IP ユニキャスト ルーティングの設定

機能情報の確認

ご使用のソフトウェア リリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、使用するプラットフォームおよびソフトウェア リリースの Bug Search Tool およびリリース ノートを参照してください。このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表を参照してください。

プラットフォームのサポートおよびシスコ ソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator には、http://www.cisco.com/go/cfn からアクセスします。Cisco.com のアカウントは必要ありません。

IP ユニキャスト ルーティングの設定に関する情報

このモジュールでは、スイッチで IP Version 4（IPv4）ユニキャスト ルーティングを設定する方法について説明します。

（注） LAN ベース フィーチャを実行しているスイッチでは、VLAN でのスタティック ルーティングのみがこのリリースでサポートされます。

スイッチ スタックは、ネットワーク内のそれ以外のルータに対して、単一のルータとして動作し、認識されます。スタティック ルーティング、Routing Information Protocol（RIP）などの基本的なルーティング機能は、 IP Base フィーチャ セットおよび IP Services フィーチャ セットの両方拡張ルーティング機能およびその他のルーティング プロトコルを使用するには、スタンドアロン スイッチやアクティブ スイッチで IP サービス フィーチャ セットをイネーブルにする必要があります。

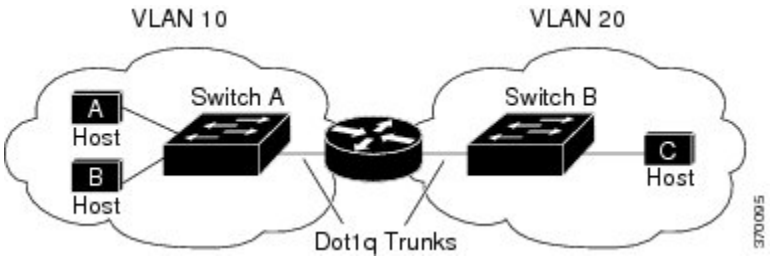
（注） IPv4 トラフィックに加えて、スイッチまたはスイッチ スタックが IP ベースまたは IP サービス フィーチャ セットを実行している場合IP バージョン 6（IPv6）ユニキャスト ルーティングをイネーブルにし、IPv6 トラフィックを転送するようにインターフェイスを設定できます。

IP ルーティングに関する情報

一部のネットワーク環境で、VLAN（仮想 LAN）は各ネットワークまたはサブネットワークに関連付けられています。IP ネットワークで、各サブネットワークは 1 つの VLAN に対応しています。VLAN を設定すると、ブロードキャスト ドメインのサイズを制御し、ローカル トラフィックをローカル内にとどめることができます。ただし、異なる VLAN 内のネットワーク デバイスが相互に通信するには、VLAN 間でトラフィックをルーティング（VLAN 間ルーティング）するレイヤ 3 デバイス（ルータ）が必要です。VLAN 間ルーティングでは、適切な宛先 VLAN にトラフィックをルーティングするため、1 つまたは複数のルータを設定します。

次の図に基本的なルーティング トポロジを示します。スイッチ A は VLAN 10 内、スイッチ B は VLAN 20 内にあります。ルータには各 VLAN のインターフェイスが備わっています。

図 *1*：ルーティング トポロジの例



VLAN 10 内のホスト A が VLAN 10 内のホスト B と通信する場合、ホスト A はホスト B 宛にアドレス指定されたパケットを送信します。スイッチ A はパケットをルータに送信せず、ホスト B に直接転送します。

ホスト A から VLAN 20 内のホスト C にパケットを送信する場合、スイッチ A はパケットをルータに転送し、ルータは VLAN 10 インターフェイスでトラフィックを受信します。ルータはルーティング テーブルを調べて正しい発信インターフェイスを判別し、VLAN 20 インターフェイスを経由してパケットをスイッチ B に送信します。スイッチ B はパケットを受信し、ホスト C に転送します。

ルーティング タイプ

ルータおよびレイヤ 3 スイッチは、次の方法でパケットをルーティングできます。

- デフォルト ルーティング

- 事前にプログラミングされているトラフィックのスタティック ルートの使用
- ルーティング プロトコルによるルートの動的な計算

デフォルト ルーティングとは、宛先がルータにとって不明であるトラフィックをデフォルトの出口または宛先に送信することです。

スタティック ユニキャスト ルーティングの場合、パケットは事前に設定されたポートから単一のパスを通り、ネットワークの内部または外部に転送されます。スタティック ルーティングは安全で、帯域幅をほとんど使用しません。ただし、リンク障害などのネットワークの変更には自動的に対応しないため、パケットが宛先に到達しないことがあります。ネットワークが拡大するにつれ、スタティック ルーティングの設定は煩雑になります。

LAN ベース フィーチャ セットを実行しているスイッチは、管理インターフェイスで使用するデフォルト ルートに加えて、ユーザが設定した 16 のスタティック ルートをサポートしています。LAN ベース イメージは、SVI でのみスタティック ルーティングをサポートしています。

ルータでは、トラフィックを転送する最適ルートを動的に計算するため、ダイナミック ルーティング プロトコルが使用されます。ダイナミック ルーティング プロトコルには次の 2 つのタイプがあります。

- ディスタンスベクトル プロトコルを使用するルータでは、ネットワーク リソースの距離の値を使用してルーティング テーブルを保持し、これらのテーブルをネイバーに定期的に渡します。ディスタンスベクトル プロトコルは 1 つまたは複数のメトリックを使用し、最適なルートを計算します。これらのプロトコルは、簡単に設定、使用できます。
- リンクステート プロトコルを使用するルータでは、ルータ間のリンクステート アドバタイズメント（LSA）の交換に基づき、ネットワーク トポロジに関する複雑なデータベースを保持します。LSA はネットワークのイベントによって起動され、コンバージェンス時間、またはこれらの変更への対応時間を短縮します。リンクステート プロトコルはトポロジの変更にすばやく対応しますが、ディスタンスベクトル プロトコルよりも多くの帯域幅およびリソースが必要になります。

スイッチでサポートされているディスタンスベクトル プロトコルは、Routing Information Protocol（RIP）および Border Gateway Protocol（BGP）です。RIP は最適パスを決定するために単一の距離メトリック（コスト）を使用し、BGP はパス ベクトル メカニズムを追加します。また、Open Shortest Path First（OSPF）リンクステート プロトコル、および従来の Interior Gateway Routing Protocol（IGRP）にリンクステート ルーティング機能の一部を追加して効率化を図った Enhanced IGRP（EIGRP）もサポートされています。

（注） スイッチまたはスイッチ スタックでサポートされるプロトコルは、アクティブ スイッチ上で稼働しているソフトウェアによって決まります。アクティブ スイッチ上で IP ベース フィーチャ セットが稼働している場合は、デフォルトのルーティング、スタティック ルーティング、および RIP だけがサポートされます。スイッチで LAN ベース フィーチャ セットが稼働している場合、SVI では 16 のスタティック ルートを設定できます。その他のすべてのルーティング プロトコルには、IP サービス フィーチャ セットが必要です。

IP ルーティングおよびスイッチ スタック

スタックのスイッチがルーティング ピアに接続されているかどうかに関係なく、スイッチ スタックはネットワークからは単一のスイッチとして認識されます。

アクティブ スイッチにより、次の機能が実行されます。

- ルーティング プロトコルを初期化し、設定します。
- ルーティング プロトコル メッセージおよびアップデートを他のルータに送信します。
- ピア ルータから受信したルーティング プロトコル メッセージおよびアップデートを処理します。
- distributed Cisco Express Forwarding（dCEF）データベースを生成および維持し、すべてのスタック メンバーに配信します。このデータベースに基づいて、スタック内のすべてのスイッチにルートがプログラミングされます。
- アクティブ スイッチの MAC アドレスはスタック全体のルータ MAC アドレスとして使用され、すべての外部デバイスはこのアドレスを使用して IP パケットをスタックに送信します。
- ソフトウェア転送またはソフトウェア処理を必要とするすべての IP パケットは、アクティブ スイッチの CPU を通ります。

スタック メンバーは、次に示す機能を実行します。

- ルーティング スタンバイ スイッチとして機能します。アクティブ スイッチに障害が発生し、新規アクティブ スイッチとして選択された場合に、処理を引き継ぐことができます。
- ルートをハードウェアにプログラムします。

アクティブ スイッチに障害が発生すると、スタックはアクティブ スイッチがダウンしていることを検出し、スタック メンバの 1 つを新規アクティブ スイッチとして選択します。この期間中に、ハードウェアは一時的な中断を除き、アクティブなプロトコルがない状態でパケットの転送を継続します。

ただし、スイッチ スタックが障害のあとハードウェア ID を維持していても、アクティブ スイッチの再起動前の短い中断の間にルータ ネイバーのルーティング プロトコルがフラップすることがあります。OSPF や EIGRP などのルーティング プロトコルは、ネイバーの移行を認識する必要があります。ルータは、次の 2 つのレベルの Nonstop Forwarding（NSF）を使用して、スイッチオーバーの検出、ネットワーク トラフィックの転送の継続、およびピアデバイスから情報の回復を行います。

- NFS 認識ルータによるネイバー ルータ障害の許容。ネイバー ルータの再起動後、NFS 認識ルータは要求を受けて自身のステート情報とルートの隣接情報を提供します。
- NFS 対応ルータによる NSF のサポート。NSF 対応ルータは、アクティブ スイッチの変更を検出した場合、NSF 認識ネイバーまたは NSF 対応ネイバーからの情報でルーティング情報を再構築します。再起動を待つことはしません。

スイッチ スタックは NSF 対応ルーティングを OSPF および EIGRP に対してサポートします。

新規アクティブ スイッチは、選択されたときに次の機能を実行します。

- ルーティング アップデートの生成、受信、および処理を開始します。
- ルーティング テーブルを構築し、CEF データベースを生成して、スタック メンバーに配信します。
- ルータ MAC アドレスとして自身の MAC アドレスを使用します。新規 MAC アドレスのネットワーク ピアに通知するために、新規ルータ MAC アドレスを使用して余分の ARP 応答を定期的に（5 分間の間、数秒おきに）送信します。

（注） 固定 MAC アドレス機能をスタックに設定していて、アクティブ スイッチに変更があった場合、設定された時間スタック MAC アドレスは変更されません。この期間に前のアクティブ スイッチがメンバ スイッチとしてスタックに再加入する場合、スタック MAC アドレスは前のアクティブ スイッチの MAC アドレスのままになります。

- ARP 要求をプロキシ ARP IP アドレスに送信し、ARP 応答を受信して、各プロキシ ARP エントリの到達可能性を判別しようとします。到達可能なプロキシ ARP IP アドレスごとに、新規ルータ MAC アドレスを使用して gratuitous ARP 応答を生成します。このプロセスは、新規アクティブ スイッチが選択されたあと、5 分間繰り返されます。

（注） アクティブ スイッチが IP サービス フィーチャ セットを実行している場合は、スタックは、Open Shortest Path First（OSPF）、Enhanced IGRP（EIGRP）、およびボーダー ゲートウェイ プロトコル（BGP）を含む、サポートされるすべてのプロトコルを実行できます。アクティブ スイッチに障害が発生し、新規に選択されたアクティブ スイッチ上で IP ベースまたは LAN ベース フィーチャ セットが稼働している場合、これらのプロトコルはスタック内で稼働しなくなります。

注意 スイッチ スタックを複数のスタックに分割すると、ネットワークが適切に動作しなくなる場合があります。

スイッチがリロードされると、NSF/SSO 機能である場合でも、そのスイッチのポートがすべてダウンし、ルーティングに関わるインターフェイスにトラフィックの損失が発生します。

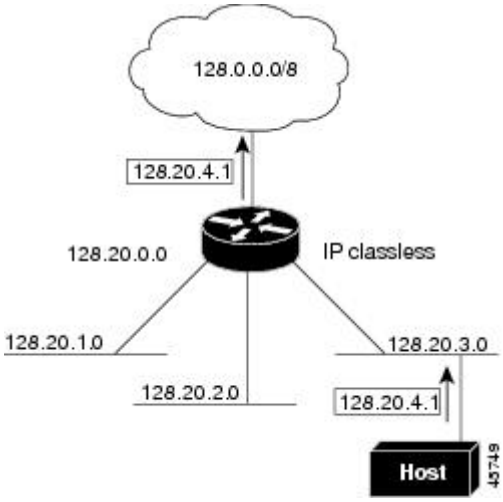
クラスレス ルーティング

ルーティングを行うように設定されたスイッチで、クラスレス ルーティング動作はデフォルトでイネーブルとなっています。クラスレスルーティングがイネーブルの場合、デフォルトルートがないネットワークのサブネット宛てパケットをルータが受信すると、ルータは最適なスーパーネッ

トルートにパケットを転送します。スーパーネットは、単一の大規模アドレス空間をシミュレートするために使用されるクラスCアドレス空間の連続ブロックで構成されています。スーパーネットは、クラス B アドレス空間の急速な枯渇を回避するために設計されました。

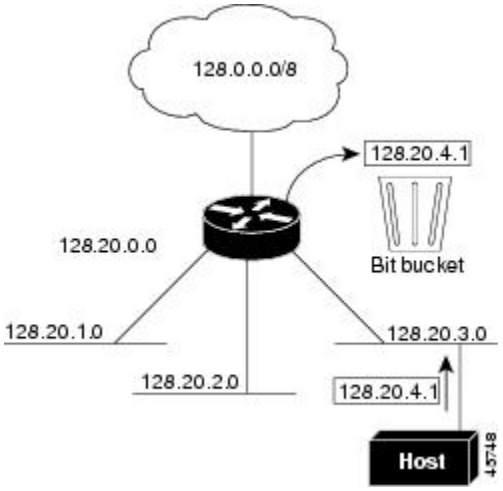
図では、クラスレス ルーティングがイネーブルとなっています。ホストがパケットを 128.20.4.1 に送信すると、ルータはパケットを廃棄せずに、最適なスーパーネット ルートに転送します。クラスレスルーティングがディセーブルの場合、デフォルトルートがないネットワークのサブネット宛てパケットを受信したルータは、パケットを廃棄します。

図 *2*：*IP* クラスレス ルーティングがイネーブルの場合



図では、ネットワーク 128.20.0.0 のルータはサブネット 128.20.1.0、128.20.2.0、128.20.3.0 に接続されています。ホストがパケットを 128.20.4.1 に送信した場合、ネットワークのデフォルト ルートが存在しないため、ルータはパケットを廃棄します。

図 *3*：*IP* クラスレス ルーティングがディセーブルの場合



スイッチが認識されないサブネット宛てのパケットを最適なスーパーネット ルートに転送しないようにするには、クラスレス ルーティング動作をディセーブルにします。

アドレス解決

インターフェイス固有の IP 処理方法を制御するには、アドレス解決を行います。IP を使用するデバイスには、ローカル セグメントまたは LAN 上のデバイスを一意に定義するローカル アドレス（MAC アドレス）と、デバイスが属するネットワークを特定するネットワーク アドレスがあります。

（注） スイッチスタックでは、スタックの単一のMACアドレスおよびIPアドレスを使用して、ネットワーク通信を行います。

ローカルアドレス（MAC アドレス）は、パケット ヘッダーのデータ リンク層（レイヤ 2）セクションに格納されて、データ リンク（レイヤ 2）デバイスによって読み取られるため、データ リンク アドレスと呼ばれます。ソフトウェアがイーサネット上のデバイスと通信するには、デバイスの MAC アドレスを学習する必要があります。IP アドレスから MAC アドレスを学習するプロセスを、アドレス解決と呼びます。MAC アドレスから IP アドレスを学習するプロセスを、逆アドレス解決と呼びます。

スイッチでは、次の形式のアドレス解決を行うことができます。

- ARP：IP アドレスを MAC アドレスと関連付けるために使用されます。ARP は IP アドレスを入力と解釈し、対応する MAC アドレスを学習します。次に、IP アドレス/MAC アドレス アソシエーションを ARP キャッシュにストアし、すぐに取り出せるようにします。その後、IP データグラムがリンク層フレームにカプセル化され、ネットワークを通じて送信されます。イーサネット以外の IEEE 802 ネットワークにおける IP データグラムのカプセル化、および ARP 要求や応答については、サブネットワーク アクセス プロトコル（SNAP）で規定されています。
- プロキシ ARP：ルーティング テーブルを持たないホストで、他のネットワークまたはサブネット上のホストの MAC アドレスを学習できるようにします。スイッチ（ルータ）が送信元と異なるインターフェイス上のホストに宛てた ARP 要求を受信した場合、そのルータに他のインターフェイスを経由してそのホストに至るすべてのルートが格納されていれば、ルータは自身のローカル データ リンク アドレスを示すプロキシ ARP パケットを生成します。ARP 要求を送信したホストはルータにパケットを送信し、ルータはパケットを目的のホストに転送します。

スイッチでは、ARP と同様の機能（ローカル MAC アドレスでなく IP アドレスを要求する点を除く）を持つ Reverse Address Resolution Protocol（RARP）を使用することもできます。RARP を使用するには、ルータ インターフェイスと同じネットワーク セグメント上に RARP サーバを設置する必要があります。サーバを識別するには、**ip rarp-server** *address* インターフェイス コンフィギュレーション コマンドを使用します。

RARP の詳細については、『*Cisco IOS Configuration Fundamentals Configuration Guide*』を参照してください。

『Proxy ARP』

プロキシ ARP は、他のルートを学習する場合の最も一般的な方法です。プロキシ ARP を使用すると、ルーティング情報を持たないイーサネット ホストと、他のネットワークまたはサブネット上のホストとの通信が可能になります。このホストでは、すべてのホストが同じローカル イーサネット上にあり、ARP を使用して MAC アドレスを学習すると想定されています。スイッチが送信元と異なるネットワーク上にあるホストに宛てた ARP 要求を受信した場合、スイッチはそのホストへの最適なルートがあるかどうかを調べます。最適なルートがある場合、スイッチはスイッチ自身のイーサネット MAC アドレスが格納された ARP 応答パケットを送信します。要求の送信元ホストはパケットをスイッチに送信し、スイッチは目的のホストにパケットを転送します。プロキシ ARP は、すべてのネットワークをローカルな場合と同様に処理し、IP アドレスごとに ARP 処理を実行します。

ICMP Router Discovery Protocol

ルータ ディスカバリを使用すると、スイッチは ICMP Router Discovery Protocol（IRDP）を使用し、他のネットワークへのルートを動的に学習します。ホストは IRDP を使用し、ルータを特定します。クライアントとして動作しているスイッチは、ルータ ディスカバリ パケットを生成します。ホストとして動作しているスイッチは、ルータ ディスカバリ パケットを受信します。スイッチは Routing Information Protocol（RIP）ルーティングのアップデートを受信し、この情報を使用してルータの場所を推測することもできます。実際のところ、ルーティング デバイスによって送信されたルーティング テーブルは、スイッチにストアされません。どのシステムがデータを送信しているのかが記録されるだけです。IRDP を使用する利点は、プライオリティと、パケットが受信されなくなってからデバイスがダウンしていると見なされるまでの期間の両方をルータごとに指定できることです。

検出された各デバイスは、デフォルト ルータの候補となります。現在のデフォルト ルータがダウンしたと宣言された場合、または再送信が多すぎて TCP 接続がタイムアウトになりつつある場合、プライオリティが上位のルータが検出されると、最も高いプライオリティを持つ新しいルータが選択されます。

UDP ブロードキャスト パケットおよびプロトコル

ユーザ データグラム プロトコル（UDP）は IP のホスト間レイヤ プロトコルで、TCP と同様です。UDP はオーバーヘッドが少ない、コネクションレスのセッションを 2 つのエンド システム間に提供しますが、受信されたデータグラムの確認応答は行いません。場合に応じてネットワーク ホストは UDP ブロードキャストを使用し、アドレス、コンフィギュレーション、名前に関する情報を検索します。このようなホストが、サーバを含まないネットワーク セグメント上にある場合、通常 UDP ブロードキャストは転送されません。この状況を改善するには、特定のクラスのブロードキャストをヘルパー アドレスに転送するように、ルータのインターフェイスを設定します。インターフェイスごとに、複数のヘルパー アドレスを使用できます。

UDP 宛先ポートを指定し、転送される UDP サービスを制御できます。複数の UDP プロトコルを指定することもできます。旧式のディスクレス Sun ワークステーションおよびネットワーク セキュリティ プロトコル SDNS で使用される Network Disk（ND）プロトコルも指定できます。

ヘルパー アドレスがインターフェイスに定義されている場合、デフォルトでは UDP と ND の両方の転送がイネーブルになっています。『*Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services*』の **ip forward-protocol** インターフェイス コンフィギュレーション コマンドの説明には、UDP ポートを指定しない場合にデフォルトで転送されるポートがリストされています。

ブロードキャスト パケットの処理

IP インターフェイス アドレスを設定したあとで、ルーティングをイネーブルにしたり、1 つまたは複数のルーティング プロトコルを設定したり、ネットワーク ブロードキャストへのスイッチの応答方法を設定したりできます。ブロードキャストは、物理ネットワーク上のすべてのホスト宛てのデータ パケットです。スイッチでは、2 種類のブロードキャストがサポートされています。

- ダイレクト ブロードキャスト パケット：特定のネットワークまたは一連のネットワークに送信されます。ダイレクト ブロードキャスト アドレスには、ネットワークまたはサブネット フィールドが含まれます。
- フラッディング ブロードキャスト パケット：すべてのネットワークに送信されます。

（注） **storm-control** インターフェイス コンフィギュレーション コマンドを使用して、トラフィック抑制レベルを設定し、レイヤ 2 インターフェイスでブロードキャスト、ユニキャスト、マルチキャスト トラフィックを制限することもできます。

ルータはローカル ケーブルまでの範囲を制限して、ブロードキャスト ストームを防ぎます。ブリッジ（インテリジェントなブリッジを含む）はレイヤ 2 デバイスであるため、ブロードキャストはすべてのネットワーク セグメントに転送され、ブロードキャスト ストームを伝播します。ブロードキャスト ストーム問題を解決する最善の方法は、ネットワーク上で単一のブロードキャスト アドレス方式を使用することです。最新の IP 実装機能ではほとんどの場合、アドレスをブロードキャスト アドレスとして使用するように設定できます。スイッチをはじめ、多数の実装機能では、ブロードキャスト メッセージを転送するためのアドレス方式が複数サポートされています。

IP ブロードキャストのフラッディング

IP ブロードキャストをインターネットワーク全体に、制御可能な方法でフラッディングできるようにするには、ブリッジング STP で作成されたデータベースを使用します。この機能を使用すると、ループを回避することもできます。この機能を使用できるようにするには、フラッディングが行われるインターフェイスごとにブリッジングを設定する必要があります。ブリッジングが設定されていないインターフェイス上でも、ブロードキャストを受信できます。ただし、ブリッジングが設定されていないインターフェイスでは、受信したブロードキャストが転送されません。

また、異なるインターフェイスで受信されたブロードキャストを送信する場合、このインターフェイスは使用されません。

IPヘルパーアドレスのメカニズムを使用して単一のネットワーク アドレスに転送されるパケットを、フラッディングできます。各ネットワーク セグメントには、パケットのコピーが 1 つだけ送信されます。

フラッディングを行う場合、パケットは次の条件を満たす必要があります（これらの条件は、IP ヘルパー アドレスを使用してパケットを転送するときの条件と同じです）。

- パケットは MAC レベルのブロードキャストでなければなりません。
- パケットは IP レベルのブロードキャストでなければなりません。
- パケットは Trivial File Transfer Protocol（TFTP）、ドメイン ネーム システム（DNS）、Time、NetBIOS、ND、または BOOTP パケット、または **ip forward-protocol udp** グローバル コンフィギュレーション コマンドで指定された UDP でなければなりません。
- パケットの存続可能時間（TTL）値は 2 以上でなければなりません。

フラッディングされた UDP データグラムには、出力インターフェイスで **ip broadcast-address** インターフェイス コンフィギュレーション コマンドによって指定された宛先アドレスが表示されます。宛先アドレスを、任意のアドレスに設定できます。このため、データグラムがネットワーク内に伝播されるにつれ、宛先アドレスが変更されることもあります。送信元アドレスは変更されません。TTL 値が減ります。

フラッディングされたUDPデータグラムがインターフェイスから送信されると（場合によっては宛先アドレスが変更される）、データグラムは通常のIP出力ルーチンに渡されます。このため、出力インターフェイスにアクセス リストがある場合、データグラムはその影響を受けます。

スイッチでは、パケットの大部分がハードウェアで転送され、スイッチのCPUを経由しません。CPU に送信されるパケットの場合は、ターボフラッディングを使用し、スパニングツリーベースの UDP フラッディングを約 4 ～ 5 倍高速化します。この機能は、ARP カプセル化用に設定されたイーサネット インターフェイスでサポートされています。

IP ルーティングの設定方法

スイッチ上で、IP ルーティングはデフォルトでディセーブルとなっているため、ルーティングを行う前に、IP ルーティングをイネーブルにする必要があります。IP ルーティングに関する設定情報については、『*Cisco IOS IP Configuration Guide*』を参照してください。

次の手順では、次に示すレイヤ 3 インターフェイスの 1 つを指定する必要があります。

- ルーテッド ポート：**no switchport** インターフェイス コンフィギュレーション コマンドを使用し、レイヤ 3 ポートとして設定された物理ポートです。
- スイッチ仮想インターフェイス（SVI）：**interface vlan** *vlan_id* グローバル コンフィギュレーション コマンドによって作成された VLAN インターフェイス。デフォルトではレイヤ 3 インターフェイスです。

（注） IP ルーティングを有効にすると、SVI として設定されている VLAN もまた、自分宛先ではないブロードキャスト ARP 要求を学習します。

- レイヤ 3 モードの EtherChannel ポート チャネル：**interface port-channel** *port-channel-number* グローバル コンフィギュレーション コマンドを使用し、イーサネット インターフェイスをチャネル グループにバインドして作成されたポートチャネル論理インターフェイス。詳細については、『Layer 2 Configuration Guide』の「Configuring Layer 3 EtherChannels」の章を参照してください。

（注） スイッチは、ユニキャスト ルーテッド トラフィックのトンネル インターフェイスをサポートしません。

ルーティングが発生するすべてのレイヤ 3 インターフェイスに、IP アドレスを割り当てる必要があります。

（注） スイッチは、各ルーテッド ポートおよび SVI に割り当てられた IP アドレスを持つことができます。

設定できるルーテッド ポートおよび SVI の個数は 128 に制限されています。推奨個数と実装されている機能の数量を超えると、ハードウェアによって制限されるため、CPU利用率が影響を受けることがあります。

ルーティングを設定するための主な手順は次のとおりです。

- VLAN インターフェイスをサポートするには、スイッチまたはスイッチ スタックで VLAN を作成および設定し、レイヤ 2 インターフェイスに VLAN メンバーシップを割り当てます。詳細については、『VLAN Configuration Guide』の「Configuring VLANs」の章を参照してください。
- レイヤ 3 インターフェイスを設定します。
- スイッチ上で IP ルーティングをイネーブルに設定します。
- レイヤ 3 インターフェイスに IP アドレスを割り当てます。
- 選択したルーティング プロトコルをスイッチ上でイネーブルにします。
- ルーティング プロトコル パラメータを設定します（任意）。

IP アドレッシングの設定方法

IP ルーティングを設定するには、レイヤ 3 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てインターフェイスをイネーブルにし、IP を使用するインターフェイスを経由してホストと

の通信を許可する必要があります。次の項では、さまざまなIPアドレス指定機能の設定方法について説明します。IP アドレスをインターフェイスに割り当てる手順は必須ですが、その他の手順は任意です。

- アドレス指定のデフォルト設定
- ネットワーク インターフェイスへの IP アドレスの割り当て
- アドレス解決方法の設定
- IP ルーティングがディセーブルの場合のルーティング支援機能
- ブロードキャスト パケットの処理方法の設定
- IP アドレスのモニタリングおよびメンテナンス

IP アドレス指定のデフォルト設定

表 *1*：アドレス指定のデフォルト設定

機能	デフォルト設定
IP アドレス	未定義
『ARP』	ARP キャッシュに永続的なエントリはありません カプセル化：標準イーサネット形式の ARP 14400 秒（4 時間）
IP ブロードキャスト アドレス	255.255.255.255（すべて 1）
IP クラスレス ルーティング	イネーブル
IP デフォルト ゲートウェイ	ディセーブル
IP ダイレクト ブロードキャスト	ディセーブル（すべてのIPダイレクトブロードキャストがドロップされます）
IP ドメイン	ドメイン リスト：ドメイン名は未定義 ドメイン検索：イネーブル ドメイン名：イネーブル

機能	デフォルト設定
IP 転送プロトコル	ヘルパー アドレスが定義されているか、またはユーザ データグラム プロトコル（UDP）フラッディングが設定されている場合、デフォルト ポートでは UDP 転送がイネーブルとなります ローカル ブロードキャスト：ディセーブル スパニングツリー プロトコル（STP）：ディセーブル ターボフラッディング：ディセーブル
IP ヘルパー アドレス	ディセーブル
IP ホスト	ディセーブル
ICMP Router Discovery Protocol（IRDP）	ディセーブル イネーブルの場合のデフォルト： • ブロードキャスト IRDP アドバタイズメント • アドバタイズメント間の最大インターバル：600 秒 • アドバタイズ間の最小インターバル：最大インターバルの 0.75 倍 • プリファレンス：0
IP プロキシ ARP	イネーブル
IP ルーティング	ディセーブル
IP サブネットゼロ	ディセーブル

ネットワーク インターフェイスへの IP アドレスの割り当て

IP アドレスは IP パケットの送信先を特定します。一部の IP アドレスは特殊な目的のために予約されていて、ホスト、サブネット、またはネットワーク アドレスには使用できません。RFC 1166 の『Internet Numbers』には IP アドレスに関する公式の説明が記載されています。

インターフェイスには、1 つのプライマリ IP アドレスを設定できます。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する場合、そのマスクをサブネット マスクと呼びます。割り当てられているネットワーク番号については、インターネット サービス プロバイダーにお問い合わせください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**no switchport** 例： `Device(config-if)# no switchport`	レイヤ 2 コンフィギュレーション モードからインターフェイスを削除します（物理インターフェイスの場合）。
ステップ 5	**ip address** *ip-address subnet-mask* 例： `Device(config-if)# ip address 10.1.5.1 255.255.255.0`	IP アドレスおよび IP サブネット マスクを設定します。
ステップ 6	**no shutdown** 例： `Device(config-if)# no shutdown`	物理インターフェイスをイネーブルにします。
ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip route** 例： `Device# show ip route`	入力を確認します。

	コマンドまたはアクション	目的
ステップ 9	**show ip interface** [*interface-id*] 例： `Device# show ip interface gigabitethernet 1/0/1`	入力を確認します。
ステップ 10	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

サブネット ゼロの使用

サブネット アドレスがゼロであるサブネットを作成しないでください。同じアドレスを持つネットワークおよびサブネットがある場合に問題が発生することがあります。たとえば、ネットワーク 131.108.0.0 のサブネットが 255.255.255.0 の場合、サブネット ゼロは 131.108.0.0 と記述され、ネットワーク アドレスと同じとなってしまいます。

すべてが 1 のサブネット（131.108.255.0）は使用可能です。また、IP アドレス用にサブネット スペース全体が必要な場合は、サブネット ゼロの使用をイネーブルにできます（ただし推奨できません）。

デフォルトに戻して、サブネット ゼロの使用をディセーブルにするには、**no ip subnet-zero** グローバル コンフィギュレーション コマンドを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip subnet-zero** 例： `Device(config)# ip subnet-zero`	インターフェイス アドレスおよびルーティングのアップデート時にサブネット ゼロの使用をイネーブルにします。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

クラスレス ルーティングのディセーブル化

スイッチが認識されないサブネット宛てのパケットを最適なスーパーネット ルートに転送しないようにするには、クラスレス ルーティング動作をディセーブルにします。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**no ip classless** 例： `Device(config)#no ip classless`	クラスレス ルーティング動作をディセーブルにします。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

アドレス解決方法の設定

アドレス解決を設定するために必要な作業は次のとおりです。

スタティック ARP キャッシュの定義

ARP および他のアドレス解決プロトコルを使用すると、IP アドレスと MAC アドレス間をダイナミックにマッピングできます。ほとんどのホストではダイナミック アドレス解決がサポートされているため、通常の場合、スタティック ARP キャッシュ エントリを指定する必要はありません。スタティック ARP キャッシュ エントリを定義する必要がある場合は、グローバルにそれを定義できます。グローバルに定義すると、IP アドレスを MAC アドレスに変換するためにスイッチが使用するARP キャッシュに永続的なエントリをインストールします。また、指定されたIPアドレス

に属しているかのように、スイッチがARP要求に応答するように指定することもできます。ARPエントリを永続的なエントリにしない場合は、ARPエントリのタイムアウト期間を指定できます。

手順

	コマンドまたはアクション	目的
ステップ1	**enable** 例： `Device> enable`	特権EXECモードをイネーブルにします。パスワードを入力します（要求された場合）。
ステップ2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ3	**arp** *ip-address hardware-address type* 例： `Device(config)# ip 10.1.5.1 c2f3.220a.12f4 arpa`	ARP キャッシュ内でIPアドレスをMAC（ハードウェア）アドレスに関連付け、次に示すカプセル化タイプのいずれかを指定します。 • **arpa**：ARP カプセル化（イーサネット インターフェイス用） • **snap**：SNAPカプセル化（トークン リングおよびFDDI インターフェイス用） • **sap**：HP の ARP タイプ
ステップ4	**arp** *ip-address hardware-address type* [**alias**] 例： `Device(config)# ip 10.1.5.3 d7f3.220d.12f5 arpa alias`	（任意）指定されたIPアドレスがスイッチに属する場合と同じ方法で、スイッチがARP要求に応答するように指定します。
ステップ5	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。
ステップ6	**arp** *timeout seconds* 例： `Device(config-if)# arp 20000`	（任意）ARPキャッシュ エントリがキャッシュに保持される期間を設定します。デフォルト値は14400秒（4時間）です。指定できる範囲は0 ～ 2147483秒です。

	コマンドまたはアクション	目的
ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show interfaces** [*interface-id*] 例： `Device# show interfaces gigabitethernet 1/0/1`	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で使用される ARP のタイプおよびタイムアウト値を確認します。
ステップ 9	**show arp** 例： `Device# show arp`	ARP キャッシュの内容を表示します。
ステップ 10	**show ip arp** 例： `Device# show ip arp`	ARP キャッシュの内容を表示します。
ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ARP のカプセル化の設定

IP インターフェイスでは、イーサネット ARP カプセル化（**arpa** キーワードで表される）がデフォルトでイネーブルに設定されています。ネットワークの必要性に応じて、カプセル化方法を SNAP に変更できます。

カプセル化タイプをディセーブルにするには、**no arp arpa** または **no arp snap** インターフェイス コンフィギュレーション コマンドを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/2`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**arp** {**arpa** | **snap**} 例： `Device(config-if)# arp arpa`	ARP カプセル化方法を指定します。 • **arpa**：Address Resolution Protocol • **snap**：Subnetwork Address Protocol
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show interfaces** [*interface-id*] 例： `Device# show interfaces`	すべてのインターフェイスまたは指定されたインターフェイスの ARP カプセル化設定を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

プロキシ ARP のイネーブル化

デフォルトでは、プロキシARPがスイッチで使用されます。ホストが他のネットワークまたはサブネット上のホストの MAC アドレスを学習できるようにするためです。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/2`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ3インターフェイスを指定します。
ステップ 4	**ip proxy-arp** 例： `Device(config-if)# ip proxy-arp`	インターフェイス上でプロキシ ARP をイネーブルにします。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip interface** [*interface-id*] 例： `Device# show ip interface gigabitethernet 1/0/2`	指定されたインターフェイスまたはすべてのインターフェイスの設定を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IP ルーティングがディセーブルの場合のルーティング支援機能

次のメカニズムを使用することで、スイッチは IP ルーティングがイネーブルでない場合、別のネットワークへのルートを学習できます。

- 『Proxy ARP』
- デフォルト ゲートウェイ
- ICMP Router Discovery Protocol（IRDP）

『Proxy ARP』

プロキシARPは、デフォルトでイネーブルに設定されています。ディセーブル化されたプロキシARPをイネーブルにするには、「プロキシARPのイネーブル化」の項を参照してください。プロキシ ARP は、他のルータでサポートされているかぎり有効です。

デフォルト ゲートウェイ

ルートを特定するもう 1 つの方法は、デフォルト ルータ、つまりデフォルト ゲートウェイを定義する方法です。ローカルでないすべてのパケットはこのルータに送信されます。このルータは適切なルーティングを行う、または IP 制御メッセージ プロトコル（ICMP）リダイレクト メッセージを返信するという方法で、ホストが使用するローカル ルータを定義します。スイッチはリダイレクト メッセージをキャッシュに格納し、各パケットをできるだけ効率的に転送します。この方法には、デフォルト ルータがダウンした場合、または使用できなくなった場合に、検出が不可能となる制限があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権EXECモードをイネーブルにします。パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**ip default-gateway** *ip-address* 例： `Device(config)# ip default gateway 10.1.5.1`	デフォルト ゲートウェイ（ルータ）を設定します。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show ip redirects** 例： `Device# show ip redirects`	設定を確認するため、デフォルト ゲートウェイ ルータのアドレスを表示します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ICMP Router Discovery Protocol（IRDP）

インターフェイスでIRDPルーティングを行う場合は、インターフェイスでIRDP処理をイネーブルにしてください。IRDP処理をイネーブルにすると、デフォルトのパラメータが適用されます。

これらのパラメータを変更することもできます。**maxadvertinterval** 値を変更すると、**holdtime** 値および **minadvertinterval** 値も変更されます。最初に **maxadvertinterval** 値を変更し、次に **holdtime** 値または **minadvertinterval** 値のいずれかを手動で変更することが重要です。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権EXECモードをイネーブルにします。パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**ip irdp** 例： `Device(config-if)# ip irdp`	インターフェイスで IRDP 処理をイネーブルにします。
ステップ 5	**ip irdp multicast** 例： `Device(config-if)# ip irdp multicast`	（任意）IP ブロードキャストの代わりとして、マルチキャスト アドレス（224.0.0.1）に IRDP アドバタイズを送信します。 （注） このコマンドを使用すると、IRDP パケットをマルチキャストとして送信するサン マイクロシステムズ社の Solaris との互換性を維持できます。実装機能の中には、これらのマルチキャストを受信できないものも多くあります。このコマンドを使用する前に、エンドホストがこの機能に対応していることを確認してください。
ステップ 6	**ip irdp holdtime** *seconds* 例： `Device(config-if)# ip irdp holdtime 1000`	（任意）アドバタイズが有効である IRDP 期間を設定します。デフォルト値は **maxadvertinterval** 値の 3 倍です。**maxadvertinterval** 値よりも大きな値（9000 秒以下）を指定する必要があります。**maxadvertinterval** 値を変更すると、この値も変更されます。
ステップ 7	**ip irdp maxadvertinterval** *seconds* 例： `Device(config-if)# ip irdp maxadvertinterval 650`	（任意）アドバタイズメントの IRDP 最大間隔を設定します。デフォルトは 600 秒です。

	コマンドまたはアクション	目的
ステップ 8	**ip irdp minadvertinterval** *seconds* 例： `Device(config-if)# ip irdp minadvertinterval 500`	（任意）アドバタイズ間のIRDPの最小インターバルを設定します。デフォルト値は**maxadvertinterval**値の 0.75 倍です。**maxadvertinterval** を変更すると、この値も新しいデフォルト値（**maxadvertinterval** の 0.75 倍）に変更されます。
ステップ 9	**ip irdp preference** *number* 例： `Device(config-if)# ip irdp preference 2`	（任意）デバイスの IRDP プリファレンス レベルを設定します。指定できる範囲は-231～231です。デフォルトは0です。大きな値を設定すると、ルータのプリファレンス レベルも高くなります。
ステップ 10	**ip irdp address** *address* [*number*] 例： `Device(config-if)# ip irdp address 10.1.10.10`	（任意）プロキシアドバタイズを行うためのIRDPアドレスとプリファレンスを設定します。
ステップ 11	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 12	**show ip irdp** 例： `Device# show ip irdp`	IRDP 値を表示し、設定を確認します。
ステップ 13	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ブロードキャスト パケットの処理方法の設定

これらの方式をイネーブルにするには、次に示す作業を実行します。

- ダイレクト ブロードキャストから物理ブロードキャストへの変換のイネーブル化
- UDP ブロードキャスト パケットおよびプロトコルの転送

- IP ブロードキャスト アドレスの確立
- IP ブロードキャストのフラッディング

ダイレクト ブロードキャストから物理ブロードキャストへの変換のイネーブル化

デフォルトでは、IP ダイレクト ブロードキャストがドロップされるため、転送されることはありません。IP ダイレクト ブロードキャストがドロップされると、ルータが DoS 攻撃（サービス拒絶攻撃）にさらされる危険が少なくなります。

ブロードキャストが物理（MAC レイヤ）ブロードキャストになるインターフェイスでは、IP ダイレクト ブロードキャストの転送をイネーブルにできます。**ip forward-protocol** グローバル コンフィギュレーション コマンドを使用し、設定されたプロトコルだけを転送できます。

転送するブロードキャストを制御するアクセス リストを指定できます。アクセス リストを指定すると、アクセス リストで許可されている IP パケットだけが、ダイレクト ブロードキャストから物理ブロードキャストに変換できるようになります。アクセス リストの詳細については、『Security Configuration Guide』の「Information about Network Security with ACLs」の項を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/2`	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。
ステップ 4	**ip directed-broadcast** [*access-list-number*] 例： `Device(config-if)# ip directed-broadcast 103`	インターフェイス上で、ダイレクト ブロードキャストから物理ブロードキャストへの変換をイネーブルにします。転送するブロードキャストを制御するアクセス リストを指定できます。アクセス リストを指定すると、アクセス リストで許可されている IP パケットだけが変換可能になります。

	コマンドまたはアクション	目的
		（注） **ip directed-broadcast** インターフェイス コンフィギュレーション コマンドはVPN ルーティングおよび転送（VRF）インターフェイスで設定でき、こうするとVRF対応になります。ダイレクトブロードキャスト トラフィックが VRF 内でだけルーティングされます。
ステップ 5	**exit** 例： `Device(config-if)# exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 6	**ip forward-protocol** {**udp** [*port*] | **nd** | **sdns**} 例： `Device(config)# ip forward-protocol nd`	ブロードキャスト パケットを転送するとき、ルータによって転送されるプロトコルおよびポートを指定します。 • **udp**：UPD データグラムを転送します。 port：（任意）転送される UDP サービスを制御する宛先ポートです。 • **nd**：ND データグラムを転送します。 • **sdns**：SDNS データグラムを転送します。
ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip interface** [*interface-id*] 例： `Device# show ip interface`	指定されたインターフェイスまたはすべてのインターフェイスの設定を確認します。
ステップ 9	**show running-config** 例： `Device# show running-config`	入力を確認します。

	コマンドまたはアクション	目的
ステップ 10	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

UDP ブロードキャスト パケットおよびプロトコルの転送

UDP ブロードキャストの転送を設定するときに UDP ポートを指定しないと、ルータは BOOTP フォワーディング エージェントとして動作するように設定されます。BOOTP パケットは Dynamic Host Configuration Protocol（DHCP）情報を伝達します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	**ip helper-address** *address* 例： `Device(config-if)# ip helper address 10.1.10.1`	転送をイネーブルにし、BOOTP などの UDP ブロードキャスト パケットを転送するための宛先アドレスを指定します。

	コマンドまたはアクション	目的
ステップ 5	**exit** 例： `Device(config-if)# exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 6	**ip forward-protocol** {**udp** [*port*] | **nd** | **sdns**} 例： `Device(config)# ip forward-protocol sdns`	ブロードキャスト パケットを転送するときに、ルータによって転送されるプロトコルを指定します。
ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip interface** [*interface-id*] 例： `Device# show ip interface gigabitethernet 1/0/1`	指定されたインターフェイスまたはすべてのインターフェイスの設定を確認します。
ステップ 9	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 10	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IP ブロードキャスト アドレスの確立

最も一般的な（デフォルトの）IP ブロードキャスト アドレスは、すべて 1 で構成されているアドレス（255.255.255.255）です。ただし、任意の形式の IP ブロードキャスト アドレスを生成するようにスイッチを設定することもできます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。
ステップ 4	**ip broadcast-address** *ip-address* 例： `Device(config-if)# ip broadcast-address 128.1.255.255`	デフォルト値と異なるブロードキャスト アドレス（128.1.255.255 など）を入力します。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip interface** [*interface-id*] 例： `Device# show ip interface`	指定されたインターフェイスまたはすべてのインターフェイスのブロードキャスト アドレスを確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IP ブロードキャストのフラッディング

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip forward-protocol spanning-tree** 例： `Device(config)# ip forward-protocol spanning-tree`	ブリッジング スパニングツリー データベースを使用し、UDP データグラムをフラッディングします。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。
ステップ 7	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 8	**ip forward-protocol turbo-flood** 例： `Device(config)# ip forward-protocol turbo-flood`	スパニングツリー データベースを使用し、UDP データグラムのフラッディングを高速化します。
ステップ 9	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IP アドレスのモニタリングおよびメンテナンス

特定のキャッシュ、テーブル、またはデータベースの内容が無効になっている場合、または無効である可能性がある場合は、**clear** 特権 EXEC コマンドを使用し、すべての内容を削除できます。次の表に、内容をクリアするために使用するコマンドを示します。

表 *2*：キャッシュ、テーブル、データベースをクリアするコマンド

clear arp-cache	IP ARP キャッシュおよび高速スイッチング キャッシュをクリアします。
clear host {*name* | ***}**	ホスト名およびアドレス キャッシュから 1 つまたはすべてのエントリを削除します。
clear ip route {**network** [*mask*] | ***}**	IP ルーティング テーブルから 1 つまたは複数のルートを削除します。

IP ルーティング テーブル、キャッシュ、データベースの内容、ノードへの到達可能性、ネットワーク内のパケットのルーティング パスなど、特定の統計情報を表示できます。次の表に、IP 統計情報を表示するために使用する特権 EXEC コマンドを示します。

表 *3*：キャッシュ、テーブル、データベースを表示するコマンド

show arp	ARP テーブル内のエントリを表示します。
show hosts	デフォルトのドメイン名、検索サービスの方式、サーバ ホスト名、およびキャッシュに格納されているホスト名とアドレスのリストを表示します。
show ip aliases	TCP ポートにマッピングされた IP アドレスを表示します（エイリアス）。
show ip arp	IP ARP キャッシュを表示します。
show ip interface [*interface-id*]	インターフェイスの IP ステータスを表示します。
show ip irdp	IRDP 値を表示します。
show ip masks *address*	ネットワーク アドレスに対して使用されるマスクおよび各マスクを使用するサブネット番号を表示します。
show ip redirects	デフォルト ゲートウェイのアドレスを表示します。
show ip route [*address* [*mask*]] | [*protocol*]	ルーティング テーブルの現在の状態を表示します。
show ip route summary	サマリー形式でルーティング テーブルの現在のステータスを表示します。

IP ユニキャスト ルーティングの設定方法

IP ユニキャスト ルーティングのイネーブル化

デフォルトで、スイッチはレイヤ 2 スイッチング モード、IP ルーティングはディセーブルとなっています。スイッチのレイヤ 3 機能を使用するには、IP ルーティングをイネーブルにする必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip routing** 例： `Device(config)# ip routing`	IP ルーティングをイネーブルにします。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IP ルーティングのイネーブル化の例

次に、ルーティング プロトコルとして RIP を使用し、上で IP ルーティングをイネーブルにする例を示します。

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# ip routing
```

```
Device(config)# router rip
Device(config-router)# network 10.0.0.0
Device(config-router)# end
```

次の作業

ここで、選択したルーティング プロトコルのパラメータを設定できます。具体的な手順は次のとおりです。

- RIP
- OSPF
- EIGRP
- BGP
- ユニキャスト Reverse Path Forwarding
- プロトコル独立機能（任意）

RIP 情報

RIP は、小規模な同種ネットワーク間で使用するために作成された Interior Gateway Protocol（IGP）です。RIP は、ブロードキャスト ユーザ データグラム プロトコル（UDP）データ パケットを使用してルーティング情報を交換するディスタンスベクトルルーティング プロトコルです。このプロトコルは RFC 1058 に文書化されています。RIP の詳細については、『*IP Routing Fundamentals*』（Cisco Press 刊）を参照してください。

（注） RIP は IP Base Network Essentials 機能セットでサポートされています。

スイッチは RIP を使用し、30 秒ごとにルーティング情報アップデート（アドバタイズメント）を送信します。180 秒以上を経過しても別のルータからアップデートがルータに届かない場合、該当するルータから送られたルートは使用不能としてマークされます。240 秒後もまだ更新がない場合、ルータは更新のないルータのルーティング テーブル エントリをすべて削除します。

RIP では、各ルートの値を評価するためにホップ カウントが使用されます。ホップ カウントは、ルート内で経由されるルータ数です。直接接続されているネットワークのホップ カウントは 0 です。ホップ カウントが 16 のネットワークに到達できません。このように範囲（0 ～ 15）が狭いため、RIP は大規模ネットワークには適していません。

ルータにデフォルトのネットワーク パスが設定されている場合、RIP はルータを疑似ネットワーク 0.0.0.0 にリンクするルートをアドバタイズします。0.0.0.0 ネットワークは存在しません。RIP はデフォルトのルーティング機能を実行するためのネットワークとして、このネットワークを処理します。デフォルト ネットワークが RIP によって学習された場合、またはルータにラスト リゾート ゲートウェイがあり、RIP がデフォルトのメトリックによって設定されている場合、スイッチはデフォルト ネットワークをアドバタイズします。RIP は指定されたネットワーク内のインター

フェイスにアップデートを送信します。インターフェイスのネットワークを指定しなければ、RIP のアップデート中にアドバタイズされません。

サマリー アドレスおよびスプリット ホライズン

ブロードキャストタイプの IP ネットワークに接続され、ディスタンスベクトル ルーティング プロトコルを使用するルータでは、通常ルーティングループの発生を抑えるために、スプリットホライズンメカニズムが使用されます。スプリットホライズンは、ルートに関する情報の発信元であるインターフェイス上の、ルータによって、その情報がアドバタイズされないようにします。この機能を使用すると、通常の場合は複数のルータ間通信が最適化されます（特にリンクが壊れている場合）。

RIP の設定方法

RIP のデフォルト設定

表 *4* : *RIP* のデフォルト設定

機能	デフォルト設定
自動サマリー	イネーブル
デフォルト情報送信元	ディセーブル
デフォルト メトリック	自動メトリック変換（組み込み）
IP RIP 認証キーチェーン	認証なし 認証モード：クリア テキスト
IP RIP の起動	ディセーブル
IP スプリット ホライズン	メディアにより異なる
Neighbor	未定義
ネットワーク	指定なし
オフセット リスト	ディセーブル
出力遅延	0 ミリ秒

機能	デフォルト設定
タイマー基準	• 更新：30 秒 • 無効：180 秒 • ホールドダウン：180 秒 • フラッシュ：240 秒
アップデート送信元の検証	イネーブル
Version	RIP バージョン 1 およびバージョン 2 パケットを受信し、バージョン 1 パケットを送信します。

基本的な RIP パラメータの設定

RIP を設定するには、ネットワークに対して RIP ルーティングをイネーブルにします。他のパラメータを設定することもできます。スイッチでは、ネットワーク番号を設定するまでRIP コンフィギュレーション コマンドは無視されます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権EXECモードをイネーブルにします。パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip routing** 例： `Device(config)# ip routing`	IPルーティングをイネーブルにします。（IPルーティングがディセーブルになっている場合だけ、必須です）。

	コマンドまたはアクション	目的
ステップ 4	**router rip** 例： `Device(config)# router rip`	RIP ルーティング プロセスをイネーブルにし、ルータ コンフィギュレーション モードを開始します。
ステップ 5	**network** *network number* 例： `Device(config)# network 12`	ネットワークを RIP ルーティング プロセスと関連付けます。複数の **network** コマンドを指定できます。RIP ルーティング アップデートの送受信は、これらのネットワークのインターフェイスを経由する場合だけ可能です。 （注） RIP コマンドを有効にするには、ネットワーク番号を設定する必要があります。
ステップ 6	**neighbor** *ip-address* 例： `Device(config)# neighbor 10.2.5.1`	（任意）ルーティング情報を交換する隣接ルータを定義します。このステップを使用すると、RIP（通常はブロードキャスト プロトコル）からのルーティング アップデートが非ブロードキャスト ネットワークに到達するようになります。
ステップ 7	**offset-list** [*access-list number* | *name*] {**in** | **out**} *offset* [*type number*] 例： `Device(config)# offset-list 103 in 10`	（任意）オフセット リストをルーティング メトリックに適用し、RIP によって取得したルートへの着信および発信メトリックを増加します。アクセス リストまたはインターフェイスを使用し、オフセット リストを制限できます。
ステップ 8	**timers basic** *update invalid holddown flush* 例： `Device(config)# timers basic 45 360 400 300`	（任意）ルーティング プロトコル タイマーを調整します。すべてのタイマーの有効範囲は 0 ～ 4294967295 秒です。 • *update*：ルーティング アップデートの送信間隔。デフォルトは 30 秒です。 • *invalid*：ルートが無効と宣言されるまでの時間。デフォルト値は 180 秒です。 • *holddown*：ルートがルーティング テーブルから削除されるまでの時間。デフォルト値は 180 秒です。 • *flush*：ルーティング アップデートが延期される時間。デフォルトは 240 秒です。

	コマンドまたはアクション	目的
ステップ 9	**version** {**1** | **2**} 例： `Device(config)# version 2`	（任意）RIP バージョン 1 または RIP バージョン 2 のパケットだけを送受信するようにスイッチを設定します。デフォルトの場合、スイッチではバージョン 1 および 2 を受信しますが、バージョン 1 だけを送信します。インターフェイス コマンド **ip rip** {**send** | **receive**} **version 1** | **2** | **1 2**} を使用し、インターフェイスでの送受信に使用するバージョンを制御することもできます。
ステップ 10	**no auto summary** 例： `Device(config)# no auto summary`	（任意）自動要約をディセーブルにします。デフォルトでは、クラスフルネットワーク境界を通過するときにサブプレフィックスがサマライズされます。サマライズをディセーブルにし（RIP バージョン 2 だけ）、クラスフルネットワーク境界にサブネットおよびホストルーティング情報をアドバタイズします。
ステップ 11	**no validate-update-source** 例： `Device(config)# no validdate-update-source`	（任意）着信 RIP ルーティング アップデートの送信元 IP アドレスの検証をディセーブルにします。デフォルトでは、スイッチが着信 RIP ルーティング アップデートの送信元 IP アドレスを検証します。送信元アドレスが無効な場合は、アップデートが廃棄されます。通常の環境で使用する場合は、この機能をディセーブルにしないでください。ただし、ネットワークに接続されていないルータがあり、そのルータのアップデートを受信する場合は、このコマンドを使用できます。
ステップ 12	**output-delay** *delay* 例： `Device(config)# output-delay 8`	（任意）送信する RIP アップデートにパケット間遅延を追加します。デフォルトでは、複数のパケットからなる RIP アップデートのパケットに、パケット間遅延が追加されません。パケットを低速なデバイスに送信する場合は、8 ～ 50 ミリ秒のパケット間遅延を追加できます。
ステップ 13	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 14	**show ip protocols** 例： `Device# show ip protocols`	入力を確認します。

	コマンドまたはアクション	目的
ステップ 15	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

RIP 認証の設定

RIP Version 1 は認証をサポートしていません。RIP バージョン 2 のパケットを送受信する場合は、インターフェイスでRIP認証をイネーブルにできます。インターフェイスで使用できる一連のキーは、キーチェーンによって指定されます。キーチェーンが設定されていないと、デフォルトの場合でも認証は実行されません。

RIP 認証がイネーブルであるインターフェイスでは、プレーン テキストと MD5 という 2 つの認証モードがスイッチでサポートされます。デフォルトはプレーン テキストです。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権EXECモードをイネーブルにします。パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。
ステップ 4	**ip rip authentication key-chain** *name-of-chain*	RIP 認証をイネーブルにします。

	コマンドまたはアクション	目的
	例： `Device(config-if)# ip rip authentication key-chain trees`	
ステップ 5	**ip rip authentication mode {text | md5}** 例： `Device(config-if)# ip rip authentication mode md5`	プレーン テキスト認証（デフォルト）またはMD5ダイジェスト認証を使用するように、インターフェイスを設定します。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

サマリー アドレスおよびスプリット ホライズンの設定

（注） ルートを適切にアドバタイズするため、アプリケーションがスプリットホライズンをディセーブルにする必要がある場合を除き、通常はこの機能をディセーブルにしないでください。

ダイヤルアップ クライアント用のネットワーク アクセス サーバで、サマライズされたローカル IP アドレス プールをアドバタイズするように、RIP が動作しているインターフェイスを設定する場合は、**ip summary-address rip** インターフェイス コンフィギュレーション コマンドを使用します。

（注） スプリット ホライズンがイネーブルの場合、自動サマリーとインターフェイス IP サマリー アドレスはともにアドバタイズされません。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ3インターフェイスを指定します。
ステップ 4	**ip address** *ip-address subnet-mask* 例： `Device(config-if)# ip address 10.1.1.10 255.255.255.0`	IP アドレスおよび IP サブネットを設定します。
ステップ 5	**ip summary-address rip ip address** *ip-network mask* 例： `Device(config-if)# ip summary-address rip ip address 10.1.1.30 255.255.255.0`	サマライズする IP アドレスおよび IP ネットワーク マスクを設定します。
ステップ 6	**no ip split horizon** 例： `Device(config-if)# no ip split horizon`	インターフェイスでスプリット ホライズンをディセーブルにします。

	コマンドまたはアクション	目的
ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show ip interface** *interface-id* 例： `Device# show ip interface gigabitethernet 1/0/1`	入力を確認します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

スプリット ホライズンの設定

ブロードキャストタイプの IP ネットワークに接続され、ディスタンスベクトル ルーティング プロトコルを使用するルータでは、通常ルーティングループの発生を抑えるために、スプリットホライズンメカニズムが使用されます。スプリットホライズンは、ルートに関する情報の発信元であるインターフェイス上の、ルータによって、その情報がアドバタイズされないようにします。この機能を使用すると、複数のルータ間通信が最適化されます（特にリンクが壊れている場合）。

（注） ルートを適切にアドバタイズするために、アプリケーションがスプリット ホライズンをディセーブルにする必要がある場合を除き、通常この機能をディセーブルにしないでください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するインターフェイスを指定します。
ステップ 4	**ip address** *ip-address subnet-mask* 例： `Device(config-if)# ip address 10.1.1.10 255.255.255.0`	IP アドレスおよび IP サブネットを設定します。
ステップ 5	**no ip split-horizon** 例： `Device(config-if)# no ip split-horizon`	インターフェイスでスプリット ホライズンをディセーブルにします。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip interface** *interface-id* 例： `Device# show ip interface gigabitethernet 1/0/1`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

サマリーアドレスおよびスプリットホライズンの設定例

次の例では、主要ネットは 10.0.0.0 です。自動サマリー アドレス 10.0.0.0 はサマリー アドレス 10.2.0.0 によって上書きされるため、10.2.0.0 はインターフェイス ギガビット イーサネット ポート 2 からアドバタイズされますが、10.0.0.0 はアドバタイズされません。次の例では、インターフェイスがまだレイヤ 2 モード（デフォルト）の場合、**no switchport** インターフェイス コンフィギュレーション コマンドを入力してから、**ip address** インターフェイス コンフィギュレーション コマンドを入力する必要があります。

（注） スプリット ホライズンがイネーブルである場合、（**ip summary-address rip** ルータ コンフィギュレーション コマンドによって設定される）自動サマリーとインターフェイス サマリー アドレスはともにアドバタイズされません。

```
Device(config)# router rip
Device(config-router)# interface gigabitethernet1/0/2
Device(config-if)# ip address 10.1.5.1 255.255.255.0
Device(config-if)# ip summary-address rip 10.2.0.0 255.255.0.0
Device(config-if)# no ip split-horizon
Device(config-if)# exit
Device(config)# router rip
Device(config-router)# network 10.0.0.0
Device(config-router)# neighbor 2.2.2.2 peer-group mygroup
Device(config-router)# end
```

OSPF に関する情報

OSPF は IP ネットワーク専用の IGP で、IP サブネット化、および外部から取得したルーティング情報のタグ付けをサポートしています。OSPF を使用するとパケット認証も可能になり、パケットを送受信するときに IP マルチキャストが使用されます。シスコの実装では、RFC1253 の OSPF 管理情報ベース（MIB）がサポートされています。

（注） OSPF は IP ベースではサポートされません。

シスコの実装は、次の主要機能を含む OSPF バージョン 2 仕様に準拠します。

- スタブ エリアの定義がサポートされています。
- 任意の IP ルーティング プロトコルによって取得されたルートは、別の IP ルーティング プロトコルに再配信されます。つまり、ドメイン内レベルで、OSPF は EIGRP および RIP によって取得したルートを取り込むことができます。OSPF ルートを RIP に伝達することもできます。
- エリア内の隣接ルータ間でのプレーン テキスト認証および MD5 認証がサポートされています。

- 設定可能なルーティング インターフェイス パラメータには、インターフェイス出力コスト、再送信インターバル、インターフェイス送信遅延、ルータ プライオリティ、ルータのデッド インターバルと hello インターバル、認証キーなどがあります。
- 仮想リンクがサポートされています。
- RFC 1587 に基づく Not-So-Stubby-Area（NSSA）がサポートされています。

通常、OSPF を使用するには、多くの内部ルータ、複数のエリアに接続された Area Border Router（ABR; エリア境界ルータ）、および自律システム境界ルータ（ASBR）間で調整する必要があります。最小設定では、すべてのデフォルト パラメータ値、エリアに割り当てられたインターフェイスが使用され、認証は行われません。環境をカスタマイズする場合は、すべてのルータの設定を調整する必要があります。

OSPF NSF

スイッチまたはスイッチ スタックは 2 つのレベルのノンストップ フォワーディング（NSF）をサポートしています。

- OSPF NSF 認識, （47 ページ）
- OSPF NSF 対応, （47 ページ）

OSPF NSF 認識

IP サービス フィーチャ セットでは、IPv4 の OSPF NSF 認識がサポートされます。隣接ルータが NSF 対応である場合、レイヤ 3 スイッチでは、ルータに障害（クラッシュ）が発生してプライマリ ルート プロセッサ（RP）がバックアップ RP によって引き継がれる間、または処理を中断させずにソフトウェア アップグレードを行うためにプライマリ RP を手動でリロードしている間、隣接ルータからパケットを転送し続けます。

この機能をディセーブルにできません。

OSPF NSF 対応

IP サービス フィーチャ セットでは、前のリリースでサポートされていた OSPFv2 NSF Cisco フォーマットに加えて、OSPFv2 NSF IETF フォーマットもサポートされます。この機能の詳細については、『*NSF—OSPF (RFC 3623 OSPF Graceful Restart)*』を参照してください。

IP サービス フィーチャ セットは、OSPF NSF 対応ルーティングも IPv4 に対してサポートし、スタック マスター変更後のコンバージェンスの向上と、トラフィック損失の低減を実現します。OSPF NSF 対応スタックでスタック マスターの変更が生じた場合、新しいスタック マスターは自身のリンクステート データベースを OSPF ネイバーと再同期化するために、次の 2 つの処理をする必要があります。

- ネイバー関係をリセットせずにネットワーク上の使用可能な OSPF ネイバーを解放します。
- ネットワークのリンクステート データベースの内容を再取得します。

スタック マスターの変更後、新しいマスターは隣接する NSF 認識デバイスに OSPF NSF 信号を送信します。デバイスはこの信号を、スタックとのネイバー関係をリセットしない指示として認識します。NSF 対応スタック マスターは、ネットワーク上の他のルータから信号を受け取ると、自身のネイバー リストの再構築を開始します。

NSF 対応スタック マスターはネイバー関係を再確立すると、自身のデータベースを NSF 認識ネイバーと再同期化し、OSPF ネイバー間でルーティング情報を交換します。新しいスタック マスターはこのルーティング情報を使用して、新しい情報を基に古いルートの削除、ルーティング情報ベース（RIB）の更新、転送情報ベース（FIB）のアップデートを行います。これで OSPF プロトコルは完全に収束します。

（注） OSPF NSF では、すべてのネイバー ネットワーク デバイスが NSF 認識である必要があります。ネットワーク セグメント上に非 NSF 認識ネイバーが検出された場合、NSF 対応ルータはそのセグメントに対する NSF 機能をディセーブルにします。すべてのデバイスが NSF 認識または NSF 対応デバイスとなっているその他のネットワーク セグメントでは、NSF 対応機能が継続して提供されます。

OSPF NSF ルーティングをイネーブルにするには、**nsf** OSPF ルーティング コンフィギュレーション コマンドを使用します。OSPF NSF ルーティングがイネーブルになっていることを確認するには、**show ip ospf** 特権 EXEC コマンドを使用します。

詳細については、次の URL の『*Cisco Nonstop Forwarding*』を参照してください。 http://www.cisco.com/en/US/docs/ios/ha/configuration/guide/ha-nonstp_fwdg.html

OSPF エリア パラメータ

複数の OSPF エリア パラメータを設定することもできます。設定できるパラメータには、エリア、スタブ エリア、および NSSA への無許可アクセスをパスワードによって阻止する認証用パラメータがあります。スタブ エリアは、外部ルートの情報が送信されないエリアです。が、代わりに、自律システム（AS）外の宛先に対するデフォルトの外部ルートが、ABR によって生成されます。NSSA ではコアからそのエリアへ向かう LSA の一部がフラッディングされませんが、再配信することによって、エリア内の AS 外部ルートをインポートできます。

経路集約は、アドバタイズされたアドレスを、他のエリアでアドバタイズされる単一のサマリー ルートに統合することです。ネットワーク番号が連続する場合は、**area range** ルータ コンフィギュレーション コマンドを使用し、範囲内のすべてのネットワークを対象とするサマリー ルートをアドバタイズするように ABR を設定できます。

その他の OSPF パラメータ

ルータ コンフィギュレーション モードで、その他の OSPF パラメータを設定することもできます。

- ルート集約：他のプロトコルからルートを再配信すると、各ルートは外部 LSA 内で個別にアドバタイズされます。OSPF リンク ステート データベースのサイズを小さくするには、

summary-address ルータ コンフィギュレーション コマンドを使用し、指定されたネットワーク アドレスおよびマスクに含まれる、再配信されたすべてのルートを単一のルータにアドバタイズします。

- 仮想リンク：OSPF では、すべてのエリアがバックボーン エリアに接続されている必要があります。バックボーンが不連続である場合に仮想リンクを確立するには、2 つの ABR を仮想リンクのエンドポイントとして設定します。設定情報には、他の仮想エンドポイント（他の ABR）の ID、および 2 つのルータに共通する非バックボーン リンク（通過エリア）などがあります。仮想リンクをスタブ エリアから設定できません。
- デフォルト ルート：OSPF ルーティング ドメイン内へのルート再配信を設定すると、ルータは自動的に自律システム境界ルータ（ASBR）になります。ASBR を設定し、強制的に OSPF ルーティング ドメインにデフォルト ルートを生成できます。
- すべての OSPF **show** 特権 EXEC コマンドで使用されるドメイン ネーム サーバ（DNS）名を使用すると、ルータ ID やネイバー ID を指定して表示する場合に比べ、ルータを簡単に特定できます。
- デフォルト メトリック：OSPF は、インターフェイスの帯域幅に従ってインターフェイスの OSPF メトリックを計算します。メトリックは、帯域幅で分割された *ref-bw* として計算されます。ここでの *ref* のデフォルト値は 10 で、帯域幅（*bw*）は **bandwidth** インターフェイス コンフィギュレーション コマンドによって指定されます。大きな帯域幅を持つ複数のリンクの場合は、大きな数値を指定し、これらのリンクのコストを区別できます。
- アドミニストレーティブ ディスタンスは、ルーティング情報送信元の信頼性を表す数値です。0 ～ 255 の整数を指定でき、値が大きいほど信頼性は低下します。アドミニストレーティブ ディスタンスが 255 の場合はルーティング情報の送信元をまったく信頼できないため、無視する必要があります。OSPF では、エリア内のルート（エリア内）、別のエリアへのルート（エリア間）、および再配信によって学習した別のルーティング ドメインからのルート（外部）の 3 つの異なるアドミニストレーティブ ディスタンスが使用されます。どのアドミニストレーティブ ディスタンスの値でも変更できます。
- 受動インターフェイス：イーサネット上の 2 つのデバイス間のインターフェイスは 1 つのネットワーク セグメントしか表しません。このため、OSPF が送信側インターフェイスに hello パケットを送信しないようにするには、送信側デバイスを受動インターフェイスに設定する必要があります。両方のデバイスは受信側インターフェイス宛ての hello パケットを使用することで、相互の識別を可能にします。
- ルート計算タイマー：OSPF がトポロジ変更を受信してから SPF 計算を開始するまでの遅延時間、および 2 つの SPF 計算の間のホールド タイムを設定できます。
- ネイバー変更ログ：OSPF ネイバー ステートが変更されたときに Syslog メッセージを送信するようにルータを設定し、ルータの変更を詳細に表示できます。

LSA グループ ペーシング

OSPF LSA グループ ペーシング機能を使用すると、OSPF LSA をグループ化し、リフレッシュ、チェックサム、エージング機能の同期を取って、ルータをより効率的に使用できるようになりま

す。デフォルトでこの機能はイネーブルとなっています。デフォルトのペーシング インターバルは 4 分間です。通常は、このパラメータを変更する必要はありません。最適なグループ ペーシング インターバルは、ルータがリフレッシュ、チェックサム、エージングを行うLSA数に反比例します。たとえば、データベース内に約 10000 個の LSA が格納されている場合は、ペーシング インターバルを短くすると便利です。小さなデータベース（40 ～ 100 LSA）を使用する場合は、ペーシング インターバルを長くし、10 ～ 20 分に設定してください。

ループバック インターフェイス

OSPF は、インターフェイスに設定されている最大の IP アドレスをルータ ID として使用します。このインターフェイスがダウンした場合、または削除された場合、OSPF プロセスは新しいルータ ID を再計算し、すべてのルーティング情報をそのルータのインターフェイスから再送信します。ループバックインターフェイスがIPアドレスによって設定されている場合、他のインターフェイスにより大きな IP アドレスがある場合でも、OSPF はこの IP アドレスをルータ ID として使用します。ループバック インターフェイスに障害は発生しないため、安定性は増大します。OSPF は他のインターフェイスよりもループバック インターフェイスを自動的に優先し、すべてのループバック インターフェイスの中で最大の IP アドレスを選択します。

OSPF の設定方法

OSPF のデフォルト設定

表 *5*：*OSPF* のデフォルト設定

機能	デフォルト設定
インターフェイス パラメータ	コスト：1 再送信インターバル：5 秒 送信遅延：1 秒 プライオリティ：1 hello インターバル：10 秒 デッド インターバル：hello インターバルの 4 倍 認証なし パスワードの指定なし MD5 認証はディセーブル

機能	デフォルト設定
エリア	認証タイプ：0（認証なし） デフォルト コスト：1 範囲：ディセーブル スタブ：スタブ エリアは未定義 NSSA：NSSA エリアは未定義
自動コスト	100 Mb/s
デフォルト情報送信元	ディセーブルイネーブルの場合、デフォルトのメトリック設定は 10 で、外部ルート タイプのデフォルトはタイプ 2 です。
デフォルト メトリック	各ルーティング プロトコルに適切な、組み込みの自動メトリック変換
距離 OSPF	dist1（エリア内のすべてのルート）：110。 dist2（エリア間のすべてのルート）：110。 および dist3（他のルーティング ドメインからのルート）：110。
OSPF データベース フィルタ	ディセーブルすべての発信 LSA がインターフェイスにフラッディングされます。
IP OSPF 名検索	ディセーブル
隣接関係変更ログ	イネーブル
Neighbor	指定なし
ネイバーデータベース フィルタ	ディセーブルすべての発信 LSA はネイバーにフラッディングされます。
ネットワーク エリア	ディセーブル
ノンストップ フォワーディング（NSF）認識	イネーブルレイヤ 3 スイッチでは、ハードウェアやソフトウェアの変更中に、隣接する NSF 対応ルータからのパケットを転送し続けることができます。
NSF 対応	ディセーブル （注） スイッチ スタックは OSPF NSF 対応ルーティングを IPv4 に対してサポートします。
ルータ ID	OSPF ルーティング プロセスは未定義
サマリー アドレス	ディセーブル

機能	デフォルト設定
タイマー LSA グループのペーシング	240 秒
タイマー Shortest Path First（SPF）	spf 遅延：5 秒; spf ホールドタイム：10 秒
仮想リンク	エリア ID またはルータ ID は未定義 hello インターバル：10 秒 再送信インターバル：5 秒 送信遅延：1 秒 デッド インターバル：40 秒 認証キー：キーは未定義 メッセージ ダイジェスト キー（MD5）：キーは未定義

基本的な OSPF パラメータの設定

OSPF をイネーブルにするには、OSPF ルーティング プロセスを作成し、そのルーティング プロセスに関連付けられる IP アドレスの範囲を指定し、その範囲に関連付けられるエリア ID を割り当てます。IP サービス イメージを実行しているスイッチでは、Cisco OSPFv2 NSF フォーマットまたは IETF OSPFv2 NSF フォーマットのいずれかを設定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router ospf** process-id 例： `Device(config)# router ospf 15`	OSPF ルーティングをイネーブルにし、ルータ コンフィギュレーション モードを開始します。プロセス ID はローカルに割り当てられ、内部で使用される識別パラメータで、任意の正の整数を指定できます。各 OSPF ルーティング プロセスには一意の値があります。

	コマンドまたはアクション	目的
		（注） OSPF for Routed Access は、OSPFv2 インスタンスと OSPFv3 インスタンスをそれぞれ 1 つずつと、最大 200 のダイナミックに学習されるルートをサポートします。
ステップ 3	**nsf cisco** [**enforce global**] 例： `Device(config)# nsf cisco enforce global`	（任意）OSPF での Cisco NSF 動作をイネーブルにします。**enforce global** キーワードを指定すると、非 NSF 認識のネイバー ネットワーキング デバイスが検出されたときに NSF 再起動がキャンセルされます。 （注） ステップ 3 またはステップ 4 でコマンドを入力し、ステップ 5 に進みます。
ステップ 4	**nsf ietf** [**restart-interval** *seconds*] 例： `Device(config)# nsf ietf restart-interval 60`	（任意）OSPF での IETF NSF 動作をイネーブルにします。**restart-interval** キーワードでは、グレースフル リスタート間隔の長さを秒単位で指定します。範囲は 1 ～ 1800 です。デフォルトは 120 です。 （注） ステップ 3 またはステップ 4 でコマンドを入力し、ステップ 5 に進みます。
ステップ 5	**network** address wildcard-mask **area** area-id 例： `Device(config)# network 10.1.1.1 255.240.0.0 area 20`	OSPF が動作するインターフェイス、およびそのインターフェイスのエリア ID を定義します。単一のコマンドにワイルドカードマスクを指定し、特定の OSPF エリアに関連付けるインターフェイスを 1 つまたは複数定義できます。エリア ID には 10 進数または IP アドレスを指定できます。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip protocols** 例： `Device# show ip protocols`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

OSPF インターフェイスの設定

ip ospf インターフェイス コンフィギュレーション コマンドを使用すると、インターフェイス固有の OSPF パラメータを変更できます。これらのパラメータを変更する必要はありませんが、一部のインターフェイス パラメータ（hello インターバル、デッド インターバル、認証キーなど）については、接続されたネットワーク内のすべてのルータで統一性を維持する必要があります。これらのパラメータを変更した場合は、ネットワーク内のすべてのルータの値も同様に変更してください。

（注） **ip ospf** インターフェイス コンフィギュレーション コマンドはすべてオプションです。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** interface-id 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 3	**ip ospf** cost 例： `Device(config-if)# ip ospf 8`	（任意）インターフェイスでパケットを送信するコストを明示的に指定します。
ステップ 4	**ip ospf retransmit-interval** seconds 例： `Device(config-if)# ip ospf transmit-interval 10`	（任意）LSA 送信間隔を秒数で指定します。指定できる範囲は 1 ～ 65535 秒です。デフォルト値は 5 秒です。
ステップ 5	**ip ospf transmit-delay** seconds 例： `Device(config-if)# ip ospf transmit-delay 2`	（任意）リンク ステートアップデート パケットを送信するまでの予測待機時間を秒数で設定します。指定できる範囲は 1 ～ 65535 秒です。デフォルト値は 1 秒です。

	コマンドまたはアクション	目的
ステップ 6	**ip ospf priority** number 例： `Device(config-if)# ip ospf priority 5`	（任意）ネットワークに対して、OSPF で指定されたルータを検索するときに役立つ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。
ステップ 7	**ip ospf hello-interval** seconds 例： `Device(config-if)# ip ospf hello-interval 12`	（任意）OSPF インターフェイスで hello パケットの送信間隔を秒数で設定します。ネットワークのすべてのノードで同じ値を指定する必要があります。指定できる範囲は 1 ～ 65535 秒です。デフォルトは 10 秒です。
ステップ 8	**ip ospf dead-interval** seconds 例： `Device(config-if)# ip ospf dead-interval 8`	（任意）最後のデバイスで hello パケットが確認されてから、OSPF ルータがダウンしていることがネイバーによって宣言されるまでの時間を秒数で設定します。ネットワークのすべてのノードで同じ値を指定する必要があります。指定できる範囲は 1 ～ 65535 秒です。デフォルト値は hello インターバルの 4 倍です。
ステップ 9	**ip ospf authentication-key** キー 例： `Device(config-if)# ip ospf authentication-key password`	（任意）隣接 OSPF ルータで使用されるパスワードを割り当てます。パスワードには、キーボードから入力した任意の文字列（最大 8 バイト長）を指定できます。同じネットワーク上のすべての隣接ルータには、OSPF 情報を交換するため、同じパスワードを設定する必要があります。
ステップ 10	**ip ospf message digest-key** keyid **md5** key 例： `Device(config-if)# ip ospf message digest-key 16 md5 your1pass`	（任意）MDS 認証をイネーブルにします。 • *keyid*：1 ～ 255 の ID。 • *key*：最大 16 バイトの英数字パスワード
ステップ 11	**ip ospf database-filter all out** 例： `Device(config-if)# ip ospf database-filter all out`	（任意）インターフェイスへの OSPF LSA パケットのフラッディングを阻止します。デフォルトでは、OSPF は、LSA が到着したインターフェイスを除き、同じエリア内のすべてのインターフェイスで新しい LSA をフラッドします。
ステップ 12	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 13	**show ip ospf interface** [*interface-name*] 例： `Device# show ip ospf interface`	OSPF に関連するインターフェイス情報を表示します。
ステップ 14	**show ip ospf neighbor detail** 例： `Device# show ip ospf neighbor detail`	ネイバー スイッチの NSF 認証ステータスを表示します。出力には、次のいずれかが表示されます。 • *Options is 0x52* *LLS Options is 0x1 (LR)* これらの行の両方が表示される場合、ネイバー スイッチが NSF 認識です。 • *Options is 0x42*：ネイバー スイッチが NSF 認識でないことを示します。
ステップ 15	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

OSPF エリア パラメータの設定

はじめる前に

（注） OSPF **area** ルータ コンフィギュレーション コマンドはすべて任意です。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router ospf** process-id 例： `Device(config)# router ospf 109`	OSPF ルーティングをイネーブルにし、ルータ コンフィギュレーション モードを開始します。
ステップ 3	**area** *area-id***authentication** 例： `Device(config-router)# area 1 authentication`	（任意）特定のエリアへの無許可アクセスに対して、パスワードベースの保護を可能にします。ID には 10 進数または IP アドレスのいずれかを指定できます。
ステップ 4	**area** *area-id***authentication message-digest** 例： `Device(config-router)# area 1 authentication message-digest`	（任意）エリアに関して MD5 認証をイネーブルにします。
ステップ 5	**area** *area-id***stub** [**no-summary**] 例： `Device(config-router)# area 1 stub`	（任意）エリアをスタブ エリアとして定義します。**no-summary** キーワードを指定すると、ABR はサマリー リンク アドバタイズメントをスタブ エリアに送信できなくなります。
ステップ 6	**area** *area-id***nssa** [**no-redistribution**] [**default-information-originate**] [**no-summary**] 例： `Device(config-router)# area 1 nssa default-information-originate`	（任意）エリアを NSSA として定義します。同じエリア内のすべてのルータは、エリアが NSSA であることを認識する必要があります。次のキーワードのいずれかを選択します。 • **no-redistribution**：ルータが NSSA ABR の場合、**redistribute** コマンドを使用して、ルートを NSSA エリアでなく通常のエリアに取り込む場合に使用します。 • **default-information-originate**：LSA タイプ 7 を NSSA に取り込めるようにする場合に、ABR で選択します。 • **no-redistribution**：サマリー LSA を NSSA に送信しない場合に選択します。

	コマンドまたはアクション	目的
ステップ 7	**area** *area-id***range** *address mask* 例： `Device(config-router)# area 1 range 255.240.0.0`	（任意）単一のルートをアドバタイズするアドレス範囲を指定します。このコマンドは、ABR に対してだけ使用します。
ステップ 8	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 9	**show ip ospf** [*process-id*] 例： `Device# show ip ospf`	設定を確認するため、一般的な OSPF ルーティング プロセスまたは特定のプロセス ID に関する情報を表示します。
ステップ 10	**show ip ospf** [*process-id* [*area-id*]] **database** 例： `Device# show ip osfp database`	特定のルータの OSPF データベースに関連する情報のリストを表示します。
ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

その他の OSPF パラメータの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**router ospf** *process-id* 例： `Device(config)# router ospf 10`	OSPF ティン ネー し、ル コン レー モー しま
ステップ 3	**summary-address** *address mask* 例： `Device(config)# summary-address 10.1.1.1 255.255.255.0`	（任意 のサ ルー アド され に、再 れた アド び IP ネッ クを指 す。
ステップ 4	**area** *area-id router-id* [*seconds*] [*seconds*] [] [[*key*]	*keyid key*]]**virtual-linkhello-intervalretransmit-intervaltransauthentication-keymessage-digest-keymd5** 例： `Device(config)# area 2 virtual-link 192.168.255.1 hello-interval 5`	（任意 リンク し、ル タを設 す。
ステップ 5	**default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** *type-value*] [**route-map** *map-name*] 例： `Device(config)# default-information originate metric 100 metric-type 1`	（任意 的に C ルーテ ドメイ フォル ルー する。 ASBR しま メー て任意
ステップ 6	**ip ospf name-lookup** 例： `Device(config)# ip ospf name-lookup`	（任意 名検索 しま フォル

	コマンドまたはアクション	目的
		ディセー になって す。
ステップ 7	**ip auto-cost reference-bandwidth** *ref-bw* 例： `Device(config)# ip auto-cost reference-bandwidth 5`	（任意） のルート ドバタイ るアドレ 囲を指定 す。この ンドは、 に対して 使用しま
ステップ 8	**distance ospf** {[**inter-area** *dist1*] [**inter-area** *dist2*] [**external** *dist3*]} 例： `Device(config)# distance ospf inter-area 150`	（任意） OSPF の距 の値を変 ます。各 プのルー デフォル 離は 110 す。指定 る範囲は 255 です。
ステップ 9	**passive-interface** *type number* 例： `Device(config)# passive-interface gigabitethernet 1/0/6`	（任意） されたイ ターフェ 経由の ho パケット 信を抑制 す。
ステップ 10	**timers throttle spf** *spf-delay spf-holdtime spf-wait* 例： `Device(config)# timers throttle spf 200 100 100`	（任意） ト計算タ マーを設 ます。 • *spf-de* SPF 算の 更を 信す 間の 延。

	コマンドまたはアクション	目的
ステップ 11	**ospf log-adj-changes** 例： `Device(config)# ospf log-adj-changes`	（任意 バー トが変 たとき syslog セーシ しま

	コマンドまたはアクション	目的
ステップ 12	**end** 例： `Device(config)# end`	特権 EXE モードに ます。
ステップ 13	**show ip ospf** [*process-id* [*area-id*]] **database** 例： `Device# show ip ospf database`	特定のル の OSPF タベース 連する情 リストを します。
ステップ 14	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意） フィギュ ション フ イルに設 保存しま

LSA グループ ペーシングの変更

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router ospf** *process-id* 例： `Device(config)# router ospf 25`	OSPF ルーティングをイネーブルにし、ルータ コンフィギュレーション モードを開始します。
ステップ 3	**timers lsa-group-pacing** *seconds* 例： `Device(config-router)# timers lsa-group-pacing 15`	LSA の グループ ペーシングを変更します。

	コマンドまたはアクション	目的
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ループバック インターフェイスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface loopback 0** 例： `Device(config)# interface loopback 0`	ループバック インターフェイスを作成し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ip address** address mask 例： `Device(config-if)# ip address 10.1.1.5 255.255.240.0`	このインターフェイスに IP アドレスを割り当てます。

	コマンドまたはアクション	目的
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show ip interface** 例： `Device# show ip interface`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

OSPF のモニタリング

IP ルーティング テーブルの内容、キャッシュの内容、およびデータベースの内容など、特定の統計情報を表示できます。

表 *6*：*IP OSPF* 統計情報の表示コマンド

show ip ospf [*process-id*]	OSPF ルーティング プロセスに関する一般情報を表示します。
show ip ospf [*process-id*] **database** [**router**] [*link-state-id*] **show ip ospf** [*process-id*] **database** [**router**] [**self-originate**] **show ip ospf** [*process-id*] **database** [**router**] [**adv-router** [*ip-address*]] **show ip ospf** [*process-id*] **database** [**network**] [*link-state-id*] **show ip ospf** [*process-id*] **database** [**summary**] [*link-state-id*] **show ip ospf** [*process-id*] **database** [**asbr-summary**] [*link-state-id*] **show ip ospf** [*process-id*] **database** [**external**] [*link-state-id*] **show ip ospf** [*process-id area-id*] **database** [**database-summary**]	OSPF データベースに関連する情報のリストを表示します。

show ip ospf border-routes	内部の OSPF ルーティング ABR および ASBR テーブル エントリを表示します。
show ip ospf interface [*interface-name*]	OSPF に関連するインターフェイス情報を表示します。
show ip ospf neighbor [*interface-name*] [*neighbor-id*] **detail**	OSPF インターフェイス ネイバー情報を表示します。
show ip ospf virtual-links	OSPF に関連する仮想リンク情報を表示します。

OSPF の設定例

例：基本的な OSPF パラメータの設定

次に、OSPF ルーティング プロセスを設定し、プロセス番号 109 を割り当てる例を示します。

```
Device(config)# router ospf 109
Device(config-router)# network 131.108.0.0 255.255.255.0 area 24
```

EIGRP に関する情報

EIGRP は IGRP のシスコ独自の拡張バージョンです。EIGRP は IGRP と同じディスタンス ベクトル アルゴリズムおよび距離情報を使用しますが、EIGRP では収束性および動作効率が大幅に改善されています。

コンバージェンス テクノロジーには、拡散更新アルゴリズム（DUAL）と呼ばれるアルゴリズムが採用されています。DUAL を使用すると、ルート計算の各段階でループが発生しなくなり、トポロジの変更に関連するすべてのデバイスを同時に同期できます。トポロジ変更の影響を受けないルータは、再計算に含まれません。

IP EIGRP を導入すると、ネットワークの幅が広がります。RIP の場合、ネットワークの最大幅は 15 ホップです。EIGRP メトリックは数千ホップをサポートするほど大きいため、ネットワークを拡張するときに問題となるのは、トランスポート レイヤのホップ カウンタだけです。IP パケットが 15 台のルータを経由し、宛先方向のネクスト ホップが EIGRP によって取得されている場合だけ、EIGRP は転送制御フィールドの値を増やします。RIP ルートを宛先へのネクスト ホップとして使用する場合、転送制御フィールドでは、通常どおり値が増加します。

EIGRP の機能

EIGRP には次の機能があります。

- 高速コンバージェンス
- 差分更新：宛先のステートが変更された場合、ルーティングテーブルの内容全体を送信する代わりに差分更新を行い、EIGRP パケットに必要な帯域幅を最小化します。
- 低い CPU 使用率：完全更新パケットを受信ごとに処理する必要がないため、CPU 使用率が低下します。
- プロトコルに依存しないネイバー探索メカニズム：このメカニズムを使用し隣接ルータに関する情報を取得します。
- 可変長サブネット マスク（VLSM）
- 任意のルート集約
- 大規模ネットワークへの対応

EIGRP コンポーネント

EIGRP には次に示す 4 つの基本コンポーネントがあります。

- ネイバー探索および回復：直接接続されたネットワーク上の他のルータに関する情報を動的に取得するために、ルータで使用されるプロセスです。また、ネイバーが到達不能または動作不能になっていることを検出するためにも使用されます。ネイバー探索および回復は、サイズの小さな hello パケットを定期的に送信することにより、わずかなオーバーヘッドで実現されます。hello パケットが受信されているかぎり、Cisco ISO ソフトウェアは、ネイバーが有効に機能していると学習します。このように判別された場合、隣接ルータはルーティング情報を交換できます。
- Reliable Transport Protocol：EIGRP パケットをすべてのネイバーに確実に、順序どおりに配信します。マルチキャスト パケットとユニキャスト パケットが混在した伝送もサポートされます。EIGRP パケットには確実に送信する必要があるものと、そうでないものがあります。効率化のため、信頼性は必要時にのみ提供されます。たとえば、マルチキャスト機能があるマルチアクセス ネットワーク（イーサネットなど）では、すべてのネイバーにそれぞれ hello パケットを確実に送信する必要はありません。そのため、EIGRP は、1 つのマルチキャスト hello を送信し、パケットに確認応答が必要ないという通知をそのパケットに含めます。他のタイプのパケット（アップデートなど）の場合は、確認応答（ACK パケット）を要求します。信頼性の高い伝送であれば、ペンディング中の未確認応答パケットがある場合、マルチキャスト パケットを迅速に送信できます。このため、リンク速度が変化する場合でも、コンバージェンス時間を短く保つことができます。
- DUAL 有限状態マシンには、すべてのルート計算の決定プロセスが組み込まれており、すべてのネイバーによってアドバタイズされたすべてのルートが追跡されます。DUAL は距離情報（メトリックともいう）を使用して、効率的な、ループのないパスを選択し、さらに DUAL

は適切な後継ルータに基づいて、ルーティング テーブルに挿入するルートを選択します。後継ルータは、宛先への最小コスト パス（ルーティング ループに関連しないことが保証されている）を持つ、パケット転送に使用される隣接ルータです。適切な後継ルータが存在しなくても、宛先にアドバタイズするネイバーが存在する場合は再計算が行われ、この結果、新しい後継ルータが決定されます。ルートの再計算に要する時間によって、コンバージェンス時間が変わります。再計算はプロセッサに負荷がかかるため、必要でない場合は、再計算しないようにしてください。トポロジが変更されると、DUALはフィジブルサクセサの有無を調べます。適切なフィジブルサクセサが存在する場合は、それらを探して使用し、不要な再計算を回避します。

- プロトコル依存モジュールは、ネットワーク層プロトコル固有のタスクを実行します。たとえば、IP EIGRP モジュールは、IP でカプセル化された EIGRP パケットを送受信します。また、EIGRPパケットを解析したり、DUALに受信した新しい情報を通知したりします。EIGRP は DUAL にルーティング決定を行うように要求しますが、結果は IP ルーティング テーブルに格納されます。EIGRP は、他の IP ルーティング プロトコルによって取得したルートの再配信も行います。

（注） EIGRP をイネーブルにするには、スイッチ またはスタック マスター上で IP サービス フィーチャ セットが稼働している必要があります。

EIGRP NSF

スイッチ スタックは、次の 2 つのレベルの EIGRP ノンストップ フォワーディングをサポートします。

- EIGRP NSF 認識
- EIGRP NSF 対応

EIGRP NSF 認識

IP サービス フィーチャ セットは、EIGRP NSF 認識を IPv4 に対してサポートしています。隣接ルータが NSF 対応である場合、レイヤ 3 スイッチでは、ルータに障害が発生してプライマリ RP がバックアップ RP によって引き継がれる間、または処理を中断させずにソフトウェア アップグレードを行うためにプライマリ RP を手動でリロードしている間、隣接ルータからパケットを転送し続けます。

この機能をディセーブルにできません。この機能の詳細については、『*Cisco IOS IP Routing Protocols Configuration Guide, Release 12.4*』の「EIGRP Nonstop Forwarding (NSF) Awareness」を参照してください。

EIGRP NSF 対応

IP サービス フィーチャ セットでは、EIGRP Cisco NSF ルーティングがサポートされています。それにより、コンバージェンスの時間が短くなり、スタック マスター変更後のトラフィック損失がなくなります。この NSF 機能の詳細については、『*High Availability Configuration Guide, Cisco IOS XE Release 3S*』の「Configuring Nonstop Forwarding」を参照してください。

IP サービス フィーチャ セットは、EIGRP NSF 対応ルーティングも IPv4 に対してサポートし、スタック マスター変更後のコンバージェンスの向上と、トラフィック損失の低減を実現します。EIGRP NSF 対応のスタック マスターが再起動したとき、または新しいスタック マスターが起動してNSFが再起動したとき、このスイッチにはネイバーが存在せず、トポロジテーブルは空の状態です。スイッチは、スイッチ スタックに対するトラフィックを中断することなく、インターフェイスの起動、ネイバーの再取得、およびトポロジテーブルとルーティングテーブルの再構築を行う必要があります。EIGRPピアルータは新しいスタックマスターから学習したルートを維持し、NSF の再起動処理の間トラフィックの転送を継続します。

ネイバーによる隣接リセットを防ぐために、新しいスタックマスターはEIGRPパケットヘッダーの新しいRestart（RS）ビットを使用して再起動を示します。これを受信したネイバーは、ピア リスト内のスタックと同期を取り、スタックとの隣接関係を維持します。続いてネイバーは、RS ビットがセットされているスタック マスターにトポロジ テーブルを送信して、自身が NSF 認識デバイスであることおよび新しいスタック マスターを補助していることを示します。

スタックのピア ネイバーの少なくとも 1 つが NFS 認識デバイスであれば、スタック マスターはアップデート情報を受信してデータベースを再構築します。各 NSF 認識ネイバーは、最後のアップデート パケットに End of Table（EOT）マーカーを付けて送信して、テーブル情報の最後であることを示します。スタック マスターは、EOTマーカーを受信したときにコンバージェンスを認識し、続いてアップデートの送信を始めます。スタック マスターがネイバーからすべてのEOTマーカーを受信した場合、または NSF コンバージ タイマーが期限切れになった場合、EIGRP は RIB にコンバージェンスを通知し、すべてのNSF認識ピアにトポロジテーブルをフラッディングします。

EIGRP スタブ ルーティング

EIGRP スタブ ルーティング機能は、すべてのフィーチャ セットで使用でき、エンド ユーザの近くにルーテッド トラフィックを移動することでリソースの利用率を低減させます。

（注） IP Base フィーチャ セットに含まれる EIGRP スタブ ルーティング機能では、ルーティング テーブルからの接続ルートまたはサマリー ルートをネットワーク内の他のスイッチにアドバタイズすることだけを行います。スイッチはアクセス レイヤで EIGRP スタブ ルーティングを使用することにより、ほかのタイプのルーティング アドバタイズメントの必要性を排除しています。拡張機能および完全な EIGRP ルーティングを使用するには、スイッチで IP ベース フィーチャ セットを稼働させる必要があります。IP ベース フィーチャ セットが稼働するスイッチ上で、Multi-VRF-CE と EIGRP スタブ ルーティングを同時に設定しようとすると、設定は許可されません。IPv6 EIGRP スタブ ルーティングは、IP ベース フィーチャ セットではサポートされません。

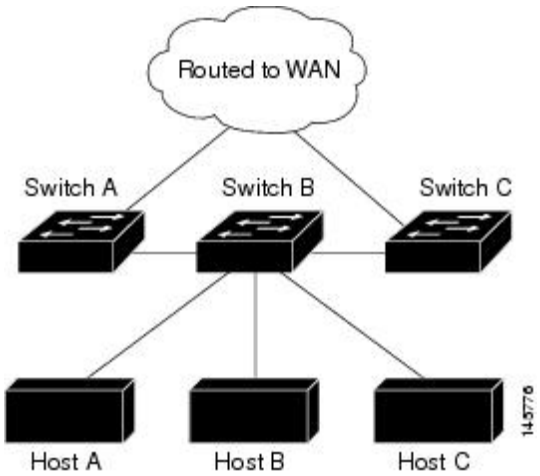
EIGRP スタブ ルーティングを使用するネットワークでは、ユーザに対する IP トラフィックの唯一の許容ルートは、EIGRP スタブ ルーティングを設定しているスイッチ経由です。スイッチは、ユーザ インターフェイスとして設定されているインターフェイスまたは他のデバイスに接続されているインターフェイスにルーテッド トラフィックを送信します。

EIGRP スタブ ルーティングを使用しているときは、EIGRP を使用してスイッチだけをスタブとして設定するように、分散ルータおよびリモート ルータを設定する必要があります。指定したルートだけがスイッチから伝播されます。スイッチは、サマリー、接続ルート、およびルーティング アップデートに対するすべてのクエリーに応答します。

スタブ ルータの状態を通知するパケットを受信した隣接ルータは、ルートについてはスタブ ルータに照会しません。また、スタブ ピアを持つルータは、そのピアについては照会しません。スタブ ルータは、ディストリビューション ルータを使用して適切なアップデートをすべてのピアに送信します。

次の図では、スイッチ B は EIGRP スタブ ルータとして設定されています。スイッチ A および C は残りの WAN に接続されています。スイッチ B は、接続ルート、スタティック ルート、再配信ルート、およびサマリー ルートをスイッチ A と C にアドバタイズします。スイッチ B はスイッチ A から学習したルートをアドバタイズしません（逆の場合も同様です）。

図 *4*：*EIGRP* スタブ ルータ設定



EIGRP スタブ ルーティングの詳細については、『*Cisco IOS IP Configuration Guide, Volume 2 of 3: Routing Protocols*』の「Configuring EIGRP Stub Routing」の項を参照してください。

EIGRP の設定方法

EIGRP ルーティング プロセスを作成するには、EIGRP をイネーブルにし、ネットワークを関連付ける必要があります。EIGRP は指定されたネットワーク内のインターフェイスにアップデートを送信します。インターフェイス ネットワークを指定しないと、どの EIGRP アップデートでもアドバタイズされません。

（注） ネットワーク上に IGRP 用に設定されているルータがあり、この設定を EIGRP に変更する場合は、IGRP と EIGRP の両方が設定された移行ルータを指定する必要があります。この場合は、この次の項に記載されているステップ 1 ～ 3 を実行し、さらに「スプリット ホライゾンの設定」も参照してください。ルートを自動的に再配信するには、同じ AS 番号を使用する必要があります。

EIGRP のデフォルト設定

表 *7*：*EIGRP* のデフォルト設定

機能	デフォルト設定
自動サマリー	ディセーブル
デフォルト情報	再配信中は外部ルートが許可され、EIGRP プロセス間でデフォルト情報が渡されます。
デフォルト メトリック	デフォルト メトリックなしで再配信できるのは、接続されたルートおよびインターフェイスのスタティック ルートだけです。デフォルト メトリックは次のとおりです。 • 帯域幅：0 以上の kb/s • 遅延（10 マイクロ秒）：0 または 39.1 ナノ秒の倍数である任意の正の数値 • 信頼性：0 ～ 255 の任意の数値（255 の場合は信頼性が 100%） • 負荷：0 ～ 255 の数値で表される有効帯域幅（255 の場合は 100% の負荷） • MTU：バイトで表されたルートの MTU サイズ（0 または任意の正の整数）

機能	デフォルト設定
ディスタンス	内部距離：90 外部距離：170
EIGRP の隣接関係変更ログ	ディセーブル隣接関係の変更はロギングされません。
IP 認証キーチェーン	認証なし
IP 認証モード	認証なし
IP 帯域幅比率	50%
IP hello 間隔	低速非ブロードキャストマルチアクセス（NBMA）ネットワークの場合：60 秒、それ以外のネットワークの場合：5 秒
IP ホールドタイム	低速 NBMA ネットワークの場合：180 秒、それ以外のネットワークの場合：15 秒
IP スプリットホライズン	イネーブル
IP サマリー アドレス	サマリー集約アドレスは未定義
メトリック重み	tos：0、k1 および k3：1、k2、k4、および k5：0
ネットワーク	指定なし
ノンストップ フォワーディング（NSF）認識	IP サービス フィーチャ セットを実行するスイッチ上で IPv4 対してイネーブルになっています。レイヤ 3 スイッチでは、ハードウェアやソフトウェアの変更中に、隣接する NSF 対応ルータからのパケットを転送し続けることができます。
NSF 対応	ディセーブル （注） スイッチは EIGRP NSF 対応ルーティングを IPv4 に対してサポートします。
オフセットリスト	ディセーブル
ルータ EIGRP	ディセーブル
メトリック設定	ルート マップにはメトリック設定なし
トラフィック共有	メトリックの比率に応じて配分

機能	デフォルト設定
バリアンス	1（等コスト ロード バランシング）

基本的な EIGRP パラメータの設定

はじめる前に

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router eigrp** autonomous-system 例： `Device(config)# router eigrp 10`	EIGRP ルーティング プロセスをイネーブルにし、ルータ コンフィギュレーション モードを開始します。AS 番号によって他の EIGRP ルータへのルートを特定し、ルーティング情報をタグ付けします。
ステップ 3	**nsf** 例： `Device(config)# nsf`	（任意）EIGRP NSF をイネーブルにします。スタック マスターおよびそのすべてのピア上でこのコマンドを入力します。
ステップ 4	**network** network-number 例： `Device(config)# network 192.168.0.0`	ネットワークを EIGRP ルーティング プロセスに関連付けます。EIGRP は指定されたネットワーク内のインターフェイスにアップデートを送信します。
ステップ 5	**eigrp log-neighbor-changes** 例： `Device(config)# eigrp log-neighbor-changes`	（任意）EIGRP 隣接関係変更のロギングをイネーブルにし、ルーティング システムの安定性をモニタします。
ステップ 6	**metric weights** *tos k1 k2 k3 k4 k5* 例： `Device(config)# metric weights 0 2 0 2 0 0`	（任意）EIGRP メトリックを調整します。デフォルト値はほとんどのネットワークで適切に動作するよう入念に設定されていますが、調整することも可能です。

	コマンドまたはアクション	目的
		注意 メトリックを設定する作業は複雑です。熟練したネットワーク設計者の指導がない場合は、行わないでください。
ステップ 7	**offset-list** [*access-list number* | *name*] {**in** | **out**} *offset* [*type number*] 例： `Device(config)# offset-list 21 out 10`	（任意）オフセット リストをルーティング メトリックに適用し、EIGRP によって取得したルートへの着信および発信メトリックを増加します。アクセス リストまたはインターフェイスを使用し、オフセット リストを制限できます。
ステップ 8	**auto-summary** 例： `Device(config)# auto-summary`	（任意）ネットワーク レベル ルートへのサブネット ルートの自動サマライズをイネーブルにします。
ステップ 9	**ip summary-address eigrp** *autonomous-system-number address mask* 例： `Device(config)# ip summary-address eigrp 1 192.168.0.0 255.255.0.0`	（任意）サマリー集約を設定します。
ステップ 10	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 11	**show ip protocols** 例： `Device# show ip protocols`	入力を確認します。 NSF 認識の場合、出力に次のように表示されます。 *** IP Routing is NSF aware *** EIGRP NSF enabled
ステップ 12	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

EIGRP インターフェイスの設定

インターフェイスごとに、他の EIGRP パラメータを任意で設定できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 3	**ip bandwidth-percent eigrp** *percent* 例： `Device(config-if)# ip bandwidth-percent eigrp 60`	（任意）インターフェイスでEIGRPが使用できる帯域幅の割合を設定します。デフォルト値は 50% です。
ステップ 4	**ip summary-address eigrp** *autonomous-system-number address mask* 例： `Device(config-if)# ip summary-address eigrp 109 192.161.0.0 255.255.0.0`	（任意）指定されたインターフェイスのサマリー集約アドレスを設定します（auto-summary がイネーブルの場合は、通常設定する必要はありません）。
ステップ 5	**ip hello-interval eigrp** *autonomous-system-number seconds* 例： `Device(config-if)# ip hello-interval eigrp 109 10`	（任意）EIGRP ルーティング プロセスの hello タイムインターバルを変更します。指定できる範囲は 1 ～ 65535 秒です。低速 NBMA ネットワークの場合のデフォルト値は60秒、その他のすべてのネットワークでは 5 秒です。
ステップ 6	**ip hold-time eigrp** *autonomous-system-number seconds* 例： `Device(config-if)# ip hold-time eigrp 109 40`	（任意）EIGRPルーティングプロセスのホールドタイムインターバルを変更します。指定できる範囲は1 ～ 65535 秒です。低速NBMA ネットワークの場合のデフォルト値は 180 秒、その他のすべてのネットワークでは 15 秒です。

	コマンドまたはアクション	目的
		注意 ホールド タイムを調整する前に、シスコのテクニカルサポートにお問い合わせください。
ステップ 7	**no ip split-horizon eigrp** *autonomous-system-number* 例： `Device(config-if)# no ip split-horizon eigrp 109`	（任意）スプリットホライズンをディセーブルにし、ルート情報が情報元インターフェイスからルータによってアドバタイズされるようにします。
ステップ 8	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 9	**show ip eigrp interface** 例： `Device# show ip eigrp interface`	EIGRP がアクティブであるインターフェイス、およびそれらのインターフェイスに関連する EIGRP の情報を表示します。
ステップ 10	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

EIGRP ルート認証の設定

EIGRP ルート認証を行うと、EIGRP ルーティング プロトコルからのルーティング アップデートに関する MD5 認証が可能になり、承認されていない送信元から無許可または問題のあるルーティング メッセージを受け取ることがなくなります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** interface-id 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 3	**ip authentication mode eigrp** *autonomous-system***md5** 例： `Device(config-if)# ip authentication mode eigrp 104 md5`	IP EIGRP パケットの MD5 認証をイネーブルにします。
ステップ 4	**ip authentication key-chain eigrp** *autonomous-system key-chain* 例： `Device(config-if)# ip authentication key-chain eigrp 105 chain1`	IP EIGRP パケットの認証をイネーブルにします。
ステップ 5	**exit** 例： `Device(config-if)# exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 6	**key chain** *name-of-chain* 例： `Device(config)# key chain chain1`	キー チェーンを識別し、キーチェーン コンフィギュレーション モードを開始します。ステップ 4 で設定した名前を指定します。
ステップ 7	**key** *number* 例： `Device(config-keychain)# key 1`	キーチェーン コンフィギュレーション モードで、キー番号を識別します。

	コマンドまたはアクション	目的
ステップ 8	**key-string** *text* 例： `Device(config-keychain-key)# key-string key1`	キーチェーン コンフィギュレーション モードで、キー ストリングを識別します。
ステップ 9	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `Device(config-keychain-key)# accept-lifetime 13:30:00 Jan 25 2011 duration 7200`	（任意）キーを受信できる期間を指定します。 *start-time* および *end-time* 構文には、*hh:mm:ss Month date year* または *hh:mm:ss date Month year* のいずれかを使用できます。デフォルトは、デフォルトの *start-time* 以降、無制限です。指定できる最初の日付は 1993 年 1 月 1 日です。デフォルトの *end-time* および **duration** は **infinite** です。
ステップ 10	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `Device(config-keychain-key)# send-lifetime 14:00:00 Jan 25 2011 duration 3600`	（任意）キーを送信できる期間を指定します。 *start-time* および *end-time* 構文には、*hh:mm:ss Month date year* または *hh:mm:ss date Month year* のいずれかを使用できます。デフォルトは、デフォルトの *start-time* 以降、無制限です。指定できる最初の日付は 1993 年 1 月 1 日です。デフォルトの *end-time* および **duration** は **infinite** です。
ステップ 11	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 12	**show key chain** 例： `Device# show key chain`	認証キーの情報を表示します。
ステップ 13	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

EIGRP のモニタリングおよびメンテナンス

ネイバーテーブルからネイバーを削除できます。さらに、各種EIGRPルーティング統計情報を表示することもできます。下の図に、ネイバーを削除し、統計情報を表示する特権 EXEC コマンドを示します。表示されるフィールドの詳細については、『*Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.4*』を参照してください。

表 *8*： *IP EIGRP* の *clear* および *show* コマンド

clear ip eigrp neighbors [*if-address* | *interface*]	ネイバーテーブルからネイバーを削除します。
show ip eigrp interface [*interface*] [*as number*]	EIGRP に設定されているインターフェイスに関する情報を表示します。
show ip eigrp neighbors [*type-number*]	EIGRP によって検出されたネイバーを表示します。
show ip eigrp topology [*autonomous-system-number*] | [[*ip-address*] *mask*]]	指定されたプロセスの EIGRP トポロジ テーブルを表示します。
show ip eigrp traffic [*autonomous-system-number*]	すべてまたは指定された EIGRP プロセスの送受信パケット数を表示します。

BGP に関する情報

ボーダー ゲートウェイ プロトコル（BGP）は、Exterior Gateway Protocol です。自律システム間で、ループの発生しないルーティング情報交換を保証するドメイン間ルーティング システムを設定するために使用されます。自律システムは、同じ管理下で動作して RIP や OSPF などの Interior Gateway Protocol（IGP）を境界内で実行し、Exterior Gateway Protocol（EGP）を使用して相互接続されるルータで構成されます。BGP バージョン 4 は、インターネット内でドメイン間ルーティングを行うための標準 EGP です。このプロトコルは、RFC 1163、1267、および 1771 で定義されています。BGP の詳細については、『*Internet Routing Architectures*』（Cisco Press 刊）、および『*Cisco IP and IP Routing Configuration Guide*』の「Configuring BGP」を参照してください。

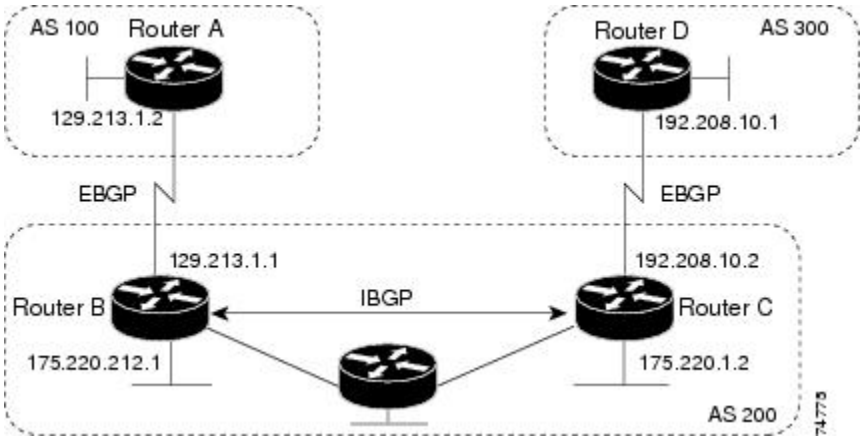
BGP コマンドおよびキーワードの詳細については、『*Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols*』の「IP Routing Protocols」を参照してください。

BGP ネットワーク トポロジ

同じ自律システム（AS）に属し、BGPアップデートを交換するルータは内部BGP（IBGP）を実行し、異なる自律システムに属し、BGP アップデートを交換するルータは外部 BGP（EBGP）を実

行します。大部分のコンフィギュレーション コマンドは、EBGP と IBGP で同じですが、ルーティング アップデートが自律システム間で交換されるか（EBGP）、または AS 内で交換されるか（IBGP）という点で異なります。下の図に、EBGP と IBGP の両方を実行しているネットワークを示します。

図 *5*：*EBGP*、*IBGP*、および複数の自律システム



外部 AS と情報を交換する前に、BGP は AS 内のルータ間で内部 BGP ピアリングを定義し、IGRP や OSPF など AS 内で稼働する IGP に BGP ルーティング情報を再配信して、AS 内のネットワークに到達することを確認します。

BGP ルーティング プロセスを実行するルータは、通常 BGP スピーカーと呼ばれます。BGP はトランスポート プロトコルとして伝送制御プロトコル（TCP）を使用します（特にポート 179）。ルーティング情報を交換するため相互に TCP 接続された 2 つの BGP スピーカーを、ピアまたはネイバーと呼びます。上の図では、ルータ A と B が BGP ピアで、ルータ B と C、ルータ C と D も同様です。ルーティング情報は、宛先ネットワークへの完全パスを示す一連の AS 番号です。BGP はこの情報を使用し、ループのない自律システム マップを作成します。

このネットワークの特徴は次のとおりです。

- ルータ A および B では EBGP が、ルータ B および C では IBGP が稼働しています。EBGP ピアは直接接続されていますが、IBGP ピアは直接接続されていないことに注意してください。IGP が稼働し、2 つのネイバーが相互に到達するかぎり、IBGP ピアを直接接続する必要はありません。
- AS 内のすべての BGP スピーカーは、相互にピア関係を確立する必要があります。つまり、AS 内の BGP スピーカーは、論理的な完全メッシュ型に接続する必要があります。BGP4 は、論理的な完全メッシュに関する要求を軽減する 2 つの技術（連合およびルート リフレクタ）を提供します。
- AS 200 は AS 100 および AS 300 の中継 AS です。つまり、AS 200 は AS 100 と AS 300 間でパケットを転送するために使用されます。

BGP ピアは完全な BGP ルーティング テーブルを最初に交換し、差分更新だけを送信します。BGP ピアはキープアライブ メッセージ（接続が有効であることを確認）、および通知メッセージ（エラーまたは特殊条件に応答）を交換することもできます。

BGP の場合、各ルートはネットワーク番号、情報が通過した自律システムのリスト（自律システム パス）、および他のパス属性リストで構成されます。BGP システムの主な機能は、AS パスのリストに関する情報など、ネットワークの到達可能性情報を他の BGP システムと交換することです。この情報は、AS が接続されているかどうかを判別したり、ルーティング ループをプルーニングしたり、AS レベル ポリシー判断を行うために使用できます。

Cisco IOS が稼働しているルータまたはスイッチが IBGP ルートを選択または使用するのは、ネクストホップ ルータで使用可能なルートがあり、IGP から同期信号を受信している（IGP 同期がディセーブルの場合は除く）場合です。複数のルートが使用可能な場合、BGP は属性値に基づいてパスを選択します。BGP 属性については、「BGP 判断属性の設定」の項を参照してください。

BGP バージョン 4 ではクラスレス ドメイン間ルーティング（CIDR）がサポートされているため、集約ルートを作成してスーパーネットを構築し、ルーティング テーブルのサイズを削減できます。CIDR は、BGP 内部のネットワーク クラスの概念をエミュレートし、IP プレフィックスのアドバタイズをサポートします。

NSF 認識

BGP NSF 認識機能は、IP サービス フィーチャ セットで IPv4 に対してサポートされます。。BGP ルーティングでこの機能をイネーブルにするには、グレースフル リスタートをイネーブルにする必要があります。隣接ルータが NSF 対応で、この機能がイネーブルである場合、レイヤ 3 スイッチでは、ルータに障害が発生してプライマリ RP がバックアップ RP によって引き継がれる間、または処理を中断させずにソフトウェア アップグレードを行うためにプライマリ RP を手動でリロードしている間、隣接ルータからパケットを転送し続けます。

この機能の詳細については、『*Cisco IOS IP Routing Protocols Configuration Guide, Release 12.4*』の「BGP Nonstop Forwarding (NSF) Awareness」を参照してください。

BGP ルーティングに関する情報

BGP ルーティングをイネーブルにするには、BGP ルーティング プロセスを確立し、ローカル ネットワークを定義します。BGP はネイバーとの関係を完全に認識する必要があるため、BGP ネイバーも指定する必要があります。

BGP は、内部および外部の 2 種類のネイバーをサポートします。内部ネイバーは同じ AS 内に、外部ネイバーは異なる AS 内にあります。通常の場合、外部ネイバーは相互に隣接し、1 つのサブネットを共有しますが、内部ネイバーは同じ AS 内の任意の場所に存在します。

スイッチではプライベート AS 番号を使用できます。プライベート AS 番号は通常サービス プロバイダーによって割り当てられ、ルートが外部ネイバーにアドバタイズされないシステムに設定されます。プライベート AS 番号の範囲は 64512 ～ 65535 です。AS パスからプライベート AS 番号を削除するように外部ネイバーを設定するには、**neighbor remove-private-as** ルータ コンフィ

ギュレーション コマンドを使用します。この結果、外部ネイバーにアップデートを渡すとき、AS パス内にプライベート AS 番号が含まれている場合は、これらの番号が削除されます。

AS が別の AS からさらに別の AS にトラフィックを渡す場合は、アドバタイズ対象のルートに矛盾が存在しないことが重要です。BGP がルートをアドバタイズしてから、ネットワーク内のすべてのルータがIGPを通してルートを学習した場合、ASは一部のルータがルーティングできなかったトラフィックを受信することがあります。このような事態を避けるため、BGP は IGP が AS に情報を伝播し、BGP が IGP と同期化されるまで、待機する必要があります。同期化は、デフォルトでイネーブルに設定されています。AS が特定の AS から別の AS にトラフィックを渡さない場合、または自律システム内のすべてのルータでBGPが稼働している場合は、同期化をディセーブルにし、IGP内で伝送されるルート数を少なくして、BGPがより短時間で収束するようにします。

ルーティング ポリシーの変更

ピアのルーティング ポリシーには、インバウンドまたはアウトバウンド ルーティング テーブル アップデートに影響する可能性があるすべての設定が含まれます。BGP ネイバーとして定義された 2 台のルータは、BGP 接続を形成し、ルーティング情報を交換します。このあとで BGP フィルタ、重量、距離、バージョン、またはタイマーを変更する場合、または同様の設定変更を行う場合は、BGP セッションをリセットし、設定の変更を有効にする必要があります。

リセットには、ハード リセットとソフト リセットの 2 種類があります。Cisco IOS Release 12.1 以降では、事前に設定を行わなくても、ソフト リセットを使用できます。事前設定なしにソフト リセットを使用するには、両方の BGP ピアでソフト ルート リフレッシュ機能がサポートされていなければなりません。この機能は、ピアによって TCP セッションが確立されたときに送信される OPEN メッセージに格納されてアドバタイズされます。ソフト リセットを使用すると、BGP ルータ間でルート リフレッシュ要求およびルーティング情報を動的に交換したり、それぞれのアウトバウンド ルーティング テーブルをあとで再アドバタイズできます。

- ソフト リセットによってネイバーからインバウンド アップデートが生成された場合、このリセットはダイナミック インバウンド ソフト リセットといいます。
- ソフト リセットによってネイバーに一連のアップデートが送信された場合、このリセットはアウトバウンド ソフト リセットといいます。

ソフト インバウンド リセットが発生すると、新規インバウンド ポリシーが有効になります。ソフトアウトバウンドリセットが発生すると、BGPセッションがリセットされずに、新規ローカルアウトバウンドポリシーが有効になります。アウトバウンドポリシーのリセット中に新しい一連のアップデートが送信されると、新規インバウンド ポリシーも有効になる場合があります。

下の表に、ハード リセットとソフト リセットの利点および欠点を示します。

表 *9*：ハード リセットとソフト リセットの利点および欠点

リセット タイプ	利点	欠点
ハード リセット	メモリ オーバーヘッドが発生しません。	ネイバーから提供されたBGP、IP、およびFIBテーブルのプレフィックスが失われます。推奨しません。

リセット タイプ	利点	欠点
発信ソフト リセット	ルーティング テーブル アップデートが設定、保管されません。	インバウンド ルーティング テーブル アップデートがリセットされません。
ダイナミック インバウンドソフトリセット	BGP セッションおよびキャッシュがクリアされません。 ルーティング テーブル アップデートを保管する必要がなく、メモリ オーバーヘッドが発生しません。	両方の BGP ルータでルート リフレッシュ機能をサポートする必要があります（Cisco IOS Release 12.1 以降）。

BGP 判断属性

BGP スピーカーが複数の自律システムから受信したアップデートが、同じ宛先に対して異なるパスを示している場合、BGP スピーカーはその宛先に到達する最適パスを 1 つ選択する必要があります。選択されたパスはBGPルーティング テーブルに格納され、ネイバーに伝播されます。この判断は、アップデートに格納されている属性値、およびBGPで設定可能な他の要因に基づいて行われます。

BGP ピアはネイバー AS からプレフィックスに対する 2 つの EBGP パスを学習するとき、最適パスを選択してIPルーティング テーブルに挿入します。BGPマルチパス サポートがイネーブルで、同じネイバー自律システムから複数の EBGP パスを学習する場合、単一の最適パスの代わりに、複数のパスがIPルーティング テーブルに格納されます。そのあと、パケット スイッチング中に、複数のパス間でパケット単位または宛先単位のロード バランシングが実行されます。maximum-paths**maximum-paths** ルータ コンフィギュレーション コマンドは、許可されるパス数を制御します。

これらの要因により、BGP が最適パスを選択するために属性を評価する順序が決まります。

1. パスで指定されているネクストホップが到達不能な場合、このアップデートは削除されます。BGP ネクスト ホップ属性（ソフトウェアによって自動判別される）は、宛先に到達するために使用されるネクストホップのIPアドレスです。EBGPの場合、通常このアドレスは**neighbor remote-as router** ルータ コンフィギュレーション コマンドで指定されたネイバーの IP アドレスです。ネクストホップの処理をディセーブルにするには、ルート マップまたは **neighbor next-hop-self** ルータ コンフィギュレーション コマンドを使用します。
2. 最大の重みのパスを推奨します（シスコ独自のパラメータ）。ウェイト属性はルータにローカルであるため、ルーティング アップデートで伝播されません。デフォルトでは、ルータ送信元のパスに関するウェイト属性は 32768 で、それ以外のパスのウェイト属性は 0 です。最大の重みのルートを推奨します。重みを設定するには、アクセス リスト、ルート マップ、または **neighbor weight** ルータ コンフィギュレーション コマンドを使用します。
3. ローカル プリファレンス値が最大のルートを推奨します。ローカル プリファレンスはルーティング アップデートに含まれ、同じ AS 内のルータ間で交換されます。ローカル初期設定属性の

デフォルト値は 100 です。ローカル プリファレンスを設定するには、**bgp default local-preference** ルータ コンフィギュレーション コマンドまたはルート マップを使用します。

4 ローカル ルータ上で稼働する BGP から送信されたルートを推奨します。

5 AS パスが最短のルートを推奨します。

6 送信元タイプが最小のルートを推奨します。内部ルートまたは IGP は、EGP によって学習されたルートよりも小さく、EGP で学習されたルートは、未知の送信元のルートまたは別の方法で学習されたルートよりも小さくなります。

7 想定されるすべてのルートについてネイバー AS が同じである場合は、MED メトリック属性が最小のルートを推奨します。MED を設定するには、ルート マップまたは **default-metric** ルータ コンフィギュレーション コマンドを使用します。IBGP ピアに送信されるアップデートには、MED が含まれます。

8 内部（IBGP）パスより、外部（EBGP）パスを推奨します。

9 最も近い IGP ネイバー（最小の IGP メトリック）を通って到達できるルートを推奨します。ルータは、AS 内の最短の内部パス（BGP のネクストホップへの最短パス）を使用し、宛先に到達するためです。

10 次の条件にすべて該当する場合は、このパスのルートを IP ルーティング テーブルに挿入してください。

最適ルートと目的のルートがともに外部ルートである

最適ルートと目的のルートの両方が、同じネイバー自律システムからのルートである

maximum-paths がイネーブルである

11 マルチパスがイネーブルでない場合は、BGP ルータ ID の IP アドレスが最小であるルートを推奨します。通常、ルータ ID はルータ上の最大の IP アドレスまたはループバック（仮想）アドレスですが、実装に依存することがあります。

ルート マップ

BGP 内でルート マップを使用すると、ルーティング情報を制御、変更したり、ルーティング ドメイン間でルートを再配信する条件を定義できます。ルート マップの詳細については、「Using Route Maps to Redistribute Routing Information」の項を参照してください。各ルート マップには、ルート マップを識別する名前（マップ タグ）およびオプションのシーケンス番号が付いています。

BGP フィルタリング

BGP アドバタイズメントをフィルタリングするには、**as-path access-list** グローバル コンフィギュレーション コマンドや **neighbor filter-list** ルータ コンフィギュレーション コマンドなどの AS パス フィルタを使用します。**neighbor distribute-list** ルータ コンフィギュレーション コマンドとアクセス リストを併用することもできます。distribute-list フィルタはネットワーク番号に適用され

ます。**distribute-list** コマンドの詳細については、「ルーティング アップデートのアドバタイズおよび処理の制御」の項を参照してください。

ネイバー単位でルート マップを使用すると、アップデートをフィルタリングしたり、さまざまな属性を変更したりできます。ルートマップは、インバウンドアップデートまたはアウトバウンドアップデートのいずれかに適用できます。ルート マップを渡すルートだけが、アップデート内で送信または許可されます。着信および発信の両方のアップデートで、ASパス、コミュニティ、およびネットワーク番号に基づくマッチングがサポートされています。AS パスのマッチングには **match as-path access-list** ルート マップ コマンド、コミュニティに基づくマッチングには **match community-list** ルートマップ コマンド、ネットワークに基づくマッチングには **ip access-list** グローバル コンフィギュレーション コマンドが必要です。

BGP フィルタリングのプレフィックス リスト

neighbor distribute-list ルータ コンフィギュレーション コマンドを含む多数の BGP ルート フィルタリング コマンドでは、アクセス リストの代わりにプレフィックス リストを使用できます。プレフィックス リストを使用すると、大規模リストのロードおよび検索パフォーマンスが改善し、差分更新がサポートされ、コマンドライン インターフェイス（CLI）設定が簡素化され、柔軟性が増すなどの利点が生じます。

プレフィックスリストによるフィルタリングでは、アクセスリストの照合の場合と同様に、プレフィックス リストに記載されたプレフィックスとルートのプレフィックスが照合されます。一致すると、一致したルートが使用されます。プレフィックスが許可されるか、または拒否されるかは、次に示すルールに基づいて決定されます。

- 空のプレフィックス リストはすべてのプレフィックスを許可します。
- 特定のプレフィックスがプレフィックス リストのどのエントリとも一致しなかった場合、実質的に拒否されたものと見なされます。
- 指定されたプレフィックスと一致するエントリがプレフィックス リスト内に複数存在する場合は、シーケンス番号が最小であるプレフィックス リスト エントリが識別されます。

デフォルトでは、シーケンス番号は自動生成され、5 ずつ増分します。シーケンス番号の自動生成をディセーブルにした場合は、エントリごとにシーケンス番号を指定する必要があります。シーケンス番号を指定する場合の増分値に制限はありません。増分値が 1 の場合は、このリストに追加エントリを挿入できません。増分値が大きい場合は、値がなくなることがあります。

BGP コミュニティ フィルタリング

BGP コミュニティ フィルタリングは、COMMUNITIES 属性の値に基づいてルーティング情報の配信を制御する BGP の方法の 1 つです。この属性によって、宛先はコミュニティにグループ化され、コミュニティに基づいてルーティング判断が適用されます。この方法を使用すると、ルーティング情報の配信制御を目的とする BGP スピーカーの設定が簡単になります。

コミュニティは、共通するいくつかの属性を共有する宛先のグループです。各宛先は複数のコミュニティに属します。AS管理者は、宛先が属するコミュニティを定義できます。デフォルトでは、

すべての宛先が一般的なインターネット コミュニティに属します。コミュニティは、過渡的でグローバルなオプションの属性である、COMMUNITIES 属性（1 ～ 4294967200 の数値）によって識別されます。事前に定義された既知のコミュニティの一部を、次に示します。

- **internet**：このルートをインターネット コミュニティにアドバタイズします。すべてのルータが所属します。
- **no-export**：EBGP ピアにこのルートをアドバタイズしません。
- **no-advertise**：どのピア（内部または外部）にもこのルートをアドバタイズしません。
- **local-as**：ローカルな AS 外部のピアにこのルートをアドバタイズしません。

コミュニティに基づき、他のネイバーに許可、送信、配信するルーティング情報を制御できます。BGP スピーカーは、ルートを学習、アドバタイズ、または再配信するときに、ルートのコミュニティを設定、追加、または変更します。ルートを集約すると、作成された集約内の COMMUNITIES 属性に、すべての初期ルートの全コミュニティが含まれます。

コミュニティ リストを使用すると、ルート マップの match 句で使用されるコミュニティ グループを作成できます。さらに、アクセス リストの場合と同様、一連のコミュニティ リストを作成することもできます。ステートメントは一致が見つかるまでチェックされ、1 つのステートメントが満たされると、テストは終了します。

コミュニティに基づいて COMMUNITIES 属性および match 句を設定するには、「ルート マップによるルーティング情報の再配信」に記載されている **match community-list** および **set community** ルート マップ コンフィギュレーション コマンドを参照してください。

BGP ネイバーおよびピア グループ

通常、BGP ネイバーの多くは同じアップデート ポリシー（同じアウトバウンド ルート マップ、配信リスト、フィルタ リスト、アップデート送信元など）を使用して設定されます。アップデート ポリシーが同じネイバーをピア グループにまとめると設定が簡単になり、アップデートの効率が高まります。多数のピアを設定した場合は、この方法を推奨します。

BGP ピア グループを設定するには、ピア グループを作成し、そこにオプションを割り当てて、ピア グループ メンバーとしてネイバーを追加します。ピア グループを設定するには、**neighbor** ルータ コンフィギュレーション コマンドを使用します。デフォルトでは、ピア グループ メンバーは remote-as（設定されている場合）、version、update-source、out-route-map、out-filter-list、out-dist-list、minimum-advertisement-interval、next-hop-self など、ピア グループの設定オプションをすべて継承します。すべてのピア グループ メンバーは、ピア グループに対する変更を継承します。また、アウトバウンド アップデートに影響しないオプションを無効にするように、メンバーを設定することもできます。

集約ルート

クラスレス ドメイン間ルーティング（CIDR）を使用すると、集約ルート（またはスーパーネット）を作成して、ルーティング テーブルのサイズを最小化できます。BGP 内に集約ルートを設定するには、集約ルートを BGP に再配信するか、または BGP ルーティング テーブル内に集約エン

トリを作成します。BGP テーブル内に特定のエントリがさらに 1 つまたは複数存在する場合は、BGP テーブルに集約アドレスが追加されます。

ルーティング ドメイン コンフェデレーション

IBGP メッシュを削減する方法の 1 つは、自律システムを複数のサブ自律システムに分割して、単一の自律システムとして認識される単一の連合にグループ化することです。各自律システムは内部で完全にメッシュ化されていて、同じコンフェデレーション内の他の自律システムとの間には数本の接続があります。異なる自律システム内にあるピアでは EBGP セッションが使用されますが、ルーティング情報は IBGP ピアと同様な方法で交換されます。具体的には、ネクストホップ、MED、およびローカル プリファレンス情報は維持されます。すべての自律システムで単一の IGP を使用できます。

BGP ルート リフレクタ

BGP では、すべての IBGP スピーカーを完全メッシュ構造にする必要があります。外部ネイバーからルートを受信したルータは、そのルートをすべての内部ネイバーにアドバタイズする必要があります。ルーティング情報のループを防ぐには、すべての IBGP スピーカーを接続する必要があります。内部ネイバーは、内部ネイバーから学習されたルートを他の内部ネイバーに送信しません。

ルート リフレクタを使用すると、学習されたルートをネイバーに渡す場合に他の方法が使用されるため、すべての IBGP スピーカーを完全メッシュ構造にする必要はありません。IBGP ピアをルート リフレクタに設定すると、その IBGP ピアは IBGP によって学習されたルートを一連の IBGP ネイバーに送信するようになります。ルート リフレクタの内部ピアには、クライアント ピアと非クライアント ピア（AS 内の他のすべてのルータ）の 2 つのグループがあります。ルート リフレクタは、これらの 2 つのグループ間でルートを反映させます。ルート リフレクタおよびクライアント ピアは、クラスタを形成します。非クライアント ピアは相互に完全メッシュ構造にする必要がありますが、クライアント ピアはその必要はありません。クラスタ内のクライアントは、そのクラスタ外の IBGP スピーカーと通信しません。

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、ネイバーに応じて、次のいずれかのアクションを実行します。

- 外部 BGP スピーカーからのルートをすべてのクライアントおよび非クライアント ピアにアドバタイズします。
- 非クライアント ピアからのルートをすべてのクライアントにアドバタイズします。
- クライアントからのルートをすべてのクライアントおよび非クライアント ピアにアドバタイズします。したがって、クライアントを完全メッシュ構造にする必要はありません。

通常、クライアントのクラスタにはルート リフレクタが 1 つあり、クラスタはルート リフレクタのルータ ID で識別されます。冗長性を高めて、シングル ポイントでの障害を回避するには、クラスタに複数のルート リフレクタを設定する必要があります。このように設定した場合は、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、ク

ラスタ内のすべてのルート リフレクタに同じクラスタ ID（4 バイト）を設定する必要があります。クラスタを処理するすべてのルート リフレクタは完全メッシュ構造にし、一連の同一なクライアント ピアおよび非クライアント ピアを設定する必要があります。

ルート ダンプニング

ルート フラップ ダンプニングは、インターネットワーク内でフラッピング ルートの伝播を最小化するためのBGP機能です。ルートの状態が使用可能、使用不可能、使用可能、使用不可能という具合に、繰り返し変化する場合、ルートはフラッピングと見なされます。ルート ダンプニングがイネーブルの場合は、フラッピングしているルートにペナルティ値が割り当てられます。ルートの累積ペナルティが、設定された制限値に到達すると、ルートが稼働している場合であっても、BGP はルートのアドバタイズメントを抑制します。再使用限度は、ペナルティと比較される設定可能な値です。ペナルティが再使用限度より小さくなると、起動中の抑制されたルートのアドバタイズメントが再開されます。

IBGP によって取得されたルートには、ダンプニングが適用されません。このポリシーにより、IBGP ピアのペナルティが AS 外部のルートよりも大きくなることはありません。

BGP の追加情報

BGP 設定の詳しい説明については、『*Cisco IOS IP Configuration Guide, Release 12.4*』の「IP Routing Protocols」にある「BGPの設定」を参照してください。特定コマンドの詳細については、『*Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.4*』を参照してください。

BGP の設定方法

BGP のデフォルト設定

下の表に、BGP のデフォルト設定を示します。すべての特性の詳細については、『*Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.4*』の特定のコマンドを参照してください。

表 *10* : *BGP* のデフォルト設定

機能	デフォルト設定
集約アドレス	ディセーブル：未定義
AS パス アクセス リスト	未定義
自動サマリー	ディセーブル

機能	デフォルト設定
最適パス	• ルータはルートを選択する場合に *as-path* を考慮し、外部 BGP ピアからの類似ルートは比較しません。 • ルータ ID の比較：ディセーブル
BGP コミュニティ リスト	• 番号：未定義。コミュニティ番号を示す特定の値を許可すると、許可されていないその他すべてのコミュニティ番号は、暗黙の拒否にデフォルト設定されます。 • フォーマット：シスコデフォルト フォーマット（32 ビット番号）
BGP 連合 ID/ピア	• ID：未設定 • ピア：識別なし
BGP 高速外部フォールオーバー	イネーブル
BGP ローカル初期設定	100。指定できる範囲は 0～4294967295 です（大きな値を推奨）。
BGP ネットワーク	指定なし。バックドア ルートのアドバタイズなし
BGP ルート ダンプニング	デフォルトでは、ディセーブルです。イネーブルの場合は、次のようになります。 • 半減期は 15 分 • 再使用は 750（10 秒増分） • 抑制は 2000（10 秒増分） • 最大抑制時間は半減期の 4 倍（60 分）
BGP ルータ ID	ループバック インターフェイスに IP アドレスが設定されている場合は、ループバック インターフェイスの IP アドレス、またはルータの物理インターフェイスに対して設定された最大の IP アドレス
デフォルトの情報送信元（プロトコルまたはネットワーク再配信）	ディセーブル
デフォルト メトリック	自動メトリック変換（組み込み）

機能	デフォルト設定
ディスタンス	• 外部ルート アドミニストレーティブ ディスタンス：20（有効値は 1 ～ 255） • 内部ルート アドミニストレーティブ ディスタンス：200（有効値は 1 ～ 255） • ローカル ルート アドミニストレーティブ ディスタンス：200（有効値は 1 ～ 255）
ディストリビュートリスト	• 入力（アップデート中に受信されたネットワークをフィルタリング）：ディセーブル • 出力（アップデート中のネットワークのアドバタイズを抑制）：ディセーブル
内部ルート再配信	ディセーブル
IP プレフィックス リスト	未定義
Multi Exit Discriminator（MED）	• 常に比較：ディセーブル。異なる自律システム内のネイバーからのパスに対して、MED を比較しません。 • 最適パスの比較：ディセーブル • 最悪パスである MED の除外：ディセーブル • 決定的な MED 比較：ディセーブル

機能	デフォルト設定
Neighbor	• アドバタイズメント インターバル：外部ピアの場合は 30 秒、内部ピアの場合は 5 秒 • ロギング変更：イネーブル • 条件付きアドバタイズ：ディセーブル • デフォルト送信元：ネイバーに送信されるデフォルト ルートはなし • 説明：なし • ディストリビュート リスト：未定義 • 外部 BGP マルチホップ：直接接続されたネイバーだけを許可 • フィルタ リスト：使用しない • 受信したプレフィックスの最大数：制限なし • ネクストホップ（BGP ネイバーのネクストホップとなるルータ）：ディセーブル • パスワード：ディセーブル • ピア グループ：定義なし、割り当てメンバーなし • プレフィックス リスト：指定なし • リモート AS（ネイバー BGP テーブルへのエントリ追加）：ピア定義なし • プライベート AS 番号の削除：ディセーブル • ルート マップ：ピアへの適用なし • コミュニティ属性送信：ネイバーへの送信なし。 • シャットダウンまたはソフト再設定：ディセーブル • タイマー：60 秒、ホールドタイム：180 秒 • アップデート送信元：最適ローカル アドレス • バージョン：BGP バージョン 4 • 重み：BGP ピアによって学習されたルート：0、ローカル ルータから取得されたルート：32768
NSF[1] 認識	[2]イネーブル状態の場合、レイヤ 3 スイッチでは、ハードウェアやソフトウェアの変更中に、隣接する NSF 対応ルータからのパケットを転送し続けることができます。

機能	デフォルト設定
ルート リフレクタ	未設定
同期化（BGP および IGP）	ディセーブル
テーブルマップ アップデート	ディセーブル
タイマー	キープアライブ：60 秒、ホールドタイム：180 秒

1 Nonstop Forwarding

2 NSF 認識は、グレースフル リスタートをイネーブルにすることにより、IP サービス フィーチャ セット ライセンスを実行するスイッチ上で IPv4 に対してイネーブルにできます。

BGP ルーティングのイネーブル化

はじめる前に

（注） BGP をイネーブルにするには、スイッチまたはスタック マスター上で IP サービス フィーチャ セットが稼働している必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ip routing** 例： `Device(config)# ip routing`	IP ルーティングをイネーブルにします。
ステップ 3	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 45000`	BGP ルーティング プロセスをイネーブルにして AS 番号を割り当て、ルータ コンフィギュレーション モードを開始します。指定できる AS 番号は 1～65535 です。64512～65535 は、プライベート AS 番号専用です。

	コマンドまたはアクション	目的
ステップ 4	**network** *network-number* [**mask** *network-mask*] [**route-map** *route-map-name*] 例： `Device(config)# network 10.108.0.0`	この AS に対してローカルとなるようにネットワークを設定し、BGPテーブルにネットワークを格納します。
ステップ 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *number* 例： `Device(config)# neighbor 10.108.1.2 remote-as 65200`	BGP ネイバー テーブルに設定を追加し、IP アドレスによって識別されるネイバーが、指定された AS に属することを示します。 EBGP の場合、通常ネイバーは直接接続されており、IPアドレスは接続のもう一方の端におけるインターフェイスのアドレスです。 IBGP の場合、IP アドレスにはルータ インターフェイス内の任意のアドレスを指定できます。
ステップ 6	**neighbor** {*ip-address* | *peer-group-name*} **remove-private-as** 例： `Device(config)# neighbor 172.16.2.33 remove-private-as`	（任意）発信ルーティング アップデート内の AS パスからプライベート AS 番号を削除します。
ステップ 7	**synchronization** 例： `Device(config)# synchronization`	（任意）BGP と IGP の同期化をイネーブルにします。
ステップ 8	**auto-summary** 例： `Device(config)# auto-summary`	（任意）自動ネットワーク サマライズをイネーブルにします。IGP から BGP にサブネットが再配信された場合、ネットワーク ルートだけがBGP テーブルに挿入されます。
ステップ 9	**bgp graceful-restart** 例： `Device(config)# bgp graceful-start`	（任意）NSF認識をスイッチでイネーブルにします。NSF 認識はデフォルトではディセーブルです。

	コマンドまたはアクション	目的
ステップ 10	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 11	**show ip bgp network** *network-number* 例： `Device# show ip bgp network 10.108.0.0`	設定を確認します。
ステップ 12	**show ip bgp neighbor** 例： `Device# show ip bgp neighbor`	NSF認識（グレースフル リスタート）がネイバーでイネーブルにされていることを確認します。 スイッチおよびネイバーで NSF 認識がイネーブルである場合は、次のメッセージが表示されます。 グレースフル リスタート機能*:* アドバタイズおよび受信される スイッチで NSF 認識がイネーブルであり、ネイバーでディセーブルである場合は、次のメッセージが表示されます。 グレースフル リスタート機能*:* アドバタイズされる
ステップ 13	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルーティング ポリシー変更の管理

BGP ピアがルート リフレッシュ機能をサポートするかどうかを学習して、BGP セッションをリセットするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**show ip bgp neighbors** 例： `Device# show ip bgp neighbors`	ネイバーがルート リフレッシュ機能をサポートするかどうかを表示します。サポートされている場合は、ルータに関する次のメッセージが表示されます。 *Received route refresh capability from peer*
ステップ 2	**clear ip bgp** {***** | *address* | *peer-group-name*} 例： `Device# clear ip bgp *`	指定された接続上でルーティング テーブルをリセットします。 • すべての接続をリセットする場合は、アスタリスク（*）を入力します。 • 特定の接続をリセットする場合は、IP アドレスを入力します。 • ピア グループをリセットする場合は、ピア グループ名を入力します。
ステップ 3	**clear ip bgp** {***** | *address* | *peer-group-name*} **soft out** 例： `Device# clear ip bgp * soft out`	（任意）指定された接続上でインバウンド ルーティング テーブルをリセットするには、アウトバウンド ソフト リセットを実行します。このコマンドは、ルート リフレッシュがサポートされている場合に使用してください。 • すべての接続をリセットする場合は、アスタリスク（*）を入力します。 • 特定の接続をリセットする場合は、IP アドレスを入力します。 • ピア グループをリセットする場合は、ピア グループ名を入力します。
ステップ 4	**show ip bgp** 例： `Device# show ip bgp`	ルーティング テーブルおよび BGP ネイバーに関する情報をチェックし、リセットされたことを確認します。
ステップ 5	**show ip bgp neighbors** 例： `Device# show ip bgp neighbors`	ルーティング テーブルおよび BGP ネイバーに関する情報をチェックし、リセットされたことを確認します。

BGP 判断属性の設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 4500`	BGP ルーティング プロセスをイネーブルにして AS 番号を割り当て、ルータ コンフィギュレーション モードを開始します。
ステップ 3	**bgp best-path as-path ignore** 例： `Device(config-router)# bgp bestpath as-path ignore`	（任意）ルート選択中に AS パス長を無視するようにルータを設定します。
ステップ 4	**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self** 例： `Device(config-router)# neighbor 10.108.1.1 next-hop-self`	（任意）ネクストホップ アドレスの代わりに使用される特定の IP アドレスを入力し、ネイバーへの BGP アップデートに関するネクストホップの処理をディセーブルにします。
ステップ 5	**neighbor** {*ip-address* | *peer-group-name*} **weight** *weight* 例： `Device(config-router)# neighbor 172.16.12.1 weight 50`	（任意）ネイバー接続に重みを割り当てます。指定できる値は 0 ～ 65535 です。最大の重みのルートを推奨します。別の BGP ピアから学習されたルートのデフォルトの重みは 0 です。ローカル ルータから送信されたルートのデフォルトの重みは 32768 です。
ステップ 6	**default-metric** *number* 例： `Device(config-router)# default-metric 300`	（任意）推奨パスを外部ネイバーに設定するように MED メトリックを設定します。MED を持たないすべてのルータも、この値に設定されます。指定できる範囲は 1 ～ 4294967295 です。最小値を推奨します。

	コマンドまたはアクション	目的
ステップ 7	**bgp bestpath med missing-as-worst** 例： `Device(config-router)# bgp bestpath med missing-as-worst`	（任意）MED がない場合は無限の値が指定されていると見なし、MED 値を持たないパスが最も望ましくないパスになるように、スイッチを設定します。
ステップ 8	**bgp always-compare med** 例： `Device(config-router)# bgp always-compare-med`	（任意）異なる AS 内のネイバーからのパスに対して、MED を比較するようにスイッチを設定します。デフォルトでは、MED は同じ AS 内のパス間でだけ比較されます。
ステップ 9	**bgp bestpath med confed** 例： `Device(config-router)# bgp bestpath med confed`	（任意）連合内の異なるサブ AS によってアドバタイズされたパスから特定のパスを選択する場合に、MED を考慮するようにスイッチを設定します。
ステップ 10	**bgp deterministic med** 例： `Device(config-router)# bgp deterministic med`	（任意）同じ AS 内の異なるピアによってアドバタイズされたルートから選択する場合に、MED 変数を考慮するようにスイッチを設定します。
ステップ 11	**bgp default local-preference** *value* 例： `Device(config-router)# bgp default local-preference 200`	（任意）デフォルトのローカル プリファレンス値を変更します。指定できる範囲は 0 ～ 4294967295 で、デフォルト値は 100 です。最大のローカル プリファレンス値を推奨します。
ステップ 12	**maximum-paths** *number* 例： `Device(config-router)# maximum-paths 8`	（任意）IP ルーティング テーブルに追加するパスの数を設定します。デフォルトでは、最適パスだけがルーティング テーブルに追加されます。指定できる範囲は 1 ～ 16 です。複数の値を指定すると、パス間のロード バランシングが可能になります。スイッチ ソフトウェア では最大 32 の等コスト ルートが許可されていますが、スイッチ ハードウェアはルートあたり 17 パス以上は使用しません。
ステップ 13	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 14	**show ip bgp** 例： `Device# show ip bgp`	ルーティング テーブルおよびBGP ネイバーに関する情報をチェックし、リセットされたことを確認します。
ステップ 15	**show ip bgp neighbors** 例： `Device# show ip bgp neighbors`	ルーティング テーブルおよびBGP ネイバーに関する情報をチェックし、リセットされたことを確認します。
ステップ 16	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルート マップによる BGP フィルタリングの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] 例： `Device(config)# route-map set-peer-address permit 10`	ルートマップを作成し、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。
ステップ 3	**set ip next-hop** *ip-address* [*...ip-address*] [**peer-address**] 例： `Device(config)# set ip next-hop 10.1.1.3`	（任意）ネクストホップ処理をディセーブルにするようにルート マップを設定します。 • インバウンドルートマップの場合は、一致するルートのネクストホップをネイバー ピ

	コマンドまたはアクション	目的
		アアドレスに設定し、サードパーティのネクスト ホップを上書きします。 • BGP ピアのアウトバウンド ルート マップの場合は、ネクストホップをローカル ルータのピア アドレスに設定して、ネクスト ホップ計算をディセーブルにします。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show route-map** [*map-name*] 例： `Device# show route-map`	設定を確認するため、設定されたすべてのルート マップ、または指定されたルート マップだけを表示します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ネイバーによる BGP フィルタリングの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 109`	BGP ルーティング プロセスをイネーブルにして AS 番号を割り当て、ルータ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**neighbor** {*ip-address* | *peer-group name*} **distribute-list**{**access-list-number** | *name*} {**in** | **out**} 例： `Device(config-router)# neighbor 172.16.4.1 distribute-list 39 in`	（任意）アクセス リストの指定に従って、ネイバーに対して送受信されるBGPルーティングアップデートをフィルタリングします。 （注） **neighbor prefix-list** ルータ コンフィギュレーション コマンドを使用して、アップデートをフィルタリングすることもできますが、両方のコマンドを使用して同じBGP ピアを設定することはできません。
ステップ 4	**neighbor** {*ip-address* | *peer-group name*} **route-map** *map-tag* {**in** | **out**} 例： `Device(config-router)# neighbor 172.16.70.24 route-map internal-map in`	（任意）ルート マップを適用し、着信または発信ルートをフィルタリングします。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip bgp neighbors** 例： `Device# show ip bgp neighbors`	設定を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

アクセス リストおよびネイバーによる BGP フィルタリングの設定

BGP 自律システム パスに基づいて着信および発信の両方のアップデートにアクセス リスト フィルタを指定して、フィルタリングすることもできます。各フィルタは、正規表現を使用するアクセス リストです。（正規表現の作成方法については、『*Cisco IOS Dial Technologies Command Reference, Release 12.4*』の付録「Regular Expressions」を参照してください）。この方法を使用す

るには、自律システムパスのアクセス リストを定義し、特定のネイバーとの間のアップデートに適用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ip as-path access-list** *access-list-number* {**permit** | **deny**} *as-regular-expressions* 例： `Device(config)# ip as-path access-list 1 deny _65535_`	BGP-related アクセス リストを定義します。
ステップ 3	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 110`	BGP ルータ コンフィギュレーション モードを開始します。
ステップ 4	**neighbor** {*ip-address* | *peer-group name*} **filter-list** {*access-list-number* | *name*} {**in** | **out** | **weight** *weight*} 例： `Device(config-router)# neighbor 172.16.1.1 filter-list 1 out`	アクセス リストに基づいて、BGP フィルタを確立します。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip bgp neighbors** [**paths** *regular-expression*] 例： `Device# show ip bgp neighbors`	設定を確認します。
ステップ 7	**copy running-config startup-config** 例： `Device# copy running-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

	コマンドまたはアクション	目的
	startup-config	

BGP フィルタリング用のプレフィックス リストの設定

コンフィギュレーション エントリを削除する場合は、シーケンス番号を指定する必要はありません。**Show** コマンドの出力には、シーケンス番号が含まれます。

コマンド内でプレフィックスリストを使用する場合は、あらかじめプレフィックスリストを設定しておく必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ip prefix-list** *list-name* [**seq** *seq-value*] **deny** | **permit** *network/len* [**ge** *ge-value*] [**le** *le-value*] 例： `Device(config)# ip prefix-list BLUE permit 172.16.1.0/24`	一致条件に合わせてアクセスを拒否（**deny**）または許可（**permit**）するプレフィックスリストを作成します。シーケンス番号を指定することもできます。少なくとも 1 つの **permit** コマンドまたは **deny** コマンドを入力する必要があります。 • *network/len* は、ネットワーク番号およびネットワークマスクの長さ（ビット単位）です。 •（任意）**ge** および **le** の値は、照合するプレフィックス長の範囲を指定します。指定された *ge-value* および *le-value* は、次の条件を満たす必要があります。*len* < *ge-value* < *le-value*< 32

	コマンドまたはアクション	目的
ステップ 3	**ip prefix-list** *list-name***seq** *seq-value***deny** | **permit** *network/len* [**ge** *ge-value*] [**le** *le-value*] 例： `Device(config)# ip prefix-list BLUE seq 10 permit 172.24.1.0/24`	（任意）プレフィックス リストにエントリを追加し、そのエントリにシーケンス番号を割り当てます。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show ip prefix list** [**detail** | **summary**] *name* [*network/len*] [**seq** *seq-num*] [**longer**] [**first-match**] 例： `Device# show ip prefix list summary test`	プレフィックス リストまたはプレフィックス リストエントリに関する情報を表示して、設定を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

BGP コミュニティ フィルタリングの設定

デフォルトでは、COMMUNITIES 属性はネイバーに送信されません。COMMUNITIES 属性が特定の IP アドレスのネイバーに送信されるように指定するには、**neighbor send-community** ルータ コンフィギュレーション コマンドを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**ip community-list** *community-list-number* {**permit** | **deny**} *community-number* 例： `Device(config)# ip community-list 1 permit 50000:10`	コミュニティ リストを作成し、番号を割り当てます。 • *community-list-number* は 1 ～ 99 の整数です。この値は、コミュニティの 1 つ以上の許可または拒否グループを識別します。 • *community-number* は、**set community** ルートマップ コンフィギュレーション コマンドで設定される番号です。
ステップ 3	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 108`	BGP ルータ コンフィギュレーション モードを開始します。
ステップ 4	**neighbor** {*ip-address* | *peer-group name*} **send-community** 例： `Device(config-router)# neighbor 172.16.70.23 send-community`	この IP アドレスのネイバーに送信する COMMUNITIES 属性を指定します。
ステップ 5	**set comm-list** *list-num***delete** 例： `Device(config-router)# set comm-list 500 delete`	（任意）ルートマップで指定された標準または拡張コミュニティ リストと一致する着信または発信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。
ステップ 6	**exit** 例： `Device(config-router)# end`	グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**ip bgp-community new-format** 例： `Device(config)# ip bgp-community new format`	（任意）AA:NN の形式で、BGP コミュニティを表示、解析します。 BGP コミュニティは、2 つの部分からなる 2 バイト長形式で表示されます。シスコのデフォルトのコミュニティ形式は、NNAA です。BGP に関する最新の RFC では、コミュニティは AA:NN の形式をとります。最初の部分は AS 番号で、その次の部分は 2 バイトの数値です。

	コマンドまたはアクション	目的
ステップ 8	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 9	**show ip bgp community** 例： `Device# show ip bgp community`	設定を確認します。
ステップ 10	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

BGP ネイバーおよびピア グループの設定

各ネイバーに設定オプションを割り当てるには、ネイバーの IP アドレスを使用し、次に示すルータ コンフィギュレーション コマンドのいずれかを指定します。ピア グループにオプションを割り当てるには、ピア グループ名を使用し、いずれかのコマンドを指定します。**neighbor shutdown** ルータ コンフィギュレーション コマンドを使用すると、すべての設定情報を削除せずに、BGP ピアまたはピア グループをディセーブルにできます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router bgp** *autonomous-system*	BGP ルータ コンフィギュレーション モードを開始します。
ステップ 3	**neighbor** *peer-group-name***peer-group**	BGP ピア グループを作成します。

	コマンドまたはアクション	目的
ステップ 4	**neighbor** *ip-address***peer-group** *peer-group-name*	BGP ネイバーをピア グループのメンバにします。
ステップ 5	**neighbor** {*ip-address* | *peer-group-name*} **remote-as** *number*	BGP ネイバーを指定します。**remote-as** *number* を使用してピア グループが設定されていない場合は、このコマンドを使用し、EBGP ネイバーを含むピア グループを作成します。指定できる範囲は 1 ～ 65535 です。
ステップ 6	**neighbor** {*ip-address* | *peer-group-name*} **description** *text*	（任意）ネイバーに説明を関連付けます。
ステップ 7	**neighbor** {*ip-address* | *peer-group-name*} **default-originate** [**route-map** *map-name*]	（任意）BGP スピーカー（ローカル ルータ）にネイバーへのデフォルト ルート 0.0.0.0 の送信を許可して、このルートがデフォルト ルートとして使用されるようにします。
ステップ 8	**neighbor** {*ip-address* | *peer-group-name*} **send-community**	（任意）この IP アドレスのネイバーに送信する COMMUNITIES 属性を指定します。
ステップ 9	**neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface*	（任意）内部 BGP セッションに、TCP 接続に関するすべての操作インターフェイスの使用を許可します。
ステップ 10	**neighbor** {*ip-address* | *peer-group-name*} **ebgp-multihop**	（任意）ネイバーがセグメントに直接接続されていない場合でも、BGP セッションを使用可能にします。マルチホップ ピア アドレスへの唯一のルートがデフォルト ルート（0.0.0.0）の場合、マルチホップ セッションは確立されません。
ステップ 11	**neighbor** {*ip-address* | *peer-group-name*} **local-as** *number*	（任意）ローカル AS として使用する AS 番号を指定します。指定できる範囲は 1 ～ 65535 です。
ステップ 12	**neighbor** {*ip-address* | *peer-group-name*} **advertisement-interval** *seconds*	（任意）BGP ルーティング アップデートを送信する最小インターバルを設定します。
ステップ 13	**neighbor** {*ip-address* | *peer-group-name*} **maximum-prefix** *maximum* [*threshold*]	（任意）ネイバーから受信できるプレフィックス数を制御します。指定できる範囲は 1 ～ 4294967295 です。*threshold*（任意）は、警告メッセージが生成される基準となる最大値（パーセンテージ）です。デフォルトは 75% です。

	コマンドまたはアクション	目的
ステップ 14	**neighbor** {*ip-address* | *peer-group-name*} **next-hop-self**	（任意）ネイバー宛てのBGPアップデートに関して、ネクストホップでの処理をディセーブルにします。
ステップ 15	neighbor {*ip-address* | *peer-group-name*} password *string*	（任意）TCP 接続での MD5 認証を BGP ピアに設定します。両方のBGPピアに同じパスワードを設定する必要があります。そうしないと、BGP ピア間に接続が作成されません。
ステップ 16	**neighbor** {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}	（任意）着信または発信ルートにルートマップを適用します。
ステップ 17	**neighbor** {*ip-address* | *peer-group-name*} **send-community**	（任意）このIPアドレスのネイバーに送信するCOMMUNITIES 属性を指定します。
ステップ 18	**neighbor** {*ip-address* | *peer-group-name*} **timers** *keepalive holdtime*	（任意）ネイバーまたはピア グループ用のタイマーを設定します。 • *keepalive* インターバルは、キープアライブ メッセージがピアに送信される間隔です。指定できる範囲は 1 ～ 4294967295 秒です。デフォルト値は 60 秒です。 • *holdtime* は、キープアライブ メッセージを受信しなかった場合、ピアが非アクティブと宣言されるまでのインターバルです。指定できる範囲は 1 ～ 4294967295 秒です。デフォルト値は 180 秒です。
ステップ 19	**neighbor** {*ip-address* | *peer-group-name*} **weight** *weight*	（任意）ネイバーからのすべてのルートに関する重みを指定します。
ステップ 20	**neighbor** {*ip-address* | *peer-group-name*} **distribute-list** {*access-list-number* | *name*} {**in** | **out**}	（任意）アクセス リストの指定に従って、ネイバーに対して送受信される BGP ルーティング アップデートをフィルタリングします。
ステップ 21	**neighbor** {*ip-address* | *peer-group-name*} **filter-list** *access-list-number* {**in** | **out** | **weight** *weight*}	（任意）BGP フィルタを確立します。
ステップ 22	**neighbor** {*ip-address* | *peer-group-name*} **version** *value*	（任意）ネイバーと通信するときに使用する BGP バージョンを指定します。

	コマンドまたはアクション	目的
ステップ 23	**neighbor** {*ip-address* | *peer-group-name*} **soft-reconfiguration inbound**	（任意）受信したアップデートのストアを開始するようにソフトウェアを設定します。
ステップ 24	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 25	**show ip bgp neighbors**	設定を確認します。
ステップ 26	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルーティング テーブルでの集約アドレスの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 106`	BGP ルータ コンフィギュレーション モードを開始します。
ステップ 3	**aggregate-address** *address mask* 例： `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0`	BGP ルーティング テーブル内に集約エントリを作成します。集約ルートは AS からのルートとしてアドバタイズされます。情報が失われた可能性があることを示すため、アトミック集約属性が設定されます。

	コマンドまたはアクション	目的
ステップ 4	**aggregate-address** *address mask***as-set** 例： `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 as-set`	（任意）AS 設定パス情報を生成します。このコマンドは、この前のコマンドと同じルールに従う集約エントリを作成します。ただし、アドバタイズされるパスは、すべてのパスに含まれる全要素で構成される AS_SET です。多くのパスを集約するときは、このキーワードを使用しないでください。このルートは絶えず取り消され、アップデートされます。
ステップ 5	**aggregate-address** *address-mask***summary-only** 例： `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 summary-only`	（任意）サマリーアドレスだけをアドバタイズします。
ステップ 6	**aggregate-address** *address mask***suppress-map** *map-name* 例： `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 suppress-map map1`	（任意）選択された、より具体的なルートを抑制します。
ステップ 7	**aggregate-address** *address mask***advertise-map** *map-name* 例： `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 advertise-map map2`	（任意）ルートマップによって指定された設定に基づいて集約を生成します。
ステップ 8	**aggregate-address** *address mask***attribute-map** *map-name* 例： `Device(config-router)# aggregate-address 10.0.0.0 255.0.0.0 attribute-map map3`	（任意）ルートマップで指定された属性を持つ集約を生成します。
ステップ 9	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 10	**show ip bgp neighbors** [**advertised-routes**] 例： `Device# show ip bgp neighbors`	設定を確認します。
ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルーティング ドメイン連合の設定

自律システムのグループの自律システム番号として機能する連合 ID を指定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 100`	BGP ルータ コンフィギュレーション モードを開始します。
ステップ 3	**bgp confederation identifier** *autonomous-system* 例： `Device(config)# bgp confederation identifier 50007`	BGP 連合 ID を設定します。

	コマンドまたはアクション	目的
ステップ 4	**bgp confederation peers** *autonomous-system* [*autonomous-system ...*] 例： `Device(config)# bgp confederation peers 51000 51001 51002`	連合に属する AS、および特殊な EBGP ピアとして処理する AS を指定します。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip bgp neighbor** 例： `Device# show ip bgp neighbor`	設定を確認します。
ステップ 7	**show ip bgp network** 例： `Device# show ip bgp network`	設定を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

BGP ルート リフレクタの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 101`	BGP ルータ コンフィギュレーション モードを開始します。
ステップ 3	**neighbor** {*ip-address* | *peer-group-name*} **route-reflector-client** 例： `Device(config-router)# neighbor 172.16.70.24 route-reflector-client`	ローカル ルータを BGP ルート リフレクタとして、指定されたネイバーをクライアントとして、それぞれ設定します。
ステップ 4	**bgp cluster-id** *cluster-id* 例： `Device(config-router)# bgp cluster-id 10.0.1.2`	（任意）クラスタに複数のルート リフレクタが存在する場合、クラスタ ID を設定します。
ステップ 5	**no bgp client-to-client reflection** 例： `Device(config-router)# no bgp client-to-client reflection`	（任意）クライアント間のルート反映をディセーブルにします。デフォルトでは、ルート リフレクタ クライアントからのルートは、他のクライアントに反映されます。ただし、クライアントが完全メッシュ構造の場合、ルート リフレクタはルートをクライアントに反映させる必要がありません。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip bgp** 例： `Device# show ip bgp`	設定を確認します。送信元 ID およびクラスタ リスト属性を表示します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルート ダンプニングの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router bgp** *autonomous-system* 例： `Device(config)# router bgp 100`	BGP ルータ コンフィギュレーション モードを開始します。
ステップ 3	**bgp dampening** 例： `Device(config-router)# bgp dampening`	BGP ルート ダンプニングをイネーブルにします。
ステップ 4	**bgp dampening** *half-life reuse suppress max-suppress* [**route-map** *map*] 例： `Device(config-router)# bgp dampening 30 1500 10000 120`	（任意）ルート ダンプニング係数のデフォルト値を変更します。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show ip bgp flap-statistics** [{**regexp** *regexp*} | {**filter-list** *list*} | {*address mask* [**longer-prefix**]}] 例： `Device# show ip bgp flap-statistics`	（任意）フラッピングしているすべてのパスのフラップを監視します。ルートの抑制が終了し、安定状態になると、統計情報が削除されます。
ステップ 7	**show ip bgp dampened-paths** 例： `Device# show pi bgp dampened-paths`	（任意）抑制されるまでの時間を含めて、ダンプニングされたルートを表示します。

	コマンドまたはアクション	目的
ステップ 8	**clear ip bgp flap-statistics** [{ *regexp*} | { *list*} | {*address mask* []}**regexpfilter-listlonger-prefix** 例： `Device# clear ip bgp flap-statistics`	（任意）BGP フラップ統計情報を消去して、ルートがダンプニングされる可能性を小さくします。
ステップ 9	**clear ip bgp dampening** 例： `Device# clear ip bgp dampening`	（任意）ルート ダンプニング情報を消去して、ルートの抑制を解除します。
ステップ 10	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

BGP のモニタリングおよびメンテナンス

特定のキャッシュ、テーブル、またはデータベースのすべての内容を削除できます。この作業は、特定の構造の内容が無効になる場合、または無効である疑いがある場合に必要となります。

BGP ルーティング テーブル、キャッシュ、データベースの内容など、特定の統計情報を表示できます。さらに、リソースの利用率を取得したり、ネットワーク問題を解決するための情報を使用することもできます。さらに、ノードの到達可能性に関する情報を表示し、デバイスのパケットが経由するネットワーク内のルーティング パスを検出することもできます。

下の図に、BGP を消去および表示するために使用する特権 EXEC コマンドを示します。表示されるフィールドの詳細については、『*Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.4*』を参照してください。

表 *11*：*IP BGP* の *clear* および *show* コマンド

clear ip bgp *address*	特定の BGP 接続をリセットします。
clear ip bgp *	すべての BGP 接続をリセットします。
clear ip bgp peer-group *tag*	BGP ピア グループのすべてのメンバを削除します。

show ip bgp *prefix*	プレフィックスがアドバタイズされるピア グループ、またはピア グループに含まれないピアを表示します。ネクスト ホップやローカル プレフィックスなどのプレフィックス属性も表示されます。
show ip bgp cidr-only	サブネットおよびスーパーネット ネットワーク マスクを含むすべての BGP ルートを表示します。
show ip bgp community [*community-number*] [**exact**]	指定されたコミュニティに属するルートを表示します。
show ip bgp community-list *community-list-number* [**exact-match**]	コミュニティ リストで許可されたルートを表示します。
show ip bgp filter-list *access-list-number*	指定された AS パス アクセス リストによって照合されたルートを表示します。
show ip bgp inconsistent-as	送信元の AS と矛盾するルートを表示します。
show ip bgp regexp *regular-expression*	コマンドラインに入力された特定の正規表現と一致する AS パスを持つルートを表示します。
show ip bgp	BGP ルーティング テーブルの内容を表示します。
show ip bgp neighbors [*address*]	各ネイバーとの BGP 接続および TCP 接続に関する詳細情報を表示します。
show ip bgp neighbors [*address*] [**advertised-routes** | **dampened-routes** | **flap-statistics** | **paths** *regular-expression* | **received-routes** | **routes**]	特定の BGP ネイバーから取得されたルートを表示します。
show ip bgp paths	データベース内のすべての BGP パスを表示します。
show ip bgp peer-group [*tag*] [**summary**]	BGP ピア グループに関する情報を表示します。
show ip bgp summary	BGP 接続すべての状況を表示します。

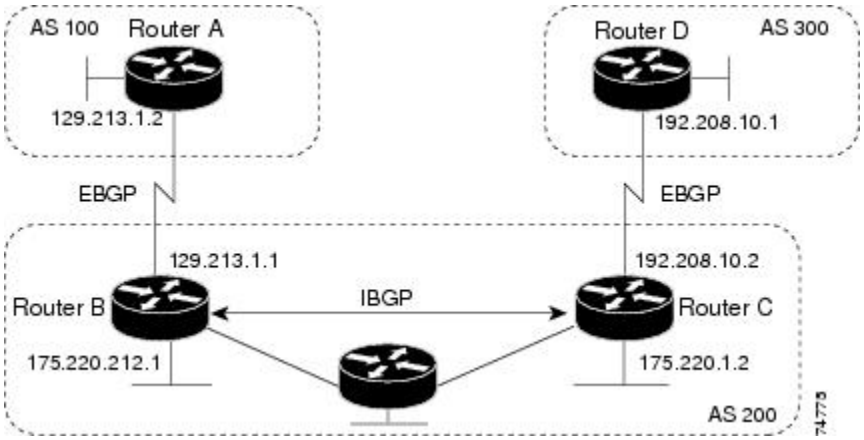
bgp log-neighbor changes コマンドは、デフォルトでイネーブルです。そのため、BGP ネイバーのリセット、起動、またはダウン時に生成されるメッセージをログに記録できます。

BGP の設定例

例：ルータでの BGP の設定

次に、下の図のルータでの BGP の設定例を示します。

図 *6*：*EBGP*、*IBGP*、および複数の自律システム



ルータ A：

```
Device(config)# router bgp 100
Device(config-router)# neighbor 129.213.1.1 remote-as 200
```

ルータ B：

```
Device(config)# router bgp 200
Device(config-router)# neighbor 129.213.1.2 remote-as 100
Device(config-router)# neighbor 175.220.1.2 remote-as 200
```

ルータ C：

```
Device(config)# router bgp 200
Device(config-router)# neighbor 175.220.212.1 remote-as 200
Device(config-router)# neighbor 192.208.10.1 remote-as 300
```

ルータ D：

```
Device(config)# router bgp 300
Device(config-router)# neighbor 192.208.10.2 remote-as 200
```

BGP ピアが稼働していることを確認するには、show ip bgp neighbors 特権 EXEC コマンドを使用します。次に、ルータ A にこのコマンドを実行した場合の出力例を示します。

```
Device# show ip bgp neighbors

BGP neighbor is 129.213.1.1, remote AS 200, external link
 BGP version 4, remote router ID 175.220.212.1
 BGP state = established, table version = 3, up for 0:10:59
 Last read 0:00:29, hold time is 180, keepalive interval is 60 seconds
 Minimum time between advertisement runs is 30 seconds
 Received 2828 messages, 0 notifications, 0 in queue
```

```
 Sent 2826 messages, 0 notifications, 0 in queue
 Connections established 11; dropped 10
```

state = established 以外の情報が出力された場合、ピアは稼働していません。リモート ルータ ID は、ルータ（または最大のループバック インターフェイス）上の最大の IP アドレスです。テーブルが新規情報でアップデートされるたびに、テーブルのバージョン番号は増加します。継続的にテーブル バージョン番号が増加している場合は、ルートがフラッピングし、ルーティング アップデートが絶えず発生しています。

外部プロトコルの場合、**network** ルータ コンフィギュレーション コマンドから IP ネットワークへの参照によって制御されるのは、アドバタイズされるネットワークだけです。これは、**network** コマンドを使用してアップデートの送信先を指定する IGP（EIGRP など）と対照的です。

BGP 設定の詳細については、『*Cisco IOS IP Configuration Guide, Release 12.4*』の「IP Routing Protocols」を参照してください。特定コマンドの詳細については、『*Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.4*』を参照してください。

ISO CLNS ルーティングに関する情報

コネクションレス型ルーティング

国際標準化機構（ISO）コネクションレス型ネットワーク サービス（CLNS）プロトコルとは、オープン システム インターコネクション（OSI）モデルのネットワーク層の標準の 1 つです。ISO ネットワーク アーキテクチャ内のアドレスは、ネットワーク サービス アクセス ポイント（NSAP）アドレスおよび Network Entity Titles（NETs）と呼ばれます。OSI ネットワークの各ノードには、1 つ以上の NETs が含まれます。さらに、各ノードには、多数の NSAP アドレスが含まれます。

スイッチ上で、**clns routing** グローバル コンフィギュレーション コマンドを使用してコネクションレス型ルーティングをイネーブルにすると、スイッチはルーティング関連の機能を果たさず、転送の決定だけを行います。ダイナミック ルーティングには、ルーティング プロトコルもイネーブルにする必要があります。スイッチは、Intermediate System-to-Intermediate System（IS-IS）ダイナミック ルーティング プロトコルをサポートします。このプロトコルは、ISO CLNS ネットワーク用の OSI ルーティング プロトコルに基づいています。

動的にルーティングを行う場合は、IS-IS を使用します。このルーティング プロトコルは、エリアの概念をサポートします。1 つのエリア内部では、すべてのルータがすべてのシステム ID に到達する方法を認識しています。エリア間では、ルータは適切なエリアに到達する方法を認識しています。IS-IS は、ステーション ルーティング（1 つのエリア内）およびエリア ルーティング（エリア間）という 2 つのレベルのルーティングをサポートします。

ISO IGRP と IS-IS NSAP アドレス方式の主な違いは、エリア アドレスの定義にあります。両方ともレベル 1 ルーティング（1 つのエリア内）にはシステム ID を使用します。ただし、エリア ルーティングに関してアドレスが指定される方法が異なります。ISO IGRP NSAP アドレスには、ドメイン、エリア、およびシステム ID という 3 つの異なるフィールドが含まれます。IS-IS アドレスには、単一の連続的エリア フィールド（ドメイン フィールドおよびエリア フィールドから成る）とシステム ID という 2 つのフィールドが含まれます。

（注） ISO CLNS の詳細については、『*Cisco IOS Apollo Domain, Banyan VINES, DECnet, ISO CLNS and XNS Configuration Guide, Release 12.4*』を参照してください。この章で使用するコマンドの構文および使用方法の詳細については、『*Cisco IOS Apollo Domain, Banyan VINES, DECnet, ISO CLNS and XNS Command Reference, Release 12.4*』を参照するか、IOS コマンド リファレンス マスター インデックスを使用するか、オンライン検索を行ってください。

IS-IS Dynamic Routing

IS-IS は、ISO ダイナミック ルーティング プロトコルの 1 つです（ISO 105890 で説明されている）。その他のルーティング プロトコルとは異なり、IS-IS をイネーブルするには、IS-IS ルーティング プロセスを作成し、それをネットワークではなく特定のインターフェイスに割り当てる必要があります。マルチエリア IS-IS コンフィギュレーション構文を使用することで、レイヤ 3 スイッチまたはルータごとに複数の IS-IS ルーティング プロトコルを指定できます。その後、IS-IS ルーティング プロセスのインスタンスごとにパラメータを設定します。

小規模の IS-IS ネットワークは、ネットワーク内にすべてのルータが含まれる単一のエリアとして構築されます。ネットワークの規模が大きくなるに従って、このネットワークは、すべてのエリアに属する、接続されたすべてのレベル 2 ルータのセットから構成されるバックボーン エリア内に再編成され、その後、このネットワークはローカルエリアに接続されます。ローカルエリア内部では、すべてのルータがすべてのシステム ID に到達する方法を認識しています。エリア間では、ルータはバックボーンへの到達方法を認識しており、バックボーン ルータは他のエリアに到達する方法を認識しています。

ルータは、ローカル エリア内でルーティングを実行するために、レベル 1 の隣接関係を確立します（ステーション ルーティング）。ルータは、レベル 1 のエリア間でルーティングを実行するために、レベル 2 の隣接関係を確立します（エリア ルーティング）。

1 つの Cisco ルータは、最大 29 エリアのルーティングに参加でき、バックボーンでレベル 2 ルーティングを実行できます。一般に、ルーティング プロセスごとに 1 つのエリアに対応します。デフォルトでは、ルーティング プロセスの最初のインスタンスが、レベル 1 および レベル 2 両方のルーティングを実行するように設定されます。追加のルーティング インスタンスを設定できます。このインスタンスは、自動的にレベル 1 エリアとして扱われます。IS-IS ルーティング プロセスの各インスタンスごとに個別にパラメータを設定する必要があります。

IS-IS マルチエリア ルーティングでは、シスコの各装置に対して最大 29 個の レベル 1 エリアを定義できますが、レベル 2 ルーティングを実行するプロセスは 1 つだけ設定できます。レベル 2 ルーティングが任意のプロセス上に設定されている場合、追加のプロセスは、すべて自動的にレベル 1 に設定されます。同時に、このプロセスがレベル 1 ルーティングを実行するように設定することもできます。ルータ インスタンスにレベル 2 ルーティングが必要でない場合は、**is-type** グローバル コンフィギュレーション コマンドを使用してレベル 2 の機能を削除します。別のルータ インスタンスをレベル 2 ルータとして設定する場合にも **is-type** コマンドを使用します。

（注） IS-IS の詳細については、『*Cisco IOS IP Configuration Guide, Release 12.4*』の「IP Routing Protocols」を参照してください。ここで使用するコマンドの構文および使用方法の詳細については、『*Cisco IOS IP Command Reference, Release 12.4*』を参照してください。

NSF 認識

統合型 IS-IS NSF 認識機能は IPv4G でサポートされています。この機能により、NSF を認識する顧客宅内装置（CPE）ルータが、NFS 対応ルータによるパケットのノンストップ転送を実現します。ローカル ルータでは、必ずしも NSF を実行している必要はありませんが、このルータが NSF を認識していると、スイッチオーバープロセス時にルーティング データベースの整合性と精度、および隣接 NSF 対応ルータ上のリンクステート データベースが保持されます。

この機能は、自動的にイネーブルにされ、設定は必要ありません。この機能の詳細については、『*Integrated IS-IS Nonstop Forwarding (NSF) Awareness Feature Guide*』を参照してください。

IS-IS グローバル パラメータ

設定可能ないくつかのオプションの IS-IS グローバル パラメータを次に示します。

- ルート マップによって制御されるデフォルト ルートを設定することで、デフォルト ルートを IS-IS ルーティング ドメイン内に強制的に設定できます。ルート マップで設定可能な、その他のフィルタリング オプションも指定できます。
- 内部チェックサム エラーとともに受信された IS-IS LSP を無視したり、破損した LSP を消去するようにルータを設定できます。これにより、LSP の発信側は、LSP を再生成します。
- エリアおよびドメインにパスワードを割り当てられます。
- サマリー アドレスを使用して、ルーティング テーブル内に表示される集約アドレスを作成できます（経路集約）。他のルーティング プロトコルから学習したルートも集約できます。サマリーをアドバタイズするのに使用されるメトリックは、すべての個別ルートにおける最小のメトリックです。
- 過負荷ビットを設定できます。
- LSP リフレッシュ インターバルおよび LSP がリフレッシュなしでルータ データベース内にとどまることができる最大時間を設定できます。
- LSP 生成に対するスロットリング タイマー、最短パス優先計算、および部分ルート計算を設定できます。
- IS-IS 隣接関係がステートを変更（アップまたはダウン）する際に、スイッチがログ メッセージを生成するように設定できます。
- ネットワーク内のリンクが、1500 バイト未満の最大伝送単位（MTU）サイズの場合、それでもルーティングが行われるように LSP MTU の値を低くできます。

- パーティション回避ルータ コンフィギュレーション コマンドは、レベル 1-2 境界ルータ、隣接レベル 1 ルータ、およびエンド ホスト間で完全な接続が失われた場合に、エリアがパーティション化されるのを防ぎます。

IS-IS インターフェイス パラメータ

任意で、特定のインターフェイス固有の IS-IS パラメータを、付加されている他のルータとは別に設定できます。ただし、一部の値（乗数およびタイム インターバルなど）をデフォルトから変更する場合、複数のルータおよびインターフェイス上でもこれを変更する必要があります。ほとんどのインターフェイス パラメータは、レベル 1、レベル 2、またはその両方で設定できます。

次に、設定可能なインターフェイス レベル パラメータの一部を示します。

- インターフェイスのデフォルト メトリック：Quality of Service（QoS）ルーティングが実行されない場合に、IS-IS メトリックの値として使用され、割り当てられます。
- hello インターバル（インターフェイスから送信される hello パケットの間隔）またはデフォルトの hello パケット乗数：インターフェイス上で使用されて、IS-IS hello パケットで送信されるホールド タイムを決定します。ホールド タイムは、ネイバーがダウンしていると宣言するまでに、別の hello パケットを待機する時間を決定します。これにより、障害リンクまたはネイバーが検出される速さも決定し、ルートを再計算できるようになります。hello パケットが頻繁に失われ、IS-IS 隣接に無用な障害が発生する場合は、hello 乗数を変更してください。hello 乗数を大きくし、それに対応して hello インターバルを小さくすると、リンク障害を検出するのに必要な時間を増やすことなく、helloプロトコルの信頼性を高めることができます。
- その他のタイム インターバル：
  - Complete Sequence Number PDU（CSNP）インターバルCSNP は、指定ルータにより送信され、データベースの同期を維持します。
  - 再送信インターバルこれは、ポイントツーポイント リンクの IS-IS LSP の再送信間隔です。
  - IS-IS LSP 再送信スロットル インターバルこれは、IS-IS LSP がポイントツーポイント リンクで再送信される最大レート（パケット間のミリ秒数）です。このインターバルは、同じ LSP が連続する再送信間隔である再送信インターバルとは異なります。
- 指定ルータの選択プライオリティ：マルチアクセス ネットワークで必要な隣接数を削減し、その代わりに、ルーティング プロトコル トラフィックの量およびトポロジ データベースのサイズを削減できます。
- インターフェイス回線タイプ：指定されたインターフェイス上のネイバーに必要な隣接タイプです。
- インターフェイスのパスワード認証

ISO CLNS ルーティングの設定方法

IS-IS のデフォルト設定

表 *12* : *IS-IS* のデフォルト設定

機能	デフォルト設定
リンクステート PDU（LSP）エラーを無視	イネーブル
IS-IS タイプ	従来型の IS-IS：ルータは、レベル 1（ステーション）とレベル 2（エリア） 両方のルータとして機能します。 マルチエリア IS-IS：IS-IS ルーティング プロセスの最初のインスタンスが レベル 1-2 ルータです。残りのインスタンスは、レベル 1 ルータです。
デフォルト情報送信元	ディセーブル
IS-IS 隣接関係のステート変更を記録	ディセーブル
LSP 生成スロットリング タイマー	連続で生成した 2 つの間の最大インターバル：5 秒 初期 LSP 生成遅延：50 ミリ秒 1 番目と 2 番目の LSP 生成間のホールド タイム：5000 ミリ秒
LSP 最大ライフ タイム（リフレッシュなし）	LSP パケットが削除されるまで 1200 秒（20 分）
LSP リフレッシュ インターバル	LSP リフレッシュを 900 秒（15 分）ごとに送信
最大 LSP パケット サイズ	1497 バイト
NSF 認識	イネーブルレイヤ 3 スイッチでは、ハードウェアやソフトウェアの変更中に、隣接する NSF 対応ルータからのパケットを転送し続けることができます。
部分ルート計算（PRC）スロットリング タイマー	最大 PRC 待機インターバル：5 秒 トポロジの変更後の初期 PRC 計算遅延：2000 ミリ秒 1 番目と 2 番目の PRC 計算間のホールド タイム：5000 ミリ秒

機能	デフォルト設定
パーティション回避	ディセーブル
パスワード	エリアまたはドメインのパスワードが定義されておらず、認証はディセーブルになっています。
過負荷ビットの設定	ディセーブルイネーブルの際に引数が入力されない場合、過負荷ビットがただちに設定され、**no set-overload-bit** コマンドが入力されるまで設定されたままになります。
Shortest Path First（SPF）スロットリング タイマー	連続した SFP 間の最大インターバル：10 秒 トポロジの変更後の初期 SFP 計算：5500 ミリ秒 1 番目と 2 番目の SFP 計算間のホールド タイム：5500 ミリ秒
サマリー アドレス	ディセーブル

IS-IS ルーティングのイネーブル化

IS-IS をイネーブルにするには、各ルーティング プロセスに名前と NET を指定します。その後、インターフェイス上で IS-IS ルーティングをイネーブルにし、ルーティング プロセスの各インスタンスに対してエリアを指定します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**clns routing** 例： `Device(config)# clns routing`	スイッチ上でISO コネクションレス型ルーティングをイネーブルに設定します。
ステップ 3	**router isis** [*area tag*] 例： `Device(config)# router isis tag1`	指定したルーティング プロセスに対して IS-IS ルーティングをイネーブルにし、IS-IS ルーティング コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
		（任意）*area tag* 引数を使用して、IS-IS ルータが割り当てられているエリアを特定します。複数の IS-IS エリアを設定する場合は、値を入力する必要があります。 最初に設定された IS-IS インスタンスは、デフォルトでレベル 1-2 です。後のインスタンスは、自動的にレベル 1 になります。**is-type** グローバル コンフィギュレーション コマンドを使用してルーティングのレベルを変更できます。
ステップ 4	**net** network-entity-title 例： `Device(config-router)# net 47.0004.004d.0001.0001.0c11.1111.00`	ルーティング プロセスに NET を設定します。マルチエリア IS-IS を設定する場合、各ルーティング プロセスに NET を指定します。NET およびアドレスの名前を指定できます。
ステップ 5	**is-type** {**level-1** | **level-1-2** | **level-2-only**} 例： `Device(config-router)# is-type level-2-only`	（任意）レベル 1（ステーション）ルータ、マルチエリア ルーティング用のレベル 2（エリア）ルータ、または両方（デフォルト）として機能するようにルータを設定します。 • **level-1**：ステーション ルータとしてだけ機能します。 • **level-1-2**：ステーション ルータおよびエリア ルータの両方として機能します。 • **level 2**：エリア ルータとしてだけ機能します。
ステップ 6	**exit** 例： `Device(config-router)# end`	グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	IS-IS をルーティングするインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。インターフェイスがまだレイヤ 3 インターフェイスとして設定されていない場合は、**no switchport** コマンドを入力し、インターフェイスをレイヤ 3 モードにします。

	コマンドまたはアクション	目的
ステップ 8	**ip router isis** [*area tag*] 例： `Device(config-if)# ip router isis tag1`	インターフェイスに ISO CLNS の IS-IS ルーティング プロセスを設定し、エリア指示子をルーティング プロセスに割り当てます。
ステップ 9	**clns router isis** [*area tag*] 例： `Device(config-if)# clns router isis tag1`	インターフェイス上で ISO CLNS をイネーブルにします。
ステップ 10	**ip address** *ip-address-mask* 例： `Device(config-if)# ip address 10.0.0.5 255.255.255.0`	インターフェイスの IP アドレスを定義します。インターフェイスのいずれかで IS-IS ルーティングが設定されている場合は、IS-IS がイネーブルになっているエリアに含まれるすべてのインターフェイスで IP アドレスが必要です。
ステップ 11	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 12	**show isis** [*area tag*] **database detail** 例： `Device# show isis database detail`	入力を確認します。
ステップ 13	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IS-IS グローバル パラメータの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**clns routing** 例： `Device(config)# clns routing`	スイッチ上で ISO コネクションレス型ルーティングをイネーブルに設定します。
ステップ 3	**router isis** 例： `Device(config)# router isis`	IS-IS ルーティング プロトコルを指定し、ルータ コンフィギュレーション モードを開始します。
ステップ 4	**default-information originate** [**route-map** *map-name*] 例： `Device(config-router)# default-information originate route-map map1`	（任意）デフォルト ルートを IS-IS ルーティング ドメインに強制的に設定します。**route-map** *map-name* を入力すると、ルート マップが条件に一致している場合にルーティング プロセスによってデフォルト ルートが生成されます。
ステップ 5	**ignore-lsp-errors** 例： `Device(config-router)# ignore-lsp-errors`	（任意）LSP を消去する代わりに、内部チェックサムにエラーがある LSP を無視するようにルータを設定します。このコマンドは、デフォルトでイネーブルになっています（破損した LSP はドロップされます）。破損した LSP を消去するには、**no ignore-lsp-errors** ルータ コンフィギュレーション コマンドを入力します。
ステップ 6	**area-password** *password* 例： `Device(config-router)# area-password 1password`	（任意）レベル 1（ステーション ルータ レベル）LSP に挿入されるエリア認証パスワードを設定します。

	コマンドまたはアクション	目的
ステップ 7	**domain-password** *password* 例： `Device(config-router)# domain-password 2password`	（任意）レベル 2（エリア ルータ レベル）LSP に挿入されるルーティング ドメイン認証パスワードを設定します。
ステップ 8	**summary-address** *address mask* [**level-1** | **level-1-2** | **level-2**] 例： `Device(config-router)# summary-address 10.1.0.0 255.255.0.0 level-2`	（任意）所定のレベルのアドレスのサマリーを作成します。
ステップ 9	**set-overload-bit** [**on-startup** {*seconds* | **wait-for-bgp**}] 例： `Device(config-router)# set-overload-bit on-startup wait-for-bgp`	（任意）ルータに問題がある場合に、他のルータが最短パス優先（SPF）計算でこのルータを無視するように過負荷ビット（hippity ビット）を設定します。 • （任意）**on-startup**：起動時だけ過負荷ビットを設定します。**on-startup** が指定されない場合、過負荷ビットが即座に設定され、**no set-overload-bit** コマンドを入力するまで設定されたままになります。**on-startup** が指定された場合、秒数または **wait-for-bgp** を入力する必要があります。 • *seconds*：**on-startup** キーワードが設定されている場合、システム起動時に過負荷ビットが設定されて、この秒数の間設定されたままになります。指定できる範囲は 5 ～ 86400 秒です。 • **wait-for-bgp**：**on-startup** キーワードが設定されている場合、システム起動時に過負荷ビットが設定されて、BGP が収束するまで設定されたままになります。BGP が収束されたことが IS-IS に通知されない場合、IS-IS は 10 分後に過負荷ビットをオフにします。
ステップ 10	**lsp-refresh-interval** *seconds* 例： `Device(config-router)# lsp-refresh-interval 1080`	（任意）LSP リフレッシュ インターバル（秒）を設定します。範囲は 1 ～ 65535 秒です。デフォルトでは、LSP リフレッシュを 900 秒（15 分）ごとに送信します。

	コマンドまたはアクション	目的
ステップ 11	**max-lsp-lifetime** *seconds* 例： `Device(config-router)# max-lsp-lifetime 1000`	（任意）LSP パケットがリフレッシュされずにルータ データベース内に存続する最大時間を設定します。範囲は 1 ～ 65535 秒です。デフォルト値は 1200 秒（20 分）です。指定されたタイムインターバルのあと、LSP パケットは削除されます。
ステップ 12	**lsp-gen-interval** [**level-1** | **level-2**] *lsp-max-wait* [*lsp-initial-wait* *lsp-second-wait*] 例： `Device(config-router)# lsp-gen-interval level-2 2 50 100`	（任意）IS-IS 生成スロットリング タイマーを設定します。 • *lsp-max-wait*：2 つの連続する LSP 生成間の最大インターバル（秒）。指定できる範囲は 1 ～ 120 秒です。デフォルト値は 5 秒です。 • *lsp-initial-wait*：最初の LSP 生成遅延（ミリ秒）。指定できる範囲は 1 ～ 10000 ミリ秒です。デフォルト値は 50 ミリ秒です。 • *lsp-second-wait*：最初と 2 番目の LSP 生成間（ミリ秒）のホールド タイム。指定できる範囲は 1 ～ 10000 ミリ秒です。デフォルト値は 5000 ミリ秒です。
ステップ 13	**spf-interval** [**level-1** | **level-2**] *spf-max-wait* [*spf-initial-wait* *spf-second-wait*] 例： `Device(config-router)# spf-interval level-2 5 10 20`	（任意）IS-IS SPF スロットリング タイマーを設定します。 • *spf-max-wait*：連続する SFP 間（秒）の最大インターバル。指定できる範囲は 1 ～ 120 で、デフォルトは 10 です。 • *spf-initial-wait*：トポロジ変更後の最初の SFP 計算（ミリ秒）。指定できる値の範囲は 1 ～ 10000 です。デフォルトは 5500 です。 • *spf-second-wait*：最初と 2 番目の SFP 計算間（ミリ秒）のホールド タイム。指定できる値の範囲は 1 ～ 10000 です。デフォルトは 5500 です。
ステップ 14	**prc-interval** *prc-max-wait* [*prc-initial-wait* *prc-second-wait*] 例： `Device(config-router)# prc-interval 5 10 20`	（任意）IS-IS PRC スロットリング タイマーを設定します。 • *prc-max-wait*：2 つの連続する PRC 計算間の最大インターバル（秒）。指定できる範囲は 1 ～ 120 秒です。デフォルト値は 5 秒です。 • *prc-initial-wait*：トポロジ変更後の最初の PRC 計算遅延（ミリ秒）。指定できる範囲は 1 ～ 10,000 ミリ秒です。デフォルト値は 2000 ミリ秒です。

	コマンドまたはアクション	目的
		• *prc-second-wait*：最初と 2 番目の PRC 計算間（ミリ秒）のホールド タイム。指定できる範囲は 1 ～ 10,000 ミリ秒です。デフォルト値は 5000 ミリ秒です。
ステップ 15	**log-adjacency-changes** [**all**] 例： `Device(config-router)# log-adjacency-changes all`	（任意）IS-IS 隣接ステート変更をログするようルータを設定します。End System-to-Intermediate System PDU およびリンクステート パケット（LSP）など、IS-IS Hello に関連しないイベントにより生成されたすべての変更をログに含めるには、**all** を入力します。
ステップ 16	**lsp-mtu** *size* 例： `Device(config-router)# lsp mtu 1560`	（任意）最大 LSP パケット サイズ（バイト）を指定します。指定できる範囲は 128 ～ 4352 バイトです。デフォルト値は 1497 バイトです。 （注） ネットワーク内の任意のリンクで MTU サイズが縮小された場合、ネットワーク内のすべてのルータで LSP MTU サイズを変更する必要があります。
ステップ 17	**partition avoidance** 例： `Device(config-router)# partition avoidance`	（任意）境界ルータ、すべての隣接レベル 1 ルータ、およびエンド ホスト間で、フル接続が切断された場合、IS-IS レベル 1-2 境界ルータがレベル 1 エリア プレフィックスをレベル 2 バックボーンにアドバタイズしないようにします。
ステップ 18	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 19	**show clns** 例： `Device# show clns`	入力を確認します。
ステップ 20	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IS-IS インターフェイス パラメータの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	設定するインターフェイスを指定して、インターフェイス コンフィギュレーション モードを開始します。インターフェイスがまだレイヤ3インターフェイスとして設定されていない場合は、**no switchport** コマンドを入力し、インターフェイスをレイヤ3モードにします。
ステップ 3	**isis metric** *default-metric* [**level-1** | **level-2**] 例： `Device(config-if)# isis metric 15`	（任意）指定したインターフェイスにメトリック（またはコスト）を設定します。範囲は 0 ～ 63 です。デフォルトは 10 です。レベルが入力されない場合は、レベル 1 およびレベル 2 ルータの両方にデフォルト値が適用されます。
ステップ 4	**isis hello-interval** {*seconds* | **minimal**} [**level-1** | **level-2**] 例： `Device(config-if)# isis hello-interval minimal`	（任意）スイッチが hello パケットを送信する間隔を指定します。デフォルトでは、hello インターバル *seconds* の 3 倍の値が、送信される hello パケットの *holdtime* としてアドバタイズされます。hello インターバルが狭まると、トポロジ変更の検出も速くなりますが、ルーティング トラフィック量は増大します。 • **minimal**：ホールドタイムが 1 秒になるように、システムが hello 乗数に基づいて hello インターバルを計算するようにします。 • *seconds*：範囲は 1 ～ 65535 秒です。デフォルトは 10 秒です。
ステップ 5	**isis hello-multiplier** *multiplier* [**level-1** | **level-2**] 例： `Device(config-if)# isis hello-multiplier 5`	（任意）ルータが隣接装置のダウンを宣言するまでに、ネイバーが損失する IS-IS hello パケット数を指定します。指定できる範囲は 3 ～ 1000 です。デフォルトは3です。hello 乗数を小さくすると、高速コンバージェンスとなりますが、ルーティングが不安定になる場合があります。

	コマンドまたはアクション	目的
ステップ 6	**isis csnp-interval** *seconds* [**level-1** | **level-2**] 例： `Device(config-if)# isis csnp-interval 15`	（任意）インターフェイスに IS-IS CSNP を設定します。範囲は 0 ～ 65535 です。デフォルトは 10 秒です。
ステップ 7	**isis retransmit-interval** *seconds* 例： `Device(config-if)# isis retransmit-interval 7`	（任意）ポイントツーポイント リンクの IS-IS LSP の再送信間隔（秒）を設定します。指定する値は、ネットワーク上の任意の 2 つのルータ間の予測ラウンドトリップ遅延よりも大きい整数である必要があります。範囲は 0 ～ 65535 です。デフォルトは 5 秒です。
ステップ 8	**isis retransmit-throttle-interval** *milliseconds* 例： `Device(config-if)# isis retransmit-throttle-interval 4000`	（任意）IS-IS LSP 再送信スロットル インターバルを設定します。これは、IS-IS LSP がポイントツーポイント リンク上で再送信される最大レート（パケット間のミリ秒数）です。範囲は 0 ～ 65535 です。デフォルト値は、**isis lsp-interval** コマンドにより決定します。
ステップ 9	**isis priority** *value* [**level-1** | **level-2**] 例： `Device(config-if)# isis priority 50`	（任意）指定ルータ選択で使用する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。
ステップ 10	**isis circuit-type** {**level-1** | **level-1-2** | **level-2-only**} 例： `Device(config-if)# isis circuit-type level-1-2`	（任意）指定されたインターフェイス上のネイバーで必要な隣接タイプを設定します（インターフェイスの回線タイプを指定します）。 • **level-1**：このノードとネイバーの両方に共通のエリア アドレスが少なくとも 1 つある場合、レベル 1 隣接関係が確立されます。 • **level-1-2**：ネイバーもレベル 1 およびレベル 2 の両方として設定されていて、少なくとも 1 つの共通のエリアがある場合、レベル 1 およびレベル 2 隣接関係が確立されます。共通のエリアがない場合は、レベル 2 隣接関係が確立されます。これはデフォルト設定です。これはデフォルトです。 • **level 2**：レベル 2 隣接関係が確立されます。ネイバー ルータがレベル 1 ルータである場合、隣接関係は確立されません。

	コマンドまたはアクション	目的
ステップ 11	**isis password** *password* [**level-1** | **level-2**] 例： `Device(config-if)# isis password secret`	（任意）インターフェイスの認証パスワードを設定します。デフォルトでは、認証はディセーブルに設定されています。レベル 1 またはレベル 2 を指定すると、それぞれレベル 1 またはレベル 2 ルーティング用のパスワードだけがイネーブルになります。レベルを指定しない場合、デフォルトはレベル 1 およびレベル 2 です。
ステップ 12	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 13	**show clns interface** *interface-id* 例： `Device# show clns interface gigabitethernet 1/0/1`	入力を確認します。
ステップ 14	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ISO IGRP と IS-IS のモニタリングおよびメンテナンス

CLNS キャッシュのすべての内容または特定のネイバーまたはルートの情報を削除できます。ルーティング テーブル、キャッシュ、およびデータベースの内容など、特定の CLNS または IS-IS の統計情報を表示できます。また、特定のインターフェイス、フィルタ、またはネイバーに関する情報も表示できます。

次の表に、ISO CLNS および IS-IS ルーティングを消去および表示するために使用する特権 EXEC コマンドを示します。出力フィールドの詳細については、『*Cisco IOS Apollo Domain, Banyan VINES, DECnet, ISO CLNS and XNS Command Reference,*』を参照するか、Cisco IOS コマンド リファレンス マスター インデックスを使用するか、オンライン検索を行ってください。

表 *13* : *ISO CLNS* と *IS-IS* の *clear* および *show* コマンド

コマンド	目的
clear clns cache	CLNS ルーティング キャッシュをクリアおよび再初期化します。
clear clns es-neighbors	隣接データベースから End System（ES）ネイバー情報を削除します。
clear clns is-neighbors	隣接データベースから Intermediate System（IS）ネイバー情報を削除します。
clear clns neighbors	隣接データベースから CLNS ネイバー情報を削除します。
clear clns route	動的に派生した CLNS ルーティング情報を削除します。
show clns	CLNS ネットワークに関する情報を表示します。
show clns cache	CLNS ルーティング キャッシュ内のエントリを表示します。
show clns es-neighbors	ES ネイバー エントリ（関連のあるエリアなど）を表示します。
show clns filter-expr	フィルタ式を表示します。
show clns filter-set	フィルタ セットを表示します。
show clns interface [*interface-id*]	各インターフェイスの CLNS 固有の情報または ES-IS 情報を表示します。

コマンド	目的
show clns neighbor	IS-IS ネイバーに関する情報を表示します。
show clns protocol	このルータの IS-IS または ISO IGRP ルーティング プロセスごとにプロトコル固有の情報を表示します。
show clns route	このルータがCLNSパケットをルーティングする方法を把握している宛先をすべて表示します。
show clns traffic	このルータで確認されたCLNSパケットに関する情報を表示します。
show ip route isis	ISIS IP ルーティング テーブルの現在のステートを表示します。
show isis database	IS-IS リンクステート データベースを表示します。
show isis routes	IS-IS レベル 1 ルーティング テーブルを表示します。
show isis spf-log	IS-IS の Shortest Path First（SPF）計算の履歴を表示します。
show isis topology	すべてのエリアで接続済みルータのリストを表示します。
show route-map	設定されたすべてのルートマップ、または指定した1つのルート マップだけを表示します。
trace clns *destination*	ネットワークのパケットが指定された宛先までに経由するパスを検出します。

コマンド	目的
which-route {*nsap-address* | *clns-name*}	指定された CLNS の宛先が見つかったルーティング テーブルを表示します。

ISO CLNS ルーティングの設定例

例：IS-IS ルーティングの設定

次に、従来型の IS-IS を IP ルーティング プロトコルとして実行するために 3 つのルータを設定する方法を示します。従来型の IS-IS では、すべてのルータはレベル 1 およびレベル 2 のルータとして機能します（デフォルト）。

ルータ A：

```
Device(config)# clns routing
Device(config)# router isis
Device(config-router)# net 49.0001.0000.0000.000a.00
Device(config-router)# exit
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip router isis
Device(config-if)# clns router isis
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# ip router isis
Device(config-if)# clns router isis
Device(config-router)# exit
```

ルータ B：

```
Device(config)# clns routing
Device(config)# router isis
Device(config-router)# net 49.0001.0000.0000.000b.00
Device(config-router)# exit
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip router isis
Device(config-if)# clns router isis
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# ip router isis
Device(config-if)# clns router isis
Device(config-router)# exit
```

ルータ C：

```
Device(config)# clns routing
Device(config)# router isis
Device(config-router)# net 49.0001.0000.0000.000c.00
Device(config-router)# exit
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# ip router isis
Device(config-if)# clns router isis
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# ip router isis
Device(config-if)# clns router isis
Device(config-router)# exit
```

Multi-VRF CE に関する情報

バーチャル プライベート ネットワーク（VPN）は、ISP バックボーン ネットワーク上でお客様にセキュアな帯域幅共有を提供します。VPNは、共通ルーティングテーブルを共有するサイトの集合です。カスタマー サイトは、1 つまたは複数のインターフェイスでサービス プロバイダー ネットワークに接続され、サービス プロバイダーは、VRF テーブルと呼ばれる VPN ルーティング テーブルと各インターフェイスを関連付けます。

スイッチ上で IP サービスまたは拡張 IP サービス フィーチャ セットが稼働している場合、スイッチはカスタマー エッジ（CE）デバイスの複数の VRF ルーティング/転送（Multi-VRF）インスタンスをサポートします（Multi-VRF CE）。サービス プロバイダーは、Multi-VRF CE により、重複する IP アドレスで複数の VPN をサポートできます。

（注） スイッチでは、VPN のサポートのためにマルチプロトコル ラベル スイッチング（MPLS）が使用されません。

Multi-VRF CE の概要

Multi-VRF CE は、サービス プロバイダーが複数の VPN をサポートし、VPN 間で IP アドレスを重複して使用できるようにする機能です。Multi-VRF CE は入力インターフェイスを使用して、さまざまな VPN のルートを区別し、1 つまたは複数のレイヤ 3 インターフェイスと各 VRF を関連付けて仮想パケット転送テーブルを形成します。VRF 内のインターフェイスは、イーサネット ポートのように物理的なもの、または VLAN SVI のように論理的なものにもできますが、複数の VRF に属すことはできません。

（注） Multi-VRF CE インターフェイスは、レイヤ 3 インターフェイスである必要があります。

Multi-VRF CE には、次のデバイスが含まれます。

- お客様は、CEデバイスにより、1つまたは複数のプロバイダーエッジ（PE）ルータへのデータ リンクを介してサービス プロバイダー ネットワークにアクセスできます。CE デバイスは、サイトのローカル ルートをルータにアドバタイズし、リモート VPN ルートをそこから学習します。スイッチを CE に設定することができます。
- PE ルータは、スタティック ルーティング、または BGP、RIPv2、OSPF、EIGRP などのルーティング プロトコルを使用して、CE デバイスとルーティング情報を交換します。PE は、直接接続している VPN に対する VPN ルートのみを保守する必要があります。そのため、すべてのサービス プロバイダー VPN ルートを PE が保守する必要はありません。各 PE ルータは、直接接続しているサイトごとに VRF を維持します。すべてのサイトが同じ VPN に存在する場合は、PE ルータの複数のインターフェイスを 1 つの VRF に関連付けることができます。各 VPN は、指定された VRF にマッピングされます。PE ルータは、ローカル VPN ルー

トを CE から学習したあとで、IBGP を使用して別の PE ルータと VPN ルーティング情報を交換します。

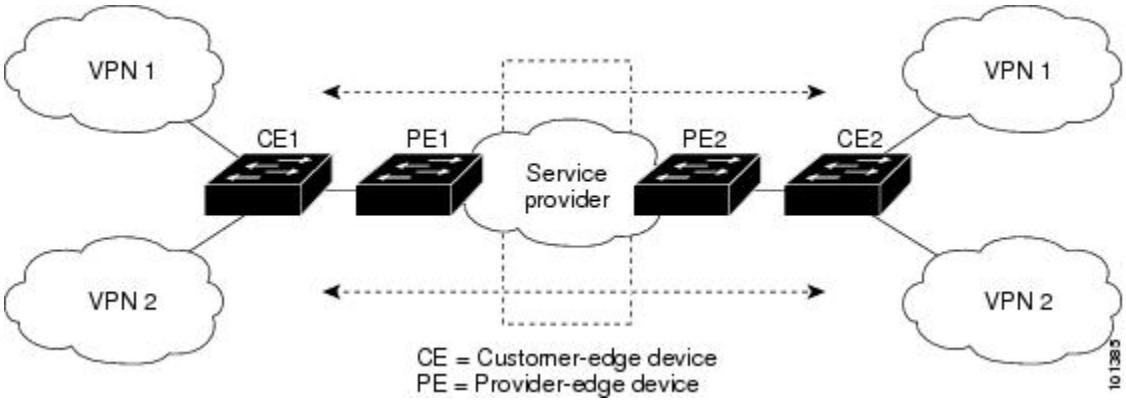
- CE デバイスに接続していないサービス プロバイダー ネットワークのルータは、プロバイダー ルータやコア ルータになります。

Multi-VRF CE では、複数のお客様が 1 つの CE を共有でき、CE と PE の間で 1 つの物理リンクだけが使用されます。共有 CE は、お客様ごとに別々の VRF テーブルを維持し、独自のルーティング テーブルに基づいて、お客様ごとにパケットをスイッチングまたはルーティングします。Multi-VRF CE は、制限付きの PE 機能を CE デバイスに拡張して、別々の VRF テーブルを維持し、VPN のプライバシーおよびセキュリティをブランチ オフィスに拡張します。

ネットワーク トポロジ

次の図に、スイッチを複数の仮想 CE として使用した構成例を示します。このシナリオは、中小企業など、VPN サービスの帯域幅要件の低いお客様に適しています。この場合、スイッチにはマルチ VRF CE のサポートが必要です。Multi-VRF CE はレイヤ 3 機能なので、VRF のそれぞれのインターフェイスはレイヤ 3 インターフェイスである必要があります。

図 *7*: 複数の仮想 *CE* として機能するスイッチ



CE スイッチは、レイヤ 3 インターフェイスを VRF に追加するコマンドを受信すると、Multi-VRF CE 関連のデータ構造で VLAN ID と Policy Label（PL）の間に適切なマッピングを設定し、VLAN ID と PL を VLAN データベースに追加します。

Multi-VRF CE を設定すると、レイヤ 3 フォワーディング テーブルは、次の 2 つのセクションに概念的に分割されます。

- Multi-VRF CE ルーティング セクションには、さまざまな VPN からのルートが含まれます。
- グローバル ルーティング セクションには、インターネットなど、VPN 以外のネットワークへのルートが含まれます。

さまざまな VRF の VLAN ID はさまざまな PL にマッピングされ、処理中に VRF を区別するために使用されます。レイヤ 3 設定機能では、学習した新しい VPN ルートごとに、入力ポートの VLAN ID を使用して PL を取得し、Multi-VRF CE ルーティング セクションに PL および新しいルートを

挿入します。ルーテッド ポートからパケットを受信した場合は、ポート内部 VLAN ID 番号が使用されます。SVI からパケットを受信した場合は、VLAN 番号が使用されます。

パケット転送処理

Multi-VRF CE 対応ネットワークのパケット転送処理は次のとおりです。

- スイッチは、VPN からパケットを受信すると、入力 PL 番号に基づいてルーティング テーブルを検索します。ルートが見つかると、スイッチはパケットを PE に転送します。
- 入力 PE は、CE からパケットを受信すると、VRF 検索を実行します。ルートが見つかると、ルータは対応する MPLS ラベルをパケットに追加し、MPLS ネットワークに送信します。
- 出力 PE は、ネットワークからパケットを受信すると、ラベルを除去してそのラベルを使用し、正しい VPN ルーティング テーブルを識別します。次に、通常のルート検索を実行します。ルートが見つかると、パケットを正しい隣接デバイスに転送します。
- CE は、出力 PE からパケットを受信すると、入力 PL を使用して正しい VPN ルーティング テーブルを検索します。ルートが見つかると、パケットを VPN 内で転送します。

ネットワーク コンポーネント

VRF を設定するには、VRF テーブルを作成し、VRF に関連するレイヤ 3 インターフェイスを指定します。次に、VPN、および CE と PE 間でルーティング プロトコルを設定します。プロバイダーのバックボーンで VPN ルーティング情報を配信する場合は、BGP が優先ルーティング プロトコルです。Multi-VRF CE ネットワークには、次の 3 つの主要コンポーネントがあります。

- VPN ルート ターゲット コミュニティ：VPN コミュニティのその他すべてのメンバのリスト。VPN コミュニティ メンバーごとに VPN ルート ターゲットを設定する必要があります。
- VPN コミュニティ PE ルータのマルチプロトコル BGP ピアリング：VPN コミュニティのすべてのメンバーに VRF 到達可能性情報を伝播します。VPN コミュニティのすべての PE ルータで BGP ピアリングを設定する必要があります。
- VPN 転送：VPN サービス プロバイダー ネットワークを介し、全 VPN コミュニティ メンバー間で、全トラフィックを伝送します。

VRF 認識サービス

IP サービスはグローバル インターフェイスに設定可能で、グローバル ルーティング インスタンスで稼働します。IP サービスは複数のルーティング インスタンス上で稼働するように拡張されます。これが、VRF 認識です。システム内の任意の設定済み VRF であればいずれも、VRF 認識サービス用に指定できます。

VRF 認識サービスは、プラットフォームに依存しないモジュールに実装されます。VRF とは、Cisco IOS 内の複数のルーティング インスタンスを意味します。各プラットフォームには、サポートする VRF 数に関して独自の制限があります。

VRF 認識サービスには、次の特性があります。

- ユーザは、ユーザ指定の VRF 内のホストに ping を実行できます。
- ARP エントリは、個別の VRF で学習されます。ユーザは、特定の VRF の ARP エントリを表示できます。

Multi-VRF CE の設定方法

Multi-VRF CE のデフォルト設定

表 *14* : *VRF* のデフォルト設定

機能	デフォルト設定
VRF	ディセーブルVRF は定義されていません。
マップ	インポートマップ、エクスポートマップ、ルートマップは定義されていません。
VRF 最大ルート数	ファスト イーサネット スイッチ：8000 ギガビット イーサネット スイッチ：12000
転送テーブル	インターフェイスのデフォルトは、グローバル ルーティング テーブルです。

Multi-VRF CE の設定時の注意事項

（注） Multi-VRF CE を使用するには、スイッチで IP サービスまたは拡張 IP サービス フィーチャ セットをイネーブルにする必要があります。

- Multi-VRF CE を含むスイッチは複数のお客様によって共有され、各お客様には独自のルーティング テーブルがあります。
- お客様は別々の VRF テーブルを使用するので、同じ IP アドレスを再利用できます。別々の VPN では IP アドレスの重複が許可されます。
- Multi-VRF CE では、複数のお客様が、PE と CE の間で同じ物理リンクを共有できます。複数の VLAN を持つトランク ポートでは、パケットがお客様間で分離されます。それぞれのお客様には独自の VLAN があります。
- Multi-VRF CE ではサポートされない MPLS-VRF 機能があります。ラベル交換、LDP 隣接関係、ラベル付きパケットはサポートされません。
- PE ルータの場合、Multi-VRF CE の使用と複数の CE の使用に違いはありません。図 41-6 では、複数の仮想レイヤ 3 インターフェイスが Multi-VRF CE デバイスに接続されています。
- スイッチでは、物理ポートか VLAN SVI、またはその両方の組み合わせを使用して、VRF を設定できます。SVI は、アクセス ポートまたはトランク ポートで接続できます。
- お客様は、別のお客様と重複しないかぎり、複数の VLAN を使用できます。お客様の VLAN は、スイッチに保存されている適切なルーティング テーブルの識別に使用される特定のルーティング テーブル ID にマッピングされます。
- スイッチは、1 つのグローバル ネットワークおよび最大 26 の VRF をサポートします。
- CE と PE の間では、ほとんどのルーティング プロトコル（BGP、OSPF、RIP、およびスタティック ルーティング）を使用できます。ただし、次の理由から External BGP（EBGP）を使用することを推奨します。
  - BGP では、複数の CE とのやり取りに複数のアルゴリズムを必要としません。
  - BGP は、さまざまな管理者によって稼働するシステム間でルーティング情報を渡すように設計されています。
  - BGP では、ルートの属性を CE に簡単に渡すことができます。
- Multi-VRF CE は、パケットのスイッチング レートに影響しません。
- VPN マルチキャストはサポートされません。
- プライベート VLAN で VRF をイネーブルにできます（逆も同様です）。
- インターフェイスでポリシーベース ルーティング（PBR）がイネーブルになっている場合は、VRF をイネーブルにできません（逆も同様です）。
- インターフェイスで Web Cache Communication Protocol（WCCP）がイネーブルになっている場合は、VRF をイネーブルにできません（逆も同様です）。

VRF の設定

コマンドの完全な構文と使用方法については、このリリースに対応するスイッチコマンドリファレンスおよび『*Cisco IOS Switching Services Command Reference*』を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ip routing** 例： `Device(config)# ip routing`	IP ルーティングをイネーブルにします。
ステップ 3	**ip vrf** *vrf-name* 例： `Device(config)# ip vrf vpn1`	VRF 名を指定し、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 4	**rd** *route-distinguisher* 例： `Device(config-vrf)# rd 100:2`	ルート識別子を指定して VRF テーブルを作成します。AS 番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。
ステップ 5	**route-target** {**export** | **import** | **both**} *route-target-ext-community* 例： `Device(config-vrf)# route-target both 100:2`	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポート ルート ターゲット コミュニティのリストを作成します。AS システム番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。*route-target-ext-community* は、ステップ 4 で入力した *route-distinguisher* と同一にする必要があります。
ステップ 6	**import map** ルート マップ 例： `Device(config-vrf)# import map importmap1`	（任意）VRF にルート マップを対応付けます。

	コマンドまたはアクション	目的
ステップ 7	**interface** *interface-id* 例： `Device(config-vrf)# interface gigabitethernet 1/0/1`	VRF に関連付けるレイヤ 3 インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。インターフェイスにはルーテッドポートまたはSVIを設定できます。
ステップ 8	**ip vrf forwarding** *vrf-name* 例： `Device(config-if)# ip vrf forwarding vpn1`	VRF をレイヤ 3 インターフェイスに対応付けます。 （注） **ip vrf forwarding** が管理インターフェイスで有効になっている場合、アクセス ポイントは加入しません。
ステップ 9	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `Device# show ip vrf interfaces vpn1`	設定を確認します。設定した VRF に関する情報を表示します。
ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

VRF 認識サービスの設定

次のサービスは、VRF 認識です。

- 『ARP』
- ping
- 簡易ネットワーク管理プロトコル（SNMP）
- ユニキャスト RPF（uRPF）
- Syslog
- traceroute

- FTP および TFTP

（注） このスイッチでは、ユニキャスト RPF（uRPF）およびネットワーク タイム プロトコル（NTP）に対して VRF 認識のサービスはサポートされません。

ARP 用 VRF 認識サービスの設定

コマンドの完全な構文と使用方法については、このリリースに対応するスイッチコマンドリファレンスおよび『*Cisco IOS Switching Services Command Reference, Release 12.4*』を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**show ip arp vrf** *vrf-name* 例： `Device# show ip arp vrf vpn1`	指定された VRF 内の ARP テーブルを表示します。

ping 用 VRF 認識サービスの設定

コマンドの完全な構文と使用方法については、このリリースに対応するスイッチコマンドリファレンスおよび『*Cisco IOS Switching Services Command Reference, Release 12.4*』を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**ping vrf***vrf-name***ip-host** 例： `Device# ping vrf vpn1 ip-host`	指定された VRF 内の ARP テーブルを表示します。

SNMP 用 VRF 認識サービスの設定

コマンドの完全な構文と使用方法については、このリリースに対応するスイッチコマンドリファレンスおよび『*Cisco IOS Switching Services Command Reference, Release 12.4*』を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**snmp-server trap authentication vrf** 例： `Device(config)# snmp-server trap authentication vrf`	VRF で、パケットに対して SNMP トラップをイネーブルにします。
ステップ 3	**snmp-server engineID remote** *host***vrf** *vpn-instance engine-id string* 例： `Device(config)# snmp-server engineID remote 172.16.20.3 vrf vpn1 80000009030000B064EFE100`	スイッチ上で、リモート SNMP エンジンの名前を設定します。
ステップ 4	**snmp-server host** *host***vrf** *vpn-instance***traps** *community* 例： `Device(config)# snmp-server host 172.16.20.3 vrf vpn1 traps comaccess`	SNMP トラップ動作の受信側、および SNMP トラップの送信に使用される VRF テーブルを指定します。
ステップ 5	**snmp-server host** *host***vrf** *vpn-instance***informs** *community* 例： `Device(config)# snmp-server host 172.16.20.3 vrf vpn1 informs comaccess`	SNMP 通知動作の受信先を指定し、SNMP 通知の送信に使用される VRF テーブルを指定します。
ステップ 6	**snmp-server user** *user group***remote** *host***vrf** *vpn-instance security model* 例： `Device(config)# snmp-server user abcd remote 172.16.20.3 vrf vpn1 priv v2c 3des secure3des`	SNMP アクセス用に、VRF 上にあるリモート ホストの SNMP グループにユーザを追加します。
ステップ 7	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

uRPF 用 VRF 認識サービスの設定

uRPF は、VRF に割り当てられたインターフェイス上で設定でき、送信元検索が VRF テーブルで実行されます。

コマンドの完全な構文と使用方法については、このリリースに対応するスイッチコマンドリファレンスおよび『*Cisco IOS Switching Services Command Reference, Release 12.4*』を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 3	**no switchport** 例： `Device(config-if)# no switchport`	レイヤ 2 コンフィギュレーション モードからインターフェイスを削除します（物理インターフェイスの場合）。
ステップ 4	**ip vrf forwarding** *vrf-name* 例： `Device(config-if)# ip vrf forwarding vpn2`	インターフェイス上で VRF を設定します。
ステップ 5	**ip address** *ip-address* 例： `Device(config-if)# ip address 10.1.5.1`	インターフェイスの IP アドレスを入力します。
ステップ 6	**ip verify unicast reverse-path** 例： `Device(config-if)# ip verify unicast reverse-path`	インターフェイス上で uRPF をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 7	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

VRF 認識 RADIUS の設定

VRF 認識 RADIUS を設定するには、まず RADIUS サーバ上で AAA をイネーブルにする必要があります。『Per VRF AAA Feature Guide』で説明されているとおり、スイッチで **ip vrf forwarding** *vrf-name* サーバグループ コンフィギュレーション コマンドと **ip radius source-interface** グローバル コンフィギュレーション コマンドがサポートされます。

syslog 用 VRF 認識サービスの設定

コマンドの完全な構文と使用方法については、このリリースに対応するスイッチコマンドリファレンスおよび『*Cisco IOS Switching Services Command Reference, Release 12.4*』を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**logging on** 例： `Device(config)# logging on`	ストレージルータイベントメッセージのロギングを、イネーブルまたは一時的にディセーブルにします。
ステップ 3	**logging host** *ip-address***vrf** *vrf-name* 例： `Device(config)# logging host 10.10.1.0 vrf vpn1`	ロギング メッセージが送信される Syslog サーバのホスト アドレスを指定します。

	コマンドまたはアクション	目的
ステップ 4	**logging buffered** *logging buffered size***debugging** 例： `Device(config)# logging buffered critical 6000 debugging`	メッセージを内部バッファにロギングします。
ステップ 5	**logging trap debugging** 例： `Device(config)# logging trap debugging`	Syslog サーバに送信されるロギング メッセージを制限します。
ステップ 6	**logging facility** *facility* 例： `Device(config)# logging facility user`	ロギング ファシリティにシステム ロギング メッセージを送信します。
ステップ 7	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

traceroute 用 VRF 認識サービスの設定

コマンドの完全な構文と使用方法については、このリリースに対応するスイッチ コマンド リファレンスおよび『Cisco IOS Switching Services Command Reference, Release 12.4』を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**traceroute vrf** *vrf-name ipaddress* 例： `Device(config)# traceroute vrf vpn2 10.10.1.1`	宛先アドレスを取得する VPN VRF の名前を指定します。

FTP および TFTP 用 VRF 認識サービスの設定

FTP および TFTP を VRF 認識とするには、いくつかの FTP/TFTP CLI を設定する必要があります。たとえば、インターフェイスに付加される VRF テーブルを使用する場合、E1/0 であれば、ip tftp source-interface E1/0 コマンドまたは ip ftp source-interface E1/0 コマンドを設定して、特定のルーティング テーブルを使用するように TFTP または FTP サーバに通知する必要があります。この例では、VRF テーブルが宛先 IP アドレスを検索するのに使用されます。これらの変更には下位互換性があり、既存の動作には影響を及ぼしません。つまり、VRF がそのインターフェイスに設定されていない場合でも、送信元インターフェイス CLI を使用して、特定のインターフェイスにパケットを送信できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ip ftp source-interface** *interface-type interface-number* 例： `Device(config)# ip ftp source-interface gigabitethernet 1/0/2`	FTP 接続の発信元 IP アドレスを指定します。
ステップ 3	**end** 例： `Device(config)#end`	特権 EXEC モードに戻ります。
ステップ 4	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 5	**ip tftp source-interface** *interface-type interface-number* 例： `Device(config)# ip tftp source-interface gigabitethernet 1/0/2`	TFTP 接続用の送信元 IP アドレスを指定します。

	コマンドまたはアクション	目的
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

マルチキャスト VRF の設定

コマンドの完全な構文と使用方法については、このリリースに対応するスイッチコマンドリファレンスおよび『*Cisco IOS IP Multicast Command Reference*』を参照してください。

Multi-VRF CE 内でのマルチキャスト設定の詳細については、『*IP Routing: Protocol-Independent Configuration Guide, Cisco IOS Release 15S*』を参照してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ip routing** 例： `Device(config)# ip routing`	IP ルーティング モードをイネーブルにします
ステップ 3	**ip vrf** *vrf-name* 例： `Device(config)# ip vrf vpn1`	VRF 名を指定し、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 4	**rd** *route-distinguisher* 例： `Device(config-vrf)# rd 100:2`	ルート識別子を指定して VRF テーブルを作成します。AS 番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。

	コマンドまたはアクション	目的
ステップ 5	**route-target** {**export** | **import** | **both**} *route-target-ext-community* 例： `Device(config-vrf)# route-target import 100:2`	指定された VRF のインポート、エクスポート、またはインポートおよびエクスポート ルート ターゲット コミュニティのリストを作成します。AS システム番号と任意の番号（xxx:y）または IP アドレスと任意の番号（A.B.C.D:y）を入力します。 *route-target-ext-community* は、ステップ 4 で入力した *route-distinguisher* と同一にする必要があります。
ステップ 6	**import map** ルート マップ 例： `Device(config-vrf)# import map importmap1`	（任意）VRF にルート マップを対応付けます。
ステップ 7	**ip multicast-routing vrf** *vrf-name***distributed** 例： `Device(config-vrf)# ip multicast-routing vrf vpn1 distributed`	（任意）VRF テーブルでグローバル マルチキャスト ルーティングをイネーブルにします。
ステップ 8	**interface** *interface-id* 例： `Device(config-vrf)# interface gigabitethernet 1/0/2`	VRF に関連付けるレイヤ 3 インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。インターフェイスはルーテッドポートまたはSVIに設定できます。
ステップ 9	**ip vrf forwarding** *vrf-name* 例： `Device(config-if)# ip vrf forwarding vpn1`	VRF をレイヤ 3 インターフェイスに対応付けます。
ステップ 10	**ip address** *ip-address***mask** 例： `Device(config-if)# ip address 10.1.5.1 255.255.255.0`	レイヤ3インターフェイスのIPアドレスを設定します。
ステップ 11	**ip pim sparse-dense mode** 例： `Device(config-if)# ip pim sparse-dense mode`	VRF に関連付けられているレイヤ 3 インターフェイス上で、PIM をイネーブルにします。

	コマンドまたはアクション	目的
ステップ 12	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 13	**show ip vrf** [**brief** | **detail** | **interfaces**] [*vrf-name*] 例： `Device# show ip vrf detail vpn1`	設定を確認します。設定した VRF に関する情報を表示します。
ステップ 14	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

VPN ルーティング セッションの設定

VPN 内のルーティングは、サポートされている任意のルーティング プロトコル（RIP、OSPF、EIGRP、BGP）、またはスタティック ルーティングで設定できます。ここで説明する設定は OSPF のものですが、その他のプロトコルでも手順は同じです。

（注） VRF インスタンス内で EIGRP ルーティング プロセスが実行されるように設定するには、**autonomous-system** *autonomous-system-number* アドレス ファミリ コンフィギュレーション モード コマンドを入力して、自律システム番号を設定する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**router ospf** *process-id***vrf** *vrf-name* 例： `Device(config)# router ospf 1 vrf vpn1`	OSPF ルーティングをイネーブルにして VPN 転送テーブルを指定し、ルータ コンフィギュレーション モードを開始します。
ステップ 3	**log-adjacency-changes** 例： `Device(config-router)# log-adjacency-changes`	（任意）隣接ステートの変更を記録します。これは、デフォルトの状態です。
ステップ 4	**redistribute bgp** *autonomous-system-number***subnets** 例： `Device(config-router)# redistribute bgp 10 subnets`	BGP ネットワークから OSPF ネットワークに情報を再配布するようにスイッチを設定します。
ステップ 5	**network** *network-number***area** *area-id* 例： `Device(config-router)# network 1 area 2`	OSPF が動作するネットワーク アドレスとマスク、およびそのネットワーク アドレスのエリア ID を定義します。
ステップ 6	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show ip ospf** *process-id* 例： `Device# show ip ospf 1`	OSPF ネットワークの設定を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

BGP PE/CE ルーティング セッションの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router bgp** *autonomous-system-number* 例： `Device(config)# router bgp 2`	その他の BGP ルータに AS 番号を渡す BGP ルーティング プロセスを設定し、ルータ コンフィギュレーション モードを開始します。
ステップ 3	**network** *network-number***mask** *network-mask* 例： `Device(config-router)# network 5 mask 255.255.255.0`	BGP を使用してアナウンスするネットワークおよびマスクを指定します。
ステップ 4	**redistribute ospf** *process-id***match internal** 例： `Device(config-router)# redistribute ospf 1 match internal`	OSPF 内部ルートを再配布するようにスイッチを設定します。
ステップ 5	**network** *network-number***area** *area-id* 例： `Device(config-router)# network 5 area 2`	OSPF が動作するネットワーク アドレスとマスク、およびそのネットワーク アドレスのエリア ID を定義します。
ステップ 6	**address-family ipv4 vrf** *vrf-name* 例： `Device(config-router)# address-family ipv4 vrf vpn1`	PE/CE ルーティング セッションの BGP パラメータを定義し、VRF アドレス ファミリ モードを開始します。
ステップ 7	**neighbor** *address***remote-as** *as-number* 例： `Device(config-router)# neighbor 10.1.1.2 remote-as 2`	PE と CE ルータの間の BGP セッションを定義します。

	コマンドまたはアクション	目的
ステップ 8	**neighbor** *address***activate** 例： `Device(config-router)# neighbor 10.2.1.1 activate`	IPv4アドレスファミリのアドバタイズメントをアクティブ化します。
ステップ 9	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show ip bgp** [**ipv4**] [**neighbors**] 例： `Device# show ip bgp ipv4 neighbors`	BGP 設定を確認します。
ステップ 11	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

Multi-VRF CE のモニタリング

表 *15*：***Multi-VRF CE*** 情報を表示するコマンド

show ip protocols vrf *vrf-name*	VRF に対応付けられたルーティング プロトコル情報を表示します。
show ip route vrf *vrf-name* [**connected**] [*protocol* [*as-number*]] [**list**] [**mobile**] [**odr**] [**profile**] [**static**] [**summary**] [**supernets-only**]	VRF に対応付けられた IP ルーティング テーブル情報を表示します。
show ip vrf [**brief** | **detail** | **interfaces**] [*vrf-name*]	定義された VRF インスタンスに関する情報を表示します。

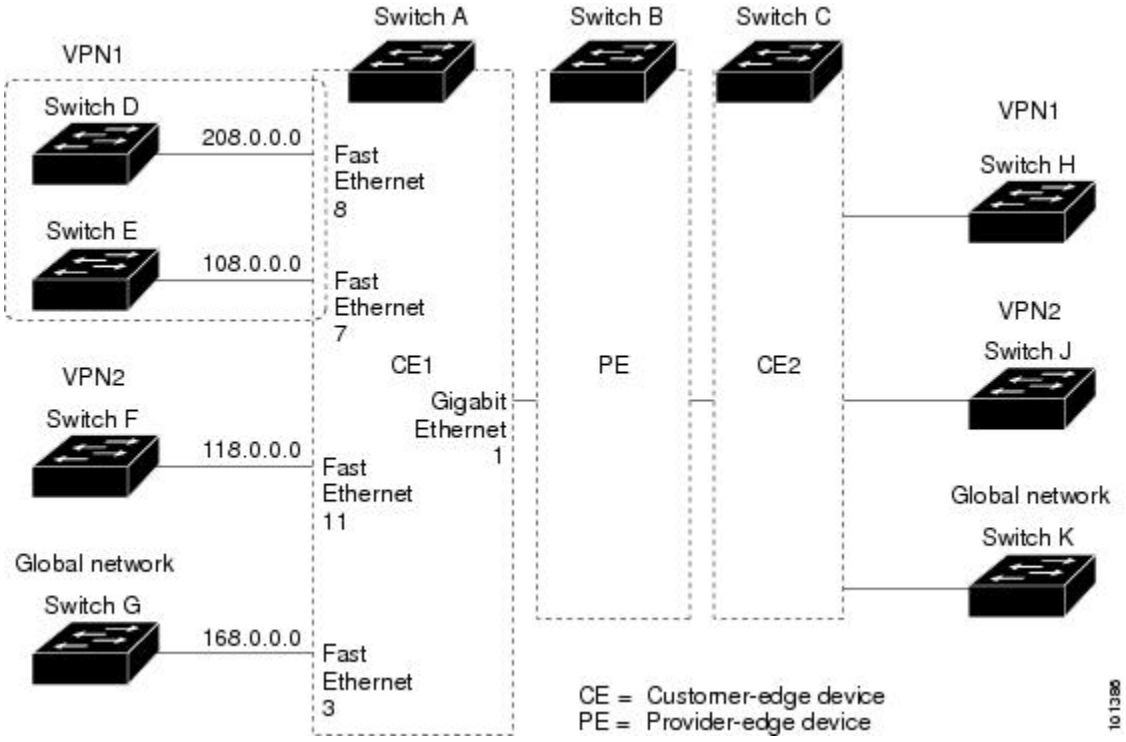
表示される情報の詳細については、『*Cisco IOS Switching Services Command Reference, Release 12.4*』を参照してください。

Multi-VRF CE の設定例

Multi-VRF CE の設定例

VPN1、VPN2、およびグローバル ネットワークで使用されるプロトコルは OSPF です。CE/PE 接続には BGP が使用されます。図のあとに続く出力は、スイッチを CE スイッチ A として設定する例、およびカスタマー スイッチ D と F の VRF 設定を示しています。CE スイッチ C とその他のカスタマー スイッチを設定するコマンドは含まれていませんが、内容は同様です。この例には、PE ルータとして動作する Catalyst 6000 スイッチまたは Catalyst 6500 スイッチのスイッチ A へのトラフィックを設定するコマンドも含まれています。

図 *8*：*Multi-VRF CE* の設定例



スイッチ A では、ルーティングをイネーブルにして VRF を設定します。

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# ip routing
Device(config)# ip vrf v11
Device(config-vrf)# rd 800:1
Device(config-vrf)# route-target export 800:1
Device(config-vrf)# route-target import 800:1
Device(config-vrf)# exit
Device(config)# ip vrf v12
Device(config-vrf)# rd 800:2
Device(config-vrf)# route-target export 800:2
```

```
Device(config-vrf)# route-target import 800:2
Device(config-vrf)# exit
```

スイッチ A のループバックおよび物理インターフェイスを設定します。ギガビット イーサネット ポート 1 は PE へのトランク接続です。ギガビット イーサネット ポート 8 と 11 は VPN に接続されます。

```
Device(config)# interface loopback1
Device(config-if)# ip vrf forwarding v11
Device(config-if)# ip address 8.8.1.8 255.255.255.0
Device(config-if)# exit

Device(config)# interface loopback2
Device(config-if)# ip vrf forwarding v12
Device(config-if)# ip address 8.8.2.8 255.255.255.0
Device(config-if)# exit

Device(config)# interface gigabitethernet1/0/5
Device(config-if)# switchport trunk encapsulation dot1q
Device(config-if)# switchport mode trunk
Device(config-if)# no ip address
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/8
Device(config-if)# switchport access vlan 208
Device(config-if)# no ip address
Device(config-if)# exit
Device(config)# interface gigabitethernet1/0/11
Device(config-if)# switchport trunk encapsulation dot1q
Device(config-if)# switchport mode trunk
Device(config-if)# no ip address
Device(config-if)# exit
```

スイッチ A で使用する VLAN を設定します。VLAN 10 は、CE と PE 間の VRF 11 によって使用されます。VLAN 20 は、CE と PE 間の VRF 12 によって使用されます。VLAN 118 と 208 は、それぞれスイッチ F とスイッチ D を含む VPN に使用されます。

```
Device(config)# interface vlan10
Device(config-if)# ip vrf forwarding v11
Device(config-if)# ip address 38.0.0.8 255.255.255.0
Device(config-if)# exit
Device(config)# interface vlan20
Device(config-if)# ip vrf forwarding v12
Device(config-if)# ip address 83.0.0.8 255.255.255.0
Device(config-if)# exit
Device(config)# interface vlan118
Device(config-if)# ip vrf forwarding v12
Device(config-if)# ip address 118.0.0.8 255.255.255.0
Device(config-if)# exit
Device(config)# interface vlan208
Device(config-if)# ip vrf forwarding v11
Device(config-if)# ip address 208.0.0.8 255.255.255.0
Device(config-if)# exit
```

VPN1 と VPN2 で OSPF ルーティングを設定します。

```
Device(config)# router ospf 1 vrf v11
Device(config-router)# redistribute bgp 800 subnets
Device(config-router)# network 208.0.0.0 0.0.0.255 area 0
Device(config-router)# exit
Device(config)# router ospf 2 vrf v12
Device(config-router)# redistribute bgp 800 subnets
Device(config-router)# network 118.0.0.0 0.0.0.255 area 0
Device(config-router)# exit
```

CE/PE ルーティングに BGP を設定します。

```
Device(config)# router bgp 800
Device(config-router)# address-family ipv4 vrf v12
Device(config-router-af)# redistribute ospf 2 match internal
Device(config-router-af)# neighbor 83.0.0.3 remote-as 100
```

```
Device(config-router-af)# neighbor 83.0.0.3 activate
Device(config-router-af)# network 8.8.2.0 mask 255.255.255.0
Device(config-router-af)# exit
Device(config-router)# address-family ipv4 vrf v11
Device(config-router-af)# redistribute ospf 1 match internal
Device(config-router-af)# neighbor 38.0.0.3 remote-as 100
Device(config-router-af)# neighbor 38.0.0.3 activate
Device(config-router-af)# network 8.8.1.0 mask 255.255.255.0
Device(config-router-af)# end
```

スイッチ D は VPN 1 に属します。次のコマンドを使用して、スイッチ A への接続を設定します。

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# ip routing
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# no switchport
Device(config-if)# ip address 208.0.0.20 255.255.255.0
Device(config-if)# exit

Device(config)# router ospf 101
Device(config-router)# network 208.0.0.0 0.0.0.255 area 0
Device(config-router)# end
```

スイッチ F は VPN 2 に属します。次のコマンドを使用して、スイッチ A への接続を設定します。

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# ip routing
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# switchport trunk encapsulation dot1q
Device(config-if)# switchport mode trunk
Device(config-if)# no ip address
Device(config-if)# exit

Device(config)# interface vlan118
Device(config-if)# ip address 118.0.0.11 255.255.255.0
Device(config-if)# exit

Device(config)# router ospf 101
Device(config-router)# network 118.0.0.0 0.0.0.255 area 0
Device(config-router)# end
```

このコマンドをスイッチ B（PE ルータ）で使用すると、CE デバイス、スイッチ A に対する接続だけが設定されます。

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip vrf v1
Router(config-vrf)# rd 100:1
Router(config-vrf)# route-target export 100:1
Router(config-vrf)# route-target import 100:1
Router(config-vrf)# exit

Router(config)# ip vrf v2
Router(config-vrf)# rd 100:2
Router(config-vrf)# route-target export 100:2
Router(config-vrf)# route-target import 100:2
Router(config-vrf)# exit
Router(config)# ip cef
Router(config)# interface Loopback1
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 3.3.1.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface Loopback2
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 3.3.2.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface gigabitethernet1/1/0.10
Router(config-if)# encapsulation dot1q 10
```

```
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 38.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface gigabitethernet1/1/0.20
Router(config-if)# encapsulation dot1q 20
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 83.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# router bgp 100
Router(config-router)# address-family ipv4 vrf v2
Router(config-router-af)# neighbor 83.0.0.8 remote-as 800
Router(config-router-af)# neighbor 83.0.0.8 activate
Router(config-router-af)# network 3.3.2.0 mask 255.255.255.0
Router(config-router-af)# exit
Router(config-router)# address-family ipv4 vrf vl
Router(config-router-af)# neighbor 38.0.0.8 remote-as 800
Router(config-router-af)# neighbor 38.0.0.8 activate
Router(config-router-af)# network 3.3.1.0 mask 255.255.255.0
Router(config-router-af)# end
```

ユニキャスト リバース パス転送の設定

uRPF 機能は、検証可能な送信元 IP アドレスが不足している IP パケットを廃棄することで、間違ったまたは偽造（スプーフィングされた）送信元 IP アドレスがネットワークに流れて発生する問題を軽減するのに役立ちます。たとえば、Smurf や Tribal Flood Network（TFN）など、多くの一般的なタイプの DoS 攻撃は、偽造された、または次々に変わる送信元 IP アドレスを使用して、攻撃を突き止めたりフィルタすることを攻撃者が阻止できるようにします。パブリック アクセスを提供するインターネット サービス プロバイダー（ISP）の場合、uRPF が IP ルーティング テーブルと整合性の取れた有効な送信元アドレスを持つパケットだけを転送することによって、そのような攻撃をそらします。この処理により、ISP のネットワーク、その顧客、および残りのインターネットが保護されます。

（注）

- uRPF は、IP サービス でサポートされます。
- スイッチが複数のスイッチ タイプが混在する混合ハードウェア スタック内にある場合は、uRPF を設定しないでください。たとえば、Catalyst 3750-X、Catalyst 3750-E、Catalyst 3750 スイッチです。

IP uRPF 設定の詳細については、『*Cisco IOS Security Configuration Guide*』の「*Other Security Features*」の章を参照してください。

プロトコル独立機能

この項では、IP ルーティング プロトコルに依存しない機能について説明します。これらの機能は、IP Base または IP Services フィーチャ セットが稼働するスイッチ上で使用できますが、IP Base フィーチャ セット付属のプロトコル関連機能は RIP でだけ使用できます。この章の IP ルーティング プロトコル独立コマンドの詳細については、『*Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols*』の「IP Routing Protocol-Independent Commands」の章を参照してください。

分散型シスコ エクスプレス フォワーディング

シスコ エクスプレス フォワーディングに関する情報

シスコエクスプレスフォワーディング（CEF）は、ネットワーク パフォーマンスを最適化するために使用されるレイヤ3 IP スイッチング技術です。CEF には高度な IP 検索および転送アルゴリズムが実装されているため、レイヤ3スイッチングのパフォーマンスを最大化できます。高速スイッチング ルート キャッシュよりも CPU にかかる負担が少ないため、CEF はより多くの CPU 処理能力をパケット転送に割り当てることができます。スイッチ スタックでは、ハードウェアによって distributed CEF（dCEF）が使用されます。動的なネットワークでは、ルーティングの変更によって、高速スイッチング キャッシュ エントリが頻繁に無効になります。高速スイッチング キャッシュエントリが無効になると、トラフィックがルートキャッシュによって高速スイッチングされずに、ルーティングテーブルによってプロセス スイッチングされることがあります。CEF および dCEF は転送情報ベース（FIB）検索テーブルを使用して、宛先ベースの IP パケット スイッチングを実行します。

CEF および dCEF での 2 つの主要なコンポーネントは、分散 FIB と分散隣接テーブルです。

- FIB はルーティング テーブルや情報ベースと同様、IP ルーティング テーブルに転送情報のミラーイメージが保持されます。ネットワーク内でルーティングまたはトポロジが変更されると、IP ルーティング テーブルがアップデートされ、これらの変更が FIB に反映されます。FIB には、IP ルーティング テーブル内の情報に基づいて、ネクストホップのアドレス情報が保持されます。FIB にはルーティング テーブル内の既知のルートがすべて格納されているため、CEF はルート キャッシュをメンテナンスする必要がなく、トラフィックのスイッチングがより効率化され、トラフィック パターンの影響も受けません。
- リンク層上でネットワーク内のノードが 1 ホップで相互に到達可能な場合、これらのノードは隣接関係にあると見なされます。CEF は隣接テーブルを使用し、レイヤ 2 アドレッシング情報を付加します。隣接テーブルには、すべての FIB エントリに対する、レイヤ 2 のネクストホップのアドレスが保持されます。

スイッチまたはスイッチスタックは、ギガビット速度の回線レート IP トラフィックを達成するため特定用途向け集積回路（ASIC）を使用しているので、CEF または dCEF 転送はソフトウェア転送パス（CPU により転送されるトラフィック）にだけ適用されます。

シスコ エクスプレス フォワーディングの設定方法

デフォルトで、CEF または dCEF はグローバルにイネーブルに設定されています。何らかの理由でこれがディセーブルになった場合は、**ip cef** または **ip cef distributed** グローバル コンフィギュレーション コマンドを使用し、再度イネーブルに設定できます。

デフォルト設定では、すべてのレイヤ3インターフェイスでCEFまたはdCEFがイネーブルです。**no ip route-cache cef** インターフェイス コンフィギュレーション コマンドを入力すると、ソフトウェアが転送するトラフィックに対してCEFがディセーブルになります。このコマンドは、ハードウェア転送パスには影響しません。CEF をディセーブルにして **debug ip packet detail** 特権 EXEC

コマンドを使用すると、ソフトウェア転送トラフィックをデバッグするのに便利です。ソフトウェア転送パス用のインターフェイスで CEF をイネーブルにするには、**ip route-cache cef** インターフェイス コンフィギュレーション コマンドを使用します。

注意 CLI には、インターフェイス上で CEF をディセーブルにする **no ip route-cache cef** インターフェイス コンフィギュレーション コマンドが表示されますが、デバッグ以外の目的でインターフェイス上で CEF または dCEF をディセーブルにしないようにしてください。

ディセーブルである CEF または dCEF をグローバルにイネーブルにしたり、ソフトウェア転送トラフィックのインターフェイス上でイネーブルにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ip cef** 例： `Device(config)# ip cef`	非スタッキング スイッチで CEF の動作をイネーブルにします。 ステップ 4 に進みます。
ステップ 3	**ip cef distributed** 例： `Device(config)# ip cef distributed`	アクティブ スイッチで CEF の動作をイネーブルにします。
ステップ 4	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 5	**ip route-cache cef** 例： `Device(config-if)# ip route-cache cef`	ソフトウェア転送トラフィック用のインターフェイスで CEF をイネーブルにします。
ステップ 6	**end** 例： `Device(config-if)# end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 7	**show ip cef** 例： `Device# show ip cef`	すべてのインターフェイスの CEF ステータスを表示します。
ステップ 8	**show cef linecard** [**detail**] 例： `Device# show cef linecard detail`	（任意）非スタッキング スイッチの CEF 関連インターフェイス情報を表示します。
ステップ 9	**show cef linecard** [*slot-number*] [**detail**] 例： `Device# show cef linecard 5 detail`	（任意）スタック内のすべてのスイッチ、または指定されたスイッチに対して、スイッチの CEF 関連インターフェイス情報をスタック メンバ別に表示します。 （任意）*slot-number* には、スタック メンバーのスイッチ番号を入力します。
ステップ 10	**show cef interface** [*interface-id*] 例： `Device# show cef interface gigabitethernet 1/0/1`	すべてのインターフェイスまたは指定されたインターフェイスの詳細な CEF 情報を表示します。
ステップ 11	**show adjacency** 例： `Device# show adjacency`	CEF の隣接テーブル情報を表示します。
ステップ 12	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

等コスト ルーティング パスの個数

等コスト ルーティング パスに関する情報

同じネットワークへ向かう同じメトリックのルートが複数ルータに格納されている場合、これらのルートは等価コストを保有していると見なされます。ルーティング テーブルに複数の等コスト ルートが含まれる場合は、これらをパラレル パスと呼ぶこともあります。ネットワークへの等コ

スト パスがルータに複数格納されている場合、ルータはこれらを同時に使用できます。パラレル パスを使用すると、パスに障害が発生した場合に冗長性を確保できます。また、使用可能なパスにパケットの負荷を分散し、使用可能な帯域幅を有効利用することもできます。等コスト ルートは、スタック内の各スイッチでサポートされます。

等コスト ルートはルータによって自動的に取得、設定されますが、ルーティング テーブルの IP ルーティング プロトコルでサポートされるパラレル パスの最大数は制御可能です。スイッチ ソフトウェア では最大 32 の等コスト ルーティングが許可されていますが、スイッチ ハードウェアはルートあたり 17 パス以上は使用しません。

等コスト ルーティング パスの設定方法

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router** {**bgp** | **rip** | **ospf** | **eigrp**} 例： `Device(config)# router eigrp`	ルータ コンフィギュレーション モードを開始します。
ステップ 3	**maximum-paths** *maximum* 例： `Device(config-router)# maximum-paths 2`	プロトコル ルーティング テーブルのパラレル パスの最大数を設定します。指定できる範囲は 1 ～ 16 です。ほとんどの IP ルーティング プロトコルでデフォルトは 4 ですが、BGP の場合だけ 1 です。
ステップ 4	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show ip protocols** 例： `Device# show ip protocols`	*Maximum path* フィールドの設定を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

スタティック ユニキャスト ルート

スタティック ユニキャスト ルートに関する情報

スタティックユニキャストルートは、特定のパスを通過して送信元と宛先間でパケットを送受信するユーザ定義のルートです。ルータが特定の宛先へのルートを構築できない場合、スタティック ルートは重要で、到達不能なすべてのパケットが送信される最終ゲートウェイを指定する場合に有効です。

ユーザによって削除されるまで、スタティック ルートはスイッチに保持されます。ただし、アドミニストレーティブディスタンスの値を割り当て、スタティック ルートをダイナミック ルーティング情報で上書きできます。各ダイナミック ルーティング プロトコルには、デフォルトのアドミニストレーティブ ディスタンスが設定されています（表 10 を参照）。ダイナミック ルーティング プロトコルの情報でスタティック ルートを上書きする場合は、スタティック ルートのアドミニストレーティブ ディスタンスがダイナミック プロトコルのアドミニストレーティブ ディスタンスよりも大きな値になるように設定します。

表 *16*：ダイナミック ルーティング プロトコルのデフォルトのアドミニストレーティブ ディスタンス

ルートの送信元	デフォルト距離
接続されているインターフェイス	0
スタティック ルート	1
EIGRP サマリー ルート	5
外部 BGP	20
内部 EIGRP	90
IGRP	100
OSPF	110
内部 BGP	200
不明（Unknown）	225

インターフェイスを指し示すスタティック ルートは、RIP、IGRP、およびその他のダイナミック ルーティング プロトコルを通してアドバタイズされます。**redistribute** スタティック ルータ コンフィギュレーション コマンドが、これらのルーティング プロトコルに対して指定されているかど

うかは関係ありません。これらのスタティック ルートがアドバタイズされるのは、インターフェイスを指し示すスタティック ルートが接続された結果、静的な性質を失ったとルーティング テーブルで見なされるためです。ただし、network コマンドで定義されたネットワーク以外のインターフェイスに対してスタティック ルートを定義する場合は、ダイナミック ルーティング プロトコルに **redistribute** スタティック コマンドを指定しない限り、ルートはアドバタイズされません。

インターフェイスがダウンすると、ダウンしたインターフェイスを経由するすべてのスタティック ルートが IP ルーティング テーブルから削除されます。転送ルータのアドレスとして指定されたアドレスへ向かう有効なネクスト ホップがスタティック ルート内に見つからない場合は、IP ルーティング テーブルからそのスタティック ルートも削除されます。

スタティック ユニキャスト ルートの設定

スタティックユニキャストルートは、特定のパスを通過して送信元と宛先間でパケットを送受信するユーザ定義のルートです。ルータが特定の宛先へのルートを構築できない場合、スタティック ルートは重要で、到達不能なすべてのパケットが送信される最終ゲートウェイを指定する場合に有効です。

スタティック ルートを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip route prefix mask** {*address* | *interface*} [*distance*] 例： `Device(config)# ip route prefix mask gigabitethernet 1/0/4`	スタティック ルートを確立します。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 5	**show ip route** 例： `Device# show ip route`	設定を確認するため、ルーティングテーブルの現在の状態を表示します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

次の作業

スタティック ルートを削除するには、**no ip route** *prefix mask* {*address*| *interface*} グローバル コンフィギュレーション コマンドを使用します。ユーザによって削除されるまで、スタティック ルートはスイッチに保持されます。

デフォルトのルートおよびネットワーク

デフォルトのルートおよびネットワークに関する情報

ルータは、他のすべてのネットワークへのルートを学習できません。完全なルーティング機能を実現するには、一部のルータをスマート ルータとして使用し、それ以外のルータのデフォルト ルートをスマート ルータ宛てに指定します（スマート ルータにはインターネットワーク全体のルーティング テーブルに関する情報が格納されます）。これらのデフォルト ルートは動的に学習できますが、ルータごとに設定することもできます。ほとんどのダイナミックな内部ルーティング プロトコルには、スマート ルータを使用してデフォルト情報を動的に生成し、他のルータに転送するメカニズムがあります。

指定されたデフォルト ネットワークに直接接続されたインターフェイスがルータに存在する場合は、そのデバイス上で動作するダイナミック ルーティング プロトコルによってデフォルト ルートが生成されます。RIP の場合は、疑似ネットワーク 0.0.0.0 がアドバタイズされます。

ネットワークのデフォルトを生成しているルータには、そのルータ自身のデフォルト ルートも指定する必要があります。ルータが自身のデフォルト ルートを生成する方法の 1 つは、適切なデバイスを経由してネットワーク 0.0.0.0 に至るスタティック ルートを指定することです。

ダイナミック ルーティング プロトコルによってデフォルト情報を送信するときは、特に設定する必要はありません。ルーティング テーブルは定期的にスキャンされ、デフォルト ルートとして最適なデフォルト ネットワークが選択されます。IGRP ネットワークでは、システムのデフォルト ネットワークの候補が複数存在する場合もあります。Cisco ルータでは、デフォルト ルートまた

は最終ゲートウェイを設定するため、アドミニストレーティブ ディスタンスおよびメトリック情報を使用します。

ダイナミックなデフォルト情報がシステムに送信されない場合は、**ip default-network** グローバル コンフィギュレーション コマンドを使用し、デフォルト ルートの候補を指定します。このネットワークが任意の送信元のルーティング テーブルに格納されている場合は、デフォルト ルートの候補としてフラグ付けされます。ルータにデフォルト ネットワークのインターフェイスが存在しなくても、そこへのパスが格納されている場合、そのネットワークは 1 つの候補と見なされ、最適なデフォルト パスへのゲートウェイが最終ゲートウェイになります。

デフォルトのルートおよびネットワークの設定方法

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**ip default-network** *network number* 例： `Device(config)# ip default-network 1`	デフォルト ネットワークを指定します。
ステップ 3	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 4	**show ip route** 例： `Device# show ip route`	最終ゲートウェイで選択されたデフォルト ルートを表示します。
ステップ 5	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルーティング情報を再配信するためのルート マップ

ルート マップの概要

スイッチでは複数のルーティング プロトコルを同時に実行し、ルーティング プロトコル間で情報を再配信できます。ルーティング プロトコル間での情報の再配信は、サポートされているすべての IP ベース ルーティング プロトコルに適用されます。

2 つのドメイン間で拡張パケット フィルタまたはルート マップを定義することにより、ルーティング ドメイン間でルートの再配信を条件付きで制御することもできます。**match** および **set** ルート マップ コンフィギュレーション コマンドは、ルート マップの条件部を定義します。**match** コマンドは、条件が一致する必要があることを指定しています。**set** コマンドは、ルーティング アップデートが match コマンドで定義した条件を満たす場合に行われる処理を指定します。再配信はプロトコルに依存しない機能ですが、**match** および **set** ルート マップ コンフィギュレーション コマンドの一部は特定のプロトコル固有のものです。

match コマンドのあとに、**set** コマンドおよび **route-map** コマンドをそれぞれ 1 つまたは複数指定します。**match** コマンドを指定しない場合は、すべて一致すると見なされます。**set** コマンドを指定しない場合、一致以外の処理はすべて実行されません。このため、少なくとも 1 つの **match** または **set** コマンドを指定する必要があります。

（注） **set** ルート マップ コンフィギュレーション コマンドを使用しないルート マップは、CPU に送信されるので、CPU の使用率が高くなります。

ルートマップ ステートメントは、**permit** または **deny** として識別することもできます。ステートメントが拒否としてマークされている場合、一致基準を満たすパケットは通常の転送チャネルを通じて送り返されます（宛先ベース ルーティング）、ステートメントが許可としてマークされている場合は、一致基準を満たすパケットに set コマンドが適用されます。一致基準を満たさないパケットは、通常のルーティング チャネルを通じて転送されます。

ルート マップの設定方法

次に示すステップ 3 ～ 14 はそれぞれ任意ですが、少なくとも 1 つの **match** ルート マップ コンフィギュレーション コマンド、および 1 つの **set** ルート マップ コンフィギュレーション コマンドを入力する必要があります。

（注） キーワードは、ルート配信を制御する手順で定義されているものと同じです。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**route-map***map-tag* [**permit** | **deny**] [*sequence number*] 例： `Device(config)# route-map rip-to-ospf permit 4`	再配信を制御するために使用するルートマップを定義し、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。 *map-tag*：ルート マップ用のわかりやすい名前を指定します。**redistribute** ルータ コンフィギュレーション コマンドはこの名前を使用して、このルート マップを参照します。複数のルート マップで同じマップ タグ名を共有できます。 （任意）**permit** が指定され、このルート マップの一致条件が満たされている場合は、set アクションの制御に従ってルートが再配信されます。**deny** が指定されている場合、ルートは再配信されません。 *sequence number*（任意）：同じ名前によってすでに設定されているルート マップのリスト内で、新しいルートマップの位置を指定する番号です。
ステップ 3	**match as-path** *path-list-number* 例： `Device(config-route-map)#match as-path 10`	BGP AS パス アクセス リストと照合します。
ステップ 4	**match community-list** *community-list-number* [**exact**] 例： `Device(config-route-map)# match community-list 150`	BGP コミュニティ リストのマッチングを行います。
ステップ 5	**match ip address** {*access-list-number* | *access-list-name*} [*...access-list-number* | *...access-list-name*]	名前または番号を指定し、標準アクセス リストと照合します。1 ～ 199 の整数を指定できます。

	コマンドまたはアクション	目的
	例： `Device(config-route-map)# match ip address 5 80`	
ステップ 6	**match metric** metric-value 例： `Device(config-route-map)# match metric 2000`	指定されたルート メトリックと一致させます。*metric-value* には、0 ～ 4294967295 の値が指定された、EIGRP のメトリックを指定できます。
ステップ 7	**match ip next-hop** {*access-list-number* | *access-list-name*} [*...access-list-number* | *...access-list-name*] 例： `Device(config-route-map)# match ip next-hop 8 45`	指定されたアクセス リスト（番号 1 ～ 199）のいずれかで送信される、ネクストホップのルータ アドレスと一致させます。
ステップ 8	**match tag** *tag value* [*...tag-value*] 例： `Device(config-route-map)# match tag 3500`	1 つまたは複数のルート タグ値からなるリスト内の指定されたタグ値と一致させます。0 ～ 4294967295 の整数を指定できます。
ステップ 9	**match interface***type number* [*...type-number*] 例： `Device(config-route-map)# match interface gigabitethernet 1/0/1`	指定されたインターフェイスの 1 つから、指定されたネクストホップへのルートと一致させます。
ステップ 10	**match ip route-source** {*access-list-number* | *access-list-name*} [*...access-list-number* | *...access-list-name*] 例： `Device(config-route-map)# match ip route-source 10 30`	アドバタイズされた指定のアクセス リストによって指定したアドレスに一致します。
ステップ 11	**match route-type** {**local** | **internal** | **external** [**type-1** | **type-2**]}	指定された **route-type** と一致させます。 • **local**：ローカルに生成された BGP ルート。

	コマンドまたはアクション	目的
	例： `Device(config-route-map)# match route-type local`	• **internal**：OSPF エリア内およびエリア間ルート、または EIGRP 内部ルート。 • **external**：OSPF 外部ルート（タイプ 1 またはタイプ 2）または EIGRP 外部ルート。
ステップ 12	**set dampening** *halflife reuse suppress max-suppress-time* 例： `Device(config-route-map)# set dampening 30 1500 10000 120`	BGP ルート ダンプニング係数を設定します。
ステップ 13	**set local-preference** *value* 例： `Device(config-route-map)# set local-preference 100`	ローカル BGP パスに値を割り当てます。
ステップ 14	**set origin** {**igp** | **egp** *as* | **incomplete**} 例： `Device(config-route-map)#set origin igp`	BGP 送信元コードを設定します。
ステップ 15	**set as-path** {**tag** | **prepend** *as-path-string*} 例： `Device(config-route-map)# set as-path tag`	BGP の自律システム パスを変更します。
ステップ 16	**set level** {**level-1** | **level-2** | **level-1-2** | **stub-area** | **backbone**} 例： `Device(config-route-map)# set level level-1-2`	ルーティング ドメインの指定エリアにアドバタイズされるルートのレベルを設定します。**stub-area** および **backbone** は、OSPF NSSA およびバックボーン エリアです。
ステップ 17	**set metric** *metric value* 例： `Device(config-route-map)# set metric 100`	再配布されるルートを指定するためのメトリック値を設定します（EIGRP のみ）。*metric value* は -294967295 ～ 294967295 の整数です。

	コマンドまたはアクション	目的
ステップ 18	**set metric***bandwidth delay reliability loading mtu* 例： `Device(config-route-map)# set metric 10000 10 255 1 1500`	再配布されるルートを指定するためのメトリック値を設定します（EIGRP のみ）。 • *bandwidth*：0 ～ 4294967295 の範囲のルートのメトリック値または IGRP 帯域幅（キロビット/秒単位）。 • *delay*：0 ～ 4294967295 の範囲のルート遅延（10 マイクロ秒単位）。 • *reliability*：0 ～ 255 の数値で表されるパケット伝送の成功可能性。255 は信頼性が 100% であること、0 は信頼性がないことを意味します。 • *loading*：0 ～ 255 の数値で表されるルートの有効帯域幅（255 は 100% の負荷）。 • *mtu*：ルートの MTU の最小サイズ（バイト単位）。範囲は 0 ～ 4294967295 です。
ステップ 19	**set metric-type** {**type-1** | **type-2**} 例： `Device(config-route-map)# set metric-type type-2`	再配信されるルートに OSPF 外部メトリック タイプを設定します。
ステップ 20	**set metric-type internal** 例： `Device(config-route-map)# set metric-type internal`	ネクストホップの IGP メトリックと一致するように、EBGP ネイバーにアドバタイズされるプレフィックスの Multi-Exit 識別子（MED）値を設定します。
ステップ 21	**set weight** *number* 例： `Device(config-route-map)# set weight 100`	ルーティング テーブルの BGP 重みを設定します。指定できる値は 1 ～ 65535 です。
ステップ 22	**end** 例： `Device(config-route-map)# end`	特権 EXEC モードに戻ります。
ステップ 23	**show route-map** 例： `Device# show route-map`	設定を確認するため、設定されたすべてのルート マップ、または指定されたルート マップだけを表示します。

	コマンドまたはアクション	目的
ステップ 24	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルート配信の制御方法

次に示すステップ 3 ～ 14 はそれぞれ任意ですが、少なくとも 1 つの **match** ルート マップ コンフィギュレーション コマンド、および 1 つの **set** ルート マップ コンフィギュレーション コマンドを入力する必要があります。

（注） キーワードは、再配信用にルート マップを設定する手順で定義されているものと同じです。

ルーティング プロトコルのメトリックを、必ずしも別のルーティング プロトコルのメトリックに変換する必要はありません。たとえば、RIP メトリックはホップ カウントで、IGRP メトリックは 5 つの特性の組み合わせです。このような場合は、メトリックを独自に設定し、再配信されたルートに割り当てます。ルーティング情報を制御せずにさまざまなルーティング プロトコル間で交換するとルーティング ループが発生し、ネットワーク動作が著しく低下することがあります。

メトリック変換の代わりに使用されるデフォルトの再配信メトリックが定義されていない場合は、ルーティング プロトコル間で自動的にメトリック変換が発生することがあります。

- RIP はスタティック ルートを自動的に再配信できます。スタティック ルートにはメトリック 1（直接接続）が割り当てられます。
- デフォルト モードになっている場合、どのプロトコルも他のルーティング プロトコルを再配信できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 2	**router** {**rip** | **ospf** | **eigrp**} 例： `Device(config)# router eigrp 10`	ルータ コンフィギュレーション モードを開始します。
ステップ 3	**redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** *type-value*] [**match internal** | **external** *type-value*] [**tag** *tag-value*] [**route-map** *map-tag*] [**weight** *weight*] [**subnets**] 例： `Device(config-router)# redistribute eigrp 1`	ルーティング プロトコル間でルートを再配信します。route-map を指定しないと、すべてのルートが再配信されます。キーワード **route-map** に *map-tag* を指定しないと、ルートは配信されません。
ステップ 4	**default-metric** *number* 例： `Device(config-router)# default-metric 1024`	現在のルーティング プロトコルが、再配信されたすべてのルートに対して同じメトリック値を使用するように設定します（RIP、OSPF）。
ステップ 5	**default-metric** bandwidth delay reliability loading mtu 例： `Device(config-router)# default-metric 1000 100 250 100 1500`	EIGRP ルーティング プロトコルが、EIGRP 以外で再配信されたすべてのルートに対して同じメトリック値を使用するように設定します。
ステップ 6	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show route-map** 例： `Device# show route-map`	設定を確認するため、設定されたすべてのルート マップ、または指定されたルート マップだけを表示します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

Policy-Based Routing：ポリシーベース ルーティング

ポリシーベース ルーティングの概要

PBR を使用すると、トラフィック フローに定義済みポリシーを設定できます。PBR を使用してルーティングをより細かく制御するには、ルーティング プロトコルから取得したルートの信頼度を小さくします。PBR は、次の基準に基づいて、パスを許可または拒否するルーティング ポリシーを設定したり、実装したりできます。

- 特定のエンド システムの ID
- アプリケーション
- プロトコル

PBR を使用すると、等価アクセスや送信元依存ルーティング、インタラクティブ対バッチ トラフィックに基づくルーティング、専用リンクに基づくルーティングを実現できます。たとえば、在庫記録を本社に送信する場合は高帯域で高コストのリンクを短時間使用し、電子メールなど日常的に使用するアプリケーション データは低帯域で低コストのリンクで送信できます。

PBRがイネーブルの場合は、アクセス コントロール リスト（ACL）を使用してトラフィックを分類し、各トラフィックがそれぞれ異なるパスを経由するようにします。PBR は着信パケットに適用されます。PBR がイネーブルのインターフェイスで受信されたすべてのパケットは、ルート マップを通過します。ルート マップで定義された基準に基づいて、パケットは適切なネクスト ホップに転送（ルーティング）されます。

- 許可とマークされているルート マップ文は次のように処理されます。
  - match コマンドは長さまたは複数の ACL で照合できます。ルート マップ文には複数の match コマンドを含めることができます。論理関数またはアルゴリズム関数は、許可または拒否の決定がされるまで、すべての match コマンドで実行されます。
    次に例を示します。

    match length A B

    match ip address acl1 acl2

    match ip address acl3

  パケットは、match length A B または acl1 または acl2 または acl3 により許可される場合に許可されます。

  - 決定が許可の場合は、set コマンドで指定されたアクションがパケットで適用されます。
  - 下された決定が拒否の場合は、PBR アクション（set コマンドで指定された）が適用されません。代わりに、処理ロジックが、シーケンス内の次のルート マップ文（シーケンス番号が次に高い文）に移動します。次の文が存在しない場合は、PBR 処理が終了し、パケットがデフォルトの 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。

- PBR では、拒否としてマークされているルートマップ ステートメントはサポートされません。

標準 IP ACL を使用すると、アプリケーション、プロトコル タイプ、またはエンド ステーションに基づいて一致基準を指定するように、送信元アドレスまたは拡張IP ACLの一致基準を指定できます。一致が見つかるまで、ルート マップにこのプロセスが行われます。一致が見つからない場合、通常の宛先ベース ルーティングが行われます。match ステートメント リストの末尾には、暗黙の拒否ステートメントがあります。

match 句が満たされた場合は、set 句を使用して、パス内のネクスト ホップ ルータを識別する IP アドレスを指定できます。

PBR コマンドおよびキーワードの詳細については、『*Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols*』を参照してください。

PBR の設定方法

- PBR を使用するには、スイッチまたはスタック マスター上で IP Base フィーチャ セットをイネーブルにしておく必要があります。
- マルチキャスト トラフィックには、ポリシーによるルーティングが行われません。PBR が適用されるのはユニキャスト トラフィックだけです。
- ルーテッド ポートまたは SVI 上で、PBR をイネーブルにできます。
- スイッチは一致長に基づき PBR をサポートします。
- レイヤ 3 モードの EtherChannel ポート チャネルにはポリシー ルート マップを適用できますが、EtherChannel のメンバーである物理インターフェイスには適用できません。適用しようとすると、コマンドが拒否されます。ポリシー ルート マップが適用されている物理インターフェイスは、EtherChannel のメンバーになることができません。
- スイッチまたはスイッチ スタックには最大 128 個の IP ポリシー ルート マップを定義できます。
- スイッチまたはスイッチ スタックには、PBR 用として最大 512 個のアクセス コントロール エントリ（ACE）を定義できます。
- ルート マップに一致基準を設定する場合は、次の注意事項に従ってください。
  - ローカル アドレス宛てのパケットを許可する ACL と照合させないでください。
- VRF と PBR は、スイッチ インターフェイス上で相互に排他的です。PBR がインターフェイスでイネーブルになっているときは、VRF をイネーブルにはできません。その反対の場合も同じで、VRF がインターフェイスでイネーブルになっているときは、PBR をイネーブルにできません。
- WCCP と PBR は、スイッチ インターフェイスで相互に排他的です。PBR がインターフェイスでイネーブルになっているときは、WCCP をイネーブルにできません。その反対の場合も同じで、WCCP がインターフェイスでイネーブルになっているときは、PBR をイネーブルにできません。

- PBR で使用されるハードウェア エントリ数は、ルート マップ自体、使用される ACL、ACL およびルート マップ エントリの順序によって異なります。
- TOS、DSCP、および IP Precedence に基づく PBR はサポートされません。
- set interface、set default next-hop、および set default interface はサポートされません。
- **ip next-hop recursive** および **ip next-hop verify availability** 機能は使用できません。next-hop は、直接接続される必要があります。
- setアクションのないポリシーマップはサポートされます。一致パケットは通常どおりにルーティングされます。
- match 句のないポリシーマップはサポートされます。setアクションはすべてのパケットに適用されます。

デフォルトでは、PBR はスイッチ上でディセーブルです。PBR をイネーブルにするには、一致基準および結果アクションを指定するルート マップを作成する必要があります。次に、特定のインターフェイスでそのルートマップ用のPBRをイネーブルにします。指定したインターフェイスに着信したパケットのうち、match 句と一致したものはすべて PBR の対象になります。

スイッチで生成されたパケットまたはローカルパケットは、通常どおりにポリシールーティングされません。スイッチ上でローカルPBRをグローバルにイネーブルにすると、そのスイッチから送信されたすべてのパケットがローカル PBR の影響を受けます。ローカル PBR は、デフォルトでディセーブルに設定されています。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**route-map** *map-tag* [**permit**] [*sequence number*] 例： `Device(config)# route-map pbr-map permit`	パケットの出力場所を制御するために使用するルート マップを定義し、ルート マップのコンフィギュレーション モードを開始します。 • *map-tag* — ルート マップ用のわかりやすい名前。 **ip policy route-map** インターフェイス コンフィギュレーション コマンドは、この名前を使用して、このルートマップを参照します。同じ map-tag がある複数の route-map 文は、1 つの route-map を定義します。 • （任意） **permit** — **permit** が指定され、このルートマップの一致条件が満たされている場合は、setアクションの制御に従ってルートがポリシー ルーティングされます。

	コマンドまたはアクション	目的
		• （任意） *sequence number* — シーケンス番号は、特定のルート マップ内の route-map ステートメントの位置を示します。
ステップ 3	**match ip address** {*access-list-number* | *access-list-name*} [*access-list-number* |*...access-list-name*] 例： `Device(config-route-map)# match ip address 110 140`	1 つ以上の標準または拡張アクセス リストで許可されている送信元および宛先 IP アドレスを照合します。ACL は、複数の送信元および宛先 IP アドレスでも照合できます。 **match** コマンドを指定しない場合、ルート マップはすべてのパケットに適用されます。
ステップ 4	**match length min max** 例： `Device(config-route-map)# match length 64 1500`	パケット長と照合します。
ステップ 5	**set ip next-hop** *ip-address* [*...ip-address*] 例： `Device(config-route-map)# set ip next-hop 10.1.6.2`	基準と一致するパケットの動作を指定します。パケットのルーティング先となるネクスト ホップを設定します（ネクスト ホップは隣接している必要があります）。
ステップ 6	**exit** 例： `Device(config-route-map)# exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet 1/0/1`	インターフェイス コンフィギュレーション モードを開始し、設定するインタフェースを指定します。
ステップ 8	**ip policy route-map** *map-tag* 例： `Device(config-if)# ip policy route-map pbr-map`	レイヤ 3 インターフェイス上で PBR をイネーブルにし、使用するルート マップを識別します。1 つのインターフェイスに設定できるルート マップは、1 つだけです。ただし、異なるシーケンス番号を持つ複数のルート マップ エントリを設定できます。これらのエントリは、最初の一致が見つかるまで、シーケンス番号順に評価されます。一致が見つからない場合、パケットは通常どおりにルーティングされます。

	コマンドまたはアクション	目的
ステップ 9	**ip route-cache policy** 例： `Device(config-if)# ip route-cache policy`	（任意）PBR の高速スイッチングをイネーブルにします。PBR の高速スイッチングをイネーブルにするには、PBR をイネーブルにする必要があります。
ステップ 10	**exit** 例： `Device(config-if)# exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 11	**ip local policy route-map** *map-tag* 例： `Device(config)# ip local policy route-map local-pbr`	（任意）ローカル PBR をイネーブルにして、スイッチから送信されるパケットに PBR を実行します。ローカル PBR は、スイッチによって生成されるパケットに適用されます。着信パケットには適用されません。
ステップ 12	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 13	**show route-map** [*map-name*] 例： `Device# show route-map`	（任意）設定を確認するため、設定されたすべてのルートマップ、または指定されたルートマップだけを表示します。
ステップ 14	**show ip policy** 例： `Device# show ip policy`	（任意）インターフェイスに付加されたポリシー ルート マップを表示します。
ステップ 15	**show ip local policy** 例： `Device# show ip local policy`	（任意）ローカル PBR がイネーブルであるかどうか、およびイネーブルである場合は使用されているルート マップを表示します。

ルーティング情報のフィルタリング

ルーティング プロトコル情報をフィルタリングする場合は、以下の作業を実行します。

（注） OSPF プロセス間でルートが再配信される場合、OSPF メトリックは保持されません。

受動インターフェイスの設定

ローカル ネットワーク上の他のルータが動的にルートを取得しないようにするには、**passive-interface** ルータ コンフィギュレーション コマンドを使用し、ルーティング アップデート メッセージがルータ インターフェイスから送信されないようにします。OSPF プロトコルでこのコマンドを使用すると、パッシブに指定したインターフェイス アドレスが OSPF ドメインのスタブ ネットワークとして表示されます。OSPF ルーティング情報は、指定されたルータ インターフェイスから送受信されません。

多数のインターフェイスが存在するネットワークで、インターフェイスを手動でパッシブに設定する作業を回避するには、**passive-interface default** ルータ コンフィギュレーション コマンドを使用し、すべてのインターフェイスをデフォルトでパッシブになるように設定します。このあとで、隣接関係が必要なインターフェイスを手動で設定します。

パッシブとしてイネーブルにしたインターフェイスを確認するには、**show ip ospf interface** などのネットワーク モニタリング用特権 EXEC コマンドを使用します。アクティブとしてイネーブルにしたインターフェイスを確認するには、**show ip interface** 特権 EXEC コマンドを使用します。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router** {**rip** | **ospf** | **eigrp**} 例： `Device(config)# router ospf`	ルータ コンフィギュレーション モードを開始します。
ステップ 3	**passive-interface** *interface-id* 例： `Device(config-router)# passive-interface gigabitethernet 1/0/1`	指定されたレイヤ 3 インターフェイス経由のルーティング アップデートの送信を抑制します。
ステップ 4	**passive-interface default** 例： `Device(config-router)# passive-interface default`	（任意）すべてのインターフェイスを、デフォルトでパッシブとなるように設定します。
ステップ 5	**no passive-interface** *interface type* 例： `Device(config-router)# no`	（任意）隣接関係を送信する必要があるインターフェイスだけをアクティブにします。

	コマンドまたはアクション	目的
	`passive-interface gigabitethernet1/0/3 gigabitethernet 1/0/5`	
ステップ 6	**network** *network-address* 例： `Device(config-router)# network 10.1.1.1`	（任意）ルーティング プロセス用のネットワーク リストを指定します。 *network-address* は IP アドレスです。
ステップ 7	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

ルーティング アップデートのアドバタイズおよび処理の制御

アクセス コントロール リストと **distribute-list** ルータ コンフィギュレーション コマンドを組み合わせて使用すると、ルーティング アップデート中にルートのアドバタイズを抑制し、他のルータが 1 つまたは複数のルートを取得しないようにできます。この機能を OSPF で使用した場合は外部ルートにだけ適用されるため、インターフェイス名を指定できません。

distribute-list ルータ コンフィギュレーション コマンドを使用し、着信したアップデートのリストのうち特定のルートを処理しないようにすることもできます。（OSPF にこの機能は適用されません）。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router** {**rip** | **eigrp**} 例： `Device(config)# router eigrp 10`	ルータ コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**distribute-list** {*access-list-number* | *access-list-name*} **out** [*interface-name* | *routing process* | *autonomous-system-number*] 例： `Device(config-router)# distribute 120 out gigabitethernet 1/0/7`	アクセスリスト内のアクションに応じて、ルーティング アップデート内のルートのアドバタイズを許可または拒否します。
ステップ 4	**distribute-list** {*access-list-number* | *access-list-name*} **in** [*type-number*] 例： `Device(config-router)# distribute-list 125 in`	アップデートにリストされたルートの処理を抑制します。
ステップ 5	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

ルーティング情報の送信元のフィルタリング

一部のルーティング情報が他の情報よりも正確な場合があるため、フィルタリングを使用して、さまざまな送信元から送られる情報にプライオリティを設定できます。「アドミニストレーティブ ディスタンス」は、ルータやルータのグループなど、ルーティング情報の送信元の信頼性を示す数値です。大規模ネットワークでは、他のルーティング プロトコルよりも信頼できるルーティングプロトコルが存在する場合があります。アドミニストレーティブディスタンスの値を指定すると、ルータはルーティング情報の送信元をインテリジェントに区別できるようになります。常にルーティング プロトコルのアドミニストレーティブ ディスタンスが最短（値が最小）であるルートが選択されます。

各ネットワークには独自の要件があるため、アドミニストレーティブ ディスタンスを割り当てる一般的な注意事項はありません。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**router** {**rip** | **ospf** | **eigrp**} 例： `Device(config)# router eigrp 10`	ルータ コンフィギュレーション モードを開始します。
ステップ 3	**distance** *weight* {*ip-address* {*ip-address mask*}} [*ip access list*] 例： `Device(config-router)# distance 50 10.1.5.1`	アドミニストレーティブ ディスタンスを定義します。 *weight*：アドミニストレーティブ ディスタンスは 10 ～ 255 の整数です。単独で使用した場合、*weight* はデフォルトのアドミニストレーティブ ディスタンスを指定します。ルーティング情報の送信元に他の指定がない場合に使用されます。アドミニストレーティブ ディスタンスが 255 のルートはルーティング テーブルに格納されません。 （任意）*ip access list*：着信ルーティング アップデートに適用される IP 標準または IP 拡張アクセス リストです。
ステップ 4	**end** 例： `Device(config-router)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show ip protocols** 例： `Device# show ip protocols`	指定されたルーティング プロセス用のデフォルトのアドミニストレーティブ ディスタンスを表示します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

認証キーの管理

キー管理を使用すると、ルーティング プロトコルで使用される認証キーを制御できます。一部のプロトコルでは、キー管理を使用できません。認証キーは EIGRP および RIP バージョン 2 で使用できます。

前提条件

認証キーを管理する前に、認証をイネーブルにする必要があります。プロトコルに対して認証をイネーブルにする方法については、該当するプロトコルについての説明を参照してください。認証キーを管理するには、キー チェーンを定義してそのキー チェーンに属するキーを識別し、各キーの有効期間を指定します。各キーには、ローカルにストアされる独自のキー ID（**key** *number* キー チェーン コンフィギュレーション コマンドで指定）があります。キー ID、およびメッセージに関連付けられたインターフェイスの組み合わせにより、使用中の認証アルゴリズムおよび Message Digest 5（MD5）認証キーが一意に識別されます。

認証キーの設定方法

有効期間が指定された複数のキーを設定できます。存在する有効なキーの数にかかわらず、送信される認証パケットは 1 つだけです。最小の番号から順にキー番号が調べられ、最初に見つかった有効なキーが使用されます。キー変更中は、有効期間が重なっても問題ありません。これらの有効期間は、ルータに通知する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure terminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**key chain** *name-of-chain* 例： `Device(config)# key chain key10`	キー チェーンを識別し、キーチェーン コンフィギュレーション モードを開始します。
ステップ 3	**key** *number* 例： `Device(config-keychain)# key 2000`	キー番号を識別します。指定できる範囲は 0 ～ 2147483647 です。

	コマンドまたはアクション	目的
ステップ 4	**key-string** *text* 例： `Device(config-keychain)# Room 20, 10th floor`	キー ストリングを確認します。ストリングには 1 ～ 80 文字の大文字および小文字の英数字を指定できますが、最初の文字に数字を指定できません。
ステップ 5	**accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `Device(config-keychain)# accept-lifetime 12:30:00 Jan 25 1009 infinite`	（任意）キーを受信できる期間を指定します。 *start-time* および *end-time* 構文には、*hh:mm:ss Month date year* または *hh:mm:ss date Month year* のいずれかを使用できます。デフォルトは、デフォルトの *start-time* 以降、無制限です。指定できる最初の日付は 1993 年 1 月 1 日です。デフォルトの *end-time* および **duration** は **infinite** です。
ステップ 6	**send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*} 例： `Device(config-keychain)# accept-lifetime 23:30:00 Jan 25 1019 infinite`	（任意）キーを送信できる期間を指定します。 *start-time* および *end-time* 構文には、*hh:mm:ss Month date year* または *hh:mm:ss date Month year* のいずれかを使用できます。デフォルトは、デフォルトの *start-time* 以降、無制限です。指定できる最初の日付は 1993 年 1 月 1 日です。デフォルトの *end-time* および **duration** は **infinite** です。
ステップ 7	**end** 例： `Device(config-keychain)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show key chain** 例： `Device# show key chain`	認証キーの情報を表示します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IP ネットワークのモニタリングおよびメンテナンス

特定のキャッシュ、テーブル、またはデータベースのすべての内容を削除できます。特定の統計情報を表示することもできます。

表 *17* : *IP* ルートの削除またはルート ステータスの表示を行うコマンド

コマンド	目的
clear ip route {*network* [*mask* | *****]}	1 つまたは複数のルートを IP ルーティング テーブルから消去します。
show ip protocols	アクティブなルーティング プロトコル プロセスのパラメータおよびステートを表示します。
show ip route [*address* [*mask*] [**longer-prefixes**]] | [*protocol* [*process-id*]]	ルーティング テーブルの現在の状態を表示します。
show ip route summary	サマリー形式でルーティング テーブルの現在のステータスを表示します。
show ip route supernets-only	スーパーネットを表示します。
show ip cache	IP トラフィックのスイッチングに使用されるルーティング テーブルを表示します。
show route-map [*map-name*]	設定されたすべてのルート マップ、または指定した 1 つのルート マップだけを表示します。

統合プラットフォーム コンフィギュレーション ガイド、Cisco IOS XE 3.3SE（Catalyst 3850 スイッチ）