルーティングおよび転送（VRF）インスタンスで、異なる DNS クライアント設定を使用できます。

DNS クライアントのライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	DNS にはライセンスは不要です。ライセンス パッケージに含まれていない機能は nx-os イメージにバンドルされており、無料で提供されます。NX-OS ライセンス方式の詳細については、『*Cisco NX-OS Licensing Guide*』を参照してください。

DNS クライアントの前提条件

DNS クライアントには次の前提条件があります。

- ネットワーク上に DNS ネーム サーバが必要です。

DNS に関する注意事項および制限事項

DNS クライアントの設定時の注意事項および制約事項は、次のとおりです。

- DNS クライアントは特定の VRF で設定します。VRF を指定しない場合、Cisco NX-OS はデフォルトの VRF を使用します。

デフォルト設定値

表 4-1 は、DNS クライアント パラメータのデフォルト設定の一覧です。

表 4-1　デフォルト DNS クライアント パラメータ

パラメータ	デフォルト
DNS クライアント	イネーブル

DNS クライアントの設定

この項では、次のトピックについて取り上げます。

- DNS クライアントの設定（4-3 ページ）
- 仮想化の設定（4-5 ページ）

（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。

DNS クライアントの設定

ネットワーク上の DNS サーバを使用するよう、DNS クライアントを設定できます。

はじめる前に

ネットワーク上にドメイン ネーム サーバがあることを確認します。

手順の概要

1. **configure terminal**
2. **ip host** *name address1* [*address2...address6*]
3. （任意）**ip domain-name** *name* [**use-vrf** *vrf-name*]
4. （任意）**ip domain-list** *name* [**use-vrf** *vrf-name*]

5. （任意）**ip name-server** *address1* [*address2... address6*] [**use-vrf** *vrf-name*]
6. （任意）**ip domain lookup**
7. （任意）**show hosts**
8. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	**ip host** *name address1* [*address2... address6*] **例:** `switch(config)# ip host cisco-rtp 192.0.2.1`	ホスト名キャッシュに、6 つまでのスタティック ホスト名/アドレス マッピングを定義します。使用可能なアドレスは、IPv4 アドレスまたは IPv6 アドレスです。
ステップ 3	**ip domain-name** *name* [**use-vrf** *vrf-name*] **例:** `switch(config)# ip domain-name myserver.com`	（任意）Cisco NX-OS が無条件ホスト名を完成するために使用するデフォルト ドメイン ネームを定義します。このドメイン名を設定した VRF でこのドメイン ネームを解決できない場合は、任意で、Cisco NX-OS がこのドメイン ネームを解決するために使用する VRF を定義することもできます。 Cisco NX-OS は、ドメイン名ルックアップを開始する前に、完全なドメイン名を含まないあらゆるホスト名にデフォルト ドメイン名を付加します。
ステップ 4	**ip domain-list** *name* [**use-vrf** *vrf-name*] **例:** `switch(config)# ip domain-list mycompany.com`	（任意）Cisco NX-OS が修飾されていないホスト名を完成するために使用できる追加のドメイン ネームを定義します。このドメイン名を設定した VRF でこのドメイン ネームを解決できない場合は、任意で、Cisco NX-OS がこのドメイン ネームを解決するために使用する VRF を定義することもできます。 Cisco NX-OS は、ドメイン名ルックアップを開始する前に、ドメイン リスト内の各エントリを使用して、完全なドメイン名を含まないすべてのホスト名にそのドメイン名を付加します。Cisco NX-OS は、一致するものが見つかるまで、ドメイン リストの各エントリにこのプロセスを実行します。
ステップ 5	**ip name-server** *address1* [*address2... address6*] [**use-vrf** *vrf-name*] **例:** `switch(config)# ip name-server 192.0.2.22`	（任意）最大 6 つのネーム サーバを定義します。使用可能なアドレスは、IPv4 アドレスまたは IPv6 アドレスです。 このネーム サーバを設定した VRF でこのネーム サーバに到達できない場合は、任意で、Cisco NX-OS がこのネーム サーバに到達するために使用する VRF を定義することもできます。

	コマンド	目的
ステップ 6	**ip domain-lookup** **例**: switch(config)# ip domain-lookup	(任意)DNS ベースのアドレス変換をイネーブルにします。この機能は、デフォルトでイネーブルにされています。
ステップ 7	**show hosts** **例**: switch(config)# show hosts	(任意)DNS に関する情報を表示します。
ステップ 8	**copy running-config startup-config** **例**: switch(config)# copy running-config startup-config	(任意)この設定の変更を保存します。

次に、デフォルト ドメイン名を設定し、DNS ルックアップをイネーブルにする例を示します。

```
switch# configure terminal
switch(config)# ip domain-name cisco.com
switch(config)# ip name-server 192.0.2.1 use-vrf management
switch(config)# ip domain-lookup
switch(config)# copy running-config startup-config
```

仮想化の設定

DNS クライアントを VRF 内で設定できます。VRF コンフィギュレーション モードを使用しない場合は、ご使用の DNS クライアント設定がデフォルト VRF に適用されます。

または、DNS クライアントを設定した VRF 以外の、指定した VRF をバックアップ VRF として使用するよう、DNS クライアントを設定することもできます。たとえば、DNS クライアントを赤の VRF で設定していても、赤の VRF で DNS サーバに到達できない場合は、青の VRF を使用して DNS サーバと通信できます。

はじめる前に

ネットワーク上にドメイン ネーム サーバがあることを確認します。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. (任意)**ip domain-name** *name* [**use-vrf** *vrf-name*]
4. (任意)**ip domain-list** *name* [**use-vrf** *vrf-name*]
5. (任意)**ip name-server** *server-address1* [*server-address2... server-address6*] [**use-vrf** *vrf-name*]
6. (任意)**show hosts**
7. (任意)**copy running-config startup-config**

手順の詳細

	コマンド	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	**vrf context** *vrf-name* **例:** switch(config)# vrf context Red switch(config-vrf)#	VRF を作成し、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 domain-name** *name* [**use-vrf** *vrf-name*] **例:** switch(config-vrf)# ip domain-name myserver.com	(任意)Cisco NX-OS が無条件ホスト名を完成するために使用するデフォルト ドメイン ネーム サーバを定義します。このドメイン名を設定した VRF でこのドメイン ネーム サーバを解決できない場合は、任意で、Cisco NX-OS がこのドメイン ネーム サーバを解決するために使用する VRF を定義することもできます。 Cisco NX-OS は、ドメイン名ルックアップを開始する前に、完全なドメイン名を含まないあらゆるホスト名にデフォルト ドメイン名を付加します。
ステップ 4	**ip domain-list** *name* [**use-vrf** *vrf-name*] **例:** switch(config-vrf)# ip domain-list mycompany.com	(任意)Cisco NX-OS が修飾されていないホスト名を完成するために使用できる追加のドメイン ネーム サーバを定義します。このドメイン名を設定した VRF でこのドメイン ネーム サーバを解決できない場合は、任意で、Cisco NX-OS がこのドメイン ネーム サーバを解決するために使用する VRF を定義することもできます。 Cisco NX-OS は、ドメイン名ルックアップを開始する前に、ドメイン リスト内の各エントリを使用して、完全なドメイン名を含まないすべてのホスト名にそのドメイン名を付加します。Cisco NX-OS は、一致するものが見つかるまで、ドメイン リストの各エントリにこのプロセスを実行します。
ステップ 5	**ip name-server** *address1* [*address2... address6*] [**use-vrf** *vrf-name*] **例:** switch(config-vrf)# ip name-server 192.0.2.22	(任意)最大 6 つのネーム サーバを定義します。使用可能なアドレスは、IPv4 アドレスまたは IPv6 アドレスです。 このネーム サーバを設定した VRF でこのネーム サーバに到達できない場合は、任意で、Cisco NX-OS がこのネーム サーバに到達するために使用する VRF を定義することもできます。

	コマンド	目的
ステップ 6	**show hosts** **例**: switch(config-vrf)# show hosts	(任意)DNS に関する情報を表示します。
ステップ 7	**copy running-config startup-config** **例**: switch(config-vrf)# copy running-config startup-config	(任意)この設定の変更を保存します。

次に、デフォルト ドメイン名を設定し、VRF 内の DNS ルックアップをイネーブルにする例を示します。

```
switch# configure terminal
switch(config)# vrf context Red
switch(config-vrf)# ip domain-name cisco.com
switch(config-vrf)# ip name-server 192.0.2.1 use-vrf management
switch(config-vrf)# copy running-config startup-config
```

DNS クライアント設定の確認

DNS クライアント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 hosts	DNS に関する情報を表示します。

DNS クライアントの設定例

次に、複数の代替ドメイン名のドメイン リストを設定する例を示します。

```
ip domain-list csi.com
ip domain-list telecomprog.edu
ip domain-list merit.edu
```

次に、ホスト名とアドレス間のマッピング プロセスを設定し、IP DNS ベースの変換を指定する例を示します。例では、ネーム サーバとデフォルトのドメイン名のアドレスを設定します。

```
ip domain-lookup
ip name-server 192.168.1.111 192.168.1.2
ip domain-name cisco.com
```

OSPFv2 の設定

この章では、Cisco NX-OS デバイスで 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」セクション（1-9 ページ）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。これらの隣接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されます（「コンバージェンス」セクション（1-6 ページ）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、OSPFv3 は IPv6 をサポートしています。詳細については、第 6 章「OSPFv3 の設定」を参照してください。

（注）　Cisco NX-OS の OSPFv2 では、RFC 2328 をサポートしています。この RFC では、ルート サマリー コストの計算に、RFC1583 で使用する計算と互換性がない別の方法が導入されました。また RFC 2328 では、AS-external パスに対して異なる選択基準が導入されました。すべてのルータが同じ RFC をサポートしていることを確認することが重要です。RFC1583 だけに対応しているルータがネットワークに含まれる場合は、**rfc1583compatibility** コマンドを使用します。デフォルトでサポートされている OSPFv2 用の RFC 標準は、Cisco NX-OS と Cisco IOS とで異なる場合があります。値が同じになるように設定するには、調整が必要です。詳細については、「OSPF RFC 互換モードの例」セクション（5-51 ページ）を参照してください。

この項では、次のトピックについて取り上げます。

- hello パケット（5-2 ページ）
- ネイバー（5-3 ページ）
- 隣接関係（5-3 ページ）
- 指定ルータ（5-4 ページ）
- エリア（5-5 ページ）
- リンクステート アドバタイズメント（5-6 ページ）
- OSPFv2 とユニキャスト RIB（5-8 ページ）
- 認証（5-8 ページ）
- 高度な機能（5-9 ページ）

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
- 双方向通信
- 指定ルータの選定（「指定ルータ」セクション（5-4 ページ）を参照）

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（「ネイバー」セクション（5-3 ページ）を参照）。

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）で 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。

ネイバー

ネイバーと見なされるためには、OSPFv2 インターフェイスがリモート インターフェイスとの互換性を持つように設定されている必要があります。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」セクション（5-5 ページ）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」セクション（5-4 ページ）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」セクション（5-4 ページ）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（5-4 ページ）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含まれるのは、ネイバーのリンクステート データベースからの LSA ヘッダーだけです（「リンクステート データベース」セクション（5-7 ページ）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応じて、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場合があります（「エリア」セクション（5-5 ページ）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。

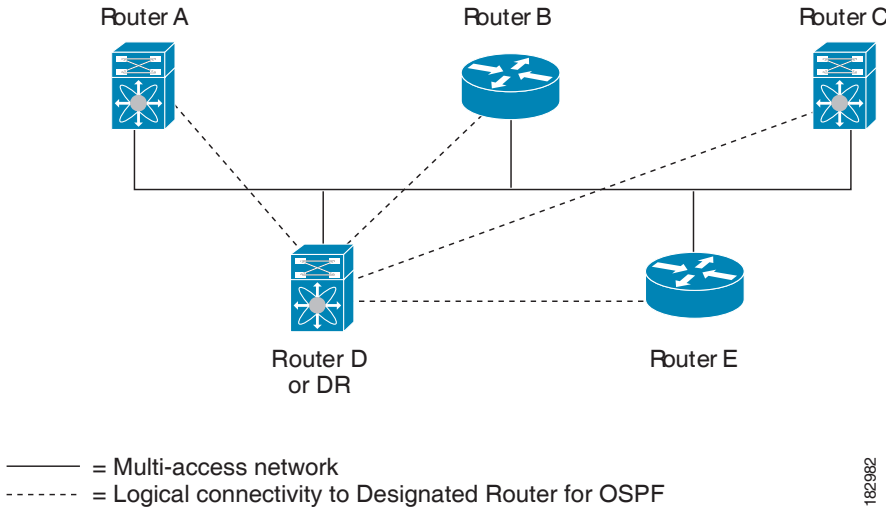
- ポイントツーポイント：2 台のルータ間にのみ存在するネットワーク。ポイントツーポイント ネットワーク上の全ネイバーは隣接関係を確立し、DR は存在しません。
- ブロードキャスト：ブロードキャスト トラフィックが可能なイーサネットなどの共有メディア上で通信できる複数のルータを持つネットワーク。OSPFv2 ルータは DR および BDR を確立し、これらにより、ネットワーク上の LSA フラッディングを制御します。OSPFv2 は、よく知られている IPv4 マルチキャスト アドレス 224.0.0.5 および MAC アドレス 0100.5300.0005 を使用して、ネイバーと通信します。

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。

他のルータはすべて DR および BDR と隣接関係を確立し、IPv4 マルチキャスト アドレス 224.0.0.6 を使用して、LSA 更新情報を DR と BDR に送信します。図 5-1 すべてのルータと DR の間のこの隣接関係を示します。

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 5-1　マルチアクセス ネットワークの DR



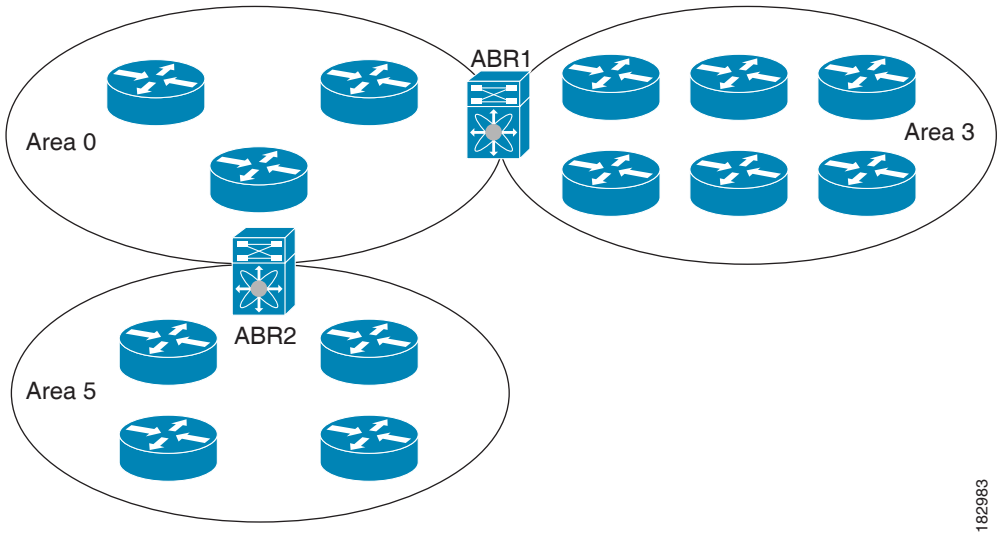
エリア

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（図 5-2を参照）。

図 5-2　　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（5-11 ページ）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。図 5-2 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」セクション（5-9 ページ）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、自身のルーティング テーブルを構築します。

この項では、次のトピックについて取り上げます。

LSA タイプ

表 5-1 は、Cisco NX-OS でサポートされる LSA タイプを示します。

表 5-1　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」セクション（5-4 ページ）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」セクション（5-5 ページ）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」セクション（5-5 ページ）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」セクション（5-5 ページ）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」セクション（5-5 ページ）を参照してください。
9–11	不透明 LSA	OSPF の拡張に使用される LSA。「不透明 LSA」セクション（5-7 ページ）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、OSPFv2 エリアをこの情報で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」セクション（5-5 ページ）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 またはバッファの高い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 10 秒以内の 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 データベースを維持しています。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 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」セクション（5-7 ページ）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。OSPFv2 は不透明 LSA を使用して、OSPFv2 グレースフル リスタート機能（「ハイ アベイラビリティおよびグレースフル リスタート」セクション（5-12 ページ）を参照）をサポートしています。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 AS にフラッディングされます。

OSPFv2 とユニキャスト 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 ルーティング情報ベース（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」セクション（5-13 ページ）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　では、次の 認証方式がサポートされています。

- 簡易パスワード認証
- 暗号化認証

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。

暗号化認証

暗号化認証では、OSPFv2 認証に暗号化パスワードを使用します。トランスミッタは、送信するパケットとキー文字列を使用してコードを計算し、そのコードとキー ID をパケットに挿入してパケットを送信します。レシーバは、受信したパケットとローカルに設定されたキー文字列（パケットのキー ID に対応）を使用してコードをローカル上で計算し、これによってパケット内のコードを検証します。

Message Digest 5（MD5）とハッシュ ベースの Message Authentication Code Secure Hash Algorithm（HMAC-SHA）の両方がサポートされます。

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。

HMAC-SHA 認証

Cisco NX-OS Release 7.0(3)I3(1) 以降では、OSPFv2 は RFC 5709 をサポートしており、MD5 より高度なセキュリティを提供する HMAC-SHA アルゴリズムを使用できます。OSPFv2 認証では、HMAC-SHA-1、HMAC-SHA-256、HMAC-SHA-384、HMAC-SHA-512 アルゴリズムがサポートされています。

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高度な OSPFv2 機能をサポートしています。この項では、次のトピックについて取り上げます。

- スタブ エリア（5-9 ページ）
- Not-So-Stubby エリア（5-10 ページ）
- 仮想リンク（5-10 ページ）
- ルートの再配布（5-11 ページ）
- ルート集約（5-11 ページ）
- ハイ アベイラビリティおよびグレースフル リスタート（5-12 ページ）
- OSPFv2 スタブ ルータ アドバタイズメント（5-13 ページ）
- 複数の OSPFv2 インスタンス（5-13 ページ）
- SPF 最適化（5-13 ページ）
- BFD（5-13 ページ）
- 仮想化のサポート（5-13 ページ）

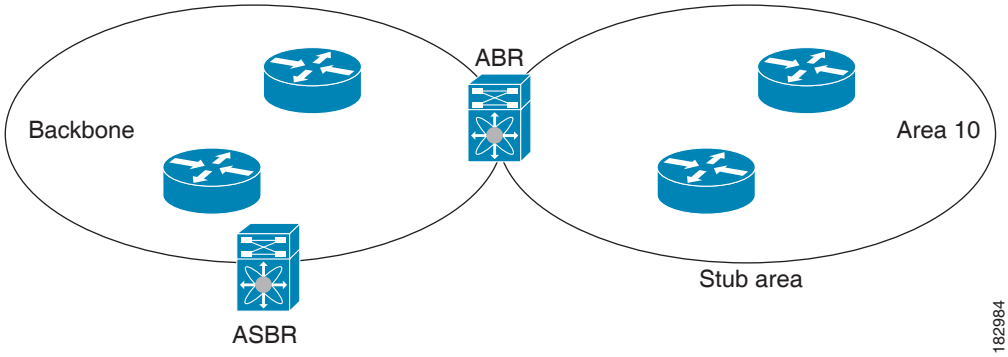
スタブ エリア

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」セクション（5-6 ページ）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」セクション（1-7 ページ）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。

図 5-3 は、外部自律システム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 自律システムの例を示します。エリア 0.0.0.10 は、スタブ エリアとして設定できます。

図 5-3　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 エリア

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、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 に関する情報については、「リンクステート アドバタイズメント」セクション（5-6 ページ）を参照してください。

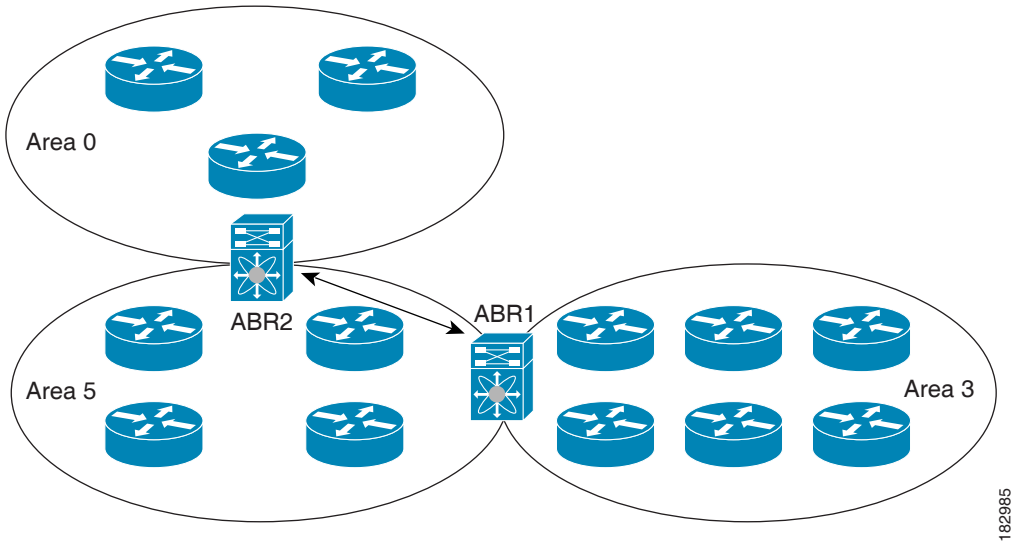
たとえば、OSPFv2 を使用する中央サイトを、異なるルーティング プロトコルを使用するリモート サイトに接続するときに NSSA を使用すると、管理作業を簡素化できます。リモート サイトへのルートはスタブ エリア内に再配布できないため、NSSA を使用する前に、企業サイトの境界ルータとリモート ルータの間の接続を OSPFv2 スタブ エリアとして実行できません。NSSA を使用すると、企業のルータとリモート ルータ間のエリアを NSSA として定義する（「NSSA の設定」セクション（5-30 ページ）を参照）ことで、OSPFv2 を拡張してリモート接続性をサポートできます。

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。図 5-4 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 5-4　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」セクション（1-6 ページ）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。再配布を指定したルート マップを設定して、どのルートが OSPFv2 に渡されるかを制御する必要があります。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。ルート マップを使用して、ローカル OSPFv2 AS 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 マップの設定の詳細については、第 15 章「Route Policy Manager の設定」を参照してください。

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。

ハイ アベイラビリティおよびグレースフル リスタート

Cisco NX-OS では、複数レベルのハイ アベイラビリティ アーキテクチャを提供します。OSPFv2 は、ステートフル リスタートをサポートしています。これは、ノンストップ ルーティング（NSR）とも呼ばれます。OSPFv2 で問題が発生した場合は、以前の実行時状態からの再起動を試みます。この場合、ネイバーはいずれのネイバー イベントも登録しません。最初の再起動が正常ではなく、別の問題が発生した場合、OSPFv2 はグレースフル リスタートを試みます。

グレースフル リスタート、つまり、Nonstop Forwarding（NSF）では、処理の再起動中も OSPFv2 がデータ転送パス上に存在し続けます。OSPFv2 はグレースフル リスタートを実行する必要がある場合、猶予 LSA と呼ばれるリンクローカル不透明（タイプ 9）LSA（「不透明 LSA」セクション（5-7 ページ）を参照）を送信します。この再起動中の OSPFv2 プラットフォームは NSF 対応と呼ばれます。

猶予 LSA には猶予期間が含まれます。猶予期間とは、ネイバー OSPFv2 インターフェイスが再起動中の OSPFv2 インターフェイスからの LSA を待つよう指定された時間です（通常、OSPFv3 は隣接関係を解消し、ダウンした、または再起動中の OSPFv3 インターフェイスが発信するすべての LSA を廃棄します）。関与するネイバーは NSF ヘルパーと呼ばれ、再起動中の OSPFv2 インターフェイスが発信するすべての LSA を、このインターフェイスが隣接したままであるかのように維持します。

再起動中の OSPFv2 インターフェイスが稼働を再開すると、ネイバーを再探索して隣接関係を確立し、LSA 更新情報の送信を再開します。この時点で、NSF ヘルパーは、グレースフル リスタートが完了したと認識します。

ステートフル リスタートは次のシナリオで使用されます。

- プロセスでの問題発生後の最初の回復試行
- **system switchover** コマンドによる手動でのスイッチオーバー

グレースフル リスタートは次のシナリオで使用されます。

- プロセスでの問題発生後の 2 回目の回復試行（4 分以内）
- **restart ospf** コマンドによるプロセスの手動での再開
- アクティブ スーパーバイザの削除
- **reload module** *active-sup* コマンドによるアクティブ スーパーバイザのリロード

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設定できません。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。サポートされる OSPFv2 インスタンスの数については、『*Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*』を参照してください。

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。

BFD

この機能は、IPv4 の Bidirectional Forwarding Detection（BFD）をサポートします。BFD は、転送パスの障害を高速で検出することを目的にした検出プロトコルです。BFD は 2 台の隣接デバイス間のサブセカンド障害を検出し、BFD の負荷の一部を、サポートされるモジュール上のデータ プレーンに分散できるため、プロトコル hello メッセージよりも CPU を使いません。詳細については、『*Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*』を参照してください。

仮想化のサポート

Cisco NX-OS は、OSPFv2 用の複数のプロセス インスタンスをサポートします。各 OSPFv2 インスタンスは、システム制限まで、複数の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できます。サポートされる OSPFv2 インスタンスの数については、『*Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*』を参照してください。

OSPFv2 のライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	OSPFv2 には Enterprise Services ライセンスが必要です。Cisco NX-OS ライセンス方式について、およびライセンスの取得方法と適用方法の詳細については、『*Cisco NX-OS Licensing Guide*』を参照してください。

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。
- Enterprise 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 機能がイネーブルにされている（「OSPFv2 のイネーブル化」セクション（5-16 ページ）を参照）。

OSPFv2 に関する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。
- すべての OSPFv2 ルータが、同じ RFC 互換モードで動作する必要があります。Cisco NX-OS の OSPFv2 は RFC 2328 に準拠しています。ネットワークに RFC 1583 だけに対応しているルータが含まれる場合は、ルータ コンフィギュレーション モードで **rfc1583compatibility** コマンドを使用します。
- スケール シナリオでは、インターフェイスと OSPF プロセスのリンク ステート アドバタイズメントの数が大きい場合、OSPF MIB オブジェクトの SNMP エージェントのタイムアウト値が小さい SNMP ウォークは、タイムアウトになると予想されます。OSPF MIB オブジェクトのポーリング中に問い合わせる SNMP エージェントのタイムアウトを確認する場合は、ポーリングする SNMP エージェントのタイムアウト値を増加してください。
- アドミニストレーティブ ディスタンス機能には、次のガイドラインと制限事項が適用されます。
  - OSPF ルートに複数の等コスト パスがある場合、アドミニストレーティブ ディスタンスを設定しても **match ip route-source** コマンドに対しては決定性を持ちません。
  - アドミニストレーティブ ディスタンスの設定は、**match route-type**、**match ip address prefix-list**、および **match ip route-source prefix-list** コマンドのみでサポートされます。別の match 文は無視されます。

  - OSPF ルートのアドミニストレーティブ ディスタンスを設定するための **match route-type**、**match ip address**、および **match ip route-source** コマンドにはプリファレンスがありません。このように、Cisco NX-OS OSPF アドミニストレーティブ ディスタンスを設定するためのテーブル マップの動作は、Cisco IOS OSPF の場合と異なります。
  - 廃棄ルートには、アドミニストレーティブ ディスタンス 220 が常に割り当てられます。テーブル マップの設定は OSPF の廃棄ルートには適用されません。
- vPC コンフィギュレーション モードで **delay restore** *seconds* コマンドを設定する場合や、マルチシャーシ EtherChannel トランク（MCT）上の VLAN がスイッチ仮想インターフェイス（SVI）を使用して OSPFv2 または OSPFv3 によって通知される場合、これらの SVI は設定された時間の間、vPC セカンダリ ノード上で MAX_LINK_COST で通知されます。その結果、すべてのルートまたはホストのプログラミングは、トラフィックを引き込む前に（セカンダリ vPC ノードのピア リロードで）vPC の同期操作後に完了します。この動作により、ノースサウス トラフィックのパケット損失を最小にできます。

（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。

デフォルト設定値

表 5-2 に、OSPFv2 パラメータのデフォルト設定を示します。

表 5-2　デフォルトの OSPFv2 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	110
hello 間隔	10 秒
デッド間隔	40 秒
廃棄ルート	イネーブル
グレースフル リスタートの猶予期間	60 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
LSA グループ ペーシング	10 秒
SPF 計算初期遅延時間	200 ミリ秒
SPF の最小ホールド タイム	5000 ミリ秒
SPF 計算初期遅延時間	1000 ミリ秒

基本的 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。

この項では、次のトピックについて取り上げます。

OSPFv2 のイネーブル化

OSPFv2 を設定するには、その前に OSPFv2 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 ospf**
3. （任意）**show feature**
4. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 ospf** **例:** `switch(config)# feature ospf`	OSPFv2 機能をイネーブルにします。
ステップ 3	**show feature** **例:** `switch(config)# show feature`	（任意）イネーブルおよびディセーブルにされた機能を表示します。
ステップ 4	**copy running-config startup-config** **例:** `switch(config)# copy running-config startup-config`	（任意）この設定の変更を保存します。

OSPFv2 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 **no feature ospf** コマンドを使用します。

コマンド	目的

`no feature ospf`
 	
例:
`switch(config)# no feature ospf`
	OSPFv2 機能をディセーブルにして、関連付けられた設定をすべて削除します。

OSPFv2 インスタンスの作成

OSPFv2 設定の最初のステップは OSPFv2 インスタンスの作成です。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」セクション（5-26 ページ）を参照してください。

はじめる前に

OSPF 機能がイネーブルにされていることを確認します（「OSPFv2 のイネーブル化」セクション（5-16 ページ）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. **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	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` *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	目的
ステップ 3	**router-id** *ip-address* **例**: `switch(config-router)# router-id 192.0.2.1`	(任意)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	**show ip ospf** *instance-tag* **例**: `switch(config-router)# show ip ospf 201`	(任意)OSPF 情報を表示します。
ステップ 5	**copy running-config startup-config** **例**: `switch(config)# copy running-config startup-config`	(任意)この設定の変更を保存します。

OSPFv2 インスタンスと、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 **no feature 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 インスタンス上のオプション 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」セクション(5-26 ページ)を参照してください。

はじめる前に

OSPF 機能がイネーブルにされていることを確認します(「OSPFv2 のイネーブル化」セクション(5-16 ページ)を参照)。

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。

手順の詳細

ルータ コンフィギュレーション モードで、次の OSPFv2 用オプション パラメータを設定できます。

コマンド	目的
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。
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。
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。範囲は 1 ～ 64 です。デフォルト値は 8 です。
passive-interface default **例:** `switch(config-router)# passive-interface default`	すべての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。このコマンドは、VRF またはインターフェイス コマンド モードの設定によって上書きされます。

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(「ネイバー」セクション(5-3 ページ)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。

はじめる前に

OSPF 機能がイネーブルにされていることを確認します(「OSPFv2 のイネーブル化」セクション(5-16 ページ)を参照)。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ip address** *ip-prefix/length*
4. **ip router ospf** *instance-tag* **area** *area-id* [**secondaries none**]
5. （任意）**show ip ospf** *instance-tag* **interface** *interface-type slot/port*
6. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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** *interface-type 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。
ステップ 3	**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 4	**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。
ステップ 5	**show ip ospf** *instance-tag* **interface** *interface-type slot/port* **例**: switch(config-if)# show ip ospf 201 interface ethernet 1/2	(任意)OSPF 情報を表示します。
ステップ 6	**copy running-config startup-config** **例**: switch(config)# copy running-config startup-config	(任意) この設定の変更を保存します。

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 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。
[**default** | **no**] **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。このコマンドによって、ルータまたは VRF コマンド モードの設定が上書きされます。**default** オプションは、このインターフェイス モード コマンドを削除して、ルータまたは VRF の設定がある場合にはそれに戻します。
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」セクション（5-4 ページ）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)# 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
```

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

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。

はじめる前に

OSPF 機能がイネーブルにされていることを確認します（「OSPFv2 のイネーブル化」セクション（5-16 ページ）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 9000 Series NX-OS Security Configuration Guide』を参照してください。*

（注） OSPFv2 の場合、**key** *key-id* コマンドのキー ID の値は 0 ～ 255 です。

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. （任意）**ip ospf authentication-key** [**0** | **3**] *key*

   または

   ip ospf message-digest-key *key-id* **md5** [**0** | **3**] *key*
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	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** *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。

	コマンド	目的

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 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	**ip ospf authentication-key** [**0** | **3**] *key* **例:** `switch(config-if)# ip ospf authentication-key 0 mypass`	(任意) 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 *key-id* **md5** [**0** | **3**] *key* **例:** `switch(config-if)# ip ospf message-digest-key 21 md5 0 mypass`	(任意) 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 6	**show ip ospf** *instance-tag* **interface** *interface-type slot/port* **例:** `switch(config-if)# show ip ospf 201 interface ethernet 1/2`	(任意) OSPF 情報を表示します。
ステップ 7	**copy running-config startup-config** **例:** `switch(config)# copy running-config startup-config`	(任意) この設定の変更を保存します。

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。

はじめる前に

OSPF 機能がイネーブルにされていることを確認します(「OSPFv2 のイネーブル化」セクション (5-16 ページ)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。OSPFv2 HMAC-SHA 認証を設定するには、キーに使用する HMAC-SHA アルゴリズムを指定する必要があります。キーチェーンを使用する暗号化認証が暗号化アルゴリズムの選択なしで設定されている場合、OSPFv2 は MD5 暗号化アルゴリズムを使用します。『*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*』を参照してください。

(注)　OSPFv2 の場合、**key** *key-id* コマンドのキー ID の値は 0 ～ 255 です。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ip ospf authentication** [**message-digest**]
4. （任意）**ip ospf authentication key-chain** *key-id*
5. （任意）**ip ospf authentication-key** [**0** | **3** | **7**] *key*
6. （任意）**ip ospf message-digest-key** *key-id* **md5** [**0** | **3** | **7**] *key*
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	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** *interface-type 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。

ステップ 3	**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。このインターフェイスのエリアに基づく認証を上書きするには、このコマンドを使用します。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 4	**ip ospf authentication key-chain** *key-id* **例**: `switch(config-if)# ip ospf authentication key-chain Test1`	（任意）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。キーチェーンの詳細については、*『Cisco Nexus 9000 Series NX-OS Security Configuration Guide』*を参照してください。
ステップ 5	**ip ospf authentication-key** [**0** | **3** | **7**] *key* **例**: `switch(config-if)# ip ospf authentication-key 0 mypass`	（任意）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	目的
ステップ 6	**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	(任意)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 7	**show ip ospf** *instance-tag* **interface** *interface-type slot/port* **例**: switch(config-if)# show ip ospf 201 interface ethernet 1/2	(任意)OSPF 情報を表示します。
ステップ 8	**copy running-config startup-config** **例**: switch(config)# copy running-config startup-config	(任意)この設定の変更を保存します。

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)# 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
```

次に、OSPFv2 HMAC-SHA-1 および MD5 暗号化認証を設定する例を示します。

```
switch# configure terminal
switch(config)# key chain chain1
switch(config-keychain)# key 1
switch(config-keychain-key)# key-string 7 070724404206
switch(config-keychain-key)# accept-lifetime 01:01:01 Jan 01 2015  infinite
switch(config-keychain-key)# send-lifetime 01:01:01 Jan 01 2015  infinite
switch(config-keychain-key)# cryptographic-algorithm HMAC-SHA-1
switch(config-keychain-key)# exit
switch(config-keychain)# key 2
switch(config-keychain-key)# key-string 7 070e234f1f5b4a
switch(config-keychain-key)# accept-lifetime 10:51:01 Jul 24 2015  infinite
switch(config-keychain-key)# send-lifetime 10:51:01 Jul 24 2015  infinite
switch(config-keychain-key)# cryptographic-algorithm MD5
switch(config-keychain-key)# exit
switch(config-keychain)# exit

switch(config)# interface ethernet 1/1
switch(config-if)# ip router ospf 1 area 0.0.0.0
switch(config-if)# ip ospf authentication message-digest
switch(config-if)# ip ospf authentication key-chain chain1
```

```
switch(config-if)# show key chain chain1
Key-Chain chain1
  Key 1 -- text 7 "070724404206"
    cryptographic-algorithm HMAC-SHA-1
    accept lifetime UTC (01:01:01 Jan 01 2015)-(always valid) [active]
    send lifetime UTC (01:01:01 Jan 01 2015)-(always valid) [active]
  Key 2 -- text 7 "070e234f1f5b4a"
    cryptographic-algorithm MD5
    accept lifetime UTC (10:51:00 Jul 24 2015)-(always valid) [active]
    send lifetime UTC (10:51:00 Jul 24 2015)-(always valid) [active]

switch(config-if)# show ip ospf interface ethernet 1/1
Ethernet1/1 is up, line protocol is up
    IP address 11.11.11.1/24
    Process ID 1 VRF default, area 0.0.0.3
    Enabled by interface configuration
    State BDR, Network type BROADCAST, cost 40
    Index 6, Transmit delay 1 sec, Router Priority 1
    Designated Router ID: 33.33.33.33, address: 11.11.11.3
    Backup Designated Router ID: 1.1.1.1, address: 11.11.11.1
    2 Neighbors, flooding to 2, adjacent with 2
    Timer intervals: Hello 10, Dead 40, Wait 40, Retransmit 5
      Hello timer due in 0:00:08
    Message-digest authentication, using keychain key1 (ready)
      Sending SA: Key id 2, Algorithm MD5
    Number of opaque link LSAs: 0, checksum sum 0
```

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。

この項では、次のトピックについて取り上げます。

- 境界ルータのフィルタ リストの設定（5-27 ページ）
- スタブ エリアの設定（5-28 ページ）
- Totally Stubby エリアの設定（5-29 ページ）
- NSSA の設定（5-30 ページ）
- 仮想リンクの設定（5-32 ページ）
- 再配布の設定（5-34 ページ）
- 再配布されるルート数の制限（5-36 ページ）
- ルート集約の設定（5-38 ページ）
- スタブ ルート アドバタイズメントの設定（5-39 ページ）
- ルートのアドミニストレーティブ ディスタンスの設定（5-40 ページ）
- デフォルト タイマーの変更（5-43 ページ）
- グレースフル リスタートの設定（5-45 ページ）
- OSPFv2 インスタンスの再起動（5-47 ページ）

境界ルータのフィルタ リストの設定

OSPFv2 ドメインを、関連性のある各ネットワークを含む一連のエリアに分離できます。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」セクション（5-5 ページ）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。「ルート集約の設定」セクション（5-38 ページ）を参照してください。
- Filter list：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。

はじめる前に

OSPF 機能がイネーブルにされていることを確認します（「OSPFv2 のイネーブル化」セクション（5-16 ページ）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。第 15 章「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	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` *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： `switch(config-router)# area 0.0.0.10 filter-list route-map FilterLSAs in`	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	目的
ステップ 4	**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	(任意)OSPF ポリシー情報を表示します。
ステップ 5	**copy running-config startup-config** **例**: switch(config)# copy running-config startup-config	(任意)この設定の変更を保存します。

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部分にスタブ 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、選択したネットワークへの往復の不要なルーティングを制限します。「スタブ エリア」セクション(5-9 ページ)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。

はじめる前に

OSPF 機能がイネーブルにされていることを確認します(「OSPFv2 のイネーブル化」セクション(5-16 ページ)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	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** *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	**area** *area-id* **default-cost** *cost* **例**: switch(config-router)# area 0.0.0.10 default-cost 25	（任意）このスタブ エリア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。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	**show ip ospf** *instance-tag* **例**: switch(config-if)# show ip ospf 201	（任意）OSPF 情報を表示します。
ステップ 6	**copy running-config startup-config** **例**: switch(config)# copy running-config startup-config	（任意）この設定の変更を保存します。

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 エリアを経由しないようにすることができます。

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、ある程度の外部トラフィックが必要な部分に NSSA を設定できます。NSSA の詳細については、「Not-So-Stubby エリア」セクション（5-10 ページ）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が 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。

はじめる前に

OSPF 機能がイネーブルにされていることを確認します（「OSPFv2 のイネーブル化」セクション（5-16 ページ）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	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** *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	**area** *area-id* **default-cost** *cost* **例**: switch(config-router)# area 0.0.0.10 default-cost 25	(任意)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	**show ip ospf** *instance-tag* **例**: switch(config-if)# show ip ospf 201	(任意)OSPF 情報を表示します。
ステップ 6	**copy running-config startup-config** **例**: switch(config)# copy running-config startup-config	(任意)この設定の変更を保存します。

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、中継エリア経由でバックボーン エリアに接続します。「仮想リンク」セクション（5-10 ページ）を参照してください。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。

はじめる前に

OSPF がイネーブルになっていることを確認します（「OSPFv2 のイネーブル化」セクション（5-16 ページ）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	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** *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	**show ip ospf virtual-link** [**brief**] **例**: switch(config-router-vlink)# show ip ospf virtual-link	(任意)OSPF 仮想リンク情報を表示します。
ステップ 5	**copy running-config startup-config** **例**: switch(config-router-vlink)# copy running-config startup-config	(任意)この設定の変更を保存します。

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	(任意)これにより、エリアに基づくこの仮想リンクの認証が無効となります。
authentication-key [**0** | **3**] *key* **例**: switch(config-router-vlink)# authentication-key 0 mypass	(任意)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	(任意)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。

コマンド	目的
hello-interval *seconds* **例**: `switch(config-router-vlink)# hello-interval 25`	(任意)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`	(任意)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`	(任意)OSPFv2 再送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`	(任意)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)# 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)# 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と、Cisco NX-OS はデフォルトのスタティック ルートも再配布します。

はじめる前に

OSPF がイネーブルになっていることを確認します（「OSPFv2 のイネーブル化」セクション（5-16 ページ）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* | **isis** *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	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** *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* | **isis** *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 OSPF に再配布します。 **（注）** スタティック ルートを再配布すると、Cisco NX-OS はデフォルトのスタティック ルートも再配布します。
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 ルートの 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。 **（注）** このコマンドは、ルート マップの **match** 文を無視します。

	コマンド	目的
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。このコマンドは、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	**copy running-config startup-config** **例:** `switch(config-router)# copy running-config startup-config`	(任意) この設定の変更を保存します。

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に最大制限を設定できます。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。

はじめる前に

OSPF がイネーブルになっていることを確認します(「OSPFv2 のイネーブル化」セクション(5-16 ページ)を参照)。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **redistribute** {**bgp** *id* | **direct** | **eigrp** *id* | **isis** *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	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** *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* | **isis** *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 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 が配布するプレフィックスの最大数を指定します。有効な範囲は 1 ～ 65535 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クス数のパーセンテージ。 • **warning-only**：プレフィックスの最大数を超えたとき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、すべてのルートが取り消された場合に使用します。
ステップ 5	**show running-config ospf** **例**: switch(config-router)# show running-config ospf	（任意）OSPFv2 の設定を表示します。
ステップ 6	**copy running-config startup-config** **例**: switch(config-router)# copy running-config startup-config	（任意）この設定の変更を保存します。

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 75
```

ルート集約の設定

集約されたアドレス範囲を設定して、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集約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。詳細については、「ルート集約」セクション（5-11 ページ）を参照してください。

はじめる前に

OSPF がイネーブルになっていることを確認します（「OSPFv2 のイネーブル化」セクション（5-16 ページ）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*]

   または
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	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** *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。
ステップ 4	**summary-address** *ip-prefix/length* [**no-advertise**	**tag** tag] **例：** `switch(config-router)# summary-address 10.5.0.0/16 tag 2`	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。ルート マップによる再配布で使用できるよう、この集約アドレスにタグを割り当てることもできます。

	コマンド	目的
ステップ 5	**show ip ospf summary-address** **例**: `switch(config-router)# show ip ospf summary-address`	(任意)OSPF 集約アドレスに関する情報を表示します。
ステップ 6	**copy running-config startup-config** **例**: `switch(config-router)# copy running-config startup-config`	(任意)この設定の変更を保存します。

次に、ABR 上のエリア間の集約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 上の集約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)# no discard-route internal
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。詳細については、「OSPFv2 スタブ ルータ アドバタイズメント」セクション(5-13 ページ)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。

(注) ルータの実行コンフィギュレーションがグレースフル シャットダウンを行うよう設定されている場合は、その実行コンフィギュレーションを保存しないでください。保存すると、ルータが、リロード後に最大メトリックをアドバタイズし続けることになります。

はじめる前に

OSPF がイネーブルになっていることを確認します(「OSPFv2 のイネーブル化」セクション(5-16 ページ)を参照)。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **max-metric router-lsa** [**external-lsa** [*max-metric-value*]] [**include-stub**] [**on-startup** {*seconds* | **wait-for bgp** *tag*}] [**summary-lsa** [*max-metric-value*]]
4. (任意)**copy running-config startup-config**

手順の詳細

	コマンド	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** *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** [**external-lsa** [*max-metric-value*]] [**include-stub**] [**on-startup** {*seconds* | **wait-for bgp** *tag*}] [**summary-lsa** [*max-metric-value*]] **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	**copy running-config startup-config** **例:** switch(config-router)# copy running-config startup-config	(任意) この設定の変更を保存します。

次に、起動時にスタブ ルータ アドバタイズメントを、デフォルトの 600 秒間イネーブル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
```

ルートのアドミニストレーティブ ディスタンスの設定

OSPFv2 によって RIB に追加されるルートのアドミニストレーティブ ディスタンスを設定できます。

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評価は低くなります。一般的にルートは、複数のルーティング 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、複数のルーティング 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。

はじめる前に

OSPF がイネーブルになっていることを確認します(「OSPFv2 のイネーブル化」セクション (5-16 ページ) を参照)。

この機能に関する注意事項と制約事項については、「OSPFv2 に関する注意事項および制約事項」セクション (5-14 ページ) を参照してください。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. [**no**] **table-map** *map-name*
4. **exit**
5. **route-map** *map-name* [**permit** | **deny**] [*seq*]
6. **match route-type** *route-type*
7. **match ip route-source prefix-list** *name*
8. **match ip address prefix-list** *name*
9. **set distance** *value*
10. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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** *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	[**no**] **table-map** *map-name* **例**: `switch(config-router)# table-map foo`	OSPFv2 ルートを RIB に送信する前に、OSPFv2 ルートをフィルタリングまたは変更するポリシーを設定します。マップ名には最大 63 文字の英数字を入力できます。

ステップ 4	**exit**
	
	例:
	`switch(config-router)# exit`
	`switch(config)#`
		ルータ 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	**route-map** *map-name* [**permit** | **deny**] [*seq*]
	
	例:
	`switch(config)# route-map foo 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 コンフィギュレーション モードを開始します。*seq* を使用して、ルート マップ エントリを順序付けます。
		（注）　**permit** オプションで、ディスタンスを設定することができます。**deny** オプションを使用すると、デフォルトのディスタンスが適用されます。

	コマンド	目的
ステップ 6	**match route-type** *route-type* **例:** `switch(config-route-map)# match route-type external`	次のルート タイプのいずれかと照合します。 • external：外部ルート（BGP、EIGRP、OSPF タイプ 1 または 2） • inter-area：OSPF エリア間ルート • internal：内部ルート（OSPF エリア内またはエリア間ルートを含む） • intra-area：OSPF エリア内ルート • nssa-external：NSSA 外部ルート（OSPF タイプ 1 または 2） • type-1：OSPF 外部タイプ 1 ルート • type-2：OSPF 外部タイプ 2 ルート
ステップ 7	**match ip route-source prefix-list** *name* **例:** `switch(config-route-map)# match ip route-source prefix-list p1`	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またはルータ ID と照合します。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。
ステップ 8	**match ip address prefix-list** *name* **例:** `switch(config-route-map)# match ip address prefix-list p1`	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。
ステップ 9	**set distance** *value* **例:** `switch(config-route-map)# set distance 150`	OSPFv2 のルート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。
ステップ 10	**copy running-config startup-config** **例:** `switch(config-route-map)# copy running-config startup-config`	（任意）この設定の変更を保存します。

次に、OSPFv2 アドミニストレーティブ ディスタンスについて、エリア間ルートを 150、外部ルートを 200、およびプレフィックス リスト p1 内のすべてのプレフィックスを 190 に設定する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# table-map foo
switch(config-router)# exit
switch(config)# route-map foo permit 10
switch(config-route-map)# match route-type inter-area
switch(config-route-map)# set distance 150
switch(config)# route-map foo permit 20
switch(config-route-map)# match route-type external
switch(config-route-map)# set distance 200
switch(config)# route-map foo permit 30
switch(config-route-map)# match ip route-source prefix-list p1
switch(config-route-map)# match ip address prefix-list p1
switch(config-route-map)# set distance 190
```

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 SPF 計算を制御する数多くのタイマーが含まれます。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」セクション（5-7 ページ）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た後に 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」セクション（5-19 ページ）を参照してください。

はじめる前に

OSPF がイネーブルになっていることを確認します（「OSPFv2 のイネーブル化」セクション（5-16 ページ）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*
7. **interface** *type slot/port*
8. **ip ospf hello-interval** *seconds*
9. **ip ospf dead-interval** *seconds*
10. **ip ospf retransmit-interval** *seconds*
11. **ip ospf transmit-delay** *seconds*
12. （任意）**show ip ospf**
13. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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** *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 200`	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 10 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`	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 0 ～ 5000 ミリ秒です。デフォルト値は 0 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-wait* **例**: `switch(config-router)# timers throttle spf 3000 2000 4000`	SPF 最適パス スケジュール初期遅延時間と、各 SPF 最適パス計算間の最小ホールド タイム（秒単位）を設定します。指定できる範囲は 1 ～ 600000 です。デフォルトは、遅延時間なし、およびホールド タイム 5000 ミリ秒です。

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 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	**ip ospf hello-interval** *seconds* **例**: `switch(config-if)# ip ospf hello-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 9	**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。

	コマンド	目的
ステップ 10	**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 11	**ip ospf transmit-delay** *seconds*
	
	例:
	`switch(config-if)# ip ospf transmit-delay 600`
	`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 12	**show ip ospf** **例**: `switch(config-if)# show ip ospf`	（任意）OSPF に関する情報を表示します。
ステップ 13	**copy running-config startup-config** **例**: `switch(config-if)# copy running-config startup-config`	（任意）この設定の変更を保存します。

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
```

グレースフル リスタートの設定

グレースフル リスタートは、デフォルトでイネーブルにされています。OSPFv2 インスタンスのグレースフル リスタートには、省略可能な次のパラメータを設定できます。

- Grace period：グレースフル リスタートの開始後に、ネイバーが隣接関係を解消するまでに待つ時間を設定します。
- Helper mode disabled：ローカル OSPFv2 インスタンスのヘルパー モードをディセーブルにします。OSPFv2 は、ネイバーのグレースフル リスタートには関与しません。
- Planned graceful restart only：予定された再起動の場合にだけグレースフル リスタートがサポートされるように OSPFv2 を設定します。

はじめる前に

OSPF がイネーブルになっていることを確認します（「OSPFv2 のイネーブル化」セクション（5-16 ページ）を参照）。

すべてのネイバーで、一致した省略可能なパラメータ一式とともにグレースフル リスタートが設定されていることを確認します。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **graceful-restart**
4. （任意）**graceful-restart grace-period** *seconds*
5. （任意）**graceful-restart helper-disable**
6. （任意）**graceful-restart planned-only**
7. （任意）**show ip ospf** *instance-tag*
8. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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** *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	**graceful-restart** **例**: `switch(config-router)# graceful-restart`	グレースフル リスタートをイネーブルにします。グレースフル リスタートは、デフォルトでイネーブルにされています。
ステップ 4	**graceful-restart grace-period** *seconds* **例**: `switch(config-router)# graceful-restart grace-period 120`	（任意）猶予期間を秒で設定します。指定できる範囲は 5 ～ 1800 です。デフォルトは 60 秒です。
ステップ 5	**graceful-restart helper-disable** **例**: `switch(config-router)# graceful-restart helper-disable`	（任意）ヘルパー モードをディセーブルにします。この機能は、デフォルトでイネーブルにされています。
ステップ 6	**graceful-restart planned-only** **例**: `switch(config-router)# graceful-restart planned-only`	（任意）予定された再起動時にだけグレースフル リスタートを設定します。
ステップ 7	**show ip ospf** *instance-tag* **例**: `switch(config-if)# show ip ospf 201`	（任意）OSPF 情報を表示します。
ステップ 8	**copy running-config startup-config** **例**: `switch(config)# copy running-config startup-config`	（任意）この設定の変更を保存します。

次に、ディセーブルにされているグレースフル リスタートをイネーブルにし、猶予期間を 120 秒に設定する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# graceful-restart
switch(config-router)# graceful-restart grace-period 120
switch(config-router)# copy running-config startup-config
```

OSPFv2 インスタンスの再起動

OSPFv2 インスタンスを再起動できます。この処理では、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 OSPFv2 インスタンスを設定できます。また、複数の VRF を作成し、各 VRF で同じ OSPFv2 インスタンスまたは複数の OSPFv2 インスタンスを使用することもできます。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。

はじめる前に

OSPF がイネーブルになっていることを確認します（「OSPFv2 のイネーブル化」セクション（5-16 ページ）を参照）。

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-type slot/port*
7. **vrf member** *vrf-name*
8. **ip-address** *ip-prefix/length*
9. **router ospf** *instance-tag* **area** *area-id*
10. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	**vrf context** *vrf-name* **例:** switch(config)# vrf context RemoteOfficeVRF switch(config-vrf)#	新しい VRF を作成し、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	**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 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	**maximum-paths** *paths* **例:** switch(config-router-vrf)# maximum-paths 4	（任意）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。この機能は、ロード バランシングに使用されます。
ステップ 6	**interface** *interface-type slot/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	**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 8	**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 9	**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 10	**copy running-config startup-config** **例:** switch(config)# copy running-config startup-config	（任意）この設定の変更を保存します。

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)# 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

OSPFv2 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 ip ospf [*instance-tag*] [**vrf** *vrf-name*]	1 つまたは複数の OSPF ルーティング インスタンスに関する情報が表示されます。出力には、次のエリア レベル カウントが含まれます。 • このエリア内のインターフェイス: このエリアに追加されたすべてのインターフェイスの数（設定済みインターフェイス）。 • アクティブ インターフェイス: ルータ リンク ステートと SPF であると認識されているすべてのインターフェイスの数（アップ インターフェイス）。 • パッシブ インターフェイス: OSPF パッシブであると認識されているすべてのインターフェイスの数（隣接関係は形成されません）。 • ループバック インターフェイス: すべてのローカル ループバック インターフェイスの数。
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-related インターフェイスの情報を表示します。
show ip ospf lsa-content-changed-list *neighbor-id interface-type number* [**vrf** {*vrf-name* | **all** | **default** | **management**}]	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* [**vrf** {*vrf-name* | **all** | **default** | **management**}]	OSPFv2 リンクステート要求の一覧を表示します。

コマンド	目的
show ip ospf retransmission-list *neighbor-id interface-type number* [**vrf** {*vrf-name* | **all** | **default** | **management**}]	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 集約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 のモニタリング

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* | **isis** *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
 ip router ospf 201 area 0.0.0.10
 ip ospf authentication
 ip ospf authentication-key 0 mypass
```

OSPF RFC 互換モードの例

次に、RFC 1583 互換ルータと互換性を持つように OSPF を設定する例を示します。

（注）　RFC1583 互換の OSPF のみを実行するルータに接続するすべての VRF で、RFC 1583 の互換性を設定する必要があります。

```
switch# configure terminal
switch(config)# feature ospf
switch(config)# router ospf Test1
switch(config-router)# rfc1583compatibility
switch(config-router)# vrf A
switch(config-router-vrf)# rfc1583compatibility
```

その他の関連資料

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（5-51 ページ）
- MIB（5-51 ページ）

関連資料

関連項目	マニュアル タイトル
キーチェーン	*『Cisco Nexus 9000 Series NX-OS Security Configuration Guide』*
IPv6 ネットワーク向け OSPFv3	第 6 章「OSPFv3 の設定」
ルート マップ	第 15 章「Route Policy Manager の設定」

MIB

MIB	MIB のリンク
OSPFv2 に関連する MIB	サポートされている MIB を検索およびダウンロードするには、次の URL にアクセスしてください。 ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

OSPFv3 の設定

この章では、Cisco NX-OS デバイスで IPv6 ネットワーク用の Open Shortest Path First version 3（OSPFv3）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。

OSPFv3 について

OSPFv3 は、IETF リンクステート プロトコル（「概要」セクション（1-1 ページ）を参照）です。OSPFv3 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 イネーブル インターフェイスに送信して、他の OSPFv3 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 OSPFv3 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 イネーブル 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 OSPFv3 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 OSPFv3 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されます（「コンバージェンス」セクション（1-6 ページ）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。

OSPFv3 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。

OSPFv3 は IPv6 をサポートしています。IPv4 向けの OSPF の詳細については、第 5 章「OSPFv2 の設定」を参照してください。

この項では、次のトピックについて取り上げます。

OSPFv3 と OSPFv2 の比較

OSPFv3 プロトコルの大半は OSPFv2 と同じです。OSPFv3 は RFC 2740 に記載されています。

OSPFv3 プロトコルと OSPFv2 プロトコルの重要な相違点は、次のとおりです。

- OSPFv2 を拡張した OSPFv3 では、IPv6 ルーティング プレフィックスとサイズの大きい IPv6 アドレスのサポートを提供しています。
- OSPFv3 の LSA は、アドレスとマスクではなく、プレフィックスとプレフィックス長として表現されます。
- ルータ ID とエリア ID は 32 ビット数で、IPv6 アドレスとは無関係です。
- OSPFv3 では、ネイバー探索およびその他の機能にリンクローカル IPv6 アドレスを使用します。
- OSPFv3 は、IPv6 認証トレーラ（RFC 6506）または IPSec（RFC 4552）を使用できます。ただし、Cisco NX-OS Release 7.0(3)I3(1) 以降では、RFC 6506 はサポートされず、RFC 4552 が部分的にサポートされるだけです。
- OSPFv3 では、LSA タイプが再定義されています。

hello パケット

OSPFv3 ルータは、すべての OSPF イネーブル 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。OSPFv3 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

- 双方向通信
- 指定ルータの選定（「指定ルータ」セクション（6-4 ページ）を参照）

hello パケットには、リンクの OSPFv3 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 OSPFv3 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 OSPFv3 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（「ネイバー」セクション（6-3 ページ）を参照）。

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。

OSPFv3 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）で 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。

ネイバー

ネイバーと見なされるためには、OSPFv3 インターフェイスがリモート インターフェイスとの互換性を持つよう設定されている必要があります。この 2 つの OSPFv3 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」セクション（6-5 ページ）を参照）
- 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 ルータのルータ ID
- 優先度：ネイバー ルータの優先度。プライオリティは、指定ルータの選定（「指定ルータ」セクション（6-4 ページ）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あとに経過した時間を示します。
- リンクローカル IPv6 アドレス：ネイバーのリンクローカル IPv6 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」セクション（6-4 ページ）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。

最初の hello パケットが新規ネイバーから受信されると、そのネイバーは、初期化状態のネイバー テーブルに入力されます。いったん双方向通信が確立されると、ネイバー状態は双方向となります。2 つのインターフェイスが互いのリンクステート データベースを交換するため、次に ExStart および交換状態となります。これらがすべて完了すると、ネイバーは完全な状態へと移行し、これが完全な隣接関係となります。ネイバーがデッド間隔内にいずれかの hello パケットを送信できない場合、ネイバーはダウン状態に移行され、隣接とは見なされなくなります。

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（6-4 ページ）を参照してください。

隣接関係は、OSPFv3 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 ヘッダーが含まれます（「リンクステート データベース」セクション（6-8 ページ）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、OSPFv3 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応じて、OSPFv3 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、OSPFv3 の残りの部分に対してネットワークを代表する場合があります（「エリア」セクション（6-5 ページ）を参照）。DR がダウンした場合、OSPFv3 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、OSPFv3 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。

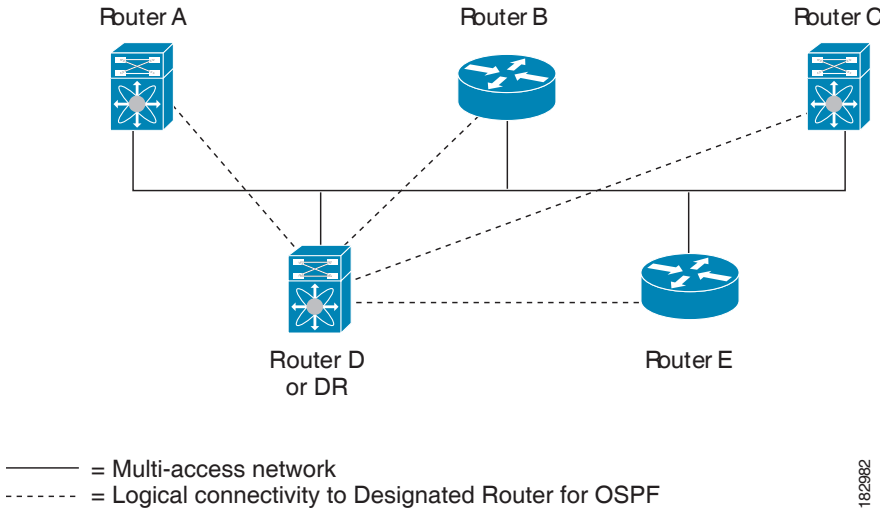
- ポイントツーポイント：2 台のルータ間にのみ存在するネットワーク。ポイントツーポイント ネットワーク上の全ネイバーは隣接関係を確立し、DR は存在しません。
- ブロードキャスト：ブロードキャスト トラフィックが可能なイーサネットなどの共有メディア上で通信できる複数のルータを持つネットワーク。OSPFv3 ルータは DR および BDR を確立し、これらにより、ネットワーク上の LSA フラッディングを制御します。OSPFv3 は、よく知られている IPv6 マルチキャスト アドレス FF02::5 および MAC アドレス 0100.5300.0005 を使用して、ネイバーと通信します。

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 OSPFv3 は、最も大きいルータ ID を DR および BDR として選択します。

他のルータはすべて DR および BDR と隣接関係を確立し、IPv6 マルチキャスト アドレス FF02::6 を使用して、LSA 更新情報を DR と BDR に送信します。図 6-1 は、すべてのルータと DR の間のこの隣接関係を示します。

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 6-1　マルチアクセス ネットワークの DR



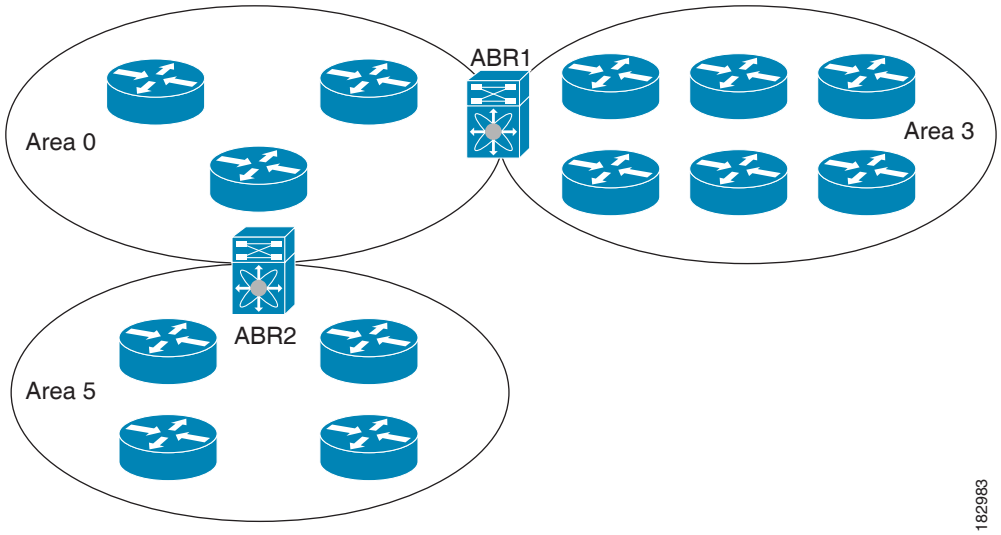
エリア

OSPFv3 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 OSPFv3 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、OSPFv3 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 進表記で表現される 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。

OSPFv3 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（図 6-2を参照）。

図 6-2　OSPFv3 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 エリアにエリア間プレフィックス（タイプ 3）LSA（「ルート集約」セクション（6-12 ページ）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。図 6-2 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。

OSPFv3 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、OSPFv3 エリアを別の自律システム（AS）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。OSPFv3 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」セクション（6-10 ページ）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

OSPFv3 はリンクステート アドバタイズメント（LSA）を使用して、自身のルーティング テーブルを構築します。

この項では、次のトピックについて取り上げます。

- LSA タイプ（6-7 ページ）
- リンク コスト（6-7 ページ）
- フラッディングと LSA グループ ペーシング（6-8 ページ）
- リンクステート データベース（6-8 ページ）

LSA タイプ

表 6-1 は、Cisco NX-OSでサポートされる LSA タイプを示します。

表 6-1　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が含まれますが、プレフィックス情報は含まれません。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 OSPFv3 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が、プレフィックス情報は含まれません。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」セクション (6-4 ページ) を参照してください。
3	エリア間プレフィックス LSA	ABR 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、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」セクション (6-5 ページ) を参照してください。
4	エリア間ルータ 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」セクション (6-5 ページ) 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」セクション (6-5 ページ) を参照してください。
7	タイプ 7 LSA	ASBR が 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。タイプ 7 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」セクション (6-5 ページ) を参照してください。
8	リンク LSA	すべてのルータが、リンクローカル フラッディング スコープを使用して送信する LSA (「フラッディングと LSA グループ ペーシング」セクション (6-8 ページ) を参照)。この LSA には、このリンクのリンクローカル アドレスと IPv6 アドレスが含まれます。
9	エリア内プレフィックス LSA	すべてのルータが送信する LSA。この LSA には、プレフィックスまたはリンク状態へのあらゆる変更が含まれます。エリア内プレフィックス LSA はローカル OSPFv3 エリアにフラッディングされます。この LSA は SPF 再計算をトリガーしません。
11	猶予 LSA	再起動されるルータが、リンクローカル フラッディング スコープを使用して送信する LSA。この LSA は、OSPFv3 のグレースフル リスタートに使用されます。「ハイ アベイラビリティおよびグレースフル リスタート」セクション (6-13 ページ) を参照してください。

リンク コスト

各 OSPFv3 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

OSPFv3 は、LSA タイプに応じて、ネットワークのさまざまな部分に LSA 更新をフラッディングします。OSPFv3 は、次のフラッディング スコープを使用します

- リンク ローカル：LSA は、ローカル リンク上でのみフラッディングされます。リンク LSA および猶予 LSA に使用されます。
- エリアローカル：LSA は、単一の OSPF エリア全体にのみフラッディングされます。ルータ LSA、ネットワーク LSA、エリア間プレフィックス LSAs、エリア間ルータ LSA、およびエリア内プレフィックス LSA に使用されます。
- AS スコープ：LSA は、ルーティング ドメイン全体にフラッディングされます。AS スコープは AS 外部 LSA に使用されます。

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、OSPFv3 エリアの設定により異なります（「エリア」セクション（6-5 ページ）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、OSPFv3 で、複数の LSA を 1 つの OSPFv3 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 10 秒以内の 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 OSPFv3 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、OSPFv3 ネットワーク用のリンクステート データベースを維持しています。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。OSPFv3 は、この情報を使用して、各宛先への最適なパスを計算し、この最適な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」セクション（6-8 ページ）を参照してください。

マルチエリア隣接関係（Multi-Area Adjacency）

OSPFv3 マルチエリア隣接関係により、複数のエリアにあるプライマリ インターフェイス上にリンクを設定できます。このリンクは、それらのエリア内の優先されるエリア内リンクになります。マルチエリア隣接関係では、OSPFv3 エリアにポイントツーポイントの番号なしリンクを確立し、そのエリアにトポロジー パスを提供します。プライマリ隣接関係はリンクを使用して、ネイバー ステートが full の場合に、ルータ LSA で対応するエリアの番号なしポイントツーポイント リンクをアドバタイズします。

マルチエリア インターフェイスは、OSPF の既存のプライマリ インターフェイス上の論理構成体として存在しますが、プライマリ インターフェイス上のネイバー ステートは、マルチエリア インターフェイスと無関係です。マルチエリア インターフェイスはネイバー ルータ上の対応するマルチエリア インターフェイスとの隣接関係を確立します。詳細については、「マルチエリア隣接関係の設定」セクション（6-31 ページ）を参照してください。

OSPFv3 と IPv6 ユニキャスト RIB

OSPFv3 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。選択された各宛先への最短パスが OSPFv3 ルート テーブルに入力されます。OSPFv3 ネットワークが収束すると、このルート テーブルは IPv6 ユニキャスト ルーティング情報ベース（RIB）にデータを提供します。OSPFv3 は IPv6 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 OSPFv3 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（「複数の OSPFv3 インスタンス」セクション（6-14 ページ）を参照）

さらに OSPFv3 は、変更済みダイクストラ アルゴリズムを実行して、エリア間プレフィックス、エリア間ルータ、AS 外部、タイプ 7、およびエリア内プレフィックス（タイプ 3、4、5、7、8）の各 LSA の変更の高速再計算を行います。

アドレス ファミリのサポート

Cisco NX-OS は、ユニキャスト IPv6 やマルチキャスト IPv6 などの複数のアドレス ファミリをサポートしています。アドレス ファミリに特有の OSPFv3 機能は、次のとおりです。

- デフォルト ルート
- ルート集約
- ルートの再配布
- 境界ルータのフィルタ リスト
- SPF 最適化

これらの機能の設定時に IPv6 ユニキャスト アドレス ファミリ コンフィギュレーション モードを開始するには、**address-family ipv6 unicast** コマンドを使用します。

認証

OSPFv3 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。

RFC 4552 は、IPv6認証ヘッダー（AH）または Encapsulating Security Payload（ESP）拡張ヘッダーを使用して OSPFv3 に認証を提供します。Cisco NX-OS 7.0(3)I3(1) 以降では、IPv6 AH ヘッダーを使用して OSPFv3 パケットを認証することで RFC 4552 を部分的にサポートします。

Cisco NX-OS では、IP Security（IPSec）認証方式と Message Digest 5（MD5）または Secure Hash Algorithm 1（SHA1）アルゴリズムを使用した OSPFv3 パケットの認証がサポートされます。OSPFv3 IPSec 認証では、スタティック キーのみがサポートされます。

IPSec 認証は、OSPFv3 プロセス、エリア、またはインターフェイスに対して設定できます。

高度な機能

Cisco NX-OS は、ネットワークでの OSPFv3 の可用性やスケーラビリティを向上させる高度な OSPFv3 機能をサポートしています。

この項では、次のトピックについて取り上げます。

- スタブ エリア（6-10 ページ）
- Not-So-Stubby エリア（6-11 ページ）
- 仮想リンク（6-11 ページ）
- ルートの再配布（6-12 ページ）
- ルート集約（6-12 ページ）
- ハイ アベイラビリティおよびグレースフル リスタート（6-13 ページ）
- 複数の OSPFv3 インスタンス（6-14 ページ）
- SPF 最適化（6-14 ページ）
- BFD（6-14 ページ）
- 仮想化のサポート（6-14 ページ）

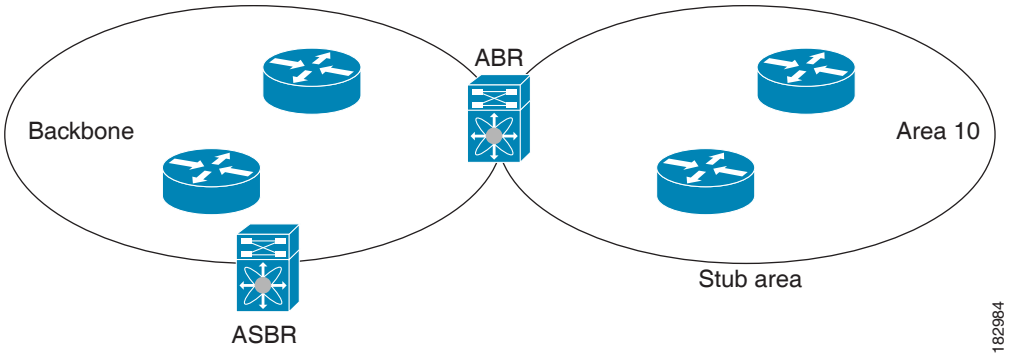
スタブ エリア

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」セクション（6-6 ページ）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」セクション（1-7 ページ）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。

図 6-3 は、外部自律システム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 OSPFv3 自律システムの例を示します。エリア 0.0.0.10 は、スタブ エリアとして設定できます。

図 6-3　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。デフォルト ルートは、プレフィックス長が IPv6 向けに 0 に設定されたエリア間プレフィックス LSA です。

Not-So-Stubby エリア

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、タイプ 7 LSA を生成して NSSA 全体にフラッディングします。または、このタイプ 7 LSA を AS 外部（タイプ 5）LSA に変換するよう、NSSA を他のエリアに接続する ABR を設定することができます。こうすると、ABR は、これらの AS 外部 LSA を OSPFv3 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。タイプ 7 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」セクション（6-6 ページ）を参照してください。

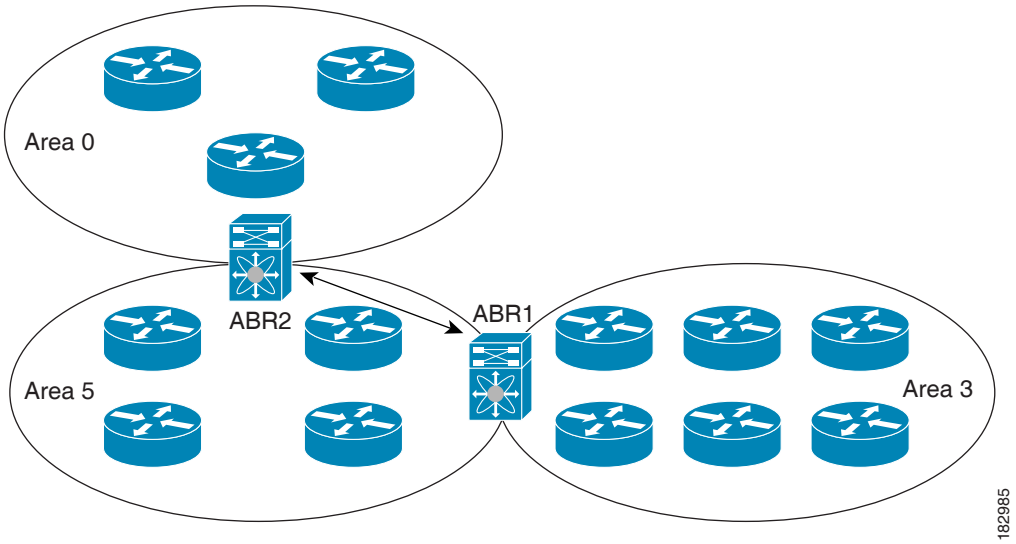
たとえば、OSPFv3 を使用する中央サイトを、異なるルーティング プロトコルを使用するリモート サイトに接続するときに NSSA を使用すると、管理作業を簡素化できます。NSSA を使用する前は、企業サイトの境界ルータとリモート ルータの間の接続を OSPFv3 スタブ エリアとして実行できませんでした。これは、リモート サイトへのルートはスタブ エリア内に再配布できないためです。NSSA を使用すると、企業のルータとリモート ルータ間のエリアを NSSA として定義する（「NSSA の設定」セクション（6-29 ページ）を参照）ことで、OSPFv3 を拡張してリモート接続性をサポートできます。

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、OSPFv3 エリア ABR をバックボーン エリア ABR に接続できます。図 6-4 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 6-4　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

OSPFv3 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」セクション（1-6 ページ）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に割り当てるよう、OSPFv3 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。再配布を指定したルート マップを設定して、どのルートが OSPFv2 に渡されるかを制御する必要があります。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。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 OSPFv3 AS 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。詳細については、第 15 章「Route Policy Manager の設定」を参照してください。

ルート集約

OSPFv3 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、2010:11:22:0:1000::1 と 2010:11:22:0:2000:679:1 を 1 つの集約アドレス 2010:11:22::/32 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ます。

外部ルート集約は、ルート再配布を使用して OSPFv3 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。

ハイ アベイラビリティおよびグレースフル リスタート

Cisco NX-OS では、複数レベルのハイ アベイラビリティ アーキテクチャを提供します。OSPFv3 は、ステートフル リスタートをサポートしています。これは、ノンストップ ルーティング（NSR）とも呼ばれます。OSPFv3 で問題が発生した場合は、以前の実行時状態からの再起動を試みます。この場合、ネイバーはいずれのネイバー イベントも登録しません。最初の再起動が正常ではなく、別の問題が発生した場合、OSPFv3 はグレースフル リスタートを試みます。

グレースフル リスタート、つまり、Nonstop Forwarding（NSF）では、処理の再起動中も OSPFv3 がデータ転送パス上に存在し続けます。OSPFv3 はグレースフル リスタートの実行が必要になると、リンクローカル猶予（タイプ 11）LSA を送信します。この再起動中の OSPFv3 プラットフォームは NSF 対応と呼ばれます。

猶予 LSA には猶予期間が含まれます。猶予期間とは、ネイバー OSPFv3 インターフェイスは再起動中の OSPFv3 インターフェイスからの LSA を待つよう指定された時間です（通常、OSPFv3 は隣接関係を解消し、ダウンした、または再起動中の OSPFv3 インターフェイスが発信するすべての LSA を廃棄します）。関与するネイバーは NSF ヘルパーと呼ばれ、再起動中の OSPFv3 インターフェイスが発信するすべての LSA を、このインターフェイスが隣接したままであるかのように維持します。

再起動中の OSPFv3 インターフェイスが稼働を再開すると、ネイバーを再探索して隣接関係を確立し、LSA 更新情報の送信を再開します。この時点で、NSF ヘルパーは、グレースフル リスタートが完了したと認識します。

ステートフル リスタートは次のシナリオで使用されます。

- プロセスでの問題発生後の最初の回復試行
- **system switchover** コマンドによる手動でのスイッチオーバー

グレースフル リスタートは次のシナリオで使用されます。

- プロセスでの問題発生後の 2 回目の回復試行（4 分以内）
- **restart ospfv3** コマンドによるプロセスの手動での再開
- アクティブ スーパーバイザの削除
- **reload module** *active-sup* コマンドによるアクティブ スーパーバイザのリロード

複数の OSPFv3 インスタンス

Cisco NX-OS は、OSPFv3 プロトコルの複数インスタンスをサポートしています。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。複数のインスタンスが同じ OSPFv3 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。サポートされる OSPFv3 インスタンスの数については、*『Cisco Nexus 9000 Series NX-OS Verified Scalability Guide』*を参照してください。

OSPFv3 ヘッダーには、特定の OSPFv3 インスタンスの OSPFv3 パケットを識別するためのインスタンス ID フィールドが含まれます。この OSPv3 インスタンスを割り当てることができます。インターフェイスは、パケット ヘッダーの OSPFv3 インスタンス ID が一致しない OSPFv3 パケットをすべてドロップします。

Cisco NX-OS では、インターフェイス上に 1 つの OSPFv3 インスタンスのみが許可されます。

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、エリア間プレフィックス（タイプ 3）LSA、および AS 外部（タイプ 5）LSA 用部分 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。

BFD

この機能は、IPv6 の Bidirectional Forwarding Detection（BFD）をサポートします。BFD は、転送パスの障害を高速で検出することを目的にした検出プロトコルです。BFD は 2 台の隣接デバイス間のサブセカンド障害を検出し、BFD の負荷の一部を、サポートされるモジュール上のデータプレーンに分散できるため、プロトコル hello メッセージよりも CPU を使いません。詳細については、*『Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide』*を参照してください。

仮想化のサポート

Cisco NX-OS は、OSPFv3 の複数のプロセス インスタンスをサポートします。各 OSPFv3 インスタンスは、システム制限まで複数の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できます。サポートされる OSPFv3 インスタンスの数については、*『Cisco Nexus 9000 Series NX-OS Verified Scalability Guide』*を参照してください。

OSPFv3 のライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	OSPFv3 には Enterprise Services ライセンスが必要です。Cisco NX-OS ライセンス方式について、およびライセンスの取得方法と適用方法の詳細については、*『Cisco NX-OS Licensing Guide』*を参照してください。

OSPFv3 の前提条件

OSPFv3 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。

- OSPFv3 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。
- スイッチにログオンしている。
- リモート OSPFv3 ネイバーと通信可能な 1 つ以上の IPv6 用インターフェイスが設定されている。
- Enterprise Services ライセンスがインストールされている。
- OSPFv3 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 がイネーブルになっている（「OSPFv3 のイネーブル化」セクション（6-17 ページ）を参照）。
- IPv6 アドレス指定および基本設定に関する詳しい知識がある。IPv6 ルーティングおよびアドレス指定の詳細については、第 3 章「IPv6 の設定」を参照してください。

OSPFv3 の注意事項および制約事項

OSPFv3 には、次の注意事項および制限事項があります。

- 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。
- 仮想ポート チャネル（vPC）環境で OSPFv3 を設定する場合は、コア スイッチ上のルータ コンフィギュレーション モードで次のタイマー コマンドを使用することにより、vPC ピアリンクがシャットダウンしたときに OSPF の高速コンバージェンスを実現します。

```
switch (config-router)# timers throttle spf 1 50 50
switch (config-router)# timers lsa-arrival 10
```

- スケール シナリオでは、インターフェイスと OSPF プロセスのリンク ステート アドバタイズメントの数が大きい場合、OSPF MIB オブジェクトの SNMP エージェントのタイムアウト値が小さい SNMP ウォークは、タイムアウトになると予想されます。OSPF MIB オブジェクトのポーリング中に問い合わせる SNMP エージェントのタイムアウトを確認する場合は、ポーリングする SNMP エージェントのタイムアウト値を増加してください。
- アドミニストレーティブ ディスタンス機能には、次のガイドラインと制限事項が適用されます。
  - OSPF ルートに複数の等コスト パスがある場合、アドミニストレーティブ ディスタンスを設定しても **match ip route-source** コマンドに対しては決定性を持ちません。
  - OSPFv3 ルートに一致したルート ソースに関しては、OSPFv3 のルートの送信元とルータが IPv4 アドレスであるため、**match ip route-source** を **match ipv6 route-source** の代わりに設定する必要があります。
  - アドミニストレーティブ ディスタンスの設定は、**match route-type**、**match ipv6 address prefix-list**、および **match ip route-source prefix-list** コマンドのみでサポートされます。別の match 文は無視されます。
  - 廃棄ルートには、アドミニストレーティブ ディスタンス 220 が常に割り当てられます。テーブル マップの設定は OSPF の廃棄ルートには適用されません。

  - OSPF ルートのアドミニストレーティブ ディスタンスを設定するための **match route-type**、**match ipv6 address**、および **match ip route-source** コマンドにはプリファレンスがありません。このように、Cisco NX-OS OSPF アドミニストレーティブ ディスタンスを設定するためのテーブル マップの動作は、Cisco IOS OSPF の場合と異なります。

- vPC コンフィギュレーション モードで **delay restore** *seconds* コマンドを設定する場合や、マルチシャーシ EtherChannel トランク（MCT）上の VLAN がスイッチ仮想インターフェイス（SVI）を使用して OSPFv2 または OSPFv3 によって通知される場合、これらの SVI は設定された時間の間、vPC セカンダリ ノード上で MAX_LINK_COST で通知されます。その結果、すべてのルートまたはホストのプログラミングは、トラフィックを引き込む前に（セカンダリ vPC ノードのピア リロードで）vPC の同期操作後に完了します。この動作により、ノースサウス トラフィックのパケット損失を最小にできます。

（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。

デフォルト設定値

表 6-2 は、OSPFv3 パラメータのデフォルト設定の一覧です。

表 6-2　デフォルト OSPFv3 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	110
hello 間隔	10 秒
デッド間隔	40 秒
廃棄ルート	イネーブル
グレースフル リスタートの猶予期間	60 秒
グレースフル リスタートの通知期間	15 秒
OSPFv3 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
LSA グループ ペーシング	10 秒
SPF 計算初期遅延時間	0 ミリ秒
SPF 計算ホールド タイム	5000 ミリ秒
SPF 計算初期遅延時間	0 ミリ秒

基本的 OSPFv3 の設定

OSPFv3 は、OSPFv3 ネットワークを設計したあとに設定します。

この項では、次のトピックについて取り上げます。

OSPFv3 のイネーブル化

OSPFv3 を設定する前に、OSPFv3 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 ospfv3**
3. （任意）**show feature**
4. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 ospfv3`
	
	例：
	`switch(config)# feature ospfv3`
		OSPFv3 をイネーブルにします。

ステップ 3	`show feature`
	
	例：
	`switch(config)# show feature`
		（任意）イネーブルおよびディセーブルにされた機能を表示します。

ステップ 4	`copy running-config startup-config`
	
	例：
	`switch(config)# copy running-config startup-config`
		（任意）この設定の変更を保存します。

OSPFv3 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 ospfv3`
 	
例:
`switch(config)# no feature ospfv3`
	OSPFv3 機能をディセーブルにして、関連付けられた設定をすべて削除します。

OSPFv3 インスタンスの作成

OSPFv3 設定の最初のステップは、インスタンス、つまり OSPFv3 インスタンスの作成です。作成した OSPFv3 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。各 OSPFv3 インスタンスには、省略可能な次のパラメータも設定できます。

- Router ID: この OSPFv3 インスタンスのルータ ID を設定します。このパラメータを使用しない場合は、ルータ ID 選択アルゴリズムが使用されます。詳細については、「ルータ ID」セクション（1-5 ページ）を参照してください。
- Administrative distance: ルーティング情報の送信元の信頼性をランク付けします。詳細については、「アドミニストレーティブ ディスタンス」セクション（1-7 ページ）を参照してください。
- Log adjacency changes: OSPFv3 ネイバーの状態が変化するたびにシステム メッセージを作成します。
- 名前のルックアップ: ローカル ホストのデータベースを検索または IPv6 の DNS 名を照会することでホスト名に OSPF ルータ ID を変換します。
- Maximum paths: OSPFv3 が、特定の宛先についてルート テーブルにインストールする同等パスの最大数を設定します。このパラメータは、複数パス間のロード バランシングに使用します。
- Reference bandwidth: ネットワークの算出 OSPFv3 コスト メトリックを制御します。算出コストは、参照帯域幅をインターフェイス帯域幅で割った値です。算出コストは、ネットワークが OSPFv3 インスタンスに追加されるときにリンク コストを割り当てると、無効にすることができます。詳細については、「OSPFv3 でのネットワークの設定」セクション（6-20 ページ）を参照してください。

OSPFv3 インスタンス パラメータの詳細については、
「高度な OSPFv3 の設定」セクション（6-25 ページ）を参照してください。

はじめる前に

OSPFv3 をイネーブルにします（「OSPFv3 のイネーブル化」セクション（6-17 ページ）を参照）。

使用する予定の OSPFv3 インスタンス タグが、このルータ上では使用されていないことを確認します。

インスタンス タグが使用されていないことを確認するには、**show ospfv3** *instance-tag* コマンドを使用します。

OSPFv3 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 ospfv3** *instance-tag*
3. （任意）**router-id** *ip-address*
4. （任意）**show ipv6 ospfv3** *instance-tag*
5. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 ospfv3** *instance-tag*
	
	例:
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 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	**router-id** *ip-address* **例**: `switch(config-router)# router-id 192.0.2.1`	(任意)OSPFv3 ルータ ID を設定します。このドット付き 10 進表記の ID で、この OSPFv3 インスタンスが識別されます。この ID 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	**show ipv6 ospfv3** *instance-tag* **例**: `switch(config-router)# show ipv6 ospfv3 201`	(任意)OSPFv3 情報を表示します。
ステップ 5	**copy running-config startup-config** **例**: `switch(config)# copy running-config startup-config`	(任意) この設定の変更を保存します。

OSPFv3 インスタンスおよび関連するすべての設定を削除するには、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 router ospfv3 *instance-tag* **例**: `switch(config)# no router ospfv3 201`	OSPFv3 インスタンスと、関連付けられたすべての設定を削除します。

（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 OSPFv3 コマンドはいずれも、手動で削除する必要があります。

ルータ コンフィギュレーション モードで、次の OSPFv3 用オプション パラメータを設定できます。

コマンド	目的
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。
passive-interface default **例:** switch(config-router)# passive-interface default	すべての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。このコマンドは、VRF またはインターフェイス コマンド モードの設定によって上書きされます。

アドレス ファミリ コンフィギュレーション モードでは、OSPFv3 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 *number* **例:** switch(config-router-af)# distance 25	この OSPFv3 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。
maximum-paths *paths* **例:** switch(config-router-af)# maximum-paths 4	ルート テーブル内の宛先への同じ OSPFv3 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。範囲は 1 ～ 64 です。デフォルト値は 8 です。

次の例は、OSPFv3 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 ospfv3 201
switch(config-router)# copy running-config startup-config
```

OSPFv3 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、OSPFv3 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（「ネイバー」セクション（6-3 ページ）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） インターフェイスの有効な IPv6 アドレスを設定するまでは、インターフェイス上で OSPFv3 がイネーブルになりません。

はじめる前に

OSPFv3 をイネーブルにします（「OSPFv3 のイネーブル化」セクション（6-17 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ipv6 address** *ipv6-prefix/length*
4. **ipv6 router ospfv3** *instance-tag* **area** *area-id* [**secondaries none**]
5. （任意）**show ipv6 ospfv3** *instance-tag* **interface** *interface-type slot/port*
6. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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** *interface-type 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。

ステップ 3	**ipv6 address** *ipv6-prefix/length* **例:** `switch(config-if)# ipv6 address 2001:0DB8::1/48`	このインターフェイスに IPv6 アドレスを割り当てます。
ステップ 4	**ipv6 router ospfv3** *instance-tag* **area** *area-id* [**secondaries none**] **例:** `switch(config-if)# ipv6 router ospfv3 201 area 0`	OSPFv3 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。
ステップ 5	**show ipv6 ospfv3** *instance-tag* **interface** *interface-type slot/port* **例:** `switch(config-if)# show ipv6 ospfv3 201 interface ethernet 1/2`	（任意）OSPFv3 情報を表示します。
ステップ 6	**copy running-config startup-config** **例:** `switch(config)# copy running-config startup-config`	（任意）この設定の変更を保存します。

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 OSPFv3 パラメータを設定できます。

コマンド	目的
ospfv3 cost *number* **例:** switch(config-if)# ospfv3 cost 25	このインターフェイスの OSPFv3 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。
ospfv3 dead-interval *seconds* **例:** switch(config-if)# ospfv3 dead-interval 50	OSPFv3 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。
ospfv3 hello-interval *seconds* **例:** switch(config-if)# ospfv3 hello-interval 25	OSPFv3 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。
ospfv3 instance *instance* **例:** switch(config-if)# ospfv3 instance 25	OSPFv3 インスタンス ID を設定します。有効な範囲は 0 ～ 255 です。デフォルトは 0 です。インスタンス ID のスコープはリンクローカルです。
ospfv3 mtu-ignore **例:** switch(config-if)# ospfv3 mtu-ignore	OSPFv3 で、ネイバーとのあらゆる IP 最大伝送単位（MTU）不一致が無視されるよう設定します。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。
ospfv3 network {**broadcast** | **point-point**} **例:** switch(config-if)# ospfv3 network broadcast	OSPFv3 ネットワーク タイプを設定します。
[**default** | **no**] **ospfv3 passive-interface** **例:** switch(config-if)# ospfv3 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。このコマンドによって、ルータまたは VRF コマンド モードの設定が上書きされます。**default** オプションは、このインターフェイス モード コマンドを削除して、ルータまたは VRF の設定がある場合にはそれに戻します。
ospfv3 priority *number* **例:** switch(config-if)# ospfv3 priority 25	エリアの DR の決定に使用される OSPFv3 優先度を設定します。有効な範囲は 0 ～ 255 です。デフォルトは 1 です。「指定ルータ」セクション (6-4 ページ) を参照してください。
ospfv3 shutdown **例:** switch(config-if)# ospfv3 shutdown	このインターフェイス上の OSPFv3 インスタンスをシャットダウンします。

次に、OSPFv3 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)# ipv6 address 2001:0DB8::1/48
switch(config-if)# ipv6 ospfv3 201 area 0.0.0.10
switch(config-if)# copy running-config startup-config
```

OSPFv3 IPSec 認証の設定

OSPFv3 IP Security（IPSec）認証は、プロセス、エリア、またはインターフェイスに対して設定できます。

認証設定は、プロセス、エリア、インターフェイス レベルの順に継承されます。認証が 3 つのレベルすべてで設定されている場合は、インターフェイス設定がプロセスおよびエリア設定よりも優先されます。

はじめる前に

OSPFv3 がイネーブルになっていることを確認します（「OSPFv3 のイネーブル化」セクション（6-17 ページ）を参照）。

feature imp コマンドを使用してインターネット メッセージング プログラム（IMP）がイネーブルになっていることを確認します。

手順の概要

1. **configure terminal**
2. **router ospfv3** *instance-tag*
3. **exit**
4. **authentication ipsec spi** *spi auth* [**0** | **3** | **7**] *key*

   または

   area *area* **authentication ipsec spi** *spi auth* [**0** | **3** | **7**] *key*

   または

   interface *interface-type slot/port*

   ospfv3 authentication ipsec spi *spi auth* [**0** | **3** | **7**] *key*
5. （任意）**show ospfv3** *process*
6. （任意）**show ospfv3 interface** *interface-type slot/port*
7. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 ospfv3` *instance-tag*
	
	例:
	`switch(config)# router ospfv3 100`
	`switch(config-router)#`
		新規 OSPFv3 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	`exit`
	
	例:
	`switch(config-router)# exit`
	`switch(config)#`
		OSPFv3 ルータ コンフィギュレーション モードを終了します。

	コマンド	目的
ステップ 4	**authentication ipsec spi** *spi auth* [**0** | **3** | **7**] *key* **例:** `switch(config)# authentication ipsec spi 475 md5 11111111111111112222222222222222`	OSPFv3 IPSec 認証をプロセス（または VRF）レベルで設定します。 *spi* 引数はセキュリティ パラメータ インデックス（SPI）を指定します。範囲は 256 ～ 4294967295 です。 *auth* 引数は認証タイプを指定します。サポートされる値は md5 または sha1 です。 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。7 の場合は、キーを Cisco タイプ 7 暗号化として設定します。 クリアテキスト オプション（0）を使用する場合、*key* 引数の長さは md5 の場合は 32 文字、sha1 の場合は 40 文字でなければなりません。
	area *area* **authentication ipsec spi** *spi auth* [**0** | **3** | **7**] *key* **例:** `switch(config)# area 0 authentication ipsec spi 475 md5 11111111111111112222222222222222`	OSPFv3 IPSec 認証をエリア レベルで設定します。 *spi* 引数はセキュリティ パラメータ インデックス（SPI）を指定します。範囲は 256 ～ 4294967295 です。 *auth* 引数は認証タイプを指定します。サポートされる値は md5 または sha1 です。 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。7 の場合は、キーを Cisco タイプ 7 暗号化として設定します。 クリアテキスト オプション（0）を使用する場合、*key* 引数の長さは md5 の場合は 32 文字、sha1 の場合は 40 文字でなければなりません。 **注**　**area** *area* **authentication disable** コマンドを使用して、OSPFv3 IPSec 認証をエリア レベルでディセーブルにします。
	interface *interface-type slot/port* **ospfv3 authentication ipsec spi** *spi auth* [**0** | **3** | **7**] *key* **例:** `switch(config)# interface ethernet 1/1 switch(config-if)# ospfv3 authentication ipsec spi 475 md5 11111111111111112222222222222222`	指定したインターフェイスに OSPFv3 IPSec 認証を設定します。 *spi* 引数はセキュリティ パラメータ インデックス（SPI）を指定します。範囲は 256 ～ 4294967295 です。 *auth* 引数は認証タイプを指定します。サポートされる値は md5 または sha1 です。 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。7 の場合は、キーを Cisco タイプ 7 暗号化として設定します。 クリアテキスト オプション（0）を使用する場合、*key* 引数の長さは md5 の場合は 32 文字、sha1 の場合は 40 文字でなければなりません。 **注**　**ospfv3 authentication disable** コマンドを使用して、指定したインターフェイスの OSPFv3 IPSec 認証をディセーブルにします。

	コマンド	目的
ステップ 5	**show ospfv3** *process* **例:** `switch(config)# show ospfv3 100`	(任意)プロセス レベルの OSPFv3 認証設定を表示します。
ステップ 6	**show ospfv3 interface** *interface-type slot/port* **例:** `switch(config)# show ospfv3 interface ethernet 1/1`	(任意)インターフェイス レベルの OSPFv3 認証設定を表示します。
ステップ 7	**copy running-config startup-config** **例:** `switch(config)# copy running-config startup-config`	(任意)この設定の変更を保存します。

高度な OSPFv3 の設定

OSPFv3 は、OSPFv3 ネットワークを設計したあとに設定します。

この項では、次のトピックについて取り上げます。

- 境界ルータのフィルタ リストの設定(6-25 ページ)
- スタブ エリアの設定(6-27 ページ)
- Totally Stubby エリアの設定(6-28 ページ)
- NSSA の設定(6-29 ページ)
- マルチエリア隣接関係の設定(6-31 ページ)
- 仮想リンクの設定(6-32 ページ)
- 再配布の設定(6-34 ページ)
- 再配布されるルート数の制限(6-36 ページ)
- ルート集約の設定(6-38 ページ)
- ルートのアドミニストレーティブ ディスタンスの設定(6-40 ページ)
- デフォルト タイマーの変更(6-43 ページ)
- グレースフル リスタートの設定(6-45 ページ)
- OSPFv3 インスタンスの再起動(6-47 ページ)
- 仮想化による OSPFv3 の設定(6-47 ページ)

境界ルータのフィルタ リストの設定

OSPFv3 ドメインを、関連性のある各ネットワークを含む一連のエリアに分離できます。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。OSPFv3 ドメインは、自律システム境界ルータ(ASBR)を介して、外部ドメインにも接続可能です。「エリア」セクション(6-5 ページ)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。詳細については、「ルート集約の設定」セクション（6-38 ページ）を参照してください。
- Filter list：ABR 上で、外部エリアから受信したエリア間プレフィックス（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。

はじめる前に

フィルタ リストが、着信または発信エリア間プレフィックス（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。第 15 章「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 ospfv3** *instance-tag*
3. **address-family ipv6 unicast**
4. **area** *area-id* **filter-list route-map** *map-name* **{in | out}**
5. （任意）**show ipv6 ospfv3 policy statistics area** *id* **filter-list {in | out}**
6. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 ospfv3** *instance-tag*
	
	例：
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 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	**address-family ipv6 unicast**
	
	例：
	`switch(config-router)# address-family ipv6 unicast`
	`switch(config-router-af)#`
		IPv6 ユニキャスト アドレス ファミリ モードを開始します。

ステップ 4	**area** *area-id* **filter-list route-map** *map-name* {**in** | **out**} **例：** `switch(config-router-af)# area 0.0.0.10 filter-list route-map FilterLSAs in`	ABR 上で着信または発信エリア間プレフィックス（タイプ 3）LSA をフィルタリングします。

	コマンド	目的
ステップ 5	**show ipv6 ospfv3 policy statistics area** *id* **filter-list** {**in** | **out**} **例**: switch(config-if)# show ipv6 ospfv3 policy statistics area 0.0.0.10 filter-list in	(任意)OSPFv3 ポリシー情報を表示します。
ステップ 6	**copy running-config startup-config** **例**: switch(config-router)# copy running-config startup-config	(任意)この設定の変更を保存します。

次に、ディセーブルにされているグレースフル リスタートをイネーブルにする例を示します。

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)# area 0.0.0.10 filter-list route-map FilterLSAs in
switch(config-router-af)# copy running-config startup-config
```

スタブ エリアの設定

OSPFv3 ドメインの、外部トラフィックが不要な部分にスタブ 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」セクション(6-10 ページ)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。

はじめる前に

OSPF をイネーブルにします(「OSPFv3 のイネーブル化」セクション(6-17 ページ)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 ospfv3** *instance-tag*
3. **area** *area-id* **stub**
4. (任意)**address-family ipv6 unicast**
5. (任意)**area** *area-id* **default-cost** *cost*
6. (任意)**copy running-config startup-config**

手順の詳細

	コマンド	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 ospfv3** *instance-tag*
	
	例:
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 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	**address-family ipv6 unicast**
	
	例:
	`switch(config-router)# address-family ipv6 unicast`
	`switch(config-router-af)#`
		(任意)IPv6 ユニキャスト アドレス ファミリ モードを開始します。

ステップ 5	**area** *area-id* **default-cost** *cost* **例:** `switch(config-router-af)# area 0.0.0.10 default-cost 25`	(任意)このスタブ エリア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。指定できる範囲は 0 ～ 16777215 です。
ステップ 6	**copy running-config startup-config** **例:** `switch(config-router)# copy running-config startup-config`	(任意)この設定の変更を保存します。

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

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# area 0.0.0.10 stub no-summary
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 エリアを経由しないようにすることができます。

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

OSPFv3 ドメインの、ある程度の外部トラフィックが必要な部分に NSSA を設定できます。「Not-So-Stubby エリア」セクション（6-11 ページ）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 OSPFv3 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：NSSA をバイパスして OSPFv3 AS 内の他のエリアに到達するルートを再配布します。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：目的のルートのみが NSSA および他のエリア全体でフラッディングされるよう、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、タイプ 7 LSA を AS 外部 LSA（タイプ 5）に変換します。再配布されたルートを OSPFv3 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。
- 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。

はじめる前に

OSPF をイネーブルにします（「OSPFv3 のイネーブル化」セクション（6-17 ページ）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 ospfv3** *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）**address-family ipv6 unicast**
5. （任意）**area** *area-id* **default-cost** *cost*
6. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 ospfv3** *instance-tag* **例**: switch(config)# router ospfv3 201 switch(config-router)#	新規 OSPFv3 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	**address-family ipv6 unicast** **例**: switch(config-router)# address-family ipv6 unicast switch(config-router-af)#	(任意)IPv6 ユニキャスト アドレス ファミリ モードを開始します。
ステップ 5	**area** *area-id* **default-cost** *cost* **例**: switch(config-router-af)# area 0.0.0.10 default-cost 25	(任意)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。指定できる範囲は 0 ～ 16777215 です。
ステップ 6	**copy running-config startup-config** **例**: switch(config-router)# copy running-config startup-config	(任意)この設定の変更を保存します。

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 ospfv3 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 ospfv3 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 ospfv3 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、常にタイプ 7 LSA を AS 外部（タイプ 5）LSA に変換する NSSA を作成する例を示します。

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# area 0.0.0.10 nssa translate type 7 always
switch(config-router)# copy running-config startup-config
```

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 ospfv3 201
switch(config-router)# area 0.0.0.10 nssa no-summary
switch(config-router)# copy running-config startup-config
```

マルチエリア隣接関係の設定

既存の OSPFv3 インターフェイスには複数のエリアを追加できます。追加の論理インターフェイスはマルチエリア隣接関係をサポートしています。

はじめる前に

OSPFv3 をイネーブルにします（「OSPFv3 のイネーブル化」セクション（6-17 ページ）を参照）。

インターフェイスにプライマリ エリアが設定されていることを確認します（「OSPFv3 でのネットワークの設定」セクション（6-20 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ipv6 router ospfv3** *instance-tag* **area** *area-id*
4. （任意）**show ipv6 ospfv3** *instance-tag* **interface** *interface-type slot/port*
5. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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** *interface-type 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。
ステップ 3	**ipv6 router ospfv3** *instance-tag* **multi-area** *area-id* **例:** switch(config-if)# ipv6 router ospfv3 201 multi-area 3	別のエリアにインターフェイスを追加します。
ステップ 4	**show ipv6 ospfv3** *instance-tag* **interface** *interface-type slot/port* **例:** switch(config-if)# show ipv6 ospfv3 201 interface ethernet 1/2	(任意)OSPFv3 情報を表示します。
ステップ 5	**copy running-config startup-config** **例:** switch(config)# copy running-config startup-config	(任意)この設定の変更を保存します。

次に、OSPFv3 インターフェイスに別のエリアを追加する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ipv6 address 2001:0DB8::1/48
switch(config-if)# ipv6 ospfv3 201 area 0.0.0.10
switch(config-if)# ipv6 ospfv3 201 multi-area 20
switch(config-if)# 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、中継エリア経由でバックボーン エリアに接続します。「仮想リンク」セクション(6-11 ページ)を参照してください。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。

- 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。

はじめる前に

OSPF をイネーブルにします（「OSPFv3 のイネーブル化」セクション（6-17 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **router ospfv3** *instance-tag*
3. **area** *area-id* **virtual-link** *router-id*
4. （任意）**show ipv6 ospfv3 virtual-link** [**brief**]
5. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 ospfv3** *instance-tag*
	
	例:
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 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 2001:0DB8::1`
	`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	**show ipv6 ospfv3 virtual-link** [**brief**] **例:** `switch(config-if)# show ipv6 ospfv3 virtual-link`	（任意）OSPFv3 仮想リンク情報を表示します。
ステップ 5	**copy running-config startup-config** **例:** `switch(config-router)# copy running-config startup-config`	（任意）この設定の変更を保存します。

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
dead-interval *seconds* **例**: `switch(config-router-vlink)# dead-interval 50`	(任意)OSPFv3 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`	(任意)OSPFv3 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。
retransmit-interval *seconds* **例**: `switch(config-router-vlink)# retransmit-interval 50`	(任意)OSPFv3 再送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`	(任意)OSPFv3 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 2001:0DB8::1)の設定は、次のとおりです。

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# area 0.0.0.10 virtual-link 2001:0DB8::10
switch(config-router)# copy running-config startup-config
```

ABR 2(ルータ ID 2001:0DB8::10)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 2001:0DB8::1
switch(config-router)# 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 OSPFv3 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と、Cisco NX-OS はデフォルトのスタティック ルートも再配布します。

はじめる前に

再配布で使用する、必要なルート マップを作成します。

OSPF をイネーブルにします（「OSPFv3 のイネーブル化」セクション（6-17 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **router ospfv3** *instance-tag*
3. **address-family ipv6 unicast**
4. **redistribute** {**bgp** *id* | **direct** | **isis** *id* | **rip** *id* | **static**} **route-map** *map-name*
5. **default-information originate** [**always**] [**route-map** *map-name*]
6. **default-metric** *cost*
7. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 ospfv3** *instance-tag*
	
	例:
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 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	**address-family ipv6 unicast**
	
	例:
	`switch(config-router)# address-family ipv6 unicast`
	`switch(config-router-af)#`
		IPv6 ユニキャスト アドレス ファミリ モードを開始します。

ステップ 4	**redistribute** {**bgp** *id* | **direct** | **isis** *id* | **rip** *id* | **static**} **route-map** *map-name* **例:** `switch(config-router-af)# 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 OSPFv3 に再配布します。 **注**　スタティック ルートを再配布すると、Cisco NX-OS はデフォルトのスタティック ルートも再配布します。

	コマンド	目的
ステップ 5	**default-information originate** [**always**] [**route-map** map-name] **例:** switch(config-router-af)# default-information-originate route-map DefaultRouteFilter	デフォルトのルートが RIB に存在する場合、この OSPFv3 ドメインにデフォルトの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 ルートの 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。 **注**　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 6	**default-metric** *cost* **例:** switch(config-router-af)# 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。指定できる範囲は 1 ～ 16777214 です。このコマンドは、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 7	**copy running-config startup-config** **例:** switch(config-router)# copy running-config startup-config	(任意) この設定の変更を保存します。

次に、ボーダー ゲートウェイ プロトコル(BGP)を OSPFv3 に再配布する例を示します。

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)# redistribute bgp route-map FilterExternalBGP
switch(config-router-af)# copy running-config startup-config
```

再配布されるルート数の制限

ルート再配布によって、OSPFv3 ルート テーブルに多数のルートを追加できます。外部プロトコルから受け取るルートの数に最大制限を設定できます。OSPFv3 には、再配布されるルート制限を設定するための次のオプションがあります。

- 上限固定:OSPFv3 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。OSPFv3 はそれ以上の再配布されたルートを受け付けません。任意で、最大値のしきい値パーセンテージを設定して、OSPFv3 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:OSPFv3 が最大値に達したときのみ、警告のログを記録します。OSPFv3 は、再配布されたルートを受け入れ続けます。
- 取り消し:OSPFv3 が最大値に達したときに設定した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、OSPFv3 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、OSPFv3 はすべての再配布されたルートを取り消します。OSPFv3 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。

はじめる前に

OSPF をイネーブルにします（「OSPFv3 のイネーブル化」セクション（6-17 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **router ospfv3** *instance-tag*
3. **address-family ipv6 unicast**
4. **redistribute** {**bgp** *id* | **direct** | **isis** *id* | **rip** *id* | **static**} **route-map** *map-name*
5. **redistribute maximum-prefix** *max* [*threshold*] [**warning-only** | **withdraw** [*num-retries timeout*]]
6. （任意）**show running-config ospfv3**
7. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 ospfv3** *instance-tag*
	
	例:
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 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	**address-family ipv6 unicast**
	
	例:
	`switch(config-router)#`
	`address-family ipv6 unicast`
	`switch(config-router-af)#`
		IPv6 ユニキャスト アドレス ファミリ モードを開始します。

ステップ 4	**redistribute** {**bgp** *id* | **direct** | **isis** *id* | **rip** *id* | **static**} **route-map** *map-name*
	
	例:
	`switch(config-router-af)#`
	`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 OSPFv3 に再配布します。

	コマンド	目的
ステップ 5	**redistribute maximum-prefix** *max* [*threshold*] [**warning-only** | **withdraw** [*num-retries timemout*]] **例:** `switch(config-router)# redistribute maximum-prefix 1000 75 warning-only`	OSPFv2 が配布するプレフィックスの最大数を指定します。有効な範囲は 1 ～ 65535 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数のパーセンテージ。 • **warning-only**：プレフィックスの最大数を超えたとき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、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* の範囲は 60 ～ 600 秒です。デフォルトは 300 秒です。
ステップ 6	**show running-config ospfv3** **例:** `switch(config-router)# show running-config ospf`	（任意）OSPFv3 の設定を表示します。
ステップ 7	**copy running-config startup-config** **例:** `switch(config-router)# copy running-config startup-config`	（任意）この設定の変更を保存します。

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 ospfv3 201
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)# redistribute bgp route-map FilterExternalBGP
switch(config-router-af)# redistribute maximum-prefix 1000 75
```

ルート集約の設定

集約されたアドレス範囲を設定して、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集約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。詳細については、「ルート集約」セクション（6-12 ページ）を参照してください。

はじめる前に

OSPF をイネーブルにします（「OSPFv3 のイネーブル化」セクション（6-17 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **router ospfv3** *instance-tag*
3. **address-family ipv6 unicast**
4. **area** *area-id* **range** *ipv6-prefix/length* [**no-advertise**] [**cost** *cost*]

または

5. **summary-address** *ipv6-prefix/length* [**no-advertise**] [**tag** *tag*]
6. （任意）**show ipv6 ospfv3 summary-address**
7. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 ospfv3** *instance-tag*
	
	例:
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 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	**address-family ipv6 unicast**
	
	例:
	`switch(config-router)# address-family ipv6 unicast`
	`switch(config-router-af)#`
		IPv6 ユニキャスト アドレス ファミリ モードを開始します。

ステップ 4	**area** *area-id* **range** *ipv6-prefix/length* [**no-advertise**] [**cost** *cost*] **例**: `switch(config-router-af)# area 0.0.0.10 range 2001:0DB8::/48 advertise`	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エリア間プレフィックス（タイプ 3）LSA にアドバタイズすることもできます。*cost* の範囲は 0 ～ 16777215 です。

ステップ 5	**summary-address** *ipv6-prefix/length* [**no-advertise**][**tag** *tag*]
	例:
	`switch(config-router-af)#`
	`summary-address 2001:0DB8::/48 tag 2`
		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。ルート マップによる再配布で使用できるよう、この集約アドレスにタグを割り当てることもできます。

ステップ 6	**show ipv6 ospfv3 summary-address** **例**: `switch(config-router)# show ipv6 ospfv3 summary-address`	（任意）OSPFv3 集約アドレスに関する情報を表示します。
ステップ 7	**copy running-config startup-config** **例**: `switch(config-router)# copy running-config startup-config`	（任意）この設定の変更を保存します。

次に、ABR 上のエリア間の集約アドレスを作成する例を示します。

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# address-family ipv6 unicast
switch(config-router)# area 0.0.0.10 range 2001:0DB8::/48
switch(config-router)# copy running-config startup-config
```

次に、ASBR 上の集約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)# address-family ipv6 unicast
switch(config-router)# summary-address 2001:0DB8::/48
switch(config-router)# no discard route internal
switch(config-router)# copy running-config startup-config
```

ルートのアドミニストレーティブ ディスタンスの設定

OSPFv3 によって RIB に追加されるルートのアドミニストレーティブ ディスタンスを設定できます。

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評価は低くなります。一般的にルートは、複数のルーティング 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、複数のルーティング 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。

はじめる前に

OSPF がイネーブルになっていることを確認します（「OSPFv3 のイネーブル化」セクション（6-17 ページ）を参照）。

この機能に関する注意事項と制約事項については、「OSPFv3 の注意事項および制約事項」セクション（6-15 ページ）を参照してください。

手順の概要

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **address-family ipv6 unicast**
4. [**no**] **table-map** *map-name*
5. **exit**
6. **exit**
7. **route-map** *map-name* [**permit** | **deny**] [*seq*]
8. **match route-type** *route-type*
9. **match ip route-source prefix-list** *name*
10. **match ipv6 address prefix-list** *name*
11. **set distance** *value*
12. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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** *instance-tag* **例**: switch(config)# router ospf 201 switch(config-router)#	新規 OSPFv3 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	**address-family ipv6 unicast** **例**: switch(config-router)# address-family ipv6 unicast switch(config-router-af)#	IPv6 ユニキャスト アドレス ファミリ モードを開始します。
ステップ 4	[**no**] **table-map** *map-name* **例**: switch(config-router-af)# table-map foo	OSPFv3 ルートを RIB に送信する前に、OSPFv2 ルートをフィルタリングまたは変更するポリシーを設定します。マップ名には最大 63 文字の英数字を入力できます。
ステップ 5	**exit** **例**: switch(config-router-af)# exit switch(config-router)#	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 6	**exit** **例**: switch(config-router)# exit switch(config)#	ルータ 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	**route-map** *map-name* [**permit** | **deny**] [*seq*] **例**: switch(config)# route-map foo 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 コンフィギュレーション モードを開始します。*seq* を使用して、ルート マップ エントリを順序付けます。 **注**　**permit** オプションで、ディスタンスを設定することができます。**deny** オプションを使用すると、デフォルトのディスタンスが適用されます。

	コマンド	目的
ステップ 8	**match route-type** *route-type* **例:** `switch(config-route-map)# match route-type external`	次のルート タイプのいずれかと照合します。 • external：外部ルート（BGP、EIGRP、OSPF タイプ 1 または 2） • inter-area：OSPF エリア間ルート • internal：内部ルート（OSPF エリア内またはエリア間ルートを含む） • intra-area：OSPF エリア内ルート • nssa-external：NSSA 外部ルート（OSPF タイプ 1 または 2） • type-1：OSPF 外部タイプ 1 ルート • type-2：OSPF 外部タイプ 2 ルート
ステップ 9	**match ip route-source prefix-list** *name* **例:** `switch(config-route-map)# match ip route-source prefix-list p1`	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 IPv6 ルート送信元アドレスまたはルータ ID と照合します。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。 **注**　OSPFv3 では、ルータ ID は 4 バイトです。
ステップ 10	**match ipv6 address prefix-list** *name* **例:** `switch(config-route-map)# match ipv6 address prefix-list p1`	1 つまたは複数の IPv6 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。
ステップ 11	**set distance** *value* **例:** `switch(config-route-map)# set distance 150`	OSPFv3 のルート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。
ステップ 12	**copy running-config startup-config** **例:** `switch(config-route-map)# copy running-config startup-config`	（任意）この設定の変更を保存します。

次に、OSPFv3 アドミニストレーティブ ディスタンスについて、エリア間ルートを 150、外部ルートを 200、およびプレフィックス リスト p1 内のすべてのプレフィックスを 190 に設定する例を示します。

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)# table-map foo
switch(config-router)# exit
switch(config)# exit
switch(config)# route-map foo permit 10
switch(config-route-map)# match route-type inter-area
switch(config-route-map)# set distance 150
switch(config)# route-map foo permit 20
switch(config-route-map)# match route-type external
switch(config-route-map)# set distance 200
switch(config)# route-map foo permit 30
switch(config-route-map)# match ip route-source prefix-list p1
switch(config-route-map)# match ipv6 address prefix-list p1
switch(config-route-map)# set distance 190
```

デフォルト タイマーの変更

OSPFv3 には、プロトコル メッセージの動作および SPF 計算を制御する数多くのタイマーが含まれます。OSPFv3 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」セクション（6-8 ページ）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た後に 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、「OSPFv3 でのネットワークの設定」セクション（6-20 ページ）を参照してください。

手順の概要

1. **configure terminal**
2. **router ospfv3** *instance-tag*
3. **timers lsa-arrival** *msec*
4. **timers lsa-group-pacing** *seconds*
5. **timers throttle lsa** *start-time hold-interval max-time*
6. **address-family ipv6 unicast**
7. **timers throttle spf** *delay-time hold-time*
8. **interface** *type slot/port*
9. **ospfv3 retransmit-interval** *seconds*
10. **ospfv3 transmit-delay** *seconds*
11. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 ospfv3** *instance-tag*
	
	例:
	`switch(config)# router ospfv3 201`
	`switch(config-router)#`
		新規 OSPFv3 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 200`	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 10 秒です。
ステップ 5	**timers throttle lsa** *start-time hold-interval max-time* **例**: `switch(config-router)# timers throttle lsa network 350 5000 6000`	LSA 生成のレート制限をミリ秒で設定します。次のタイマーを設定できます。 *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	**address-family ipv6 unicast**
	
	例:
	`switch(config-router)# address-family ipv6 unicast`
	`switch(config-router-af)#`
		IPv6 ユニキャスト アドレス ファミリ モードを開始します。

ステップ 7	**timers throttle spf** *delay-time hold-time* **例**: `switch(config-router)# timers throttle spf 3000 2000`	SPF 最適パス スケジュール初期遅延時間と、各 SPF 最適パス計算間の最小ホールド タイム（秒単位）を設定します。指定できる範囲は 1 ～ 600000 です。デフォルトは、遅延時間なし、およびホールド タイム 5000 ミリ秒です。

ステップ 8	**interface** *type 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。

ステップ 9	**ospfv3 retransmit-interval** *seconds* **例**: `switch(config-if)# ospfv3 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。

	コマンド	目的

ステップ 10	**ospfv3 transmit-delay** *seconds*
	
	例:
	`switch(config-if)# ospfv3 transmit-delay 600`
	`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 11	**copy running-config startup-config** **例:** `switch(config-if)# copy running-config startup-config`	(任意) この設定の変更を保存します。

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
```

グレースフル リスタートの設定

グレースフル リスタートは、デフォルトでイネーブルにされています。OSPFv3 インスタンスのグレースフル リスタートには、省略可能な次のパラメータを設定できます。

- Grace period: グレースフル リスタートの開始後に、ネイバーが隣接関係を解消するまでに待つ時間を設定します。
- Helper mode disabled: ローカル OSPFv3 インスタンスのヘルパー モードをディセーブルにします。OSPFv3 は、ネイバーのグレースフル リスタートには関与しません。
- Planned graceful restart only: 予定された再起動の場合にのみグレースフル リスタートがサポートされるよう、OSPFv3 を設定します。

はじめる前に

OSPFv3 をイネーブルにします(「OSPFv3 のイネーブル化」セクション (6-17 ページ)を参照)。

すべてのネイバーで、一致した省略可能なパラメータ一式とともにグレースフル リスタートが設定されていることを確認します。

手順の概要

1. **configure terminal**
2. **router ospfv3** *instance-tag*
3. **graceful-restart**
4. **graceful-restart grace-period** *seconds*
5. **graceful-restart helper-disable**
6. **graceful-restart planned-only**
7. (任意) **show ipv6 ospfv3** *instance-tag*
8. (任意) **copy running-config startup-config**

手順の詳細

	コマンド	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 ospfv3** *instance-tag* **例**: switch(config)# router ospfv3 201 switch(config-router)#	新規 OSPFv3 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	**graceful-restart** **例**: switch(config-router)# graceful-restart	グレースフル リスタートをイネーブルにします。グレースフル リスタートは、デフォルトでイネーブルにされています。
ステップ 4	**graceful-restart grace-period** *seconds* **例**: switch(config-router)# graceful-restart grace-period 120	猶予期間を秒で設定します。指定できる範囲は 5 ～ 1800 です。デフォルトは 60 秒です。
ステップ 5	**graceful-restart helper-disable** **例**: switch(config-router)# graceful-restart helper-disable	ヘルパー モードをディセーブルにします。デフォルトでは、イネーブルです。
ステップ 6	**graceful-restart planned-only** **例**: switch(config-router)# graceful-restart planned-only	予定された再起動時にのみグレースフル リスタートを設定します。
ステップ 7	**show ipv6 ospfv3** *instance-tag* **例**: switch(config-if)# show ipv6 ospfv3 201	(任意)OSPFv3 情報を表示します。
ステップ 8	**copy running-config startup-config** **例**: switch(config)# copy running-config startup-config	(任意)この設定の変更を保存します。

次に、ディセーブルにされているグレースフル リスタートをイネーブルにし、猶予期間を 120 秒に設定する例を示します。

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# graceful-restart
switch(config-router)# graceful-restart grace-period 120
switch(config-router)# copy running-config startup-config
```

OSPFv3 インスタンスの再起動

OSPv3 インスタンスを再起動できます。この処理では、インスタンスのすべてのネイバーが消去されます。

OSPFv3 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 ospfv3 instance-tag **例:** switch(config)# restart ospfv3 201	OSPFv3 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 OSPFv3 の設定

複数の OSPFv3 インスタンスを設定できます。また、複数の VRF を作成し、各 VRF で同じ OSPFv3 インスタンスまたは複数の OSPFv3 インスタンスを使用することもできます。VRF には OSPFv3 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。

はじめる前に

OSPF をイネーブルにします（「OSPFv3 のイネーブル化」セクション（6-17 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf_name*
3. **router ospfv3** *instance-tag*
4. **vrf** *vrf-name*
5. （任意）**maximum-paths** *paths*
6. **interface** *type slot/port*
7. **vrf member** *vrf-name*
8. **ipv6 address** *ipv6-prefix/length*
9. **ipv6 ospfv3** *instance-tag* **area** *area-id*
10. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	**vrf context** *vrf-name* **例**: switch(config)# vrf context RemoteOfficeVRF switch(config-vrf)#	新しい VRF を作成し、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	**router ospfv3** *instance-tag* **例**: switch(config)# router ospfv3 201 switch(config-router)#	新規 OSPFv3 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 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	**maximum-paths** *paths* **例**: switch(config-router-vrf)# maximum-paths 4	(任意) この VRF のルート テーブル内の宛先への、同じ OSPFv3 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します。
ステップ 6	**interface** *type 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。
ステップ 7	**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 8	**ipv6 address** *ipv6-prefix/length* **例**: switch(config-if)# ipv6 address 2001:0DB8::1/48	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 9	**ipv6 ospfv3** *instance-tag* **area** *area-id* **例**: switch(config-if)# ipv6 ospfv3 201 area 0	設定した OSPFv3 インスタンスおよびエリアに、このインターフェイスを割り当てます。
ステップ 10	**copy running-config startup-config** **例**: switch(config)# copy running-config startup-config	(任意) この設定の変更を保存します。

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)# exit
switch(config)# router ospfv3 201
switch(config-router)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# vrf member NewVRF
switch(config-if)# ipv6 address 2001:0DB8::1/48
switch(config-if)# ipv6 ospfv3 201 area 0
switch(config-if)# copy running-config startup-config
```

OSPFv3 設定の確認

OSPFv3 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 ipv6 ospfv3 [*instance-tag*] [**vrf** *vrf-name*]	1 つまたは複数の OSPFv3 ルーティング インスタンスに関する情報が表示されます。出力には、次のエリア レベル カウントが含まれます。 • このエリア内のインターフェイス: このエリアに追加されたすべてのインターフェイスの数（設定済みインターフェイス）。 • アクティブ インターフェイス: ルータ リンク ステートと SPF であると認識されているすべてのインターフェイスの数（アップ インターフェイス）。 • パッシブ インターフェイス: OSPF パッシブであると認識されているすべてのインターフェイスの数（隣接関係は形成されません）。 • ループバック インターフェイス: すべてのローカル ループバック インターフェイスの数。
show ipv6 ospfv3 border-routers	ABR および ASBR への内部 OSPF ルーティング テーブル エントリを表示します。
show ipv6 ospfv3 database	特定のルータの OSPFv3 データベースに関する情報のリストを表示します。
show ipv6 ospfv3 interface *type number* [**vrf** {*vrf-name* | **all** | **default** | **management**}]	OSPFv3 関連のインターフェイス情報を表示します。
show ipv6 ospfv3 neighbors	ネイバー情報を表示します。**clear ospfv3 neighbors** コマンドを使用すると、すべてのネイバーとの隣接関係を削除できます。
show ipv6 ospfv3 request-list	ルータから要求されている LSA の一覧を表示します。
show ipv6 ospfv3 retransmission-list	再送を待っている LSA の一覧を表示します。

コマンド	目的
show ipv6 ospfv3 summary-address	OSPFv3 インスタンスで設定されている、すべての集約アドレス再配布情報の一覧を表示します。
show ospfv3 *process*	プロセス レベルの OSPFv3 認証設定を表示します。
show ospfv3 interface *interface-type slot/port*	インターフェイス レベルの OSPFv3 認証設定を表示します。
show running-configuration ospfv3	現在実行中の OSPFv3 コンフィギュレーションを表示します。

OSPFv3 のモニタリング

OSPFv3 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 ipv6 ospfv3 memory	OSPFv3 メモリ使用状況の統計情報を表示します。
show ipv6 ospfv3 policy statistics area *area-id* **filter-list** {**in** | **out**} [**vrf** {*vrf-name* | **all** | **default** | **management**}]	エリアの OSPFv3 ルート ポリシー統計情報を表示します。
show ipv6 ospfv3 policy statistics redistribute {**bgp** *id*| **direct** | **isis** *id* | **rip** *id* | **static**} **vrf** {*vrf-name* | **all** | **default** | **management**}]	OSPFv3 ルート ポリシー統計を表示します。
show ipv6 ospfv3 statistics [**vrf** {*vrf-name* | **all** | **default** | **management**}]	OSPFv3 イベント カウンタを表示します。
show ipv6 ospfv3 traffic [*interface-type number*] [**vrf** {*vrf-name* | **all** | **default** | **management**}]	OSPFv3 パケット カウンタを表示します。

OSPFv3 の設定例

次に、OSPFv3 を設定する例を示します。

```
feature ospfv3
router ospfv3 201
 router-id 290.0.2.1

interface ethernet 1/2
 ipv6 address 2001:0DB8::1/48
 ipv6 ospfv3 201 area 0.0.0.10
```

関連項目

次の項目には、OSPF に関する詳細情報が含まれています。

- 第 5 章「OSPFv2 の設定」
- 第 15 章「Route Policy Manager の設定」

その他の関連資料

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。

- MIB（6-51 ページ）

MIB

MIB	MIB のリンク
OSPFv3 に関連する MIB	サポートされている MIB を検索およびダウンロードするには、次の URL にアクセスしてください。 ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

EIGRP の設定

この章では、Cisco NX-OS デバイスで 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 について

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、EIGRP が使用する帯域幅が最小限になります。

この項では、次のトピックについて取り上げます。

EIGRP のコンポーネント

EIGRP には、次の基本コンポーネントがあります。

- Reliable Transport Protocol（7-2 ページ）
- ネイバー探索およびネイバー回復（7-2 ページ）
- 拡散更新アルゴリズム（7-3 ページ）

Reliable Transport Protocol

Reliable Transport Protocol 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」セクション（7-2 ページ）を参照）。Reliable Transport Protocol は、マルチキャスト パケットとユニキャスト パケットの混合伝送をサポートしています。この転送は信頼性が高く、未確認パケットが保留されているときにも、マルチキャスト パケットの迅速な送信が可能です。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。マルチキャスト パケットとユニキャスト 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」セクション（7-16 ページ）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：EIGRP が使用する拡散更新アルゴリズム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」セクション（7-3 ページ）を参照してください。

EIGRP はネイバーへのキープアライブとして、Hello メッセージも使用します。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 または IPv6 アドレス/マスク：この宛先のマスクのネットワーク 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。

この項では、次のトピックについて取り上げます。

- 内部ルート メトリック（7-4 ページ）
- ワイド メトリック（7-4 ページ）
- 外部ルート メトリック（7-5 ページ）
- EIGRP とユニキャスト RIB（7-5 ページ）

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。遅延は 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。

ワイド メトリック

EIGRP は、より高速なインターフェイスまたはバンドルされたインターフェイス上でのルート選択を改善するためのワイド（64 ビット）メトリックをサポートします。ワイド メトリックをサポートしているルータは、次のように、ワイド メトリックをサポートしていないルータと相互運用できます。

- ワイド メトリックをサポートするルータ:ローカル ワイド メトリック値を受信した値に追加し、情報を送信します。
- ワイド メトリックをサポートしないルータ:値を変更せずに受信したメトリックを送信します。

EIGRP は、ワイド メトリックのパス コストを計算するために、次の式を使用します。

メトリック = [k1 × 帯域幅 +（k2 × 帯域幅）/（256 – 負荷）+ k3 × 遅延 + k6 × 拡張属性] x [k5/（信頼性 + k4）]

ユニキャスト RIB が 64 ビットのメトリック値をサポートできないため、EIGRP ワイド メトリックは RIB スケール係数で次の式を使用して、64 ビット メトリック値を 32 ビット値に変換します。

RIB メトリック =（ワイド メトリック/RIB スケール値）。

RIB スケール値は設定可能なパラメータです。

EIGRP ワイド メトリックは、EIGRP メトリックの設定の k6 として、次の 2 種類の新しいメトリック値を導入します。

- ジッタ:（マイクロ秒単位で測定）ルート パス上のすべてのリンクにわたって累積します。ルートの低い方のジッター値は、EIGRP パス選択に優先されます。
- エネルギー:（キロ ビット単位のワットで測定）ルート パス上のすべてのリンクにわたって累積します。ルートの低い方のエネルギー値は、EIGRP パス選択に優先されます。

EIGRP は、より高い値のパスを持つパスよりも、ジッターやエネルギー メトリック値を持たないパス、またはより低いジッターやエネルギー メトリック値を持つパスを優先します。

（注）　EIGRP ワイド メトリックは、TLV バージョン 2 で送信されます。詳細については、「ワイド メトリックの有効化」セクション（7-27 ページ）を参照してください。

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。
- 自律システム番号：宛先の自律システム番号。
- 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、学習したルートをすべて、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。

この項では、次のトピックについて取り上げます。

アドレス ファミリ

EIGRP では、IPv4 と IPv6 の両方のアドレス ファミリをサポートしています。下位互換性を保つために、ルート コンフィギュレーション モードまたは IPV4 アドレス ファミリ モードで EIGRPv4 を設定できます。アドレス ファミリ モードで IPv6 の EIGRP を設定する必要があります。

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。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/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。キーチェーン作成の詳細については、*『Cisco Nexus 9000 Series NX-OS Security Configuration Guide』*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して、ネットワークの安定性を向上させ、リソースの使用を削減し、スタブ ルータ設定を簡素化することができます。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」セクション（1-7 ページ）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 リンク上の通信で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 を使用すると、スタティック ルート、他の EIGRP AS が学習したルート、またはほかのプロトコルからのルートを再配布できます。再配布を指定したルート マップを設定して、どのルートが EIGRP に渡されるかを制御する必要があります。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。第 15 章「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。

ルーティング アップデートからルートをフィルタリングするには、配布リストを使用します。これらのフィルタ処理されたルートは、**ip distribute-list eigrp** コマンドで各インターフェイスに適用されます。

Load Balancing

ロード バランシングを使用すると、ルータによって、宛先アドレスから同じ距離にあるすべてのルータ ネットワーク ポートにトラフィックが分散されます。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、EIGRP ルート テーブルおよびユニキャスト RIB 中の 16 までの等コスト パスを使用する等コスト マルチパス（ECMP）機能をサポートしています。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。

スプリット ホライズン

スプリット ホライズンを使用して、EIGRP が、ルートを伝えたインターフェイスからそのルートをアドバタイズしないようにすることができます。

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。

BFD

この機能は、IPv4 および IPv6 の Bidirectional Forwarding Detection（BFD）をサポートします。BFD は、転送パスの障害を高速で検出することを目的にした検出プロトコルです。BFD は 2 台の隣接デバイス間のサブセカンド障害を検出し、BFD の負荷の一部を、サポートされるモジュール上のデータ プレーンに分散できるため、プロトコル hello メッセージよりも CPU を使いません。詳細については、*『Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide』*を参照してください。

仮想化のサポート

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。

グレースフル リスタートおよびハイ アベイラビリティ

Cisco NX-OS は、EIGRP の無停止フォワーディングおよびグレースフル リスタートをサポートします。

EIGRP の NSF を使用すると、フェールオーバー後に EIGRP ルーティング プロトコル情報が復元される間に、データ パケットを FIB 内の既存のルートで転送できます。ノンストップ フォワーディング（NSF）を使用すると、ピア ネットワーキング デバイスでルーティング フラップが発生することがありません。フェールオーバー時に、データ トラフィックはインテリジェント モジュール経由で転送され、スタンバイ スーパーバイザがアクティブになります。

Cisco NX-OS システムでコールド リブートが発生した場合、デバイスはシステムへのトラフィック転送を中止し、ネットワーク トポロジからシステムを削除します。このシナリオでは、EIGRP でステートレス再起動が発生し、すべてのネイバーが削除されます。Cisco NX-OS はスタートアップ コンフィギュレーションを適用し、EIGRP がネイバーを再検出して、完全な EIGRP ルーティング情報を再度共有します。

Cisco NX-OS を実行するデュアル スーパーバイザ プラットフォームで、ステートフル スーパーバイザ スイッチオーバーが発生します。このスイッチオーバーが発生する前に、EIGRP はグレースフル リスタートを使用して、EIGRP がしばらく使用不可であることを宣言します。スイッチオーバーの間、EIGRP は無停止フォワーディングを使用して FIB の情報に基づいてトラフィックを転送し続け、システムがネットワーク トポロジから取り除かれることはありません。

グレースフル リスタート対応ルータは、Hello メッセージを使用して、グレースフル リスタート動作が開始されたことをネイバーに通知します。グレースフル リスタート認識ルータが、グレースフル リスタート対応ネイバーからグレースフル リスタート動作が進行中であるという通知を受信すると、両方のルータは各トポロジ テーブルをただちに交換します。グレースフル リスタート認識ルータは、ルータの再起動を支援するための次のアクションを実行します。

- ルータは、EIGRP Hello 保持時間を失効し、Hello メッセージにセットされる間隔を短くします。このプロセスにより、グレースフル リスタート認識ルータは再起動中のルータにより早く応答し、再起動中のルータがネイバーを再検出し、トポロジ テーブルを再構築するために必要な時間を短縮します。
- ルータは、ルート保留タイマーを開始します。このタイマーで、グレースフル リスタート認識ルータが、再起動中のネイバー ルータのために既知のルートを保留する時間の長さが設定されます。デフォルトの期間は 240 秒です。
- ルータは、ネイバーが再起動していることをピア リストに記載する、隣接関係を維持する、グレースフル リスタート認識ルータのトポロジ テーブルを送信する準備ができたことを知らせるシグナルをネイバーが送信するか、ルートホールド タイマーが期限切れになるまで再起動中のネイバーを保持する、ということを行います。グレースフル リスタート認識ルータ上でルート保留タイマーの期限が切れた場合、グレースフル リスタート認識ルータは保留ルートを破棄し、再起動中のルータをネットワークに参加する新しいルータとして扱い、隣接関係を再確立します。

スイッチオーバー後に、Cisco NX-OS は実行コンフィギュレーションを適用し、EIGRP は、自身が再び稼働していることをネイバーに通知します。

複数の EIGRP インスタンス

Cisco NX-OS は、同じシステム上で動作する、EIGRP プロトコルの複数インスタンスをサポートしています。すべてのインスタンスで同じシステム 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。サポートされる EIGRP インスタンスの数については、『*Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*』を参照してください。

EIGRP のライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	EIGRP には Enterprise Services ライセンスが必要です。Cisco NX-OS ライセンス方式について、およびライセンスの取得方法と適用方法の詳細については、『*Cisco NX-OS Licensing Guide*』を参照してください。

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 をイネーブルにする必要があります（「EIGRP 機能のイネーブル化」セクション（7-12 ページ）を参照）。

EIGRP に関する注意事項および制限事項

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（第 15 章「Route Policy Manager の設定」を参照）。
- グレースフル スタートについては、NSF 認識ルータが動作中であり、ネットワークで完全に収束している場合にのみ、このルータが NSF 対応ルータのグレースフル リスタート動作を支援できます。
- グレースフル リスタートについては、グレースフル リスタートに関係する隣接デバイスが NSF 認識、または NSF 対応である必要があります。
- Cisco NX-OS 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。
- 1 ギガビット以上のインターフェイス速度の EIGRP ネットワークでの標準メトリックとワイド メトリックの組み合わせは、最適なルーティングになる可能性があります。
- 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 | ipv6} 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 | ipv6} 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、デフォルトで無効となっており、有効にはできません。
- Cisco NX-OS は IP のみをサポートしています。
- ハイ アベイラビリティは、EIGRP 集約タイマーでサポートされません。

（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。

デフォルト設定値

表 7-1 は、各 EIGRP パラメータに対するデフォルト設定を示します。

表 7-1　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	• 帯域幅：100000 Kb/s • 遅延：100（10 マイクロ秒単位） • 信頼性：255 • ロード：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 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
NSF コンバージェンス時間	120
NSF ルート保留時間	240
NSF 信号送信時間	20
再分配	ディセーブル
スプリット ホライズン	イネーブル

基本的 EIGRP の設定

この項では、次のトピックについて取り上げます。

- EIGRP 機能のイネーブル化（7-12 ページ）
- EIGRP インスタンスの作成（7-12 ページ）
- EIGRP インスタンスの再起動（7-15 ページ）
- EIGRP インスタンスのシャットダウン（7-15 ページ）
- EIGRP のパッシブ インターフェイスの設定（7-15 ページ）
- インターフェイスでの EIGRP のシャットダウン（7-16 ページ）

EIGRP 機能のイネーブル化

EIGRP を設定するには、その前に EIGRP 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	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 eigrp`
	
	例:
	`switch(config)# feature eigrp`
		EIGRP 機能をイネーブルにします。

ステップ 3	`show feature`
	
	例:
	`switch(config)# show feature`
		（任意）イネーブルにされた機能の情報を表示します。

ステップ 4	`copy running-config startup-config`
	
	例:
	`switch(config)# copy running-config startup-config`
		（任意）この設定の変更を保存します。

EIGRP 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、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 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」セクション（1-5 ページ）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。

はじめる前に

EIGRP をイネーブルにする必要があります（「EIGRP 機能のイネーブル化」セクション（7-12 ページ）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。

自律システム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、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。IPv6 の場合、この番号は、アドレス ファミリの下で設定する必要があります。

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. （任意）**log-neighbor-warnings** [***seconds***]
6. **interface** ***interface-type slot/port***
7. **{ip | ipv6} router eigrp** ***instance-tag***
8. （任意）**show {ip | ipv6} eigrp interfaces**
9. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 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	`autonomous-system` *as-number*
	
	例:
	`switch(config-router)#`
	`autonomous-system 33`
		（任意）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。

	コマンド	目的
ステップ 4	**log-adjacency-changes** **例**: `switch(config-router)# log-adjacency-changes`	(任意)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`	(任意)ネイバー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 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確認できます。

ステップ 7	{**ip** | **ipv6**} **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 8	**show** {**ip** | **ipv6**} **eigrp interfaces** **例**: `switch(config-if)# show ip eigrp interfaces`	(任意)EIGRP インターフェイスに関する情報を表示します。
ステップ 9	**copy running-config startup-config** **例**: `switch(config)# copy running-config startup-config`	(任意)この設定の変更を保存します。

EIGRP プロセスおよび関連する設定を削除するには、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 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)# interface ethernet 1/2
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」セクション(7-16 ページ)を参照してください。

EIGRP インスタンスの再起動

EIGRP インスタンスは再起動できます。この処理では、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`	(任意)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削除されますが、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
`switch(config-router)#` **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** | **ipv6**} **passive-interface eigrp** *instance-tag* **例**: `switch(config-if)# ip passive-interface eigrp tag10`	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。*instance-tag* 引数には、大文字と小文字が区別される最大 20 文字の任意の英数字文字列を指定できます。

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 をディセーブルにするには、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

`switch(config-if)# {ip | ipv6} eigrp instance-tag shutdown`
 	
例:
`switch(config-router)# ip eigrp Test1 shutdown`
	このインターフェイスで EIGRP をディセーブルにします。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

この項では、次のトピックについて取り上げます。

- EIGRP での認証の設定（7-16 ページ）
- EIGRP スタブ ルーティングの設定（7-18 ページ）
- EIGRP のサマリー集約アドレスの設定（7-19 ページ）
- EIGRP へのルート再配布（7-20 ページ）
- 再配布されるルート数の制限（7-21 ページ）
- EIGRP でのロードバランスの設定（7-23 ページ）
- EIGRP のグレースフル リスタートの設定（7-24 ページ）
- hello パケットとホールド タイムの間隔調整（7-26 ページ）
- スプリット ホライズンのディセーブル化（7-27 ページ）
- ワイド メトリックの有効化（7-27 ページ）
- EIGRP の調整（7-28 ページ）

EIGRP での認証の設定

EIGRP のネイバー間での認証を設定できます。「認証」セクション（7-6 ページ）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優先されます。

はじめる前に

EIGRP をイネーブルにする必要があります（「EIGRP 機能のイネーブル化」セクション（7-12 ページ）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。

この認証設定のためのキー チェーンを作成します。詳細については、『*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*』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 | ipv6} unicast**
4. **authentication key-chain** *key-chain*
5. **authentication mode md5**
6. **interface** *interface-type slot/port*
7. **{ip | ipv6} router eigrp** *instance-tag*
8. **{ip | ipv6} authentication key-chain eigrp** *instance-tag key-chain*
9. **{ip | ipv6} authentication mode eigrp** *instance-tag* **md5**
10. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 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** | **ipv6**} **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。

	コマンド	目的

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 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、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	{**ip** | **ipv6**} **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 8	{**ip** | **ipv6**} **authentication key-chain eigrp** *instance-tag key-chain* **例**: `switch(config-if)# ip authentication key-chain eigrp Test1 routeKeys`	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。
ステップ 9	{**ip** | **ipv6**} **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 10	**copy running-config startup-config** **例**: `switch(config)# copy running-config startup-config`	(任意) この設定の変更を保存します。

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)# 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
`switch(config-router-af)#` **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 ipv6 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 のサマリー集約アドレスの設定

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。より具体的なルートがルーティング テーブルにある場合、EIGRP は、より具体的なすべてのルートの最小に等しいメトリックを持つ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。「ルート集約」セクション（7-7 ページ）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
switch(config-if)# {**ip** | **ipv6**} **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。

この例は、EIGRP がネットワーク 192.0.2.0 をイーサネット 1/2 だけに集約するようにする方法を示しています。

```
switch(config)# interface ethernet 1/2
switch(config-if)# ip summary-address eigrp Test1 192.0.2.0 255.255.255.0
```

EIGRP へのルート再配布

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。

はじめる前に

EIGRP をイネーブルにする必要があります（「EIGRP 機能のイネーブル化」セクション（7-12 ページ）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。第 15 章「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 eigrp** *instance-tag*
3. **address-family {ipv4 | ipv6} unicast**
4. **redistribute {bgp** *as* **| {eigrp | isis | ospf | ospfv3 | rip}** *instance-tag* **| direct | static} route-map** *name*
5. **default-metric** *bandwidth delay reliability loading mtu*
6. （任意）**show {ip | ipv6} eigrp route-map statistics redistribute**
7. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 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 | ipv6} 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。

	コマンド	目的
ステップ 4	**redistribute** {**bgp** *as*| {**eigrp** | **isis** | **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。 • 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** | **ipv6**} **eigrp route-map statistics redistribute** **例**: `switch(config-router-af)# show ip eigrp route-map statistics redistribute bgp`	（任意）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	**copy running-config startup-config** **例:** `switch(config)# copy running-config startup-config`	（任意）この設定の変更を保存します。

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に最大制限を設定できます。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。任意で、最大値のしきい値パーセンテージを設定して、EIGRP 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：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 が最大値に達したときにタイムアウト期間を開始します。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。

はじめる前に

EIGRP をイネーブルにする必要があります（「EIGRP 機能のイネーブル化」セクション（7-12 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. **redistribute {bgp** *id* **| direct | eigrp** *id* **| isis** *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	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 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 インスタンスを作成します。

ステップ 3	**redistribute** {**bgp** *id* | **direct** | **eigrp** *id* | **isis** *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 が配布するプレフィックスの最大数を指定します。有効な範囲は 1 ～ 65535 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数のパーセンテージ。 • **warning-only**：プレフィックスの最大数を超えたとき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 eigrp redistribution** コマンドは、すべてのルートが取り消された場合に使用します。

	コマンド	目的
ステップ 5	**show running-config eigrp** **例:** `switch(config-router)# show running-config eigrp`	(任意)EIGRP の設定を表示します。
ステップ 6	**copy running-config startup-config** **例:** `switch(config-router)# copy running-config startup-config`	(任意)この設定の変更を保存します。

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。「EIGRP でのロードバランスの設定」セクション(7-23 ページ)を参照してください。

はじめる前に

EIGRP をイネーブルにする必要があります(「EIGRP 機能のイネーブル化」セクション(7-12 ページ)を参照)。

手順の概要

1. **configure terminal**
2. **router eigrp** ***instance-tag***
3. **address-family {ipv4 | ipv6} unicast**
4. **maximum-paths** ***num-paths***
5. (任意)**copy running-config startup-config**

手順の詳細

	コマンド	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 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** | **ipv6**} **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	**copy running-config startup-config** **例:** switch(config-router-af)# copy running-config startup-config	(任意) この設定の変更を保存します。

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
```

EIGRP のグレースフル リスタートの設定

EIGRP のグレースフル リスタートまたは NSF を設定できます。「グレースフル リスタートおよびハイ アベイラビリティ」セクション (7-8 ページ) を参照してください。

(注) デフォルトでは、グレースフル リスタートはイネーブルです。

はじめる前に

EIGRP をイネーブルにする必要があります(「EIGRP 機能のイネーブル化」セクション (7-12 ページ)を参照)。

NSF 認識ルータが動作中であり、ネットワークで完全に収束している場合にのみ、このルータが NSF 対応ルータのグレースフル リスタート動作を支援できます。

グレースフル リスタートに関与するネイバー デバイスが NSF 認識または NSF 対応である必要があります。

手順の概要

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. **address-family {ipv4 | ipv6} unicast**
4. **graceful-restart**
5. **timers nsf converge** *seconds*
6. **timers nsf route-hold** *seconds*
7. **timers nsf signal** *seconds*
8. (任意)**copy running-config startup-config**

手順の詳細

	コマンド	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 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** | **ipv6**} **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	**graceful-restart** **例:** `switch(config-router-af)# graceful-restart`	グレースフル リスタートをイネーブルにします。この機能は、デフォルトでイネーブルにされています。

	コマンド	目的
ステップ 5	**timers nsf converge** *seconds* **例**: `switch(config-router-af)# timers nsf converge 100`	スイッチオーバー後にコンバージェンスするまでの制限時間を設定します。範囲は 60 ～ 180 秒です。デフォルトは 120 です。
ステップ 6	**timers nsf route-hold** *seconds* **例**: `switch(config-router-af)# timers nsf route-hold 200`	グレースフル リスタート認識ピアから学習したルートのホールド タイムを設定します。範囲は 20 ～ 300 秒です。デフォルトは 240 です。
ステップ 7	**timers nsf signal** *seconds* **例**: `switch(config-router-af)# timers nsf signal 15`	グレースフル リスタートの信号を送信する時間制限を設定します。範囲は 10 ～ 30 秒です。デフォルトは 20 です。
ステップ 8	**copy running-config startup-config** **例**: `switch(config-router-af)# copy running-config startup-config`	(任意) この設定の変更を保存します。

次に、デフォルト タイマー値を使用して IPv6 上で EIGRP のグレースフル リスタートを設定する例を示します。

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# address-family ipv6 unicast
switch(config-router-af)# graceful-restart
switch(config-router-af)# copy running-config startup-config
```

hello パケットとホールド タイムの間隔調整

各 Hello メッセージの間隔とホールド タイムを調整できます。

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、送信者が有効であると見なすまでの時間をネイバーに示します。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
`switch(config-if)# {`**ip** | **ipv6**`}` **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
switch(config-if)# {**ip** | **ipv6**} **hold-time eigrp** *instance-tag seconds* **例**: switch(config-if)# ipv6 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 ホライズンのディセーブル化

スプリット ホライズンを使用して、ルート情報がルータにより、その情報の送信元インターフェイスの外部にアドバタイズされないようにすることができます。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 デバイス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イネーブルになっています。

スプリット ホライズンをディセーブルにするには、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
switch(config-if)# **no** {**ip** | **ipv6**} **split-horizon eigrp** *instance-tag* **例**: switch(config-if)# no ip split-horizon eigrp Test1	スプリット ホライズンをディセーブルにします。

ワイド メトリックの有効化

ワイド メトリックをイネーブルにするには、ルータ コンフィギュレーション モードまたはアドレス 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
switch(config-router)# **metrics version 64bit** **例**: switch(config-router)# metrics version 64bit	64 ビット メトリック値を有効にします。

オプション選択で RIB のスケール係数を設定するには、ルータ コンフィギュレーション モードまたはアドレス 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

`switch(config-router)# metrics rib-scale value`
 	
例:
`switch(config-router)# metrics rib-scale 128`
	(任意)RIB の 64 ビットのメトリック値を 32 ビットに変換するために使用されるスケール係数を設定します。範囲は 1 ～ 255 です。デフォルト値は 128 です。

EIGRP の調整

省略可能なパラメータを設定して、EIGRP をネットワークに合わせて調整できます。

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 max-hops *hop-count* **例:** `switch(config-router-af)# metric max-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。

コマンド	目的
metric weights *tos k1 k2 k3 k4 k5 k6* **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。 メトリック = [k1 × 帯域幅 + (k2 × 帯域幅)/(256 – 負荷) + k3 × 遅延 + k6 × 拡張属性] x [k5/(信頼性 + 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。 • k6:0。有効な範囲は 0 ～ 255 です。
nsf await-redist-proto-convergence **例**: `switch(config-router-af)# nsf await-redist-proto-convergence`	ノンストップ フォワーディング(NSF)中、EIGRP は再配布されたプロトコルのコンバージェンスを待機してからルーティング情報ベース(RIB)に固有のルートをインストールします。 このコマンドは、スイッチオーバーで、NSF が進行中であり、EIGRP が BGP のコンバージェンスを待ってからそのルートをインストールするようにしたい場合に便利です。この場合、BGP のコンバージェンスが行われて EIGRP が宛先への代替パスを探す前に EIGRP が一時的なルートをインストールし、転送情報ベース(FIB)エントリを変更する必要がなくなります。 **注**　EIGRP と BGP の間で相互再配布を設定するときに(たとえば PE-CE 環境で)このコマンドを使用すると、トラフィック損失が発生する可能性があります。これは、BGP ルートが使用可能になるまで、プロバイダー エッジ(PE)ルータは RIB に EIGRP ルートをインストールしないためです。この動作により、カスタマー エッジ(CE)ルータが EIGRP から学習し、ピア PE ルータにアドバタイズするルートが遅延します。
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** | **ipv6**} **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** | **ipv6**} **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** | **ipv6**} **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** | **ipv6**} **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** | **ipv6**} **next-hop-self eigrp** *instance-tag* **例:** `switch(config-if)# ipv6 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。
{**ip** | **ipv6**} **offset-list eigrp** *instance-tag* {**prefix-list** *name*| **route-map** name} {**in** | **out**} *offset* **例:** `switch(config-if)# ip offset-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** | **ipv6**} **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 EIGRP の processe のを設定し、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。

はじめる前に

EIGRP をイネーブルにする必要があります（「EIGRP 機能のイネーブル化」セクション（7-12 ページ）を参照）。

VRF を作成します。

手順の概要

1. **configure terminal**
2. **vrf context** ***vrf-name***
3. **router eigrp** ***instance-tag***
4. **interface ethernet** ***slot/port***
5. **vrf member** ***vrf-name***
6. **{ip | ipv6} router eigrp** ***instance-tag***
7. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	`vrf context` *vrf-name*
	
	例:
	`switch(config)# vrf context RemoteOfficeVRF`
	`switch(config-vrf)#`
		新しい VRF を作成し、VRF コンフィギュレーション モードを開始します。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 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。

	コマンド	目的

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確認できます。

ステップ 5	**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 6	{**ip** | **ipv6**} **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 7	**copy running-config startup-config** **例**: `switch(config-if)# copy running-config startup-config`	(任意) この設定の変更を保存します。

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)# interface ethernet 1/2
switch(config-if)# ip router eigrp Test1
switch(config-if)# vrf member NewVRF
switch(config-if)# copy running-config startup-config
```

EIGRP 設定の確認

EIGRP 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 {**ip** | **ipv6**} **eigrp** [*instance-tag*]	設定した EIGRP プロセスの要約を表示します。
show {**ip** | **ipv6**} **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** | **ipv6**} **eigrp** *instance-tag* **neighbors** [*type number*] [**detail**]	すべての EIGRP ネイバーに関する情報を表示します。EIGRP ネイバー設定を確認するには、次のコマンドを使用します。
show {**ip** | **ipv6**} **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。

コマンド	目的
show {**ip** | **ipv6**} **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 のモニタリング

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** | **ipv6**} **eigrp** [*instance-tag*] **accounting** [**vrf** *vrf-name*]	EIGRP の課金統計情報を表示します。
show {**ip** | **ipv6**} **eigrp** [*instance-tag*] **route-map statistics redistribute**	EIGRP の再配布統計情報を表示します。
show {**ip** | **ipv6**} **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
 ip address 192.0.2.55/24
 ip router eigrp Test1
  no shutdown
router eigrp Test1
  router-id 192.0.2.1
```

次に、EIGRP ピアから動的に受信した（または EIGRP ピアへアドバタイズした）ルートをフィルタリングするために、**distribute-list** コマンドでルート マップを使用する例を示します。例では、EIGRP の外部プロトコル メトリック ルートを、有効な偏差の 100、BGP のソース プロトコル、および自律システム 45000 と照合するための、ルート マップの設定をします。2 つの match 句が true の場合、対象のルーティング プロトコルのタグ値が 5 に設定されます。ルート マップを使用して、着信パケットを EIGRP プロセスへ配布します。

```
switch(config)# route-map metric-range
switch(config-route-map)# match metric external 500 +- 100
switch(config-route-map)# match source-protocol bgp 45000
switch(config-route-map)# set tag 5
switch(config-route-map)# exit
switch(config)# router eigrp 1
switch(config-router)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# ip address 172.16.0.0
switch(config-if)# ip router eigrp 1
switch(config-if)# ip distribute-list eigrp 1 route-map metric-range in
```

次の例は、EIGRP トポロジ テーブルに許可される前に、ルート マップでフィルタリングされるルーティング テーブルから再配布されるルートが受け入れられるよう、**redistribute** コマンドでルート マップを使用する方法を示します。この例は、EIGRP ルートを、110、200、または 700 ～ 800 の範囲のメトリックと照合するために、ルート マップを設定する方法を示しています。この match 句が true の場合、対象のルーティング プロトコルのタグ値が 10 に設定されます。ルート マップを使用して、EIGRP パケットを再配布します。

```
switch(config)# route-map metric-eigrp
switch(config-route-map)# match metric 110 200 750 +- 50
switch(config-route-map)# set tag 10
switch(config-route-map)# exit
switch(config)# router eigrp 1
switch(config-router)# redistribute eigrp route-map metric-eigrp
switch(config-router)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# ip address 172.16.0.0
switch(config-if)# ip router eigrp 1
```

関連項目

ルート マップの詳細については、第 15 章「Route Policy Manager の設定」、を参照してください。

その他の関連資料

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（7-34 ページ）
- MIB（7-34 ページ）

関連資料

関連項目	参照先
http://www.cisco.com/warp/public/103/1.html	『*Introduction to EIGRP Tech Note*』
http://www.cisco.com/en/US/tech/tk365/technologies_q_and_a_item09186a008012dac4.shtml	EIGRP Frequently Asked Questions

MIB

MIB	MIB のリンク
EIGRP に関連する MIB	サポートされている MIB を検索およびダウンロードするには、次の URL にアクセスしてください。 ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

IS-IS の設定

この章では、Cisco NX-OS デバイスの Integrated Intermediate System-to-Intermediate System（IS-IS）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。

IS-IS について

IS-IS は、ISO（国際標準化機構）/IEC（国際電気標準化会議）10589 に基づく IGP です。Cisco NX-OS インターネット プロトコル バージョン 4（IPv4）および IPv6 をサポートします。IS-IS はネットワーク トポロジの変化を検出し、ネットワーク上の他のノードへのループフリー ルートを計算できる、ダイナミック リンクステート ルーティング プロトコルです。各ルータは、ネットワークの状態を記述するリンクステート データベースを維持し、設定された各リンクにパケットを送信してネイバーを検出します。IS-IS はネットワークを介して各ネイバーにリンクステート情報をフラッディングします。ルータもすべての既存ネイバーを通じて、リンクステート データベースのアドバタイズメントおよびアップデートを送信します。

この項では、次のトピックについて取り上げます。

IS-IS の概要

IS-IS は、設定されている各インターフェイスに hello パケットを送信し、IS-IS ネイバー ルータを検出します。hello パケットには認証、エリア、サポート対象プロトコルなど、受信側インターフェイスが発信側インターフェイスとの互換性を判別するために使用する情報が含まれます。また、一致する最大転送ユニット（MTU）設定を持つインターフェイスだけを使用して IS-IS が隣接関係を確立できるように、hello パケットがパディングされます。互換インターフェイスは隣接関係を形成し、リンクステート アップデート メッセージ（LSP）を使用して、リンクステート データベースのルーティング情報をアップデートします。ルータはデフォルトで、10 分間隔で定期的に LSP リフレッシュを送信し、LSP は 20 分間（LSP ライフタイム）リンクステート データベースに残ります。LSP ライフタイムが終了するまでにルータが LSP リフレッシュを受信しなかった場合、ルータはデータベースから LSP を削除します。

LSP 間隔は、LSP ライフタイムより短くする必要があります。そうしないと、リフレッシュ前に LSP がタイムアウトします。

IS-IS は、隣接ルータに定期的に hello パケットを送信します。hello パケットに対して一時モードを設定すると、IS-IS が隣接関係を確立する前に使用された余分なパディングがこれらの hello パケットに含まれなくなります。隣接ルータの MTU 値が変更された場合、IS-IS はこの変更を検出し、パディングされた hello パケットを一定期間送信できます。IS-IS はこの機能を使用して、隣接ルータ上の一致しない MTU 値を検出します。詳細については、「hello パディングの一時モードの設定」セクション（8-19 ページ）を参照してください。

IS-IS エリア

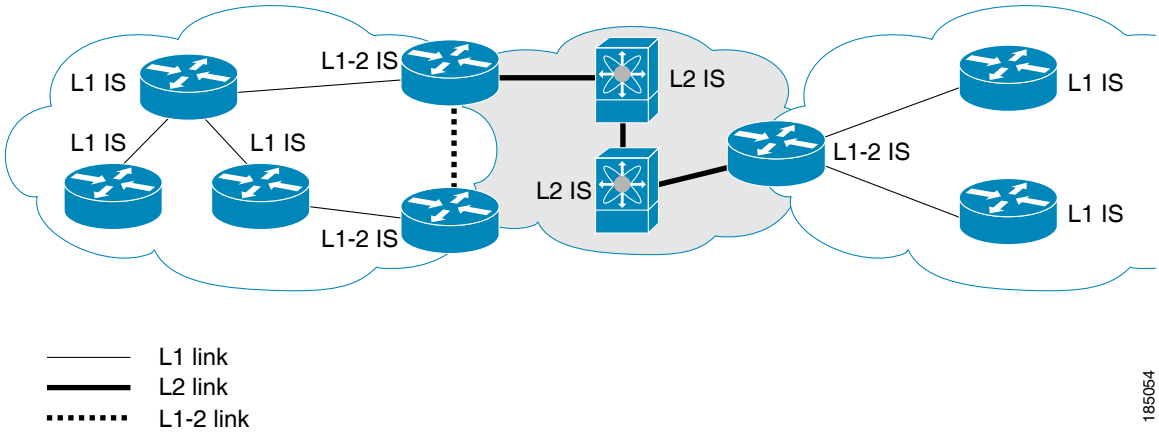
IS-IS ネットワークは、ネットワーク内のすべてのルータが含まれるシングル エリアとして設計することも、バックボーンまたはレベル 2 エリアに接続する複数のエリアとして設計することもできます。非バックボーン エリアのルータはレベル 1 ルータで、ローカル エリア内で隣接関係を確立します（エリア内ルーティング）。レベル 2 エリアのルータは、他のレベル 2 ルータと隣接関係を確立し、レベル 1 エリア間のルーティングを実行します（エリア間ルーティング）。1 つのルータにレベル 1 エリアとレベル 2 エリアの両方を設定できます。これらのレベル 1/レベル 2 ルータは、エリア境界ルータとして動作し、ローカル エリアからレベル 2 バックボーン エリアに情報をルーティングします（図 8-1を参照）。

レベル 1 エリア内のルータは、そのエリア内の他のすべてのルータに対する到達方法を認識します。レベル 2 ルータは、他のエリア境界ルータおよび他のレベル 2 ルータへの到達方法を認識します。レベル 1/レベル 2 ルータは 2 つのエリアの境界にまたがり、レベル 2 バックボーン エリアとの間で双方向にトラフィックをルーティングします。レベル 1/レベル 2 ルータはレベル 1 ルータの Attached（ATT）ビット信号を使用して、レベル 2 エリアに接続するため、このレベル 1/レベル 2 ルータへのデフォルト ルートを設定します。

エリア内に 2 台以上のレベル 1/レベル 2 ルータがある場合など、場合によっては、レベル 1 ルータがレベル 2 エリアへのデフォルト ルートとして使用するレベル 1/レベル 2 ルータを制御することもできます。Attached ビットを設定するレベル 1/レベル 2 ルータを設定できます。詳細については、「IS-IS 設定の確認」セクション（8-30 ページ）を参照してください。

Cisco NX-OS の IS-IS インスタンスは、レベル 1 またはレベル 2 エリアを 1 つだけサポートするか、またはそれぞれのエリアを 1 つずつサポートします。デフォルトでは、すべての IS-IS インスタンスが自動的にレベル 1 およびレベル 2 ルーティングをサポートします。

図 8-1　エリアに分割された IS-IS ネットワーク



ASBR（自律システム境界ルータ）は、IS-IS AS（自律システム）全体に外部宛先をアドバタイズします。外部ルートは、他のプロトコルから IS-IS に再配布されたルートです。

NET およびシステム ID

IS-IS インスタンスごとに NET が関連付けられています。NET は、その IS-IS インスタンスをエリア内で一意に特定する IS-IS システム ID とエリア ID からなります。たとえば、NET が 47.0004.004d.0001.0001.0c11.1111.00 の場合、システム ID は 0000.0c11.1111.00、エリア ID は 47.0004.004d.0001 です。

DIS

IS-IS はブロードキャスト ネットワーク内で代表中継システム（designated intermediate system）を使用し、各ルータがブロードキャスト ネットワーク上の他のすべてのルータと不要なリンクを形成することがないようにします。IS-IS ルータは DIS に LSP を送信し、DIS がブロードキャスト ネットワークのあらゆるリンクステート情報を管理します。エリア内で DIS を選択するために IS-IS に使用させる IS-IS プライオリティをユーザ側で設定できます。

（注）　ポイントツーポイント ネットワークでは DIS は不要です。

IS-IS 認証

隣接関係および LSP 交換を制御するために、認証を設定できます。ネイバーになろうとするルータは、設定されている認証レベルの同じパスワードを交換する必要があります。パスワードが無効なルータは、IS-IS によってブロックされます。IS-IS 認証はグローバルに設定することも、レベル 1、レベル 2、またはレベル 1/レベル 2 両方のルーティングに対応する個々のインターフェイスに設定することもできます。

IS-IS がサポートする認証方式は、次のとおりです。

- クリア テキスト：交換するすべてのパケットで、クリアテキストの 128 ビット パスワードが伝送されます。
- MD5 ダイジェスト：交換するすべてのパケットで、128 ビット キーに基づくメッセージ ダイジェストが伝送されます。

受動的攻撃から保護するために、IS-IS はネットワークを介してクリア テキストとして MD5 秘密キーを送信します。また、リプレイ アタックから保護するために、IS-IS は各パケットにシーケンス番号を組み込みます。

hello および LSP 認証用のキーチェーンも使用できます。キーチェーン管理の詳細については、*『Cisco Nexus 9000 Series NX-OS Security Configuration Guide』*を参照してください。

メッシュ グループ

メッシュ グループは、一連のインターフェイスであり、それらのインターフェイスを介して到達可能なすべてのルータは、他の各ルータとの間に 1 つ以上のリンクがあります。多数のリンクで障害が発生しても、ネットワークから 1 つまたは複数のルータが切り離されることはありません。

通常のフラッディングでは、新しい LSP を受信したインターフェイスは、その LSP をルータ上の他のすべてのインターフェイスにフラッディングします。メッシュ グループを使用する場合、メッシュ グループに含まれているインターフェイスは新しい LSP を受信しても、メッシュ グループ内の他のインターフェイスには、新しい LSP をフラッディングしません。

（注） 特定のメッシュ ネットワーク トポロジーで、ネットワークのスケーラビリティを向上させるために、LSP を制限しなければならない場合があります。LSP フラッディングを制限すると、ネットワークの信頼性も下がります（障害発生時）。したがって、メッシュ グループはどうしても必要な場合に限り、慎重にネットワークを設計したうえで使用することを推奨します。

ルータ間のパラレル リンクに、ブロック モードでメッシュ グループを設定することもできます。このモードでは、各ルータがそれぞれリンクステート情報を最初に交換すると、それ以後はメッシュ グループのそのインターフェイスですべての LSP がブロックされます。

過負荷ビット

IS-IS は過負荷ビットを使用して、トラフィックの転送にはローカル ルータを使用しないが、引き続き、そのローカル ルータ宛てのトラフィックをルーティングすることを他のルータに指示します。

過負荷ビットを使用する状況は、次のとおりです。

- ルータがクリティカル条件下にある。
- ネットワークに対して通常手順でルータの追加および除去を行う。
- その他（管理上またはトラフィック エンジニアリング上）の理由。BGP コンバージェンスの待機中など。

ルート集約

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。

IS-IS 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小メトリックと同じメトリックを指定して、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

IS-IS を使用すると、スタティック ルート、他の IS-IS AS が学習したルート、またはほかのプロトコルからのルートを再配布できます。再配布を指定したルート マップを設定して、どのルートが IS-IS に渡されるかを制御する必要があります。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。詳細については、第 15 章「Route Policy Manager の設定」を参照してください。

IS-IS 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、IS-IS 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。IS-IS でデフォルト ルートを発生させ、ルート ポリシーでそのルートを制御できます。

IS-IS 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、ルータによって、宛先アドレスから同じ距離にあるすべてのルータ ネットワーク ポートにトラフィックが分散されます。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、ECMP（等コスト マルチパス）機能をサポートします。IS-IS 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 16 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、IS-IS を設定できます。

BFD

この機能は、IPv4 および IPv6 の Bidirectional Forwarding Detection（BFD）をサポートします。BFD は、転送パスの障害を高速で検出することを目的にした検出プロトコルです。BFD は 2 台の隣接デバイス間のサブセカンド障害を検出し、BFD の負荷の一部を、サポートされるモジュール上のデータ プレーンに分散できるため、プロトコル hello メッセージよりも CPU を使いません。詳細については、*『Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide』*を参照してください。

仮想化のサポート

Cisco NX-OS は、IS-IS の複数のプロセス インスタンスをサポートします。各 IS-IS インスタンスは、システム制限まで複数の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できます。サポートされる IS-IS インスタンスの数については、『*Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*』を参照してください。

ハイ アベイラビリティおよびグレースフル リスタート

Cisco NX-OS では、複数レベルのハイ アベイラビリティ アーキテクチャを提供します。IS-IS は、ステートフル リスタートをサポートしています。これは、ノンストップ ルーティング（NSR）とも呼ばれます。IS-IS で問題が発生した場合は、以前の実行時状態からの再起動を試みます。この場合、ネイバーはいずれのネイバー イベントも登録しません。最初の再起動が正常ではなく、別の問題が発生した場合、RFC 3847 のとおり、IS-IS はグレースフル リスタートを試みます。グレースフル リスタート、つまり、Nonstop Forwarding（NSF）では、処理の再起動中も IS-IS がデータ転送パス上に存在し続けます。再起動中の IS-IS インターフェイスが稼働を再開すると、ネイバーを再探索して隣接関係を確立し、更新情報の送信を再開します。この時点で、NSF ヘルパーは、グレースフル リスタートが完了したと認識します。

ステートフル リスタートは次のシナリオで使用されます。

- プロセスでの問題発生後の最初の回復試行
- **system switchover** コマンドによる手動でのスイッチオーバー

グレースフル リスタートは次のシナリオで使用されます。

- プロセスでの問題発生後の 2 回目の回復試行（4 分以内）
- **restart isis** コマンドによるプロセスの手動での再開
- アクティブ スーパーバイザの削除
- **reload module** *active-sup* コマンドによるアクティブ スーパーバイザのリロード

（注） グレースフル リスタートがデフォルトとなっており、ディセーブルにしないことを強く推奨します。

複数の IS-IS インスタンス

Cisco NX-OS は、同じノード上で動作する、IS-IS 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。すべてのインスタンスで同じシステム ルータ ID を使用します。サポートされる IS-IS インスタンスの数については、『*Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*』を参照してください。

IS-IS のライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	IS-IS には Enterprise Services ライセンスが必要です。Cisco NX-OS ライセンス方式について、およびライセンスの取得方法と適用方法の詳細については、*『Cisco NX-OS Licensing Guide』*を参照してください。

IS-IS の前提条件

IS-IS の前提条件は、次のとおりです。

- IS-IS をイネーブルにします（「IS-IS 機能のイネーブル化」セクション（8-9 ページ）を参照）。

IS-IS に関する注意事項および制限事項

IS-IS 設定時の注意事項および制約事項は、次のとおりです。

- デフォルトの参照帯域幅が Cisco NX-OS と Cisco IOS では異なるため、アドバタイズされたトンネル IS-IS メトリックは、これら 2 つのオペレーティング システムによって異なります。

デフォルト設定

表 8-1 に、IS-IS パラメータのデフォルト設定を示します。

表 8-1　デフォルトの IS-IS 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	115
エリア レベル	Level-1-2
DIS プライオリティ	64
グレースフル リスタート	イネーブル
hello 乗数	3
hello パディング	イネーブル
hello タイム	10 秒
IS-IS 機能	ディセーブル
LSP 間隔	33
LSP MTU	1492
最大 LSP ライフタイム	1200 秒
最大パス	8
メトリック	40
参照帯域幅	40 Gbps

IS-IS の設定

IS-IS を設定する手順は、次のとおりです。

ステップ 1　IS-IS 機能をイネーブルにします（「IS-IS 機能のイネーブル化」セクション（8-9 ページ）を参照）。

ステップ 2　IS-IS インスタンスを作成します（「IS-IS インスタンスの作成」セクション（8-10 ページ）を参照）。

ステップ 3　IS-IS インスタンスにインターフェイスを追加します（「インターフェイス上での IS-IS の設定」セクション（8-13 ページ）を参照）。

ステップ 4　認証、メッシュ グループ、ダイナミック ホスト交換などのオプション機能を設定します。

ここでは、次の内容について説明します。

- IS-IS コンフィギュレーション モード（8-8 ページ）
- IS-IS 機能のイネーブル化（8-9 ページ）
- IS-IS インスタンスの作成（8-10 ページ）
- IS-IS インスタンスの再起動（8-12 ページ）
- IS-IS のシャットダウン（8-12 ページ）
- インターフェイス上での IS-IS の設定（8-13 ページ）
- インターフェイスでの IS-IS のシャットダウン（8-14 ページ）
- エリアでの IS-IS 認証の設定（8-14 ページ）
- インターフェイス上での IS-IS 認証の設定（8-16 ページ）
- メッシュ グループの設定（8-17 ページ）
- DIS の設定（8-17 ページ）
- ダイナミック ホスト交換の設定（8-17 ページ）
- 過負荷ビットの設定（8-18 ページ）
- Attached ビットの設定（8-18 ページ）
- hello パディングの一時モードの設定（8-19 ページ）
- サマリー アドレスの設定（8-19 ページ）
- 再配布の設定（8-20 ページ）
- 再配布されるルート数の制限（8-22 ページ）
- 厳密な隣接モードのディセーブル化（8-23 ページ）
- グレースフル リスタートの設定（8-25 ページ）
- 仮想化の設定（8-26 ページ）
- IS-IS の調整（8-28 ページ）

（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。

IS-IS コンフィギュレーション モード

ここでは各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ことで、そのモードで使用可能なコマンドを表示できます。

この項では、次のトピックについて取り上げます。

- ルータ コンフィギュレーション モード（8-9 ページ）
- 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（8-9 ページ）

ルータ コンフィギュレーション モード

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#: configure terminal
switch(config)# router isis isp
switch(config-router)#
```

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

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

```
switch(config)# router isis isp
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)#
```

IS-IS 機能のイネーブル化

IS-IS を設定する前に、IS-IS 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 isis**
3. （任意）**show feature**
4. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 isis** **例**: `switch(config)# feature isis`	IS-IS 機能をイネーブルにします。
ステップ 3	**show feature** **例**: `switch(config)# show feature`	(任意) イネーブルおよびディセーブルにされた機能を表示します。
ステップ 4	**copy running-config startup-config** **例**: `switch(config)# copy running-config startup-config`	(任意) この設定の変更を保存します。

IS-IS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、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 isis **例:** `switch(config)# no feature isis`	IS-IS 機能をディセーブルにし、関連付けられたすべての設定を削除します。

IS-IS インスタンスの作成

IS-IS インスタンスを作成し、そのインスタンスのエリア レベルを設定できます。

はじめる前に

IS-IS をイネーブルにします（「IS-IS 機能のイネーブル化」セクション（8-9 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **router isis** *instance-tag*
3. **net** *network-entity-title*
4. （任意）**is-type** {**level-1** | **level-2** | **level-1-2**}
5. （任意）**show isis** [**vrf** *vrf-name*] **process**
6. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 isis** *instance-tag*
	
	例:
	`switch(config)# router isis Enterprise`
	`switch(config-router)#`
		instance tag を設定して、新しい IS-IS インスタンスを作成します。

ステップ 3	**net** *network-entity-title* **例:** `switch(config-router)# net 47.0004.004d.0001.0001.0c11.1111.00`	この IS-IS インスタンスに対応する NET を設定します。
ステップ 4	**is-type** {**level-1** | **level-2** | **level-1-2**} **例:** `switch(config-router)# is-type level-2`	(任意) この IS-IS インスタンスのエリア レベルを設定します。デフォルトは level-1-2 です。

	コマンド	目的
ステップ 5	**show isis** [**vrf** *vrf-name*] **process** **例**: `switch(config)# show isis process`	(任意)すべての IS-IS インスタンスについて、IS-IS 要約情報を表示します。
ステップ 6	**copy running-config startup-config** **例**: `switch(config)# copy running-config startup-config`	(任意)この設定の変更を保存します。

IS-IS インスタンスおよび関連する設定を削除するには、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 router isis *instance-tag* **例**: `switch(config)# no router isis Enterprise`	IS-IS 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） IS-IS インスタンスに関するすべての設定を完全に削除するには、インターフェイス モードで設定した IS-IS コマンドも削除する必要があります。

IS-IS には次のオプション パラメータを設定できます。

コマンド	目的
distance *value* **例**: `switch(config-router)# distance 30`	IS-IS 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 115 です。
log-adjacency-changes **例**: `switch(config-router)# log-adjacency-changes`	IS-IS ネイバーのステートが変化するたびに、システム メッセージを送信します。
lsp-mtu *size* **例**: `switch(config-router)# lsp-mtu 600`	この IS-IS インスタンスにおける LSP の MTU を設定します。指定できる範囲は 128 ～ 4352 バイトです。デフォルトは 1492 です。
maximum-paths *number* **例**: `switch(config-router)# maximum-paths 6`	IS-IS 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。範囲は 1 ～ 64 です。デフォルト値は 8 です。
reference-bandwidth *bandwidth-value* {**Mbps** | **Gbps**} **例**: `switch(config-router)# reference-bandwidth 100 Gbps`	IS-IS コスト メトリックの計算に使用する、デフォルトの基準帯域幅を設定します。指定できる範囲は 1 ～ 4000 Gbps です。デフォルトは 40 Gbps です。

レベル 2 エリアで IS-IS インスタンスを作成する例を示します。

```
switch# configure terminal
switch(config)# router isis Enterprise
switch(config-router)# net 47.0004.004d.0001.0001.0c11.1111.00
switch(config-router)# is-type level 2
switch(config-router)# copy running-config startup-config
```

ネイバーの統計情報を消去し、隣接関係を削除するには、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
clear isis [*instance-tag*] **adjacency** [***** | *system-id* | *interface*] **例**: `switch(config-if)# clear isis adjacency *`	ネイバーの統計情報を消去し、この IS-IS インスタンスの隣接関係を削除します。

IS-IS インスタンスの再起動

IS-IS インスタンスは再起動が可能です。この処理では、インスタンスのすべてのネイバーが消去されます。

IS-IS 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 isis instance-tag **例**: `switch(config)# restart isis Enterprise`	IS-IS 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。

IS-IS のシャットダウン

IS-IS インスタンスをシャットダウンできます。シャットダウンすると、その IS-IS インスタンスがディセーブルになり、設定が保持されます。

IS-IS インスタンス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
シャットダウン **例**: `switch(config-router)# shutdown`	IS-IS インスタンスをディセーブルにします。

インターフェイス上での IS-IS の設定

IS-IS インスタンスにインターフェイスを追加できます。

はじめる前に

IS-IS をイネーブルにします（「IS-IS 機能のイネーブル化」セクション（8-9 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. （任意）**medium** {**broadcast** | **p2p**}
4. {**ip** | **ipv6**} **router isis** *instance-tag*
5. （任意）**show isis** [**vrf** *vrf-name*] [*instance-tag*] **interface** [*interface-type slot/port*]
6. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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** *interface-type 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。
ステップ 3	**medium** {**broadcast** | **p2p**} **例**: switch(config-if)# medium p2p	(任意) インターフェイスのブロードキャスト モードまたはポイントツーポイント モードを設定します。IS-IS はこのモードを継承します。
ステップ 4	{**ip** | **ipv6**} **router isis** *instance-tag* **例**: switch(config-if)# ip router isis Enterprise	この IPv4 または IPv6 インターフェイスを IS-IS インスタンスに関連付けます。
ステップ 5	**show isis** [**vrf** *vrf-name*] [*instance-tag*] **interface** [*interface-type slot/port*] **例**: switch(config)# show isis Enterprise ethernet 1/2	(任意) VRF のインターフェイスの IS-IS 情報を表示します。
ステップ 6	**copy running-config startup-config** **例:** switch(config)# copy running-config startup-config	(任意) この設定の変更を保存します。

インターフェイス モードでは、IS-IS 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
isis circuit-type {**level-1** | **level-2** | **level-1-2**} **例**: `switch(config-if)# isis circuit-type level-2`	このインターフェイスが参加する隣接関係のタイプを設定します。このコマンドを使用するのは、レベル 1 とレベル 2 の両方のエリアにルータが関係する場合だけです。
isis metric *value* {**level-1** | **level-2**} **例**: `switch(config-if)# isis metric 30`	このインターフェイスの IS-IS メトリックを設定します。指定できる範囲は 1 ～ 16777214 です。デフォルトは 10 です。
isis passive {**level-1** | **level-2** | **level-1-2**} **例**: `switch(config-if)# isis passive level-2`	インターフェイスが隣接関係を形成しないようにしながら、なおかつ、インターフェイスに関連付けられたプレフィックスをアドバタイズするようにします。

次に、IS-IS インスタンスに Ethernet 1/2 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)# ip router isis Enterprise
switch(config-if)# copy running-config startup-config
```

インターフェイスでの IS-IS のシャットダウン

インターフェイス上で IS-IS 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 IS-IS トラフィックが停止しますが、IS-IS 設定は保持されます。

インターフェイス上で IS-IS をディセーブルにするには、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
`switch(config-if)#` **isis shutdown** **例**: `switch(config-router)# isis shutdown`	このインターフェイスで IS-IS をディセーブルにします。IS-IS インターフェイスの設定は保持されます。

エリアでの IS-IS 認証の設定

エリアで LSP を認証するように IS-IS を設定できます。

はじめる前に

IS-IS をイネーブルにします（「IS-IS 機能のイネーブル化」セクション（8-9 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **router isis** *instance-tag*
3. **authentication-type** {**cleartext** | **md5**} {**level-1** | **level-2**}
4. **authentication key-chain** *key* {**level-1** | **level-2**}
5. （任意）**authentication-check** {**level-1** | **level-2**}
6. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 isis** *instance-tag* **例**: switch(config)# router isis Enterprise switch(config-router)#	*instance tag* を設定して、新しい IS-IS インスタンスを作成します。
ステップ 3	**authentication-type** {**cleartext** | **md5**} {**level-1** | **level-2**} **例**: switch(config-router)# authentication-type cleartext level-2	クリアテキストまたは MD5 認証ダイジェストとして、レベル 1 またはレベル 2 エリアに使用する認証方式を設定します。
ステップ 4	**authentication key-chain** *key* {**level-1** | **level-2**} **例**: switch(config-router)# authentication key-chain ISISKey level-2	IS-IS エリアレベル認証に使用する認証キーを設定します。
ステップ 5	**authentication-check** {**level-1** | **level-2**} **例**: switch(config-router)# authentication-check level-2	(任意)受信パケットの認証パラメータ チェックをイネーブルにします。
ステップ 6	**copy running-config startup-config** **例**: switch(config-router)# copy running-config startup-config	(任意)この設定の変更を保存します。

IS-IS インスタンスにクリアテキスト認証を設定する例を示します。

```
switch# configure terminal
switch(config)# router isis Enterprise
switch(config-router)# authentication-type cleartext level-2
switch(config-router)# authentication key-chain ISISKey level-2
switch(config-router)# copy running-config startup-config
```

インターフェイス上での IS-IS 認証の設定

インターフェイス上で hello パケットを認証するように IS-IS を設定できます。

はじめる前に

IS-IS をイネーブルにします（「IS-IS 機能のイネーブル化」セクション（8-9 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **isis authentication-type** {**cleartext** | **md5**} {**level-1** | **level-2**}
4. **isis authentication key-chain** *key* {**level-1** | **level-2**}
5. （任意）**isis authentication-check** {**level-1** | **level-2**}
6. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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** *interface-type 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。
ステップ 3	**isis authentication-type** {**cleartext** | **md5**} {**level-1** | **level-2**} **例**: switch(config-if)# isis authentication-type cleartext level-2	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 IS-IS 認証タイプを設定します。
ステップ 4	**isis authentication key-chain** *key* {**level-1** | **level-2**} **例**: switch(config-if)# isis authentication-key ISISKey level-2	このインターフェイス上で IS-IS に使用する認証キーを設定します。
ステップ 5	**isis authentication-check** {**level-1** | **level-2**} **例**: switch(config-if)# isis authentication-check	（任意）受信パケットの認証パラメータ チェックをイネーブルにします。

	コマンド	目的
ステップ 6	**copy running-config startup-config** **例**: `switch(config-if)# copy running-config startup-config`	(任意) この設定の変更を保存します。

IS-IS インスタンスにクリアテキスト認証を設定する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# isis authentication-type cleartext level-2
switch(config-if)# isis authentication key-chain ISISKey
switch(config-if)# copy running-config startup-config
```

メッシュ グループの設定

メッシュ グループにインターフェイスを追加することによって、そのメッシュ グループ内のインターフェイスに対する LSP フラッディング量を制限できます。任意で、メッシュ グループ内のインターフェイスに対して、すべての LSP フラッディングをブロックすることもできます。

メッシュ グループにインターフェイスを追加するには、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
isis mesh-group {**blocked** | *mesh-id*} **例**: `switch(config-if)# isis mesh-group 1`	メッシュ グループにこのインターフェイスを追加します。範囲は 1 ～ 4294967295 です。

DIS の設定

インターフェイス プライオリティを設定することによって、ルータがマルチアクセス ネットワークの DIS (代表中継システム) になるように設定できます。

DIS 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
isis priority *number* {**level-1** | **level-2**} **例**: `switch(config-if)# isis priority 100 level-1`	DIS 選択のためのプライオリティを設定します。指定できる範囲は 0 ～ 127 です。デフォルトは 64 です。

ダイナミック ホスト交換の設定

ダイナミック ホスト交換を使用することによって、システム ID とルータのホスト名がマッピングされるように IS-IS を設定できます。

ダイナミック ホスト交換を設定するには、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
hostname dynamic **例**: `switch(config-router)# hostname dynamic`	ダイナミック ホスト交換をイネーブルにします。

過負荷ビットの設定

最短パス優先(SPF)を計算するときの中間ホップとしてこのルータを使用しないことを他のルータに伝えるように、ルータを設定できます。任意で、起動時に BGP がコンバージェンスするまで、一時的に過負荷ビットを設定することもできます。

過負荷ビットを設定する以外に、レベル 1 またはレベル 2 トラフィックに関して、LSP からの特定タイプの IP プレフィックス アドバタイズメントを抑制することが必要な場合もあります。

過負荷ビットを設定するには、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
set-overload-bit {**always** | **on-startup** {*seconds* | **wait-for bgp** *as-number*}} [**suppress** [**interlevel** | **external**]] **例**: `switch(config-router)# set-overload-bit on-startup 30`	IS-IS に過負荷ビットを設定します。*seconds* の範囲は 5 ～ 86400 です。

Attached ビットの設定

Attached ビットを設定すると、レベル 1 ルータがレベル 2 エリアへのデフォルト ルートとして使用するレベル 1/レベル 2 ルータを制御できます。Attached ビットの設定をディセーブルにすると、レベル 1 ルータはこのレベル 1/レベル 2 ルータを使用してレベル 2 エリアに接続しなくなります。

レベル 1/レベル 2 ルータの Attached ビットを設定するには、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
[**no**] **attached-bit** **例**: `switch(config-router)# no attached-bit`	Attached ビットを設定するようにレベル 1/レベル 2 ルータを設定します。この機能は、デフォルトでイネーブルにされています。

hello パディングの一時モードの設定

hello パディングの一時モードを設定すると、IS-IS が隣接関係を確立するときに hello パケットをパディングし、IS-IS が隣接関係を確立したあとでそのパディングを削除できます。

hello パディングのモードを設定するには、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
[no] **isis hello-padding** **例**: `switch(config-if)# no isis hello-padding`	完全な最大伝送単位（MTU）に hello パケットをパディングします。デフォルトではイネーブルになっています。hello パディングの一時モードを設定するには、このコマンドの **no** 形式を使用します。

サマリー アドレスの設定

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。1 つのサマリー アドレスに、特定のレベルのアドレス グループを複数含めることができます。Cisco NX-OS は固有性の強いすべてのルートのうち、最小メトリックをアドバタイズします。

はじめる前に

IS-IS をイネーブルにします（「IS-IS 機能のイネーブル化」セクション（8-9 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **router isis** *instance-tag*
3. **address-family** {**ipv4** | **ipv6**} **unicast**
4. **summary-address** *ip-prefix*/*mask-len* {**level-1** / **level-2** / **level-1-2**}
5. （任意）**show isis** [**vrf** *vrf-name*] {**ip** | **ipv6**} **summary-address** *ip-prefix* [**longer-prefixes**]
6. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 isis** *instance-tag*
	
	例:
	`switch(config)# router isis Enterprise`
	`switch(config-router)#`
		instance tag を設定して、新しい IS-IS インスタンスを作成します。

	コマンド	目的

ステップ 3	**address-family** {**ipv4** | **ipv6**} **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。

ステップ 4	**summary-address** *ip-prefix/mask-len* {**level-1** | **level-2** | **level-1-2**}
	
	例:
	`switch(config-router-af)#`
	`summary-address 192.0.2.0/24 level-2`
		IPv4 アドレスまたは IPv6 アドレスに対応する、IS-IS エリア用のサマリー アドレスを設定します。

ステップ 5	**show isis** [**vrf** *vrf-name*] {**ip** | **ipv6**} **summary-address** *ip-prefix* [**longer-prefixes**] **例**: `switch(config-if)# show isis ip summary-address`	（任意）IS-IS IPv4 または IPv6 サマリー アドレス情報を表示します。
ステップ 6	**copy running-config startup-config** **例**: `switch(config-if)# copy running-config startup-config`	（任意）この設定の変更を保存します。

次に、IS-IS の IPv4 ユニキャスト サマリー アドレスを設定する例を示します。

```
switch# configure terminal
switch(config)# router isis Enterprise
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# summary-address 192.0.2.0/24 level-2
switch(config-router-af)# copy running-config startup-config
```

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、IS-IS ネットワークを通じてその情報を再配布するように、IS-IS 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。

はじめる前に

IS-IS をイネーブルにします（「IS-IS 機能のイネーブル化」セクション（8-9 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **router isis** *instance-tag*
3. **address-family** {**ipv4** | **ipv6**} **unicast**
4. **redistribute** {**bgp** *as* | **direct** |{**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-name*
5. （任意）**default-information originate** [**always**] [**route-map** *map-name*]
6. （任意）**distribute** {**level-1** | **level-2**} **into** {**level-1** | **level-2**} {**route-map** *route-map* | **all**}
7. （任意）**show isis** [**vrf** *vrf-name*] {**ip** | **ipv6**} **route** *ip-prefix* [*detail* | **longer-prefixes** [**summary** | **detail**]]
8. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 isis** *instance-tag* **例:** switch(config)# router isis Enterprise switch(config-router)#	*instance tag* を設定して、新しい IS-IS インスタンスを作成します。
ステップ 3	**address-family** {**ipv4** | **ipv6**} **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。
ステップ 4	**redistribute** {**bgp** *as* | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static** | **direct**} **route-map** *map-name* **例**: switch(config-router-af)# redistribute eigrp 201 route-map ISISmap	他のプロトコルからのルートを IS-IS に再配布します。ルート マップの詳細については、「ルート マップの設定」セクション（15-13 ページ）を参照してください。
ステップ 5	**default-information originate** [**always**] [**route-map** *map-name*] **例**: switch(config-router-af)# default-information originate always	（任意）IS-IS へのデフォルト ルートを作成します。
ステップ 6	**distribute** {**level-1** | **level-2**} **into** {**level-1** | **level-2**} {**route-map** *route-map* | **all**} **例**: switch(config-router-af)# distribute level-1 into level-2 all	（任意）一方の IS-IS レベルから他方の IS-IS レベルへ、ルートを再配布します。
ステップ 7	**show isis** [**vrf** *vrf-name*] {**ip** | **ipv6**} **route** *ip-prefix* [*detail* | **longer-prefixes** [**summary** | **detail**]] **例:** switch(config-router-af)# show isis ip route	（任意）IS-IS ルートを示します。
ステップ 8	**copy running-config startup-config** **例:** switch(config-router-af)# copy running-config startup-config	（任意）この設定の変更を保存します。

次に、EIGRP を IS-IS に再配布する例を示します。

```
switch# configure terminal
switch(config)# router isis Enterprise
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# redistribute eigrp 201 route-map ISISmap
switch(config-router-af)# 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、IS-IS 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に最大制限を設定できます。IS-IS 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：IS-IS 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。IS-IS 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、IS-IS 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：IS-IS が最大値に達したときのみ、警告のログを記録します。IS-IS は引き続き再配布ルートを受け取ります。
- 取り消し：IS-IS 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。タイムアウト期間の経過後、現在の再配布ルートの数が最大制限より少ない場合、IS-IS はすべての再配布ルートを要求します。現在の再配布ルートの数が最大制限に達している場合、IS-IS はすべての再配布ルートを取り消します。IS-IS が以降の再配布ルートを受け取るには、この状態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。

はじめる前に

IS-IS をイネーブルにします（「IS-IS 機能のイネーブル化」セクション（8-9 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **router isis** *instance-tag*
3. **redistribute** {**bgp** *id* | **direct** | **eigrp** *id* | **isis** *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 isis**
6. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 isis** *instance-tag*
	
	例：
	`switch(config)# router isis Enterprise`
	`switch(config-router)#`
		instance tag を設定して、新しい IS-IS インスタンスを作成します。

ステップ 3	**redistribute** {**bgp** *id* | **direct** | **eigrp** *id* | **isis** *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 IS-IS に再配布します。

	コマンド	目的
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`	IS-IS が配布するプレフィックスの最大数を指定します。有効な範囲は 1 ～ 65535 です。次の項目を任意で指定できます。 • *threshold*：警告メッセージをトリガーする最大プレフィックス数のパーセンテージ。 • **warning-only**：プレフィックスの最大数を超えたとき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。オプション選択で、再配布されたルートの取得を試みることができます。*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 isis redistribution** コマンドは、すべてのルートが取り消された場合に使用します。
ステップ 5	**show running-config isis** **例:** `switch(config-router)# show running-config isis`	(任意) IS-IS の設定を表示します。
ステップ 6	**copy running-config startup-config** **例:** `switch(config-router)# copy running-config startup-config`	(任意) この設定の変更を保存します。

次に、IS-IS 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 isis Enterprise
switch(config-router)# redistribute bgp route-map FilterExternalBGP
switch(config-router)# redistribute maximum-prefix 1000 75
```

厳密な隣接モードのディセーブル化

IPv4 と IPv6 の両方のアドレス ファミリがイネーブルの場合、厳格な隣接モードはデフォルトでイネーブルです。このモードでは、デバイスが両方のアドレス ファミリにイネーブルでない任意のルータとの隣接関係を形成しません。厳格な隣接モードは、**no adjacency-check** コマンドを使用してディセーブルにできます。

はじめる前に

IS-IS をイネーブルにします(「IS-IS 機能のイネーブル化」セクション (8-9 ページ)を参照)。

手順の概要

1. **configure terminal**
2. **router isis** *instance-tag*
3. **address-family ipv4 unicast**
4. **no adjacency-check**

5. **exit**
6. **address-family ipv6 unicast**
7. **no adjacency-check**
8. （任意）**show running-config isis**
9. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 isis** *instance-tag*
	
	例:
	`switch(config)# router isis Enterprise`
	`switch(config-router)#`
		instance tag を設定して、新しい IS-IS 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)#`
		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	**no adjacency-check** **例**: `switch(config-router-af)# no adjacency-check`	IPv4 アドレス ファミリに関する厳格な隣接モードをディセーブルにします。

ステップ 5	**exit**
	
	例:
	`switch(config-router-af)# exit`
	`switch(config-router)#`
		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 6	**address-family ipv6 unicast**
	
	例:
	`switch(config-router)# address-family ipv6 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。

ステップ 7	**no adjacency-check** **例**: `switch(config-router-af)# no adjacency-check`	IPv6 アドレス ファミリに関する厳格な隣接モードをディセーブルにします。

	コマンド	目的
ステップ 8	**show running-config isis** **例:** `switch(config-router-af)# show running-config isis`	(任意)IS-IS の設定を表示します。
ステップ 9	**copy running-config startup-config** **例:** `switch(config-router-af)# copy running-config startup-config`	(任意)この設定の変更を保存します。

グレースフル リスタートの設定

IS-IS にグレースフル リスタートを設定できます。

はじめる前に

IS-IS をイネーブルにします(「IS-IS 機能のイネーブル化」セクション(8-9 ページ)を参照)。
VRF を作成します。

手順の概要

1. **configure terminal**
2. **router isis** *instance-tag*
3. **graceful-restart**
4. **graceful-restart t3 manual** *time*
5. (任意)**show running-config isis**
6. (任意)**copy running-config startup-config**

手順の詳細

	コマンド	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 isis** *instance-tag*
	
	例:
	`switch(config)# router isis Enterprise`
	`switch(config-router)#`
		名前を設定して、新しい IS-IS プロセスを作成します。

ステップ 3	**graceful-restart** **例:** `switch(config-router)# graceful-restart`	グレースフル リスタートおよびグレースフル リスタート ヘルパー機能をイネーブルにします。デフォルトでは、イネーブルです。

	コマンド	目的
ステップ 4	**graceful-restart t3 manual** *time* **例:** `switch(config-router)# graceful-restart t3 manual 300`	グレースフル リスタート T3 タイマーを設定します。有効な範囲は 30 ～ 65535 秒です。デフォルトは 60 です。
ステップ 5	**show running-config isis** **例:** `switch(config-router)# show running-config isis`	(任意)IS-IS の設定を表示します。
ステップ 6	**copy running-config startup-config** **例:** `switch(config-router)# copy running-config startup-config`	(任意)この設定の変更を保存します。

次に、グレースフル リスタートをイネーブルにする例を示します。

```
switch# configure terminal
switch(config)# router isis Enterprise
switch(config-router)# graceful-restart
switch(config-router)# copy running-config startup-config
```

仮想化の設定

複数の IS-IS インスタンスと複数の VRF を設定できます。また、各 VRF で同じまたは複数の IS-IS インスタンスを使用することもできます。VRF に IS-IS インターフェイスを割り当てます。

設定した VRF に NET を設定する必要があります。

(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。

はじめる前に

IS-IS をイネーブルにします(「IS-IS 機能のイネーブル化」セクション(8-9 ページ)を参照)。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf_name*
3. **exit**
4. **router isis** *instance-tag*
5. (任意) **vrf** *vrf_name*
6. **net** *network-entity-title*
7. **exit**
8. **interface** *type slot/port*
9. **vrf member** *vrf-name*

10. {**ip** | **ipv6**} **address** *ip-prefix*/*length*
11. {**ip** | **ipv6**} **router isis** *instance-tag*
12. （任意）**show isis** [**vrf** *vrf-name*] [*instance-tag*] **interface** [*interface-type slot*/*port*]
13. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	**vrf context** *vrf-name*
	
	例:
	`switch(config)# vrf context RemoteOfficeVRF`
	`switch(config-vrf)#`
		新しい VRF を作成し、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	**exit**
	
	例:
	`switch(config-vrf)# exit`
	`switch(config)#`
		VRF コンフィギュレーション モードを終了します。

ステップ 4	**router isis** *instance-tag*
	
	例:
	`switch(config)# router isis Enterprise`
	`switch(config-router)#`
		instance tag を設定して、新しい IS-IS インスタンスを作成します。

ステップ 5	**vrf** *vrf-name*
	
	例:
	`switch(config-router)# vrf RemoteOfficeVRF`
	`switch(config-router-vrf)#`
		（任意）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 6	**net** *network-entity-title* **例**: `switch(config-router-vrf)# net 47.0004.004d.0001.0001.0c11.1111.00`	この IS-IS インスタンスに対応する NET を設定します。

ステップ 7	**exit**
	
	例:
	`switch(config-router-vrf)# exit`
	`switch(config-router)#`
		ルータ VRF 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	**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。

ステップ 9	**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。

	コマンド	目的
ステップ 10	{**ip** | **ipv6**} **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 11	{**ip** | **ipv6**} **router isis** *instance-tag* **例**: `switch(config-if)# ip router isis Enterprise`	この IPv4 または IPv6 インターフェイスを IS-IS インスタンスに関連付けます。
ステップ 12	**show isis** [**vrf** *vrf-name*] [*instance-tag*] **interface** [*interface-type slot/port*] **例**: `switch(config-if)# show isis Enterprise ethernet 1/2`	(任意) VRF のインターフェイスに関する IS-IS 情報を表示します。
ステップ 13	**copy running-config startup-config** **例:** `switch(config-if)# copy running-config startup-config`	(任意) この設定の変更を保存します。

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)# exit
switch(config)# router isis Enterprise
switch(config-router)# vrf NewVRF
switch(config-router-vrf)# net 47.0004.004d.0001.0001.0c11.1111.00
switch(config-router-vrf)# interface ethernet 1/2
switch(config-if)# vrf member NewVRF
switch(config-if)# ip address 192.0.2.1/16
switch(config-if)# ip router isis Enterprise
switch(config-if)# copy running-config startup-config
```

IS-IS の調整

ネットワーク要件に合わせて IS-IS を調整できます。

IS-IS 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
lsp-gen-interval [**level-1** | **level-2**] *lsp-max-wait* [*lsp-initial-wait lsp-second-wait*] **例**: `switch(config-router)# lsp-gen-interval level-1 500 500 500`	LSP 発生に関する IS-IS スロットルを設定します。オプション パラメータは次のとおりです。 • lsp-max-wait:トリガーから LSP 発生までの最大待ち時間。指定できる範囲は 500 ～ 65535 ミリ秒です。 • lsp-initial-wait:トリガーから LSP 発生までの初期待ち時間。指定できる範囲は 50 ～ 65535 ミリ秒です。 • lsp-second-wait:バックオフ時の LSP スロットルに使用する第 2 待ち時間。指定できる範囲は 50 ～ 65535 ミリ秒です。
max-lsp-lifetime *lifetime* **例**: `switch(config-router)# max-lsp-lifetime 500`	LSP の最大ライフタイムを秒数で設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 1200 です。
metric-style transition **例**: `switch(config-router)# metric-style transition`	IS-IS がナロー メトリック スタイルのタイプ、長さ、値(TLV)オブジェクトとワイド メトリック スタイルの TLV オブジェクトの両方を生成して受け取ることができるようにします。デフォルトではディセーブルになっています。
spf-interval [**level-1** | **level-2**] *spf-max-wait* [*spf-initial-wait spf-second-wait*] **例**: `switch(config-router)# spf-interval level-2 500 500 500`	LSA 到着までのインターバルを設定します。オプション パラメータは次のとおりです。 • lsp-max-wait:トリガーから SPF 計算までの最大待ち時間。指定できる範囲は 500 ～ 65535 ミリ秒です。 • lsp-initial-wait:トリガーから SPF 計算までの初期待ち時間。指定できる範囲は 50 ～ 65535 ミリ秒です。 • lsp-second-wait:バックオフ時の SPF 計算に使用する第 2 待ち時間。指定できる範囲は 50 ～ 65535 ミリ秒です。

ルータ アドレス コンフィギュレーション モードで次のオプション コマンドを使用できます。

コマンド	目的
adjacency-check **例**: `switch(config-router-af)# adjacency-check`	隣接関係チェックを実行し、IS-IS インスタンスが同じアドレス ファミリをサポートするリモート IS-IS エンティティに限って隣接関係を形成していることを確認します。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。

IS-IS 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
isis csnp-interval *seconds* [**level-1** | **level-2**] **例**: `switch(config-if)# isis csnp-interval 20`	IS-IS に Complete Sequence Number PDU（CNSP）インターバルを秒数で設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 10 です。
isis hello-interval *seconds* [**level-1** | **level-2**] **例**: `switch(config-if)# isis hello-interval 20`	IS-IS に hello 間隔を秒数で設定します。有効な範囲は 1 ～ 65535 です。デフォルトは 10 です。
isis hello-multiplier *num* [**level-1** | **level-2**] **例**: `switch(config-if)# isis hello-multiplier 20`	ルータが隣接関係を破棄するまでに、ネイバーが見逃さなければならない IS-IS hello パケットの数を指定します。指定できる範囲は 3 ～ 1000 です。デフォルトは 3 です。
isis lsp-interval *milliseconds* **例**: `switch(config-if)# isis lsp-interval 20`	フラッディング時にこのインターフェイスで LSP が送信される間隔をミリ秒数で設定します。指定できる範囲は 10 ～ 65535 です。デフォルトは 33 です。

IS-IS 設定の確認

IS-IS 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 isis [*instance-tag*] **adjacency** [*interface*] [**detail** | **summary**] [**vrf** *vrf-name*]	IS-IS の隣接関係を表示します。これらの統計情報を消去するには、**clear isis adjacency** コマンドを使用します。
show isis [*instance-tag*] **database** [**level-1** | **level-2**] [**detail** | **summary**] [*LSP ID*] [{**ip** | **ipv6**} **prefix** *ip-prefix*] | [**router-id** *router-id*] | [**adjacency** *node-id*] | [**zero-sequence**]} [**vrf** *vrf-name*]	IS-IS LSP データベースを表示します。
show isis [*instance-tag*] *hostname* [**vrf** *vrf-name*]	ダイナミック ホスト交換情報を表示します。
show isis [*instance-tag*] **interface** [**brief** | *interface*] [**level-1** | **level-2**] [**vrf** *vrf-name*]	IS-IS インターフェイス情報を表示します。
show isis [*instance-tag*] **mesh-group** [*mesh-id*] [**vrf** *vrf-name*]	メッシュ グループ情報を表示します。
show isis [*instance-tag*] **protocol** [**vrf** *vrf-name*]	IS-IS プロトコルに関する情報を表示します。
show isis [*instance-tag*] {**ip** | **ipv6**} **redistribute route** [*ip-address* / **summary**] [[*ip-prefix*] [**longer-prefixes** [**summary**]] [**vrf** *vrf-name*]	IS-IS のルート再配布情報を表示します。

コマンド	目的
show isis [*instance-tag*] {**ip** | **ipv6**} **route** [*ip-address* / **summary**] [*ip-prefix* [**longer-prefixes** [**summary**]] [**detail**] [**vrf** *vrf-name*]	IS-IS ルート テーブルを表示します。
show isis [*instance-tag*] **rrm** [*interface*] [**vrf** *vrf-name*]	IS-IS インターフェイスの再送信情報を表示します。
show isis [*instance-tag*] **srm** [*interface*] [**vrf** *vrf-name*]	IS-IS インターフェイスのフラッディング情報を表示します。
show isis [*instance-tag*] **ssn** [*interface*] [**vrf** *vrf-name*]	IS-IS インターフェイスの PSNP 情報を表示します。
show isis [*instance-tag*] {**ip** | **ipv6**} **summary-address** [*ip-address*] | [*ip-prefix*] [**vrf** *vrf-name*]	IS-IS のサマリー アドレス情報を表示します。
show running-configuration isis	現在の実行中の IS-IS 設定を表示します。
show tech-support isis [**detail**]	IS-IS のテクニカル サポートの詳細情報を表示します。

IS-IS のモニタリング

IS-IS 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 isis [*instance-tag*] **adjacency** [*interface*] [**system-ID**] [**detail**] [**summary**] [**vrf** *vrf-name*]	IS-IS 隣接関係の統計情報を表示します。
show isis [*instance-tag*] **database** [**level-1** | **level-2**] [**detail** | **summary**] [*lsip*] {[**adjacency** *id*] {**ip** | **ipv6**} **prefix** *prefix*] [**router-id** *id*] [zero-sequence]} [**vrf** *vrf-name*]	IS-IS データベースの統計情報を表示します。
show isis [*instance-tag*] **statistics** [*interface*] [**vrf** *vrf-name*]	IS-IS インターフェイスの統計情報を表示します。
show isis {**ip** | **ipv6**} **route-map statistics redistribute** {**bgp** *id* | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} [**vrf** *vrf-name*]	IS-IS 再配布の統計情報を表示します。
show isis route-map statistics distribute {**level-1** | **level-2**} **into** {**level-1** | **level-2**}} [**vrf** *vrf-name*]	レベル間で配布されたルートに関する、IS-IS 配布統計情報を表示します。
show isis [*instance-tag*] **spf-log** [**detail**] [**vrf** *vrf-name*]	IS-IS SPF 計算の統計情報を表示します。
show isis [*instance-tag*] **traffic** [*interface*] [**vrf** *vrf-name*]	IS-IS トラフィックの統計情報を表示します。

IS-IS 設定の統計情報を消去するには、次のいずれかの作業を行います。

コマンド	目的
clear isis [*instance-tag*] **adjacency** [***** | [*interface*] [**system-id** *id*]] [**vrf** *vrf-name*]	IS-IS 隣接関係の統計情報を消去します。
clear isis {**ip** | **ipv6**} **route-map statistics redistribute** {**bgp** *id* | **direct** | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} [**vrf** *vrf-name*]	IS-IS 再配布の統計情報を消去します。
clear isis route-map statistics distribute {**level-1** | **level-2**} **into** {**level-1** | **level-2**} [**vrf** *vrf-name*]	レベル間で配布されたルートに関する、IS-IS 配布統計情報を消去します。
clear isis [*instance-tag*] **statistics** [***** | *interface*] [**vrf** *vrf-name*]	IS-IS インターフェイスの統計情報を消去します。
clear isis [*instance-tag*] **traffic** [***** | *interface*] [**vrf** *vrf-name*]	IS-IS トラフィックの統計情報を消去します。

IS-IS の設定例

IS-IS を設定する例を示します。

```
router isis Enterprise
 is-type level-1
 net 49.0001.0000.0000.0003.00
 graceful-restart
 address-family ipv4 unicast
  default-information originate

interface ethernet 2/1
 ip address 192.0.2.1/24
 isis circuit-type level-1
 ip router isis Enterprise
```

関連項目

ルート マップの詳細については、第 15 章「Route Policy Manager の設定」を参照してください。

ベーシック BGP の設定

この章では、デバイス上Cisco NX-OS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。

基本的な BGP について

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 マルチキャスト 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 対応デバイスとの間で 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 デバイスまたは 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」セクション（10-3 ページ）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」セクション（10-7 ページ）を参照してください。

この項では、次のトピックについて取り上げます。

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」セクション（1-5 ページ）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 は、プレーン テキスト表記法または AS ドット付き表記法の 2 バイトの自律システム（AS）番号、もしくはプレーン テキスト表記法の 4 バイトの AS 番号をサポートします。

4 バイトの AS 番号を使用して BGP が設定されている場合は、**route-target auto** VXLAN コマンドを使用できません。これは、AS 番号とともに（すでに 3 バイト値である）VNI がルート ターゲットの生成に使用されるためです。詳細については、『*Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide*』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。BGP はデフォルトで、表 9-1のアドミニストレーティブ ディスタンスを使用します。

表 9-1　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
External	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	220	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」セクション（1-7 ページ）を参照してください。

BGP ピア

BGP スピーカが別の BGP スピーカを自動的に検出することはありません。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 または IPv4 アドレス：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 または IPv6 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、第 10 章「Configuring Advanced BGP」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、第 10 章「Configuring Advanced BGP」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。ルータ ID はデフォルトで、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、ルータ上の物理インターフェイスに設定されている最大の IPv4 アドレスが BGP ルータ ID を表すものとして、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、プレフィックスごとの複数パスの送受信と、このパスのアドバタイジングをサポートします。追加 BGP パスの設定については、第 10 章「Configuring Advanced BGP」を参照してください。

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。

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」セクション（9-5 ページ）を参照）。

ステップ 2　すべてのパスを探索し、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」セクション（9-6 ページ）を参照）。

ステップ 3　新しいベスト パスを使用するに足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」セクション（9-7 ページ）を参照）。

（注）　重要なのは、ステップ 2 で決定される比較順序です。3 つのパス A、B、および C がある場合を考えます。A と B を比較した場合、Cisco NX-OS は A を選択します。B と C を比較した場合、Cisco NX-OS は B を選択します。しかし、A と C を比較した場合、Cisco NX-OS は A を選択しません。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。

（注）　VXLAN 展開では、リモート パスよりローカル パスを優先する通常の選択プロセスとは異なる BGP パス選択プロセスを使用します。EVPN アドレス ファミリの場合、BGP は MAC モビリティ属性内のシーケンス番号（存在する場合）を比較し、より大きいシーケンス番号を持つパスを選択します。比較対象の両方のパスの属性とシーケンス番号が同じである場合、BGP はローカル パスではなくリモート ピアから学習したパスを選択します。詳細については、『*Cisco Nexus 9000 Series NX-OS VXLAN Configuration Guide*』を参照してください。

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」セクション(10-4 ページ)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」セクション(10-11 ページ)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。

   a. パスに AS パスまたは AS_SET から始まる AS パスがない場合、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。
   b. 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。
   c. AS-path 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。
   d. 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 がパスの指定された MED 属性を受信しなかった場合、欠落 MED が使用可能な最大値になるように、ユーザがベストパス アルゴリズムを設定していない限り、Cisco NX-OS は 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」セクション(10-11 ページ)を参照してください。

   e. 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」セクション(10-11 ページ)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」セクション（10-11 ページ）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 アドレスが小さいほう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」セクション（10-7 ページ）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」セクション（9-5 ページ）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決定します。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 が新しい最適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」セクション（10-11 ページ）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 は IPv6 ユニキャスト RIB と通信し、IPv6 ルートについて、これらの動作を実行します。

BGP プレフィックス独立コンバージェンス

BGP プレフィックス独立コンバージェンス（PIC）のコア機能を導入しました。この機能を使用すると、ネットワークのコア部分で障害が発生した場合に、同じリモート ネクスト ホップを共有する BGP プレフィックス宛てのトラフィックの、高速コンバージェンスが可能になります。純粋な IP トラフィックは、この機能の利点を活用できます。デフォルトでイネーブルであり、ディセーブルにすることはできません。

BGP の仮想化

BGP 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。

ベーシック BGP のライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	BGP には Enterprise Services ライセンスが必要です。Cisco NX-OS ライセンス方式の詳細と、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。

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 をイネーブルにします（「BGP の有効化」セクション（9-12 ページ）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 に関する注意事項および制限事項

BGP 設定時の注意事項および制約事項は、次のとおりです。

- 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、BGP/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 ポリシーを指定します。
- VRF 内で BGP ルータ ID を定義します。
- IPv6 ネイバーの場合は、VRF 単位でルータ ID を設定することをお勧めします。VRF に IPv4 インターフェイスが存在しない場合、IPv6 BGP ネイバーはオンライン状態になりません。これは、そのルータ ID は IPv4 アドレスである必要があるためです。数値が最も小さいループバック IPv4 アドレスがルータ ID として選択されます。ループバック アドレスが存在しない場合は、VRF インターフェイスの最も小さい IP アドレスが選択されます。そのアドレスが存在しない場合、BGP ネイバー関係は確立されません。

- 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。
- すべての iBGP および eBGP セッションの BGP の最小ルート アドバタイズメント インターバル（MRAI）値はゼロであり、設定できません。
- VRF を設定する場合は、Advanced Services ライセンスをインストールし、所定の VRF を開始してください（第 13 章「レイヤ 3 仮想化の設定」を参照）。
- **show ip bgp** コマンドは BGP 設定の確認に使用できますが、代わりに **show bgp** コマンドを使用することを推奨します。

デフォルト設定値

表 9-2 に、BGP パラメータのデフォルト設定を示します。

表 9-2　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
BGP PIC コア	イネーブル
Auto-summary	常に無効
同期	常に無効

CLI コンフィギュレーション モード

ここでは 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ことで、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、第 10 章「Configuring Advanced BGP」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 VRF をサポートしています。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。詳細については、「仮想化の設定」セクション（10-52 ページ）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 コンフィギュレーション 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 コンフィギュレーション 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 コンフィギュレーション 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 コンフィギュレーション モードからアドレス 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)# address-family ipv6 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 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)# address-family ipv6 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。

Cisco NX-OS Release 7.0(3)I2(1) には RFC 5549 が導入されているため、IPv6 アドレスを持つネイバーに IPv4 アドレス ファミリを設定できます。

次に、IPv4 アドレスを持つネイバーに対して IPv4 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、IPv6 アドレスを持つネイバーに対して IPv4 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 2001:db8::/64 eui64
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)#
```

次に、IPv4 アドレスを持つネイバーに対して VRF IPv4 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)#
```

次に、IPv6 アドレスを持つネイバーに対して VRF IPv4 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 2001:db8::/64 eui64
switch(config-router-vrf-neighbor)# address-family ipv4 unicast
switch(config-router-vrf-neighbor-af)#
```

ベーシック BGP の設定

ベーシック BGP を設定するには、BGP を有効にして、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。

この項では、次のトピックについて取り上げます。

（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。

BGP の有効化

BGP を設定する前に、BGP 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	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 bgp** **例:** `switch(config)# feature bgp`	BGP をイネーブルにします。
ステップ 3	**show feature** **例:** `switch(config)# show feature`	（任意）イネーブルおよびディセーブルにされた機能を表示します。
ステップ 4	**copy running-config startup-config** **例:** `switch(config)# copy running-config startup-config`	（任意）この設定の変更を保存します。

BGP をディセーブルにして、関連するすべての設定を削除する場合は、**no feature bgp** コマンドを使用します。

コマンド	目的
no feature bgp **例:** `switch(config)# no feature bgp`	BGP 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。詳細については、「BGP ルータ ID」セクション（9-4 ページ）を参照してください。

はじめる前に

BGP をイネーブルにします（「BGP の有効化」セクション（9-12 ページ）を参照）。
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** | **ipv6** } { **unicast** | **multicast** }
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	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** *autonomous-system-number* **例**: switch(config)# router bgp 64496 switch(config-router)#	BGP をイネーブルにして、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	**router-id** *ip-address* **例**: switch(config-router)# router-id 192.0.2.255	（任意）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。
ステップ 4	**address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **例**: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	（任意）IPv4 または IPv6 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	**network** {*ip-address*/*length* | *ip-address* **mask** *mask*} [**route-map** *map-name*] **例**: switch(config-router-af)# network 10.10.10.0/24 **例**: switch(config-router-af)# network 10.10.10.0 mask 255.255.255.0	（任意）この自律システムにローカルとしてネットワークを指定し、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。内部プロトコルは **network** コマンドを使用して、アップデートの送信先を判断します。

	コマンド	目的
ステップ 6	**show bgp all** **例**: switch(config-router-af)# show bgp all	(任意)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	**copy running-config startup-config** **例**: switch(config-router-af)# copy running-config startup-config	(任意)この設定の変更を保存します。

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

設定を維持しながら、BGP をシャット ダウンして BGP を正常にディセーブルにできます。

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

BGP ピアの設定

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。

はじめる前に

BGP をイネーブルにします（「BGP の有効化」セクション（9-12 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor** { *ip-address* | *ipv6-address* } **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** | **ipv6**} { **unicast** | **multicast** }
8. （任意）weight value
9. （任意）**show bgp** { **ipv4** | **ipv6** } { **unicast** | **multicast** } **neighbors**
10. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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** *autonomous-system-number*
	
	例:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		BGP をイネーブルにして、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* | *ipv6-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 アドレスまたは IPv6 アドレスおよび AS 番号を設定します。*ip-address* の形式は x.x.x.x です。*ipv6-address* の形式は A:B::C:D です。

ステップ 4	**description** *text*
	
	例:
	`switch(config-router-neighbor)# description Peer Router B`
	`switch(config-router-neighbor)#`
		(任意) 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	**timers** *keepalive-time hold-time* **例**: `switch(config-router-neighbor)# timers 30 90`	(任意) 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	**シャットダウン** **例**: `switch(config-router-neighbor)# shutdown`	(任意)。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** | **ipv6**} {**unicast** | **multicast**}
	
	例:
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		IPv4 または IPv6 アドレス ファミリに対してネイバー 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。

ステップ 8	**weight** *value* **例**: `switch(config-router-neighbor-af)# weight 100`	(任意) このネイバーからのルートのデフォルトの重みを設定します。範囲は 0 ～ 65535 です。 このネイバーから学習したすべてのルートに、まず重みが割り当てられます。特定のネットワークへのルートが複数ある場合、最大の重みを持つルートが優先ルートとして選ばれます。**set weight route-map** コマンドで割り当てられた重みは、このコマンドで割り当てられた重みを上書きします。 BGP ピア ポリシー テンプレートを指定した場合、テンプレートのメンバーすべてが、このコマンドで設定された特性を継承します。

	コマンド	目的
ステップ 9	**show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors** **例**: switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors	(任意)BGP ピアの情報を表示します。
ステップ 10	**copy running-config startup-config** **例**: switch(config-router-neighbor-af) copy running-config startup-config	(任意)この設定の変更を保存します。

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)# weight 100
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 番号を使用して設定された BGP セッションでは、**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 番号のリストを変更できますが、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。

はじめる前に

BGP をイネーブルにします(「BGP の有効化」セクション(9-12 ページ)を参照)。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor** *prefix* **remote-as route-map** *map-name*
4. (任意)**show bgp** { **ipv4** | **ipv6** } { **unicast** | **multicast** } **neighbors**
5. (任意)**copy running-config startup-config**

手順の詳細

	コマンド	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** *autonomous-system-number*
	
	例:
	`switch(config)# router bgp 64496`
	`switch(config-router)#`
		BGP をイネーブルにして、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 プレフィックスまたは IPv6 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 マップを設定します。IPv4 の場合の *prefix* の形式は「x.x.x.x/長さ」です。長さの範囲は 1 ～ 32 です。IPv6 の場合の *prefix* の形式は「A:B::C:D/長さ」です。長さの範囲は 1 ～ 128 です。
		
		map-name には最大 63 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。

ステップ 4	**show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors** **例**: `switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors`	(任意) BGP ピアの情報を表示します。
ステップ 5	**copy running-config startup-config** **例**: `switch(config-router-neighbor-af)# copy running-config startup-config`	(任意) この設定の変更を保存します。

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
```

ルート マップについては、第 15 章「Route Policy Manager の設定」を参照してください。

BGP 情報の消去

BGP 情報をクリアするには、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。***** は、すべてのアドレス ファミリのすべてのネイバーをクリアします。引数は次のとおりです。 • *neighbor*：ネイバーの IPv4 または IPv6 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 または IPv6 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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} **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** | **ipv6**} {**unicast** | **multicast**} {*neighbor* | * | *as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	選択したアドレス ファミリから 1 つ以上のネイバーをクリアします。* は、アドレス ファミリのすべてのネイバーをクリアします。引数は次のとおりです。 • *neighbor*：ネイバーの IPv4 または IPv6 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 または IPv6 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 ip bgp {**ip** {**unicast** | **multicast**}} {*neighbor* | * | *as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	1 つ以上のネイバーをクリアします。* は、アドレス ファミリのすべてのネイバーをクリアします。引数は次のとおりです。 • *neighbor*：ネイバーの IPv4 または IPv6 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 または IPv6 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-neighbor*：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-neighbor*：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** | **multicast**}} {*neighbor* | ***** | *as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	1 つ以上のネイバーをクリアします。***** は、アドレス ファミリのすべてのネイバーをクリアします。引数は次のとおりです。 • *neighbor*：ネイバーの IPv4 または IPv6 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 または IPv6 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-neighbor*：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-neighbor*：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 の設定確認

BGP 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 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** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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*] {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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。

コマンド	目的
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] {**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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [i*p-address* | *ipv6-prefix*] **neighbors** [*ip-address* | *ipv6-prefix*] [**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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] {**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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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** | **ipv6**} **bgp** *options*	BGP のステータスと構成情報を表示します。
show {**ip** | **ipv6**} **mbgp** *options*	BGP のステータスと構成情報を表示します。
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 統計情報のモニタリング

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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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消去するには、**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消去するには、**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。

```
switch(config)# feature bgp
switch(config)# router bgp 64496
switch(config-router)# neighbor 2001:DB8:0:1::55 remote-as 64496
switch(config-router-neighbor)# address-family ipv6 unicast
switch(config-router-af)# next-hop-self
```

関連項目

BGP の関連項目は、次のとおりです。

- 第 10 章「Configuring Advanced BGP」
- 第 15 章「Route Policy Manager の設定」

次の作業

次の機能の詳細について、第 10 章「Configuring Advanced BGP」、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

その他の関連資料

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。

- MIB（9-25 ページ）

MIB

MIB	MIB のリンク
BGP に関連する MIB	サポートされている MIB を検索およびダウンロードするには、次の URL にアクセスしてください。 ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

Configuring Advanced BGP

この章では、Cisco NX-OS デバイスでボーダー 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 について

BGP は、組織または自律システム（AS）間のループフリー ルーティングを実現する、ドメイン間ルーティング 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 マルチキャスト 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 対応デバイス（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 external BGP（eBGP; 外部 BGP）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。

この項では、次のトピックについて取り上げます。

ピア テンプレート

BGP ピア テンプレートを使用すると、共通のコンフィギュレーション ブロックを作成し、類似している BGP ピア間で再利用できます。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 タイマーといった 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）　BGP ピア間で MD5 パスワードを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、次のサポートをします。

- 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。ルート マップの詳細については、第 15 章「Route Policy Manager の設定」を参照してください。

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。

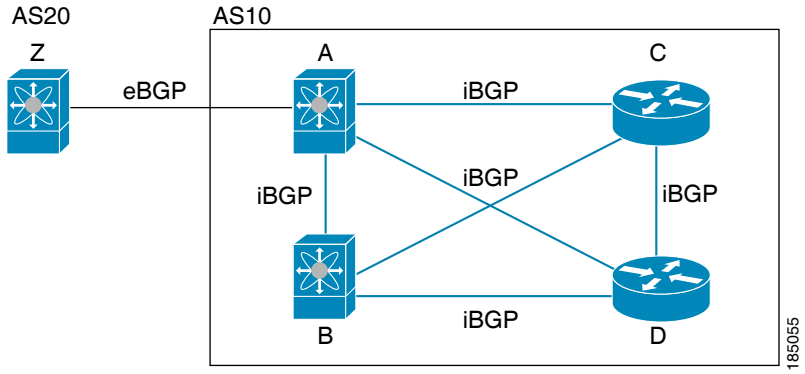
通常、インターフェイスがダウンしたときにコンバージェンスが高速化されるように、eBGP ピアリングは直接接続されたインターフェイスで行われる必要があります。

iBGP

iBGP を使用すると、同じ AS 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。

図 10-1 に、規模の大きい BGP ネットワークの中の iBGP ネットワークを示します。

図 10-1　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。

ネイバー コンフィギュレーション モードで update-source が設定された単一ホップ iBGP ピアでは、ピアは高速外部フォールオーバーをサポートします。

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」セクション（10-31 ページ）を参照してください。

（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。

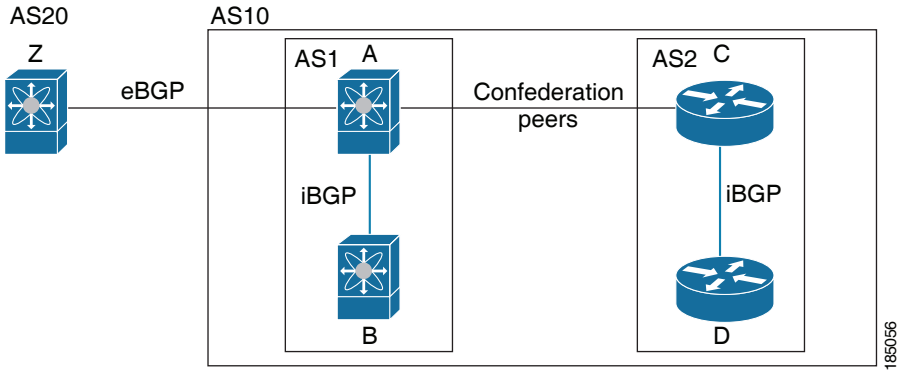
この項では、次のトピックについて取り上げます。

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。AS を複数のサブ AS 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 10-2 に、図 10-1 の BGP ネットワークを 2 つのサブ AS に分割し、1 つの連合にしたものを示します。

図 10-2　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 連合を使用することによって、図 10-1 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

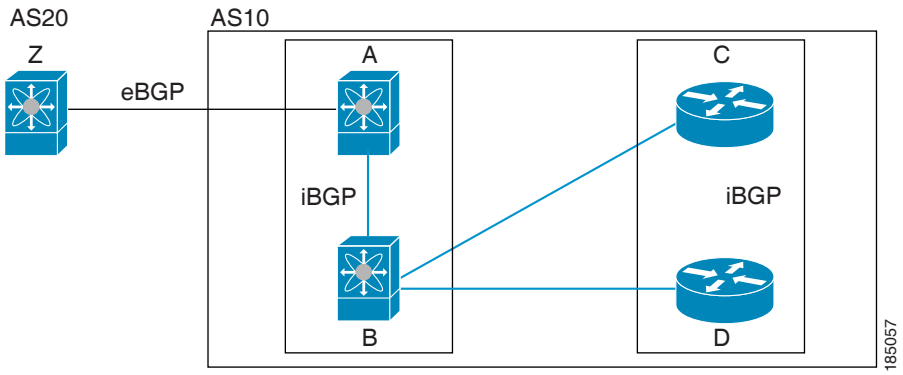
すべての iBGP ピアが完全に一致する必要がないように、ルート リフレクタが学習したルートをネイバーに渡すルート リフレクタ構成を使用することによって、iBGP メッシュを削減できます。

図 10-1 に、メッシュの iBGP スピーカを 4 つ（ルータ A、B、C、D）使用する、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。

図 10-3 では、ルータ B がルート リフレクタです。ルータ A からアドバタイズされたルートを受信したルート リフレクタは、そのルートをルータ C および D にアドバタイズ（リフレクション）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はなくなります。

図 10-3　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。他のマルチプロトコル設定（IPv6 など）の場合は、機能ネゴシエーションが不可欠です。

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 ルートの伝播を最小限に抑える 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 AS 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、プレフィックスごとの複数パスの送受信と、このパスのアドバタイジングをサポートします。詳細については、「BGP の追加パス」を参照してください。

（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。

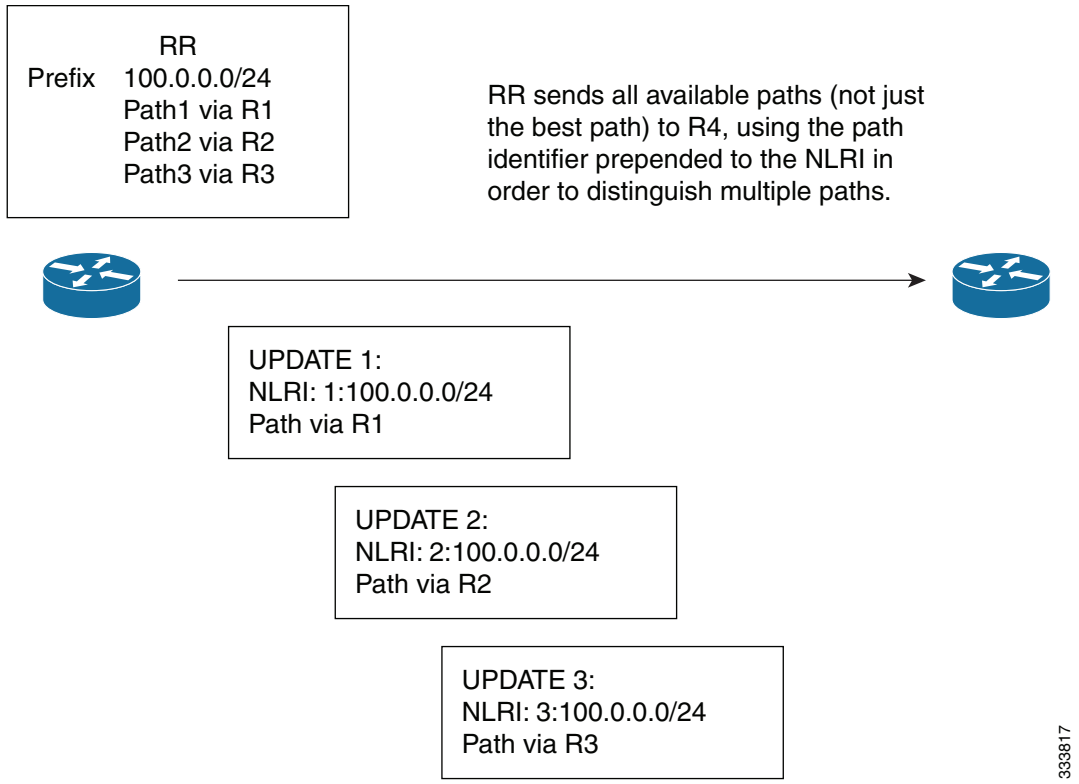
（注）　iBGP マルチパスに関してルート リフレクタを設定すると、ルート リフレクタが、選択されたベスト パスをピアにアドバタイズします。そのパスのネクスト ホップは変更されません。

BGP の追加パス

1 つの BGP 最良パスだけがアドバタイズされ、BGP スピーカは特定ピアからの特定プレフィックスの 1 パスだけを受け入れます。BGP スピーカが同じセッション内で同じプレフィックスの複数のパスを受信した場合、最新のアドバタイズメントを使用します。

BGP は、以前のパスに代わる新しいパスなしで、BGP スピーカが同じプレフィックスに対して複数のパスを伝播し、受け入れることを可能にする追加のパス機能をサポートします。この機能は、BGP スピーカのピアが、プレフィックスごとの複数パスのアドバタイズおよび受信をサポートし、また、そのパスのアドバタイズをサポートするかどうかネゴシエートすることを可能にします。特別な 4 バイトのパス ID は、ピア セッションを介して送信される同じプレフィックスに対して複数のパスを区別するため、ネットワーク層到達可能性情報（NLRI）に追加されます。図 10-4 は、BGP パスの追加機能について説明します。

図 10-4　追加パスの機能を持つ BGP ルート アドバタイズメント



BGP 追加パス設定の詳細については、「BGP 追加パスの設定」セクション（10-28 ページ）を参照してください。

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 AS 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。詳細については、「BGP 条件付きアドバタイズメントの設定」セクション（10-41 ページ）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 最適パスの選択、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ルーティング情報ベース（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 ホップへの完全再帰のインテリア ゲートウェイ プロトコル（IGP）メトリックは変更されます。
- 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。

- クリティカルなイベントとは、異なるパスに対してスイッチオーバーの原因となるネクストホップの消失など、ネクスト ホップの到達可能性に関連しています。異なるパスに対してスイッチオーバーの原因となるネクスト ホップの IGP メトリックの変更は、クリティカルなイベントと見なすことができます。

- 非クリティカルなイベントとは、最適パスに影響を与えたり、単一のネクスト ホップに IGP メトリックを変更したりせずに追加されるネクスト ホップに関連しています。

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」セクション（10-26 ページ）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。再配布を指定したルート マップを設定して、どのルートが BGP に渡されるかを制御する必要があります。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。詳細については、第 15 章「Route Policy Manager の設定」を参照してください。デフォルトでは、iBGP は IGP に再配布されません。

ルート マップを使用して両シナリオのデフォルト動作を無効にできますが、ルート マップの正しくない使用によってネットワーク ループが発生することがあるため、そうする場合は注意が必要です。次に、デフォルトの動作の変更にルート マップを使用する例を示します。

ルート マップの変更によって、シナリオ 1 のデフォルトの動作を次のように変更できます。

```
route-map foo permit 10
   match route-type internal
router ospf 1
   redistribute bgp 100 route-map foo
```

同様に、ルート マップの変更によって、シナリオ 2 のデフォルトの動作を次のように変更できます。

```
route-map foo deny 10
  match route-type internal
router ospf 1
   vrf bar
     redistribute bgp 100 route-map foo
```

BFD

この機能は、IPv4 および IPv6 の Bidirectional Forwarding Detection（BFD）をサポートします。BFD は、転送パスの障害を高速で検出することを目的にした検出プロトコルです。BFD は 2 台の隣接デバイス間のサブセカンド障害を検出し、BFD の負荷の一部を、サポートされるモジュール上のデータ プレーンに分散できるため、プロトコル hello メッセージよりも CPU を使いません。

BGP の BFD は eBGP ピアおよび iBGP シングルホップ ピアでサポートされます。BFD を使用している iBGP シングル ホップ ピアのネイバー コンフィギュレーション モードでアップデート送信元オプションを設定します。

（注）　BFD は他の iBGP ピアまたはマルチ ホップ eBGP ピアではサポートされていません。

詳細については、*『Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide』*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。

この項では、次のトピックについて取り上げます。

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 Multi-Exit Discriminator（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。BGP ではたとえば、IPv4 ユニキャスト ルーティング用のルート セットを 1 つ、IPv4 マルチキャスト ルーティング用のルート セットを 1 つ、さらに IPv6 マルチキャスト ルーティング用のルート セットを 1 つ伝送できます。

（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。

RFC 5549

Cisco NX-OS Release 7.0(3)I2(1) 以降では、BGP は IPv4 プレフィクスを IPv6 ネクスト ホップで伝送できる RFC 5549 をサポートしています。BGP がすべてのホップで動作し、すべてのルータが IPv4 および IPv6 トラフィックを転送できるため、ルータ間の IPv6 トンネルをサポートする必要はありません。BGP は、IPv6 ルートを介した IPv4 を Unicast Route Information Base（URIB）にインストールします。

グレースフル リスタートおよびハイ アベイラビリティ

Cisco NX-OS は、BGP の無停止フォワーディングおよびグレースフル リスタートをサポートします。

BGP ルーティング プロトコル情報がフェールオーバー後に復元されている間に、転送情報ベース（FIB）内の既知のルートでデータ パケットを転送するように、BGP の無停止フォワーディング（NSF）を使用できます。NSF では、BGP ピアはルーティング フラップと無縁です。フェールオーバー時に、データ トラフィックはインテリジェント モジュール経由で転送され、スタンバイ スーパーバイザがアクティブになります。

Cisco NX-OS ルータでコールド リブートが発生した場合、ネットワークはルータにトラフィックを転送しないで、ネットワーク トポロジからルータを削除します。この状況では、BGP は非グレースフル リスタートになり、すべてのルートが削除されます。Cisco NX-OS はスタートアップ コンフィギュレーションを適用し、BGP はピアリング セッションを再び確立して、ルートを再学習します。

Cisco NX-OS デュアル スーパーバイザ構成のルータでは、ステートフル スーパーバイザ スイッチオーバーが実行されます。スイッチオーバーの間、BGP は無停止フォワーディングを使用し、FIB の情報に基づいてトラフィックを転送します。システムがネットワーク トポロジから取り除かれることはありません。ネイバーが再起動しているルータは、「ヘルパー」と呼ばれます。スイッチオーバー後、グレースフル リスタート動作が開始されます。この処理が進行中の際、2 つのルータはネイバー関係を再確立し、これらの BGP ルートを交換します。それらネイバー関係が再起動したとしても、ヘルパーは再起動中のピアを指すプレフィックスを転送し続け、再起動中のルータはピアへトラフィックを転送し続けます。再起動中のルータがグレースフル リスタート可能なすべての BGP ピアを持つ場合、グレースフル リスタートが完了し、BGP は再び動作可能なネイバーを通知します。

ネイバー コンフィギュレーション モードで update-source が設定された単一ホップ iBGP ピアでは、ピアは高速外部フェールオーバーをサポートします。

追加 BGP パス機能により、特定のプレフィックスにアドバタイズされるパス数が再起動の前後で同じ場合、パス ID の選択は古いパスの最終状態および削除を保証します。いくつかのパスが指定されたプレフィックスにアドバタイズされる場合、古いパスがグレースフル リスタート ヘルパー ピアに発生する可能性があります。

メモリ不足の処理

BGP は、次の条件でメモリ不足に対処します。

- マイナー アラート：BGP は新しい eBGP ピアを確立しません。BGP は新しい iBGP ピアおよび連合ピアの確立は続行します。確立されたピアは存続しますが、リセット ピアは再確立されません。

- 重大アラート：BGP は、メモリ アラートがマイナーになるまで、選択した確立済み eBGP ピアを 2 分おきにシャット ダウンします。eBGP ピアごとに、受信したパスの合計数と最適パスとして選択されたパスの数の比率が計算されます。比率が最高のピアが、メモリ使用状況を削減するためのシャット ダウン対象として選択されます。オシレーションを回避するために、シャットダウンされた eBGP ピアを復帰する前にその eBGP ピアをクリアする必要があります。

（注）　重要な eBGP ピアをこの選択プロセスから除外できます。

- クリティカル アラート：BGP は確立されたすべてのピアを正常にシャット ダウンします。シャットダウンされた eBGP ピアを復帰する前にその eBGP ピアをクリアする必要があります。

メモリ不足状態によるシャットダウンから BGP ピアを除外する詳細については、「BGP の調整」セクション（10-47 ページ）を参照してください。

仮想化のサポート

1 台の BGP インスタンスを設定できます。BGP は、仮想ルーティング/転送（VRF）インスタンスをサポートします。

拡張 BGP のライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	BGP には、Enterprise Services ライセンスが必要です。Cisco NX-OS のライセンス スキームとライセンスの取得および適用方法の詳細は、*『Cisco NX-OS Licensing Guide』*を参照してください。

拡張 BGP の前提条件

拡張 BGP の前提条件は次のとおりです。

- BGP をイネーブルにします（「BGP の有効化」セクション（9-12 ページ）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

拡張 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 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。
- Cisco NX-OS は、マルチ ホップ BFD をサポートしません。BGP 用 BFD に関する制約事項は、次のとおりです。
  - BFD は、eBGP ピアおよび iBGP シングル ホップ ピアでのみサポートされます。
  - iBGP の単一ホップ ピアに対して BFD をイネーブルにするには、物理インターフェイスの update-source オプションを設定します。
  - BFD は、マルチ ホップ iBGP ピアおよびマルチ ホップ eBGP ピアではサポートされません。
  - BGP はプレフィクスベースのピアをサポートしますが、BFD はプレフィクスベースのピアではサポートされません。
- **remove-private-as** コマンドには、次のガイドラインと制限事項が適用されます。
  - これは、eBGP ピアにだけ適用されます。
  - ネイバー コンフィギュレーション モードだけで設定可能となり、ネイバー アドレス ファミリ モードでは設定できません。
  - AS パスにプライベートとパブリック AS 番号を含める場合、プライベート AS 番号は削除されません。
  - AS パスに eBGP ネイバーの AS 番号が含まれている場合、プライベート AS 番号は削除されません。
  - その AS パス内のすべての AS 番号がプライベート AS 番号範囲に属する場合のみ、プライベート AS 番号は削除されます。ピアの AS 番号または非プライベート AS 番号が AS パス セグメントに存在する場合、プライベート AS 番号は削除されません。
- **aggregate-address** コマンドを使用して集約アドレスを設定し、**suppress-fib-pending** コマンドを使用して BGP ルートを抑制すると、集約の無損失トラフィックは BGP またはシステムのトリガーで保証できません。

- スイッチで FIB 抑制をイネーブルにした場合、ルートのプログラミングがハードウェアで失敗すると、BGP はハードウェアでローカルにプログラミングされていないルートをアドバタイズします。

拡張 BGP のデフォルト設定

表 10-1 に、拡張 BGP パラメータのデフォルト設定値を示します。

表 10-1　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
BGP の追加パス	ディセーブル
ホールド タイマー	180 秒
キープアライブ インターバル	60 秒
ダイナミック機能	イネーブル

Configuring Advanced BGP

この項では、次のトピックについて取り上げます。

- ロード シェアリングおよび ECMP の設定（10-39 ページ）
- 最大プレフィックス数の設定（10-39 ページ）
- ダイナミック機能の設定（10-39 ページ）
- 集約アドレスの設定（10-40 ページ）
- BGP ルートの抑制（10-40 ページ）
- BGP 条件付きアドバタイズメントの設定（10-41 ページ）
- ルートの再配布の設定（10-43 ページ）
- デフォルト ルートのアドバタイズ（10-44 ページ）
- マルチプロトコル BGP の設定（10-46 ページ）
- BGP の調整（10-47 ページ）
- グレースフル リスタートの設定（10-51 ページ）
- 仮想化の設定（10-52 ページ）

（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。

インターフェイスでの IP 転送のイネーブル化

RFC 5549 を使用するには、少なくとも 1 つの IPv4 アドレスを設定する必要があります。IPv4 アドレスを設定しない場合は、IP 転送機能をイネーブルにして RFC 5549 を使用する必要があります。

1. **configure terminal**
2. **interface** *type slot/port*
3. **ip forward**
4. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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:
	`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。

	コマンド	目的

ステップ 3	`ip forward`
	
	例:
	`switch(config-if)# ip forward`
		対象のインターフェイスに IP アドレスが設定されていない場合でも、そのインターフェイス上の IPv4 トラフィックを許可します。

ステップ 4	`copy running-config startup-config`
	
	例:
	`switch(config-if)# copy running-config startup-config`
		(任意)この設定の変更を保存します。

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、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。

はじめる前に

BGP をイネーブルにします(「BGP の有効化」セクション(9-12 ページ)を参照)。

(注) テンプレートを編集するときには、ピアまたはテンプレートのレベルで **no** 形式のコマンドを使用すると、テンプレートの設定を明示的に上書きできます。属性をデフォルトの状態にリセットするには、**default** 形式のコマンドを使用する必要があります。

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	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** *autonomous-system-number*
	
	例:
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGP をイネーブルにして、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。

ステップ 4	**password** *number password* **例**: `switch(config-router-stmp)# password 0 test`	(任意)ネイバーにクリアテキスト パスワード *test* 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	**timers** *keepalive hold* **例**: `switch(config-router-stmp)# timers 30 90`	(任意)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 65535`
	`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	**description** *text*
	
	例:
	`switch(config-router-neighbor)# description Peer Router A`
	`switch(config-router-neighbor)`
		(任意)ネイバーの説明を追加します。

	コマンド	目的
ステップ 10	**show bgp peer-session** *template-name* **例:** `switch(config-router-neighbor)# show bgp peer-session BaseSession`	(任意) peer-policy テンプレートを表示します。
ステップ 11	**copy running-config startup-config** **例:** `switch(config-router-neighbor)# copy running-config startup-config`	(任意) この設定の変更を保存します。

適用されたテンプレートを確認するには、**show bgp neighbor** コマンドを使用します。

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 65535
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 65535
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。

はじめる前に

BGP をイネーブルにします(「BGP の有効化」セクション (9-12 ページ)を参照)。

(注) テンプレートを編集するときには、ピアまたはテンプレートのレベルで **no** 形式のコマンドを使用すると、テンプレートの設定を明示的に上書きできます。属性をデフォルトの状態にリセットするには、default 形式のコマンドを使用する必要があります。

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** | **ipv6**} {**multicast** | **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	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** *autonomous-system-number*
	
	例:
	`switch(config)# router bgp 65535`
	`switch(config-router)#`
		BGP をイネーブルにして、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	**advertise-active-only** **例**: `switch(config-router-ptmp)# advertise-active-only`	(任意)アクティブ ルートだけをピアにアドバタイズします。
ステップ 5	**maximum-prefix** *number* **例**: `switch(config-router-ptmp)# maximum-prefix 20`	(任意)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。

	コマンド	目的

ステップ 7	**neighbor** *ip-address* **remote-as** *as-number*
	
	例:
	`switch(config-router)# neighbor 192.168.1.2 remote-as 65535`
	`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	**address-family** {**ipv4** | **ipv6**} {**multicast** | **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	**show bgp peer-policy** *template-name* **例**: `switch(config-router-neighbor-af)# show bgp peer-policy BasePolicy`	(任意) peer-policy テンプレートを表示します。
ステップ 11	**copy running-config startup-config** **例**: `switch(config-router-neighbor-af)# copy running-config startup-config`	(任意) この設定の変更を保存します。

適用されたテンプレートを確認するには、**show bgp neighbor** コマンドを使用します。

BGP peer-policy 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 65535
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 65535
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。

はじめる前に

BGP をイネーブルにします（「BGP の有効化」セクション（9-12 ページ）を参照）。

（注）　テンプレートを編集するときには、ピアまたはテンプレートのレベルで **no** 形式のコマンドを使用すると、テンプレートの設定を明示的に上書きできます。属性をデフォルトの状態にリセットするには、default 形式のコマンドを使用する必要があります。

手順の概要

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer** *template-name*
4. **inherit peer-session** *template-name*
5. **address-family** {**ipv4** | **ipv6**} {**multicast** | **unicast**}
6. **inherit peer** *template-name*
7. **exit**
8. **timers** *keepalive hold*
9. **exit**
10. **neighbor** *ip-address* **remote-as** *as-number*
11. **inherit peer** *template-name*
12. **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	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** *autonomous-system-number* **例**: `switch(config)# router bgp 65535`	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	**template peer** *template-name*
	
	例:
	`switch(config-router)# template peer BasePeer`
	`switch(config-router-neighbor)#`
		peer 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	**inherit peer-session** *template-name* **例**: `switch(config-router-neighbor)# inherit peer-session BaseSession`	（任意）peer テンプレートで peer-session テンプレートを継承します。

	コマンド	目的

ステップ 5	**address-family** {**ipv4** | **ipv6**} {**multicast** | **unicast**}
	
	例:
	`switch(config-router-neighbor)# address-family ipv4 unicast`
	`switch(config-router-neighbor-af)#`
		(任意)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	**inherit peer** *template-name* **例**: `switch(config-router-neighbor-af)# inherit peer BasePolicy`	(任意)ネイバー アドレス ファミリ設定に peer 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	**timers** *keepalive hold* **例**: `switch(config-router-neighbor)# timers 45 100`	(任意)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-neighbor)# 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 65535`
	`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	**timers** *keepalive hold* **例**: `switch(config-router-neighbor)# timers 60 120`	(任意)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。
ステップ 13	**show bgp peer-template** *template-name* **例**: `switch(config-router-neighbor-af)# show bgp peer-template BasePeer`	(任意)peer テンプレートを表示します。
ステップ 14	**copy running-config startup-config** **例**: `switch(config-router-neighbor-af)# copy running-config startup-config`	(任意)この設定の変更を保存します。

適用されたテンプレートを確認するには、**show bgp neighbor** コマンドを使用します。

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 65535
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 65535
switch(config-router-neighbor)# inherit peer BasePeer
switch(config-router-neighbor)# copy running-config startup-config
```

プレフィックス ピアリングの設定

BGP では IPv4 および IPv6 の両方のプレフィックスを使用したピア セットの定義がサポートされます。この機能を使用すると、各ネイバーを設定に追加する必要がありません。

プレフィックス ピアリングを定義する場合は、プレフィックスとともにリモート自律システム番号を指定する必要があります。プレフィックス ピアリングが設定されている許容最大ピア数を超えない場合、BGP はプレフィックスおよび自律システムから接続するピアを受け付けます。

プレフィックス ピアリングに含まれている BGP ピアが切断されると、Cisco NX-OS は定義されているプレフィックス ピア タイムアウト値まで、ピア構造を維持します。この場合、そのプレフィックス ピアリングのすべてのスロットを他のピアが使い果たした結果、ブロックされるという危険性を伴わずに、確立されたピアのリセットまたは再接続が可能になります。

BGP プレフィクス ピアリング タイムアウト値を設定するには、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
timers prefix-peer-timeout *value* **例**: `switch(config-router-neighbor)# timers prefix-peer-timeout 120`	プレフィックス ピアリングのタイムアウト値を設定します。有効な範囲は 0 ～ 1200 秒です。デフォルト値は 30 です。

ピアの最大数を設定するには、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
maximum-peers *value* **例**: `switch(config-router-neighbor)# maximum-peers 120`	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。

最大 10 のピアを受け付けるプレフィックス ピアリングの設定例を示します。

```
switch# configure terminal
switch(config)# router bgp 65535
switch(config-router)# timers prefix-peer-timeout 120
switch(config-router)# neighbor 10.100.200.0/24 remote-as 65535
```

```
switch(config-router-neighbor)# maximum-peers 10
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)#
```

show bgp ipv4 unicast neighbors コマンドを使用すると、所定のプレフィックス ピアリングの設定の詳細とともに、現在受け付けられているインスタンスのリスト、アクティブ ピア数、最大同時ピア数、および受け付けたピアの合計数を表示できます。

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 アップデートを認証するように 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 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
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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} *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 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
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イネーブルであり、ディセーブル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 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。

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。

セッションがダウンした場合のネクストホップ グループの縮小

セッションがダウンしたときに迅速な方法で ECMP グループを縮小するように BGP を設定できます。

この機能は、次の BGP パス障害イベントに適用されます。

- 1 つまたは複数のレイヤ 3 リンクの障害
- ラインカード障害
- BGP ネイバーの BFD 障害検出
- BGP ネイバーの管理上のシャットダウン（**shutdown** コマンドを使用）

最初の 2 つのイベント（レイヤ 3 リンク障害とラインカード障害）の迅速な処理はデフォルトでイネーブルになっており、イネーブルにするためのコンフィギュレーション コマンドは必要ありません。

最後の 2 つのイベントの迅速な処理を設定するには、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
neighbor-down fib-accelerate **例**: `switch(config-router)# neighbor-down fib-accelerate`	BGP セッションがダウンするたびに、すべてのネクストホップ グループ（ECMP グループと単一のネクストホップ ルート）から対応する次のネクスト ホップを取り消します。 **注**　このコマンドは、IPv4 と IPv6 の両方のアドレスファミリ ルートに適用されます。

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。

ポリシーのバッチ処理のディセーブル化

プレフィクスが固有の属性を持っている BGP 展開では、BGP は類似の属性を持つルートを識別して同じ BGP アップデート メッセージにバンドルしようとします。この追加の BGP プロセスのオーバーヘッドを回避するには、バッチ処理をディセーブルにします。

シスコは、固有のネクスト ホップを持つルートが大量に存在する BGP 展開では、ポリシーのバッチ処理をディセーブルにすることを推奨します。

ポリシーのバッチ処理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

disable-policy-batching
 	
例:
`switch(config-router)#`
`disable-policy-batching`
	すべてのピアへのプレフィクスのアドバタイズメントのバッチ処理評価をディセーブルにします。

BGP 追加パスの設定

BGP は、プレフィックスごとの複数パスの送受信と、このパスのアドバタイジングをサポートします。ここでは、次の内容について説明します。

- 追加パスの送受信機能のアドバタイズ（10-29 ページ）
- 追加パスの送受信の設定（10-29 ページ）
- アドバタイズされたパスの設定（10-30 ページ）
- 追加パス選択の設定（10-31 ページ）

追加パスの送受信機能のアドバタイズ

BGP ピア間の追加パスの送受信機能をアドバタイズするように BGP を設定できます。これを行うには、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
[**no**] **capability additional-paths send** [**disable**] **例**: `switch(config-router-neighbor-af)# capability additional-paths send`	BGP ピアに追加パスを送信する機能をアドバタイズします。**disable** オプションは、追加パス送信機能のアドバタイズをディセーブルにします。 このコマンドの **no** 形式は、追加パスの送信機能をディセーブルにします。
[**no**] **capability additional-paths receive** [**disable**] **例**: `switch(config-router-neighbor-af)# capability additional-paths receive`	BGP ピアから追加パスを受信する機能をアドバタイズします。**disable** オプションは、追加パス受信機能のアドバタイズをディセーブルにします。 このコマンドの **no** 形式は、追加パスの受信機能をディセーブルにします。
show bgp neighbor **例**: `switch(config-router-neighbor-af)# show bgp neighbor`	ローカル ピアがリモート ピアへの追加パス送受信機能をアドバタイズしたかを表示します。

BGP ピアに追加のパスを送受信する機能をアドバタイズする BGP の設定例を示します。

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# neighbor 10.131.31.2 remote-as 100
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# capability additional-paths send
switch(config-router-neighbor-af)# capability additional-paths receive
```

追加パスの送受信の設定

BGP ピア間の追加パスの送受信機能を設定できます。これを行うには、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
[**no**] **additional-paths send** **例**: `switch(config-router-af)# additional-paths send`	機能がディセーブルになっていないこのアドレス ファミリで、すべてのネイバーの追加パスの送信機能をイネーブルにします。 このコマンドの **no** 形式を使用すると、送信機能がディセーブルになります。
[**no**] **additional-paths receive** **例**: `switch(config-router-af)# additional-paths receive`	機能がディセーブルになっていないこのアドレス ファミリで、すべてのネイバーの追加パスの受信機能をイネーブルにします。 このコマンドの **no** 形式を使用すると、受信機能がディセーブルになります。

コマンド	目的
show bgp neighbor **例**: `switch(config-router-af)# show bgp neighbor`	ローカル ピアがリモート ピアへの追加パス送受信機能をアドバタイズしたかを表示します。

機能がディセーブルになっていない指定されたアドレス ファミリで、すべてのネイバーの追加パスの受信機能をイネーブルにする例を示します。

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# additional-paths send
switch(config-router-af)# additional-paths receive
```

アドバタイズされたパスの設定

BGP にアドバタイズされたパスを指定できます。これを行うには、ルート マップ 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**] **set ip next-hop unchanged** **例**: `switch(config-route-map)# set ip next-hop unchanged`	不変のネクスト ホップ IP アドレスを指定します。
[**no**] **set path-selection all advertise** **例**: `switch(config-route-map)# set path-selection all advertise`	すべてのパスが指定されたプレフィックスにアドバタイズされるよう指定します。 このコマンドの **no** 形式は、最適パスだけがアドバタイズされるよう指定します。
show bgp {**ipv4** | **ipv6**} **unicast** [*ip-address* | *ipv6-prefix*] [**vrf** *vrf-name*] **例**: `switch(config-route-map)# show bgp ipv4 unicast`	プレフィックスの追加パスのパス ID とこれらのパスのアドバタイズメント情報を表示します。

すべてのパスが指定されたプレフィックスにアドバタイズされるよう指定する例を示します。

```
switch# configure terminal
switch(config)# route-map PATH_SELECTION_RMAP
switch(config-route-map)# match ip address prefix-list p1
switch(config-route-map)# set path-selection all advertise
```

追加パス選択の設定

プレフィックスに追加のパスを選択する機能を設定できます。これを行うには、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
[**no**] **additional-paths selection route-map** *map-name* **例**: `switch(config-router-af)# additional-paths selection route-map map1`	プレフィックスに追加のパスを選択する機能を設定します。 このコマンドの **no** 形式は、追加パス選択機能をディセーブルにします。
show bgp {**ipv4** | **ipv6**} **unicast** [*ip-address* | *ipv6-prefix*] [**vrf** *vrf-name*] **例**: `switch(config-route-af)# show bgp ipv4 unicast`	プレフィックスの追加パスのパス ID とこれらのパスのアドバタイズメント情報を表示します。

指定されたアドレス ファミリで追加パス選択を設定する例を示します。

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# additional-paths selection route-map PATH_SELECTION_RMAP
```

eBGP の設定

ここでは、次の内容について説明します。

- eBGP シングルホップ チェックのディセーブル化（10-31 ページ）
- eBGP マルチホップの設定（10-32 ページ）
- 高速外部フォールオーバーのディセーブル化（10-32 ページ）
- AS パス属性の制限（10-32 ページ）
- ローカル AS サポートの設定（10-33 ページ）

eBGP シングルホップ チェックのディセーブル化

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ディセーブルにするように、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
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ディセーブルにします。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 TTL（存続可能時間）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。

高速外部フォールオーバーのディセーブル化

Cisco NX-OS デバイスは、すべての VRF のネイバーおよびアドレス ファミリ（IPv4 または IPv6）の高速外部フォールオーバーをデフォルトでサポートします。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 が高速外部フォールオーバーを開始します。この高速外部フォ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。

高速外部フォ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-fallover **例**: `switch(config-router)# no fast-external-fallover`	eBGP ピアの高速外部フォ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非常に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 番号が非常に高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、別の自律システム（AS）のメンバ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。

この機能は、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`	ローカルの AS *番号*を AS_PATH 属性に追加するために eBGP を設定します。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。

次に、VRF のローカル AS サポートを設定する例を示します。

```
switch# configure terminal
switch(config)# router bgp 1
switch(config-router)# vrf test
switch(config-router-vrf)# local-as 1
switch(config-router-vrf)# show running-config bgp
```

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外部で認識されます。

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 4000`	AS 連合を表す連合 ID 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。

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 AS 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。

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。

はじめる前に

BGP をイネーブルにします（「BGP の有効化」セクション（9-12 ページ）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** | **ipv6**} {**unicast** | **multicast**}
5. （任意）**client-to-client reflection**
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
9. **route-reflector-client**
10. （任意）**show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors**
11. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 65535`
	`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	`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。

	コマンド	目的

ステップ 4	**address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
	
	例:
	`switch(config-router)# address-family ipv4 unicast`
	`switch(config-router-af)#`
		指定のアドレス ファミリに対しルータ 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	**client-to-client reflection** **例**: `switch(config-router-af)# client-to-client reflection`	(任意) 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)# exit`
	`switch(config-router)#`
		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 192.0.2.10 remote-as 65535`
	`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** | **ipv6**} {**unicast** | **multicast**}
	
	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。

ステップ 9	**route-reflector-client** **例**: `switch(config-router-neighbor-af)# route-reflector-client`	BGP ルート リフレクタとしてデバイスを設定し、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** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors** **例:** `switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors`	(任意) BGP ピアを表示します。
ステップ 11	**copy running-config startup-config** **例:** `switch(config-router-neighbor-af)# copy running-config startup-config`	(任意) この設定の変更を保存します。

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# configure terminal
switch(config)# router bgp 65535
switch(config-router)# neighbor 192.0.2.10 remote-as 65535
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
```

アウトバウンド ルート マップを使用した、反映されたルートのネクスト ホップの設定

アウトバウンド ルート マップを使用して、BGP ルート リフレクタの反映されたルートのネクスト ホップを変更できます。ネクストホップ アドレスとしてピアのローカル アドレスを指定するため、アウトバウンド ルート マップを設定できます。

（注）　**next-hop-self** コマンドは、ルート リフレクタによってクライアントに反映されるルートに対するこの機能を有効にしません。この機能は、アウトバウンド ルート マップを使用した場合にだけイネーブルにできます。

はじめる前に

BGP をイネーブルにします（「BGP の有効化」セクション（9-12 ページ）を参照）。

アドレス ファミリ固有のネクスト ホップ アドレスを設定するには、**set next-hop** コマンドを入力する必要があります。たとえば、IPv6 アドレス ファミリには、**set ipv6 next-hop peer-address** コマンドを入力します。

- ルート マップを使用して IPv4 ネクスト ホップを設定する場合：**set ip next-hop peer-address** がルート マップに一致する場合、ネクスト ホップはピアのローカル アドレスに設定されます。ネクスト ホップがルート マップで設定されていない場合、ネクスト ホップはパスに保存されているネクスト ホップに設定されます。
- ルート マップを使用して IPv6 ネクスト ホップを設定する場合：**set ipv6 next-hop peer-address** がルート マップに一致する場合、ネクスト ホップは次のとおり設定されます。
  - IPv6 ピアでは、ネクスト ホップはピアのローカル IPv6 アドレスに設定されます。
  - IPv4 ピアでは、**update-source** が設定されている場合、ネクスト ホップは、もしあれば、発信元インターフェイスの IPv6 アドレスに設定されます。IPv6 アドレスが設定されていない場合、ネクスト ホップは設定されません。
  - IPv4 ピアでは、**update-source** が設定されていない場合、ネクスト ホップは、もしあれば、発信インターフェイスの IPv6 アドレスに設定されます。IPv6 アドレスが設定されていない場合、ネクスト ホップは設定されません。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-number*
4. （任意）**update-source** *interface number*
5. **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
6. **route-reflector-client**
7. **route-map** *map-name* **out**
8. （任意）**show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors**
9. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 200`
	`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 192.0.2.12 remote-as 200`
	`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 4	**update-source** *interface number* **例**: `switch(config-router-neighbor)# update-source loopback 300`	(任意)BGP セッションの送信元を指定し、更新します。

ステップ 5	**address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
	
	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 6	**route-reflector-client** **例**: `switch(config-router-neighbor-af)# route-reflector-client`	BGP ルート リフレクタとしてデバイスを設定し、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 7	**route-map** *map-name* **out** **例**: `switch(config-router-neighbor-af)# route-map setrrnh out`	発信ルートに設定された BGP ポリシーを適用します。
ステップ 8	**show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} [**ip-address** | **ipv6-prefix**] **route-map** *map-name* [**vrf** *vrf-name*] **例**: `switch(config-router-neighbor-af)# show bgp ipv4 unicast route-map setrrnh`	(任意)ルート マップと一致する BGP ルートを表示します。
ステップ 9	**copy running-config startup-config** **例**: `switch(config-router-neighbor-af)# copy running-config startup-config`	(任意)この設定の変更を保存します。

アウトバウンド ルート マップを使用して、BGP ルート リフレクタの反映されたルートのネクスト ホップを設定する例を示します。

```
switch# configure terminal
switch(config)# interface loopback 300
switch(config-if)# ip address 192.0.2.11/32
switch(config-if)# ipv6 address 2001::a0c:1a65/64
switch(config-if)# ip router ospf 1 area 0.0.0.0
switch(config-if)# exit
switch(config)# route-map setrrnh permit 10
switch(config-route-map)# set ip next-hop peer-address
switch(config-route-map)# exit
switch(config)# route-map setrrnhv6 permit 10
switch(config-route-map)# set ipv6 next-hop peer-address
switch(config-route-map)# exit
switch(config)# router bgp 200
switch(config-router)# neighbor 192.0.2.12 remote-as 200
switch(config-router-neighbor)# update-source loopback 300
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# route-map setrrnh out
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# address-family ipv6 unicast
switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)# route-map setrrnhv6 out
```

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。

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 8`	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。デフォルトは 1 です。

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* | **warning-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。

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** キーワードで、自律システム セット パス情報および関係するパスに基づくコミュニティ情報が生成されます。 • **summary-only** キーワードによって、アップデートから固有性の強いルートがすべてフィルタリングされます。 • **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によって、固有性の強いルートを条件付きでフィルタリングします。

BGP ルートの抑制

新しく学習された BGP ルートが転送情報ベース（FIB）により確認され、ハードウェアでプログラミングされた後にのみ、これらのルートをアドバタイズするように Cisco NX-OS を設定できます。ルートがプログラミングされた後は、これらのルートに対する以降の変更にはこのハードウェア プログラミングのチェックは必要ありません。

BGP ルートを抑制するには、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
suppress-fib-pending **例**: `switch(config-router)# suppress-fib-pending`	新しく学習された BGP ルート（IPv4 または IPv6）がハードウェアでプログラミングされるまで、ダウンストリームの BGP ネイバーにアドバタイズされることを抑制します。

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。

はじめる前に

BGP をイネーブルにします（「BGP の有効化」セクション（9-12 ページ）を参照）。

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** | **ipv6**} {**unicast** | **multicast**}
5. **advertise-map** *adv-map* {**exist-map** *exist-rmap* | **non-exist-map** *nonexist-rmap*}
6. （任意）**show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors**
7. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 65535`
	`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 65534`
	`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。

	コマンド	目的

ステップ 4	**address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
	
	例:
	`switch(config-router-neighbor)# address-family ipv4 multicast`
	`switch(config-router-neighbor-af)#`
		アドレスファミリ コンフィギュレーション モードに入ります。

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** ステートメントを使用してルート マップを指定します。*adv-map* は、大文字と小文字が区別される 63 文字以下の英数字文字列です。 • *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* は、大文字と小文字が区別される 63 文字以下の英数字文字列です。 • *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* は、大文字と小文字が区別される 63 文字以下の英数字文字列です。
ステップ 6	**show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors** **例**: `switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors`	(任意) 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	**copy running-config startup-config** **例**: `switch(config-router-neighbor-af)# copy running-config startup-config`	(任意) この設定の変更を保存します。

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 65535
switch(config-router)# neighbor 192.0.2.2 remote-as 65534
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 172.16.201.0/27
```

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。

はじめる前に

BGP をイネーブルにします（「BGP の有効化」セクション（9-12 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
4. **redistribute** {**direct** | {**eigrp** | **isis** | **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	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 65535 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**	**ipv6**} {**unicast**	**multicast**} **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 モードに入ります。

	コマンド	目的
ステップ 4	**redistribute** {**direct** | {**eigrp** | **isis** | **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 マップの詳細については、「ルート マップの設定」セクション（15-13 ページ）を参照してください。
ステップ 5	**default-metric** *value* **例**: `switch(config-router-af)# default-metric 33`	（任意）BGP へのデフォルト ルートを作成します。
ステップ 6	**copy running-config startup-config** **例**: `switch(config-router-af)# copy running-config startup-config`	（任意）この設定の変更を保存します。

次に、EIGRP を BGP に再配布する例を示します。

```
switch# configure terminal
switch(config)# router bgp 65535
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
```

デフォルト ルートのアドバタイズ

デフォルトのルート（ネットワーク 0.0.0.0）をアドバタイズするように BGP を設定できます。

はじめる前に

BGP をイネーブルにします（「BGP の有効化」セクション（9-12 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **route-map allow permit**
3. **exit**
4. **ip route** *ip-address network-mask* **null** *null-interface-number*
5. **router bgp** *as-number*
6. **address-family** {**ipv4** | **ipv6**} **unicast**
7. **default-information originate**
8. **redistribute static route-map allow**
9. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	**route-map allow permit** **例**: switch(config)# route-map allow permit switch(config-route-map)#	ルータのマップ コンフィギュレーション モードを開始し、ルートを再配布する条件を定義します。
ステップ 3	**exit** **例**: switch(config-route-map)# exit switch(config)#	ルータのマップ コンフィギュレーション モードを終了します。
ステップ 4	**ip route** *ip-address network-mask* **null** *null-interface-number* **例**: switch(config)# ip route 192.0.2.1 255.255.255.0 null 0	IP アドレスを設定します。
ステップ 5	**router bgp** *as-number* **例**: switch(config)# router bgp 65535 switch(config-router)#	BGP モードを開始し、AS 番号をローカルの BGP スピーカに割り当てます。
ステップ 6	**address-family** {**ipv4** | **ipv6**} **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。
ステップ 7	**default-information originate** **例**: switch(config-router-af)# default-information originate	デフォルトのルートをアドバタイズします。
ステップ 8	**redistribute static route-map allow** **例**: switch(config-router-af)# redistribute static route-map allow	デフォルトのルートを再配布します。
ステップ 9	**copy running-config startup-config** **例**: switch(config-router-af)# copy running-config startup-config	(任意) この設定の変更を保存します。

マルチプロトコル BGP の設定

複数のアドレス ファミリ（IPv4 および IPv6 のユニキャストおよびマルチキャスト 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。

はじめる前に

BGP をイネーブルにします（「BGP の有効化」セクション（9-12 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-numbe*r
4. **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
5. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 65535`
	`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 65534`
	`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 4	**address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}
	
	例:
	`switch(config-router-neighbor)# address-family ipv4 multicast`
	`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	**copy running-config startup-config** **例**: `switch(config-router-neighbor-af)# copy running-config startup-config`	（任意）この設定の変更を保存します。

次に、ネイバーのマルチキャスト RPF に対して IPv4 および IPv6 ルートのアドバタイズおよび受信をイネーブルにする例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 2/1
switch(config-if)# ipv6 address 2001:0DB8::1
switch(config-if)# router bgp 65535
switch(config-router)# neighbor 192.168.1.2 remote-as 35537
switch(config-router-neighbor)# address-family ipv4 multicast
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# address-family ipv6 multicast
switch(config-router-neighbor-af)# copy running-config startup-config
```

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** | **as-path multipath-relax** | **compare-routerid** | **cost-community ignore** | **med** {**confed** | **missing-as-worst** | **non-deterministic**}] **例**: `switch(config-router)# bestpath always-compare-med`	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 AS からのパスの MED を比較します。 • **as-path multipath-relax**：異なる（ただし長さが等しい）AS パスを持つプロバイダー間でのロード シェアリングを許可します。このオプションを指定しないと、AS パスはロード シェアリングの場合に同一である必要があります。 • **compare-routerid**：同一の eBGP パスのルータ ID を比較します。 • **cost-community ignore**：BGP 最良パスを計算する場合に、コスト コミュニティを無視します。 • **med confed**：コンフェデレーション内を起点とするパス間でのみ MED 比較を実行するよう bestpath を強制します。 • **med missing-as-worst**：脱落 MED を最上位 MED として扱います。 • **med non-deterministic**：同じ AS からのパス間で、必ずしも最適な MED パスを選択しません。
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。 **注**　特定のネイバーのネイバー ステータス変化に関するメッセージを抑制するには、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 **log-neighbor-changes disable** コマンドを使用できます。
router-id *id* **例**: `switch(config-router)# router-id 10.165.20.1`	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。 • *delay*：再起動後の初期最適パス タイムアウト値。有効な範囲は 0 ～ 3600 秒です。デフォルト値は 300 です。 • *keepalive*：BGP セッション キープアライブ タイム。有効な範囲は 0 ～ 3600 秒です。デフォルト値は 60 です。 • *holdtime*：BGP セッション ホールド タイム。指定できる範囲は 0 ～ 3600 秒です。デフォルト値は 180 です。 • *timeout*：プレフィックス ピア タイムアウト値。有効な範囲は 0 ～ 1200 秒です。デフォルト値は 30 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。 • *ebgp-distance*：20。 • *ibgp-distance*：200。 • *local-distance*：220。ローカル ディスタンスは、集約廃棄ルートが RIB に組み込まれている場合に、集約廃棄ルートに使用するアドミニストレーティブ ディスタンスです。

コマンド	目的
log-neighbor-changes [**disable**] **例**: switch(config-router-af)# log-neighbor-changes disable	この特定のネイバーの状態が変化すると、システム メッセージを生成します。 **disable** オプションを使用すると、この特定のネイバーのネイバー ステータス変化に関するメッセージが抑制されます。

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。
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。
[**no** | **default**] **remove-private-as** [**all** | **replace-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自律システム番号を削除します。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。 オプション パラメータは次のとおりです。 • **no**：コマンドをディセーブルにします。 • **default**：デフォルト モードにコマンドを移動します。 • **all**：AS パスからすべてのプライベート AS 番号を削除します。 • **replace-as**：すべてのプライベート AS 番号を replace-as AS-path 値に置き換えます。 **注**　このコマンドの詳細については、「拡張 BGP に関する注意事項と制限事項」を参照してください。

コマンド	目的
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。単一ホップ iBGP ピアでは、**update-source** が設定されている場合に、高速外部フォールオーバーをサポートします。

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
allowas in **例**: `switch(config-router-neighbor-af)# allowas in`	BRIP にインストールする AS パスにルート自体の AS を持つことを可能にします。
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。
disable-peer-as-check **例**: `switch(config-router-neighbor-af)# disable-peer-as-check`	デバイスが同じ AS パスで一方のノードからもう一方のノードに学習されたルートをアドバタイズすると同時に、ピア AS 番号のチェックをディセーブルにします。
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。
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。

コマンド	目的
send-community extended **例**: `switch(config-router-neighbor-af)# send-community extended`	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。
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。

グレースフル リスタートの設定

BGP のグレースフル リスタートを設定し、グレースフル リスタート ヘルパー機能をイネーブルにできます。

はじめる前に

BGP をイネーブルにします（「BGP の有効化」セクション（9-12 ページ）を参照）。

VRF を作成します。

手順の概要

1. **configure terminal**
2. **router bgp** *as-number*
3. **graceful-restart**
4. **graceful-restart** [**restart-time** *time* | **stalepath-time** *time*]
5. **graceful-restart-helper**
6. （任意）**show running-config bgp**
7. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 65535`
	`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。

	コマンド	目的
ステップ 3	**graceful-restart** **例:** switch(config-router)# graceful-restart	グレースフル リスタートおよびグレースフル リスタート ヘルパー機能をイネーブルにします。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。 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	**graceful-restart** [**restart-time** *time* | **stalepath-time** *time*] **例:** switch(config-router)# graceful-restart restart-time 300	グレースフル リスタート タイマーを設定します。 オプション パラメータは次のとおりです。 • **restart-time**：BGP ピアに送信されたリスタートの最大時間。有効な範囲は 1 ～ 3600 秒です。デフォルトは 120 です。 • **stalepath-time**：BGP が再起動中の BGP ピアからの古いルートを維持する最大時間。有効な範囲は 1 ～ 3600 秒です。デフォルトは 300 です。 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 5	**graceful-restart-helper** **例:** switch(config-router)# graceful-restart-helper	グレースフル リスタート ヘルパー機能をイネーブルにします。このコマンドは、グレースフル リスタートをディセーブルにしていながら、グレースフル リスタート ヘルパー機能はイネーブルにする必要がある場合に使用します。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 6	**show running-config bgp** **例:** switch(config-router)# show running-config bgp	(任意) BGP の設定を表示します。
ステップ 7	**copy running-config startup-config** **例:** switch(config-router)# copy running-config startup-config	(任意) この設定の変更を保存します。

次に、グレースフル リスタートをイネーブルにする例を示します。

```
switch# configure terminal
switch(config)# router bgp 65535
switch(config-router)# graceful-restart
switch(config-router)# copy running-config startup-config
```

仮想化の設定

1 つの BGP プロセスを設定し、複数の VRF を作成できます。また、各 VRF で同じ BGP プロセスを使用できます。

はじめる前に

BGP をイネーブルにします(「BGP の有効化」セクション(9-12 ページ)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）**copy running-config startup-config**

手順の詳細

	コマンド	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	**vrf context** *vrf-name*
	
	例:
	`switch(config)# vrf context RemoteOfficeVRF`
	`switch(config-vrf)#`
		新しい VRF を作成し、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	**exit**
	
	例:
	`switch(config-vrf)# exit`
	`switch(config)#`
		VRF コンフィギュレーション モードを終了します。

ステップ 4	**router bgp** *as-number*
	
	例:
	`switch(config)# router bgp 65535`
	`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 コンフィギュレーション 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 65535`
	`switch(config-router--vrf-neighbor)#`
		リモート BGP ピアの IP アドレスおよび AS 番d号を設定します。

ステップ 7	**copy running-config startup-config** **例**: `switch(config-router-vrf-neighbor)# copy running-config startup-config`	(任意) この設定の変更を保存します。

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 65535
switch(config-router)# vrf NewVRF
switch(config-router-vrf)# neighbor 209.165.201.1 remote-as 65535
switch(config-router-vrf-neighbor)# copy running-config startup-config
```

拡張 BGP の設定の確認

BGP 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 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** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] {**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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **neighbors** [*ip-address* | *ipv6-prefix*] [**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。

コマンド	目的
show bgp {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] {**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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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 {**ipv4** | **ipv6**} **bgp** *options*	BGP のステータスと構成情報を表示します。
show {**ipv4** | **ipv6**} **mbgp** *options*	BGP のステータスと構成情報を表示します。
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 統計情報のモニタリング

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 {**ipv4** | **ipv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **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消去するには、**clear bgp flap-statistics** コマンドを使用します。
show bgp {**ipv4** | **ipv6**} **unicast injected-routes**	ルーティング テーブルに挿入されたルートを表示します。

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消去するには、**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消去するには、**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。

設定例

プレフィックス ベース ネイバーの MD5 認証を設定する例を示します。

```
template peer BasePeer-V6
    description BasePeer-V6
    password 3 f4200cfc725bbd28
    transport connection-mode passive
    address-family ipv6 unicast
template peer BasePeer-V4
    bfd
    description BasePeer-V4
    password 3 f4200cfc725bbd28
    address-family ipv4 unicast
--
    neighbor fc00::10:3:11:0/127 remote-as 65006
      inherit peer BasePeer-V6
    neighbor 10.3.11.0/31 remote-as 65006
      inherit peer BasePeer-V4
```

次に、ネイバー ステータスの変化に関するメッセージをグローバルに有効にし、特定のネイバーについてはメッセージを抑制する方法を示します。

```
router bgp 65100
  log-neighbor-changes
    neighbor 209.165.201.1 remote-as 65535
      description test
      address-family ipv4 unicast
        soft-reconfiguration inbound
         disable log-neighbor-changes
```

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。

- 第 9 章「ベーシック BGP の設定」
- 第 15 章「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。

- MIB（10-57 ページ）

MIB

MIB	MIB のリンク
BGP に関連する MIB	サポートされている MIB を検索およびダウンロードするには、次の URL にアクセスしてください。 ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

RIP の設定

この章では、Cisco NX-OS デバイスで Routing Information Protocol（RIP）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。

RIP について

この項では、次のトピックについて取り上げます。

RIP の概要

RIP は UDP(ユーザ データグラム プロトコル)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 の認証」セクション(11-2 ページ)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 コンフィギュレーションの確認」セクション(11-18 ページ)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 の認証

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。キーチェーン作成の詳細については、*『Cisco Nexus 9000 Series NX-OS Security Configuration Guide』*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。

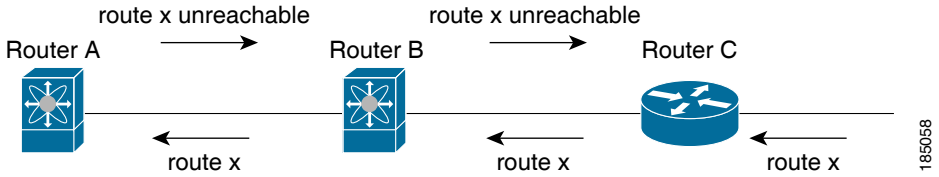
Split Horizon

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は、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。インターフェイス上でスプリット ホライズンがイネーブルの場合、Cisco NX-OS はそのインターフェイスから学習した宛先にはアップデート 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。図 11-1 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 11-1　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、ルータ 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。次に、ルート X をルータ A にアドバタイズしますが、ルータ C には、ルート X 到達不能アップデートを戻します。

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 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。再配布を指定したルート マップを設定して、どのルートが RIP に渡されるかを制御する必要があります。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、第 15 章「Route Policy Manager の設定」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 へのデフォルト ルートを発生させ、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、ルータによって、宛先アドレスから同じ距離にあるすべてのルータ ネットワーク ポートにトラフィックが分散されます。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、RIP ルート テーブルおよびユニキャスト RIB 中の 16 までの等コスト パスを使用する等コスト マルチパス（ECMP）機能をサポートしています。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。

ハイ アベイラビリティ

Cisco NX-OS は、RIP のステートレス リスタートをサポートします。リブートまたはスーパーバイザ スイッチオーバー後に、Cisco NX-OS が実行コンフィギュレーションを適用し、RIP がただちに要求パケットを送信して、ルーティング テーブルに再入力します。

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。

RIP のライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	RIP にライセンスは不要です。ライセンス パッケージに含まれていない機能は nx-os イメージにバンドルされており、無料で提供されます。NX-OS ライセンス方式の詳細については、『*Cisco NX-OS Licensing Guide*』を参照してください。

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 をイネーブルにします（「RIP のイネーブル化」セクション（11-5 ページ）を参照）。

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 は、RIPv1 をサポートしません。RIPv1 パケットを受信した Cisco NX-OS は、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 では IPv6 はサポートされていません。

デフォルト設定

表 11-1 は、各 RIP パラメータに対するデフォルト設定を示します。

表 11-1　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	イネーブル

RIP の設定

この項では、次のトピックについて取り上げます。

- RIP のイネーブル化（11-5 ページ）
- RIP インスタンスの作成（11-6 ページ）
- RIP インスタンスの再起動（11-8 ページ）
- インターフェイス上での RIP の設定（11-8 ページ）
- RIP 認証の設定（11-9 ページ）
- 受動インターフェイスの設定（11-10 ページ）
- 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（11-11 ページ）
- ルート集約の設定（11-11 ページ）
- ルートの再配布の設定（11-11 ページ）
- Cisco IOS RIP との互換性のため、Cisco NX-OS RIP を設定（11-13 ページ）
- 仮想化の設定（11-14 ページ）
- RIP の調整（11-17 ページ）

（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 のイネーブル化

RIP を設定する前に、RIP 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	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 rip** **例:** switch(config)# feature rip	RIP 機能をイネーブルにします。
ステップ 3	**show feature** **例:** switch(config)# show feature	(任意) イネーブルおよびディセーブルにされた機能を表示します。
ステップ 4	**copy running-config startup-config** **例:** switch(config)# copy running-config startup-config	(任意) この設定の変更を保存します。

RIP 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 rip **例:** switch(config)# no feature rip	RIP 機能をディセーブルにし、関連付けられた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、そのインスタンス用のアドレス ファミリを設定できます。

はじめる前に

RIP をイネーブルにします(「RIP のイネーブル化」セクション(11-5 ページ)を参照)。

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	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 rip** *instance-tag*
	
	例:
	`switch(config)# router RIP Enterprise`
	`switch(config-router)#`
		instance tag 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	**show ip rip** [**instance** *instance-tag*] [**vrf** *vrf-name*] **例**: `switch(config-router-af)# show ip rip`	(任意)すべての RIP インスタンスについて、RIP 要約情報を表示します。
ステップ 5	**copy running-config startup-config** **例**: `switch(config-router-af)# copy running-config startup-config`	(任意)この設定の変更を保存します。

RIP インスタンスおよび関連する設定を削除するには、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 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」セクション（1-7 ページ）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。有効な範囲は 1 ～ 64 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。

インターフェイス上での 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。

はじめる前に

RIP をイネーブルにします（「RIP のイネーブル化」セクション（11-5 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ip router rip** *instance-tag*
4. （任意）**show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot/port*] [**vrf** *vrf-name*] [**detail**]
5. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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** *interface-type 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。
ステップ 3	**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 4	**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	(任意) インターフェイスの RIP 情報を表示します。
ステップ 5	**copy running-config startup-config** **例**: switch(config-if)# copy running-config startup-config	(任意) この設定の変更を保存します。

次に、RIP インスタンスに Ethernet 1/2 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)# ip router rip Enterprise
switch(config)# copy running-config startup-config
```

RIP 認証の設定

インターフェイス上で RIP パケットの認証を設定できます。

はじめる前に

RIP をイネーブルにします(「RIP のイネーブル化」セクション(11-5 ページ)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 9000 Series NX-OS Security Configuration Guide』*を参照してください。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ip rip authentication mode**{**text** | **md5**}
4. **ip rip authentication key-chain** *key*
5. (任意)**copy running-config startup-config**

手順の詳細

	コマンド	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** *interface-type 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。
ステップ 3	**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 4	**ip rip authentication key-chain** *key* **例**: switch(config-if)# ip rip authentication key-chain 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 5	**copy running-config startup-config** **例**: switch(config-if)# copy running-config startup-config	(任意) この設定の変更を保存します。

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-keychain)# key 2
switch(config-keychain-key)# accept-lifetime 00:00:00 Jan 01 2000 infinite
switch(config-keychain-key)# send-lifetime 00:00:00 Jan 01 2000 infinite
switch(config-keychain-key)# exit
switch(config-keychain)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# ip rip authentication mode md5
switch(config-if)# ip rip authentication key-chain RIPKey
switch(config-if)# copy running-config startup-config
```

受動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 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 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

ポイズン リバースをイネーブルにすることによって、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。

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 rip summary-address 1.1.1.1/32`	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。

はじめる前に

RIP をイネーブルにします（「RIP のイネーブル化」セクション（11-5 ページ）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 マップ設定の詳細については、「ルート マップの設定」セクション（15-13 ページ）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. **redistribute** {**bgp** *as* | **direct** | **eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-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	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 rip** *instance-tag*
	
	例:
	`switch(config)# router rip Enterprise`
	`switch(config-router)#`
		instance tag 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)#`
		アドレスファミリ コンフィギュレーション モードに入ります。

ステップ 4	**redistribute** {**bgp** *as*	**direct** |{**eigrp**	**isis**	**ospf**	**ospfv3**	**rip**} *instance-tag*	**static**} **route-map** *map-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 マップの詳細については、「ルート マップの設定」セクション（15-13 ページ）を参照してください。
ステップ 5	**default-information originate** [**always**] [**route-map** *map-name*] **例**: `switch(config-router-af)# default-information originate always`	（任意）RIP へのデフォルト ルートを作成し、任意でルート マップで制御します。
ステップ 6	**default-metric** *value* **例**: `switch(config-router-af)# default-metric 2`	（任意）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。

	コマンド	目的
ステップ 7	**show ip rip route** [*ip-prefix* [**longer-prefixes** | **shorter-prefixes**] [**vrf** *vrf-name*] *[summary*] **例**: switch(config-router-af)# show ip rip route	(任意)RIP のルートを表示します。
ステップ 8	**copy running-config startup-config** **例**: switch(config-router-af)# copy running-config startup-config	(任意)この設定の変更を保存します。

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
```

Cisco IOS RIP との互換性のため、Cisco NX-OS RIP を設定

Cisco NX-OS RIP を、ルートがアドバタイズされ、処理される方法で Cisco IOS RIP のように動作するよう設定できます。

直接接続されたルートが、Cisco NX-OS RIP ではコスト 1 として処理され、Cisco IOS RIP ではコスト 0 として処理されます。ルートが Cisco NX-OS RIP でアドバタイズされる場合、受信デバイスはすべての受信ルートに +1 の最小のコストを増加し、ルーティング テーブルにルートをインストールします。Cisco IOS RIP において、このコストの増加は送信側ルータで実行され、受信側ルータは変更なしでルートをインストールします。Cisco NX-OS および Cisco IOS デバイスの両方が連携しているときに、この動作の違いにより問題が発生する可能性があります。Cisco IOS RIP など、ルートをアドバタイズし、処理するために、Cisco NX-OS RIP の設定に応じて、次の互換性の問題を回避できます。

はじめる前に

RIP をイネーブルにします(「RIP のイネーブル化」セクション(11-5 ページ)を参照)。

手順の概要

1. **configure terminal**
2. **router rip** *instance-tag*
3. **[no] metric direct 0**
4. (任意)**show running-config rip**
5. (任意)**copy running-config startup-config**

手順の詳細

	コマンド	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 rip** *instance-tag* **例**: switch(config)# router rip 100 switch(config-router)#	instance tag を設定して、新しい RIP インスタンスを作成します。インスタンス タグには 100、201、または 20 文字までの英数字を入力できます。
ステップ 3	[**no**] **metric direct 0** **例**: switch(config-router)# metric direct 0	ルートがアドバタイズされ、処理される方法で Cisco IOS RIP と Cisco NX-OS RIP が互換性を持つようにするため、直接接続するルータすべてをデフォルトであるコスト 1 の代わりにコスト 0 で設定します。 **注**　このコマンドは、Cisco IOS デバイスを含む RIP ネットワークに存在するすべての Cisco NX-OS デバイスで設定する必要があります。
ステップ 4	**show running-config rip** **例**: switch(config-router)# show running-config rip	(任意) 現在実行中の RIP の設定を表示します。
ステップ 5	**copy running-config startup-config** **例**: switch(config-router)# copy running-config startup-config	(任意) この設定の変更を保存します。

次に、すべての直接ルートをコスト 0 からコスト 1 に返すことによって、Cisco IOS RIP と Cisco NX-OS RIP の互換性をディセーブルにする例を示します。

```
switch# configure terminal
switch(config)# router rip 100
switch(config-router)# no metric direct 0
switch(config-router)# show running-config rip
switch(config-router)# copy running-config startup-config
```

仮想化の設定

複数の RIP インスタンスを設定し、複数の VRF を作成し、同じまたは複数の RIP インスタンスを各 VRF で使用するようにできます。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 を設定したあとに、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。

はじめる前に

RIP をイネーブルにします（「RIP のイネーブル化」セクション（11-5 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **exit**
4. **router rip** *instance-tag*
5. **vrf** *vrf-name*
6. （任意）**address-family ipv4 unicast**
7. （任意）**redistribute** {**bgp** *as* | **direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-name*
8. **interface ethernet** *slot*/*port*
9. **vrf member** *vrf-name*
10. **ip-address** *ip-prefix*/*length*
11. **ip router rip** *instance-tag*
12. （任意）**show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot*/*port*] [**vrf** *vrf-name*]
13. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	**vrf context** *vrf-name* **例**: switch(config)# vrf context RemoteOfficeVRF switch(config-vrf)#	新しい VRF を作成し、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	**exit** **例**: switch(config-vrf)# exit switch(config)#	VRF コンフィギュレーション モードを終了します。
ステップ 4	**router rip** *instance-tag* **例**: switch(config)# router rip Enterprise switch(config-router)#	instance tag 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	目的

ステップ 5	**vrf** *vrf-name*
	
	例:
	`switch(config-router)# vrf RemoteOfficeVRF`
	`switch(config-router-vrf)#`
		新しい VRF を作成します。

ステップ 6	**address-family ipv4 unicast**
	
	例:
	`switch(config-router-vrf)# address-family ipv4 unicast`
	`switch(config-router-vrf-af)#`
		(任意) 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	**redistribute** {**bgp** *as* | **direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-name* **例**: `switch(config-router-vrf-af)# redistribute eigrp 201 route-map RIPmap`	(任意) 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 マップの詳細については、「ルート マップの設定」セクション (15-13 ページ)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	**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 10	**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 11	**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 12	**show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot/port*] [**vrf** *vrf-name*] **例**: `switch(config-if)# show ip rip Enterprise ethernet 1/2`	(任意) VRF のインターフェイスに関する RIP 情報を表示します。
ステップ 13	**copy running-config startup-config** **例**: `switch(config-if)# copy running-config startup-config`	(任意) この設定の変更を保存します。

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)# 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合わせて 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ト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 コンフィギュレーションの確認

RIP 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 ip rip instance [*instance-tag*] [**vrf** *vrf-name*]	RIP インスタンスの状態を表示します。
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 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 ip rip [**instance** *instance-tag*] **policy statistics redistribute** {**bgp** *as* | **direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} [**vrf** *vrf-name*]	RIP ポリシー統計情報を表示します。
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。

ポリシー統計情報を消去するには、**clear ip rip policy statistics redistribute** *protocol process-tag* コマンドを使用します。

RIP 統計情報を消去するには、**clear ip rip statistics** コマンドを使用します。

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 ipv4 unicast
    redistribute eigrp 201 route-map RIPmap
    max-paths 10
!
interface ethernet 1/2
 vrf member NewVRF
 ip address 192.0.2.1/16
 ip router rip Enterprise
 ip rip authentication mode md5
 ip rip authentication key-chain RIPKey
```

関連項目

ルート マップの詳細については、第 15 章「Route Policy Manager の設定」を参照してください。

スタティック ルーティングの設定

この章では、Cisco NX-OS デバイス上でスタティック 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

ルータは、ユーザ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。

この項では、次のトピックについて取り上げます。

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、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）のみを指定する必要があります。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/IPv6 アドレスにする必要があります。

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。

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。

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。

BFD

この機能は、IPv4 の Bidirectional Forwarding Detection（BFD）をサポートします。BFD は、転送パスの障害を高速で検出することを目的にした検出プロトコルです。BFD は 2 台の隣接デバイス間のサブセカンド障害を検出し、BFD の負荷の一部を、サポートされるモジュール上のデータプレーンに分散できるため、プロトコル hello メッセージよりも CPU を使いません。詳細については、*『Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide』*を参照してください。

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/転送（VRF）インスタンスをサポートします。

スタティック ルーティングのライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	スタティック ルーティングにライセンスは不要です。ライセンス パッケージに含まれていない機能は nx-os イメージにバンドルされており、無料で提供されます。Cisco NX-OS のライセンス スキームの詳細については、*『Cisco NX-OS Licensing Guide』*を参照してください。

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 アドレスが到達不能な場合、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。

デフォルト設定値

表 12-1 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 12-1　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	ディセーブル

スタティック ルーティングの設定

この項では、次のトピックについて取り上げます。

- スタティック ルートの設定（12-4 ページ）
- VLAN を介したスタティック ルートの設定（12-6 ページ）
- 仮想化の設定（12-7 ページ）
- スタティック ルーティングの設定確認（12-9 ページ）

（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 ルートの設定

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*]} [**name** *nexthop-name*] [**tag** *tag-value*] [*pref*]

   または

   ipv6 route *ip6-prefix* {*nh-prefix* | *link-local-nh-prefix*} | {*nh-prefix* [*interface*] | *link-local-nh-prefix* [*interface*]} [**name** *nexthop-name*] [**tag** *tag-value*] [*pref*]
3. （任意）**show {ip | ipv6} static-route**
4. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	**ip route** {*ip-prefix* | *ip-addr*/*ip-mask*} {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**name** *nexthop-name*] [**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。サポートされているインターフェイスのリストを表示するには、**?** を使用します。**null 0** を使用すると、ヌル インターフェイスを指定できます。 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 1 です。
	ipv6 route *ip6-prefix* {*nh-prefix* | *link-local-nh-prefix*} | (*nexthop* [*interface*] | *link-local-nexthop* [*interface*]} [**name** *nexthop-name*] [**tag** *tag-value*] [*pref*] **例**: switch(config)# ipv6 route 2001:0DB8::/48 6::6 ethernet 2/1	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。サポートされているインターフェイスのリストを表示するには、**?** を使用します。**null 0** を使用すると、ヌル インターフェイスを指定できます。 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 1 です。
ステップ 3	**show** {**ip** | **ipv6**} **static-route** **例**: switch(config)# show ip static-route	(任意) スタティック ルート情報を表示します。
ステップ 4	**copy running-config startup-config** **例**: switch(config)# copy running-config startup-config	(任意) 実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

次に、ヌル インターフェイスのスタティック ルートを設定する例を示します。

```
switch# configure terminal
switch(config)# ip route 1.1.1.1/32 null 0
switch(config)# copy running-config startup-config
```

スタティック ルートを削除するには、**no** {**ip** | **ipv6**} **route** コマンドを使用します。

VLAN を介したスタティック ルートの設定

VLAN を介したネクスト ホップのサポートなしでスタティック ルートを設定できます。

はじめる前に

アクセス ポートが VLAN の一部であることを確認します。

手順の概要

1. **configure terminal**
2. **feature interface-vlan**
3. **interface vlan** *vlan-id*
4. **ip address** *ip-addr/length*
5. **ip route** *ip-addr/length vlan-id*
6. （任意）**show ip route**
7. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 interface vlan** **例**: `switch(config)# feature interface-vlan`	VLAN インターフェイス モードをイネーブルにします。
ステップ 3	**interface-vlan** *vlan-id* **例**: `switch(config)# interface-vlan 10`	SVI を作成して、インターフェイス コンフィギュレーション モードを開始します。 *vlan-id* 引数の範囲は 1 ～ 4094 ですが、内部スイッチ用に予約されている VLAN は除きます。
ステップ 4	**ip address** *ip-addr/length* **例**: `switch(config)# ip address 192.0.2.1/8`	VLAN の IP アドレスを設定します。
ステップ 5	**ip route** *ip-addr/length vlan-id* **例**: `switch(config)# ip route 209.165.200.224/27 vlan 10`	スイッチ仮想インターフェイス（SVI）上のネクスト ホップなしでインターフェイスのスタティック ルートを追加します。 IP アドレスは、スイッチに接続されたインターフェイスで設定されるアドレスです。

	コマンド	目的
ステップ 6	**show ip route** **例**: switch(config)# show ip route	(任意)Unicast Route Information Base(URIB)からルートを表示します。
ステップ 7	**copy running-config startup-config** **例**: switch(config)# copy running-config startup-config	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

次に、SVI を介したネクスト ホップなしでスタティック ルートを設定する例を示します。

```
switch# configure terminal
switch(config)# feature interface-vlan
swicth(config)# interface vlan 10
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# ip route 209.165.200.224/27 vlan 10 <===209,165.200.224 is the IP
address of the interface that is configured on the interface that is directly connected to
the switch.
switch(config-if)# copy running-config startup-config
```

スタティック ルートを削除するには、**no ip route** コマンドを使用します。

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*} [**name** *nexthop-name*] [**tag** *tag-value*] [*pref*]

   または

   ipv6 route *ip6-prefix* {*nh-prefix* | *link-local-nh-prefix*} | {*next-hop* [*interface*] | *link-local-next-hop* [*interface*]} [**name** *nexthop-name*] [**tag** *tag-value*] [*pref*]
4. (任意)**show {ip | ipv6} static-route vrf** ***vrf-name***
5. (任意)**copy running-config startup-config**

手順の詳細

	コマンド	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	**vrf context** *vrf-name* **例**: switch(config)# vrf context StaticVrf	VRF を作成し、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** {*ip-prefix* | *ip-addr ip-mask*} {*next-hop* | *nh-prefix* | *interface*} [**name** *nexthop-name*] [**tag** *tag-value*] [*pref*] **例**: switch(config-vrf)# ip route 192.0.2.0/8 ethernet 1/2	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。サポートされているインターフェイスのリストを表示するには、**?** を使用します。**null 0** を使用すると、ヌル インターフェイスを指定できます。 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 1 です。
	ipv6 route *ip6-prefix* {*nh-prefix* | *link-local-nh-prefix*} | (*nexthop* [*interface*] | *link-local-nexthop* [*interface*]} [**name** *nexthop-name*] [**tag** *tag-value*] [*pref*] **例**: switch(config)# ipv6 route 2001:0DB8::/48 6::6 ethernet 2/1	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。サポートされているインターフェイスのリストを表示するには、**?** を使用します。**null 0** を使用すると、ヌル インターフェイスを指定できます。 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 1 です。
ステップ 4	**show** {**ip** | **ipv6**} **static-route vrf** *vrf-name* **例**: switch(config-vrf)# show ip static-route	(任意) スタティック ルート情報を表示します。
ステップ 5	**copy running-config startup-config** **例**: switch(config-vrf)# copy running-config startup-config	(任意) 実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

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** | **ipv6**} **static-route**	設定されているスタティック ルートを表示します。
show ipv6 static-route vrf *vrf-name*	各 VRF のスタティック ルートの情報を表示します。
show {**ip** | **ipv6**} **static-route track-table**	IPv4 または IPv6 スタティック ルート トラック テーブルに関する情報を表示します。

スタティック ルーティングの設定例

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
```

レイヤ 3 仮想化の設定

この章では、Cisco NX-OS デバイスでレイヤ 3 仮想化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。

レイヤ 3 仮想化について

Cisco NX-OS は、複数の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 および IPv6 に対応するユニキャストおよびマルチキャスト 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。

管理 VRF

- 管理 VRF は管理専用です。
- mgmt 0 インターフェイスのみが、管理 VRF にいることができます。
- mgmt 0 インターフェイスは、異なる VRF に割り当てられることはできません。
- ルーティング プロトコルは、管理 VRF（スタティックのみ）で動作できません。

デフォルト VRF

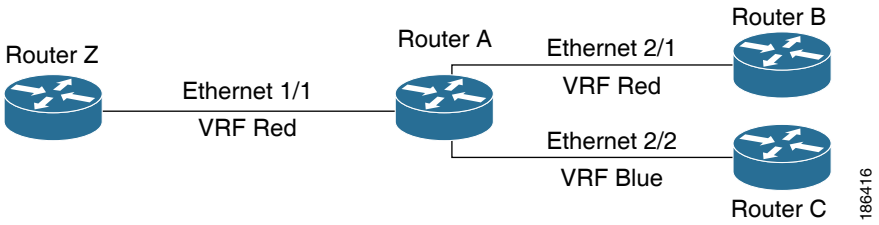
- すべてのレイヤ 3 インターフェイスは、別の VRF に割り当てられるまでデフォルト VRF に存在します。
- 異なる VRF コンテキストが指定されない限り、ルーティング プロトコルはデフォルトの VRF コンテキストで実行されます。
- デフォルト VRF は、すべての **show** コマンドにデフォルトのルーティング コンテキストを使用します。
- デフォルト VRF は、Cisco IOS のグローバル ルーティング テーブルの概念に似ています。

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。図 13-1 に、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更新を共有します。ルータ C は別の VRF で設定されているからです。

図 13-1　　ネットワーク内の VRF



Cisco NX-OS はデフォルトで、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 ルート リーク

Cisco NX-OS は、VRF 間のルート リークをサポートしています。

インポート ポリシーを使用して IP プレフィックスをグローバル ルーティング テーブル（デフォルト VRF）から他の VRF にインポートしたり、エクスポート ポリシーを使用して IP プレフィックスをデフォルト以外の VRF からデフォルトの VRF にエクスポートしたりできます。VRF インポート ポリシーおよびエクスポート ポリシーは、ルート マップを使用して VRF にインポートまたはエクスポートされるプレフィックスを指定します。これらのポリシーは、IPv4 および IPv6 ユニキャスト プレフィックスをインポートまたはエクスポートできます。

（注）　BGP のデフォルト VRF のルートは直接インポートできます。デフォルト VRF の他のルートは、最初に BGP に再配布する必要があります。

IP プレフィックスは、標準のルート ポリシー フィルタリング メカニズムでインポートまたはエクスポート ルート マップの一致基準として定義されます。たとえば、IP プレフィックス リストまたは AS パス フィルタを作成して IP プレフィックスまたは IPプ レフィックスの範囲を定義し、そのプレフィックス リストまたは AS パス フィルタをルート マップの match 文節で使用できます。ルート マップを通過したプレフィックスは、インポートまたはエクスポート ポリシーを使用して指定された VRF にインポートまたはエクスポートされます。別の VRF からインポートされるルートまたはパスは、再びインポートまたはエクスポートすることはできません。

詳細については、「VRF ルート リークの注意事項と制約事項」セクション（13-6 ページ）を参照してください。

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って、リモート サーバに接続したり、選択した VRF に基づいて情報をフィルタリングすることができます。

- AAA：詳細については、『*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*』を参照してください。
- Call Home：詳細については、『*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*』を参照してください。
- DNS（ドメイン ネーム システム）：詳細については、第 4 章「DNS の設定」を参照してください。
- HTTP：詳細については、『*Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*』を参照してください。
- HSRP：詳細については、第 17 章「HSRP の設定」、を参照してください。
- NTP：詳細については、『*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*』を参照してください。
- RADIUS：詳細については、『*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*』を参照してください。
- ping および traceroute：詳細については、『*Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*』を参照してください。
- SSH：詳細については、『*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*』を参照してください。
- SNMP：詳細については、『*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*』を参照してください。
- Syslog：詳細については、『*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*』を参照してください。
- TACACS+：詳細については、『*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*』を参照してください。
- HTTP：詳細については、『*Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*』を参照してください。
- VRRP（仮想ルータ冗長プロトコル）：詳細については、第 18 章「VRRP の設定」、を参照してください。
- XML：詳細については、『*Cisco NX-OS XML Management Interface User Guide, Release 5.x*』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詳細については、各サービスの適切なコンフィギュレーション ガイドを参照してください。

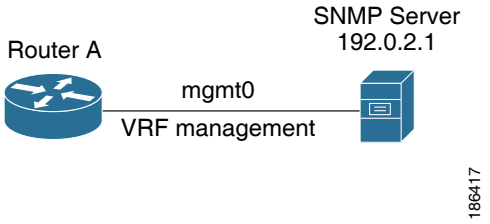
ここでは、次の内容について説明します。

到達可能性

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。ルータ上でサーバ 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 が使用しなければならない VRF も設定します。

図 13-2 に、管理 VRF を介して到達できる SNMP サーバを示します。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。

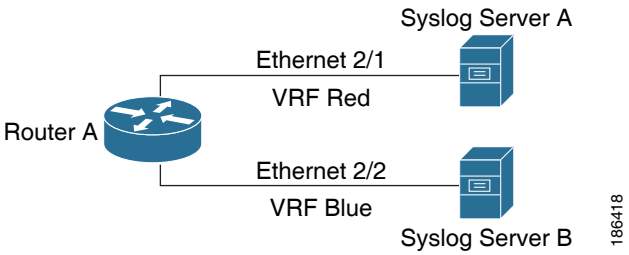
図 13-2　サービス VRF の到達可能性



フィルタリング

フィルタリングによって、VRF に基づいて VRF 認識サービスに渡す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。図 13-3 は、それぞれが 1 つ VRF をサポートしている 2 つの syslog サーバ 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 13-3　サービス VRF のフィルタリング

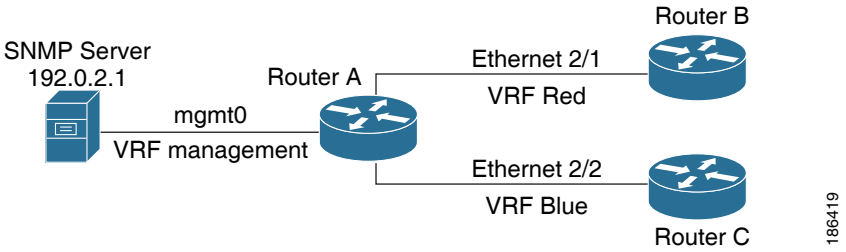


到達可能性とフィルタリングの組み合わせ

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。その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、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。

図 13-4 に、管理 VRF 上で到達できる SNMP サーバを示します。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 13-4　サービス VRF の到達可能性とフィルタリング



VRF のライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	VRF にライセンスは不要です。ライセンス パッケージに含まれていない機能は nx-os イメージにバンドルされており、無料で提供されます。Cisco NX-OS のライセンス スキームの詳細については、*『Cisco NX-OS Licensing Guide』*を参照してください。 VRF ルート リークには Enterprise Services ライセンスが必要です。Cisco NX-OS ライセンス方式の詳細と、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。

VRF の注意事項および制約事項

VRF 設定時の注意事項と制約事項は次のとおりです。

- 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 VRF および管理 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。**write erase** コマンドを使用してから **write erase boot** コマンドを使用する必要があります。

VRF ルート リークの注意事項と制約事項

VRF ルート リークの設定時の注意事項と制限は次のとおりです。

- ルート リークは、2 つのデフォルト以外の VRF 間、およびデフォルト VRF から デフォルト以外の VRF でサポートされます。Cisco NX-OS Release 7.0(3)I2(1) 以降では、デフォルト以外の VRF からデフォルト VRF へのルート リークもサポートされています。
- 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、デフォルト VRF からデフォルト以外の VRF にインポートできる IP プレフィクスの最大数は 1,000 ルートです。
- 2 つのデフォルト以外の VRF 間でリークできるルートの数に制限はありません。
- 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。

デフォルト設定値

表 13-1 に、VRF パラメータのデフォルト設定を示します。

表 13-1　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 ルート リークのプレフィクスの制限	1000

VRF の設定

ここでは、次の内容について説明します。

- VRF の作成（13-7 ページ）
- インターフェイスへの VRF メンバーシップの割り当て（13-8 ページ）
- ルーティング プロトコルに関する VRF パラメータの設定（13-9 ページ）
- グローバル VRF ルート リークの設定（13-11 ページ）
- VRF 認識サービスの設定（13-12 ページ）
- VRF スコープの設定（13-13 ページ）

（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

VRF を作成できます。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. （任意）**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	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	**vrf context** *name*
	
	例:
	`switch(config)# vrf context Enterprise`
	`switch(config-vrf)#`
		新しい VRF を作成し、VRF コンフィギュレーション 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`	（任意）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 1 です。
ステップ 4	**show vrf** [*vrf-name*] **例**: `switch(config-vrf)# show vrf Enterprise`	（任意）VRF 情報を表示します。
ステップ 5	**copy running-config startup-config** **例**: `switch(config)# copy running-config startup-config`	（任意）この設定の変更を保存します。

VRF および関連する設定を削除するには、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 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 コンフィギュレーション モードで使用できるコマンドはすべて、VRF コンフィギュレーション 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り当て

インターフェイスを VRF のメンバにできます。

はじめる前に

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. （任意）**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	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** *interface-type 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。

	コマンド	目的
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`	（任意）VRF 情報を表示します。
ステップ 6	**copy running-config startup-config** **例**: `switch(config)# copy running-config startup-config`	（任意）この設定の変更を保存します。

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 プロトコルに関する 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-type slot*/*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	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** *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 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	**maximum-paths** *paths* 例: switch(config-router-vrf)# maximum-paths 4	(任意) 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-type 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。
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。
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	**copy running-config startup-config** **例:** switch(config)# copy running-config startup-config	(任意) この設定の変更を保存します。

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
```

グローバル VRF ルート リークの設定

デフォルト VRF からデフォルト以外の VRF へ、またはデフォルト以外の VRF からデフォルト VRFへのルート リークを設定できます。

デフォルト以外の VRF 間のルート リークは(ルート ターゲット一致で)自動的に有効になります。

はじめる前に

Enterprise ライセンスがインストールされており、BGP が有効になっていることを確認してください。

手順の概要

1. **configure terminal**
2. **vrf context** [*vrf-name*]
3. **address-family** {**ipv4** | **ipv6**} **unicast**
4. {**import** | **export**} **vrf default** [*prefix-limit*] **map** *route-map*
5. （任意）**show bgp process vrf** [*vrf-name*]
6. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	`vrf context` *vrf-name*
	
	例:
	`switch(config)# vrf context vpn1`
	`switch(config-vrf)#`
		新しい VRF を作成します。

	コマンド	目的
ステップ 3	**address-family** {**ipv4** | **ipv6**} **unicast** **例:** switch(config-vrf)# address-family ipv4 unicast switch(config-vrf-af-ipv4)#	IPv4 または IPv6 アドレス ファミリに対してグローバル 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 4	{**import** | **export**} **vrf default** [*prefix-limit*] **map** *route-map* 例: switch(config-vrf-af-ipv4)# import vrf default map importmap 例: switch(config-vrf-af-ipv4)# export vrf default map exportmap	VRF ルート リークを設定します。次のオプションを使用できます。 • **import**:IPv4 または IPv6 ユニキャスト プレフィックスを含むルートを、グローバル ルーティング テーブル(デフォルト VRF)から他の VRF にコピーします。 • **export**:IPv4 または IPv6 ユニキャスト プレフィックスを含むルートを、デフォルト以外の VRF からグローバル ルーティング テーブル(デフォルト VRF)にコピーします。 • *prefix-limit*:インポートまたはエクスポートできるルートの最大数を指定します。範囲は 1 ～ 2147483647 で、デフォルト値は 1000 です。
ステップ 5	**show bgp process vrf** [*vrf-name*] **例:** switch(config-vrf-af-ipv4)# show bgp process vrf vpn1	(任意)指定した VRF の BGP プロセス情報を表示します。
ステップ 6	**copy running-config startup-config** **例:** switch(config)# copy running-config startup-config	(任意)この設定の変更を保存します。

VRF 認識サービスの設定

VRF 認識サービスの到達可能性および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」セクション(13-3 ページ)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	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	**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。選択された 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、さらに任意で、指定されたドメイン名に接続するために Cisco NX-OS が使用する VRF を設定します。
ステップ 5	**copy running-config startup-config** **例**: switch(config)# copy running-config startup-config	(任意) この設定の変更を保存します。

次に、VRF Red で到達可能な SNMP ホスト 192.0.2.1 に、すべての VRF の SNMP 情報を送信する例を示します。

```
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 context 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 コンフィギュレーションの確認

VRF 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 bgp process vrf [*vrf-name*]	指定した VRF の BGP プロセス情報を表示します。
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、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。

```
configure terminal
vrf context Red
 snmp-server host 192.0.2.12 use-vrf Red
router ospf 201
   VRF Red
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。

```
configure terminal
!Create the VRFs
vrf context Red
vrf context Blue
vrf context Green
!Create the OSPF instances and associate them with a single VRF or multiple VRFs
(recommended)
feature ospf
router ospf Lab
 VRF Red
!
router ospf Production
 vrf Blue
   router-id 1.1.1.1
vrf Green
   router-id 2.2.2.2
!Configure one interface to use ospf Lab on VRF Red
interface ethernet 1/2
 vrf member Red
 ip address 192.0.2.1/16
 ip router ospf Lab area 0
 no shutdown
!Configure another interface to use ospf Production on VRF Blue
interface ethernet 10/2
 vrf member Blue
 ip address 192.0.2.1/16
 ip router ospf Production area 0
 no shutdown
!
interface ethernet 10/3
 vrf member Green
 ip address 192.0.2.1/16
 ip router ospf Production area 0
 no shutdown

!configure the SNMP server
snmp-server user admin network-admin auth md5 nbv-12345
snmp-server community public ro
!Create the SNMP contexts for each VRF
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 this example.
```

この例で、VRF Red の OSPF インスタンス Lab の OSPF-MIB 値にアクセスするには、SNMP コンテキスト **lab** を使用します。

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

switch# 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], 0:26:46, direct
35.35.1.2/32, ubest/mbest: 1/0, attached
*via 35.35.1.2, Eth1/7, [0/0], 0:26:46, 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], 0:25:53, direct
45.45.1.2/32, ubest/mbest: 1/0, attached
*via 45.45.1.2, Eth1/11, [0/0], 0:25:53, 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
```

次に、デフォルト以外の VRF からデフォルト VRF へのルート リークを設定する例を示します。

```
feature bgp
vrf context vpn1
    address-family ipv4 unicast
        import vrf default map importmap
        export vrf default map exportmap

show bgp ipv4 unicast 123.123.123.123/32
BGP routing table information for VRF default, address family IPv4 Unicast
BGP routing table entry for 123.123.123.123/32, version 6
Paths: (1 available, best #1)
Flags: (0x8008001a) on xmit-list, is in urib, is best urib route

  Advertised path-id 1
  Path type: redist, path is valid, is best path
             Imported from 100:1:123.123.123.123/32 (VRF vpn1)
  AS-Path: NONE, path locally originated
    0.0.0.0 (metric 0) from 0.0.0.0 (1.1.1.1)
      Origin incomplete, MED 0, localpref 100, weight 32768
Extcommunity:  RT:100:1

  Path-id 1 not advertised to any peer
  Path-id 1 scheduled to be advertised to peers:
    2.2.2.2

show bgp process vrf vpn1
Information regarding configured VRFs:
```

```
BGP Information for VRF vpn1
VRF Id                           : 3
VRF state                        : UP
Router-ID                        : 20.0.0.1
Configured Router-ID             : 0.0.0.0
Confed-ID                        : 0
Cluster-ID                       : 0.0.0.0
No. of configured peers          : 2
No. of pending config peers      : 0
No. of established peers         : 2
VRF RD                           : 100:1

    Information for address family IPv4 Unicast in VRF vpn1
    Table Id                     : 3
    Table state                  : UP
Peers      Active-peers    Routes     Paths      Networks    Aggregates
    1          1               6          6          0           0

    再分配
        なし

    Export RT list:
        100:1
        1000:1
    Import RT list:
        100:1
    Label mode: per-prefix
    Aggregate label: 492287
    Import default limit         : 1000
    Import default prefix count  : 2
    Import default map           : importmap
    Export default limit         : 1000
    Export default prefix count  : 3
    Export default map           : exportmap
```

その他の関連資料

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（13-18 ページ）
- 標準（13-19 ページ）

関連資料

関連項目	マニュアル タイトル
BGP	*第 9 章「ベーシック BGP の設定」*
VRF	*『Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide』* *『Cisco Nexus 9000 Series NX-OS System Management Configuration Guide』*

標準

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

ユニキャスト RIB および FIB の管理

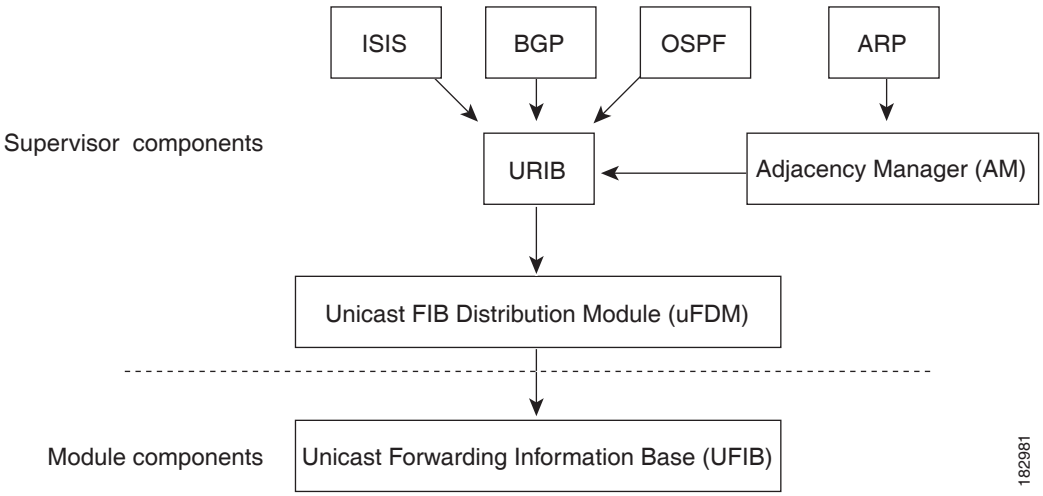
この章では、Cisco NX-OS デバイスのユニキャスト ルーティング情報ベース（RIB）および転送情報ベース（FIB）のルートを管理する方法について説明します。

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

ユニキャスト RIB（IPv4 RIB と IPv6 RIB）および FIB は、図 14-1 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 14-1　Cisco NX-OS 転送アーキテクチャ



ユニキャスト RIB は、アクティブなスーパーバイザ上にあります。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（UFDM）のサービスを使用して、モジュール上のユニキャスト 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）。

この項では、次のトピックについて取り上げます。

- レイヤ 3 整合性チェッカー（14-2 ページ）

レイヤ 3 整合性チェッカー

まれな状況において、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、スーパーバイザ モジュールのユニキャスト IPv4 RIB と各インターフェイス モジュールの FIB の間で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」セクション（14-8 ページ）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」セクション（14-8 ページ）を参照してください。

ユニキャスト RIB および FIB のライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	ユニキャスト RIB および FIB にライセンスは不要です。ライセンス パッケージに含まれていない機能は nx-os イメージにバンドルされており、無料で提供されます。Cisco NX-OS のライセンス スキームの詳細については、『*Cisco NX-OS Licensing Guide*』を参照してください。

ユニキャスト RIB および FIB の管理

この項では、次のトピックについて取り上げます。

（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。

モジュールの FIB 情報の表示

モジュールの FIB 情報を表示できます。

手順の詳細

モジュールの 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 forwarding {**ipv4** | **ipv6**} **adjacency module** *slot* **例:** `switch# show forwarding ipv6 adjacency module 2`	IPv4 または IPv6 の隣接情報を表示します。
show forwarding {**ipv4** | **ipv6**} **route module** *slot* **例:** `switch# show forwarding ipv6 route module 2`	IPv4 または IPv6 のルート テーブルを表示します。

ユニキャスト FIB のロード シェアリングの設定

Open Shortest Path First（OSPF）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 ルーティング情報ベース（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 セットを転送情報ベース（FIB）に組み込み、転送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。

（注）　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*][**rotate** *rotate*] [**concatenation**] **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** オプションは、ハッシュ アルゴリズムのランダム シードを設定し、フローをあるリンクから別のリンクにシフトします。 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。*universal-id* の範囲は 1 ～ 4294967295 です。 • **rotate** オプションを使用すると、ハッシュ アルゴリズムはネットワーク内のすべてのノードで同じリンクを継続的に選択しないように、リンク ピッキング選択を循環させます。これは、ハッシュ アルゴリズムのビット パターンに影響を与えることによって機能します。このオプションは、あるリンクから別のリンクにフローをシフトし、最初の ECMP レベルからすでにロード バランシング（極性化）されているトラフィックのロード バランシングを複数のリンク間で行います。 *rotate* 値を指定すると、64 ビットのストリームが循環回転でそのビット位置から解釈されます。*rotate* 値の範囲は 1 ～ 63 で、デフォルトは 32 です。 **注**　多層レイヤ 3 トポロジでは、極性が発生する可能性があります。極性を回避するには、トポロジの各層で異なる循環ビットを使用します。 **注**　ポート チャネルの rotation 値を設定するには、**port-channel load-balance src-dst ip-l4port rotate** *rotate* コマンドを使用します。このコマンドの詳細については、『*Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*』を参照してください。 • **concatenation** オプションを使用すると、ECMP のハッシュ タグ値とポート チャネルのハッシュ タグ値が一つに結合され、より強力な 64 ビットのハッシュを使用できるようになります。このオプションを使用しない場合、ECMP のロード バランシングおよびポート チャネルのロード バランシングを個別に制御できます。デフォルトではディセーブルになっています。

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 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:** `switch# 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、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 10.0.0.5 192.0.0.2
  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 *172.0.0.2 (hash: 0x0e), for route:
```

ルーティング情報と隣接情報の表示

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** | **ipv6**} **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、direct、static、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。

コマンド	目的
show {**ip** | **ipv6**} **adjacency** [*prefix* | *interface-type 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 または IPv6 プレフィックス アドレス。 • *interface-type number*:**?** 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。 • *vrf-id*:最大 64 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** | **ipv6**} **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、direct、static、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 Context "default"
'*' denotes best ucast next-hop       '**' denotes best mcast next-hop
'[x/y]' denotes [preference/metric]

0.0.0.0/0, 1 ucast next-hops, 0 mcast next-hops
   *via 10.1.1.1, mgmt0, [1/0], 5d21h, static
0.0.0.0/32, 1 ucast next-hops, 0 mcast next-hops
   *via Null0, [220/0], 1w6d, local, discard
10.1.0.0/22, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.1.55, mgmt0, [0/0], 5d21h, direct
10.1.0.0/32, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.0.0, Null0, [0/0], 5d21h, local
10.1.1.1/32, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.1.1, mgmt0, [2/0], 5d16h, am
10.1.1.55/32, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.1.55, mgmt0, [0/0], 5d21h, local
10.1.1.253/32, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.1.253, mgmt0, [2/0], 5d20h, am
10.1.3.255/32, 1 ucast next-hops, 0 mcast next-hops, attached
   *via 10.1.3.255, mgmt0, [0/0], 5d21h, local
255.255.255.255/32, 1 ucast next-hops, 0 mcast next-hops
   *via Eth Inband Port, [0/0], 1w6d, local
```

次に、隣接関係情報を表示する例を示します。

```
switch# show ip adjacency

IP Adjacency Table for context 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 forwarding [**ipv4** | **ipv6**] [**unicast**] **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 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 ～ 26 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** | **ipv6**] [**unicast**] **inconsistency** [**vrf** *vrf-name*] [**module** {*slot*| **all**}] **stop** **例:** `switch# 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 ～ 26 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** | **ipv6**] **inconsistency** [**vrf** *vrf-name*] [**module** {*slot*| **all**}] **例:** `switch# 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 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 ～ 26 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。FIB のエントリを消去しても、ユニキャスト RIB に影響はありません。

注意　**clear forwarding** コマンドを実行すると、デバイス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** | **ipv6**} **route** {* | *prefix*} [**vrf** *vrf-name*] **module** {*slot*| **all**} **例**: `switch# clear forwarding ipv4 route * module 1`	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 または IPv6 プレフィックス *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 ～ 26 です。

ユニキャスト RIB の最大ルート数の設定

ルーティング テーブルで許可されている最大ルート数を設定できます。

手順の概要

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **ipv4 unicast**
4. **maximum routes** *max-routes* [*threshold* [**reinstall** *threshold*] | **warning-only**]
5. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	**vrf context** *vrf-name*
	
	例:
	`switch(config)# vrf context Red`
	`switch(config-vrf)#`
		VRF を作成し、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	**ipv4 unicast**
	
	例:
	`switch(config-vrf)# ipv4 unicast`
	`switch(config-vrf-af-ipv4)#`
		アドレスファミリ コンフィギュレーション モードに入ります。

	コマンド	目的
ステップ 4	**maximum routes** *max-routes* [*threshold* [**reinstall** *threshold*] | **warning-only**] **例:** `switch(config-vrf-af-ipv4)# maximum routes 250 90`	ルーティング テーブルで許可される最大ルート数を設定します。範囲は 1 ～ 4294967295 です。 次の項目を任意で指定できます。 • *threshold*:警告メッセージをトリガーする最大ルート数のパーセンテージ。範囲は 1 ～ 100 です。 • **warning-only**:ルートの最大数を超えたときの警告メッセージを記録します。 • **reinstall** *threshold*:以前に最大ルート数の制限を超過し、拒否されたルートを再インストールして、それらを再インストールするしきい値を指定します。しきい値の範囲は 1 ～ 100 です。
ステップ 5	**copy running-config startup-config** **例:** `switch(config-vrf-af-ipv4)# copy running-config startup-config`	(任意) この設定の変更を保存します。

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 {**ipv6**} **memory estimate routes** *num-routes* **next-hops** *num-nexthops* **例:** `switch# show routing memory estimate routes 5000 next-hops 2`	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　* キーワードはルーティングに破壊的な影響を与えます。

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 {**ip** | **ipv4** | **ipv6**} **route** {* | {*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 または IPv6 ルート • *prefix/length*:任意の IP または IPv6 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 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 routing [**multicast** | **unicast**] [**ip** | **ipv4** | **ipv6**] {* | {*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 または IPv6 ルート • *prefix/length*:任意の IP または IPv6 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 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。

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 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。
show forwarding {**ip** | **ipv4** | **ipv6**} **route**	FIB 内のルートを表示します。
show {**ip** | **ipv6**} **adjacency**	隣接関係テーブルを表示します。
show {**ip** | **ipv6**} **route**	ユニキャスト RIB から受け取った IPv4 または IPv6 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。

その他の関連資料

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（14-12 ページ）

関連資料

関連項目	マニュアル タイトル
EEM の設定	*『Cisco Nexus 9000 Series NX-OS System Management Configuration Guide』*

Route Policy Manager の設定

この章では、Cisco NX-OS デバイスでの Route Policy Manager 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。

Route Policy Manager について

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 または IPv6 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。

この項では、次のトピックについて取り上げます。

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。シーケンス番号がユーザにより設定されていない場合、Cisco NX-OS によりシーケンス番号が自動設定されます。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 は指定されたプレフィック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、残りのプレフィックス リストは評価しません。

（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。

プレフィックス リストのマスク

Cisco NX-OS Release 7.0(3)I4(1) では、プレフィックス リストのマスクが導入されています。マスクでは、数値 1 と数値 0 を使用して、対応する IP アドレス ビットをどのように扱うかを指定します。

- マスク ビット 0 は、対応するビット値を無視することを示します。
- マスク ビット 1 は、対応するビット値が正確に一致しているかどうかを確認することを示します。

プレフィックス リストを使用してルート マップの IP アドレスを照合できます。この IP アドレスは再配布時にルーティング プロトコルで使用されます。IP アドレスは、マスク ビット 1 に対応するビットがプレフィックス リストで指定されたサブネットと同じであるプレフィックス リストと照合されます。

マスクを慎重に設定することにより、許可または拒否のテストに 1 つまたは複数の IP アドレスを選択できます。

プレフィックス リストのマスクを使用すると、マスクに非連続ビットを指定し、偶数または奇数の IP アドレスの範囲を定義できます。

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 マップのルートまたは 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 エントリと一致すると、設定した 1 つまたは複数の 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 または IPv6 アドレス
- Protocol
- Precedence
- ToS

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 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（最初の 2 バイトは 2 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。

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）。

（注） 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 のライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	Route Policy Manager にライセンスは不要です。ライセンス パッケージに含まれていない機能は nx-os イメージにバンドルされており、無料で提供されます。NX-OS ライセンス方式の詳細については、『*Cisco NX-OS Licensing Guide*』を参照してください。

ガイドラインと制限事項

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。

- ルート マップが空の場合は、すべてのルートが拒否されます。
- プレフィックス リストが空の場合は、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、Cisco NX-OS は match 文を失敗として、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。
- シスコは、同じルート マップ シーケンスで IPv4 と IPv6 の両方の match 文を使用し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してください。
- 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。

デフォルト設定値

表 15-1 に、Route Policy Manager のデフォルト設定を示します。

表 15-1　Route Policy Manager のデフォルト パラメータ

パラメータ	デフォルト
Route Policy Manager	イネーブル
アドミニストレーティブ ディスタンス	115

Route Policy Manager の設定

この項では、次のトピックについて取り上げます。

- IP プレフィックス リストの設定（15-7 ページ）
- AS パス リストの設定（15-9 ページ）
- コミュニティ リストの設定（15-10 ページ）
- 拡張コミュニティ リストの設定（15-11 ページ）
- ルート マップの設定（15-13 ページ）

（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 には IP プレフィックス リスト、IPv6 には IPv6 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する場合、および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）以上で **lt** キーワードの値（設定されている場合）以下の場合です。

プレフィックス アドレスとの比較に使用できる連続または非連続ルートの範囲を定義するには、**mask** キーワードを使用します。

手順の概要

1. **configure terminal**
2. {**ip** | **ipv6**} **prefix-list** *name* **description** *string*
3. **ip prefix-list** *name* [**seq** *number*] [{**permit** | **deny**} *prefix* {[**eq** *prefix-length*] | [**ge** *prefix-length*] [**le** *prefix-length*]}] [**mask** *mask*]

   または

   ipv6 prefix-list *name* [**seq** *number*] [{**permit** | **deny**} *prefix* **{[eq** *prefix-length*] | [**ge** *prefix-length*] [**le** *prefix-length*]}] [**mask** *mask*]
4. （任意）**show** {**ip** | **ipv6**} **prefix-list** *name*
5. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	{**ip** | **ipv6**} **prefix-list** *name* **description** *string* **例**: switch(config)# ip prefix-list AllowPrefix description allows engineering server	（任意）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。

	コマンド	目的

ステップ 3	**ip prefix-list** *name* [**seq** *number*] [{**permit** | **deny**} *prefix* {[**eq** *prefix-length*] | [**ge** *prefix-length*] [**le** *prefix-length*]}] [**mask** *mask*]
	
	例:
	`switch(config)# ip prefix-list AllowPrefix seq 10 permit 192.0.2.0/24 eq 24`
	
	例:
	`switch(config)# ip prefix-list even permit 0.0.0.0/32 mask 0.0.0.1`
		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。
		• 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。
		• mask:再配布時にルーティング プロトコルで使用されるプレフィックス アドレスのビットと比較する、プレフィックス リストのプレフィックス アドレスのビットを指定します。

	ipv6 prefix-list *name* [**seq** *number*] [{**permit** | **deny**} *prefix* {[**eq** *prefix-length*] | [**ge** *prefix-length*] [**le** *prefix-length*]}] [**mask** *mask*] **例:** `switch(config)# ipv6 prefix-list AllowIPv6Prefix seq 10 permit 2001:0DB8:: le 32`	IPv6 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設定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。 • 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。 • mask:再配布時にルーティング プロトコルで使用されるプレフィックス アドレスのビットと比較する、プレフィックス リストのプレフィックス アドレスのビットを指定します。
ステップ 4	**show** {**ip** | **ipv6**} **prefix-list** *name* **例**: `switch(config)# show ip prefix-list AllowPrefix`	(任意)プレフィックス リスト情報を表示します。
ステップ 5	**copy running-config startup-config** **例:** `switch# copy running-config startup-config`	(任意)この設定の変更を保存します。

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 192.0.2.0/24 eq 24
switch(config)# ip prefix-list allowprefix seq 20 permit 209.165.201.0/27 eq 27
switch(config)# router bgp 65535
switch(config-router)# neighbor 192.0.2.1/16 remote-as 65534
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# prefix-list allowprefix in
```

次に、すべての /24 奇数 IP アドレスに一致するマスクを使用して IPv4 プレフィックス リストを作成する例を示します。

```
switch# configure terminal
switch(config)# ip prefix-list list1 seq 7 permit 22.1.1.0/24 mask 255.255.1.0
switch(config)# show route-map test
route-map test, permit, sequence 7
  Match clauses:
    ip address prefix-lists: list1
  Set clauses:
    extcommunity COST:igp:10:20
switch(config)# show ip prefix-list list1
ip prefix-list list1: 1 entries
   seq 7 permit 22.1.1.0/24 mask 255.255.1.0
```

次に、サブネット プレフィックスが 17 以上で、21.1.0.0/16 のすべてのサブネットに一致する IPv4 プレフィックス リストを作成する例を示します。**mask** オプションにより、第 3 オクテットの最初のビットが未設定（偶数）の着信プレフィックスのみが一致します。

```
switch# configure terminal
switch(config)# ip prefix-list list1 seq 10 permit 21.1.0.0/16 ge 17 mask 255.255.1.0
```

AS パス リストの設定

発信および着信 BGP ルートの両方に、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** | **ipv6**} **as-path list** *name*
4. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	**ip as-path access-list** *name* {**deny** | **permit**} *expression* **例**: `switch(config)# ip as-path access-list Allow40 permit 40`	正規表現を使用して BGP AS パス リストを作成します。

	コマンド	目的
ステップ 3	**show** {**ip** | **ipv6**} **as-path-access-list** *name* **例**: switch(config)# show ip as-path-access-list Allow40	(任意)AS パス アクセス リスト情報を表示します。
ステップ 4	**copy running-config startup-config** **例**: switch# copy running-config startup-config	(任意)この設定の変更を保存します。

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 65535:20
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
```

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. **ip community-list standard** *list-name* {**deny** | **permit**} [*community-list*] [**internet**] [**local-AS**] [**no-advertise**] [**no-export**]

   または

   ip community-list expanded *list-name* {**deny** | **permit**} *expression*
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	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	**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 65535:20	標準 BGP コミュニティ リストを作成します。*list-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。*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	**show ip community-list** *name* **例**: switch(config)# show ip community-list BGPCommunity	(任意)コミュニティ リストの情報を表示します。
ステップ 4	**copy running-config startup-config** **例**: switch# copy running-config startup-config	(任意)この設定の変更を保存します。

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 65535:20
switch(config)# ip community-list standard BGPCommunity permit local-AS no-export
switch(config)# copy running-config startup-config
```

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. **ip extcommunity-list standard** *list-name* {**deny** | **permit**} **4bytegeneric** {**transitive** | **non-transitive**} *aa4:nn*

   または

   ip extcommunity-list expanded *list-name* {**deny** | **permit**} *expression*
3. （任意）**show ip extcommunity-list** *name*
4. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	**ip extcommunity-list standard** *list-name* {**deny** | **permit**} **4bytegeneric** {**transitive** | **nontransitive**} *community1* [*community2...*] **例**: `switch(config)# ip extcommunity-list standard BGPExtCommunity permit 4bytegeneric transitive 65535:20`	標準 BGP 拡張コミュニティ リストを作成します。*community* には、1 つ以上の拡張コミュニティを *aa4:nn* 形式で指定できます。
	ip extcommunity-list expanded *list-name* {**deny** | **permit**} *expression* **例**: `switch(config)# ip extcommunity-list expanded BGPExtComplex deny 1.5:[0-9][0-9]_`	正規表現を使用して拡張 BGP 拡張コミュニティ リストを作成します。
ステップ 3	**show ip community-list** *name* **例**: `switch(config)# show ip community-list BGPCommunity`	（任意）拡張コミュニティ リストの情報を表示します。
ステップ 4	**copy running-config startup-config** **例**: `switch# copy running-config startup-config`	（任意）この設定の変更を保存します。

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
65535:40 65535:60
switch(config)# copy running-config startup-config
```

ルート マップの設定

ルート マップは、ルートの再配布またはルート フィルタリングに使用できます。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. **route-map** *map-name* [**permit** | **deny**] [*seq*]
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	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	**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 コンフィギュレーション モードを開始します。*seq* を使用して、ルート マップ エントリを順序付けます。

ステップ 3	**continue** *seq* **例**: `switch(config-route-map)# continue 10`	（任意）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	**exit** **例**: `switch(config-route-map)# exit`	（任意）ルート マップ コンフィギュレーション モードを終了します。
ステップ 5	**copy running-config startup-config** **例**: `switch(config)# copy running-config startup-config`	（任意）この設定の変更を保存します。

ルート マップ コンフィギュレーション 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。

（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。
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の種類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 ipv6 address prefix-list *name* [*name...*] **例**: switch(config-route-map)# match ip address prefix-list AllowIPv6Prefix	1 つまたは複数の IPv6 プレフィックス リストと照合。プレフィックス リストは **ipv6 prefix-list** コマンドを使用して作成します。
match ip multicast [**source** *ipsource*] [[**group** *ipgroup*] [*rp iprp*]] **例**: switch(config-route-map)# match ip multicast rp 192.0.2.1	マルチキャスト送信元、グループ、またはランデブー ポイントに基づいて IPv4 マルチキャスト パケットを照合。

コマンド	目的
match ipv6 multicast [**source** *ipsource*] [[**group** *ipgroup*] [*rp iprp*]] **例**: `switch(config-route-map)# match ip multicast source 2001:0DB8::1`	マルチキャスト送信元、グループ、またはランデブー ポイントに基づいて IPv6 マルチキャスト パケットを照合。
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 ipv6 next-hop prefix-list *name* [*name...*] **例**: `switch(config-route-map)# match ipv6 next-hop prefix-list AllowIPv6Prefix`	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 IPv6 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 **ipv6 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 ipv6 route-source prefix-list *name* [*name...*] **例**: `switch(config-route-map)# match ipv6 route-source prefix-list AllowIPv6Prefix`	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 IPv6 ルート送信元アドレスを照合。プレフィックス リストは **ipv6 prefix-list** コマンドを使用して作成します。
match metric *value* [**+-** *deviation.*][*value..*] **例**: `switch(config-route-map)# match metric 50 + 10`	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* ～ *value + deviation*
match ospf-area *area-id* **例**: `switch(config-route-map)# match ospf-area 1`	OSPFv2 または OSPFv3 エリア ID と照合。 *area-id* の範囲は 0 ～ 4294967295 です。

コマンド	目的
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：外部ルート（BGP、EIGRP、OSPF タイプ 1 または 2） • inter-area：OSPF エリア間ルート • internal：内部ルート（OSPF エリア内またはエリア間ルートを含む） • intra-area：OSPF のエリア内ルート • level 1：IS-IS レベル 1 ルート • level 2：IS-IS レベル 2 ルート • local：ローカルで生成されたルート • nssa-external：NSSA 外部ルート（OSPF タイプ 1 または 2） • type-1：OSPF 外部タイプ 1 ルート • type-2：OSPF 外部タイプ 2 ルート
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 コンフィギュレーション 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*）をプリペンドできます。
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。

コマンド	目的
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 distance *value* **例**: `switch(config-route-map)# set distance 150`	OSPFv2 または OSPFv3 のルート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。
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。
set extcommunity 4byteas-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**　BGP 拡張コミュニティ属性を BGP ピアに伝達するには、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 **send-community** コマンドを使用します。

コマンド	目的
set extcommunity cost *community-id1 cost* [**igp** | **pre-bestpath**] [*community-id2...*]} **例:** switch(config-route-map)# set extcommunity cost 33 1.0:30	BGP ルート アップデートのコスト コミュニティ属性を設定します。この属性は、ローカルの自律システムまたは自律連合の BGP 最良パス選択プロセスをカスタマイズすることができます。*community-id* の範囲は 0 ～ 255 です。*cost* の範囲は 0 ～ 4294967295 です。最も低いコストを持つパスが優先されます。コストが同じ場合は、最も低いコスト コミュニティ番号を持つパスが優先されます。 **igp** キーワードは IGP コスト比較の後にコストを比較します。p**re-bestpath** キーワードは、ベスト パス アルゴリズムの他のすべてのステップの前に比較します。
set extcommunity rt *community-1* [**additive**] [community-2...]} **例:** switch(config-route-map)# set extcommunity rt 1.0:30	BGP ルート更新の拡張コミュニティ ルート ターゲット属性を設定します。*community* の値は、2 バイトの AS 番号: 4 バイトのネットワーク番号、4 バイトの AS 番号: 2 バイトのネットワーク番号、または IP アドレス: 2 バイトのネットワーク番号で指定します。 **additive** キーワードは、ルート ターゲットを既存の拡張コミュニティ ルート ターゲット属性に追加するために使用します。
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 ip next-hop unchanged **例:** switch(config-route-map)# set ip next-hop unchanged	不変のネクスト ホップ IP アドレスを指定します。このコマンドは、BGP IPv6-over-IPv4 ピアリングに必要です。
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（kbps）。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。指定できる範囲は 1 ～ 255（100% のロード）です。 • *metric4*：パスの MTU。有効な範囲は 1 ～ 16777215 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 nssa-only **例**: `switch(config-route-map)# set nssa-only`	P ビット セットを持たない ASBR で生成されたタイプ 7 LSA を設定します。これにより、OSPF で、タイプ 7 からタイプ 5 への LSA 変換が行われなくなります。
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および eBGP ネイバーのみに作用します。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 extcommunity-list [*name*]	拡張コミュニティ リストの情報を表示します。
show [**ip** | **ipv6**] **prefix-list** [*name*]	IPv4 または IPv6 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

次に、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およびマルチキャスト ルートがプレフィックス リスト AllowPrefix と一致した場合に、受け付けられるようにする例を示します。

```
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。

- 第 9 章「ベーシック BGP の設定」

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 ルーティングについて

ポリシーベース ルーティングを使用すると、IPv4 および IPv6 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 ルーティングがイネーブルのインターフェイスで受信する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。

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。
- QoS（Quality of Service）：ネットワークの周辺で IP パケット ヘッダーに優先または ToS（タイプ オブ サービス）値を設定することによって、またはキューイング メカニズムを利用して、ネットワークのコアまたはバックボーンでトラフィックにプライオリティを設定することによって、トラフィックを差別化します（『*Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide*』を参照）。
- 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 マップの各エントリで、match および set 文のコンビネーションを指定します。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。文の解釈は次のとおりです。

- 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、set 文節が適用されます。そのアクションの 1 つに、ネクスト ホップの選択が含まれます。
- 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、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、パケットがいずれのルート マップ文にも一致しない場合、そのパケットは通常の転送チャネルを介して返送され、宛先ベースのルーティングが実行されます。

(注) ポリシーベース 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

Cisco Nexus 9000シリーズ スイッチでは、ポリシーベース ルーティングで使用されるルート マップで次の **set** コマンドを使用できます。

- **set** {**ip** | **ipv6**} **next-hop** *address1* [*address2...*][**load-share**]
- **set interface null0**

これらの **set** コマンドは、ルート マップ シーケンス内で相互に排他的です。

最初のコマンドでは、IP アドレスでパケットの転送先へのパス上にある隣接ネクストホップ ルータを指定します。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) 必要に応じて、最大 32 の IP アドレスにトラフィックのロード バランシングを行うように、ネクストホップ アドレス用にこのコマンドを設定できます。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。

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。

ルート マップ処理ロジック

パケットがルート マップで設定されたインターフェイスに到着すると、転送ロジックがシーケンス番号順にそれぞれのルート マップ文を処理します。

ルート マップ文が **route-map...permit** 文の場合、パケットは **match** コマンドの一致基準と照合されます。このコマンドは、1 つ以上のアクセス コントロール エントリ(ACE)を持つ ACL を参照する場合があります。パケットが ACL の許可 ACE に一致すると、ポリシーベース ルーティング ロジックは **set** コマンドで指定されているアクションをパケットに対して実行します。

ルート マップ文が **route-map... deny** 文の場合、パケットは **match** コマンドの一致基準と照合されます。このコマンドは、1 つ以上の ACE を持つ ACL を参照する場合があります。パケットが ACL の許可 ACE に一致すると、ポリシーベース ルーティング プロセスが終了し、パケットはデフォルト IP ルーティング テーブルを使用してルーティングされます。

（注）　**set** コマンドは、**route-map... deny** 文内部に影響しません。

ルート マップ設定に match 文が含まれていない場合、ポリシーベース ルーティング ロジックは **set** コマンドで指定されているアクションをパケットに対して実行します。すべてのパケットは、ポリシーベース ルーティングを使用してルーティングされます。

ルート マップ設定が match 文を参照し、match 文が存在しない ACL または ACE エントリのない ACL を参照した場合、パケットはデフォルト ルーティング テーブルを使用してルーティングされます。

set {ip | ipv6} next-hop コマンドで指定されているネクスト ホップがダウンしているか、アクセス不能であるか、削除されている場合、パケットはデフォルト ルーティング テーブルを使用してルーティングされます。

ポリシーベース ルーティングのフィルタ オプション

追加のオプションを使用してトラフィックを識別できます。次のリストには、ほとんどの追加フィルタリング オプションが含まれていますが、すべてを網羅しているわけではありません。

ポリシーベース ルーティング ACL には、次の追加フィルタリング オプションが用意されています。

- レイヤ 3 送信元アドレスおよび宛先アドレス
- TCP/UDP ポート
- 優先レベル
- DiffServ コード ポイント（DSCP）値
- ACK、FIN、PSH、RST、SYN、または URG ビットがセットされた TCP パケット
- 確立済み TCP 接続
- パケット長

ポリシーベース ルーティングのライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	ポリシーベース ルーティングには Enterprise Services ライセンスが必要です。ライセンス パッケージに含まれていない機能は nx-os イメージにバンドルされており、無料で提供されます。NX-OS ライセンス方式の詳細については、『*Cisco NX-OS Licensing Guide*』を参照してください。

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 ルーティングを有効にします。
- 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。

注意事項および制約事項

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。

- ポリシーベース ルーティングのルート マップでは、1 つのルート マップ文に match 文または set 文を 1 つだけ指定できます。
- **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。
- 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。
- 一致基準としてのプレフィックス リストの使用はサポートされていません。ポリシーベース ルーティングのルート マップでプレフィックス リストを使用しないでください。
- ポリシーベース ルーティングは、ユニキャスト トラフィックのみをサポートしています。マルチキャスト トラフィックはサポートされていません。
- ポリシーベース ルーティングは、FEX ポートの着信トラフィックでサポートされていません。
- ポリシーベース ルーティングは、レイヤ 3 ポート チャネル サブインターフェイスではサポートされていません。
- ポリシーベース ルーティングのルート マップで使用される ACL には、拒否アクセス コントロール エントリ（ACE）を含めることができません。
- ポリシーベース ルーティングは、デフォルト システム ルーティング モードのみでサポートされます。
- Cisco Nexus 9000 シリーズ スイッチは、**set vrf** および **set default next-hop** コマンドをサポートしていません。
- ネクスト ホップが ECMP パス上で再帰的である場合は、ポリシーベース ルーティング トラフィックのロード バランシングを行うことはできません。代わりに、**set {ip | ipv6} next-hop** *ip-address* **load-share** コマンドを使用して隣接ネクスト ホップを指定してください。
- IP SLA PBR オブジェクト トラッキングの詳細については、『*Cisco Nexus 9000 Series NX-OS IP SLAs Configuration Guide*』を参照してください。
- Cisco NX-OS Release 6.1(2)I3(2) 以降では、Cisco Nexus 9000 シリーズ スイッチはポリシーベース ACL（PBACL）をサポートしています（オブジェクト グループ ACL とも呼びます）。詳細については、『*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*』を参照してください。
- 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。

デフォルト設定値

表 16-1 に、ポリシーベース ルーティングのデフォルト設定を示します。

表 16-1　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

この項では、次のトピックについて取り上げます。

- ポリシーベース ルーティング機能のイネーブル化（16-5 ページ）
- ルート ポリシーの設定（16-6 ページ）

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. **[no] feature pbr**
3. （任意）**show feature**
4. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	`[no] 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。
		ポリシーベース ルーティング機能をディセーブルにするには、このコマンドの **no** 形式を使用します。
		注　**no feature pbr** コマンドは、インターフェイスで適用されたポリシーを削除します。このコマンドによって ACL またはルート マップ設定は削除されず、システム チェックポイントも作成されません。

	コマンド	目的
ステップ 3	**show feature** **例**: `switch(config)# show feature`	(任意) イネーブルおよびディセーブルにされた機能を表示します。
ステップ 4	**copy running-config startup-config** **例**: `switch(config)# copy running-config startup-config`	(任意) この設定の変更を保存します。

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。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。

はじめる前に

IPv6 トラフィックに対してポリシーベース ルーティング ポリシーを適用する前に、IPv6 RACL TCAM リージョンを (TCAM カービングを使用して) 設定する必要があります。この手順については、『*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*』の「Configuring ACL TCAM Region Sizes」および「Configuring TCAM Carving - For Cisco NX-OS Release 6.1(2)I2(1) and Later Releases」を参照してください。

(注) スイッチには、IPv4 用の IPv4 RACL TCAM リージョンがデフォルトで用意されています。

手順の概要

1. **configure terminal**
2. **interface** *type slot*/*port*
3. {**ip** | **ipv6**} **policy route-map** *map-name*
4. **route-map** *map-name* [**permit** | **deny**] [*seq*]
5. **match** {**ip** | **ipv6**} **address access-list-name** *name* [*name*...]
6. (任意) **set ip next-hop** *address1* [*address2*...][**load-share**]
7. (Optional) **set ipv6 next-hop** *address1* [*address2*...][**load-share**]
8. (任意) **set interface null0**
9. (任意) **copy running-config startup-config**

手順の詳細

	コマンド	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:
	`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。

ステップ 3	{**ip** | **ipv6**} **policy route-map** *map-name* **例**: `switch(config-if)# ip policy route-map Testmap`	IPv4 または IPv6 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	**route-map** *map-name* [**permit** | **deny**] [*seq*]
	
	例:
	`switch(config-if)# route-map Testmap`
	`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 コンフィギュレーション モードを開始します。*seq* を使用して、ルート マップ エントリを順序付けます。

ステップ 5	**match** {**ip** | **ipv6**} **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 または IPv6 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。
ステップ 6	**set ip next-hop** *address1* [*address2...*][**load-share**] **例**: `switch(config-route-map)# set ip next-hop 192.0.2.1`	（任意）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 32 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。
ステップ 7	**set ipv6 next-hop** *address1* [*address2...*][**load-share**] **例**: `switch(config-route-map)# set ipv6 next-hop 2001:0DB8::1`	（任意）ポリシーベース ルーティング用の IPv6 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 32 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。

	コマンド	目的
ステップ 8	**set interface null0** **例**: `switch(config-route-map)# set interface null0`	(任意) ルーティングに使用するインターフェイスを設定します。パケットをドロップするには **null0** インターフェイスを使用します。
ステップ 9	**copy running-config startup-config** **例**: `switch(config-route-map)# copy running-config startup-config`	(任意) この設定の変更を保存します。

ポリシーベース ルーティングの設定確認

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、次の作業のいずれかを行います。

コマンド	目的
show [**ip** | **ipv6**] **policy** [*name*]	IPv4 または IPv6 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。

ポリシー統計をイネーブルにするには、**route-map** *map-name* **pbr-statistics** コマンドを使用します。ポリシー統計を消去するには、**clear route-map** *map-name* **pbr-statistics** コマンドを使用します。

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。

```
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。

```
switch# 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

switch# show route-map pbr-sample pbr-statistics
route-map pbr-sample, permit, sequence 10
 Policy routing matches: 84 packets
Default routing: 233 packets
```

```
switch# show ip policy
Interface   Route-map  Status  VRF-Name
Ethernet1/2 pbr-sample Active  --
```

関連資料

関連項目	マニュアル タイトル
IP SLA PBR オブジェクト トラッキング	*『Cisco Nexus 9000 Series NX-OS IP SLAs Configuration Guide』*
トラブルシューティング情報	*『Cisco Nexus 9000 Series NX-OS Troubleshooting Guide』*

HSRP の設定

この章では、Cisco NX-OS デバイスでホット 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 について

HSRP は、ファーストホップ IP ルータの透過的フェールオーバーが可能な、ファーストホップ冗長プロトコル（FHRP）です。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、さまざまな理由で現実的ではありません。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。

この項では、次のトピックについて取り上げます。

HSRP の概要

HSRP を使用する場合、HSRP 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 または IPv6 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 アドレスおよび仮想 IP アドレスを指定します。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 1 つをアクティブ ルータとして選択します。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。この時点で、選択されているスタンバイ ルータが HSRP グループの仮想 MAC および IP アドレスの制御を引き継ぎます。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。

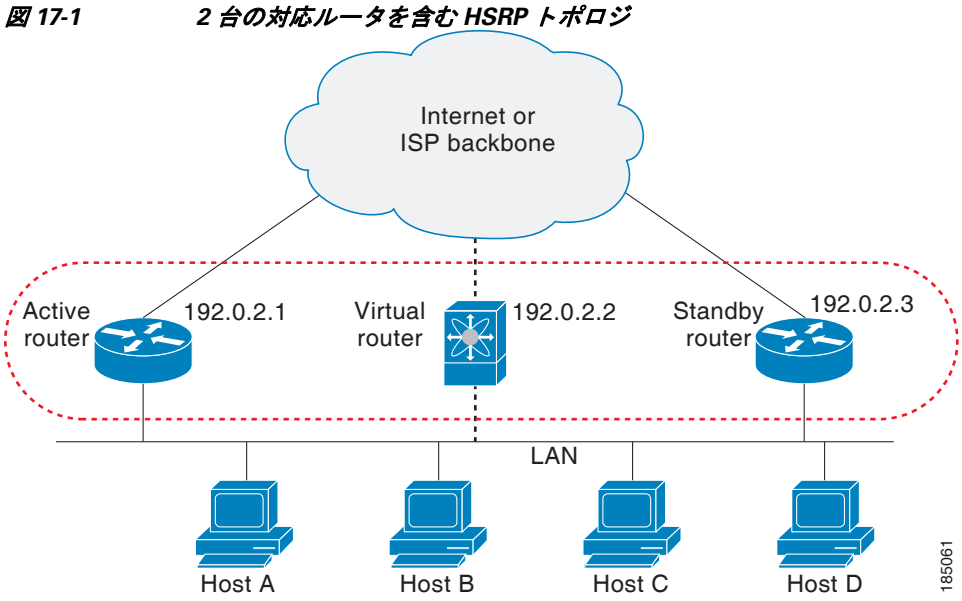
HSRP ではプライオリティ指示子を使用して、デフォルトのアクティブ ルータにする HSRP 設定インターフェイスを決定します。アクティブ ルータとしてインターフェイスを設定するには、グループ内の他のすべての HSRP 設定インターフェイスよりも高いプライオリティを与えます。デフォルトのプライオリティは 100 なので、それよりもプライオリティが高いインターフェイスを 1 つ設定すると、そのインターフェイスがデフォルトのアクティブ ルータになります。

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。

図 17-1 に、HSRP 対応として設定されたネットワークを示します。仮想 MAC アドレスおよび仮想 IP アドレスを共有することによって、2 つ以上のインターフェイスを単一の仮想ルータとして動作させることができます。

図 17-1　　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、仮想ルータの IP アドレス（仮想 IP アドレス）をホストのデフォルト ルータとして設定します。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。このプロセスには 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 アドレス宛のパケットは、アクティブ ルータ上で終端します。

HSRP のバージョン

Cisco NX-OS は、デフォルトで HSRP バージョン 1 をサポートします。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、HSRP バージョン 1 で使用する IP マルチキャスト アドレス 224.0.0.2 の代わりに、IPv4 マルチキャスト アドレス 224.0.0.102 または IPv6 マルチキャスト アドレス FF02::66 を使用して hello パケットを送信します。
- IPv4 では 0000.0C9F.F000 ～ 0000.0C9F.FFFF、IPv6 アドレスでは 0005.73A0.0000 ～ 0005.73A0.0FFF の MAC アドレス範囲を使用します。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。

IPv4 の HSRP

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 ルータが設定済みの IP アドレスと HSRP 仮想 MAC アドレスから hello パケットを取得するのに対して、スタンバイ ルータは、設定済みの IP アドレスとインターフェイス MAC アドレス（バーンド イン アドレス（BIA）である可能性があります）から hello パケットを取得します。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 for IPv6

IPv6 ホストは、IPv6 ネイバー探索（ND）ルータ アドバタイズメント（RA）メッセージを通じて使用可能な IPv6 ルータを学習します。これらのメッセージは、定期的にマルチキャストされる他、ホストによって送信要求されることもあります。ただし、デフォルト ルートがダウンしていることを検出したときの遅延時間は 30 秒以上になることもあります。IPv6 の HSRP は、IPv6 ND プロトコルを使用した場合よりも、代替デフォルト ルータへのスイッチオーバーが大幅に高速であり、ミリ秒タイマーが使用される場合は 1 秒未満になります。IPv6 の HSRP では、IPv6 ホストの仮想ファースト ホップを提供します。

HSRP の IPv6 インターフェイスを設定すると、IPv6 ND がルータのライフタイムがゼロで最終 RA を送信した後で、インターフェイスのリンクローカル アドレスに対する定期 RA が停止します。インターフェイスの IPv6 リンクローカル アドレスに制限はありません。他のプロトコルは、このアドレスへのパケットを送受信し続けます。

IPv6 ND は、HSRP グループがアクティブなときに、HSRP 仮想 IPv6 リンクローカル アドレスの定期 RA を送信します。これらの RA は、HSRP グループがアクティブ状態のままのときに、ルータのライフタイムがゼロで最終 RA が送信されると停止します。HSRP は、アクティブ HSRP グループ メッセージ（hello、coup、resign）でのみ仮想 MAC アドレスを使用します。

IPv6 の HSRP は、次のパラメータを使用します。

- HSRP バージョン 2
- UDP ポート 2029
- 0005.73A0.0000 ～ 0005.73A0.0FFF の範囲の仮想 MAC アドレス
- マルチキャスト リンクローカル IP 宛先アドレス FF02::66
- ホップ リミット 255

HSRP IPv6 アドレス

HSRP IPv6 グループには、HSRP グループ番号から導出される仮想 MAC アドレス、および HSRP 仮想 MAC アドレスからデフォルトで導出される仮想 IPv6 リンクローカル アドレスがあります。仮想 IPv6 リンクローカル アドレスを形成するために HSRP IPv6 グループのデフォルトの仮想 MAC アドレスが常に使用されます。グループによって実際に使用されている仮想 MAC アドレスは関係ありません。

表 17-1 に、IPv6 ネイバー探索パケットおよび HSRP パケットに使用される MAC アドレスおよび IP アドレスを示します。

表 17-1　HSRP および IPv6 ND アドレス

パケット	送信元 MAC アドレス	送信元 IPv6 アドレス	宛先 IPv6 アドレス	リンク層アドレス オプション
ネイバー送信要求（NS）	インターフェイス MAC アドレス	インターフェイス IPv6 アドレス	—	インターフェイス MAC アドレス
ルータ送信要求（RS）	インターフェイス MAC アドレス	インターフェイス IPv6 アドレス	—	インターフェイス MAC アドレス
ネイバー アドバタイズメント（NA）	インターフェイス MAC アドレス	インターフェイス IPv6 アドレス	仮想 IPv6 アドレス	HSRP 仮想 MAC アドレス
ルート アドバタイズメント（RA）	インターフェイス MAC アドレス	仮想 IPv6 アドレス	—	HSRP 仮想 MAC アドレス
HSRP（非アクティブ）	インターフェイス MAC アドレス	インターフェイス IPv6 アドレス	—	—
HSRP（アクティブ）	仮想 MAC アドレス	インターフェイス IPv6 アドレス	—	—

HSRP は、IPv6 リンクローカル アドレスをユニキャスト ルーティング情報ベース（URIB）に追加しません。リンクローカル アドレスには、セカンダリ仮想 IP アドレスがありません。

グローバル ユニキャスト アドレスの場合は、HSRP は URIB および IPv6 に仮想 IPv6 アドレスを追加します。

HSRP 認証

HSRP Message Digest 5（MD5）アルゴリズム方式の認証は、HSRP スプーフィング ソフトウェアから保護し、業界標準である MD5 アルゴリズムを使用して、信頼性およびセキュリティを向上させます。HSRP では、認証 TLV に IPv4 または IPv6 アドレスが含まれます。

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：このメッセージは、アクティブ ルータであるルータがシャットダウン直前、またはプライオリティの高いルータから hello または coup メッセージが送信されたときに、ルータから送信されます。

HSRP ロード シェアリング

HSRP では、1 つのインターフェイス上で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。図 17-2 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 17-2　HSRP ロード シェアリング

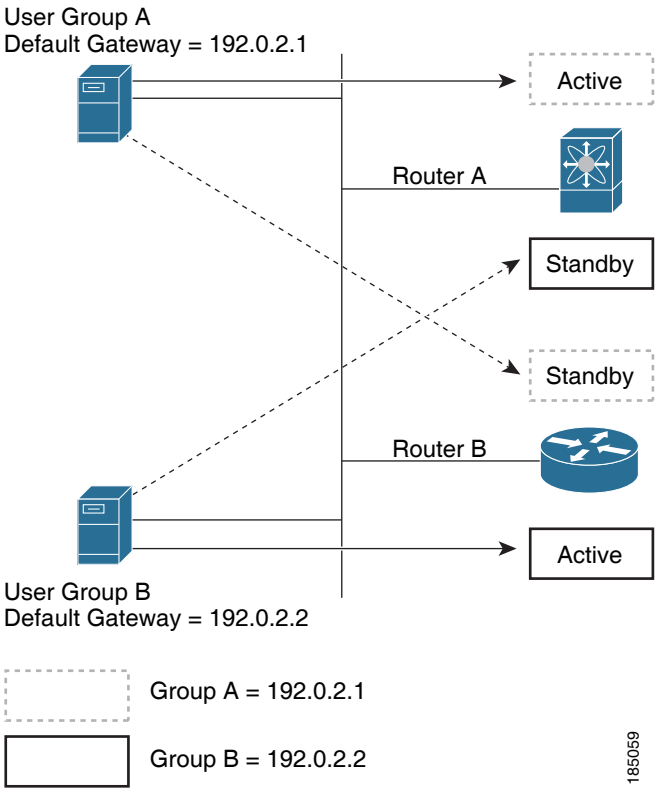


図 17-2 に、ルータ A、ルータ B、および 2 つの HSRP グループを示します。ルータ A はグループ A のアクティブ ルータですが、グループ B のスタンバイ ルータです。同様に、ルータ B はグループ B のアクティブ ルータであり、グループ A のスタンバイ ルータです。両方のルータがアクティブのままの場合、HSRP はホストからのトラフィックを両方のルータ間でロード バランシングを行います。どちらかのルータで障害が発生すると、残りのルータが引き続き、両方のホストのトラフィックを処理します。

（注）　IPv6 の HSRP では、デフォルトでロード バランシングを行います。サブネット上に 2 つの HSRP IPv6 グループが存在する場合、ホストはそれぞれのルータ アドバタイズメントから両方のグループを学習し、アドバタイズされたルータ間で負荷が共有されるように 1 つのグループを使用することを選択します。

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」セクション（17-19 ページ）を参照してください。

vPC と HSRP

HSRP は仮想ポート チャネル（vPC）と相互運用できます。vPC を使用すると、2 個の異なる Cisco Nexus 9000 シリーズ デバイスを物理的に接続し、第 3 のデバイスからは 1 つのポートとして見えるリンクが実現します。vPC の詳細については、*『Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide, Release 5.x』*を参照してください。

vPC は、アクティブ HSRP ルータとスタンバイ HSRP ルータの両方を通じてトラフィックを転送します。詳細については、「HSRP プライオリティの設定」セクション（17-21 ページ）および「HSRP の設定例」セクション（17-25 ページ）を参照してください。

（注）　HSRP アクティブは、異なる SVI のプライマリとセカンダリの両方の vPC ピアに配布できます。

vPC ピア ゲートウェイと HSRP

一部のサード パーティ製デバイスは HSRP 仮想 MAC アドレスを無視し、代わりに HSRP ルータの送信元 MAC アドレスを使用する場合があります。vPC 環境では、この送信元 MAC アドレスを使用するパケットが vPC ピア リンク経由で送信され、それによってパケットのドロップが発生する可能性があります。vPC ピア ゲートウェイを設定して、HSRP ルータで、ローカル vPC ピア MAC アドレスとリモート vPC ピア MAC アドレス、および HSRP 仮想 MAC アドレスに送信されたパケットを直接処理できるようにします。vPC ピア ゲートウェイの詳細については、*『Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide, Release 5.x』*を参照してください。

BFD

この機能は、IPv4 の Bidirectional Forwarding Detection（BFD）をサポートします。BFD は、高速転送とパス障害の検出時間を提供する検出プロトコルです。BFD は 2 台の隣接デバイス間のサブセカンド障害を検出し、BFD の負荷の一部を、サポートされるモジュール上のデータ プレーンに分散できるため、プロトコル hello メッセージよりも CPU を使いません。詳細については、『*Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*』を参照してください。

ハイ アベイラビリティおよび拡張ノンストップ フォワーディング

HSRP は、ステートフル リスタートおよびステートフル スイッチオーバーをサポートします。ステートフル リスタートは、HSRP プロセスが失敗してリスタートするときに行われます。ステートフル スイッチオーバーは、アクティブ スーパーバイザがスタンバイ スーパーバイザに切り替わるときに行われます。Cisco NX-OS スイッチオーバー後に実行時の設定を適用します。

HSRP ホールド タイマーが短時間に設定されている場合は、制御されたスイッチオーバー中に、これらのタイマーが切れる可能性があります。HSRP は、拡張型ノンストップ フォワーディング（NSF）をサポートし、制御されたスイッチオーバー

拡張 NSF を設定している場合、HSRP は延長されたタイマーを使用して hello メッセージを送信します。HSRP ピアは、この新しい値でホールド タイマーを更新します。タイマーが延長されることにより、スイッチオーバー時に不要な HSRP 状態の変更が発生することを防ぎます。スイッチオーバー後に、HSRP はホールド タイマーを元の設定値に復元します。スイッチオーバーに失敗すると、延長されたホールド タイマー値が満了してから HSRP はホールド タイマーを復元します。

詳細については、「HSRP の拡張ホールド タイマーの設定」セクション（17-23 ページ）を参照してください。

仮想化のサポート

HSRP 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。

HSRP のライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	HSRP にライセンスは不要です。ライセンス パッケージに含まれていない機能は nx-os イメージにバンドルされており、無料で提供されます。NX-OS ライセンス方式の詳細については、『*Cisco NX-OS Licensing Guide*』を参照してください。

HSRPP の前提条件

- 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 機能をデバイスでイネーブルにする必要があります。

HSRP の注意事項および制約事項

HSRP 設定時の注意事項および制約事項は、次のとおりです。

- 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。
- HSRP に IPv6 インターフェイスを設定するときは、HSRP バージョン 2 を設定する必要があります。
- 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。
- IPv4 に対する HSRP は、BFD でサポートされます。IPv6 に対する HSRP は、BFD でサポートされていません。
- HSRP IPv4 と HSRP IPv6 が同じ SVI の仮想 MAC アドレスを使用する場合、HSRP の状態は HSRP IPv4 と HSRP IPv6 の両方で同じである必要があります。フェールオーバー後に同じ状態になるようにするには、プライオリティとプリエンプションを設定する必要があります。
- 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場合、またはポート モードをレイヤ 2 に変更した場合は、Cisco NX-OS によって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。
- vPC で仮想 MAC アドレスを設定するときは、vPC ピアの両方で同じ仮想 MAC アドレスを設定する必要があります。
- vPC メンバである VLAN インターフェイスで HSRP MAC アドレスのバーンドイン オプションは使用できません。
- Release 7.0(3)I2(1) 以降、Cisco NX-OS ではダブルサイド vPC のすべてのノードで同じ HSRP グループを設定できます。
- 認証を設定していない場合、**show hsrp** コマンドは次の文字列を表示します。

  ```
  Authentication text "cisco"
  ```

  HSRP のデフォルトの動作は RFC 2281 で定義されています。

  認証データが設定されていない場合、推奨されるデフォルト値は `0x63 0x69 0x73 0x63 0x6F 0x00 0x00 0x00` **です。**

デフォルト設定値

表 17-2 に、HSRP パラメータのデフォルト設定を示します。

表 17-2　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	Version 1
プリエンプション	ディセーブル
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

HSRP の設定

この項では、次のトピックについて取り上げます。

- HSRP のイネーブル化（17-10 ページ）
- HSRP バージョン設定（17-11 ページ）
- IPv4 の HSRP グループの設定（17-11 ページ）
- IPv6 の HSRP グループの設定（17-13 ページ）
- HSRP 仮想 MAC アドレスの設定（17-15 ページ）
- HSRP の認証（17-16 ページ）
- HSRP オブジェクト トラッキングの設定（17-19 ページ）
- HSRP プライオリティの設定（17-21 ページ）
- HSRP のカスタマイズ（17-22 ページ）
- HSRP の拡張ホールド タイマーの設定（17-23 ページ）

（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 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 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`	すべてのグループの 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。

（注） IPv6 HSRP グループは、HSRP バージョン 2 として設定する必要があります。

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 インターフェイス上で HSRP グループを設定し、その HSRP グループに仮想 IP アドレスおよび仮想 MAC アドレスを設定できます。

はじめる前に

HSRP 機能がイネーブルになっていることを確認します(「HSRP のイネーブル化」セクション (17-10 ページ)を参照)。

Cisco NX-OS では、仮想 IP アドレスを設定すると HSRP グループ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. **ip address** ***ip-address/length***
4. **hsrp** ***group-number*** **[ipv4]**
5. **ip [***ip-address* **[secondary]]**
6. **exit**
7. **no shutdown**
8. （任意）**show hsrp [group** ***group-number*****] [ipv4]**
9. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 number*
	
	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。

ステップ 3	**ip address** *ip-address/length* **例**: `switch(config-if)# ip 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 4	**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 5	**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、グループをイネーブルにします。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。

	コマンド	目的
ステップ 6	**exit** **例**: switch(config-if-hsrp)# exit	HSRP 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	**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。
ステップ 8	**show hsrp** [**group** *group-number*] [**ipv4**] **例**: switch(config-if)# show hsrp group 2	(任意)HSRP 情報を表示します。
ステップ 9	**copy running-config startup-config** **例**: switch(config-if)# copy running-config startup-config	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

（注）　設定完了後にインターフェイスをイネーブルにするには、**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)# 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
```

IPv6 の HSRP グループの設定

IPv6 インターフェイス上で HSRP グループを設定し、その HSRP グループに仮想 MAC アドレスを設定できます。

IPv6 の HSRP グループを設定すると、HSRP はリンクローカル プレフィックスからリンクローカル アドレスを生成します。HSRP では、Modified EUI-64 形式のインターフェイス ID も生成します。EUI-64 インターフェイス ID は、関連の HSRP 仮想 MAC アドレスから作成されます。

はじめる前に

HSRP をイネーブルにする必要があります(「HSRP のイネーブル化」セクション(17-10 ページ)を参照)。

IPv6 HSRP グループを設定するインターフェイスで HSRP バージョン 2 がイネーブルになっていることを確認します。

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 属性を設定してあることを確認します。

手順の概要

1. **configure terminal**
2. **interface** *type number*
3. **ipv6 address** *ipv6-address/length*
4. **hsrp version 2**
5. **hsrp** *group-number* **ipv6**
6. **ip** *ipv6-address*
7. **ip autoconfig**
8. **exit**
9. **no shutdown**
10. （任意）**show hsrp [group** *group-number***] [ipv6]**
11. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 number*
	
	例:
	`switch(config)# interface ethernet 3/2`
	`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 3	**ipv6 address** *ipv6-address/length* **例**: `switch(config-if)# ipv6 address 2001:0DB8::0001:0001/64`	インターフェイスの IPv6 アドレスを設定します。
ステップ 4	**hsrp version 2** **例**: `switch(config-if-hsrp)# hsrp version 2`	HSRP バージョン 2 にこのグループを設定します。

ステップ 5	**hsrp** *group-number* **ipv6**
	
	例:
	`switch(config-if)# hsrp 10 ipv6`
	`switch(config-if-hsrp)#`
		IPv6 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 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** *ipv6-address* **例**: `switch(config-if-hsrp)# ip 2001:DB8::1`	HSRP グループの仮想 IPv6 アドレスを設定し、そのグループをイネーブルにします。
ステップ 7	**ip autoconfig** **例**: `switch(config-if-hsrp)# ip autoconfig`	計算されたリンクローカル仮想 IPv6 アドレスから HSRP グループの仮想 IPv6 アドレスを自動設定し、グループをイネーブルにします。

	コマンド	目的

ステップ 8	**exit**
	
	例:
	`switch(config-if-hsrp)# exit`
	`switch(config-if)#`
		HSRP コンフィギュレーション モードを終了します。

ステップ 9	**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。
ステップ 10	**show hsrp** [**group** *group-number*] [**ipv6**] **例**: `switch(config-if)# show hsrp group 10`	(任意)HSRP 情報を表示します。
ステップ 11	**copy running-config startup-config** **例**: `switch(config-if)# copy running-config startup-config`	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

(注) 設定完了後にインターフェイスをイネーブルにするには、**no shutdown** コマンドを使用する必要があります。

次に Ethernet 3/2 上で IPv6 HSRP グループを設定する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 3/2
switch(config-if)# ipv6 address 2001:0DB8::0001:0001/64
switch(config-if-hsrp)# hsrp version 2
switch(config-if)# hsrp 2 ipv6
switch(config-if-hsrp)# ip 2001:DB8::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

設定されたグループ番号に基づいて HSRP が生成したデフォルト仮想 MAC アドレスを変更できます。

(注) vPC リンクの vPC ピアの両方で同じ仮想 MAC アドレスを設定する必要があります。

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。任意で **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 BIA 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 認証ではキーチェーンを使用します（*『Cisco Nexus 9000 Series NX-OS Security Configuration Guide』*を参照）。

はじめる前に

HSRP をイネーブルにする必要があります（「HSRP のイネーブル化」セクション（17-10 ページ）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. **hsrp** ***group-number*** **[ipv4 | ipv6]**
4. **authentication text** ***string***

   または

   authentication md5 {**key-chain** *key-chain* | **key-string** {**0** | **7**} *text* [**timeout** *seconds*]}
5. （任意）**show hsrp [group** ***group-number***]
6. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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。

	コマンド	目的

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 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	**hsrp** *group-number* [**ipv4** | **ipv6**]
	
	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 4	**authentication text** *string* **例**: `switch(config-if-hsrp)# authentication text mypassword`	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。
	authentication md5 {**key-chain** *key-chain* | **key-string** {**0** | **7**} *text* [**timeout** *seconds*]} **例**: `switch(config-if-hsrp)# authentication md5 key-chain hsrp-keys`	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、必要に応じて、HSRP が新しいキーのみを受け入れる時間の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。
ステップ 5	**show hsrp** [**group** *group-number*] **例**: `switch(config-if-hsrp)# show hsrp group 2`	(任意) HSRP 情報を表示します。
ステップ 6	**copy running-config startup-config** **例:** `switch(config-if-hsrp)# copy running-config startup-config`	(任意) 実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

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 2013 23:59:59 Sep 12 2013
switch(config-keychain-key) send-lifetime 00:00:00 Jun 01 2013 23:59:59 Aug 12 2013
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 2013 23:59:59 Dec 12 2013
switch(config-keychain-key) send-lifetime 00:00:00 Sep 12 2013 23:59:59 Nov 12 2013
switch(config-keychain-key)# interface ethernet 1/2
switch(config-if)# hsrp 2
switch(config-if-hsrp)# authentication md5 key-chain hsrp-keys
switch(config-if-hsrp)# copy running-config startup-config
```

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 認証ではキーチェーンを使用します(*『Cisco Nexus 9000 Series NX-OS Security Configuration Guide』*を参照)。

はじめる前に

HSRP をイネーブルにする必要があります（「HSRP のイネーブル化」セクション（17-10 ページ）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. **hsrp** ***group-number*** **[ipv4 | ipv6]**
4. **authentication text** ***string***

   または

   authentication md5 {**key-chain** *key-chain* | **key-string** {**0** | **7**} *text* [**timeout** *seconds*]}
5. （任意）**show hsrp [group** ***group-number***]
6. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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** *interface-type 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。

ステップ 3	**hsrp** *group-number* [**ipv4**	**ipv6**]
		
		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 4	**authentication text** *string* **例**: `switch(config-if-hsrp)# authentication text mypassword`	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。
	authentication md5 {**key-chain** *key-chain*	**key-string** {**0**	**7**} *text* [**timeout** *seconds*]} **例**: `switch(config-if-hsrp)# authentication md5 key-chain hsrp-keys`	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、必要に応じて、HSRP が新しいキーのみを受け入れる時間の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。

	コマンド	目的
ステップ 5	**show hsrp** [**group** *group-number*] **例**: switch(config-if-hsrp)# show hsrp group 2	(任意)HSRP 情報を表示します。
ステップ 6	**copy running-config startup-config** **例**: switch(config-if-hsrp)# copy running-config startup-config	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

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 2013 23:59:59 Sep 12 2013
switch(config-keychain-key) send-lifetime 00:00:00 Jun 01 2013 23:59:59 Aug 12 2013
switch(config-keychain-key) key 1
```

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、HSRP グループ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。

手順の概要

1. **configure terminal**
2. **track** *object-id* **interface** *interface-type slot/port* **{line-protocol | ip routing | ipv6 routing}**
3. **track** *object-id* **{ip | ipv6} route** *ip-prefix/length* **reachability**
4. **exit**
5. **interface** *interface-type slot/port*
6. **hsrp** *group-number* **[ipv4 | ipv6]**
7. **priority [***value***]**
8. **track** *object-id* **[decrement** *value***]**
9. **preempt [delay [minimum** *seconds***] [reload** *seconds***] [sync** *seconds***]]**
10. （任意）**show hsrp interface** *interface-type slot/port*
11. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	**track** *object-id* **interface** *interface-type slot/port* {**line-protocol** | **ip routing** | **ipv6 routing**}
	
	例:
	`switch(config)# track 1 interface ethernet 2/2 line-protocol`
	`switch(config-track)#`
		トラック オブジェクトがトラッキング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、次のようにトラック オブジェクトのステータスを左右します。
		• グローバル 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 routing** or **ipv6 routing** 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。

ステップ 3	**track** *object-id* {**ip** | **ipv6**} **route** *ip-prefix/length* **reachability** **例**: `switch(config-track)# track 2 ip route 192.0.2.0/8 reachability`	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。

ステップ 4	**exit**
	
	例:
	`switch(config-track)# exit`
	`switch(config)#`
		トラック 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	**interface** *interface-type 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。

ステップ 6	**hsrp** *group-number* [**ipv4** | **ipv6**]
	
	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 7	**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。

	コマンド	目的
ステップ 8	**track** *object-id* [**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 9	**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。任意で、遅延を設定して、HSRP グループのプリエンプションを設定した時間だけ遅らせることができます。指定できる範囲は 0 ～ 3600 秒です。
ステップ 10	**show hsrp interface** *interface-type slot/port* **例**: `switch(config-if-hsrp)# show hsrp interface ethernet 1/2`	(任意) インターフェイスの HSRP 情報を表示します。
ステップ 11	**copy running-config startup-config** **例**: `switch(config-if-hsrp)# copy running-config startup-config`	(任意) 実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

次に、Ethernet インターフェイス 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-track)# track 2 ip route 192.0.2.0/8 reachability
switch(config-track)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# hsrp 2
switch(config-if-hsrp)# priority 254
switch(config-if-hsrp)# track 1 decrement 20
switch(config-if-hsrp)# preempt delay minimum 60
switch(config-if-hsrp)# copy running-config startup-config
```

HSRP プライオリティの設定

HSRP グループの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。vPC 対応のインターフェイスで HSRP を設定する場合は、オプションで vPC トランクにフェールオーバーする時期を制御するしきい値の上限と下限を設定できます。スタンバイ ルータのプライオリティが下限のしきい値を下回った場合、HSRP は、すべてのスタンバイ ルータ トラフィックを vPC トランク全体に送信し、アクティブな HSRP ルータを通して転送します。HSRP では、スタンバイ HSRP ルータ プライオリティが上限しきい値を超えるまで、この状況を維持します。

IPv6 HSRP グループでは、すべてのグループ メンバのプライオリティが同じ場合、HSRP は IPv6 リンクローカル アドレスに基づいてアクティブ ルータを選択します。

HSRP プライオリティを設定するには、HSRP グループ 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。オプションで、このコマンドを使用して vPC トランクにフェールオーバーする時点を決定するために vPC が使用するしきい値の上限と下限を設定できます。lower-value の範囲は 1 ～ 255 です。デフォルトは 1 です。upper-value の範囲は 1 ～ 255 です。デフォルトは 255 です。

HSRP のカスタマイズ

任意で、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*.たとえば、hsrp-Eth2/1-1 です。
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。任意で、遅延を設定して、HSRP グループのプリエンプションを設定した時間だけ遅らせることができます。指定できる範囲は 0 ～ 3600 秒です。

コマンド	目的
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。 • *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 1 ～ 254 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 3 ～ 255 です。 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	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 の拡張ホールド タイマーの設定

制御された（グレースフル）スイッチオーバー中に拡張 NSF をサポートするために拡張ホールド タイマーを使用するように HSRP を設定できます。拡張ホールド タイマーは、すべての HSRP ルータ上で設定してください（「ハイ アベイラビリティおよび拡張ノンストップ フォワーディング」セクション（17-8 ページ）を参照）。

（注）　拡張ホールド タイマーを設定する場合は、すべての HSRP ルータで拡張ホールド タイマーを設定する必要があります。デフォルトでないホールド タイマーを設定する場合は、HSRP 拡張ホールド タイマーの設定時にすべての HSRP ルータで同じ値を設定してください。

（注）　HSRP 拡張ホールド タイマーは、HSRPv1 のミリ秒の hello タイマーやホールド タイマーを設定した場合は適用されません。これは、HSRPv2 には適用されません。

HSRP 拡張ホールド タイマーを設定するには、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
hsrp timers extended-hold [*timer*] **例**: `switch(config)# hsrp timers extended-hold`	IPv4 と IPv6 両方のグループに HSRP 拡張ホールド タイマーを秒単位で設定します。タイマーの範囲は 10 ～ 255 です。デフォルトは 10 です。

拡張ホールド時間を表示するには、**show hsrp** コマンドまたは **show running-config hsrp** コマンドを使用します。

HSRP 設定の確認

HSRP 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 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 2013 23:59:59 Sep 12 2013
    send-lifetime 00:00:00 Jun 01 2013 23:59:59 Aug 12 2013
  key 1
    key-string 7 uaeqdyito
    accept-lifetime 00:00:00 Aug 12 2013 23:59:59 Nov 12 2013
    send-lifetime 00:00:00 Sep 12 2013 23:59:59 Nov 12 2013
feature hsrp
track 2 interface ethernet 2/2 ip
interface ethernet 1/2
 ip address 192.0.2.2/8
 hsrp 1
 authenticate md5 key-chain hsrp-keys
 priority 90
 track 2 decrement 20
 ip 192.0.2.10
 no shutdown
```

次に、インターフェイス上で HSRP プライオリティを設定する例を示します。

```
interface vlan 1
hsrp 0
    preempt
    priority 100 forwarding-threshold lower 80 upper 90
    ip 192.0.2.2
    track 1 decrement 30
```

その他の関連資料

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（17-25 ページ）
- MIB（17-26 ページ）

関連資料

関連項目	マニュアル タイトル
VRRP の設定	第 18 章「VRRP の設定」
ハイ アベイラビリティの設定	*『Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide』*

MIB

MIB	MIB のリンク
HSRP に関連する MIB	サポートされている MIB を検索およびダウンロードするには、次の URL にアクセスしてください。 ftp://ftp.cisco.com/pub/mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html

VRRP の設定

この章では、Cisco NX-OS デバイス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 ではそのグループのマスター ルータが選択され、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なります。残りのルータはスタンバイになり、マスター ルータで障害が発生した場合に処理を引き継ぎます。

この項では、次のトピックについて取り上げます。

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。

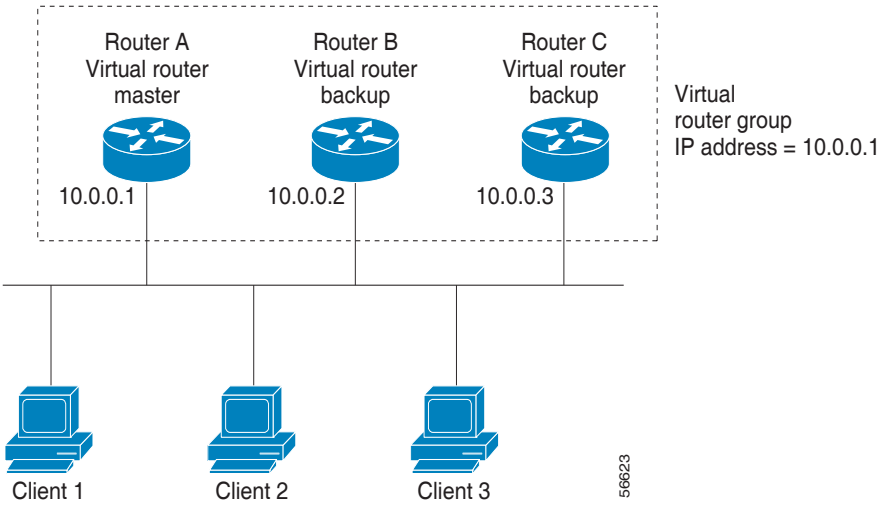
ダイナミック ディスカバリ プロトコルのデメリットは、LAN クライアントにある程度、設定および処理のオーバーヘッドが発生することです。また、ルータが故障した場合、他のルータに切り替えるプロセスも遅くなる場合があります。

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。この方法を使用すると、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。

図 18-1 基本的な VLAN トポロジを示します。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 18-1　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 アドレスを使用するので、ルータ A がマスター（別名、IP アドレス オーナー）です。ルータ A はマスターとして、VRRP グループの仮想 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。マスター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 ルータがマスターになり、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、そのルータが再びマスターになります。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」の項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。そのルータがマスター 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。これらのパケットには、ping トラフィックと Telnet トラフィックが含まれます。VRRP 仮想 IP アドレス宛のレイヤ 2（VLAN）インターフェイスで受信したパケットは、マスター ルータで終端します。

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複数の 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、複数の 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：障害マスター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、専用のインターネット割り当て番号局（IANA）規格マルチキャスト アドレス（224.0.0.18）を使用します。このアドレッシング方式によって、マルチキャストを提供するルータ数が最小限になり、テスト機器でセグメント上の VRRP パケットを正確に識別できるようになります。IANA 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マスターになることが保証されます。

マルチ VRRP グループ

物理インターフェイス上で複数の VRRP グループを設定できます。サポートされる VRRP グループの数については、『*Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*』を参照してください。

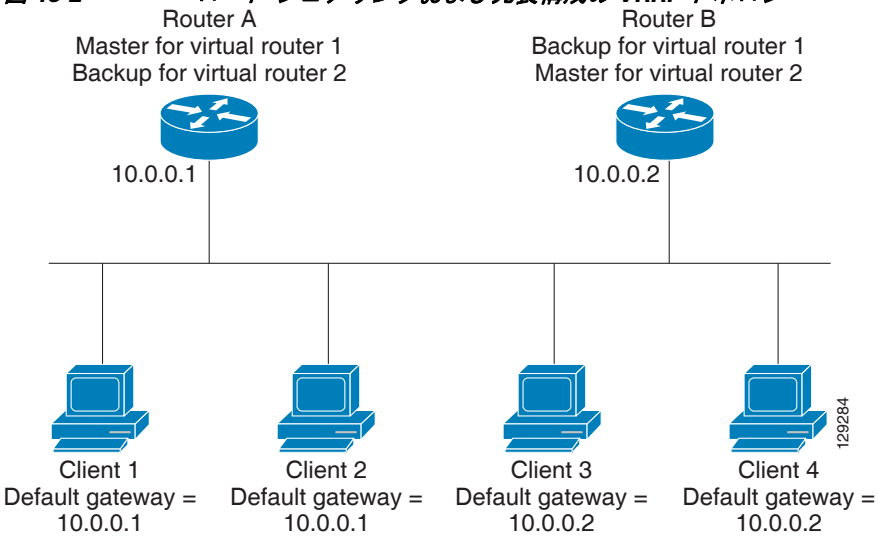
ルータ インターフェイスがサポートできる VRRP グループの数は、次の要因によって決まります。

- 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 グループのマスター、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。

図 18-2 ルータ A および B がクライアント 1 ～ 4 との間でトラフィックを共有するように VRRP が設定されている LAN トポロジを示します。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 18-2　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、マスターです。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 ～ 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、マスターです。ルータ A はルータ B のバックアップです。クライアント 3 ～ 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 冗長構成の重要なポイントは、VRRP ルータのプライオリティです。プライオリティによって、各 VRRP ルータが果たす役割が決まり、マスター ルータで障害が発生した場合のアクションが決まるからです。

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、そのルータはマスターとして機能します。マスター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、マスターで障害が発生した場合にマスターになる順序も決まります。

たとえば、ルータ A が LAN トポロジにおけるマスターであり、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 をマスター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マスター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 ルータがマスター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 は新しいマスター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 がマスターであり、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 を新しいマスターとして選択します。

プリエンプションをディセーブルにした場合、VRRP が切り替わるのは、元のマスターが回復した場合、または新しいマスターで障害が発生した場合に限られます。

vPC および VRRP

VRRP は仮想ポート チャネル（vPC）と相互運用できます。vPC を使用すると、2 個の異なる Cisco Nexus 9000 シリーズ デバイスを物理的に接続し、第 3 のデバイスからは 1 つのポートとして見えるリンクが実現します。vPC の詳細については、*『Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide, Release 5.x』*を参照してください。

vPC はマスター VRRP ルータとバックアップ VRRP ルータの両方を使用してトラフィックを転送します。「VRRP プライオリティの設定」セクション（18-12 ページ）を参照してください。

（注）　プライマリ vPC ピア デバイスの VRRP をアクティブに、セカンダリ vPC デバイスの VRRP をスタンバイにそれぞれ設定する必要があります。

VRRP のアドバタイズメント

VRRP マスター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は、マスターのプライオリティおよびステートを伝達します。Cisco NX-OS 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、次の認証機能をサポートします。

- 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、第 19 章「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 グループのマスター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」セクション（18-18 ページ）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。

BFD

この機能は、IPv4 の Bidirectional Forwarding Detection（BFD）をサポートします。BFD は、高速転送とパス障害の検出時間を提供する検出プロトコルです。BFD は 2 台の隣接デバイス間のサブセカンド障害を検出し、BFD の負荷の一部を、サポートされるモジュール上のデータ プレーンに分散できるため、プロトコル hello メッセージよりも CPU を使いません。詳細については、*『Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide』*を参照してください。

VRRPv3 と VRRS に関する情報

VRRP のバージョン 3（VRRPv3）では、スイッチのグループで単一の仮想スイッチを形成して、冗長性を実現し、ネットワーク内のシングル ポイント障害が生じる可能性を減らすことができます。これにより、仮想スイッチをデフォルト ゲートウェイとして使用するように、LAN クライアントを設定できます。スイッチのグループを表す仮想スイッチは、VRRPv3 グループとも呼ばれます。

仮想ルータ冗長サービス（VRRS）では、VRRPv3 を監視することでステートレス冗長サービスを VRRS 経路と VRRS クライアントに提供することで VRRPv3 のスケーラビリティが向上します。VRRPv3 は、VRRPv3 ステータス情報（現在および過去の冗長状態、アクティブおよび非アクティブのレイヤ 2 およびレイヤ 3 アドレスなど）を VRRS 経路とすべての登録済み VRRS クライアントに配信する VRRS サーバとして機能します。

VRRS クライアントは、VRRPv3 を使用して、グループのステートに応じてサービスやリソースを提供または抑制する他の Cisco プロセスまたはアプリケーションです。VRRS 経路は、VRRS データベース情報を使用して、拡張インターフェイス環境全体に拡張ファーストホップ ゲートウェイの冗長性を提供する特殊な VRRS クライアントです。

VRRS は、単独ではそれ自身のステートを管理することしかできません。VRRPv3 グループに VRRS クライアントをリンクすると、ステートレスまたはステートフル フェールオーバーが実装可能になるように、VRRS でクライアント アプリケーションにサービスを提供できるようにするメカニズムが提供されます。ステートフル フェールオーバーでは、フェールオーバーが発生したときに運用データが失われないように障害の前に所定バックアップとの通信が必要になります。

VRRS 経路はクライアントと同様に動作しますが、VRRS アーキテクチャと統合されます。この経路により、何百ものインターフェイス間で 1 つの仮想アドレスを設定することでファーストホップ ゲートウェイの冗長性を拡張する方法が提供されます。VRRS 経路の仮想ゲートウェイの状態は、ファーストホップ冗長プロトコル（FHRP）VRRS サーバの状態によります。

VRRPv3 は、現在の状態（マスター、バックアップ、または運用不可能な初期状態（INIT））を VRRS に通知し、その情報を経路またはクライアントに渡します。VRRPv3 グループ名は、VRRS をアクティブにし、VRRPv3 グループをクライアントまたは同じ名前の VRRS の一部として設定されている経路と関連付けます。

経路およびクライアントは、VRRPv3 サーバの状態で機能します。VRRPv3 グループの状態が変化すると、VRRS 経路とクライアントの動作（インターフェイスのシャット ダウン、アカウンティング ログの追加などのタスクの実行）が VRRS から受信した状態により変化します。

VRRPv3 の利点

VRRPv3 の利点は、次のとおりです。

- マルチベンダー環境での相互運用性。
- IPv4 および IPv6 アドレス ファミリのサポート。
- VRRS 経路によるスケーラビリティの向上。

ハイ アベイラビリティ

VRRP は、ステートフル リスタートとステートフル スイッチオーバーを通してハイ アベイラビリティをサポートします。ステートフル リスタートは、VRRP が障害を処理してリスタートするときに行われます。ステートフル スイッチオーバーは、アクティブ スーパーバイザがスタンバイ スーパーバイザに切り替わるときに行われます。Cisco NX-OS スイッチオーバー後に実行時の設定を適用します。

VRRPv3 は、ステートフル スイッチオーバー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。

VRRP のライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	VRRP にライセンスは不要です。ライセンス パッケージに含まれていない機能は nx-os イメージにバンドルされており、無料で提供されます。NX-OS ライセンス方式の詳細については、『*Cisco NX-OS Licensing Guide*』を参照してください。

VRRP の注意事項と制約事項

VRRP 設定時の注意事項および制約事項は、次のとおりです。

- 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、ネットワーク上のデバイス全体で 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。

- インターフェイス VRF メンバーシップまたはポート チャネル メンバーシップを変更した場合、またはポート モードをレイヤ 2 に変更した場合は、Cisco NX-OS によって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 BFD は、2 台のルータ間でのみ設定できます。

VRRPv3 の注意事項と制約事項

VRRPv3 設定時の注意事項と制約事項は次のとおりです。

- VRRPv3 は既存のダイナミック プロトコルの代替にはなりません。VRRPv3 は、マルチアクセス、マルチキャスト、または ブロードキャスト対応イーサネット LAN で使用するために設計されています。
- VRRPv3 は、イーサネットおよびファスト イーサネット インターフェイス、ブリッジ グループ仮想インターフェイス（BVI）、ギガビット イーサネット インターフェイス、および VLAN でのみサポートされます。
- VRRPv3 が使用中の場合、VRRPv2 は使用できません。VRRPv3 を設定するには、VRRPv2 設定をディセーブルにする必要があります。
- VRRS は現在、VRRPv3 と合わせて使用する場合にのみ使用できます。
- VRRPv3 ミリ秒タイマーは、絶対に必要な場合以外は使用しないようにし、使用する場合は慎重な検討とテストが必要です。ミリ秒の値は望ましい状況でのみ動作します。ミリ秒のタイマー値は、VRRPv3 も含めてサポートしている限り、サードパーティ ベンダーと互換性があります。
- VRRPv3 が VRRS 経路の冗長インターフェイスと同じネットワーク パス上で動作する場合にのみ、完全なネットワークの冗長性を実現できます。完全な冗長性のために、次の制約事項が適用されます。
  - VRRS 経路は、親 VRRPv3 グループと同じ物理インターフェイスを使用する必要があるか、または親 VRRPv3 グループと同じ物理インターフェイスを持つサブインターフェイス上で設定する必要があります。
  - VRRS 経路をスイッチ仮想インターフェイス（SVI）に設定できるのは、関連付けられた VLAN が親 VRRPv3 グループが設定された VLAN と同じトランクを共有する場合のみです。
- VRRPv2 とは異なり、VRRPv3 は障害検出を高速化するための双方向フォワーディング検出をサポートしていません。
- VRRPv2 とは異なり、VRRPv3 はネイティブおよびオブジェクト トラッキングをサポートしていません。

VRRP パラメータのデフォルト設定

表 18-1 に、VRRP パラメータのデフォルト設定を示します。

表 18-1　デフォルトの VRRP パラメータ

パラメータ	デフォルト
VRRP	ディセーブル
アドバタイズ インターバル	1 秒
認証	認証なし
プリエンプション	イネーブル
プライオリティ	100

VRRPv3 パラメータのデフォルト設定

表 18-1 に、VRRPv3 パラメータのデフォルト設定を示します。

表 18-2　デフォルトの VRRPv3 パラメータ

パラメータ	デフォルト
VRRPv3	ディセーブル
VRRS	ディセーブル
VRRPv3 セカンダリ アドレスの一致	イネーブル
VRRPv3 グループのプライオリティ	100
VRRPv3 アドバタイズメント タイマー	1000 ミリ秒

VRRP の設定

この項では、次のトピックについて取り上げます。

- VRRP 機能のイネーブル化（18-11 ページ）
- VRRP グループの設定（18-11 ページ）
- VRRP プライオリティの設定（18-12 ページ）
- VRRP 認証の設定（18-14 ページ）
- アドバタイズメント パケットのタイム インターバル設定（18-15 ページ）
- プリエンプションのディセーブル化（18-17 ページ）
- VRRP インターフェイス ステート トラッキングの設定（18-18 ページ）

（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、グループをイネーブル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。マスター 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 マスター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、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 マスターの場合に、これらのパケットを受け付けます。

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イネーブルにする必要があります。

はじめる前に

インターフェイス上で IP アドレスが設定されていることを確認します（「IPv4 アドレス指定の設定」セクション（2-9 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp** *number*
4. **address** ***ip-address*** **[secondary]**
5. **no shutdown**

6. （任意）**show vrrp**
7. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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** *interface-type slot/port*
	
	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。

ステップ 3	**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 4	**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 5	**no shutdown**
	
	例:
	`switch(config-if-vrrp)# no shutdown`
	`switch(config-if-vrrp)#`
		VRRP グループをイネーブルにします。デフォルトでは、ディセーブルです。

ステップ 6	**show vrrp** **例**: `switch(config-if-vrrp)# show vrrp`	（任意）VRRP 情報を表示します。
ステップ 7	**copy running-config startup-config** **例**: `switch(config-if-vrrp)# copy running-config startup-config`	（任意）この設定の変更を保存します。

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 アドレスと同じデバイス（マスター）の場合、デフォルト値は 255 です。

vPC 対応のインターフェイスで VRRP を設定する場合は、オプションで vPC トランクにフェールオーバーする時期を制御するしきい値の上限と下限を設定できます。バックアップ ルータのプライオリティが下限のしきい値を下回った場合、VRRP は、すべてのバックアップ ルータ トラフィックを vPC トランク全体に送信し、マスター VRRP ルータを通して転送します。バックアップ VRRP ルータのプライオリティがしきい値の上限を超えるまで、VRRP はこの処理を継続します。

はじめる前に

VRRP をイネーブルにする必要があります（「VRRP の設定」セクション（18-10 ページ）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」セクション（2-9 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp** *number*
4. シャットダウン
5. **priority** *level* **[forwarding-threshold lower** *lower-value* **upper** *upper-value***]**
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	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 interface-type slot/port`
	
	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。

ステップ 3	`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。

ステップ 4	**シャットダウン**
	
	例:
	`switch(config-if-vrrp)# shutdown`
	`switch(config-if-vrrp)#`
		VRRP グループをディセーブルにします。デフォルトでは、ディセーブルです。

	コマンド	目的
ステップ 5	**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。*level* の範囲は 1 ～ 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 アドレスと等しいマスターの場合は 255 です。 オプションで、vPC トランクにフェールオーバーする時点を決定するために vPC が使用するしきい値の上限と下限を設定します。*lower-value* の範囲は 1 ～ 255 です。デフォルトは 1 です。*upper-value* の範囲は 1 ～ 255 です。デフォルトは 255 です。

ステップ 6	**no shutdown**
	
	例:
	`switch(config-if-vrrp)# no shutdown`
	`switch(config-if-vrrp)#`
		VRRP グループをイネーブルにします。デフォルトでは、ディセーブルです。

ステップ 7	**show vrrp** ***例:*** `switch(config-if-vrrp)# show vrrp`	（任意）VRRP 情報の要約を表示します。
ステップ 8	**copy running-config startup-config** **例**: `switch(config-if-vrrp)# copy running-config startup-config`	（任意）この設定の変更を保存します。

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。

はじめる前に

ネットワーク上のすべての VRRP デバイスで、認証設定が同じであることを確認します。

VRRP がイネーブルになっていることを確認します（「VRRP の設定」セクション（18-10 ページ）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」セクション（2-9 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp** *number*
4. シャットダウン
5. **authentication text** ***password***
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	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** *interface-type slot/port* **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。
ステップ 3	**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。
ステップ 4	**シャットダウン** **例**: switch(config-if-vrrp)# shutdown switch(config-if-vrrp)#	VRRP グループをディセーブルにします。デフォルトでは、ディセーブルです。
ステップ 5	**authentication text** *password* **例:** switch(config-if-vrrp)# authentication text aPassword	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。
ステップ 6	**no shutdown** **例**: switch(config-if-vrrp)# no shutdown switch(config-if-vrrp)#	VRRP グループをイネーブルにします。デフォルトでは、ディセーブルです。
ステップ 7	**show vrrp** **例:** switch(config-if-vrrp)# show vrrp	(任意) VRRP 情報の要約を表示します。
ステップ 8	**copy running-config startup-config** **例:** switch(config-if-vrrp)# copy running-config startup-config	(任意) この設定の変更を保存します。

アドバタイズメント パケットのタイム インターバル設定

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。

はじめる前に

VRRP をイネーブルにする必要があります(「VRRP の設定」セクション(18-10 ページ)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」セクション(2-9 ページ)を参照)。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp** *number*
4. シャットダウン
5. **advertisement-interval** ***seconds***
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	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** *interface-type slot/port*
	
	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。

ステップ 3	**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。

ステップ 4	**シャットダウン**
	
	例:
	`switch(config-if-vrrp)# shutdown`
	`switch(config-if-vrrp)#`
		VRRP グループをディセーブルにします。デフォルトでは、ディセーブルです。

ステップ 5	**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。範囲は 1 ～ 255 です。デフォルト値は 1 秒です。

ステップ 6	**no shutdown**
	
	例:
	`switch(config-if-vrrp)# no shutdown`
	`switch(config-if-vrrp)#`
		VRRP グループをイネーブルにします。デフォルトでは、ディセーブルです。

ステップ 7	**show vrrp** **例:** `switch(config-if-vrrp)# show vrrp`	（任意）VRRP 情報の要約を表示します。

ステップ 8	**copy running-config startup-config**
	
	例:
	`switch(config-if-vrrp)# copy`
	`running-config startup-config`
		（任意）この設定の変更を保存します。

プリエンプションのディセーブル化

VRRP グループ メンバ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マスター 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イネーブルです。

はじめる前に

VRRP をイネーブルにする必要があります（「VRRP の設定」セクション（18-10 ページ）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」セクション（2-9 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp** *number*
4. シャットダウン
5. **no preempt**
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	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 interface-type slot/port`
	
	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。

ステップ 3	`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。

ステップ 4	`no shutdown`
	
	例:
	`switch(config-if-vrrp)# no shutdown`
		VRRP グループをイネーブルにします。デフォルトでは、ディセーブルです。

ステップ 5	`no preempt`
	
	例:
	`switch(config-if-vrrp)# no preempt`
		プリエンプト 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マスターが変わらないようにします。

	コマンド	目的
ステップ 6	**no shutdown** **例**: switch(config-if-vrrp)# no shutdown	VRRP グループをイネーブルにします。デフォルトでは、ディセーブルです。
ステップ 7	**show vrrp** **例**: switch(config-if-vrrp)# show vrrp	（任意）VRRP 情報の要約を表示します。
ステップ 8	**copy running-config startup-config** **例**: switch(config-if-vrrp)# copy running-config startup-config	（任意）この設定の変更を保存します。

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 ステート トラッキングは、デバイスの別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を変更します。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 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」セクション（18-12 ページ）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。

はじめる前に

VRRP をイネーブルにする必要があります（「VRRP の設定」セクション（18-10 ページ）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」セクション（2-9 ページ）を参照）。

仮想ルータがイネーブルになっていることを確認します（「VRRP グループの設定」セクション（18-11 ページ）を参照）。

手順の概要

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp** *number*
4. シャットダウン
5. **track interface** ***type*** *number* **priority** *value*
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	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** *interface-type slot/port* **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。
ステップ 3	**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。
ステップ 4	**シャットダウン** **例:** switch(config-if-vrrp)# shutdown switch(config-if-vrrp)#	VRRP グループをディセーブルにします。デフォルトでは、ディセーブルです。
ステップ 5	**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 6	**no shutdown** **例:** switch(config-if-vrrp)# no shutdown switch(config-if-vrrp)#	VRRP グループをイネーブルにします。デフォルトでは、ディセーブルです。
ステップ 7	**show vrrp** **例:** switch(config-if-vrrp)# show vrrp	(任意) VRRP 情報の要約を表示します。
ステップ 8	**copy running-config startup-config** **例:** switch(config-if-vrrp)# copy running-config startup-config	(任意) この設定の変更を保存します。

VRRPv3 の設定

この項では、次のトピックについて取り上げます。

VRRPv3 と VRRS のイネーブル化

VRRPv3 グループを設定してイネーブルにするには、その前に VRRPv3 をグローバルでイネーブルにする必要があります。

手順の概要

1. **configure terminal**
2. **[no] feature vrrpv3**
3. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	`[no] feature vrrpv3`
	
	例:
	`switch(config)# feature vrrpv3`
		VRRP バージョン 3 と仮想ルータ冗長サービス（VRRS）をイネーブルにします。このコマンドの **no** 形式は、VRRPv3 と VRRS をディセーブルにします。
		
		VRRPv2 が現在設定されている場合は、グローバル コンフィギュレーション モードで **no feature vrrp** コマンドを使用して VRRPv2 設定を削除し、その後 **feature vrrpv3** コマンドを使用して VRRPv3 を有効にします。

ステップ 3	`copy running-config startup-config`
	
	例:
	`switch(config)# copy running-config startup-config`
		（任意）この設定の変更を保存します。

VRRPv3 グループの作成

VRRPv3 グループを作成し、仮想 IP アドレスを割り当て、グループをイネーブルにすることができます。

はじめる前に

VRRPv3 がイネーブルになっていることを確認します。

インターフェイスで IP アドレスを設定したことを確認します。

手順の概要

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **vrrpv3** *number* **address-family [ipv4 | ipv6]**
4. （任意）**address** *ip-address* **[primary | secondary]**

5. （任意）**description** *description*
6. （任意）**match-address**
7. （任意）**preempt [delay minimum** *seconds***]**
8. （任意）**priority** *level*
9. （任意）**timers advertise** *interval*
10. （任意）**vrrp2**
11. （任意）**vrrs leader** *vrrs-leader-name*
12. （任意）**shutdown**
13. （任意）**show fhrp [***interface-type interface-number***] [verbose]**
14. （任意）**show vrrpv3** *interface-type interface-number*
15. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 ethernet** *slot/port*
	
	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。

ステップ 3	**vrrpv3** *number address-family* [**ipv4** | **ipv6**]
	
	例:
	`switch(config-if)# vrrpv3 5 address-family ipv4`
	`switch(config-if-vrrpv3-group)#`
		VRRPv3 グループを作成し、VRRPv3 グループ コンフィギュレーション モードを開始します。範囲は 1 ～ 255 です。

ステップ 4	**address** *ip-address* [**primary** | **secondary**] **例**: `switch(config-if-vrrpv3-group)# address 100.0.1.10 primary`	(任意) VRRPv3 グループにプライマリまたはセカンダリ IPv4 アドレスまたは IPv6 アドレスを指定します。 VRRPv3 グループでセカンダリ IP アドレスを使用するには、まず同じグループでプライマリ IP アドレスを設定する必要があります。
ステップ 5	**description** *description* **例**: `switch(config-if-vrrpv3-group)# description group3`	(任意) VRRPv3 グループの説明を指定します。最大 80 文字の英数字を入力できます。
ステップ 6	**match-address** **例**: `switch(config-if-vrrpv3-group)# match-address`	(任意) アドバタイズメント パケットのセカンダリ アドレスを設定したアドレスと照合します。

	コマンド	目的
ステップ 7	**preempt** [**delay minimum** *seconds*] **例:** `switch(config-if-vrrpv3-group)# preempt delay minimum 30`	(任意)プライオリティの低いマスター スイッチのプリエンプションをオプションの延期期間でイネーブルにします。範囲は 0 ～ 3600 です。
ステップ 8	**priority** *level* **例:** `switch(config-if-vrrpv3-group)# priority 3`	(任意)VRRPv3 グループのプライオリティを指定します。有効な範囲は 1 ～ 254 です。
ステップ 9	**timers advertise** *interval* **例:** `switch(config-if-vrrpv3-group)# timers advertise 1000`	(任意)アドバタイズメント タイマーをミリ秒で設定します。範囲は 100 ～ 40950 です。 シスコは、このタイマーを 1 秒以上の値に設定することを推奨します。
ステップ 10	**vrrp2** **例:** `switch(config-if-vrrpv3-group)# vrrp2`	(任意)VRRPv2 のみをサポートするデバイスとの相互運用性を確保するために、VRRPv2 へのサポートも同時にイネーブルにします。 VRRPv2 互換モードは、VRRPv2 から VRRPv3 にアップグレードするために提供されます。これは完全な VRRPv2 実装ではないので、アップグレードを実行する場合にのみ使用してください。
ステップ 11	**vrrs leader** *vrrs-leader-name* **例:** `switch(config-if-vrrpv3-group)# vrrs leader leader1`	(任意)VRRS に登録されるリーダーの名前を指定します。
ステップ 12	**シャットダウン** **例:** `switch(config-if-vrrpv3-group)# shutdown`	(任意)VRRPv3 グループの VRRP 設定をディセーブルにします。
ステップ 13	**show fhrp** [*interface-type interface-number*] [**verbose**] **例:** `switch(config-if-vrrpv3-group)# show fhrp ethernet 2/1 verbose`	(任意)ファースト ホップ冗長性プロトコル(FHRP)の情報を表示します。 詳細情報を表示するには、**verbose** キーワードを使用します。
ステップ 14	**show vrrpv3** *interface-type interface-number* **例:** `switch(config-if-vrrpv3-group)# show vrrpv3 ethernet 2/1`	(任意)指定したインターフェイスの VRRPv3 設定情報を表示します。
ステップ 15	**copy running-config startup-config** **例:** `switch(config-if-vrrpv3-group)# copy running-config startup-config`	(任意)この設定の変更を保存します。

VRRPv3 制御グループの設定

VRRPv3 制御グループを設定できます。

はじめる前に

VRRPv3 がイネーブルになっていることを確認します。

インターフェイスで IP アドレスを設定したことを確認します。

手順の概要

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **ip address** *ip-address mask* **[secondary]**
4. **vrrpv3** *number* **address-family [ipv4 | ipv6]**
5. （任意）**address** *ip-address* **[primary | secondary]**
6. （任意）**shutdown**
7. （任意）**show fhrp [***interface-type interface-number***] [verbose]**
8. （任意）**show vrrpv3** *interface-type interface-number*
9. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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 ethernet** *slot/port*
	
	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。

ステップ 3	**ip address** *ip-address mask* [**secondary**] **例**: `switch(config-if)# ip address 209.165.200.230 255.255.255.224`	インターフェイスの IP アドレスを設定します。 **secondary** キーワードを使用して、インターフェイスで追加の IP アドレスを設定できます。

ステップ 4	**vrrpv3** *number* **address-family** [**ipv4** | **ipv6**]
	
	例:
	`switch(config-if)# vrrpv3 5 address-family ipv4`
	`switch(config-if-vrrpv3-group)#`
		VRRPv3 グループを作成し、VRRPv3 グループ コンフィギュレーション モードを開始します。範囲は 1 ～ 255 です。

ステップ 5	**address** *ip-address* [**primary** | **secondary**] **例**: `switch(config-if-vrrpv3-group)# address 209.165.200.227 primary`	（任意）VRRPv3 グループにプライマリまたはセカンダリ IPv4 アドレスまたは IPv6 アドレスを指定します。

	コマンド	目的
ステップ 6	**シャットダウン** **例**: `switch(config-if-vrrpv3-group)# shutdown`	(任意)VRRPv3 グループの VRRP 設定をディセーブルにします。
ステップ 7	**show fhrp** [*interface-type interface-number*] [**verbose**] **例**: `switch(config-if-vrrpv3-group)# show fhrp ethernet 2/1 verbose`	(任意)ファースト ホップ冗長性プロトコル(FHRP)の情報を表示します。詳細情報を表示するには、**verbose** キーワードを使用します。
ステップ 8	**show vrrpv3** *interface-type interface-number* **例**: `switch(config-if-vrrpv3-group)# show vrrpv3 ethernet 2/1`	(任意)指定したインターフェイスの VRRPv3 設定情報を表示します。
ステップ 9	**copy running-config startup-config** **例**: `switch(config-if-vrrpv3-group)# copy running-config startup-config`	(任意)この設定の変更を保存します。

VRRS 経路の設定

仮想ルータ冗長サービス(VRRS)の経路を設定できます。拡張環境では、VRRS 経路は VRRPv3 制御グループと組み合わせて使用する必要があります。

はじめる前に

VRRPv3 がイネーブルになっていることを確認します。

インターフェイスで IP アドレスを設定したことを確認します。

手順の概要

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **ip address** *ip-address mask* **[secondary]**
4. **vrrs pathway** *vrrs-tag*
5. **mac address {***mac-address* **| inherit}**
6. **address** *ip-address*
7. (任意)**show vrrs pathway** *interface-type interface-number*
8. (任意)**copy running-config startup-config**

手順の詳細

	コマンド	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 ethernet** *slot/port* **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。
ステップ 3	**ip address** *ip-address mask* [**secondary**] **例**: switch(config-if)# ip address 209.165.200.230 255.255.255.224	インターフェイスの IP アドレスを設定します。 **secondary** キーワードを使用して、インターフェイスで追加の IP アドレスを設定できます。
ステップ 4	**vrrs pathway** *vrrs-tag* **例**: switch(config-if)# vrrs pathway path1 switch(config-if-vrrs-pw)#	VRRS グループの VRRS 経路を定義し、VRRS 経路コンフィギュレーション モードを開始します。 *vrrs-tag* 引数は、経路に関連付けられている VRRS タグの名前を指定します。
ステップ 5	**mac address** {*mac-address* | **inherit**} **例**: switch(config-if-vrrs-pw)# mac address fe24.fe24.fe24	経路の MAC アドレスを指定します。 **inherit** キーワードを使用すると、経路は関連付けられている VRRPv3 グループの仮想 MAC アドレスを継承します。
ステップ 6	**address** *ip-address* **例**: switch(config-if-vrrs-pw)# address 209.165.201.10	経路の仮想 IPv4 アドレスまたは IPv6 アドレスを定義します。 VRRPv3 グループは、複数の経路を制御できます。
ステップ 7	**show vrrs pathway** *interface-type interface-number* **例**: switch(config-if-vrrs-pw)# show vrrs pathway ethernet 1/2	(任意) 異なる経路の状態 (アクティブ、非アクティブ、非対応など) に関する VRRS 経路の情報を表示します。
ステップ 8	**copy running-config startup-config** **例**: switch(config-if-vrrs-pw)# copy running-config startup-config	(任意) この設定の変更を保存します。

VRRP の設定確認

VRRP 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 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 fhrp [*interface-type interface-number*] [**verbose**]	ファースト ホップ冗長性プロトコル（FHRP）の情報を表示します。
show interface *interface-type*	インターフェイスの仮想ルータ設定を表示します。

VRRPv3 設定の確認

VRRPv3 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 vrrpv3 [**all** | **brief** | **detail**]	VRRPv3 設定情報を表示します。
show vrrpv3 *interface-type interface-number*	特定のインターフェイスの VRRPv3 設定情報を表示します。
show vrrs client [*client-name*]	VRRS クライアント情報を表示します。
show vrrs pathway [*interface-type interface-number*]	異なる経路の状態（アクティブ、非アクティブ、非対応など）に関する VRRS 経路の情報を表示します。
show vrrs server	VRRS サーバ情報を表示します。
show vrrs tag [*tag-name*]	VRRS タグ情報を表示します。

VRRP 統計情報のモニタリングとクリア

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	VRRP の統計情報を表示します。

デバイスのすべてのインターフェイスについて、すべての VRRP 統計情報を消去するには、**clear vrrp statistics** コマンドを使用します。

VRRPv3 統計情報のモニタリングとクリア

VRRPv3 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 vrrpv3 statistics	VRRPv3 の統計情報を表示します。

デバイスのすべてのインターフェイスについて、すべての VRRPv3 統計情報を消去するには、**clear vrrpv3 statistics** コマンドを使用します。

VRRP の設定例

この例では、ルータ A およびルータ B はそれぞれ 3 つの VRRP グループに所属しています。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 はプライオリティ 120 で、このグループのマスターになります。
  - アドバタイズ インターバルは 3 秒です。
  - プリエンプションはイネーブルです。
- グループ 5:
  - ルータ B はプライオリティ 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、このグループのマスターになります。
  - アドバタイズ インターバルはデフォルトの 1 秒です。
  - プリエンプションはディセーブルです。

ルータ A

```
interface ethernet 1/0
   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
 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
```

VRRPv3 の設定例

次に、VRRPv3 をイネーブルにし VRRPv3 グループを作成およびカスタマイズする例を示します。

```
switch# configure terminal
switch(config)# feature vrrpv3
switch(config)# interface ethernet 4/6
switch(config-if)# vrrpv3 5 address-family ipv4
switch(config-if-vrrp3-group)# address 209.165.200.225 primary
switch(config-if-vrrp3-group)# description group3
switch(config-if-vrrp3-group)# match-address
switch(config-if-vrrp3-group)# preempt delay minimum 30
switch(config-if-vrrpv3-group)# show fhrp ethernet 4/6 verbose
switch(config-if-vrrpv3-group)# show vrrpv3 ethernet 4/6
```

次に、VRRPv3 制御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)# ip address 209.165.200.230 255.255.255.224
switch(config-if)# vrrpv3 5 address-family ipv4
switch(config-if-vrrpv3-group)# address 209.165.200.227 primary
switch(config-if-vrrpv3-group)# vrrs leader leader1
switch(config-if-vrrpv3-group)# shutdown
switch(config-if-vrrpv3-group)# show fhrp ethernet 1/2 verbose
switch(config-if-vrrpv3-group)# show vrrpv3 ethernet 1/2
```

次に、VRRS 経路を設定する例を示します。

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip address 209.165.200.230 255.255.255.224
switch(config-if)# vrrs pathway path1
switch(config-if-vrrs-pw)# mac address inherit
switch(config-if-vrrs-pw)# address 209.165.201.10
switch(config-if-vrrs-pw)# show vrrs pathway ethernet 1/2
```

その他の関連資料

関連資料

関連項目	マニュアル タイトル
Hot Standby Router Protocol（HSRP）の設定	第 17 章「HSRP の設定」
ハイ アベイラビリティの設定	*『Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide』*

CHAPTER 19

オブジェクト トラッキングの設定

この章では、Cisco NX-OS デバイス上でオブジェクト 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 トラッキング情報

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、デバイス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。

この項では、次のトピックについて取り上げます。

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。

- Embedded Event Manager（EEM）
- ホットスタンバイ冗長プロトコル（HSRP）
- 仮想ポート チャネル（vPC）
- 仮想ルータ冗長プロトコル（VRRP）

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 または IPv6 アドレスが設定されていて、IPv4 または IPv6 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 または IPv6 ルートが存在していて、ローカル デバイス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。リンク プロトコルがダウンした場合、影響を受ける HSRP ルータのプライオリティを変更し、よりすぐれたネットワーク接続が得られるバックアップ ルータにスイッチオーバーされるようにできます。

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。

他のエンティティ（たとえば、仮想ポート チャネル（vPC））は、オブジェクト トラッキング リストを使用することにより、vPC を作成する複数のピア リンクのステートに基づいて vPC のステートを変更できます。vPC の詳細については、『*Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*』を参照してください。

トラック リストの詳細については、「ブール式を使用したオブジェクト トラッキング リストの設定」セクション（19-7 ページ）を参照してください。

ハイ アベイラビリティ

オブジェクト トラッキングは、ステートフル リスタートを通じてハイ アベイラビリティをサポートします。ステートフル リスタートが実行されるのは、オブジェクト トラッキング プロセスがクラッシュした場合です。オブジェクト トラッキングは、デュアル スーパーバイザ システムでのステートフル スイッチオーバーもサポートします。Cisco NX-OS スイッチオーバー後に実行コンフィギュレーションを適用します。

オブジェクト トラッキングを使用して、ネットワーク全体の可用性が向上するように、クライアントの動作を変更することもできます。

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。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、その VRF のメンバとしてオブジェクトを設定する必要があります（「非デフォルト VRF のオブジェクト トラッキング設定」セクション（19-14 ページ）を参照）。

オブジェクト トラッキングのライセンス要件

次の表に、この機能のライセンス要件を示します。

製品	ライセンス要件
Cisco NX-OS	オブジェクト トラッキングにライセンスは不要です。ライセンス パッケージに含まれていない機能は nx-os イメージにバンドルされており、無料で提供されます。Cisco NX-OS のライセンス スキームの詳細は、『*Cisco NX-OS Licensing Guide*』を参照してください。

注意事項および制約事項

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。

- イーサネット、サブインターフェイス、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。

デフォルト設定値

表 19-1 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 19-1　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

この項では、次のトピックについて取り上げます。

（注）　IP SLA PBR オブジェクト トラッキングの設定の詳細については、『Cisco Nexus 9000 Series NX-OS IP SLAs Configuration Guide』を参照してください。

インターフェイスのオブジェクト トラッキング設定

インターフェイスのライン プロトコルまたは IPv4 や IPv6 ルーティングのステー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* **interface** *interface-type number* {**ip** | **ipv6** | **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	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	**track** *object-id* **interface** *interface-type number* {**ip routing** | **ipv6 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	**show track** [*object-id*] **例**: `switch(config-track)# show track 1`	(任意)オブジェクト トラッキング情報を表示します。
ステップ 4	**copy running-config startup-config** **例**: `switch(config-track)# copy running-config startup-config`	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

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
```

Ethernet 1/2 上で IPv6 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 3 interface ethernet 1/2 ipv6 routing
switch(config-track)# copy running-config startup-config
```

トラッキング対象オブジェクトの削除

トラッキング対象オブジェクトを削除できます。

手順の概要

1. **configure terminal**
2. **no track** *object-id*

手順の詳細

	コマンド	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	**no track** *object-id* **例**: switch(config)# no track 1 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削除します。*object-id* の範囲は 1 ～ 500 です。

次に、トラッキング対象オブジェクトを削除する例を示します。

```
switch# configure terminal
switch(config)# no track 1
switch(config-track)# copy running-config startup-config
```

ルート到達可能性のオブジェクト トラッキング設定

IP ルートまたは IPv6 ルートの存在および到達可能性を追跡するように Cisco NX-OS を設定できます。

手順の概要

1. **configure terminal**
2. **track** *object-id* {**ip** | **ipv6**} **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	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	**track** *object-id* {**ip** | **ipv6**} **route** *prefix/length* **reachability** **例**: switch(config)# track 3 ipv6 route 2::5/64 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。IPv4 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。IPv6 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 ～ 128 です。

	コマンド	目的
ステップ 3	**show track** [*object-id*] **例**: switch(config-track)# show track 1	(任意)オブジェクト トラッキング情報を表示します。
ステップ 4	**copy running-config startup-config** **例**: switch(config-track)# copy running-config startup-config	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

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
```

次に、デフォルト VRF で IPv6 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 5 ipv6 route 10::10/128 reachability
switch(config-track)# copy running-config startup-config
```

ブール式を使用したオブジェクト 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** [*object-id*]
5. (任意)**copy running-config startup-config**

手順の詳細

	コマンド	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。

	コマンド	目的
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 コンフィギュレーション 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の場合にリストがアップ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。 *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	**show track** [*object-id*] **例**: switch(config-track)# show track	(任意)オブジェクト トラッキング情報を表示します。
ステップ 5	**copy running-config startup-config** **例**: switch(config-track)# copy running-config startup-config	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

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使用したオブジェクト 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、トラッキング対象リストに 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、トラッキング 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. **object** *object-id*
5. （任意）**show track** [*object-id*]
6. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	**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 コンフィギュレーション 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テージを設定します。指定できる範囲は 0 ～ 100% です。
ステップ 4	**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。

	コマンド	目的
ステップ 5	**show track** [*object-id*] **例**: switch(config-track)# show track	(任意)オブジェクト トラッキング情報を表示します。
ステップ 6	**copy running-config startup-config** **例**: switch(config-track)# copy running-config startup-config	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

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
```

重みしきい値を使用したオブジェクト 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** [*object-id*]
6. (任意)**copy running-config startup-config**

手順の詳細

	コマンド	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。

	コマンド	目的

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 コンフィギュレーション 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**: `switch(config-track)# object 10 weight 15`	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。*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	**show track** [*object-id*] **例**: `switch(config-track)# show track`	(任意)オブジェクト トラッキング情報を表示します。
ステップ 6	**copy running-config startup-config** **例**: `switch(config-track)# copy running-config startup-config`	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

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** [*object-id*]
6. （任意）**copy running-config startup-config**

手順の詳細

	コマンド	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	**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。IPv4 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。IPv6 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 ～ 128 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 コンフィギュレーション 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。

	コマンド	目的
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。
ステップ 5	**show track** [*object-id*] **例**: switch(config-track)# show track 3	(任意)オブジェクト トラッキング情報を表示します。
ステップ 6	**copy running-config startup-config** **例**: switch(config-track)# copy running-config startup-config	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

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される前後の **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 のオブジェクト トラッキング設定

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。

はじめる前に

デフォルト以外の VRF が最初に作成されることを確認します。

手順の概要

1. **configure terminal**
2. **track** *object-id* {**ip** | **ipv6**} **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	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	**track** *object-id* {**ip** | **ipv6**} **route** *prefix/length* **reachability**
	
	例:
	`switch(config)# track 3 ipv6 route 1::2/64 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。IPv4 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。IPv6 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 ～ 128 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	**show track** [*object-id*] **例**: `switch(config-track)# show track 3`	(任意)オブジェクト トラッキング情報を表示します。
ステップ 5	**copy running-config startup-config** **例**: `switch(config-track)# copy running-config startup-config`	(任意)実行コンフィギュレーションをスタートアップ コンフィギュレーションにコピーします。

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、IPv6 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 3 ipv6 route 1::2/64 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** | **ipv6**} **route** [**brief**]	IPv4 または IPv6 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

次に、ルート到達可能性のオブジェクト トラッキングを設定し、VRF Red を使用してそのルー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
```

関連項目

オブジェクト トラッキングの関連情報については、次の項目を参照してください。

- 第 13 章「レイヤ 3 仮想化の設定」
- 第 17 章「HSRP の設定」

その他の関連資料

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（19-16 ページ）

関連資料

関連項目	マニュアル タイトル
Embedded Event Manager の設定	*『Cisco Nexus 9000 Series NX-OS System Management Configuration Guide』*
IP SLA オブジェクト トラッキングの設定	*『Cisco Nexus 9000 Series NX-OS IP SLAs Configuration Guide』*

Cisco NX-OS ユニキャスト機能でサポートされている IETF RFC

この付録では、Cisco NX-OS でサポートされる IETF RFC を示します。

BGP の RFC

RFC	タイトル
RFC 1997	*『BGP Communities Attribute』*
RFC 2385	*『Protection of BGP Sessions via the TCP MD5 Signature Option』*
RFC 2439	*『BGP Route Flap Damping』*
RFC 2519	*『A Framework for Inter-Domain Route Aggregation』*
RFC 2545	*『Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing』*
RFC 2858	*『Multiprotocol Extensions for BGP-4』*
RFC 2918	*『Route Refresh Capability for BGP-4』*
RFC 3065	*『Autonomous System Confederations for BGP』*
RFC 3392	*『Capabilities Advertisement with BGP-4』*
RFC 4271	*『A Border Gateway Protocol 4 (BGP-4)』*
RFC 4273	*『Definitions of Managed Objects for BGP-4』*
RFC 4456	*『BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)』*
RFC 4486	*『Subcodes for BGP Cease Notification Message』*
RFC 4724	*『Graceful Restart Mechanism for BGP』*
RFC 4893	*『BGP Support for Four-octet AS Number Space』*
RFC 5004	*『Avoid BGP Best Path Transitions from One External to Another』*
RFC 5396[1]	*『Textual Representation of Autonomous System (AS) Numbers』*
RFC 5549	*『Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop』*
RFC 5668	*『4-Octet AS Specific BGP Extended Community』*
draft-ietf-idr-add-paths-08.txt	*『Advertisement of Multiple Paths in BGP』*

RFC	タイトル
draft-ietf-idr-bgp4-mib-15.txt	*BGP4-MIB*
draft-kato-bgp-ipv6-link-local-00.txt	*『BGP4+ Peering Using IPv6 Link-local Address』*

1. RFC 5396 は部分的にサポートされます。asplain と asdot 表記はサポートされますが、asdot+ 表記はサポートされません。

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』*
RFC 4022	*TCP-MIB*
RFC 4292	*『IP-FORWARDING-TABLE-MIB』*
RFC 4293	*IP-MIB*

IPv6 の RFC

RFC	タイトル
RFC 1981	*『Path MTU Discovery for IP version 6』*
RFC 2373	*『IP Version 6 Addressing Architecture』*
RFC 2374	*『An Aggregatable Global Unicast Address Format』*
RFC 2460	*『Internet Protocol, Version 6 (IPv6) Specification』*
RFC 2461	*『Neighbor Discovery for IP Version 6 (IPv6)』*
RFC 2462	*『IPv6 Stateless Address Autoconfiguration』*
RFC 2464	*『Transmission of IPv6 Packets over Ethernet Networks』*
RFC 3152	*『Delegation of IP6.ARPA』*
RFC 3162	*『RADIUS and IPv6』*

RFC	タイトル
RFC 3513	*『Internet Protocol Version 6 (IPv6) Addressing Architecture』*
RFC 3596	*『DNS Extensions to Support IP version 6』*
RFC 4193	*『Unique Local IPv6 Unicast Addresses』*

IS-IS の RFC

RFC	タイトル
RFC 1142	*『OSI 10589 Intermediate system to intermediate system intra-domain routing exchange protocol』*
RFC 1195	*『Use of OSI IS-IS for routng in TCP/IP and dual environment』*
RFC 2763、RFC 5301	*『Dynamic Hostname Exchange Mechanism for IS-IS』*
RFC 2966、RFC 5302	*『Domain-wide Prefix Distribution with Two-Level IS-IS』*
RFC 2972	*『IS-IS Mesh Groups』*
RFC 3277	*『IS-IS Transient Blackhole Avoidance』*
RFC 3373、RFC 5303	*『Three-Way Handshake for IS-IS Point-to-Point Adjacencies』*
RFC 3567、RFC 5304	*『IS-IS Cryptographic Authentication』*
RFC 3784、RFC 5305	*『IS-IS Extensions for Traffic Engineering』*
RFC 3847、RFC 5306	*『Restart Signaling for IS-IS』*
draft-ietf-isis-igp-p2p-over-lan-06.txt	*『Internet Draft Point-to-point operation over LAN in link-state routing protocols』*

OSPF の RFC

RFC	タイトル
RFC 2328	*『OSPF Version 2』*
RFC 2370	*『The OSPF Opaque LSA Option』*
RFC 2740	*『OSPF for IPv6』*
RFC 3101	*『The OSPF Not-So-Stubby Area (NSSA) Option』*
RFC 3137	*『OSPF Stub Router Advertisement』*
RFC 3623	*『Graceful OSPF Restart』*
RFC 4552（部分的なサポート）	*『Authentication/Confidentiality for OSPFv3』*
RFC 5709	*OSPFv2 HMAC-SHA 暗号化認証*
draft-ietf-ospf-ospfv3-graceful-restart-04.txt	*『OSPFv3 Graceful Restart』*

RIP の RFC

RFC	タイトル
RFC 2082	*『RIP-2 MD5 Authentication』*
RFC 2453	*『RIP Version 2』*

Cisco NX-OS レイヤ 3 ユニキャスト機能の設定の上限

設定の制限は、『*Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*』に記載されています。