ネットワーク スタック IPv4 および IPv6 の実装

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

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

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

ネットワーク スタック IPv4 および IPv6 の実装の機能履歴

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

- ネットワーク スタック IPv4 および IPv6 の実装の前提条件，2 ページ
- ネットワーク スタック IPv4 および IPv6 の実装の制約事項，2 ページ
- ネットワーク スタック IPv4 および IPv6 の実装について，2 ページ
- ネットワーク スタック IPv4 および IPv6 の実装方法，24 ページ
- 総称ルーティング カプセル化，40 ページ
- ネットワーク スタック IPv4 および IPv6 の実装の設定例，41 ページ

- VRF big モードの設定，43 ページ
- その他の参考資料，45 ページ

ネットワーク スタック IPv4 および IPv6 の実装の前提条件

適切なタスク ID を含むタスク グループに関連付けられているユーザ グループに属している必要があります。 このコマンド リファレンスには、各コマンドに必要なタスク ID が含まれます。 ユーザ グループの割り当てが原因でコマンドを使用できないと考えられる場合、AAA 管理者に連絡してください。

ネットワーク スタック IPv4 および IPv6 の実装の制約事項

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

ネットワーク スタック IPv4 および IPv6 の実装について

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

ネットワーク スタック IPv4 および IPv6 の例外

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

- Cisco IOS XR ソフトウェアでは、**clear ipv6 neighbors** コマンドと **show ipv6 neighbors** コマンドに **location** *node-id* キーワードが含まれています。 場所を指定した場合、指定した場所の隣接エントリのみが表示されます。
- **ipv6 nd scavenge-timeout** コマンドは、stale 状態の隣接エントリの有効期間を設定します。 隣接エントリの廃棄タイマーの有効期間が切れると、そのエントリはクリアされます。
- Cisco IOS XR ソフトウェアでは、**show ipv4 interface** コマンドと **show ipv6 interface** コマンドに **location** *node-id* キーワードが含まれています。 場所を指定した場合、指定した場所のインターフェイス エントリのみが表示されます。
- Cisco IOS XR ソフトウェア では、設定するときに、競合する IP アドレス エントリを許可します。 アクティブな 2 つのインターフェイスの間に IP アドレス競合が存在する場合、Cisco

IOS XR ソフトウェアは、設定されている競合ポリシーに従って、インターフェイスを停止します（デフォルト ポリシーでは、より高いインターフェイス インスタンスを停止します）。 たとえば、GigabitEthernet 0/1/0/1 が GigabitEthernet 0/2/0/1 と競合した場合、GigabitEthernet 0/2/0/1 上の IPv4 プロトコルが停止され、GigabitEthernet 0/1/0/1 上の IPv4 はアクティブなままになります。

IPv4 および IPv6 機能

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

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

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

Cisco IOS XR ソフトウェアの IPv6

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

拡大された IPv6 アドレス空間

グローバルに一意な IP アドレスの需要は今後増加すると予想され、その需要を満たす必要があることが、IPv6 の主な目的です。モバイルインターネット対応デバイス（携帯情報端末（PDA）、電話、車両など）、Home Area Network（HAN）、ワイヤレス データ サービスなどのアプリケーションによって、グローバルに一意な IP アドレスの需要が増大しています。 IPv6 は、ネットワー

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

IPv6 アドレス形式

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

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

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

IPv6 アドレスには、通常、連続するゼロの 16 進フィールドが含まれます。 IPv6 アドレスを扱いやすくするために、2 つのコロン（::）を使用して、IPv6 アドレスの先頭、中間、最後の部分の連続したゼロの 16 進フィールドを圧縮できます。（これらのコロンは、連続したゼロの 16 進フィールドを表します）。表 1：圧縮された IPv6 アドレス形式, （4 ページ）に、圧縮された IPv6 アドレス形式を示します。

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

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

IPv6 アドレスの 16 進文字は大文字と小文字が区別されません。

表 *1*：圧縮された *IPv6* アドレス形式

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

ノードは、表 1：圧縮された IPv6 アドレス形式, （4 ページ）に示されているループバック アドレスを使用して、IPv6 パケットを自身に送信できます。 IPv6 のループバック アドレスは、IPv4 のループバック アドレス（127.0.0.1）と同じように機能します。

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

表 1：圧縮された IPv6 アドレス形式, （4 ページ）に示されている未指定アドレスは、IPv6 アドレスがないことを示します。 たとえば、IPv6 ネットワーク上で新しく初期化されたノードは、IPv6アドレスを受信するまで、パケットで未指定アドレスを送信元アドレスとして使用できます。

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

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

IPv6 アドレス タイプ：ユニキャスト

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

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

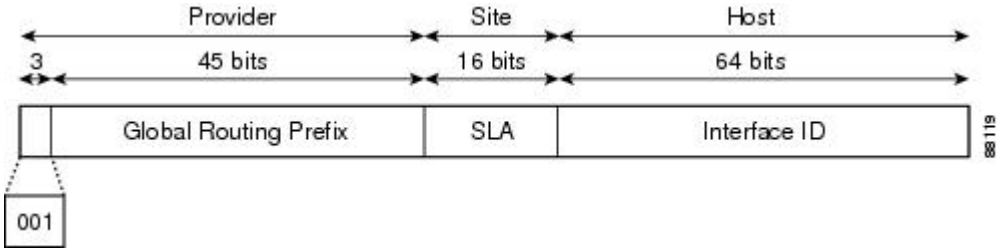
集約可能グローバル アドレス

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

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

集約可能グローバル IPv6 アドレスは、グローバル ルーティング プレフィックス、サブネット ID、およびインターフェイス ID により定義されます。 バイナリ 000 から開始するアドレスを除き、すべてのグローバル ユニキャスト アドレスには 64 ビットのインターフェイス ID があります。現在のグローバルユニキャストアドレスの割り当てには、バイナリ値 001（2000::/3）から始まるアドレスの範囲が使用されます。 図 1：集約可能グローバル アドレス形式, （6 ページ）に、集約可能グローバル アドレスの構造を示します。

図 *1*：集約可能グローバル アドレス形式



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

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

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

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

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

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

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

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

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

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

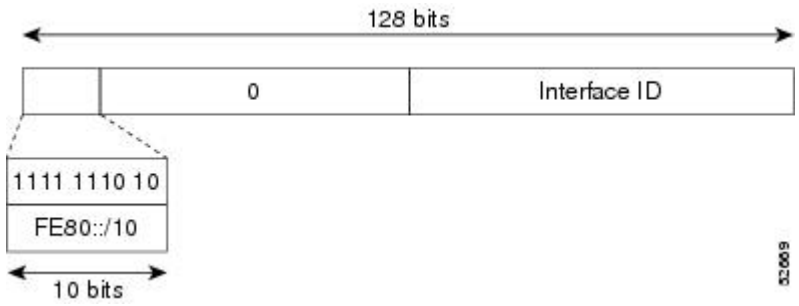
1. ルータに MAC アドレスが（ルータの MAC アドレス プールから）照会されます。
2. 使用できる MAC アドレスがない場合は、ルート プロセッサ（RP）またはラインカード（LC）のシリアル番号を使用して、リンクローカル アドレスを形成します。

リンクローカル アドレス

リンクローカル アドレスは、リンクローカル プレフィックス FE80::/10（1111 1110 10）と変更された EUI-64 形式のインターフェイス識別子を使用するすべてのインターフェイスを自動的に設定できる IPv6 ユニキャスト アドレスです。リンクローカル アドレスは、ネイバー探索プロトコルとステートレス自動設定プロセスで使用されます。ローカル リンク上のノードは、リンクローカルアドレスを使用して通信できます。ノードの通信にサイトローカルアドレスまたはグローバルに一意のアドレスは不要です。図 2：リンクローカル アドレス形式，（8 ページ）に、リンクローカル アドレスの構造を示します。

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

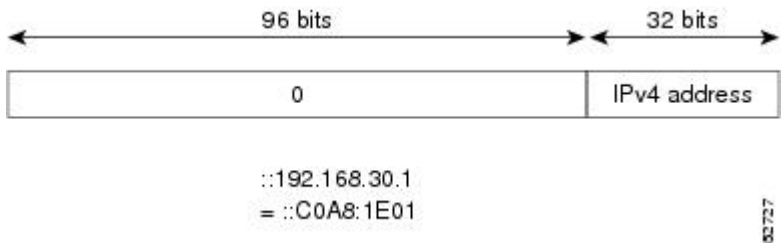
図 *2*：リンクローカル アドレス形式



IPv4 互換 IPv6 アドレス

IPv4 互換 IPv6 アドレスは、アドレスの上位 96 ビットがゼロであり、アドレスの下位 32 ビットが IPv4 アドレスである IPv6 ユニキャスト アドレスです。 IPv4 互換 IPv6 アドレスの形式は、0:0:0:0:0:0:A.B.C.D または ::A.B.C.D です。 IPv4 互換 IPv6 アドレスの 128 ビット全体がノードの IPv6 アドレスとして使用され、下位 32 ビットに埋め込まれた IPv4 アドレスがノードの IPv4 アドレスとして使用されます。 IPv4 互換 IPv6 アドレスは、IPv4 と IPv6 の両方のプロトコル スタックをサポートするノードに割り当てられ、自動トンネルで使用されます。 図 3：IPv4 互換 IPv6 アドレス形式, （8 ページ）に、IPv4 互換 IPv6 アドレスの構造と、許容されるいくつかのアドレス形式を示します。

図 *3*：*IPv4* 互換 *IPv6* アドレス形式

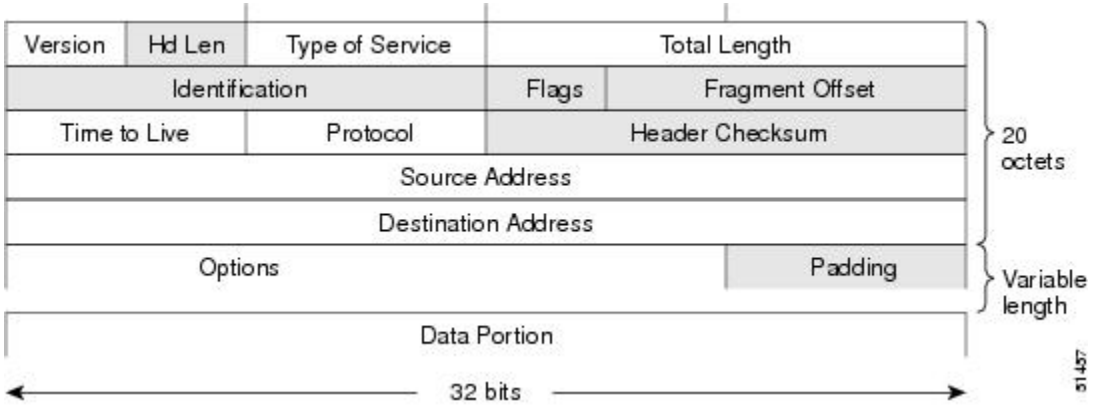


簡易 IPv6 パケット ヘッダー

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

ダーのグレーの部分のフィールドは、IPv6 パケット ヘッダーに含まれません（図 4：IPv4 パケット ヘッダー形式, （9 ページ）を参照）。

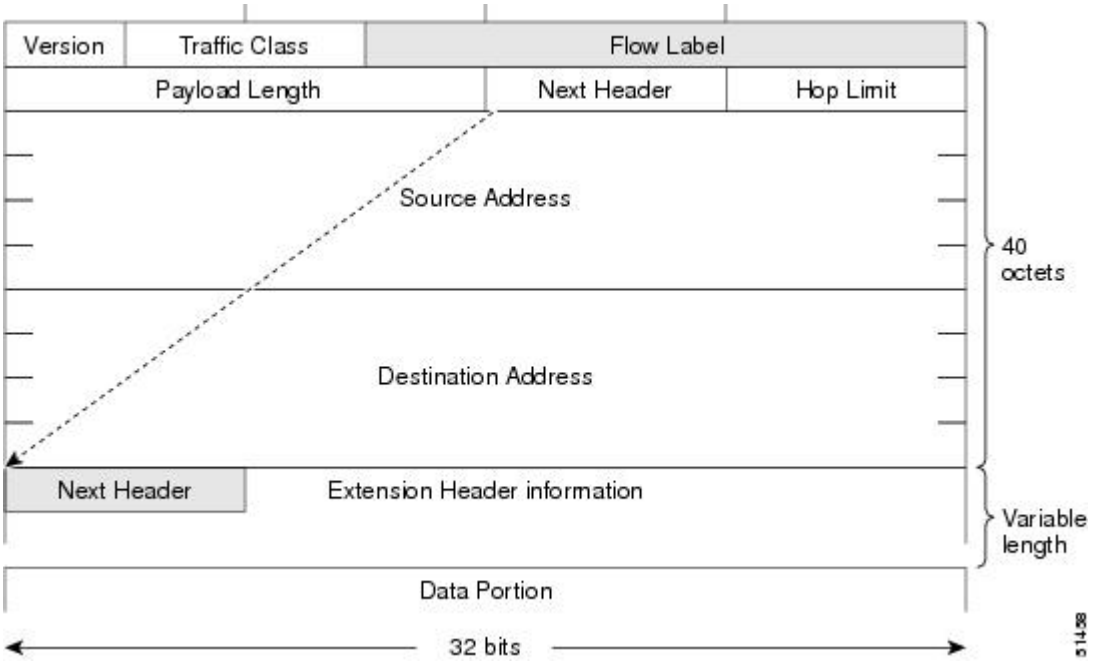
図 *4*：*IPv4* パケット ヘッダー形式



基本 IPv6 パケット ヘッダーには、合計サイズが 40 オクテット（320 ビット）の 8 つのフィールドがあります（図 5：IPv6 パケット ヘッダー形式, （10 ページ）を参照）。IPv6 では、フラグメンテーションはルータによって処理されず、チェックサムはネットワーク層で使用されないため、IPv6 ヘッダーからフィールドが除去されました。 代わりに、IPv6 のフラグメンテーションはパケットの送信元によって処理され、チェックサムはデータ リンク層とトランスポート層で使用されます （IPv4 では、ユーザ データグラム プロトコル（UDP）トランスポート層でオプションのチェックサムが使用されます。 IPv6 では、内部パケットの整合性をチェックするために UDP

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

図 *5*：*IPv6* パケット ヘッダー形式



次の表に、基本 IPv6 パケット ヘッダーのフィールドをリストします。

表 *2*：基本 *IPv6* パケット ヘッダー フィールド

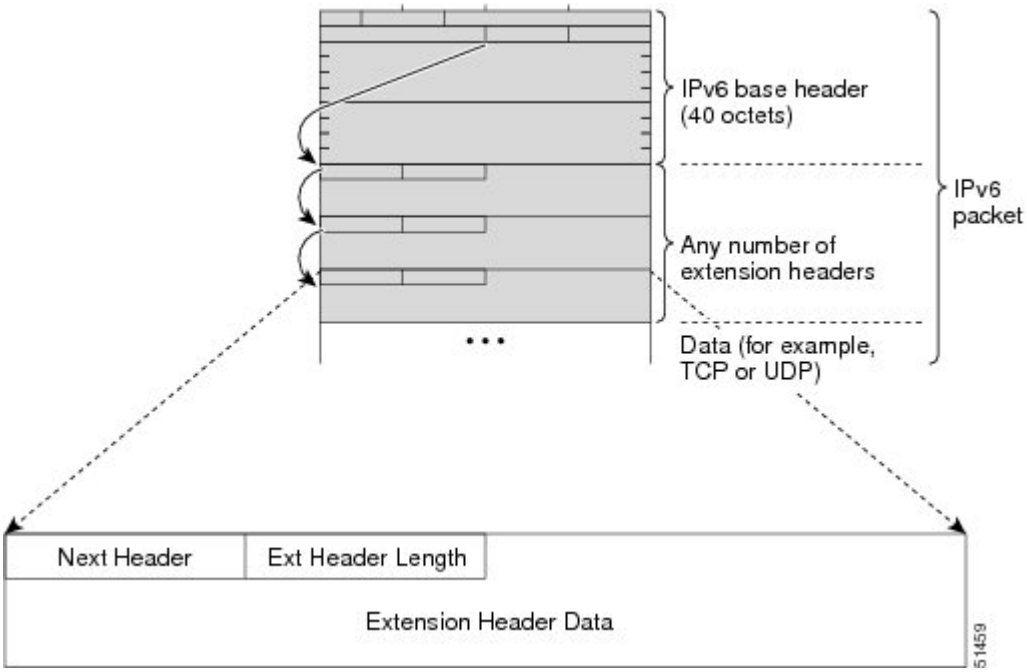
フィールド	説明
バージョン	IPv4 パケット ヘッダーのバージョン フィールドと同様ですが、IPv4 を意味する数字 4 の代わりに IPv6 を意味する数字 6 が示されます。
トラフィック クラス	IPv4 パケット ヘッダーのタイプ オブ サービス フィールドと同様です。 トラフィック クラス フィールドは、差別化されたサービスで使用されるトラフィック クラスのタグをパケットに付けます。
フロー ラベル	IPv6 パケット ヘッダーの新しいフィールドです。 フロー ラベル フィールドは、ネットワーク層でパケットを差別化する特定のフローのタグをパケットに付けます。

フィールド	説明
ペイロード長	IPv4 パケット ヘッダーの合計長フィールドと同様です。ペイロード長フィールドは、パケットのデータ部分の合計長を示します。
次ヘッダー	IPv4 パケット ヘッダーのプロトコル フィールドと同様です。 次ヘッダー フィールドの値により、基本 IPv6 ヘッダーに続く情報のタイプが決まります。 基本 IPv6 ヘッダーに続く情報のタイプは、図 6：IPv6 拡張ヘッダー形式, （12 ページ）に示すように、TCP や UDP パケットなどのトランスポートレイヤ パケット、または拡張ヘッダーです。
ホップ リミット	IPv4 パケット ヘッダーの存続可能時間フィールドと同様です。 ホップ リミット フィールドの値は、IPv6 パケットが無効と見なされる前に通過できるルータの最大数です。 各ルータを通過するたびに、この値が 1 つずつ減少します。 IPv6 ヘッダーにはチェックサムがないため、ルータは値を減らすたびにチェックサムを再計算する必要がなく、処理リソースが節約されます。
送信元アドレス	IPv4 パケット ヘッダーの送信元アドレス フィールドと同様ですが、IPv4 の 32 ビット送信元アドレスの代わりに、IPv6 では 128 ビットの送信元アドレスが含まれます。
宛先アドレス	IPv4 パケット ヘッダーの宛先アドレス フィールドと同様ですが、IPv4 の 32 ビット宛先アドレスの代わりに、IPv6 では 128 ビットの宛先アドレスが含まれます。

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

ヘッダー フィールドがあります。 図 6：IPv6 拡張ヘッダー形式, （12 ページ）に、IPv6 拡張ヘッダー形式を示します。

図 *6*：*IPv6* 拡張ヘッダー形式



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

表 *3*：*IPv6* 拡張ヘッダー タイプ

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

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

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

IPv6 のパス MTU ディスカバリ

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

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

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

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

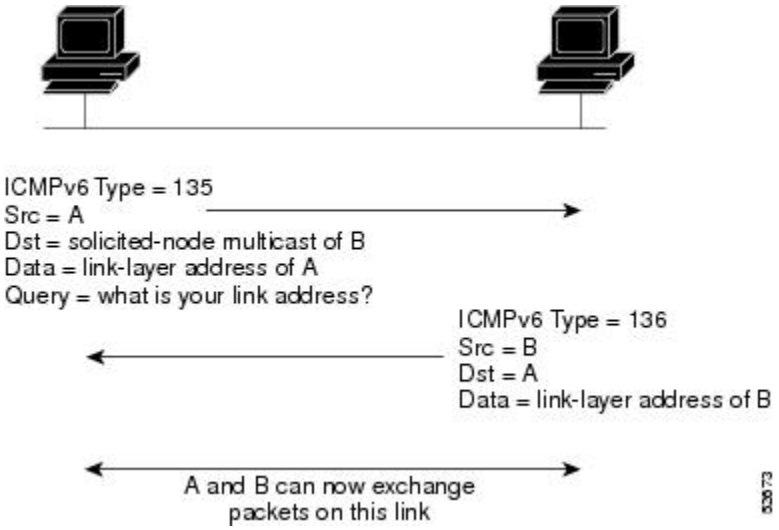
IPv6 ネイバー探索

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

IPv6 ネイバー送信要求メッセージ

ICMP パケット ヘッダーのタイプ フィールドの値 135 は、ネイバー送信要求メッセージを示します。ネイバー送信要求メッセージは、ノードが同じローカル リンク上の別のノードのリンク層アドレスを決定するときに、ローカル リンク上で送信されます（図 7：IPv6 ネイバー探索 - ネイバー送信要求メッセージ, （15 ページ）を参照）。ノードで別のノードのリンク層アドレスを特定する必要がある場合、ネイバー送信要求メッセージの送信元アドレスは、ネイバー送信要求メッセージを送信するノードの IPv6 アドレスになります。ネイバー送信要求メッセージ内の宛先アドレスは、宛先ノードの IPv6 アドレスに対応する送信要求ノード マルチキャスト アドレスです。ネイバー送信要求メッセージには、送信元ノードのリンク層アドレスも含まれます。

図 7：***IPv6* ネイバー探索 - ネイバー送信要求メッセージ**



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

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

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

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

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

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

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

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

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

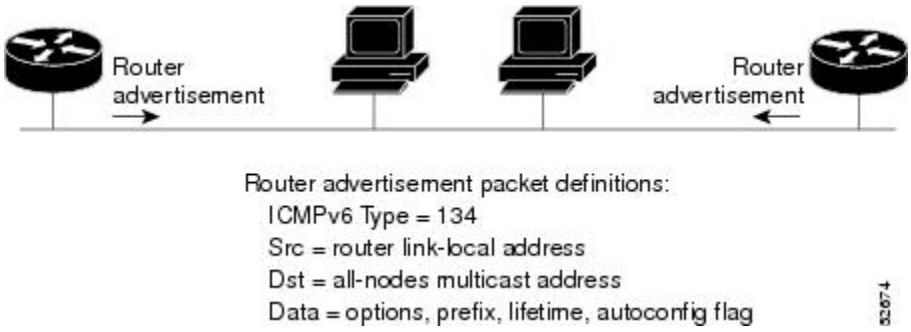
す。 ネイバー送信要求メッセージの返信としてネイバー アドバタイズメント メッセージが受信されず、同じ一時アドレスの検証を試行している他のノードからのネイバー送信要求メッセージも受信されない場合、最初のネイバー送信要求メッセージを送信したノードは、一時的なリンクローカル アドレスを一意であると見なし、そのアドレスをインターフェイスに割り当てます。

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

IPv6 ルータ アドバタイズメント メッセージ

ルータ アドバタイズメント（RA）メッセージは、ICMP パケット ヘッダーのタイプ フィールドが値 134 であり、IPv6 ルータの設定済みの各インターフェイスへ定期的に送信されます。 ルータ アドバタイズメント メッセージは全ノード マルチキャスト アドレスに送信されます（図 8：IPv6 ネイバー探索 - ルータ アドバタイズメント メッセージ, （17 ページ）を参照）。

図 *8*：*IPv6* ネイバー探索 - ルータ アドバタイズメント メッセージ



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

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

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

次のルータ アドバタイズメント メッセージ パラメータを設定できます。

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

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

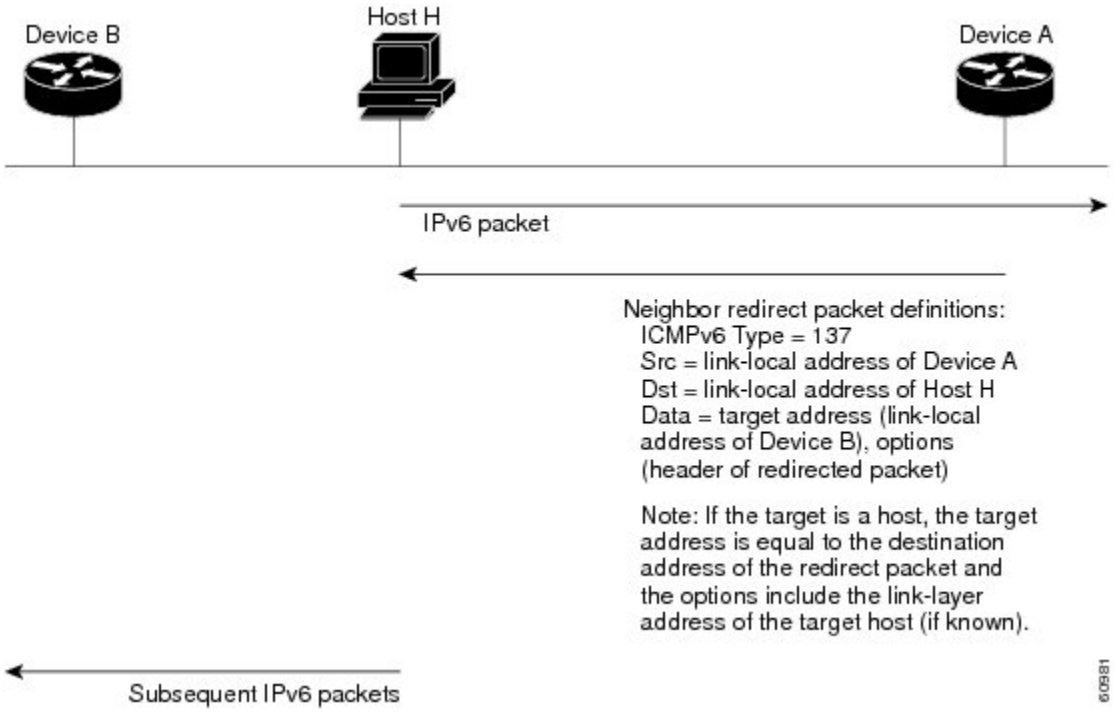
（注） ステートレス自動設定が正しく機能するには、ルータ アドバタイズメント メッセージでアドバタイズされたプレフィックス長が常に 64 ビットである必要があります。

IPv6 ネイバー リダイレクト メッセージ

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

り適切なファーストホップ ノードをホストに通知します（図 9：IPv6 ネイバー探索 - ネイバー リダイレクト メッセージ，（19 ページ）を参照）。

図 *9*：*IPv6* ネイバー探索 - ネイバー リダイレクト メッセージ



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

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

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

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

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

IPv6 の ICMP

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

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

Address Repository Manager

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

アドレス競合解決

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

競合データベース

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

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

アドレス 1：10.1.1.1/16

アドレス 2：10.2.1.1/16

アドレス 3：10.3.1.1/16

アドレス 4：10.4.1.1/8

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

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

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

複数の IP アドレス

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

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

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

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

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

競合セットの再帰的解決

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

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

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

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

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

接続ルートに対する Route-Tag のサポート

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

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

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. 次のいずれかを実行します。
   - **ipv4 address** *ipv4-address mask* [**secondary**]
4. route-tag *[route-tag value]*
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface POS 0/1/0/1`	インターフェイス コンフィギュレーション モードを開始します。

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

ステップ 5	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

ネットワーク スタック IPv4 および IPv6 の実装方法

ここでは、次の手順について説明します。

ネットワーク インターフェイスへの IPv4 アドレスの割り当て

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

IPv4 アドレス

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

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

（注） シスコでは、ネットワーク フィールドに対して左寄せの連続ビットを使用するネットワーク マスクのみをサポートしています。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ipv4 address** *ipv4-address mask* [**secondary**]
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
5. show ipv4 interface

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/1`	インターフェイス コンフィギュレーション モードを開始します。

ステップ 3	**ipv4 address** *ipv4-address mask* [**secondary**]
	
	例：
	`RP/0/RSP0/CPU0:router(config-if)# ipv4 address 192.168.1.27 255.0.0.0`
	`RP/0/RSP0/CPU0:router(config-if)# ipv4 address 192.168.1.27/8`
		インターフェイスに対するプライマリ IPv4 アドレスまたはセカンダリ IPv4 アドレスを指定します。
		• 4 分割ドット付き 10 進表記のアドレスでネットワーク マスクを指定します。 たとえば、255.0.0.0 は、1 に等しい各ビットが、ネットワーク アドレスに属した対応するアドレス ビットを意味することを示します。
		• ネットワーク マスクは、スラッシュ（/）および数字、つまり、プレフィックス長として示される場合もあります。プレフィックス長は、アドレスの高次の連続ビットのうち、何個がプレフィックス（アドレスのネットワーク部分）を構成しているかを指定する 10 進数値です。 スラッシュは 10 進数値の前に置かれ、IP アドレスとスラッシュの間にスペースは入りません。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。

	コマンドまたはアクション	目的
		◦ **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。 • 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。
ステップ 5	show ipv4 interface 例： `RP/0/RSP0/CPU0:router# show ipv4 interface`	（任意）IPv4 用に設定されたインターフェイスの使用可能性ステータスを表示します。

IPv4 仮想アドレス

IPv4 仮想アドレスを設定することにより、いずれのルートプロセッサ（RP）がアクティブであるかを事前に把握していなくても、管理ネットワークでの単一の仮想アドレスからルータにアクセスすることができます。 IPv4 仮想アドレスは、RP フェールオーバー状況間で維持されます。 このようにするには、仮想 IPv4 アドレスが、両方の RP の管理イーサネット インターフェイスで共通 IPv4 サブネットを共有する必要があります。

vrf キーワードは、VRF 単位の仮想アドレスをサポートします。

use-as-src-addr キーワードを使用すると、管理アプリケーションのために、ループバック インターフェイスを送信元インターフェイス（つまり、更新送信元）として設定する必要がなくなります。更新送信元が設定されていない場合、トランスポート プロセス（TCP、UDP、raw_ip）は、管理アプリケーションを使用して適切な送信元アドレスを選択できます。 トランスポート プロセスは、FIB を参照して、適切な送信元アドレスを選択します。 管理イーサネットの IP アドレスが送信元アドレスとして選択されており、**use-as-src-addr** キーワードが設定されている場合、トランスポートでは、管理イーサネットの IP アドレスを関連する仮想 IP アドレスに置き換えます。 この機能は、RP スイッチオーバー全体で機能します。 **use-as-src-addr** が設定されていない場合、トランスポートで選択された送信元アドレスはフェールオーバー後に変更される可能性があり、NMS ソフトウェアがこの状況を管理できなくなるおそれがあります。

（注） tacacs source-interface、snmp-server trap-source、ntp source、logging source-interface などのプロトコル コンフィギュレーションでは、送信元として仮想管理 IP アドレスをデフォルトでは使用しません。 **ipv4 virtual address use-as-src-addr** コマンドを使用して、プロトコルが仮想 IPv4 アドレスを送信元アドレスとして使用するようにします。また、指定した、または目的の IPv4 アドレスを使用してループバック アドレスを設定し、それを TACACS+ などのプロトコルの送信元として **tacacs source-interface** コマンドにより設定することもできます。

IPv6 アドレッシングの設定

このタスクでは、IPv6 アドレスを個々のルータ インターフェイスに割り当て、ルータ上で IPv6 トラフィックのグローバルな転送を可能にします。 デフォルトでは、IPv6 アドレスは設定されていません。

（注） **ipv6 address** コマンドの *ipv6-prefix* 引数は、RFC 2373 に記載された形式にする必要があり、16 ビット値をコロンで区切った 16 進でアドレスを指定します。

ipv6 address コマンドの **/prefix-length** 引数は 10 進数の値で、プレフィックスを構成しているアドレスの連続する上位ビット数（アドレスのネットワーク部）を指定します。10 進値の前にはスラッシュが必要です。

ipv6 address link-local コマンドの *ipv6-address* 引数は、RFC 2373 に記載された形式にする必要があり、16 ビット値をコロンで区切った 16 進でアドレスを指定します。

ネットワーク インターフェイスへの複数の IP アドレスの割り当て

このタスクでは、複数の IP アドレスをネットワーク インターフェイスに割り当てます。

セカンダリ IPv4 アドレス

Cisco IOS XR ソフトウェアは、インターフェイスごとに複数の IP アドレスをサポートしています。 セカンダリ アドレスは無制限に指定できます。 セカンダリ IP アドレスは、さまざまな状況で使用できます。 次に、一般的な使用状況を示します。

- 特定のネットワーク セグメントに十分なホスト アドレスがない場合があります。たとえば、サブネット化により、論理サブネットあたり最大 254 のホストを使用できますが、1 つの物理サブネットでは、300 のホスト アドレスが必要になるとします。 ルータまたはアクセス サーバでセカンダリ IP アドレスを使用すると、2 つの論理サブネットで 1 つの物理サブネットを使用できます。
- 多くの旧式ネットワークは、レベル 2 ブリッジを使用して構築され、サブネット化されませんでした。セカンダリ アドレスは、慎重に使用することで、サブネット化されたルータベー

ス ネットワークへの移行に役立ちます。 旧式のブリッジ セグメントのルータで、そのセグメントに複数のサブネットがあることを簡単に認識されるようにできます。

- 1 つのネットワークの 2 つのサブネットは、別の方法で、別のネットワークにより分離できる場合があります。別のネットワークによって物理的に分離された複数のサブネットから、セカンダリ アドレスを使用して、1 つのネットワークを作成できます。 このような場合、最初のネットワークは、2 番めのネットワークの上に拡張されます。つまり、上の階層となります。 サブネットは、同時に複数のアクティブなインターフェイス上に表示できないことに注意してください。

（注） ネットワーク セグメント上の任意のルータがセカンダリ IPv4 アドレスを使用した場合、同一のセグメント上にある他のルータもすべて、同一のネットワークまたはサブネットからセカンダリ アドレスを使用する必要があります。

注意 ネットワーク セグメント上のセカンダリ アドレスの使用に矛盾があると、ただちにルーティング ループが引き起こされる可能性があります。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ipv4 address** *ipv4-address mask* [**secondary**]
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/3`	インターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	**ipv4 address** *ipv4-address mask* [**secondary**] 例： `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 192.168.1.27 255.255.255.0 secondary`	設定されているアドレスが、セカンダリ IPv4 アドレスであることを指定します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

IPv4 および IPv6 プロトコル スタックの設定

このタスクでは、IPv4 と IPv6 の両方のプロトコル スタックをサポートするようにシスコのネットワーク デバイスのインターフェイスを設定します。

シスコのネットワーク デバイスのインターフェイスが IPv4 アドレスと IPv6 アドレスの両方で設定されている場合、インターフェイスは IPv4 トラフィックと IPv6 トラフィックの両方を転送します。インターフェイスは、IPv4 ネットワークと IPv6 ネットワークの両方でデータを送受信できます。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ipv4 address** *ip-address mask* [**secondary**]
4. **ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**]
5. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/1`	インターフェイスのタイプおよび番号を指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ipv4 address** *ip-address mask* [**secondary**] 例： `RP/0/RSP0/CPU0:router(config-if)# ipv4 address 192.168.99.1 255.255.255.0`	インターフェイスに対するプライマリ IPv4 アドレスまたはセカンダリ IPv4 アドレスを指定します。
ステップ 4	**ipv6 address** *ipv6-prefix*/*prefix-length* [**eui-64**] 例： `RP/0/RSP0/CPU0:router(config-if)# ipv6 address 2001:0DB8:c18:1::3/64`	インターフェイスに割り当てられた IPv6 アドレスを指定し、インターフェイスでの IPv6 処理をイネーブルにします。 • スラッシュ記号（/）は、*prefix-length* の前に置かれ、*ipv6-prefix* とスラッシュ記号の間にスペースは入りません。
ステップ 5	次のいずれかのコマンドを使用します。 • **end** • **commit**	設定変更を保存します。 • **end** コマンドを実行すると、変更をコミットするように要求されます。 `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`

	コマンドまたはアクション	目的

	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		◦ **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータがEXECモードに戻ります。
		◦ **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

アンナンバード インターフェイス上での IPv4 処理のイネーブル化

このタスクでは、アンナンバード インターフェイス上での IPv4 処理をイネーブルにします。

アンナンバード インターフェイス上での IPv4 処理

ここでは、明示的なIPアドレスをインターフェイスに割り当てることなく、IPv4ポイントツーポイント インターフェイスをイネーブルにするプロセスについて説明します。 アンナンバード インターフェイスがパケットを生成する場合（たとえば、ルーティング アップデートのため）は必ず、IP パケットの送信元アドレスとして指定したインターフェイスのアドレスが使用されます。また、アンナンバードインターフェイスを介してアップデートを送信するルーティングプロセスを判別する場合、指定されたインターフェイスのアドレスが使用されます。 その制限を次に示します。

- High-Level Data Link Control（HDLC）、PPP、およびフレーム リレーのカプセル化を使用するシリアル インターフェイスには、アンナンバードを設定できません。 フレーム リレー カプセル化を使用するシリアルインターフェイスにもアンナンバードを設定できますが、そのインターフェイスはポイントツーポイント サブインターフェイスである必要があります。
- インターフェイスがIPアドレスを持たないため、インターフェイスがアップ状態かどうかを判断するために **ping** EXEC コマンドは使用できません。 簡易ネットワーク管理プロトコル（SNMP）は、インターフェイス ステータスのリモートでのモニタリングに使用できます。
- IPセキュリティ オプションは、アンナンバードインターフェイス上でサポートできません。

Intermediate System-to-Intermediate System（IS-IS）をシリアル回線全体で設定する場合、シリアル インターフェイスをアンナンバードとして設定し、それにより、各インターフェイス上で IP アドレスは必須ではないことを規定している RFC 1195 に準拠することができます。

手順の概要

1. **configure**
2. **interface** *type interface-path-id*
3. **ipv4 unnumbered** *interface-type interface-instance*
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type interface-path-id* 例： `RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/1`	インターフェイス コンフィギュレーション モードを開始します。
ステップ 3	**ipv4 unnumbered** *interface-type interface-instance* 例： `RP/0/RSP0/CPU0:router(config-if)# ipv4 unnumbered loopback 5`	明示的な IPv4 アドレスをインターフェイスに割り当てることなく、ポイントツーポイント インターフェイス上での IPv4 処理をイネーブルにします。 • 指定したインターフェイスは、別のアンナンバードインターフェイスではなく、任意の IP アドレスを持つ、ルータの別のインターフェイスの名前である必要があります。 • *interface-type* および *interface-instance* 引数で指定されたインターフェイスは、イネーブルにされている必要があります（**show interfaces** コマンド出力に「up」と表示）。
ステップ 4	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ICMP レート制限の設定

このタスクでは、IPv4 または IPv6 の ICMP レート制限の設定方法について説明します。

IPv4 ICMP レート制限

IPv4 ICMP レート制限機能では、IPv4 ICMP 宛先到達不能メッセージが生成されるレートを制限します。 Cisco IOS XR ソフトウェアは、通常の宛先到達不能メッセージ用と DF 宛先到達不能メッセージ用の 2 つのタイマーを保守します。 これらは同じ時間制限およびデフォルトを共有します。 **DF** キーワードが設定されていない場合、**icmp ipv4 rate-limit unreachable** コマンドによって、DF 宛先到達不能メッセージの時間値が設定されます。 **DF** キーワードが設定されている場合、その時間値は、通常の宛先到達不能メッセージの時間値とは無関係のままになります。

IPv6 ICMP レート制限

IPv6 ICMP レート制限機能によって、IPv6 ICMP エラー メッセージがネットワークへ送信されるレートを制限するためのトークン バケット アルゴリズムが実装されます。 IPv6 ICMP レート制限の初期の実装では、エラー メッセージ間に固定の間隔が定義されていましたが、traceroute などの

一部のアプリケーションでは、間断なく送信される要求のグループへの返信が必要になる場合があります。 エラー メッセージ間の固定間隔は、traceroute などのアプリケーションで動作するのに十分な柔軟性がなく、アプリケーションが失敗する原因となることがあります。 トークン バケット方式を実装すると、複数のトークンを仮想バケットに格納できます。トークンごとに 1 つのエラー メッセージを送信できます。 バケットに格納できるトークンの最大数を指定でき、エラー メッセージが送信されるたびに 1 つのトークンがバケットから削除されます。 一連のエラー メッセージが生成された場合は、バケットが空になるまでエラー メッセージを送信できます。トークンのバケットが空になると、新しいトークンがバケットに配置されるまで、IPv6 ICMP エラー メッセージは送信されません。 トークン バケット アルゴリズムは、レート制限の平均時間間隔を増やさず、固定時間間隔方式よりも柔軟性が高くなります。

手順の概要

1. **configure**
2. 次のいずれかを実行します。
   - **icmp ipv4 rate-limit unreachable** [**DF**] *milliseconds*
   - **ipv6 icmp error-interval** *milliseconds* [*bucketsize*]
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**
4. 次のいずれかを実行します。
   - **show ipv4 traffic** [**brief**]
   - **show ipv6 traffic** [**brief**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	次のいずれかを実行します。 • **icmp ipv4 rate-limit unreachable** [**DF**] *milliseconds* • **ipv6 icmp error-interval** *milliseconds* [*bucketsize*]	IPv4 ICMP 宛先到達不能メッセージが生成されるレートを制限します。 • **DF** キーワードは、コード 4 フラグメンテーションが必要で、データ フラグメンテーション（DF）が設定されているときに、ICMP 宛先到達不能メッセージの IP ヘッダーに指定されている

	コマンドまたはアクション	目的

	例：
	`RP/0/RSP0/CPU0:router(config)# icmp ipv4 rate-limit unreachable 1000`
	または
	`RP/0/RSP0/CPU0:router(config)# ipv6 icmp error-interval 50 20`
		ように、ICMP宛先到達不能メッセージが送信されるレートを制限します。
		• *milliseconds* 引数では、ICMP 宛先到達不能メッセージを送信する間隔を指定します。
		または
		IPv6 ICMP エラー メッセージの間隔とバケット サイズを設定します。
		• *milliseconds* 引数では、トークンがバケットに追加される間隔を指定します。
		• オプションの *bucketsize* 引数では、バケットに格納されるトークンの最大数を定義します。

ステップ 3	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーションファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

ステップ 4	次のいずれかを実行します。 • **show ipv4 traffic** [**brief**] • **show ipv6 traffic** [**brief**]	（任意）ICMP到達不能情報を含む、IPv4 トラフィックに関する統計情報を表示します。 • **brief** キーワードを使用して、IPv4 および ICMPv4 のトラフィック統計情報のみを表示します。

	コマンドまたはアクション	目的

	例：
	`RP/0/RSP0/CPU0:router# show ipv4 traffic`
	または
	`RP/0/RSP0/CPU0:router# show ipv6 traffic`
		または
		（任意）IPv6 ICMP レート制限カウンタを含む、IPv6 トラフィックに関する統計情報を表示します。
		• **brief** キーワードを使用して、IPv6 および ICMPv6 のトラフィック統計情報のみを表示します。

IPARM 競合解決の設定

このタスクでは、IP Address Repository Manager（IPARM）アドレス競合解決のパラメータを設定します。

静的ポリシー解決

静的ポリシー解決の設定により、新しいアドレス設定が現在実行中のインターフェイスに影響するのを防ぎます。

手順の概要

1. **configure**
2. {**ipv4** | **ipv6**} **conflict-policy static**
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的

ステップ 2	{**ipv4** | **ipv6**} **conflict-policy static**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# ipv4 conflict-policy static`
	または
	`RP/0/RSP0/CPU0:router(config)# ipv6 conflict-policy static`
		競合ポリシーを静的に設定します。つまり、新しいインターフェイス アドレスが現在実行中のインターフェイスに影響するのを防ぎます。

ステップ 3	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。
		◦ **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

最長プレフィックス アドレス競合解決

この競合解決ポリシーでは、最も長いプレフィックス長を持つIPアドレスに最も高い優先度を付与することを試みます。

手順の概要

1. **configure**
2. **{ ipv4 | ipv6 } conflict-policy longest-prefix**
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

ステップ 2	**{ ipv4 | ipv6 } conflict-policy longest-prefix**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# ipv4 conflict-policy longest-prefix`
	または
	`RP/0/RSP0/CPU0:router(config)# ipv6 conflict-policy longest-prefix`
		競合ポリシーを最長プレフィックスに設定します。つまり、競合セット内の、現在実行中のインターフェイスの最長プレフィックスアドレスと競合しないすべてのアドレスは同様に実行することが許可されます。

ステップ 3	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。

	コマンドまたはアクション	目的
		• 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

最大 IP アドレス競合解決

この競合解決ポリシーでは、最大値を持つ IP アドレスに最も高い優先度を付与することを試みます。

手順の概要

1. **configure**
2. **{ ipv4 | ipv6 } conflict-policy highest-ip**
3. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

ステップ 2	**{ ipv4 | ipv6 } conflict-policy highest-ip**
	例：
	`RP/0/RSP0/CPU0:router(config)# ipv4 conflict-policy highest-ip`
	または
	`RP/0/RSP0/CPU0:router(config)# ipv6 conflict-policy highest-ip`
		競合ポリシーを最も高い IP 値に設定します。つまり、値が最大の IP アドレスが優先されます。

ステップ 3	次のいずれかのコマンドを使用します。	設定変更を保存します。

	コマンドまたはアクション	目的

	• **end**
	• **commit**
	
	例：
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	`RP/0/RSP0/CPU0:router(config)# commit`
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		◦ **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。
		◦ **no** と入力すると、コンフィギュレーション セッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ **cancel** と入力すると、現在のコンフィギュレーション セッションが継続します。コンフィギュレーション セッションは終了せず、設定変更もコミットされません。
		• 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーション セッションを継続するには、**commit** コマンドを使用します。

総称ルーティング カプセル化

総称ルーティング カプセル化（GRE）トンネリング プロトコルでは、カプセル化によって、1 つのプロトコルから別のプロトコルにパケットを転送する、簡易で一般的なアプローチを提供します。 転送する必要のあるパケットは、まず GRE ヘッダーでカプセル化され、さらに IPv4 や IPv6 などの別のプロトコルでカプセル化されてから、宛先に転送されます。

一般的な GRE カプセル化パケットには次のものが含まれます。

- 配信ヘッダー
- GRE ヘッダー
- ペイロード パケット

カプセル化され、宛先に送信する必要があるパケットがシステムに存在します。 これが、ペイロード パケットです。 ペイロードは、まず GRE パケットにカプセル化されます。 この GRE パケットは、次に別のプロトコルでカプセル化されてから、転送されます。 この外部プロトコルは、配信プロトコルと呼ばれます。

（注） IPv4 が GRE ペイロードとして実行される場合、Protocol Type フィールドは 0x800 に設定されている必要があります。

配信プロトコルまたはペイロード プロトコルあるいはその両方としての IPv6 は、現在配布されている GRE バージョンには含まれていません。

GRE トンネル上の IPv4 転送

GREトンネル上をトンネリングされるパケットは、通常のIPパケットとしてルータに入ります。このパケットは、この IP パケットの宛先アドレスを使用して転送（ルーティング）されます。Equal Cost Multi Path（ECMP）シナリオでは、出力インターフェイスや隣接は、プラットフォーム固有の L3 ロード バランス（LB）ハッシュに基づいて選択されます。 CRS のような 2 段階の転送プラットフォームの場合、選択した出力インターフェイスの受信隣接を使用して、そのインターフェイスをホスティングする出力ラインカードにパケットを送信します。 選択した出力インターフェイスが GRE インターフェイスである場合、入力ラインカードでは、GRE トンネル宛先への到達に使用できる実際の物理インターフェイスを決定する必要があります。 このために、2 番めのルーティング（転送）の決定が、（L3 ロード バランス ハッシュが物理インターフェイスを決定するために再度適用される）GREトンネル宛先アドレスに基づいて行われます。出力物理インターフェイスが判明すると、パケットは、GRE ヘッダーでまずカプセル化され、続いて物理インターフェイスのL2書き換えヘッダーでカプセル化された後に、そのインターフェイスから送信されます。 GRE カプセル化パケットがリモート トンネル エンドポイント ルータに到達した後、GRE パケットのカプセル化が解除されます。 外側の IP ヘッダーの宛先アドレスのルックアップ（トンネル宛先アドレスと同じ）では、入力ラインカード上のローカル アドレス（受信）エントリを検出します。

GREカプセル化解除の最初の手順は、GREパケットがルータに入ることを許可する前に、トンネルの送信元（外側の IP ヘッダーの送信元 IP アドレスと同じ）とトンネルの宛先（外側の IP ヘッダーの宛先IPアドレスと同じ）の組み合わせに基づいてトンネルエンドポイントが適格であるか調べることです。受信したパケットは、トンネルアドミタンス認定チェックに失敗すると、カプセル化解除ルータによってドロップされます。 トンネル アドミタンス チェックに成功すると、カプセル化解除により、外側の IP ヘッダーと GRE ヘッダーがパケットから取り除かれ、次に内部ペイロード パケットの処理が通常のパケットとして開始されます。

トンネル エンドポイントが、IPv4 パケットをペイロードとして持つ GRE パケットをカプセル化解除する場合、IPv4 ペイロード パケット内の宛先アドレスを使用してそのパケットを転送し、ペイロード パケットの TTL が減少する必要があります。 そのようなパケットを転送する場合は注意する必要があります。ペイロードパケットの宛先アドレスがパケットのエンカプスレータ（トンネルの反対側など）である場合、ループが発生する可能性があります。 この場合、そのパケットを廃棄する必要があります。

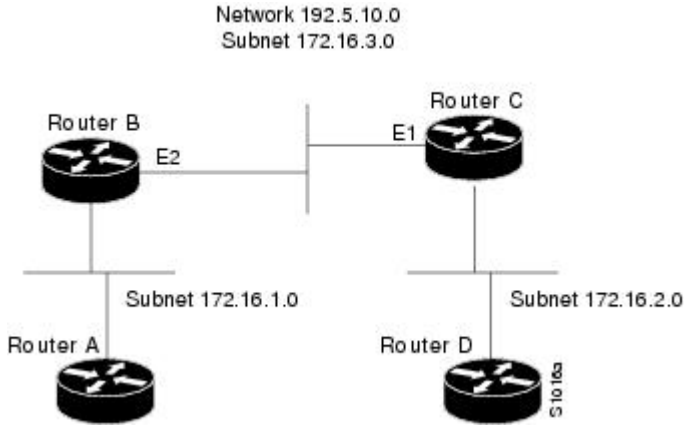
ネットワーク スタック IPv4 および IPv6 の実装の設定例

ここでは、次の設定例について説明します。

分離されたサブネットからのネットワークの作成：例

次の例では、ネットワーク 172.16.0.0 のサブネット 1 および 2 が、図 10：分離されたサブネットからのネットワークの作成，（42 ページ）に示すように、バックボーンによって分離されています。 これら 2 つのネットワークは、セカンダリ アドレスを使用して同じ論理ネットワークに入れられます。

図 *10*：分離されたサブネットからのネットワークの作成



次に、ルータ B および C の設定例を示します。

ルータ B の設定

```
configure
interface gigabitethernet 0/0/0/2
ipv4 address 192.5.10.1 255.255.255.0
ipv4 address 172.16.3.1 255.255.255.0 secondary
```

ルータ C の設定

```
configure
interface gigabitethernet 0/0/0/1
ipv4 address 192.5.10.2 255.255.255.0
ipv4 address 172.16.3.2 255.255.255.0 secondary
```

アンナンバード インターフェイスの割り当て：例

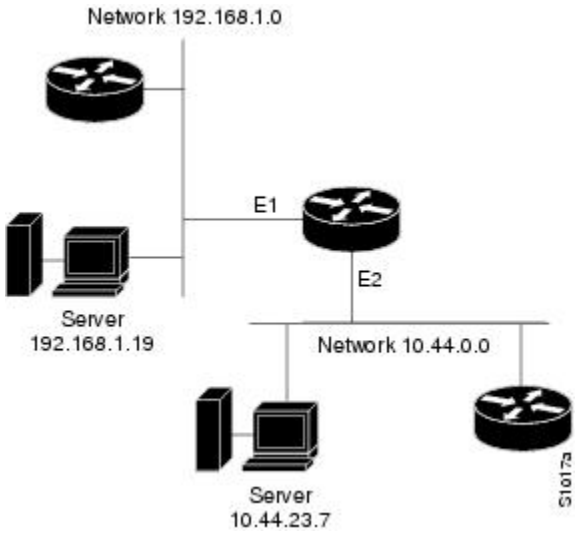
次の例では、2 番めのインターフェイス（GigabitEthernet 0/1/0/1）にループバック インターフェイス 0 のアドレスが付与されています。 このループバック インターフェイスはアンナンバードです。

```
interface loopback 0
 ipv4 address 192.168.0.5 255.255.255.0
interface gigabitethernet 0/1/0/1
 ipv4 unnumbered loopback 0
```

ヘルパー アドレスの設定：例

次の例では、1 つのルータがネットワーク 192.168.1.0 上にあり、別のルータはネットワーク 10.44.0.0 上にあり、いずれかのネットワーク セグメント上のホストからの IP ブロードキャストが両方のサーバに到達できるようにする必要があります。図 11：IP ヘルパー アドレス，（43 ページ）に、ネットワーク 10.44.0.0 をネットワーク 192.168.1.0 に接続するルータを設定する方法を示します。

図 *11*：*IP* ヘルパー アドレス



次に、設定例を示します。

```
!
interface gigabitethernet 0/0/0/1
 ipv4 helper-address 10.44.23.7
interface gigabitethernet 0/0/0/2
 ipv4 helper-address 192.168.1.19
```

VRF big モードの設定

次のタスクを実行して、VRF の big モードを設定します。

手順の概要

1. **configure**
2. **vrf** *vrf-name*
3. **mode big**
4. 次のいずれかのコマンドを使用します。
   - **end**
   - **commit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `RP/0/RSP0/CPU0:router# configure`	グローバル コンフィギュレーション モードを開始します。

ステップ 2	**vrf** *vrf-name*
	
	例：
	
	`RP/0/RSP0/CPU0:router(config)# vrf v1`
	`RP/0/RSP0/CPU0:router(config-vrf)#`
		VRF コンフィギュレーション モードを開始します。

ステップ 3	**mode big**
	
	例：
	
	`RP/0/RSP0/CPU0:router(config-vrf)# mode big`
	`RP/0/RSP0/CPU0:router(config-vrf)#`
		対応する VRF の big モードを開始します。

ステップ 4	次のいずれかのコマンドを使用します。
	• **end**
	• **commit**
	
	例：
	
	`RP/0/RSP0/CPU0:router(config)# end`
	または
	
	`RP/0/RSP0/CPU0:router(config)# commit`
		設定変更を保存します。
		• **end** コマンドを実行すると、変更をコミットするように要求されます。
		
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`
		
		◦ **yes** と入力すると、実行コンフィギュレーション ファイルに変更が保存され、コンフィギュレーションセッションが終了して、ルータがEXECモードに戻ります。
		◦ **no** と入力すると、コンフィギュレーションセッションが終了して、ルータが EXEC モードに戻ります。変更はコミットされません。
		◦ **cancel** と入力すると、現在のコンフィギュレーションセッションが継続します。コンフィギュレーションセッションは終了せず、設定変更もコミットされません。

	コマンドまたはアクション	目的
		• 実行コンフィギュレーション ファイルに変更を保存し、コンフィギュレーションセッションを継続するには、**commit** コマンドを使用します。

その他の参考資料

ここでは、ネットワーク スタック IPv4 および IPv6 の実装に関する関連資料について説明します。

関連資料

関連項目	参照先
アドレス解決の設定タスク	このマニュアルの「*ARP* の設定」の章。
ホスト名の IP アドレスへのマッピング	『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*Host Services and Applications Commands*」の章
ネットワーク スタック IPv4 および IPv6 のコマンド：コマンド構文の詳細、コマンドモード、コマンド履歴、デフォルト設定、使用上のガイドライン、および例	『*Cisco ASR 9000 Series Aggregation Services Router IP Addresses and Services Command Reference*』の「*Network Stack IPv4 and IPv6 Commands*」の項

標準

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

MIB

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

RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	—

シスコのテクニカル サポート

説明	リンク
シスコのテクニカルサポート Web サイトでは、製品、テクノロジー、ソリューション、技術的なヒント、およびツールへのリンクなどの、数千ページに及ぶ技術情報が検索可能です。Cisco.com に登録済みのユーザは、このページから詳細情報にアクセスできます。	http://www.cisco.com/en/US/support/index.html