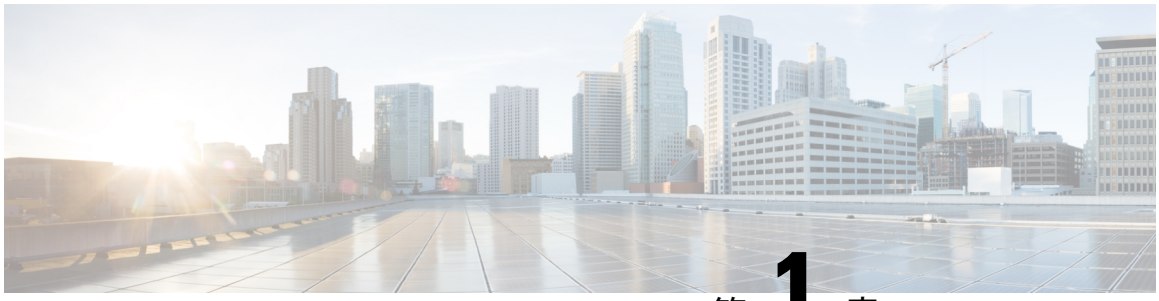
Cisco NCS 560 シリーズ ルータ（IOS XR リリース6.6.x）IP アドレスおよびサービス コンフィギュレーション ガイド

初版：2019 年 5 月 30 日

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

目 次



第 1 章

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。

制約事項

IPv6 に対応している Cisco IOS XR ソフトウェア リリースでは、1 つのインターフェイス上に複数の IPv6 グローバル アドレスを設定できます。ただし、1 つのインターフェイス上での複数の IPv6 リンクローカル アドレスはサポートされません。

フォールバック VRF の実装

仮想ルーティングおよびフォワーディング（VRF）は、同じルータ上にルーティング テーブルの複数のインスタンスが同時に存在できるようにする IP テクノロジーです。ルーティング インスタンスは独立しているため、同じ IP アドレスを競合することなく使用できます。

データ パケットの宛先プレフィックスが、設定されている VRF のどのルートとも一致しない場合、グローバル ルーティング テーブルからデフォルト ルートが識別されます。ただし、デフォルト ルートを使用するには明示的なネクスト ホップが必要なため、効率的でない可能性があります。フォールバック VRF ルートを設定することをお勧めします。宛先が VRF テーブルで一致しない場合は、フォールバック VRF テーブルが使用されます。フォールバック VRF には、グローバル ルーティング テーブルまたは非グローバル VRF テーブルを使用できます。

制約事項

フォールバック VRF ルートを設定する場合は、次の制約事項が適用されます。

- フォールバック VRF ルートは、各プライマリ VRF のアドレス ファミリごとに 1 つのみ設定できます。
- LPTS 受信トラップはサポート対象外であるため、ping、トレースルート、または低速パス アプリケーションはフォールバック VRF でサポートされません。
- Cisco NCS 560 シリーズ ルータでは、1000 の VRF と 1 つのグローバル テーブルのみがサポートされます。
- VRF へのスタティック デフォルト ルートを設定すると、このスタティック デフォルト ルートがフォールバック VRF よりも優先されます。VRF のデフォルト ルートを設定すると、ルート ルックアップにグローバル ルーティング テーブルが使用されます。デフォルト ルートは、設定済みのネクスト ホップに必ず転送されます。
- プライマリ VRF でパケットのルート ルックアップが失敗した場合、フォールバック VRF でルート ルックアップを実行するためにパケットがリサイクルされます。したがって、パケットのルーティング パフォーマンスが最大で 50% 低下します。
- パケットの ACL ベース転送（ABF）VRF リダイレクトと VRF フォールバックの両方を設定すると、パケットは2回リサイクルされます。したがって、パケットのルーティング パフォーマンスが最大で 33% 低下します。
- フォールバック VRF でパケットのルートが見つかった場合、グリーニング IPv4 およびグリーニング IPv6 隣接関係パケットのみが正常にパントされます。
- ループ設定では、パケットのルートがプライマリとフォールバックのどちらの VRF でも見つからない場合、パケットはリサイクル パスでループします。最終的にパケットはリサイクル出力キューにドロップされます。リサイクル キューの優先順位が最も高いため、ループしているトラフィックのレートが高いと、他の正常なリサイクル パケットがドロップされる可能性があります。

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** および **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** および **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、設定時に、競合する IP アドレス エントリが許可されます。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、 0/0/0/1 が 0/0/0/2 と競合する場合、 0/0/0/2 上の IPv4 プロトコルは停止され、IPv4 は 0/0/0/1 上で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 は、Open Shortest Path First（OSPF）やマルチプロトコル ボーダー ゲートウェイ プロトコル（BGP）など、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。

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。

IPv4 アドレス指定の設定

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。

設定例

IPv4 アドレス 192.168.1.27 とネットワーク マスク「/8」が、**インターフェイス HundredGigE 0/9/0/1** に割り当てられます。

（注） 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。

```
Router#configure HundredGigE0/9/0/1
Router(config)#interface HundredGigE 0/9/0/1
Router(config-if)#ipv4 address 192.168.1.27/8
Router(config-if)#commit
```

実行コンフィギュレーション

```
Router#show running-config interface HundredGigE0/9/0/1
interface HundredGigE0/9/0/1
 ipv4 address 192.168.1.27 255.0.0.0
!
```

確認

HundredGigEインターフェイスがアクティブであり、IPv4がイネーブルであることを確認します。

```
Router# show ipv4 interface HundredGigE0/9/0/1

interface HundredGigE0/9/0/1 is Up, ipv4 protocol is Up
  Vrf is default (vrfid 0x60000000)
  Internet address is 192.168.1.27/8
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Multicast reserved groups joined: 224.0.0.2 224.0.0.1
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000
```

関連コマンド

- ipv4 address
- show ipv4 interface

IPv6 アドレッシングの設定

ルータでのIPv6トラフィックのグローバル転送を可能にするため、IPv6アドレスは個々のルータインターフェイスに設定されます。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* 引数には、RFC 2373 に記載されている形式を使用する必要があります。コロンで区切った 16 ビット値を使用して 16 進数でアドレスを指定してください。

ipv6 address コマンドの **/prefix-length** 引数には、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の数（アドレスのネットワーク部）を示す 10 進数の値を指定します。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* 引数には、RFC 2373 に記載されている形式を使用する必要があります。コロンで区切った 16 ビット値を使用して 16 進数でアドレスを指定してください。

IPv6 マルチキャスト グループ

インターフェイスで IPv6 トラフィックを転送できるようにするには、そのインターフェイスで IPv6 アドレスを設定する必要があります。インターフェイスにグローバル IPv6 アドレスを設定すると、リンクローカル アドレスが自動的に設定され、そのインターフェイスに対して IPv6 がアクティブになります。

また、設定されたインターフェイスは、そのリンクに必要な次のマルチキャスト グループに自動的に加入します。

- 送信要求ノード マルチキャスト グループ FF02:0:0:0:0:1:FF00::/104（インターフェイスに割り当てられた各ユニキャスト アドレス用）
- 全ノード リンクローカル マルチキャスト グループ FF02::1
- 全ルータ リンクローカル マルチキャスト グループ FF02::2

（注） 送信要求ノード マルチキャスト アドレスは、ネイバー探索プロセスで使用されます。

設定例

IPv6 アドレス 2001:0DB8:0:1::1/64 が、**インターフェイス HundredGigE 0/9/0/1** に割り当てられます。

```
Router#configure
Router(config)#interface HundredGigE 0/9/0/1
Router(config-if)#ipv6 address 2001:0DB8:0:1::1/64
Router(config-if)#commit
```

実行コンフィギュレーション

```
Router#show running-config interface HundredGigE0/9/0/1
interface HundredGigE0/9/0/1
 ipv4 address 192.168.1.27 255.0.0.0
 ipv4 address 1.0.0.1 255.255.255.0 secondary
 ipv4 address 2.0.0.1 255.255.255.0 secondary
 ipv6 address 2001:db8:0:1::1/64
!
```

確認

HundredGigE インターフェイスがアクティブであり、IPv6 が有効であることを確認します。

```
Router#show ipv6 interface HundredGigE0/9/0/1
HundredGigE0/9/0/1 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
  IPv6 is enabled, link-local address is fe80::c672:95ff:fea6:1c75
  Global unicast address(es):
    2001:db8:0:1::1, subnet is 2001:db8:0:1::/64
  Joined group address(es): ff02::1:ff00:1 ff02::1:ffa6:1c75 ff02::2
      ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND cache entry limit is 1000000000
  ND advertised retransmit interval is 0 milliseconds
  Hosts use stateless autoconfig for addresses.
  Outgoing access list is not set
  Inbound  access list is not set
  Table Id is 0xe0800000
```

```
  Complete protocol adjacency: 0
  Complete glean adjacency: 0
  Incomplete protocol adjacency: 0
  Incomplete glean adjacency: 0
  Dropped protocol request: 0
  Dropped glean request: 0
```

関連コマンド

- ipv6 address
- interface
- show ipv6 interface

設定例

IPv6 アドレス 2001:0DB8:0:1::/64 が、**インターフェイス HundredGigE 0/9/0/1** に割り当てられます。**eui-64** キーワードを指定して、IPv6 アドレスの下位 64 ビットにインターフェイス識別子（ID）を含むサイトローカルおよびグローバル IPv6 アドレスを設定します。指定する必要があるのはアドレスの 64 ビット ネットワーク プレフィックスだけです。最後の 64 ビットはインターフェイス ID から自動的に計算されます。

```
Router#configure
Router(config)#interface HundredGigE 0/9/0/1
Router(config-if)#ipv6 address 2001:0DB8:0:1::/64 eui-64
Router(config-if)#commit
```

実行コンフィギュレーション

```
Router#show running-config interface HundredGigE0/9/0/1
interface HundredGigE0/9/0/1
 ipv4 address 192.168.1.27 255.0.0.0
 ipv4 address 1.0.0.1 255.255.255.0 secondary
 ipv4 address 2.0.0.1 255.255.255.0 secondary
 ipv6 address 2001:db8:0:1::/64 eui-64
!
```

確認

HundredGigE インターフェイスがアクティブであり、IPv6 が有効であることを確認します。

```
Router#show ipv6 interface HundredGigE0/9/0/1
HundredGigE0/9/0/1 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
  IPv6 is enabled, link-local address is fe80::c672:95ff:fea6:1c75
  Global unicast address(es):
    2001:db8:0:1:c672:95ff:fea6:1c75, subnet is 2001:db8:0:1::/64
  Joined group address(es): ff02::1:ffa6:1c75 ff02::2 ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND cache entry limit is 1000000000
  ND advertised retransmit interval is 0 milliseconds
  Hosts use stateless autoconfig for addresses.
  Outgoing access list is not set
  Inbound  access list is not set
  Table Id is 0xe0800000
```

```
  Complete protocol adjacency: 0
  Complete glean adjacency: 0
  Incomplete protocol adjacency: 0
  Incomplete glean adjacency: 0
  Dropped protocol request: 0
  Dropped glean request: 0
```

関連コマンド

- ipv6 address
- interface
- show ipv6 interface

設定例

IPv6 アドレス FE80::260:3EFF:FE11:6770 が、**インターフェイス HundredGigE 0/9/0/1** に割り当てられます。link-local キーワードを指定して、リンクローカル アドレスをインターフェイスに設定します。このアドレスは、IPv6 がインターフェイスでイネーブルになっているときに自動的に設定されるリンクローカル アドレスの代わりに使用されます。

```
Router#configure
Router(config)#interface HundredGigE 0/9/0/1
Router(config-if)#ipv6 address FE80::260:3EFF:FE11:6770 link-local
Router(config-if)#commit
```

実行コンフィギュレーション

```
Router#show running-config interface HundredGigE0/9/0/1

interface HundredGigE0/9/0/1
 ipv6 address fe80::260:3eff:fe11:6770 link-local
!
```

確認

HundredGigE インターフェイスがアクティブであり、IPv6 がリンクローカル アドレスで有効になっていることを確認します。

```
Router#show ipv6 interface HundredGigE0/9/0/1
HundredGigE0/9/0/1 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
  IPv6 is enabled, link-local address is fe80::260:3eff:fe11:6770
  Global unicast address(es):
    2001:db8:0:1:260:3eff:fe11:6770, subnet is 2001:db8:0:1::/64
  Joined group address(es): ff02::1:ff11:6770 ff02::2 ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND cache entry limit is 1000000000
  ND advertised retransmit interval is 0 milliseconds
  Hosts use stateless autoconfig for addresses.
  Outgoing access list is not set
  Inbound  access list is not set
  Table Id is 0xe0800000
  Complete protocol adjacency: 0
  Complete glean adjacency: 0
```

```
  Incomplete protocol adjacency: 0
  Incomplete glean adjacency: 0
  Dropped protocol request: 0
  Dropped glean request: 0
```

関連コマンド

- ipv6 address
- interface
- show ipv6 interface

設定例

明示的な IPv6 アドレスが設定されていない**インターフェイス HundredGigE 0/9/0/1** での IPv6 の処理を有効にします。

```
Router#configure
Router(config)#interface HundredGigE 0/9/0/1
Router(config-if)#ipv6 enable
Router(config-if)#commit
```

実行コンフィギュレーション

```
Router#show running-config interface HundredGigE0/9/0/1
interface HundredGigE0/9/0/1
ipv6 enable
!
```

確認

HundredGigE インターフェイスがアクティブであり、IPv6 が有効であることを確認します。

```
Router#show ipv6 interface HundredGigE0/9/0/1
HundredGigE0/9/0/1 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
  IPv6 is enabled, link-local address is fe80::c672:95ff:fea6:1c75
  No global unicast address is configured
  Joined group address(es): ff02::1:ffa6:1c75 ff02::2 ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND cache entry limit is 1000000000
  ND advertised retransmit interval is 0 milliseconds
  Hosts use stateless autoconfig for addresses.
  Outgoing access list is not set
  Inbound  access list is not set
  Table Id is 0xe0800000
  Complete protocol adjacency: 0
  Complete glean adjacency: 0
  Incomplete protocol adjacency: 0
  Incomplete glean adjacency: 0
  Dropped protocol request: 0
  Dropped glean request: 0
```

関連コマンド

- ipv6 enable
- interface
- show ipv6 interface

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

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（セカンダリ 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。

設定例

セカンダリ IPv4 アドレス 192.168.1.27 が Hundredgige インターフェイス 0/0/0/1 に割り当てられます。

注：IPv6 の場合は、**secondary** キーワードを指定せずに、インターフェイスに複数の IPv6 アドレスを設定できます。

```
Router# configure
Router(config)# interface HundredGigE 0/9/0/1
Router(config-if)# ipv4 address 192.168.1.27 255.255.255.0 secondary
Router(config-if)#commit
```

実行コンフィギュレーション

```
Router#show running-config interface HundredGigE0/9/0/1

interface HundredGigE0/9/0/1
 ipv4 address 192.168.1.27 255.255.255.0 secondary
!
```

確認

```
Router#show ipv4 interface HundredGigE0/9/0/1
HundredGigE0/9/0/1 is Up, ipv4 protocol is Up
  Vrf is default (vrfid 0x60000000)
  Internet address is unassigned
  Secondary address 192.168.1.27/24
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Multicast reserved groups joined: 224.0.0.2 224.0.0.1
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000
```

関連コマンド

- ipv4 address
- show ipv4 interface

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。

設定例

インターフェイス HundredGigE 0/9/0/1 に IPv4 アドレス 192.168.99.1 と IPv6 アドレス 2001:0DB8:c18:1::3/64 を設定します。

```
Router#configure
Router(config)#interface HundredGigE 0/9/0/1
Router(config-if)#ipv4 address 192.168.99.1 255.255.255.0
Router(config-if)#ipv6 address 2001:0DB8:c18:1::3/64
Router(config-if)#commit
```

実行コンフィギュレーション

```
Router# show running-config interface HundredGigE0/9/0/1
interface HundredGigE0/9/0/1
 ipv4 address 192.168.99.1 255.255.255.0
 ipv6 address 2001:db8:c18:1::3/64
!
```

確認

HundredGigE インターフェイスがアクティブであり、IPv4 と IPv6 が有効になっていることを確認します。

```
Router#show ipv4 interface HundredGigE0/9/0/1
HundredGigE0/9/0/1 is Up, ipv4 protocol is Up
  Vrf is default (vrfid 0x60000000)
  Internet address is 192.168.99.1/24
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Multicast reserved groups joined: 224.0.0.2 224.0.0.1
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  access list is not set
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000

Router#show ipv6 interface HundredGigE0/9/0/1
HundredGigE0/9/0/1 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
  IPv6 is enabled, link-local address is fe80::c672:95ff:fea6:1c75
  Global unicast address(es):
    2001:db8:c18:1::3, subnet is 2001:db8:c18:1::/64
  Joined group address(es): ff02::1:ff00:3 ff02::1:ffa6:1c75 ff02::2
      ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND cache entry limit is 1000000000
  ND advertised retransmit interval is 0 milliseconds
  Hosts use stateless autoconfig for addresses.
  Outgoing access list is not set
  Inbound  access list is not set
  Table Id is 0xe0800000
  Complete protocol adjacency: 0
  Complete glean adjacency: 0
  Incomplete protocol adjacency: 0
  Incomplete glean adjacency: 0
  Dropped protocol request: 0
  Dropped glean request: 0
```

関連コマンド

- ipv4 address
- ipv6 address
- show ipv4 interface
- show ipv6 interface

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インターフェイスには、アンナンバードを設定できます。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なければなりません。
- インターフェイスには IP アドレスがないため、ping EXEC コマンドを使用してインターフェイスがアップ状態かどうかを確認することはできません。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。

設定例

明示的な IP アドレスを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します。

```
Router#configure
Router(config)#interface HundredGigE 0/9/0/1
Router(config-if)#ipv4 unnumbered loopback 0
Router(config-if)#commit
```

実行コンフィギュレーション

```
Router#show running-config interface HundredGigE0/9/0/1
interface HundredGigE0/9/0/1
 ipv4 point-to-point
```

```
 ipv4 unnumbered Loopback0
!
```

確認

```
Router#show interface HundredGigE0/9/0/1
HundredGigE0/9/0/1 is up, line protocol is up
  Interface state transitions: 5
  Hardware is Hundredgige, address is 00e2.2a33.445b (bia 00e2.2a33.445b)
  Layer 1 Transport Mode is LAN
  Internet address is 10.0.0.2/32
  MTU 1514 bytes, BW 10000000 Kbit (Max: 10000000 Kbit)
     reliability 255/255, txload 194/255, rxload 0/255
  Encapsulation ARPA,
  Full-duplex, 10000Mb/s, link type is force-up
  output flow control is off, input flow control is off
  Carrier delay (up) is 10 msec
  loopback not set,
  Last link flapped 01:38:49
  ARP type ARPA, ARP timeout 04:00:00
  Last input 00:00:00, output 00:00:00
  Last clearing of "show interface" counters 02:34:16
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 7647051000 bits/sec, 12254894 packets/sec
     1061401410 packets input, 82789675614 bytes, 0 total input drops
     0 drops for unrecognized upper-level protocol
     Received 5 broadcast packets, 19429 multicast packets
              0 runs, 0 giants, 0 throttles, 0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     76895885948 packets output, 6192569128048 bytes, 0 total output drops
     Output 7 broadcast packets, 18916 multicast packets
     0 output errors, 0 underruns, 0 applique, 0 resets
     0 output buffer failures, 0 output buffers swapped out
     2 carrier transitions

Router #show run int lo 0
interface Loopback0
 ipv4 address 10.0.0.2 255.255.255.255
```

関連コマンド

- ipv4 unnumbered
- show interfaces

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。

設定例

IPv4 ICMP 宛先到達不能メッセージが 1,000 ミリ秒ごとに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、Don't Fragment（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ように ICMP 宛先到達不能メッセージを送信するレートを任意で制限します。

```
Router#configure
Router(config)#icmp ipv4 rate-limit unreachable 1000
Router(config)#icmp ipv4 rate-limit unreachable DF 1000
Router(config)#commit
```

実行コンフィギュレーション

```
Router#show running-config | in icmp
Building configuration...
icmp ipv4 rate-limit unreachable DF 1000
icmp ipv4 rate-limit unreachable 1000
```

確認

```
Router#show ipv4 interface HundredGigE0/9/0/1
HundredGigE0/9/0/1 is Up, ipv4 protocol is Up
  Vrf is default (vrfid 0x60000000)
  Internet address is 192.85.1.1/24
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Multicast reserved groups joined: 224.0.0.2 224.0.0.1 224.0.0.2
      224.0.0.5 224.0.0.6
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  common access list is not set, access list is not set
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000
```

送信または受信する ICMP 到達不能メッセージの数は、**show ipv4 traffic** コマンドを使用して指定できます。

```
Router# show ipv4 traffic
ICMP statistics:
  Sent: 0 admin unreachable, 5 network unreachable
        0 host unreachable, 0 protocol unreachable
        0 port unreachable, 0 fragment unreachable
        0 time to live exceeded, 0 reassembly ttl exceeded
        0 echo request, 0 echo reply
        0 mask request, 0 mask reply
        0 parameter error, 0 redirects
        5 total
  Rcvd: 0 admin unreachable, 0 network unreachable
        0 host unreachable, 0 protocol unreachable
        0 port unreachable, 0 fragment unreachable
        0 time to live exceeded, 0 reassembly ttl exceeded
        0 echo request, 0 echo reply
        0 mask request, 0 mask reply
        0 redirect, 0 parameter error
        0 source quench, 0 timestamp, 0 timestamp reply
        0 router advertisement, 0 router solicitation
        0 total, 0 checksum errors, 0 unknown
```

関連コマンド

- icmp ipv4 rate-limit unreachable
- show ipv4 traffic

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。

設定例

50 ミリ秒の間隔と、20 トークンのバケット サイズを IPv6 ICMP エラー メッセージに設定します。

- 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。

```
Router#configure
Router(config)#ipv6 icmp error-interval 50 20
Router(config)#commit
```

実行コンフィギュレーション

```
Router#show running-config
Building configuration...
!! IOS XR Configuration version = 6.0.0.26I
!! Last configuration change at Mon Dec 14 22:07:35 2015 by root
!
hostname test-83
logging console debugging
username root
 group root-lr
 group cisco-support
 secret 5 $1$d2NC$RbAdqdU7kw/kEJoMP/IJG1
!
cdp
ipv6 icmp error-interval 50 20
icmp ipv4 rate-limit unreachable DF 1000
```

```
icmp ipv4 rate-limit unreachable 1000
ipv4 conflict-policy static
```

関連コマンド

- ipv6 icmp error-interval

柔軟な送信元 IP の選択

障害に応答する Internet Control Message Protocol（ICMP）応答パケットで柔軟な送信元 IP アドレスを選択できます。

設定例

送信元アドレスの選択に関する RFC コンプライアンスをイネーブルにします。

```
Router#configure
Router(config)#icmp ipv4 source rfc
Router(config)#commit
```

実行コンフィギュレーション

```
Router#show running-config | in source rfc
Building configuration...
icmp ipv4 source rfc
```

関連コマンド

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
- 最長プレフィックス アドレス競合解決
- 最大 IP アドレス競合解決
- 接続ルートに対する Route-Tag のサポート

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。

設定例

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。

```
Router#configure
Router(config)#ipv4 conflict-policy static
*/For IPv6, use the ipv6 conflict-policy static command/*
Router(config)#commit
```

実行コンフィギュレーション

```
Router#show running-config | in ipv4 config
Building configuration...
!! IOS XR Configuration version = 6.0.0.26I
!! Last configuration change at Mon Dec 14 21:57:27 2015 by root
!
hostname sample-83
logging console debugging
username root
 group root-lr
 group test
 secret 5 $1$d2NC$RbAdqdU7kw/eKJpMo/GJI1
!
cdp
ipv4 conflict-policy static
interface Loopback0
 ipv4 address 1.1.1.1 255.255.255.255
!
…..
```

確認

```
Router#show arm ipv4 conflicts
F Forced down
| Down interface & addr             Up interface & addr VRF

F tenGigE 0/11/0/0 192.85.1.2/24   HundredGigE0/9/0/1  192.85.1.1/24 default


Forced down interface         Up interface                    VRF
```

関連コマンド

- ipv4 conflict-policy
- ipv6 conflict-policy

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は、プレフィックスが最長の 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。つまり、現在実行している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競合セット内のすべてのアドレスが、同様に実行を許可されます。

設定例

最長プレフィックス アドレス競合解決を設定します。

```
Router# configure
Router(config)# ipv4 conflict-policy longest-prefix
*/For IPv6, use the ipv6 conflict-policy command*/
Router(config)# commit
```

実行コンフィギュレーション

```
Router# show running-config | in longest-prefix
Building configuration...
ipv4 conflict-policy longest-prefix
```

確認

```
Router#show arm ipv4 conflicts
F Forced down
| Down interface & addr             Up interface & addr VRF

F tenGigE 0/11/0/0 192.85.1.2/24   HundredGigE0/9/0/1  192.85.1.1/24 default


Forced down interface         Up interface                   VRF
```

最大 IP アドレス競合解決

この競合解決ポリシーは、値が最も高い 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。つまり、値が最も高い IP アドレスが優先されます。

設定

最大 IP アドレス競合解決を設定します。

```
Router# configure
Router(config)#ipv4 conflict-policy highest-ip
*/For IPv6, use the ipv6 conflict-policy highest-ip command/*
Router(config)#commit
```

実行コンフィギュレーション

```
Router#show running-config | in highest-ip
Building configuration...
ipv4 conflict-policy highest-ip
```

確認

```
Router#show arm ipv4 conflicts
F Forced down
| Down interface & addr             Up interface & addr VRF

F tenGigE 0/11/0/0 192.85.1.2/24   HundredGigE0/9/0/1  192.85.1.1/24 default


Forced down interface         Up interface                   VRF
```

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 のサポート機能は、インターフェイスのすべての IPv4 および IPv6 アドレス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。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 ルートと接続ルート（インターフェイス）では、このルート タグ機能はすでに利用可能になっています。

設定例

インターフェイス HundredGigE 0/9/0/1 に対して、ルート タグ 20 が付いた IPv4 アドレス 10.0.54.2/30 を指定します。

```
Router#configure
Router(config)#interface HundredGigE 0/9/0/1
Router(config-if)#ipv4 address 10.0.54.2/30 route-tag 1899
Router(config)#commit
```

実行コンフィギュレーション

```
Router#show running-config interface HundredGigE0/9/0/1

interface HundredGigE0/9/0/1
 ipv4 address 10.0.54.2/30 route-tag 1899
!
```

確認

ルートのパラメータを確認します。

```
Router#show route 10.0.54.2
Routing entry for 10.0.54.2/32
  Known via "local", distance 0, metric 0 (connected)
  Tag 1899
Routing Descriptor Blocks
    directly connected, via HundredGigE0/9/0/1
      Route metric is 0
  No advertising protos.
```

関連コマンド

- route-tag

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 （21 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 ビット値がゼロで表されている場合は、 *ipv6-address* 引数の一部として 2 つのコロンを使用できます。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 （21 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 （21 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-prefix/prefix-length 形式の IPv6 アドレス プレフィックスを使用すると、アドレス空間全体のビット単位の連続ブロックを表現できます。 *ipv6-prefix* 引数には、RFC 2373 に記載されている形式を使用する必要があります。コロンで区切った 16 ビット値を使用して 16 進数でアドレスを指定してください。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るパケットは、そのアドレスで識別される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

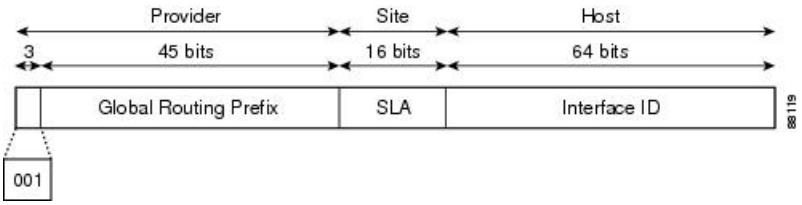
集約可能グローバル アドレス

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。次の図は、集約可能グローバルアドレスの構造を示しています。

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。

- 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。

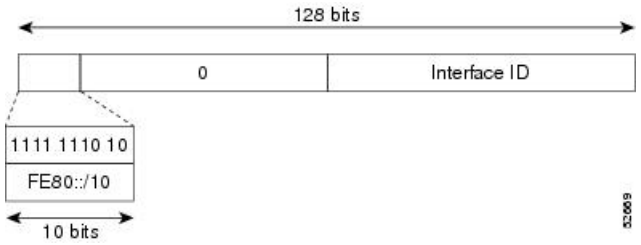
1. ルータに MAC アドレスが（ルータの MAC アドレス プールから）照会されます。
2. 使用できる MAC アドレスがない場合は、ルート プロセッサ（RP）またはラインカード（LC）のシリアル番号を使用して、リンクローカル アドレスを形成します。

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。次の図は、以下のリンク ローカル アドレスの構造を示しています。

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。

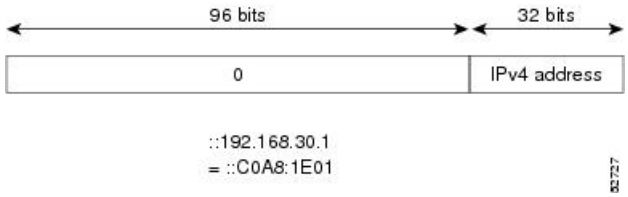
図 *2:* リンクローカル アドレス形式



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。次の図は、IPv4 互換 IPv6 アドレスの構造と、許容されるアドレス フォーマットのいくつかを示しています。

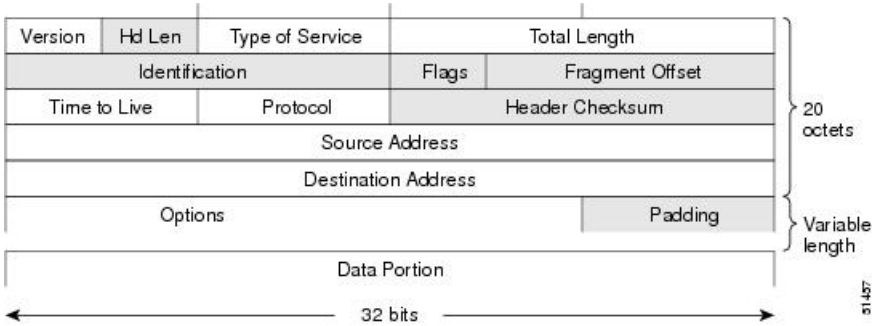
図 *3 : IPv4* 互換 *IPv6* アドレス形式



簡易 IPv6 パケット ヘッダー

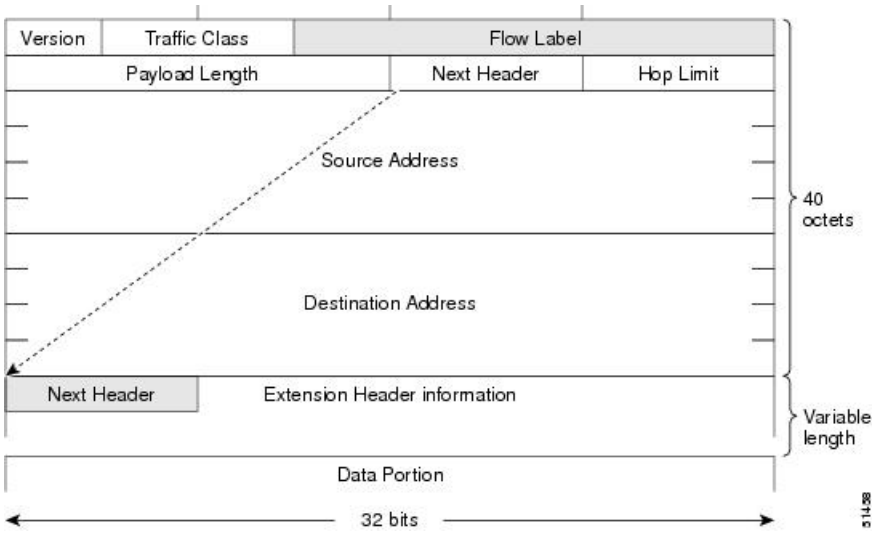
基本 IPv4 パケット ヘッダーには、合計サイズが 20 オクテット（160 ビット）の 12 のフィールドがあります。この 12 個のフィールドの後にはオプション フィールドが続く場合があり、さらにその後に、通常はトランスポートレイヤ パケットであるデータ部分が続きます。可変長のオプション フィールドは、IPv4 パケット ヘッダーの合計サイズに加算されます。IPv4 パケット ヘッダーのグレーの部分のフィールドは、IPv6 パケット ヘッダーに含まれません。

図 *4 : IPv4* パケット ヘッダー形式



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、UDP チェックサムを使用して内部パケットの完全性を確認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 ビットに揃えられています。これにより、IPv6 パケットの処理が容易になります。

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。

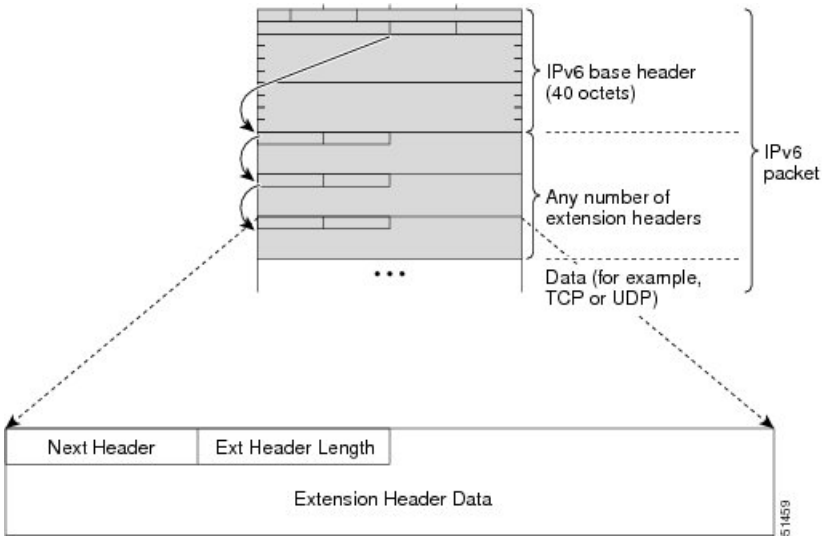
表 *2 :* 基本 *IPv6* パケット ヘッダー フィールド

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。

フィールド	説明
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。
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 ヘッダーの後ろに続く情報のタイプは、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。次の図は、IPv6 拡張ヘッダーの形式を示しています。

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。
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。

ヘッダー タイプ	次ヘッダーの値	説明
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。
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。

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 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

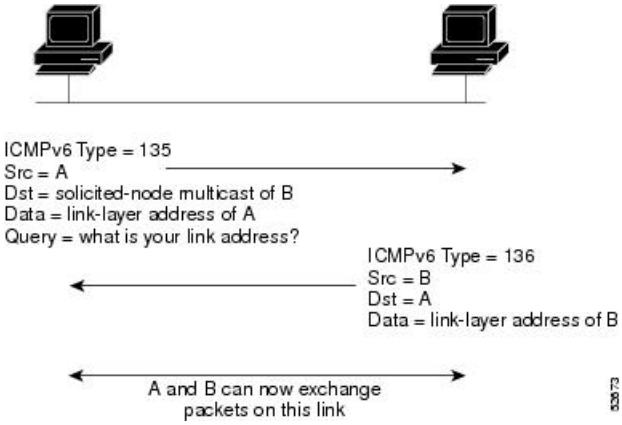
IPv6 のネイバー探索プロセスは、ICMP メッセージと送信要求ノード マルチキャスト アドレスを使用して、同じネットワーク（ローカル リンク）上のネイバーのリンク層アドレスを判別し、ネイバーの到達可能性を確認して、隣接ルータの状況を把握します。

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ノードが別のノードのリンク

層アドレスを判断する必要がある場合、ネイバー請求メッセージ内の送信元アドレスは、ネイバー請求メッセージを送信するノードの IPv6 アドレスです。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。（送信要求フラグに値1が設定されている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）。非送信要求メッセージでは、送信元ノードから宛先ノードへの一方向パスだけが確認されます。送信要求ネイバー アドバタイズメント メッセージは、両方向のパスが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。

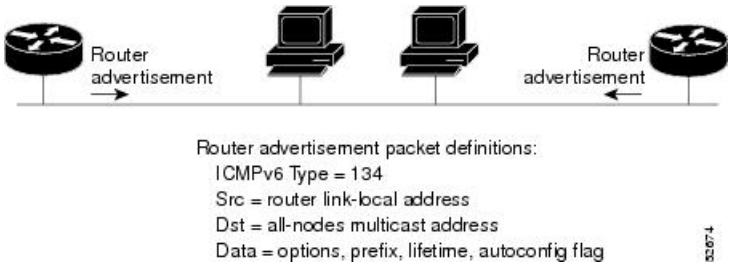
ネイバー送信要求メッセージは、ユニキャスト IPv6 アドレスがインターフェイスに割り当てられる前にそのアドレスが一意であることを確認するために、ステートレス自動設定プロセスでも使用されます。アドレスがインターフェイスに割り当てられる前に、重複アドレス検出がまず新しいリンクローカル IPv6 アドレスで実行されます（重複アドレス検出の実行時、新しいアドレスは暫定的な状態にあります）。具体的には、ノードは未指定の送信元アドレスと一時的なリンクローカルアドレスをメッセージの本文に含むネイバー送信要求メッセージを送信します。そのアドレスが別のノードですでに使用されている場合、ノードは一時的なリンクローカル 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チェックされません。

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



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

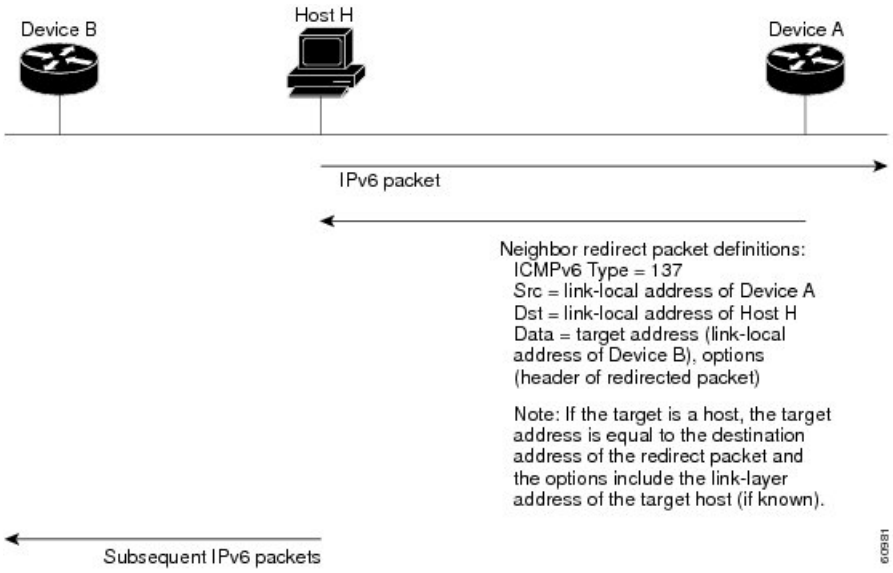
設定されたパラメータはインターフェイスに固有です。ルータ アドバタイズメント メッセージ（デフォルト値を含む）の送信は、イーサネットと FDDI インターフェイス上では自動的にイネーブルになります。その他のインターフェイス タイプの場合、ルータ アドバタイズメント メッセージの送信は、インターフェイス コンフィギュレーション モードで **no ipv6 nd suppress-ra** コマンドを使用して手動で設定する必要があります。ルータ アドバタイズメント メッセージの送信を個々のインターフェイスでディセーブルにするには、インターフェイス コンフィギュレーション モードで **ipv6 nd suppress-ra** コマンドを使用します。

（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



（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。

ルータがすべての IPv6 ICMP エラー メッセージ（ネイバー リダイレクト メッセージを含む）を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。

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

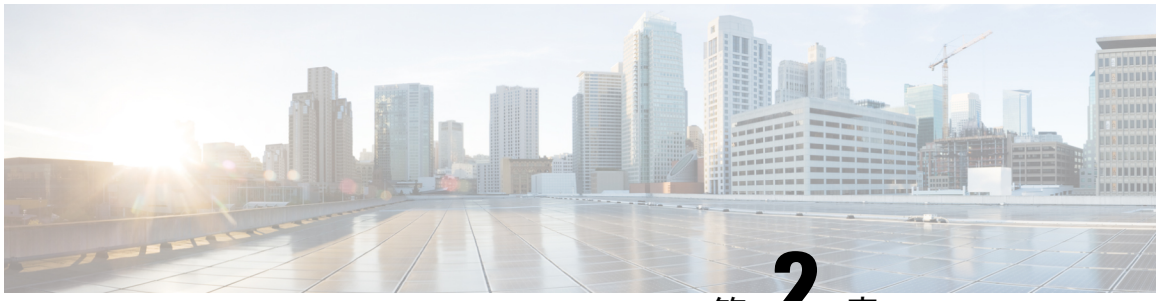
アドレス 3：10.3.1.1/16

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。



第 2 章

ARP の設定

ARP の設定

アドレス解決はネットワーク アドレスをメディア アクセス コントロール（MAC）アドレスにマッピングするプロセスです。通常は、ARP プロトコルを使用してシステムにより動的に実行されますが、スタティック ARP エントリの設定によって実行することもできます。このプロセスを実現するのに使用されるのが、アドレス解決プロトコル（ARP）です。

ARP は、IP アドレスをメディアや MAC アドレスに関連付けるために使用されます。ARP は IP アドレスを入力とし、関連するメディアのアドレスを決定します。メディアまたは MAC アドレスが決定すると、IP アドレスまたはメディア アドレスの関連付けは、すぐ取得できるように ARP のキャッシュに保管されます。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。

ARP の詳細については、次を参照してください。 ARP の設定に関する情報 （42 ページ）

ARP およびプロキシ ARP

Cisco IOS XR ソフトウェアでは、アドレス解決プロトコル（ARP）とプロキシ ARP という 2 つの形式のアドレス解決がサポートされています。これらのプロトコルは、それぞれ RFC 826 と RFC 1027 で定義されています。Cisco IOS XR ソフトウェアでは、ローカル プロキシ ARP という ARP の形式もサポートされています。

プロキシ ARP とローカル プロキシ ARP の詳細については、次を参照してください。 プロキシ ARP とローカル プロキシ ARP （38 ページ）

制約事項

ARP の設定には、次の制約事項が適用されます。

- 逆アドレス解決プロトコル（RARP）はサポートされません。
- 転送情報ベース（FIB）で ARP パケットのレートを制限する ARP スロットリングはサポートされていません。

ARP キャッシュ エントリ

ARP は、ネットワーク アドレス（IP アドレスなど）とイーサネット ハードウェア アドレスの間の通信を確立します。各通信の記録は、キャッシュ内に事前定義された期間だけ保持された後、廃棄されます。

また、明示的に削除されるまで存続するスタティック（永続）エントリを ARP キャッシュに追加することもできます。

スタティック ARP キャッシュ エントリの定義

ARP をはじめとするアドレス解決プロトコルを使用すると、IP アドレスとメディア アドレスとをダイナミックにマッピングできます。大部分のホストはダイナミック アドレス解決をサポートしているため、通常はスタティック ARP 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、グローバルに定義できます。このタスクを実行すると、ARP キャッシュにエントリが永続的にインストールされます。Cisco IOS XR ソフトウェアはこのエントリを使用して、32 ビット IP アドレスを 48 ビットのハードウェア アドレスに変換します。

また、ARP キャッシュにエイリアス エントリを作成することにより、特定の IP アドレスによって識別されたかのように、ARP 要求に応答することもできます。

設定例

キャッシュ エントリを作成して、IP アドレス **203.0.1.2** と MAC アドレス **0010.9400.000c** の間に接続を確立します。さらに、このキャッシュ エントリをエイリアス エントリとして作成すると、エントリに関連付けられているインターフェイスは、エントリ内のデータ リンク層アドレスを使って、このネットワーク層アドレスに対する ARP 要求パケットに応答します。

```
Router#config
Router(config)#arp 203.0.1.2 0010.9400.000c arPA
Router(config)#commit
```

実行コンフィギュレーション

```
Router#show run arp 203.0.1.2 0010.9400.000c arpA
arp vrf default 203.0.1.2 0010.9400.000c ARPA
```

確認

[State] が [Static] になっているかをチェックして、適切に機能していることを確認します。

```
Router#show arp location 0/RP0/CPU0
Address         Age        Hardware Addr   State      Type  Interface
203.0.1.1       -          ea28.5f0b.8024  Interface  ARPA  HundredGigE0/9/0/0
203.0.1.2       -          0010.9400.000c  Static     ARPA  HundredGigE0/9/0/0
```

プロキシ ARP とローカル プロキシ ARP

プロキシ ARP がディセーブルされると、ネットワーキングデバイスは、次のいずれかの条件が満たされる場合に限り、インターフェイスに受信された ARP 要求に応答します。

- ARP 要求のターゲット IP アドレスは、要求が受信されたインターフェイス IP アドレスと同じです。
- ARP 要求のターゲット IP アドレスには、静的に設定された ARP エイリアスがあります。

プロキシ ARP がイネーブルになると、ネットワーキング デバイスは、次の条件すべてを満たす ARP 要求にも応答します。

- ターゲット IP アドレスが、要求を受信した同一の物理ネットワーク（LAN）上にない。
- ネットワーキング デバイスに、ターゲット IP アドレスまでのルートが 1 つ以上存在する。
- ターゲット IP アドレスまでのルートすべてが、要求を受信したインターフェイスとは別のインターフェイスを通過する。

プロキシ ARP がイネーブルになっている場合、ネットワーキング デバイスは、次の条件をすべて満たす ARP 要求に応答します。

- ARP 要求のターゲット IP アドレス、ARP ソースの IP アドレス、および ARP 要求を受信するインターフェイスの IP アドレスが、同じレイヤ 3 ネットワーク上にある。
- ターゲット IP アドレスのネクストホップが、要求を受信するインターフェイスと同じインターフェイスを使用する。

通常、ローカル プロキシ ARP は、同じレイヤ 3 ネットワークで MAC アドレスを IP アドレスに解決するために使用されます。ローカル プロキシ ARP は、ARP でサポートされるあらゆるタイプのインターフェイスに加えて、アンナンバード インターフェイスに対応しています。

プロキシ ARP のイネーブル化

Cisco IOS XR ソフトウェア は（RFC 1027 で定義されている）プロキシ ARP を使用して、ルーティングに必要な情報を持たないホストでも他のネットワークやサブネット上のホストのメディア アドレスを判別できるようにします。たとえば、ARP 要求の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 要求をルータが受信した場合、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 応答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。プロキシARPはデフォルトではディセーブルになっています。このタスクでは、ディセーブルになっているプロキシ ARP をイネーブルにする方法について説明します。

設定例

プロキシ ARP を HundredGigE インターフェイス 0/9/0/0 でイネーブルにします。

```
Router#configure
Router(config)#interface HundredGigE0/9/0/0
Router(config-if)#proxy-arp
Router(config-if)#commit
```

実行コンフィギュレーション

```
Router# show running-config interface HundredGigE0/9/0/0
interface HundredGigE0/9/0/0
 mtu 4000
 ipv4 address 1.0.0.1 255.255.255.0
 proxy-arp
!
!
```

確認

プロキシ ARP が設定され、イネーブルになっていることを確認します。

```
Router#show arp idb interface HundredGigE0/9/0/0 location 0/RP0/CPU0
  interface HundredGigE0/9/0/0 (0x08000038):
  IPv4 address 1.0.0.1, Vrf ID 0x60000000
  VRF Name default
  Dynamic learning: Enable
  Dynamic entry timeout: 14400 secs
  Purge delay: off
  IPv4 caps added (state up)
  MPLS caps not added
  Interface not virtual, not client fwd ref,
  Proxy arp is configured, is enabled
  Local Proxy arp not configured
  Packet IO layer is NetIO
  Srg Role : DEFAULT
  Idb Flag : 262332
  IDB is Complete
```

ローカル プロキシ ARP のイネーブル化

ローカル プロキシ ARP は、レイヤ 2 で分離されたプライベート VLAN など、同じレイヤ 3 ネットワークで MAC アドレスを IP アドレスに解決するために使用されます。ローカル プロキシ ARP は、ARP でサポートされるあらゆるタイプのインターフェイスに加えて、アンナンバード インターフェイスに対応しています。

設定例

ローカル プロキシ ARP を HundredGigE インターフェイス 0/9/0/0 でイネーブルにします。

```
Router#configure
Router(config)#interface HundredGigE0/9/0/0
Router(config-if)#local-proxy-arp
Router(config-if)#commit
```

実行コンフィギュレーション

```
Router#show running-config interface HundredGigE0/9/0/1
 interface HundredGigE0/9/0/0
ipv4 address 1.0.0.1 255.255.255.0
local-proxy-arp
!
```

確認

ローカル プロキシ ARP が設定されていることを確認します。

```
Router#show arp idb interface HundredGigE0/9/0/0 location 0/RP0/CPU0
HundredGigE0/9/0/1 (0x08000038):
  IPv4 address 1.0.0.1, Vrf ID 0x60000000
  VRF Name default
  Dynamic learning: Enable
  Dynamic entry timeout: 14400 secs
  Purge delay: off
  IPv4 caps added (state up)
  MPLS caps not added
  Interface not virtual, not client fwd ref,
  Proxy arp not configured, not enabled
  Local Proxy arp is configured
  Packet IO layer is NetIO
  Srg Role : DEFAULT
  Idb Flag : 264332
  IDB is Complete
```

関連コマンド

- local-proxy-arp
- show arp idb

ローカル ARP エントリの学習の設定

インターフェイスまたはサブインターフェイスを設定して、ローカル サブネットから ARP エントリのみを学習することができます。

インターフェイス上にローカル ARP 学習を設定するには、次の手順を実行します。

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。

   ```
   Router(config)# interface TenGigE 0/11/0/0
   ```

2. インターフェイスの IPv4/IPv6 アドレスを設定します。

   ```
   Router(config-if)# ipv4 address 12.1.3.4 255.255.255.0
   ```

3. インターフェイス上にローカル ARP インターフェイスの学習を設定します。

   ```
   Router(config-if)# arp learning local
   ```

4. インターフェイスを有効にし、設定をコミットします。

   ```
   Router(config-if)# no shut
   Router(config-if)# commit
   RP/0/0/CPU0:Dec 12 13:41:16.580 : ifmgr[397]: %PKT_INFRA-LINK-3-UPDOWN : interface
   TenGigE 0/11/0/0, changed state to Down
   RP/0/0/CPU0:Dec 12 13:41:16.683 : ifmgr[397]: %PKT_INFRA-LINK-3-UPDOWN : interface
   TenGigE 0/11/0/0 changed state to Up
   ```

5. 設定を確認します。

   ```
   Router(config-if)# show running-configuration
   ..
   Building configuration...
   !! IOS XR Configuration 0.0.0
   !! Last configuration change at Mon Dec 12 13:41:16 2016
   !interface TenGigE 0/11/0/0
    ipv4 address 12.1.3.4 255.255.255.0
   ```

```
 arp learning local
!
```

6. ローカル ARP 学習がインターフェイスに設定したとおりに動作していることを確認します。

```
Router(config-if)# do show arp idb TenGigE 0/11/0/0 location 0/RP0/CPU0
Thu Dec 15 10:00:11.733 IST

TenGigE 0/11/0/0 (0x00000040):
  IPv4 address 12.1.3.4, Vrf ID 0x60000000
  VRF Name default
  Dynamic learning: Local
  Dynamic entry timeout: 14400 secs
  Purge delay: off
  IPv4 caps added (state up)
  MPLS caps not added
  Interface not virtual, not client fwd ref,
  Proxy arp not configured, not enabled
  Local Proxy arp not configured
  Packet IO layer is NetIO
  Srg Role : DEFAULT
  Idb Flag : 2146444
  IDB is Complete
```

7. （任意）インターフェイス上で ARP トラフィックをモニタできます。

```
Router(config-if)# do show arp idb TenGigE 0/11/0/0 location 0/RP0/CPU0
Thu Dec 15 10:13:28.964 IST

ARP statistics:
  Recv: 0 requests, 0 replies
  Sent: 0 requests, 1 replies (0 proxy, 0 local proxy, 1 gratuitous)
  Subscriber Interface:
         0 requests recv, 0 replies sent, 0 gratuitous replies sent
  Resolve requests rcvd: 0
  Resolve requests dropped: 0
  Errors: 0 out of memory, 0 no buffers, 0 out of sunbet

ARP cache:
  Total ARP entries in cache: 1
  Dynamic: 0, Interface: 1, Standby: 0
  Alias: 0,   Static: 0,    DHCP: 0

  IP Packet drop count for GigabitEthernet0_0_0_1: 0
```

ARP の設定に関する情報

アドレス解決の概要

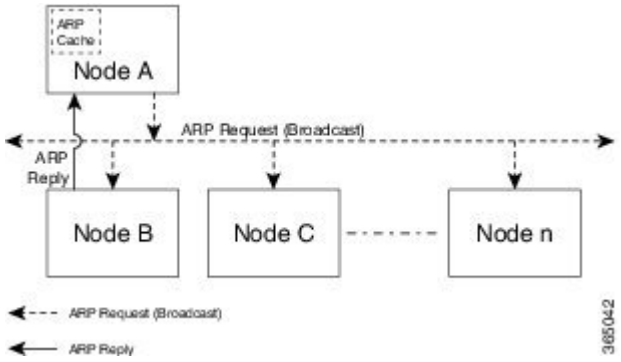
IP のデバイスは、ローカル アドレス（ローカル セグメントまたは LAN のデバイスを一意に識別）とネットワーク アドレス（デバイスが属するネットワークを識別）の両方を持つことができます。ローカル アドレスは、より正確にはデータリンク アドレスとして知られています。その理由は、ローカル アドレスはパケット ヘッダーのデータリンク層（OSI モデルの第 2 層）の部分にあり、データリンク デバイス（ブリッジやすべてのデバイス インターフェイスなど）

によって読み取られるからです。データリンク層内の MAC 副層がその層用にアドレスを処理するため、技術志向が強い人ほどローカル アドレスを *MAC* アドレスと呼びます。

たとえば、イーサネットでデバイスと通信するには、Cisco IOS XR ソフトウェアがまずそのデバイスの 48 ビットの MAC アドレスまたはローカル データリンク アドレスを特定する必要があります。IP アドレスからローカル データリンク アドレスを決定する処理は、アドレス解決と呼ばれています。

単一の LAN でのアドレス解決

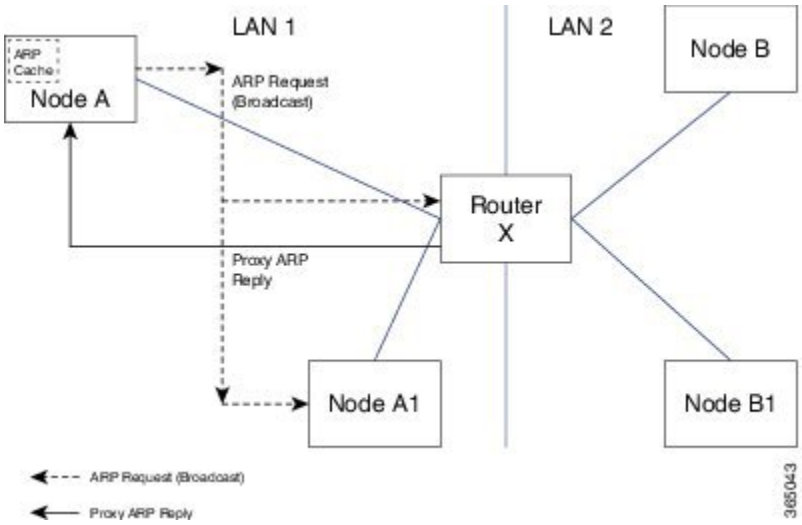
次のプロセスでは、送信元デバイスと宛先デバイスが同じ LAN に接続されている場合のアドレス解決について説明します。



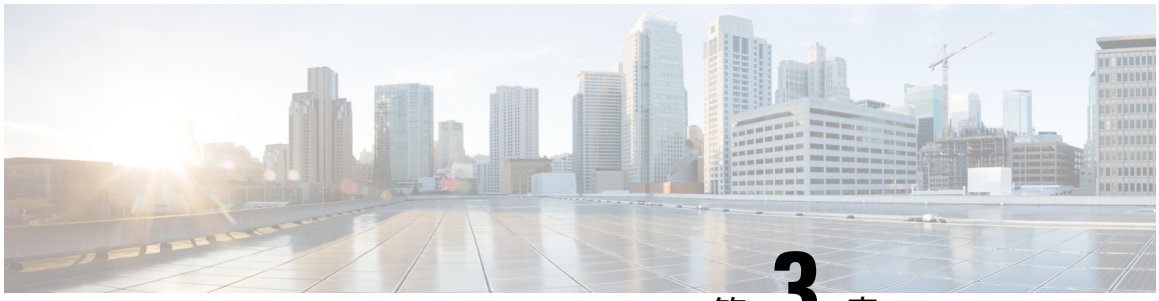
1. エンド システム A（ノード A）は ARP 要求を LAN にブロードキャストし、エンド システム B（ノード B）の MAC アドレスの学習を試みます。
2. ブロードキャストは、エンド システム B を含め LAN 上のすべてのデバイスで受信され、処理されます。
3. エンド システム B のみが、ARP 要求に応答します。ARP 応答に自身の MAC アドレスを含めてエンド システム A（ノード A）に送信します。
4. エンド システム A（ノード A）は応答を受信し、自身の ARP キャッシュにエンド システム B の MAC アドレスを保存します（ARP キャッシュ内で、ネットワーク アドレスが MAC アドレスに関連付けられます）。
5. エンド システム A（ノード A）はエンド システム B との通信が必要になるたびに、ARP キャッシュをチェックしてエンド システム B の MAC アドレスを探し、フレームを直接送信します。最初に ARP 要求を使用する必要はありません。

ルータによって相互接続されている場合のアドレス解決

次のプロセスでは、送信元デバイスと宛先デバイスが、ルータによって相互接続された異なる LAN に接続されている場合のアドレス解決について説明します（プロキシ ARP が有効になっている場合のみ）。



1. エンドシステムY（ノードA）はARP要求をLANにブロードキャストし、エンドシステムZ（ノードB）のMACアドレスの学習を試みます。
2. ブロードキャストは、ルータXを含めLAN上のすべてのデバイスで受信され、処理されます。
3. ルータXは、自身のルーティングテーブルをチェックし、エンドシステムZ（ノードB）が別のLAN上にあることを突き止めます。
4. これにより、ルータXはエンドシステムZ（ノードB）のプロキシとして動作します。ルータXはエンドシステムZ（ノードB）に属しているかのように、エンドシステムY（ノードA）からのARP要求に応答し、ARP応答に自身のMACアドレスを含めて送信します。
5. エンドシステムY（ノードA）はARP応答を受信し、自身のARPキャッシュのエンドシステムZ（ノードB）用エントリにルータXのMACアドレスを保存します。
6. エンドシステムY（ノードA）はエンドシステムZ（ノードB）との通信が必要になると、ARPキャッシュをチェックしてルータXのMACアドレスを探し、フレームを直接送信します。ARP要求は使用されません。
7. ルータXは、エンドシステムY（ノードA）からトラフィックを受信し、それを他のLAN上のエンドシステムZ（ノードB）に転送します。



第 3 章

DHCP リレーの概要

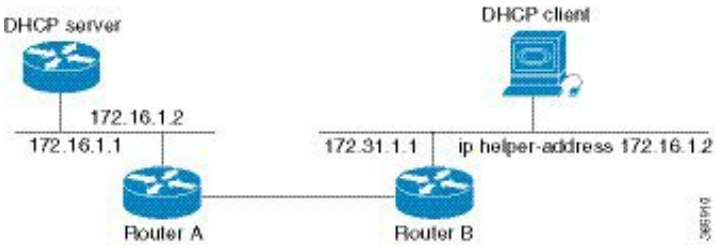
DHCP リレーエージェントは、共有の物理サブネットに存在しないクライアントとサーバとの間で DHCP パケットを転送するホストです。リレー エージェント転送は、IP ルータの通常の転送とは異なります。通常の転送では、IPデータグラムがネットワーク間で透過的にスイッチングされます。

DHCP クライアントは、自身の所属先のネットワークに関する情報を保持していないときには、ユーザデータグラムプロトコル（UDP）ブロードキャストを使用して、DHCPDISCOVER メッセージを送信します。

サーバが含まれていないネットワーク セグメント上にクライアントがある場合、DHCP パケットが別のネットワーク セグメント上のサーバに届くようにするには、そのネットワーク セグメントにリレー エージェントが必要です。ほとんどのルータはブロードキャスト トラフィックを転送するように設定されていないため、UDP ブロードキャスト パケットは転送されません。DHCP リレー プロファイルを設定することにより DHCP パケットをリモート サーバに転送するように DHCP リレー エージェントを設定し、そこに 1 つ以上のヘルパー アドレスを設定できます。プロファイルをインターフェイスまたは VRF に割り当てることができます。

次の図に、このプロセスを示します。DHCPクライアントが、IPアドレスの要求と追加設定パラメータをローカル LAN 上でブロードキャストしています。DHCP リレー エージェントとして機能するルータ B は、ブロードキャストを取得し、宛先アドレスを DHCP サーバのアドレスに変更し、別のインターフェイスにメッセージを送信します。リレーエージェントは、DHCP クライアントのパケットを受け取ったインターフェイスのIPアドレスをDHCPパケットのゲートウェイ アドレス（giaddr）フィールドに挿入します。これにより、DHCP サーバは、どのサブネットがオファーを受信するかを判断し、適切な IP アドレス範囲を特定できます。リレーエージェントは、メッセージを（リレー プロファイルのヘルパー アドレスによって指定される）サーバ アドレス、この場合は 172.16.1.2 にユニキャストします。

図 *10 :* ヘルパー アドレスを使用した *UDP* ブロードキャストの *DHCP* サーバへの転送



• DHCP リレー エージェントの設定およびイネーブル化 （46 ページ）

DHCP リレー エージェントの設定およびイネーブル化

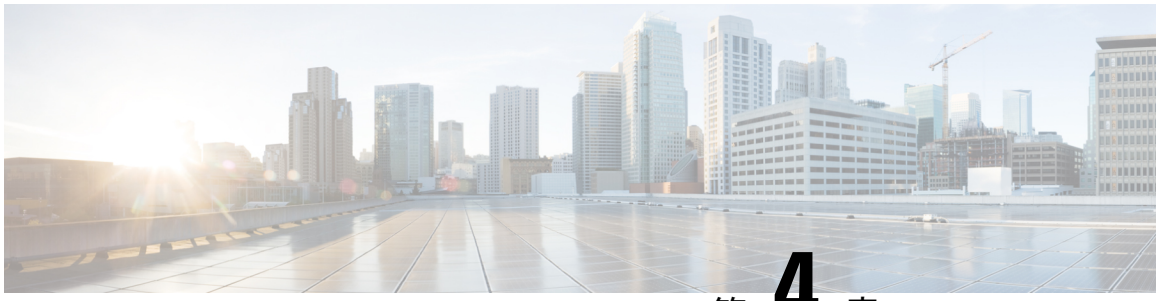
設定例

```
RP/0/RP0/CPU0:router# configure
/* Enters the global configuration mode */

RP/0/RP0/CPU0:router(config)# dhcp ipv4
/* Configures DHCP for IPv4 and enters the DHCPv4 configuration submode. */

RP/0/RP0/CPU0:router(config-dhcpv4)# profile r1 relay
/* Enables DHCP relay profile */

RP/0/RP0/CPU0:router(config-dhcpv4-relay-profile)# helper-address vrf A 10.10.10.1 giaddr
 40.1.1.2
RP/0/RP0/CPU0:router(config-dhcpv4-relay-profile)# broadcast-flag policy check
/* Configures VRF addresses for forwarding UDP broadcasts, including DHCP. */

RP/0/RP0/CPU0:router(config-dhcpv4-relay-profile)# relay information option vpn
RP/0/RP0/CPU0:router(config-dhcpv4-relay-profile)# relay information option vpn-mode rfc
/* Inserts the DHCP relay agent information option (option-82 field) in forwarded
BOOTREQUEST messages to a DHCP server. */


RP/0/RP0/CPU0:router(config-dhcpv4-relay-profile)# relay information option allow-untrusted
/* (Optional) Configures the DHCP IPv4 Relay not to discard BOOTREQUEST packets
that have an existing relay information option and the giaddr set to zero. */

RP/0/RP0/CPU0:router(config-dhcpv4-relay-profile)# exit
RP/0/RP0/CPU0:router(config-dhcpv4)# interface BVI 1 relay profile r1
RP/0/RP0/CPU0:router(config-dhcpv4)# commit
/* Configures DHCP relay on a BVI interface and commits the configuration */
```

実行コンフィギュレーション

```
RP/0/RP0/CPU0:router#show running-config
Tue May 23 10:56:14.463 IST
Building configuration...
!! IOS XR Configuration 0.0.0
!! Last configuration change at Tue May 23 10:56:08 2017 by annseque
!
dhcp ipv4
 vrf vrf1 relay profile client
 profile r1 relay
```

```
 helper-address vrf A 10.10.10.1 giaddr 40.1.1.2
 broadcast-flag policy check
 relay information option vpn
 relay information option vpn-mode rfc
 relay information option allow-untrusted
!
```



第 4 章

DHCPv6 リレー エージェントの実装

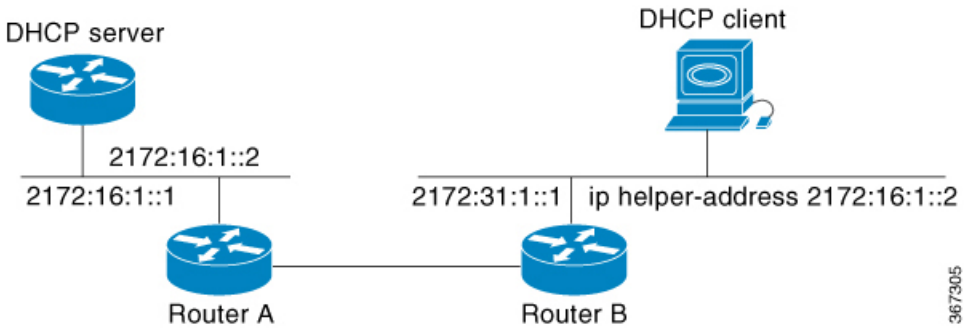
DHCPv6 リレーエージェントは、共有の物理サブネット上に存在しないクライアントとサーバの間で DHCPv6 パケットを転送するホストです。リレー エージェントの転送は、ネットワーク間で IP パケットが透過的に交換される通常の IP ルータの転送とは異なります。

DHCPv6 クライアントは、自身の所属先のネットワークに関する情報がないときに、ユーザ データグラム プロトコル（UDP）ブロードキャストを使用して DHCP DISCOVER メッセージを送信します。

サーバが含まれていないネットワーク セグメント上にクライアントがある場合、DHCPv6 パケットが別のネットワーク セグメント上のサーバに届くようにするには、そのネットワーク セグメントにリレー エージェントが必要です。ほとんどのルータはブロードキャスト トラフィックを転送するように設定されていないため、UDP ブロードキャスト パケットは転送されません。DHCPv6 リレープロファイルを設定することで、DHCPv6 パケットをリモート サーバに転送するように DHCPv6 リレー エージェントを設定し、そこに 1 つ以上のヘルパー アドレスを設定できます。プロファイルをインターフェイスまたは VRF に割り当てることができます。

次の図に、このプロセスを示します。DHCPv6 クライアントが、IPv6 アドレスの要求と追加設定パラメータをローカル LAN 上でブロードキャストしています。DHCPv6 リレー エージェントとして機能するルータ B は、ブロードキャストを取得して宛先アドレスを DHCPv6 サーバのアドレスに変更し、別のインターフェイスにメッセージを送信します。リレーエージェントは、DHCPv6 クライアントのパケットを受け取ったインターフェイスの IPv6 アドレスを DHCPv6 パケットのゲートウェイ アドレス（giaddr）フィールドに挿入します。これにより、DHCPv6 サーバは、どのサブネットがオファーを受信するかを判断し、適切な IPv6 アドレス範囲を特定できます。リレー エージェントは、リレー プロファイルのヘルパー アドレスによって指定されるサーバ アドレス（この場合は 2172:16:1::2）にメッセージをユニキャストします。

図 *11* : ヘルパー アドレスを使用した *UDP* ブロードキャストの *DHCPv6* サーバへの転送



DHCPv6 リレー エージェントの設定の前提条件

DHCPv6 リレー エージェントを設定する際の前提条件は次のとおりです。

- 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。
- DHCPv6 クライアントおよび DHCPv6 サーバが設定済みで動作している。
- リレー エージェントと DHCPv6 サーバが接続されている。

DHCPv6 リレー機能の制限事項

DHCPv6 リレー機能の実装に関する制限事項は次のとおりです。

- マルチキャスト アドレスはサポートされていません。DHCPv6 リレー プロファイル サブモードの **helper-address** コマンドは、ヘルパー アドレスとしてグローバル ユニキャスト IPv6 アドレスのみをサポートします。

- パケットを DHCPv6 サーバに転送するときにリレー エージェントによって追加される DHCPv6 オプション コードは、interface-id と remote-id のみです。

（注） DHCPv6 リレープロファイル サブモードでは、DHCPv6 オプション コードの設定はサポートされていません。

DHCPv6 リレー エージェントを設定およびイネーブルにする方法

ここでは、次のタスクについて説明します。

DHCPv6 リレー エージェントの設定およびイネーブル化

このタスクでは、DHCPv6 リレーエージェントを設定してイネーブルにする方法について説明します。

```
RP/0/RSP0/CPU0:router# configure terminal
RP/0/RSP0/CPU0:router(config)# dhcp ipv6
RP/0/RSP0/CPU0:router(config-dhcpv6)# commit
```

DHCPv6（ステートレス）リレー エージェントの設定

クライアント メッセージの転送先のアドレスを指定し、インターフェイスで IPv6 リレー サービス用にダイナミック ホスト コンフィギュレーション プロトコル（DHCP）をイネーブルにするには、このタスクを実行します。

```
RP/0/RP0/CPU0:router# configure terminal
RP/0/RP0/CPU0:router(config)# dhcp ipv6
RP/0/RP0/CPU0:router(config-dhcpv6)# interface type interface-path-id relay
RP/0/RP0/CPU0:router(config-dhcpv6-if)# destination ipv6-address
RP/0/RP0/CPU0:router(config-dhcpv6-if)# commit
```

インターフェイスでの DHCPv6 リレー エージェントのイネーブル化

このタスクでは、インターフェイスで Cisco IOS XR DHCPv6 リレー エージェントをイネーブルにする方法について説明します。

（注） Cisco IOS XR ソフトウェアでは、DHCPv6 リレー エージェントがデフォルトでディセーブルになっています。

```
RP/0/RP0/CPU0:router# configure terminal
RP/0/RP0/CPU0:router(config)# dhcp ipv6
```

```
RP/0/RP0/CPU0:router(config-dhcpv6)# interface type interface-instance relay profile
profile-name
RP/0/RP0/CPU0:router(config-dhcpv6-if)# commit
```

インターフェイスでの DHCPv6 リレーのディセーブル化

このタスクでは、インターフェイスにプロファイルを割り当てないことにより、インターフェイスで DHCPv6 リレーをディセーブルにする方法について説明します。

```
RP/0/RP0/CPU0:router# configure terminal
RP/0/RP0/CPU0:router(config)# dhcp ipv6
RP/0/RP0/CPU0:router(config-dhcpv6)# interface type name none
RP/0/RP0/CPU0:router(config-dhcpv6-if)# commit
```

VRF での DHCPv6 リレーのイネーブル化

このタスクでは、VRF で DHCPv6 リレーをイネーブルにする方法について説明します。

```
RP/0/RP0/CPU0:router# configure terminal
RP/0/RP0/CPU0:router(config)# dhcp ipv6
RP/0/RP0/CPU0:router(config-dhcpv6)# vrf vrf-name relay profile profile-name
RP/0/RP0/CPU0:router(config-dhcpv6-if)# commit
```

複数のヘルパー アドレスを使用した DHCPv6 リレー プロファイルの設定

DHCPv6 リレー プロファイルには、最大 16 のヘルパー IPv6 アドレスを設定できます。

1. DHCPv6 コンフィギュレーション モードを開始します。

   ```
   RP/0/RP0/CPU0:router(config)# dhcp ipv6
   ```

2. DHCPv6 リレー プロファイルを設定します。

   ```
   RP/0/RP0/CPU0:router(config-dhcpv6)# profile helper relay
   ```

3. ヘルパー アドレスを設定します。

（注） 最大 16 の IPv6 アドレスを設定できます。

   ```
   RP/0/RP0/CPU0:router(config-dhcpv6-relay-profile)# helper-address vrf default
   2001:1:1::2
   ```

4. 設定を確認します。

   ```
   RP/0/RP0/CPU0:router(config-dhcpv6-relay-profile)# show configuration

   !! IOS XR Configuration 0.0.0
   dhcp ipv6
    profile helper relay
     helper-address vrf default 2001:1:1::2
   ```

```
 !
!
end
```

5. 設定をコミットします。

```
RP/0/RP0/CPU0:router(config-dhcpv6-relay-profile)# commit
```

6. コンフィギュレーション モードを終了し、設定されているヘルパー アドレスを確認します。

```
RP/0/RP0/CPU0:router# show dhcp ipv6 relay profile name helper
...
Profile: helper
Helper Addresses:
        2001:1:1::2, vrf default
Information Option: Disabled
Information Option Allow Untrusted: Disabled
Information Option VPN: Disabled
Information Option VPN Mode: RFC
Information Option Policy: Replace
Information Option Check: Disabled
GIADDR Policy: Keep
Broadcast-flag Policy: Ignore
VRF References:
Interface References:
```

DHCPv6 リレー ヘルパー アドレスが正常に設定されています。

システム永続メモリへの DHCPv6 リレー バインディング データベースの書き込みの設定

システム永続メモリへの DHCPv6 リレー バインディング データベースの書き込みを設定するには、次のタスクを実行します。これは、システムのリロード後に DHCPv6 リレーバインディングテーブルを回復するのに役立ちます。完全な永続ファイルの書き込みに使用されるファイル名は、dhcpv6_srpb_{nodeid}_odd および dhcpv6_srpb_{nodeid}_even です。nodeid は、ファイルが書き込まれるノードの実際のノード ID です。増分ファイルは完全なファイルと同じ方法で命名され、_inc が付加されます。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# dhcp ipv6
RP/0/RP0/CPU0:router(config-dhcpv6)# database [relay] [full-write-interval
full-write-interval] [incremental-write-interval incremental-write-interval]
RP/0/RP0/CPU0:router(config-dhcpv6)# commit
```

DHCPv6 サーバ

DHCPv6 サーバはアドレス割り当て要求と更新を受け取り、分散アドレス プール（DAPS）内にある定義済みアドレス グループから IP アドレスを割り当てます。サブネット マスク、ドメイン名、DNS サーバの IP アドレス、デフォルト ルータ、その他の設定パラメータなどの詳細情報を要求元クライアントに提供するように DHCPv6 サーバを設定することもできます。

DHCPv6サーバは、ローカル接続されているLANセグメントからのブロードキャストや、ネットワークにある他のDHCPv6リレーエージェントから転送されたDHCPv6要求のブロードキャストを受け入れることができます。

DHCPv6プロキシはすべてのリレー機能を実行し、さらにいくつかの追加機能も提供します。DHCPv6プロキシにより、DHCPv6クライアントはDHCPv6サーバの詳細を確認できません。DHCPv6プロキシは、クライアントがプロキシをサーバと見なすようにDHCPv6応答を変更します。この状態では、クライアントはDHCPv6サーバに対するようにプロキシと相互作用します。

DHCP IPv6 サービスベース モードの選択

DHCP IPv6サービスベース モードの選択機能の一部として、DHCPベースという新しいモードが導入されました。DHCPベース モードでインターフェイスが設定されている場合、DHCPはクライアント要求のオプション60（class-identifier）値をDHCP ベース プロファイルで設定された値と照合することによって、DHCPv6 プロキシまたはDHCPv6 サーバ モードのいずれかを選択してクライアント要求を処理します。

プールは、サーバプロファイル モードおよびサーバプロファイルクラス サブモードで設定されます。クラスベースのプールの選択は、常にプロファイル プールの選択よりも優先されます。

DHCPv6 サーバプロファイルクラス サブモードは、一部（0、12、50、52、53、54、58、59、61、82、および255）を除くDHCPv6 オプションの設定をサポートします。

```
dhcp ipv6
profile DHCP_BASE base
  match option 60 41424344 profile DHCPv6_PROXY proxy
  match option 60 41424355 profile DHCPv6_SERVER server
  default profile DEFAULT_PROFILE server
  relay information authenticate inserted
   !
profile DHCPv6_PROXY proxy
  helper-address vrf default 10.10.10.1 giaddr 0.0.0.0
!
profile DHCPv6_SERVER server
  lease 1 0 0
  pool IP_POOL
!
profile DEFAULT_PROFILE server
  lease 1 0 0
  pool IP_POOL
!
!
interface TenGigE 0/11/0/0 base profile DHCP_BASE
```

DHCPv6 サーバ プロファイルの設定

DHCPv6 サーバ プロファイルを設定するには、次のタスクを実行します。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# dhcp ipv6
RP/0/RP0/CPU0:router(config-dhcpv6)# profile profile-name server
RP/0/RP0/CPU0:router(config-dhcpv6-server-profile)# bootfile boot-file-name
```

```
RP/0/RP0/CPU0:router(config-dhcpv6-server-profile)# broadcast-flag policy unicast-always
RP/0/RP0/CPU0:router(config-dhcpv6-server-profile)# class class-name
RP/0/RP0/CPU0:router(config-dhcpv6-server-profile-class)# exit
RP/0/RP0/CPU0:router(config-dhcpv6-server-profile)# default-router address1 address2 ...
 address8
RP/0/RP0/CPU0:router(config-dhcpv6-server-profile)# lease {infinite | days minutes seconds
 }
RP/0/RP0/CPU0:router(config-dhcpv6-server-profile)# limit lease {per-circuit-id |
per-interface| per-remote-id} value
RP/0/RP0/CPU0:router(config-dhcpv6-server-profile)# netbios-name server address1 address2
 ... address8
RP/0/RP0/CPU0:router(config-dhcpv6-server-profile)# netbios-node-type {number
|b-node|h-node |m-node |p-node}
RP/0/RP0/CPU0:router(config-dhcpv6-server-profile)# option option-code {ascii string |
hex string |ip address}
RP/0/RP0/CPU0:router(config-dhcpv6-server-profile)# pool pool-name
RP/0/RP0/CPU0:router(config-dhcpv6-server-profile)# requested-ip-address-check disable
RP/0/RP0/CPU0:router(config-dhcpv6-server-profile)# commit
```

プールを使用した複数のクラスの設定

複数のクラスにプールを設定するには、次のタスクを実行します。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# dhcp ipv6
RP/0/RP0/CPU0:router(config-dhcpv6)# profile profile-name server
RP/0/RP0/CPU0:router(config-dhcpv6-server-profile)# pool pool-name
RP/0/RP0/CPU0:router(config-dhcpv6-server-profile)# class class-name
RP/0/RP0/CPU0:router(config-dhcpv6-server-class)# pool pool-name
RP/0/RP0/CPU0:router(config-dhcpv6-server-class)# match option option [ sub-option
sub-option] [ ascii asciiString | hex hexString ]
RP/0/RP0/CPU0:router(config-dhcpv6-server-class)# exit
RP/0/RP0/CPU0:router(config-dhcpv6-server-profile)# class class-name
RP/0/RP0/CPU0:router(config-dhcpv6-server-class)# pool pool-name
RP/0/RP0/CPU0:router(config-dhcpv6-server-class)# match vrf vrf-name
RP/0/RP0/CPU0:router(config-dhcpv6-server-class)# commit
```

DHCPv6 クライアント

Dynamic Host Configuration Protocol（DHCP）クライアント機能を使用すると、ルータ インターフェイスで DHCP を使用して IPv6 アドレスを動的に取得できるようになります。

DHCP はインターネット ホストに設定パラメータを提供します。DHCP は 2 つのコンポーネントで構成されています。

- ホスト固有の設定パラメータを DHCP サーバからホストに伝送するプロトコル。
- ホストにネットワーク アドレスを割り当てるメカニズム。

DHCP はクライアント/サーバ モデルに基づいています。指定された DHCP サーバ ホストは、動的に設定されたホストにネットワーク アドレスを割り当て、設定パラメータを提供します。

クライアントとサーバが同じレイヤ2ネットワーク上にない場合は、リレー エージェントが必要です。通常、リレー エージェントはルータ上で実行されます。クライアント デバイスが最

初は自身の IP アドレスを認識していないため、リレー エージェントが必要です。エージェントは、この情報を持つサーバを見つけるためにレイヤ2ブロードキャストを送信します。ルータはこれらのブロードキャストをDHCPv6サーバにリレーし、正しいレイヤ2アドレスに応答を転送して、適切なデバイスが正確な設定情報を取得できるようにします。

DHCP は、リース期間と呼ばれる設定可能な期間にのみ IP アドレスを割り当てることができます。クライアントがこの IP アドレスをリース期間を超えて保持する必要がある場合は、IP アドレスが期限切れになる前にリース期間を更新する必要があります。クライアントは、サーバから送信された設定に基づいてリースを更新します。クライアントは、サーバの IP アドレスを使用して REQUEST メッセージをユニキャストします。サーバは REQUEST メッセージを受信すると、ACK メッセージで応答します。クライアントのリース期間が、ACK メッセージに設定されたリース時間で延長されます。

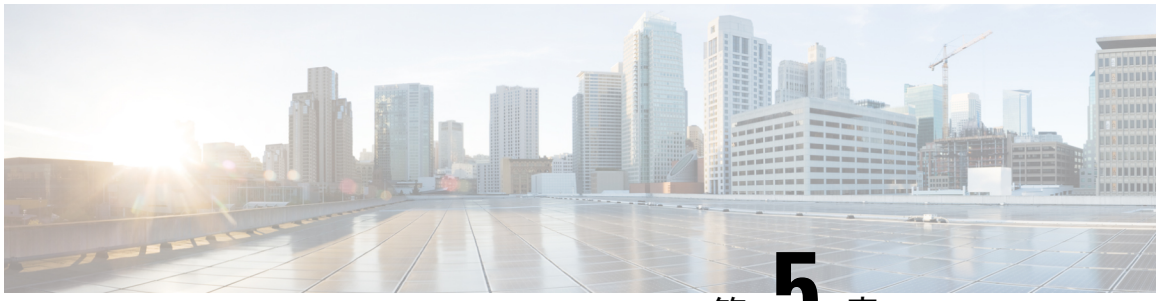
制約事項と制限

- DHCPv6 クライアントは、管理インターフェイスでのみイネーブルにできます。
- インターフェイスで DHCPv6 またはスタティック IPv6 のいずれかを設定できます。

インターフェイスでの DHCPv6 クライアントのイネーブル化

DHCPv6 クライアントは、インターフェイス レベルでイネーブルにできます。インターフェイスで DHCPv6 がイネーブル化またはディセーブル化されると、DHCP コンポーネントが通知を受信します。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface MgmtEth rack/slot/CPU0/port
RP/0/RP0/CPU0:router(config)# interface interface_name ipv6 address dhcp
```



第 5 章

ホスト サービスとアプリケーションの実装

ホスト サービスとアプリケーションの実装

ルータ上の Cisco IOS XR ソフトウェア ホスト サービスおよびアプリケーション機能は主に、ネットワークの接続性とパケットが宛先に達するまでのルートをチェックし、ホスト名を IP アドレスに（または IP アドレスをホスト名に）マッピングして、ルータと UNIX ワークステーション間でファイルを転送するために使用します。

ネットワーク接続性ツール

ネットワーク接続性ツールを使用すると、ネットワーク上のデバイスに対して traceroute や ping を実行して、デバイスの接続をチェックできます。

ping

ping コマンドは、デバイスのアクセシビリティのトラブルシューティングに広く使用されている方法です。これは、2 つのインターネット制御メッセージプロトコル（ICMP）クエリーメッセージ、ICMP エコー要求、および ICMP エコー応答を使用して、リモート ホストがアクティ

ブであるかどうかを判断します。pingコマンドは、エコー応答を受信するまでにかかる時間も測定します。

pingコマンドは、最初に1つのアドレスにエコー要求パケットを送信して応答を待ちます。pingが正常に完了するのは、エコー要求が宛先に届き、定義済みの時間内に宛先がpingの送信元にエコー応答（ホスト名が存続している）を返すことができる場合だけです。

bulkオプションが導入されたため、複数の宛先の到達可能性をチェックできるようになりました。宛先は、CLIから直接入力します。このオプションは、ipv4の宛先でのみサポートされます。

ネットワーク接続の確認

基本的なネットワーク接続性の診断を支援する手段として、多くのネットワークプロトコルがエコープロトコルをサポートしています。プロトコルでは、宛先ホストに特殊なデータグラムを送信し、そのホストからの応答データグラムを待ちます。このエコープロトコルからの結果は、ホストに至るパスの信頼性、パスの遅延、およびホストに到達できるのか、ホストが機能しているのかを評価するのに役立ちます。

ネットワーク接続を確認するための設定

次の設定は、ルータAのインターフェイスとルータBのインターフェイスから送信される拡張pingコマンドを示しています。このpingが成功する場合、ルーティング上の問題がないことを示します。ルータAはルータBのインターフェイスに到達する方法を認識していて、ルータBはルータAのインターフェイスに到達する方法を認識しています。また、両方のホストには適切に設定されたデフォルト ゲートウェイがあります。

ルータAからの拡張 **ping** コマンドが失敗する場合、ルーティング上の問題があることを意味します。3つのルータのいずれでもルーティングの問題が発生する可能性があります。ルータAでは、ルータBのインターフェイスのサブネットへのルートや、ルータCとルータB間のサブネットへのルートが不明になる可能性があります。ルータBでは、ルータAのサブネットへのルートや、ルータCとルータA間のサブネットへのルートが不明になる可能性があります。ルータCでは、ルータAまたはルータBのイーサネット セグメントのサブネットへのルートが不明になる可能性があります。ルーティングに関する問題を修正してから、ホスト1からホスト2へのpingを実行する必要があります。ホスト1からホスト2へのpingを実行できない場合は、両方のホストのデフォルト ゲートウェイを確認してください。ルータAのインターフェイスとルータBのインターフェイスとの接続は、拡張pingコマンドを使用してチェックします。

ルータAからルータBのインターフェイスへの通常のpingでは、pingパケットの送信元アドレスは発信インターフェイスのアドレス、つまりインターフェイスのアドレス（10.0.0.2）になります。ルータBがpingパケットに応答するとき、送信元アドレス（つまり、10.0.0.2）に応答します。このように、ルータAのインターフェイス（10.0.0.2）とルータBのTenGigEインターフェイス（10.0.0.1）間の接続だけがテストされます。

ルータAのインターフェイス（10.0.0.2）とルータBのインターフェイス（10.0.0.1）との接続をテストするには、拡張pingコマンドを使用します。拡張 **ping** コマンドには、**ping** パケットの送信元アドレスを指定するオプションがあります。

設定例

この使用例では、拡張 ping コマンドを使用して、2 つの IP アドレス（ルータ A（10.0.0.2）とルータ B（10.0.0.1））間の IP 接続を検証します。

```
Router# ping 10.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5)
Router#!!!!!

*/If you do not enter a hostname or an IP address on the same line as the ping command,

the system prompts you to specify the target IP address and several other command
parameters.
After specifying the target IP address, you can specify alternate values for the
remaining parameters or accept the displayed default for each parameter /*

Router# ping
Protocol [ipv4]:
Target IP address: 10.0.0.1
Repeat count [5]: 5
Datagram size [100]: 1000
Timeout in seconds [2]: 1
Interval in milliseconds [10]: 1
Extended commands? [no]: no
Sweep range of sizes? [no]:
Type escape sequence to abort.
Sending 5, 1000-byte ICMP Echos to 10.0.0.1, timeout is 1 seconds:
!!!!!
Success rate is 100 percent (5/5)
Router#!!!!!
```

関連コマンド

複数の宛先に対するネットワーク接続性のチェック

bulk オプションを使用すると、複数の宛先への到達可能性をチェックできます。宛先は、CLI から直接入力します。このオプションは、ipv4 の宛先でのみサポートされます。

設定例

次の IP アドレスを持つ IP ネットワーク上の複数のホストへの到達可能性とネットワーク接続を確認します。

- 1: 1.1.1.1
- 2: 2.2.2.2
- 3: 3.3.3.3

```
Router# ping bulk ipv4 input cli batch
*/You must hit the Enter button and then specify one destination address per line*/
Please enter input via CLI with one destination per line and when done Ctrl-D/(exit) to
 initiate pings:
1: 1.1.1.1
2: 2.2.2.2
3: 3.3.3.3
4:
```

```
Starting pings...
Target IP address: 1.1.1.1
Repeat count [5]: 5
Datagram size [100]: 1
% A decimal number between 36 and 18024.
Datagram size [100]: 1
% A decimal number between 36 and 18024.
Datagram size [100]: 1000
Timeout in seconds [2]: 1
Interval in milliseconds [10]: 10
Extended commands? [no]: no
Sweep range of sizes? [no]: q
% Please answer 'yes' or 'no'.
Sweep range of sizes? [no]: q
% Please answer 'yes' or 'no'.
Sweep range of sizes? [no]:
Type escape sequence to abort.
Sending 5, 1000-byte ICMP Echos to 1.1.1.1, vrf is default, timeout is 1 seconds:
!!!!!
Success rate is 100 percent (5/5),
Target IP address: 2.2.2.2
Repeat count [5]:
Datagram size [100]: q
% A decimal number between 36 and 18024.
Datagram size [100]:
Timeout in seconds [2]:
Interval in milliseconds [10]:
Extended commands? [no]:
Sweep range of sizes? [no]:
Sending 5, 100-byte ICMP Echos to 1.1.1.1, vrf is default, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5),
Target IP address:  3.3.3.3
Repeat count [5]: 4
Datagram size [100]: 100
Timeout in seconds [2]: 1
Interval in milliseconds [10]: 10
Extended commands? [no]: no
Sweep range of sizes? [no]: no
Sending 4, 100-byte ICMP Echos to 1.1.1.1, vrf is default, timeout is 1 seconds:
!!!!!
Success rate is 100 percent (4/5),
```

関連コマンド

traceroute

ping コマンドを使用してデバイス間の接続性を検証できる場合は、**traceroute** コマンドを使用してパケットがリモート接続先までにたどるパスおよびルーティングに障害がある場所を検出できます。

traceroute コマンドは、各 ICMP "time-exceeded" メッセージの送信元を記録して、パケットが宛先に達するまでにたどったパスを示すことができます。IP **traceroute** コマンドを使用すると、パケットがネットワーク経由でたどるパスをホップバイホップで特定できます。このコマンドを実行すると、トラフィックが宛先に到達するまでに通過するルータなどのすべてのネットワーク層（レイヤ 3）デバイスが表示されます。

traceroute コマンドは、IP ヘッダーの存続可能時間（TTL）フィールドを使用して、ルータとサーバで特定のリターン メッセージが生成されるようにします。traceroute コマンドは、TTL フィールドが 1 に設定されている宛先ホストに、ユーザ データグラム プロトコル（UDP）データグラムを送信します。ルータは 1 または 0 の TTL 値を検出すると、データグラムをドロップし、送信元に ICMP の time-exceeded メッセージを戻します。traceroute 機能は、ICMP time-exceeded メッセージの送信元アドレス フィールドを調べ、最初のホップのアドレスを判別します。

ネクスト ホップを識別するために、traceroute コマンドは TTL 値が 2 の UDP パケットを送信します。1 番めのルータは、TTL フィールドの値から 1 を差し引いて次のルータにデータグラムを送信します。2 番めのルータは、TTL 値が 1 の UDP パケットを受け取り、データグラムを廃棄して、送信元に time-exceeded メッセージを戻します。このように、データグラムが宛先ホストに到達するまで（または TTL の最大値に達するまで）TTL の値は増分され、処理が続けられます。

データグラムが宛先に到達したことを判断するために、traceroute コマンドは、宛先ホストが使用しないと予測される非常に大きな値をデータグラムの UDP 宛先ポートに設定します。ホストは、この未知のポート番号を持つデータグラムを受信すると、送信元に ICMP port unreachable error メッセージを戻します。このメッセージにより、宛先に到達したことを traceroute 機能に伝えます。

パケット ルートのチェック

traceroute コマンドを使用すると、パケットが宛先に到達するまでに実際にたどるルートをトレースできます。

設定例

10.0.0.2 から 20.1.1.1 へのルートをトレースします。

```
Router# traceroute 20.1.1.1
Type escape sequence to abort.
Tracing the route to 20.1.1.1
 1  10.0.0.1 39 msec  *  3 msec

*/If you do not enter a hostname or an IP address on the same line as the traceroute
command, the system prompts you to specify the target IP address and several other
command parameters. After specifying the target IP address, you can specify alternate
values for the remaining parameters or accept the displayed default for each
parameter/*

Router #traceroute
Protocol [ipv4]:
Target IP address: 20.1.1.1
Source address: 10.0.0.2
Numeric display? [no]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]:

Type escape sequence to abort.
```

```
Tracing the route to 20.1.1.1
 1  10.0.0.1 3 msec  *  3 msec
```

関連コマンド

ドメイン サービス

Cisco IOS XR ソフトウェア ドメイン サービスは、Berkeley Standard Distribution（BSD）ドメイン リゾルバとして機能します。ドメイン サービスは、アプリケーション（Telnet など）やコマンド（**ping**、**traceroute** など）で使用される、ホスト名対アドレスのマッピングのローカル キャッシュを保持します。ローカルキャッシュにより、ホスト名からアドレスへの変換の速度が向上します。ローカルキャッシュには、2つのタイプのエントリが存在します。スタティックとダイナミックです。domain ipv4 host または domain ipv6 host コマンドを使用して設定されたエントリはスタティック エントリとして追加され、ネーム サーバから受信したエントリはダイナミック エントリとして追加されます。

ネーム サーバは、World Wide Web（WWW）でネットワーク ノードの名前をアドレスに変換するために使用されます。ネーム サーバは、DNS サーバから DNS プロトコルを使用して、ホスト名を IP アドレスにマッピングする分散データベースを維持します。domain name-server コマンドを使用して、1 つ以上のネーム サーバを指定できます。

アプリケーションでホストの IP アドレスまたは IP アドレスのホスト名が必要になると、ドメイン サービスに対してリモート プロシージャ コール（RPC）が実行されます。ドメイン サービスは、キャッシュ内で IP アドレスまたはホスト名を探し、エントリが見つからない場合にはネーム サーバに DNS クエリーを送信します。

ドメイン名要求を完了するために Cisco IOS XR ソフトウェアで使用されるデフォルト ドメイン名を指定できます。単一のドメインまたはドメイン名のリストを指定することもできます。IP ホスト名にドメイン名が含まれていない場合には、ホスト テーブルに追加される前に指定のドメイン名が付加されます。1 つまたは複数のドメイン名を指定するには、domain name コマンドまたは domain list コマンドを使用します。

ドメイン サービスの設定

デフォルトでは、DNS によるホスト名からアドレスへの変換がイネーブルになっています。**domain lookup disable** コマンドによってホスト名からアドレスへの変換がディセーブルになっている場合は、 **no domain lookup disable** コマンドを使用して変換を再びイネーブルにします。

設定例

スタティック ホスト名とアドレスのマッピングを定義します。IPv4 アドレスを 2 つのホスト（192.168.7.18 と 10.2.0.2 192.168.7.33）に関連付けます（またはマッピングします）。ホスト名は host1 と host2 です。

```
Defining the Domain Host
==================================
Router# configure
Router(config)#domain ipv4 host host1 192.168.7.18
```

```
Router(config)#domain ipv4 host host2 10.2.0.2 192.168.7.33
Router(config)#commit

Defining the Domain Name
=========================
*/Define cisco.com as the default domain name/*
Router#configure
Router(config)#domain name cisco.com
Router(config)#commit

Specifying the Addresses of the Name Servers
=============================================
*/Specify host 192.168.1.111 as the primary name server
and host 192.168.1.2 as the secondary server/*
Router#configure
Router(config)#domain name-server 192.168.1.111
Router(config)#domain name-server 192.168.1.2
Router(config)#commit
```

確認

```
Router#show hosts
Default domain is cisco.com
Name/address lookup uses domain service
Name servers: 192.168.1.111, 192.168.1.2

Host                      Flags     Age(hr) Type Address(es)
host2                   (perm, OK)    0      IP  10.2.0.2
                                                 192.168.7.33
host1                   (perm, OK)    0      IP  192.168.7.18
```

関連コマンド

TFTP サーバ

サーバとしてだけ機能するマシンをネットワークの各セグメントに配置するのは、コストがかかり、非効率的です。しかし、すべてのセグメントにサーバがあるのではない場合、ネットワーク セグメントを超えたネットワークの操作によって相当の遅延が引き起こされることがあります。ルータを TFTP サーバとして機能するように設定すると、ルータの通常の機能を使用しながらコストと遅延時間を削減できます。

一般に、TFTP サーバとして設定されたルータは、フラッシュ メモリから他のルータにシステム イメージまたはルータ コンフィギュレーション ファイルを提供します。他のタイプのサービス要求に応答するようにルータを設定することもできます。

TFTP サーバとしてのルータの設定

TFTP 機能の実装前に、サーバとクライアント ルータは互いに到達可能である必要があります。ping コマンドを使用してサーバとクライアント ルータ間の接続を（いずれかの方向で）テストして、この接続を検証します。

このタスクを実行すると、ルータを TFTP サーバとして設定できます。これにより、TFTP クライアントとして機能する他のデバイスは、slot0: や /tmp などの特定のディレクトリ（TFTP ホーム ディレクトリ）の下にあるファイルをルータに対して読み書きできます。

（注） セキュリティを確保するため、ファイルがすでに存在していないと、TFTP サーバでは書き込み要求を正常に完了できません。

TFTP 機能の実装前に、サーバとクライアント ルータは互いに到達可能である必要があります。ping コマンドを使用してサーバとクライアント ルータ間の接続を（いずれかの方向で）テストして、この接続を検証します。

設定例

TFTP サーバとしてルータを設定します（ホーム ディレクトリの disk0:）。

```
Router#configure
Router(config)#tftp ipv4 server homedir disk0
Router(config)#commit
```

実行コンフィギュレーション

```
Router#show running-config tftp ipv4 server homedir disk0:
tftp vrf default ipv4 server homedir disk0:
```

確認

```
Router#show cinetd services
Vrf Name Family Service     Proto Port ACL  max_cnt  curr_cnt wait  Program Client Option

default  v4     tftp     udp   69      unlimited 0       wait   tftpd   sysdb disk0:

default  v4       telnet      tcp   23      10        0       nowait telnetd sysdb
```

関連コマンド

ファイル転送サービス

ファイル転送プロトコル（FTP）、簡易ファイル転送プロトコル（TFTP）、リモート コピー プロトコル（RCP）の各クライアント、およびセキュア コピープロトコル（SCP）はファイル システムまたはリソース マネージャとして実装されます。たとえば、tftp:// で始まるパス名は TFTP リソース マネージャによって処理されます。

ファイル システム インターフェイスは、URL を使用して、ファイルの場所を指定します。URL は、WWW でファイルまたは場所を指定するのに広く使用されています。ただし、Cisco ルータの URL には、ルータまたはリモート ファイル サーバ上のファイルの場所も指定されます。

ルータがクラッシュしたときは、ルータのメモリ内容全体のコピーを取得するのが便利です（これをコア ダンプと言います）。テクニカル サポート担当者が、クラッシュの原因を特定

するのに使用します。SCP、FTP、TFTP、RCP を使用すると、コア ダンプをリモート サーバに保存できます。

FTP

ファイル転送プロトコル（FTP）は、TCP/IP プロトコル スタックの一部であり、ネットワーク ノード間でファイルを転送するのに使用します。FTP は、RFC 959 で定義されています。

FTP 接続使用時のルータ設定

FTP接続を使用してネットワーク上のシステム間でファイルを転送するようにルータを設定できます。次の FTP の特性を設定できます。

- パッシブ モード FTP
- パスワード
- IP アドレス

設定例

ルータによる FTP 接続の使用をイネーブルにします。パッシブ FTP 接続を使用するようにソフトウェアを設定し、匿名ユーザのパスワードを設定して、FTP 接続の送信元 IP アドレスも指定します。

```
Router#configure
Router(config)#ftp client passive
(Optional) Router(config)#ftp client vrf vrfa
Router(config)#ftp client anonymous-password xxxx
Router(config)#ftp client source-interface HundredGigE 0/9/0/0
Router(config)#commit
```

実行コンフィギュレーション

```
Router#show running-config ftp client passive
ftp client passive
ftp client vrf vrfa
Router#show running-config ftp client anonymous-password xxxx
ftp client anonymous-password xxxx
Router#show running-config ftp client source-interface HundredGigE 0/9/0/0
ftp client source-interface HundredGigE 0/9/0/0
```

関連コマンド

- ftp client passive
- ftp client anonymous-password
- ftp client source-interface

TFTP

Trivial File Transfer Protocol（TFTP）は FTP の簡易版で、ネットワークを介して 1 つのコンピュータから別のコンピュータにファイルを転送できます。通常は、クライアント認証（ユーザ名とパスワードなど）を使用しません。

TFTP 接続使用時のルータ設定

設定例

TFTP 接続を使用するようにルータを設定し、TFTP 接続の送信元アドレスとして HundredGigE 0/9/0/0 の IP アドレスを設定します。

```
Router#configure
Router(config)#tftp client source-interface HundredGigE 0/9/0/0
Router(config)#commit
```

実行コンフィギュレーション

```
Router#show running-config tftp client source-interface HundredGigE 0/9/0/0
tftp client source-interface HundredGigE 0/9/0/0
```

確認

```
Router#show cinetd services
Vrf Name Family  Service  Proto Port ACL max_cnt  curr_cnt  wait   Program Client Option

default  v4     tftp udp    69      unlimited  0       wait   tftpd    sysdb disk0:

default  v4      telnet   tcp    23      10         0       nowait telnetd  sysdb
```

関連コマンド

- tftp client source-interface type
- show cinetd services

SCP

セキュア コピー プロトコル（SCP）は、ファイルを転送するための認証されたセキュアな方式を提供するファイル転送プロトコルです。SCP は SSHv2 に依存して、リモート ロケーションからローカル ロケーションに、またはローカル ロケーションからリモート ロケーションにファイルを転送します。

Cisco IOS XR ソフトウェアは SCP サーバ操作とクライアント操作をサポートしています。デバイスが SCP 要求を受信すると、SSH サーバ プロセスはクライアントとやり取りする SCP サーバ プロセスを生成します。各着信 SCP サブシステム要求に対して新しい SCP サーバ インスタンスが生成されます。デバイスが宛先デバイスにファイル転送要求を送信する場合、そのデバイスはクライアントとして機能します。

デバイスがファイル転送のためにリモート ホストとの SSH 接続を開始すると、リモート デバイスはソース モードまたはシンク モードで要求に応答することができます。ソース モードで

は、デバイスはファイル ソースになります。デバイスはそのローカル ディレクトリからファイルを読み取り、目的の宛先に転送します。シンク モードでは、デバイスは転送するファイルの宛先になります。

SCP を使用して、ローカル デバイスから宛先デバイスに、または宛先デバイスからローカル デバイスにファイルをコピーできます。

SCP では、個々のファイルの転送のみを実行できます。宛先デバイスから別の宛先デバイスにファイルを転送することはできません。

SCP によるファイル転送

セキュア コピー プロトコル（SCP）を使用すると、送信元デバイスと宛先デバイス間でファイルを転送できます。一度に 1 つのファイルを転送できます。宛先がサーバの場合は、SSH サーバ プロセスが実行されている必要があります。

設定例

ファイル「test123.txt」をローカル ディレクトリからリモート ディレクトリに転送します。

```
Router#scp /harddisk:/test123.txt xyz@1.75.55.1:/auto/remote/test123.txt
Connecting to 1.75.55.1...
Password:
Router#commit
```

確認

テキスト「test123.txt」ファイルがコピーされたことを確認します。

```
xyz-lnx-v1:/auto/remote> ls -altr test123.txt
-rw-r--r-- 1 xyz eng 0 Nov 23 09:46 test123.txt
```

関連コマンド

- scp

Cisco inetd

Cisco インターネット サービス プロセス デーモン（Cinetd）は、システムのブート後にシステム マネージャによって開始されるマルチスレッド サーバ プロセスです。Cinetd は、Telnet サービスや TFTP サービスなどのインターネット サービスをリッスンします。Cinetd が特定のサービスをリッスンするかどうかは、ルータ コンフィギュレーションによって異なります。たとえば、 **tftp server** コマンドを入力すると、Cinetd は TFTP サービスのリッスンを開始します。要求が届くと、Cinetd はサービスに関連付けられたサーバ プログラムを実行します。

Telnet

Telnet をイネーブルにすると、ネットワーキング デバイスで着信 Telnet 接続が許可されます。

設定例

Telnet をイネーブルにして、ルータに同時にアクセスできるユーザの数を 10 人に制限します。

```
Router# configure
Router(config)# telnet ipv4 server max-servers 10
Router(config)# commit
```

確認

```
Router# show cinetd services
Vrf Name  Family    Service   Proto Port ACL max_cnt  curr_cnt  wait   Program Client
Option
default   v4         tftp      udp  69       unlimited  0        wait   tftpd   sysdb
disk0:
default   v4     telnet  tcp  23        10  0        nowait telnetd sysdb
```

関連コマンド

Syslog の送信元インターフェイス

ロギング送信元インターフェイスを設定すると、特定のルータから VRF で発信される syslog トラフィックを、単一のデバイスからの着信として識別できます。

設定例

リモート Syslog サーバの送信元インターフェイスをイネーブルにします。デフォルトの vrf のロギング送信元インターフェイスとして loopback 2 を設定します。

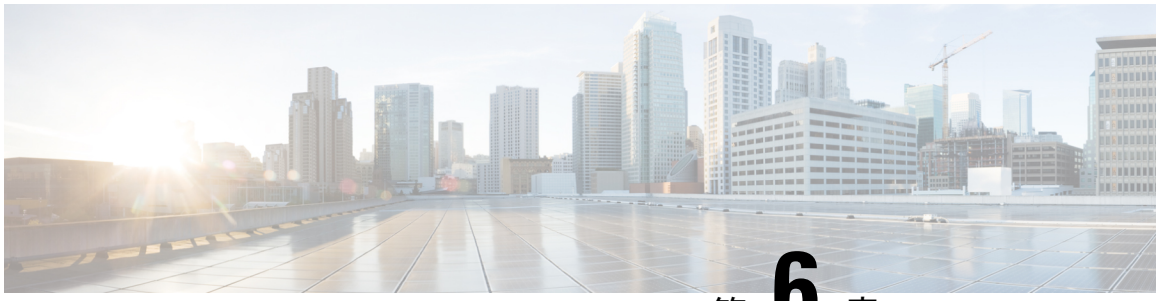
```
Router#configure
Router(config)#logging source-interface Loopback2
Router(config)#logging source-interface Loopback3 vrf vrfa
Router(config)#commit
```

実行コンフィギュレーション

```
Router#show running-config logging
/*Logging configuration after changing the source into loopback2 interface.
logging console debugging
logging monitor debugging
logging facility local4
logging 123.100.100.189 vrf default severity info port default
logging source-interface Loopback2
logging source-interface Loopback3 vrf vrfa
```

関連コマンド

- logging source-interface
- show running-configuration logging



第 6 章

アクセス リストおよびプレフィックス リストの実装

アクセス リストの概要

アクセス リストは、パケット フィルタリングを実行して、ネットワークを介して移動するパケットとその場所を制御します。この処理は、ネットワーク トラフィックを制限したり、ユーザやデバイスによるネットワークへのアクセスを制限したりするのに役立ちます。アクセス リストの用途は多様なので、多くのコマンドの構文でアクセス リストが参照されます。アクセス リストを使用して、次のようなことを実行できます。

アクセス コントロール リスト（ACL）は、ネットワーク トラフィック プロファイルをまとめて定義する 1 つ以上のアクセス コントロール エントリ（ACE）です。このプロファイルは、トラフィック フィルタリング、ルート フィルタリング、QoS 分類、アクセス コントロールなど、Cisco IOS XR ソフトウェアの機能で参照できます。次の 2 種類の ACL があります。

- 標準的な ACL：パケットの送信元 IP アドレスのみを確認します。トラフィックは、ACL に設定されたアドレスまたはプレフィックスの比較と、パケットにある送信元アドレスで制御されます。
- 拡張ACL：パケットの発信元アドレス以外の属性も確認します。確認する属性は、送信先アドレス、特定の IP プロトコル、UDP または TCP ポート番号、DSCP などです。ACL に記載されている属性と、着信パケットまたは発信パケット内の属性を比較することで、トラフィックが制御されます。

Cisco IOS XR は標準アクセス リストと拡張アクセス リストとを区別しません。標準アクセス リストをサポートしているのは、下位互換性を確保するためです。

IP アクセス リストの目的

- インターフェイスで着信パケットまたは発信パケットをフィルタリングします。
- ミラーリングのためにパケットをフィルタリングします。
- 必要に応じて、トラフィックをリダイレクトします。
- ルーティング アップデートの内容の制限
- アドレスまたはプロトコルに基づくデバッグ出力の制限
- vty へのアクセスの制御
- 輻輳回避、輻輳管理、プライオリティ キューイング、カスタム キューイングなどの高度な機能に使用されるトラフィックの特定または分類

IP アクセス リストの機能

アクセス リストは、permit ステートメントと deny ステートメントで構成される順次リストです。これらのステートメントは、IP アドレス、場合によっては上位層 IP プロトコルに適用されます。アクセス リストには、参照に使用される名前があります。多くのソフトウェア コマンドは、構文の一部としてアクセス リストを受け取ります。

アクセス リストを設定して名前を付けることは可能ですが、アクセス リストを受け取るコマンドによってアクセスリストが参照されるまで、有効にはなりません。複数のコマンドから同じアクセス リストを参照できます。アクセス リストで、ルータに到達するトラフィック、またはルータ経由で送信されるトラフィックは制御できますが、ルータが送信元のトラフィックは制御できません。

送信元アドレスと宛先アドレスは、IPパケットの最も一般的な2つのフィールドで、アクセスリストの基礎となります。送信元アドレスを指定して、特定のネットワーキングデバイスまたはホストからのパケットを制御します。宛先アドレスを指定して、特定のネットワーキングデバイスまたはホストに送信されるパケットを制御します。

また、トランスポート層の情報（パケットが TCP、UDP、ICMP、IGMP のいずれであるかなどの情報）に基づいてパケットをフィルタリングすることもできます。

ACL のワークフロー

次の図に、ACL のワークフローを示します。

IP アクセス リストのプロセスとルール

IP アクセス リストを設定するときは、次のプロセスとルールを使用してください。

- アクセスリストの条件に対してフィルタリングされる各パケットの送信元アドレスや宛先アドレス、またはプロトコルがテストされます。一度に 1 つの条件（permit ステートメントまたは deny ステートメント）がテストされます。
- パケットがアクセスリストのステートメントに一致しないと、そのパケットはリスト内の次のステートメントに対してテストされます。
- パケットとアクセスリストのステートメントが一致すると、リスト内の残りのステートメントはスキップされ、パケットは一致したステートメントに指定されたとおりに許可または拒否されます。パケットが許可されるか拒否されるかは、パケットが一致する最初のエントリによって決まります。つまり、一致すると、それ以降のエントリは考慮されません。
- アクセスリストでアドレスまたはプロトコルが拒否されると、パケットは廃棄され、インターネット制御メッセージプロトコル（ICMP）ホスト到達不能メッセージが返されます。ICMP は、Cisco IOS XR ソフトウェアで設定できます。
- 各アクセス リストの最後には暗黙の deny ステートメントがあるため、一致する条件がない場合は、パケットはドロップされます。つまり、各ステートメントに対してテストするときまでにパケットを許可または拒否しないと、パケットは拒否されます。
- アクセス リストには permit ステートメントを 1 つ以上含める必要があります。そうしないと、パケットはすべて拒否されます。
- 最初に一致が見つかった後は条件のテストが終了するため、条件の順序は重要です。同じ permit ステートメントまたは deny ステートメントでも、順序が異なる場合、ある状況では通過し、別の状況では拒否されるパケットが生じる可能性があります。
- 1 つのインターフェイス、1 つのプロトコル、1 つの方向につき、許可されるアクセス リストは 1 つだけです。
- インバウンド アクセス リストは、ルータに到達するパケットを処理します。着信パケットの処理後に、アウトバウンドインターフェイスへのルーティングが行われます。インバウンドアクセスリストが効率的なのは、フィルタリングテストで拒否されたことでパケットが廃棄される場合、ルーティング検索のオーバーヘッドが抑えられるためです。パケットがテストで許可されると、そのパケットに対してルーティングの処理が実施されます。インバウンド リストの場合、permit はインバウンド インターフェイスで受信したパケットが引き続き処理されることを意味し、 **deny** はパケットが廃棄されることを意味します。

- アウトバウンド アクセス リストの場合、パケットの処理後にルータから送信されます。着信パケットはアウトバウンドインターフェイスにルーティングされてから、アウトバウンドアクセスリストで処理されます。アウトバウンドリストの場合、許可とは、出力バッファに対して送信されることを示し、拒否とは、パケットが廃棄されることを示します。
- アクセス リストは、使用中のアクセス グループによって適用されている場合には削除できません。アクセス リストを削除するには、まずアクセス リストを参照しているアクセス グループを削除してから、アクセス リストを削除します。
- 特定のトラフィックを拒否するACLで設定されているインターフェイスを削除する前に、そのACLを削除し、設定をコミットする必要があります。これを実行しないと、**no interface <interface-name>** コマンドが設定され、コミットされるとすぐにインターフェイスを通じて一部のパケットがリークします。
- **ipv4 access group** コマンドを使用するには、アクセス リストが必要です。

ワイルドカード マスクと暗黙的なワイルドカード マスクを使用した ACL フィルタリング

アドレス フィルタリングでは、アクセス リスト エントリ内のアドレス ビットとアクセス リストに送信されるパケットを比較するときに、ワイルドカード マスクを使用して、対応する IP アドレス ビットを確認するか無視するかを指定します。管理者は、ワイルドカード マスクを慎重に設定することにより、許可または拒否のテストに 1 つまたは複数の IP アドレスを選択できます。

IP アドレス ビット用のワイルドカード マスクでは、数値 1 と数値 0 を使用して、対応する IP アドレス ビットをどのように扱うかを指定します。1 と 0 は、サブネット（ネットワーク）マスクで意味する内容が逆になるため、ワイルドカード マスクは逆マスクとも呼ばれます。

- ワイルドカード マスク ビット 0 は、対応するビット値を確認することを示します。
- ワイルドカード マスクのビット 1 は、対応するビット値を無視することを意味します。

アクセス リスト ステートメントでは、送信元アドレスまたは宛先アドレスにワイルドカード マスクを指定する必要はありません。 **host** キーワードを使用すると、ワイルドカード マスクとして 0.0.0.0 を指定したものと見なされます。

サブネットマスクでは、ネットワークとサブネットを示す隣接ビットをマスクにする必要がありますが、それとは異なり、ワイルドカード マスクではマスクに非隣接ビットを使用できます。

ワイルドカード ビットの代わりに、CIDR 形式（/x）を使用することもできます。たとえば、IPv4 アドレス 1.2.3.4 0.255.255.255 は 1.2.3.4/8 に相当し、IPv6 アドレスの場合、2001:db8:abcd:0012:0000:0000:0000:0000 は 2001:db8:abcd:0012::0/64 に相当します。

アクセス リストのコメントの組み入れ

remark アクセス リスト コンフィギュレーション コマンドを使用すると、名前付き IP アクセス リストにエントリに関するコメント（注釈）を含めることができます。コメントを含めると、ネットワーク管理者がアクセス リストを理解し、精査しやすくなります。1 つのコメント行の最大長は 255 文字です。

コメントは、**permit** ステートメントまたは **deny** ステートメントの前後どちらにでも配置できます。コメントがどの **permit** ステートメントまたは **deny** ステートメントの説明であるのかが明確になるように、コメントの位置に関して一貫性を保つようにしてください。たとえば、一部のコメントが **permit** または **deny** ステートメントの前にあり、他のコメントがステートメントの後ろにあると、混乱を招きます。コメントに順番を付けることができます。

アクセス リストの作成後、アクセス リストをインターフェイスまたは端末回線に適用することを忘れないでください。

IPv4 および IPv6 用のユーザ定義の TCAM キー

Cisco NCS 540 シリーズ ルータ上のアクセスリストは、TCAM（内部および外部）を使用して各パケットでルックアップとアクションの解決を実行します。TCAMはハードウェアに搭載された制約のある貴重なリソースであり、複数の機能で共有する必要があります。したがって、これらのキー定義で使用可能なスペース（キー幅）も制限されます。キー定義では、ルックアップの実行時に ACL 機能で使用可能な修飾子およびアクション フィールドを指定します。各キー定義に、使用可能な修飾子およびアクションフィールドをすべて含めることはできません。

キー定義は特定の ACL タイプに固有のものであり、アクセスリストの次の属性によって決まります。

- 接続の方向（入力または出力）
- プロトコル タイプ（IPv4/IPv6/L2）
- 圧縮レベル（0：非圧縮、3：圧縮）

デフォルトのキー定義には制約がある（すべての修飾子/アクションフィールドが含まれない）ため、次のタイプのユーザ定義キー（UDK）定義がサポートされます。

- 従来の入力 IPv4 ACL（非圧縮）
- 従来の入力 IPv6 ACL（非圧縮）

ユーザ定義の TCAM キー（UDK）機能を使用することで、考えられる 3 つの目的（入力用、従来用、Ipv4/Ipv6 ACL 専用）のいずれかのために独自の TCAM キーを柔軟に定義できます。

デフォルトの TCAM キーに含まれていない修飾子フィールドを含めるため。

ACL モードを共有から固有に変更して、一意の ACL および一意のカウンタなどをより多くサポートするため。

TCAM キーのサイズ（消費されるバンクの数）を減らすため。

UDK は次のコマンドを使用して設定できます。

```
hw-module profile tcam format access-list [ipv4 | ipv6] qualifiers [location
rack/slot/cpu0]
```

ユーザによって定義されたフィールド

TCAM キーは複数の修飾子で構成され、特定の ACL のパケットをフィルタリングするために一連の修飾子が使用されます。ユーザ定義フィールド（UDF）では、次の UDF コマンドを使用してフィールドの場所とサイズを指定することでカスタム修飾子を定義できます。

udf *udf-name* **header** *[inner | outer] [l2 | l3 | l4]* **offset** *byte-offset* **length** *no of bytes*

その後、次のように UDF を UDK に追加できます。

```
hw-module profile tcam format access-list [ipv4 | ipv6] qualifiers [udf1 udf-name udf2 udf-name] [location rack/slot/cpu0]
```

（注） システム全体で UDF を 8 個まで定義できます。現在、UDF はグローバルに定義されます。

従来の入力 ACL の IPv4 および IPv6 キー形式

ユーザ定義の TCAM キー（UDK）の定義は、従来（非圧縮）の入力 IPv4 および IPv6 ACL でサポートされています。

次の表に、IPv4 および IPv6 キー形式でサポートされる修飾子フィールドを示します。デフォルトの TCAM キーがイネーブルに設定されている場合、修飾子フィールドはデフォルトでイネーブルになります。デフォルトの TCAM キーがディセーブルに設定されている場合は、修飾子フィールドで UDK を使用する必要があります。

表 *4: IPv4* および *IPv6* キー形式でサポートされる修飾子フィールド

パラメータ	デフォルトの TCAM キー	
	IPv4	IPv6
Source Address	イネーブル	イネーブル
Destination Address	イネーブル	イネーブル
Source Port	イネーブル	イネーブル
Destination Port	イネーブル	イネーブル
Port Range	イネーブル	サポート対象外
Protocol/Next Header	イネーブル	イネーブル
Fragment bit	イネーブル	サポート対象外
Packet length	ディセーブル	ディセーブル
Precedence/DSCP	ディセーブル	イネーブル

パラメータ	デフォルトの TCAM キー	
	IPv4	IPv6
TCP Flags	イネーブル	イネーブル
TTL Match	ディセーブル	ディセーブル
Interface based	ディセーブル	サポート対象外
UDF 1-8	ディセーブル	ディセーブル
ACL ID	イネーブル	イネーブル
common ACL bit	共有モードの IPv4/IPv6 ではデフォルトでイネーブル 固有モードの IPv4/IPv6 ではデフォルトでディセーブル	共有モードの IPv4/IPv6 ではデフォルトでイネーブル 固有モードの IPv4/IPv6 ではデフォルトでディセーブル
Interface-based (RIF)	ディセーブル	ディセーブル

次の表に、IPv4 および IPv6 キー形式でサポートされるアクション フィールドを示します。

表 *5 : IPv4* および *IPv6* キー形式でサポートされるアクション フィールド

パラメータ	デフォルトのアクション フィールド	
	IPv4	IPv6
Permit	イネーブル	イネーブル
Deny	イネーブル	イネーブル
Log	イネーブル	イネーブル
Capture	イネーブル	イネーブル
Stats Counter	拒否統計は常にイネーブル（許可統計には独自の hw-module コマンドがある）	拒否統計は常にイネーブル
TTL Set	イネーブル	イネーブル

IPv4 ACL の設定

この項では、IPv4 の入力 ACL と出力 ACL の基本的な設定について説明します。

IPv4 入力 ACL を設定するための注意事項と制約事項

IPv4 入力 ACL は、次の動作を特徴としています。

- 入力 IPv4 ACL は、管理インターフェイス以外のすべてのインターフェイスでサポートされています。
- ACL ベースの転送（ABF）は、入力方向でのみサポートされています。
- NPU ごとにデフォルトで許可されている ACL の総数は 31 です。
- ライン カードごとに許可されている付加された ACE の数は 4,000 です。
- 入力インターフェイスによる ACL ロギング（**log-input** キーワードを使用）はサポートされていません。
- 統計情報を確認するための **show access-lists ipv4 acl_name stats** コマンドは、ログを含む ACE ではサポートされません。したがって、ログを含む ACE の統計情報を確認するには、**show access-lists** *acl-name* **hardware [ingress | egress] detail location***loc* コマンドを使用してください。

IPv4 出力 ACL を設定するための注意事項と制約事項

IPv4 出力 ACL は、次の動作を特徴としています。

- 出力 IPv4 ACL は、メインの物理インターフェイスとバンドル インターフェイスでサポートされています。

（注） 出力 ACL は、サブインターフェイスでは直接サポートされていません。ただし、サブインターフェイスがあるメインのインターフェイス上に出力 ACL を設定した場合、ACL アクションはサブインターフェイス トラフィックにも適用されます。この出力 ACL の動作は、メインのインターフェイスに ACL を適用した後でサブインターフェイスを設定した場合も同じです。

- NPU ごとに許可された出力 ACL の総数は 255 です。
- ACL は、出力方向の管理インターフェイスではサポートされていません。
- ライン カードごとに許可されている付加された ACE の数は 4,000 です。
- ACL ロギング（**log** コマンドを使用）と入力インターフェイスによる ACL ロギング（**log-input** コマンドを使用）はサポートされていません。

ギガビット イーサネット インターフェイス上での入力 IPv4 ACL の設定

GigE インターフェイス上で入力 IPv4 ACL を設定するには、次の設定を使用します。

```
/* Configure a GigE interface with an IPv4 address */
Router(config)# interface TenGigE 0/11/0/0
Router(config-if)# ipv4 address 10.1.1.1 255.255.255.0
Router(config-if)# no shut
Router(config-if)# commit
Thu Jan 25 10:07:54.700 IST
```

```
Router(config-if)# exit

/* Verify if the interface is up */
Router(config)# do show ipv4 interface brief
Thu Jan 25 10:08:49.087 IST

Interface                      IP-Address      Status          Protocol
Vrf-NameTenGigE0/11/0/0         10.1.1.1        Up              Up       default

/* Configure an IPv4 ingress ACL */
Router(config)# ipv4 access-list V4-ACL-INGRESS
Router(config-ipv4-acl)# 10 permit tcp 10.2.1.1 0.0.0.255 any
Router(config-ipv4-acl)# 20 deny udp any any
Router(config-ipv4-acl)# 30 permit ipv4 10.2.0.0 0.255.255.255 any
Router(config-ipv4-acl)# commit
Thu Jan 25 10:16:11.473 IST

/* Verify the ingress ACL creation */
Router(config)# do show access-lists ipv4
Thu Jan 25 10:25:19.896 IST
...
ipv4 access-list V4-ACL-INGRESS
 10 permit tcp 10.2.1.0 0.0.0.255 any
 20 deny udp any any
 30 permit ipv4 10.0.0.0 0.255.255.255 any


/* Apply the ingress ACL to the GigE interface */
Router(config)# interface TenGigE0/11/0/0
Router(config-if)# ipv4 access-group V4-ACL-INGRESS ingress
Router(config-if)# commit
Thu Jan 25 10:28:19.671 IST
Router(config-if)# exit

/* Verify if the ingress ACL has been successfully applied to the interface */
Router(config)# do show ipv4 interface
Thu Jan 25 10:29:44.944 IST
TenGigE0/11/0/0 is Up, ipv4 protocol is Up
  Vrf is default (vrfid 0x60000000)
  Internet address is 10.1.1.1/24
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound  common access list is not set, access list is V4-ACL-INGRESS
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000
```

ギガビット イーサネット インターフェイス上に IPv4 入力 ACL を正常に設定しました。

ギガビット イーサネット インターフェイス上での出力 IPv4 ACL の設定

GigE インターフェイス上で出力 IPv4 ACL を設定するには、次の設定を使用します。

```
/* Configure a GigE interface with an IPv4 address */
Router(config)# interface TenGigE 0/11/0/0
Router(config-if)# ipv4 address 20.1.1.1 255.255.255.0
Router(config-if)# no shut
Router(config-if)# commit
```

```
Thu Jan 25 10:08:38.767 IST
Router(config-if)# exit

/* Verify if the interface is up */
Router(config)# do show ipv4 interface brief
Thu Jan 25 10:08:49.087 IST

Interface                      IP-Address      Status          Protocol Vrf-Name
TenGigE0/11/0/0         10.1.1.1        Up              Up       default
TenGigE0/11/0/1         20.1.1.1        Up              Up       default

/* Configure an IPv4 egress ACL */
Router(config)# ipv4 access-list V4-ACL-EGRESS
Router(config-ipv4-acl)# 10 permit ipv4 10.2.0.0 0.255.255.255 20.2.0.0 0.255.255.255
Router(config-ipv4-acl)# 20 deny ipv4 any any
Router(config-ipv4-acl)# commit
Thu Jan 25 10:25:04.655 IST

/* Verify the egress ACL creation */
Router(config)# do show access-lists ipv4
Thu Jan 25 10:25:19.896 IST
ipv4 access-list V4-ACL-EGRESS
 10 permit ipv4 10.0.0.0 0.255.255.255 20.0.0.0 0.255.255.255
 20 deny ipv4 any any
...

/* Apply the egress ACL to the GigE interface */
Router(config)# interface TenGigE 0/11/0/1
Router(config-if)# ipv4 access-group V4-ACL-EGRESS egress
Router(config-if)# commit
Thu Jan 25 10:28:45.937 IST
Router(config-if)# exit

/* Verify if the egress ACL has been successfully applied to the interface */
Router(config)# do show ipv4 interface
Thu Jan 25 10:29:44.944 IST
TenGigE 0/11/0/1 is Up, ipv4 protocol is Up
  Vrf is default (vrfid 0x60000000)
  Internet address is 20.1.1.1/24
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is V4-ACL-EGRESS
  Inbound  common access list is not set, access list is not set
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000
...
```

ギガビット イーサネット インターフェイス上に IPv4 出力 ACL を正常に設定しました。

IPv6 ACL の設定

この項では、ギガビット イーサネットとバンドル イーサネットを介して入力 IPv6 ACL と出力 IPv6 ACL を設定するステップについて説明します。

IPv6 入力 ACL を設定するための注意事項と制約事項

IPv6 入力 ACL は、次の動作を特徴としています。

- 入力 IPv6 ACL はすべてのインターフェイスでサポートされています。
- ACL ベースの転送（ABF）は、入力方向でのみサポートされています。
- NPU ごとに許可された入力 ACL の総数は 31 です。
- ライン カードごとに許可されている付加された ACE の数は 2047 です。
- 入力インターフェイスによる ACL ロギング（**log-input** キーワードを使用）はサポートされていません。
- パケット長（**pkt-length** キーワードを使用）はサポートされていません。
- 統計情報を確認するための **show access-lists ipv4 acl_name stats** コマンドは、ログを含む ACE ではサポートされません。したがって、ログを含む ACE の統計情報を確認するには、**show access-lists** *acl-name* **hardware [ingress | egress] detail location***loc* コマンドを使用してください。

IPv6 出力 ACL を設定するための注意事項と制約事項

IPv6 出力 ACL は、次の動作を特徴としています。

- パケット長の設定は、出力 ACL ではサポートされていません。
- TCP フラグは出力 ACL ではサポートされていません。
- 出力 ACL は、BVI インターフェイスおよび L2 インターフェイスではサポートされていません。
- QoS グループの設定は、出力 ACL ではサポートされていません。
- 出力 ACL では、スループットの 50% 以下がサポートされます。
- スループットの制限を除き、ルータによって生成されたトラフィックが出力 IPv6 ACL の影響を受けることはありません。
- NPU ごとに許可された出力 ACL の総数は 255 です。
- ライン カードごとに許可されている付加 ACE の合計数は 2000 です。
- ダイナミック TCAM キーの設定は、出力 ACL ではサポートされていません。
- NPU ごとにサポートされる IPv6 出力 ACL の合計は最大 160 GB です。

ギガビット イーサネット インターフェイス上での入力 IPv6 ACL の設定

GigE インターフェイス上で入力 IPv6 ACL を設定するには、次の設定を使用します。

```
/* Configure a GigE interface with an IPv6 address */
Router(config)# interface TenGigE 0/11/0/0
Router(config-if)# ipv6 address 1001::1/64
```

```
Router(config-if)# no shut
Router(config-if)# commit
Thu Jan 25 10:07:54.700 IST
Router(config-if)# exit

/* Verify if the interface is up */
Router(config)# do show ipv6 interface brief
Thu Jan 25 12:38:35.742 IST
TenGigE 0/11/0/0 [Up/Up]
    fe80::bd:b9ff:fea9:5606
    1001::1
…

/* Configure an IPv6 ingress ACL */
Router(config)# ipv6 access-list V6-INGRESS-ACL
Router(config-ipv6-acl)# 10 permit ipv6 any any
Router(config-ipv6-acl)# 20 deny udp any any
Router(config-ipv6-acl)# commit
Thu Jan 25 11:31:24.488 IST
Router(config-ipv6-acl)# exit


/* Verify the ingress ACL creation */
Router(config)# do show access-lists ipv6
Thu Jan 25 11:34:56.911 IST
ipv6 access-list V6-INGRESS-ACL
 10 permit ipv6 any any
 20 deny udp any any


/* Apply the ingress ACL to the GigE interface */
Router(config)# interface TenGigE 0/11/0/0
Router(config-if)# ipv6 access-group V6-INGRESS-ACL ingress
Router(config-if)# commit
Thu Jan 25 11:32:55.194 IST
Router(config-if)# exit

/* Verify if the ingress ACL has been successfully applied to the interface */
Router(config)# do show ipv6 interface
Thu Jan 25 11:34:08.028 IST
TenGigE 0/11/0/0 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
  IPv6 is enabled, link-local address is fe80::bd:b9ff:fea9:5606
  Global unicast address(es):
    1001::1, subnet is 1001::/64
  Joined group address(es): ff02::1:ff00:1 ff02::1:ffa9:5606 ff02::2
      ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND cache entry limit is 1000000000
  ND advertised retransmit interval is 0 milliseconds
  Hosts use stateless autoconfig for addresses.
  Outgoing access list is not set
  Inbound  common access list is not set, access list is V6-INGRESS-ACL
  Table Id is 0xe0800000
  Complete protocol adjacency: 0
  Complete glean adjacency: 0
  Incomplete protocol adjacency: 0
  Incomplete glean adjacency: 0
  Dropped protocol request: 0
  Dropped glean request: 0
```

…

ギガビット イーサネット インターフェイス上に IPv6 入力 ACL を正常に設定しました。

ギガビット イーサネット インターフェイス上での出力 IPv6 ACL の設定

GigE インターフェイス上で出力 IPv6 ACL を設定するには、次の設定を使用します。

```
/* Configure a GigE interface with an IPv6 address */
Router(config)# interface TenGigE 0/11/0/1
Router(config-if)# ipv6 address 2001::1/64
Router(config-if)# no shut
Router(config-if)# commit
Thu Jan 25 11:41:25.778 IST
Router(config-if)# exit

/* Verify if the interface is up */
Router(config)# do show ipv6 interface brief
Thu Jan 25 12:38:35.742 IST
TenGigE 0/11/0/0 [Up/Up]
    fe80::bd:b9ff:fea9:5606
    1001::1
TenGigE 0/11/0/1 [Up/Up]
    fe80::23:e9ff:fea8:a44e
    2001::1

/* Configure an IPv6 egress ACL */
Router(config)# ipv6 access-list V6-EGRESS-ACL
Router(config-ipv6-acl)# 10 permit ipv6 any any
Router(config-ipv6-acl)# 20 deny udp any any
Router(config-ipv6-acl)# commit
Thu Jan 25 11:44:03.969 IST
Router(config-ipv6-acl)# exit

/* Verify the egress ACL creation */
Router(config)# do show access-lists ipv6
Thu Jan 25 11:45:53.823 IST
ipv6 access-list V6-EGRESS-ACL
 10 permit ipv6 any any
 20 deny udp any any
…

/* Apply the egress ACL to the GigE interface */
Router(config)# interface TenGigE 0/11/0/1
Router(config-if)# ipv6 access-group V6-EGRESS-ACL egress
Router(config-if)# commit
Thu Jan 25 11:45:12.682 IST
Router(config-if)# exit

/* Verify if the egress ACL has been successfully applied to the interface */
Router(config)# do show ipv6 interface
Thu Jan 25 11:46:43.234 IST
…
TenGigE 0/11/0/1 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
  IPv6 is enabled, link-local address is fe80::23:e9ff:fea8:a44e
  Global unicast address(es):
    2001::1, subnet is 2001::/64
  Joined group address(es): ff02::1:ff00:1 ff02::1:ffa8:a44e ff02::2
      ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
```

```
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND cache entry limit is 1000000000
  ND advertised retransmit interval is 0 milliseconds
   Hosts use stateless autoconfig for addresses.
  Outgoing access list is V6-EGRESS-ACL
  Inbound  common access list is not set, access list is not set
  Table Id is 0xe0800000
  Complete protocol adjacency: 0
  Complete glean adjacency: 0
  Incomplete protocol adjacency: 0
  Incomplete glean adjacency: 0
  Dropped protocol request: 0
  Dropped glean request: 0
...
```

ギガビット イーサネット インターフェイス上に IPv6 出力 ACL を正常に設定しました。

バンドル インターフェイス上での入力 IPv6 ACL と出力 IPv6 ACL の設定

バンドル インターフェイス上で入力 IPv6 ACL と出力 IPv6 ACL を設定するには、次の設定を使用します。

```
/* Configure a bundle interface with an IPv6 address */
Router(config)# interface Bundle-Ether 1
Router(config-if)# ipv6 address 3001::1/64
Router(config-if)# no shut
Router(config-if)# commit
Thu Jan 25 13:53:47.435 IST
Router(config-if)# exit

/* Configure an IPv6 egress ACL */
Router(config)# ipv6 access-list V6-EGRESS-ACL-bundle interface
Router(config-ipv6-acl)# 10 permit tcp any any range 3000 4000
Router(config-ipv6-acl)# 20 permit ipv6 any any
Router(config-ipv6-acl)# commit
Thu Jan 25 13:57:14.960 IST
Router(config-ipv6-acl)# exit

/* Configure an IPv6 ingress ACL to deny ingress traffic on the bundle interface */
Router(config)# ipv6 access-list V6-DENY-INGRESS-ACL
Router(config-ipv6-acl)# 10 deny ipv6 any any
Router(config-ipv6-acl)# commit
Thu Jan 25 13:59:23.198 IST
Router(config-ipv6-acl)# exit

/* Verify the egress and ingress ACL creation */
Router(config)# do show access-lists ipv6
Thu Jan 25 14:00:24.055 IST
ipv6 access-list V6-DENY-INGRESS-ACL
 10 deny ipv6 any any
ipv6 access-list V6-EGRESS-ACL-BI
 10 permit tcp any any range 3000 4000
 20 permit ipv6 any any
...

/* Apply the egress and ingress ACLs to the bundle interface */
Router(config)# interface Bundle-Ether 1
Router(config-if)# ipv6 access-group V6-EGRESS-ACL-BI egress
Router(config-if)# ipv6 access-group V6-DENY-INGRESS-ACL ingress
Router(config-if)# commit
```

```
Thu Jan 25 14:04:19.536 IST
Router(config-if)# exit

/* Verify if the ACLs have been successfully applied to the interface */
Router(config)# do show ipv6 interface
Thu Jan 25 11:46:43.234 IST
…
Thu Jan 25 14:04:51.322 IST
Bundle-Ether1 is Down, ipv6 protocol is Down, Vrfid is default (0x60000000)
  IPv6 is enabled, link-local address is fe80::1:10ff:fe87:8d04 [TENTATIVE]
  Global unicast address(es):
    3001::1, subnet is 3001::/64 [TENTATIVE]
  Joined group address(es): ff02::2 ff02::1
  MTU is 1514 (1500 is available to IPv6)
  ICMP redirects are disabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts 1
  ND reachable time is 0 milliseconds
  ND cache entry limit is 1000000000
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 160 to 240 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.
  Outgoing access list is V6-EGRESS-ACL-BI
  Inbound  common access list is not set, access list is V6-DENY-INGRESS-ACL
  Table Id is 0xe0800000
  Complete protocol adjacency: 0
  Complete glean adjacency: 0
  Incomplete protocol adjacency: 0
  Incomplete glean adjacency: 0
  Dropped protocol request: 0
  Dropped glean request: 0
```

バンドル インターフェイス上に入力 IPv6 ACL と出力 IPv6 ACLを正常に設定しました。

ACL の変更

この項では、ACL を変更するための設定の例について説明します。

```
*/ Create an Access List*/
Router(config)#ipv4 access-list acl_1

*/Add entries (ACEs) to the ACL*/
Router(config-ipv4-acl)#10 permit ip host 10.3.3.3 host 172.16.5.34
Router(config-ipv4-acl)#20 permit icmp any any
Router(config-ipv4-acl)#30 permit tcp any host 10.3.3.3
Router(config-ipv4-acl)#end

*/Verify the entries of the ACL*/:
Router#show access-lists ipv4 acl_1
ipv4 access-list acl_1
10 permit ip host 10.3.3.3 host 172.16.5.34
20 permit icmp any any
30 permit tcp any host 10.3.3.3

*/Add new entries, one with a sequence number "15" and another without a sequence number
 to the ACL. Delete an entry with the sequence number "30":*/
Router(config)#ipv4 access-list acl_1
Router(config-ipv4-acl)# 15 permit 10.5.5.5 0.0.0.255
Router(config-ipv4-acl)# no 30
```

```
Router(config-ipv4-acl)# permit 10.4.4.4 0.0.0.255
Router(config-ipv4-acl)# commit


*/When an entry is added without a sequence number, it is automatically given a sequence
 number
that puts it at the end of the access list. Because the default increment is 10, the
entry will have a sequence
number 10 higher than the last entry in the existing access list*/

*/Verify the entries of the ACL:*/
Router(config)#show access-lists ipv4 acl_1
ipv4 access-list acl_1
 10 permit ipv4 host 10.3.3.3 host 172.16.5.34

15 permit 10.5.5.5 0.0.0.255---*/newly added ACE (with the sequence number)*/
20 permit icmp any any
30 permit ipv4 10.4.4.0 0.0.0.255 any ---*/newly added ACE (without the sequence number)*/

*/The entry with the sequence number 30, that is, "30 permit tcp any host 10.3.3.3" is
deleted from the ACL*/
```

機能中の ACL を正常に変更しました。

ACL ベースの転送の設定

統合ネットワークは、音声、ビデオ、およびデータを伝送します。トラフィックによっては、ルーティング プロトコルが算出したパスを使用するのではなく、特定のパスにルーティングすることが必要になる場合があります。これを実現するには、ACL 設定にネクストホップ アドレスを指定します。これで、パケットベースで宛先アドレスをルックアップするのではなく、ACL に設定したネクストホップ アドレスを使用して指定の宛先にパケットを転送できるようになります。ACL 設定でネクストホップを使用して転送するというこの機能は、ACL ベース転送（ABF）と呼ばれます。

ACL ベース転送を使用すると、ブロードキャスト TV over IP、IP テレフォニー、データなどを対象としたサービスを複数のプロバイダーから選択することが可能になり、カフェテリア形式でインターネットにアクセスできます。サービス プロバイダーは、ユーザ トラフィックをさまざまなコンテンツ プロバイダーに迂回させることができます。

（注） リリース 6.2.2 では、GRE インターフェイスを介してルーティングされた IPv4 ABF のネクストホップがサポートされています。

（注） リリース 6.3.2 では、入力方向に適用された場合、IPv6 ABF ACL はバンドル インターフェイスで動作しません。

機能概要

- ABF は入力 ACL でのみサポートされています。
- ABF はネクストホップの変更をサポートしています。ネクストホップの変更、ネクストホップの削除、または既存のネクストホップ間での変更が可能です。

（注） ネクストホップがデフォルトの VRF 内にある場合を除き、ACE ルールの定義時にすべてのネクストホップの VRF を指定する必要があります。これにより、パケットがネクストホップへの適切なパスを取得するようになります。

- VRF 認識型 ABF は、最大 3 つのネクストホップがある IPv4 と IPv6 でサポートされています。
- ABF は ACL ベースであるため、ACL 内の既存のルール（ACE）に一致しないパケットはデフォルトの ACL ルール（すべてドロップ）に従います。ACL が（セキュリティ上の理由ではなく）ABF リダイレクトにのみ使用されている場合は、ACL の末尾（最下位のユーザ プライオリティ）に明示的な ACL ルールを組み込んで、すべてのトラフィックを照合して「許可」します。これにより、ABF ルールに一致しないすべてのトラフィックが許可され、通常通りに転送されるようになります。
- ABF は、許可ルールのみでサポートされています。
- VRF-select（ネクストホップに対して VRF のみが設定されている場合）はサポートされていません。
- ABF のデフォルトのルートはサポートされていません。
- 低速パスでは ABF がサポートされないため、NPU からラインカード CPU へと入力方向にパントされたパケットは ABF では処理されません。通常、これらのパケットは、ソフトウェア データプレーンによる送信先アドレスのルックアップに基づいて転送されます。これらのタイプのパケットには、IPv4 オプション、IPv6 拡張ヘッダー、および収集（未解決/不完全）隣接関係宛のパケットなどがありますが、これらに限定されません。
- ローカル IP インターフェイス宛のパケット（「for-us」パケット）は、ABF アクションが含まれているルールに一致した場合はリダイレクトの対象になります。これは、「for-us」パケットへの一致を避けるために十分な具体的なルールを作成するか、または ABF ルールの照合よりも前に（高いプライオリティの）明確な許可 ACL ルールを ACL に配置することで防ぐことができます。

設定例

ACL ベースの転送を設定するには、次のタスクを実行します。

```
/* Enter IPv4 access list configuration mode and configure an ACL: */
router# configure
router(config)# ipv4 access-list abf-acl
```

```
/* Set the conditions for the ACL and configure ABF: */
/* The next hop for this entry is specified. */
router(config-ipv4-acl)# 10 permit ipv4 192.168.18.0 0.255.255.255 any nexthop1 ipv4
192.168.20.2
router(config-ipv4-acl)# 15 permit ipv4 192.168.21.0 0.0.0.255 any
router(config-ipv4-acl)# 20 permit ipv4 192.168.22.0 0.0.255.255 any nexthop1 ipv4
192.168.23.2
/* More than two nexthops */
router(config-ipv4-acl)# 25 permit tcp any range 2000 3000 any range 4000 5000 nexthop1
 ipv4 192.168.23.1 nexthop2 ipv4 192.168.24.1 nexthop3 ipv4 192.168.25.1

/* VRF support on ABF */
router(config-ipv4-acl)# 30 permit tcp any eq www host 192.168.12.2 precedence immediate
 nexthop1 vrf vrf1_ipv4 ipv4 192.168.13.2 nexthop2 vrf vrf1_ipv4 ipv4 192.168.14.2

router(config-ipv4-acl)# 35 permit ipv4 any any

router(config-ipv4-acl)# commit

/* (Optional) Display ACL information: */
router# show access-lists ipv4 abf-acl
```

実行コンフィギュレーション

```
ipv4 access-list abf-acl
10 permit ipv4 192.168.18.0 0.255.255.255 any nexthop1 192.168.20.2
15 permit ipv4 192.168.21.0 0.0.0.255 any
20 permit ipv4 192.168.22.0 0.0.255.255 any nexthop1 192.168.23.2
25 permit tcp any range 2000 3000 any range 4000 5000 nexthop1 ipv4 192.168.23.1 nexthop2
 ipv4 192.168.24.1 nexthop3 ipv4 192.168.25.1
30 permit tcp any eq www host 192.168.12.2 precedence immediate nexthop1 vrf vrf1_ipv4
ipv4 192.168.13.2 nexthop2 vrf vrf1_ipv4 ipv4 192.168.14.2
35 permit ipv4 any any
commit
!
```

確認

ABF 内の IP ネクストホップの状態を確認し、その予想されるネクストホップが起動するようにするには、次のコマンドを使用します。

```
Router# show access-lists ipv4 abf nexthops client pfilter_ea location 0/0/CPU0
Wed Jan 24 14:18:58.667 UTC

ACL name : abf-acl
ACE seq.   NH-1   NH-2   NH-3
 --------- --------------- --------------- ---------------
 10        192.168.13.2
 status          UP
 at status  Not Present
 exist      No
 vrf        default
 track
 pd     ctx  Present
 25   192.168.14.2    192.168.11.1  192.168.12.1
 status   UP   Down   Down
 at status  Not Present  Not Present  Not Present
 exist   No   Yes   Yes
 vrf   default  default  default
 track
```

```
pd ctx   Present  Not present  Not present
30   192.168.15.1  192.168.12.7
status   Unknown  Unknown
at status  Not Present  Not Present
exist   No   Yes
vrf   vrf1_ipv4  vrf1_ipv4
track
pd ctx   Not present  Not present
```

ABFがラインカードのインターフェイスに現在付加されているかどうかを確認するには、次のコマンドを使用します。

```
show access-lists usage pfilter location all
```

ブリッジ仮想インターフェイスの ACL

ブリッジ仮想インターフェイス（BVI）は、ルータ上のルーティング ドメインとブリッジング ドメイン間にブリッジを提供します。BVI は IP アドレスで設定され、通常のルーテッド インターフェイスとして動作します。BVI 上に ACL を設定して、そのインターフェイスを使用するネットワークに対するトラフィックをフィルタリングできます。

（注） BVIインターフェイスがブリッジドメインに含まれていない場合は、BVIインターフェイスに付加されている ACL を削除しないでください。後で BVI インターフェイスをブリッジ ドメインに追加した場合は、トラフィックがトロップされます。

BVI 上での ACL の設定による TCAM 消費の増加

BVI に ACL が設定されている場合、TCAM リソースの消費は次のように影響されます。

- ACL が BVI インターフェイスに付加されていると、TCAM エントリは物理インターフェイスのメンバーシップとは関係なく、すべてのライン カード上でプログラミングされます。これにより、BVI メンバ インターフェイスがないライン カードでも TCAM リソースの消費が増加することになります。
- ACL が BVI インターフェイスに付加されていると、TCAM エントリは物理インターフェイスのメンバーシップとは関係なく、ライン カードのすべての NPU 上でプログラミングされます。これにより、BVI メンバインターフェイスがない NPU でも TCAM リソースの消費が増加することになります。
- 入力 ACL では、同じ ACL の TCAM エントリが同じ NPU 上のインターフェイス間で共有されます。
- 出力 ACL では、同じ ACL の TCAM エントリはすべてのインターフェイスに一意です。これにより、TCAM リソースの消費が増加します。

BVI 上での ACL 設定の制約事項

BVI 上での ACL の設定に進む前に、次の制約事項を認識しておく必要があります。

- BVI では、出力 IPv6 ACL はサポートされていません。
- **hw-module** コマンドを使用して BVI 上で出力 IPv4 ACL を有効にすると、その ACL では他のインターフェイス タイプがサポートされません（このモードの ACL では、非 BVI インターフェイスはサポートされません）。

BVI での IPv4 出力 ACL 設定の前提条件

デフォルトでは、BVI 上の IPv4 出力 ACL は無効になっており、ACL が BVI に付加されていても ACL のフィルタリングは実行されません。そのため、**hw-module** コマンドを使用して、ライン カードのリロード時に ACL を有効にします。

（注） IPv4 と IPv6 の入力 ACL はこの設定を必要としません。

次の設定を使用して、ハードウェア 上の BVI で IPv4 出力 ACL を有効にし、ライン カードをリロードします。

```
/* Enable an IPv4 egress ACL on BVI */
RP/0/RP0/CPU0:router(config)# hw-module profile acl egress layer3 interface-based
/* Enable permit statistics for the egress ACL (by default, only deny statistics are
shown)*/
RP/0/RP0/CPU0:router(config)# hw-module profile stats acl-permit
RP/0/RP0/CPU0:router(config)# commit
RP/0/RP0/CPU0:router(config)# end
RP/0/RP0/CPU0:router# reload location all
Wed Apr 5 23:05:46.193 UTC
Proceed with reload? [confirm]
```

設定

次の項では、BVI 上で IPv4 の入力 ACL と出力 ACL を設定する手順について説明します。

BVI で IPv4 の入力 ACL と出力 ACL を設定するには、次の設定例を使用した手順を実行します。

1. グローバル コンフィギュレーション モードを開始し、IPv4 入力 ACL を設定します。

```
RP/0/RP0/CPU0:router(config)# ipv4 access-list v4-acl-ingress
RP/0/RP0/CPU0:router(config-ipv4-acl)# 10 permit tcp any 10.1.1.0/24 dscp cs6
RP/0/RP0/CPU0:router(config-ipv4-acl)# 20 deny udp any any eq ssh
RP/0/RP0/CPU0:router(config-ipv4-acl)# 30 permit ipv4 any any
RP/0/RP0/CPU0:router(config-ipv4-acl)# commit
RP/0/RP0/CPU0:router(config-ipv4-acl)# exit
```

2. IPv4 出力 ACL を設定します。

```
RP/0/RP0/CPU0:router(config)# ipv4 access-list v4-acl-egress
RP/0/RP0/CPU0:router(config-ipv4-acl)# 10 deny ipv4 any any fragments log
RP/0/RP0/CPU0:router(config-ipv4-acl)# 20 deny tcp any any ack
RP/0/RP0/CPU0:router(config-ipv4-acl)# 30 permit ipv4 any any
RP/0/RP0/CPU0:router(config-ipv4-acl)# commit
RP/0/RP0/CPU0:router(config-ipv4-acl)# exit
```

3. BVI にマップする必要があるギガビット イーサネット インターフェイスを設定し、レイヤ 2 トランスポートに対して有効にします。

```
RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/0/0/0
RP/0/RP0/CPU0:router(config-if)# l2transport
RP/0/RP0/CPU0:router(config-if-l2)# commit
```

4. 入力 ACL と出力 ACL を BVI に付加します。

```
RP/0/RP0/CPU0:router(config)# interface BVI1
RP/0/RP0/CPU0:router(config-if)# ipv4 access-group v4-acl-ingress ingress
RP/0/RP0/CPU0:router(config-if)# ipv4 access-group v4-acl-egress egress
RP/0/RP0/CPU0:router(config-if)# commit
RP/0/RP0/CPU0:router(config-if)# exit
```

5. ギガビット イーサネット インターフェイスと BVI でブリッジ ドメインを設定します。

```
RP/0/RP0/CPU0:router(config)# l2vpn
RP/0/RP0/CPU0:router(config-l2vpn)# bridge group BG1
RP/0/RP0/CPU0:router(config-l2vpn-bg)# bridge-domain B1
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# interface GigabitEthernet 0/0/0/0
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd-ac)# routed interface BVI1

RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# commit
RP/0/RP0/CPU0:router(config-l2vpn-bg-bd)# exit
RP/0/RP0/CPU0:router(config-l2vpn-bg)# exit
RP/0/RP0/CPU0:router(config-l2vpn)# exit
```

6. 設定が正常にコミットされていることを確認します。

```
RP/0/RP0/CPU0:router(config)# show run
...
!
ipv4 access-list v4-acl-egress
 10 deny ipv4 any any fragments log
 20 deny tcp any any ack
 30 permit ipv4 any any
!
ipv4 access-list v4-acl-ingress
 10 permit tcp any 10.1.1.0/24 dscp cs6
 20 deny udp any any eq ssh
 30 permit ipv4 any any
!
interface GigabitEthernet0/0/0/0
 l2transport
 !
!
interface BVI1
ipv4 address 209.165.200.224/27
 ipv4 access-group v4-acl-ingress ingress
 ipv4 access-group v4-acl-egress egress

!
l2vpn
 bridge group BG1
  bridge-domain B1
   interface GigabitEthernet0/0/0/0
   !
   routed interface BVI1
  !
```

```
 !
!
end
```

7. エグゼクティブ特権モードに移行して、ACL が機能していることを確認します。

```
RP/0/RP0/CPU0:router# show access-lists interface bvi1
Tue May 9 10:01:25.732 EDT
Input ACL (common): GigabitEthernet 0/0/0/0 (interface): v4-acl-ingress
Output ACL: v4-acl-egress

RP/0/RP0/CPU0:router# show access-lists summary
Tue May 9 10:02:01.167 EDT
ACL Summary:
Total ACLs configured: 2
Total ACEs configured: 6


RP/0/RP0/CPU0:router# show access-lists ipv4 v4-acl-egress hardware egress location
 0/0/CPU0
ipv4 access-list v4-acl-egress
10 deny ipv4 any any fragments log (15214 matches)
20 deny tcp any any ack (15214 matches)
30 permit ipv4 any any (15214 matches)
```

設定した ACL、ACE の総数（ACL ごとに 3 つ）、およびハードウェア内の ACE の一致も出力に明確に表示されます。

IPv4 の入力および出力 ACL が BVI 上に正常に設定されました。

フラグメント制御を使用した ACL の設定

非フラグメント パケットとパケットの先頭フラグメントは、IP 拡張アクセス リストで処理されていましたが（このアクセスリストを適用した場合）、先頭以外のフラグメントはデフォルトで許可されていました。しかし、現在では、フラグメント制御機能付きの IP 拡張アクセスリストによって、パケットの先頭以外のフラグメントもさらにきめ細かく制御できるようになりました。この機能を使用すると、IP 拡張アクセス リストを適用するときに、パケットの先頭以外の IP フラグメントを調べるかどうかを指定できます。

先頭以外のフラグメントにはレイヤ 3 情報だけしか含まれていないので、レイヤ 3 情報のみを含んでいるアクセスリスト エントリを先頭以外のフラグメントに適用できるようになりました。フラグメントにはフィルタリングに必要な情報がすべて含まれているため、アクセスリスト エントリをパケットのフラグメントに適用できるのです。

この機能により、オプションの **fragments** キーワードが、IP アクセス リスト コマンドの **deny** と **permit** に追加されます。アクセスリスト エントリに **fragments** キーワードを指定すると、その特定のアクセスリストエントリはパケットの先頭以外のフラグメントにのみ適用され、指定に応じてフラグメントが許可または拒否されます。

fragments キーワードの有無に応じたアクセスリスト エントリの動作の概要は、以下のとおりです。

アクセス リスト エントリの状態	結果

fragments キーワードがなく、すべてのアクセス リスト エントリ情報が一致する	アクセス リスト エントリにレイヤ 3 情報のみが含まれている場合：
	• エントリは、非フラグメント パケット、先頭フラグメント、先頭以外のフラグメントに適用されます。
	
	アクセス リスト エントリにレイヤ 3 情報とレイヤ 4 情報が含まれている場合：
	• エントリは、非フラグメント パケットと先頭フラグメントに適用されます。
	  • エントリが一致し、かつ **permit** ステートメントである場合、パケットまたはフラグメントは許可されます。
	  • エントリが一致し、かつ **deny** ステートメントである場合、パケットまたはフラグメントは拒否されます。
	
	• エントリは、次の方法で先頭以外のフラグメントにも適用されます。先頭以外のフラグメントにはレイヤ 3 情報のみが含まれているため、アクセスリスト エントリのレイヤ 3 部分のみが適用されます。アクセス リスト エントリのレイヤ 3 の部分が一致し、
	  • エントリが **permit** ステートメントの場合、先頭以外のフラグメントが許可されます。
	  • エントリが **deny** ステートメントの場合は、次のアクセス リスト エントリが処理されます。
	
	（注） 先頭以外のフラグメントと、非フラグメントまたは先頭フラグメントとでは、deny ステートメントの処理方法が異なります。

fragments キーワードがあり、すべてのアクセス リスト エントリ情報が一致する	アクセスリスト エントリは、先頭以外のフラグメントにのみ適用されます。 （注） レイヤ 4 情報を含むアクセスリスト エントリに **fragments** キーワードは設定できません。

すべてのアクセスリスト エントリに **fragments** キーワードを追加しないでください。IP パケットの先頭フラグメントは非フラグメントと見なされ、それ以降のフラグメントとは独立して扱われるためです。先頭フラグメントは、**fragments** キーワードが含まれているアクセス リストの permit または deny エントリとは一致しないため、パケットは次のアクセスリスト エントリと比較されます。この比較は、**fragments** キーワードが含まれていないアクセスリスト エント

リによってパケットが許可または拒否されるまで続きます。したがって、deny エントリごとに、2 つのアクセス リスト エントリが必要になる場合があります。ペアの最初の deny エントリは **fragments** キーワードを含んでおらず、先頭フラグメントに適用されます。ペアの 2 番目の deny エントリは **fragments** キーワードを含んでおり、以降のフラグメントに適用されます。同じホストに対して複数の **deny** アクセスリスト エントリがあり、それぞれのレイヤ 4 ポートが異なる場合は、そのホスト用として、**fragments** キーワードを含む deny アクセスリスト エントリを 1 つだけ追加する必要があります。このように、パケットのすべてのフラグメントは、アクセス リストによって同様に扱われます。

IP データグラムのパケット フラグメントは個々のパケットと見なされ、各フラグメントはアクセス リスト アカウンティングとアクセス リスト違反カウントの 1 つのパケットとして個別にカウントされます。

（注） アクセス リストおよび IP フラグメントに関するあらゆるケースを **fragments** キーワードで解決できるわけではありません。

（注） ACL 処理の範囲内で、レイヤ 3 情報は IPv4 ヘッダー内のフィールド（送信元、宛先、プロトコルなど）を参照します。レイヤ 4 情報は、TCP または UDP の送信元ポートと宛先ポート、TCP のフラグ、ICMP のタイプとコードなど、IPv4 ヘッダーの後ろに含まれるその他のデータを参照します。

フラグメント タイプでの照合を実行するための IPv4 ACL の設定

ほとんどの DoS（サービス妨害）攻撃は、フラグメント化されたパケットでネットワークをフラッディングさせることで機能します。ネットワーク内のパケットの着信フラグメントをフィルタリングすることで、このような攻撃に対する保護レイヤを余分に追加できます。

フラグメント タイプを照合する IPv4 ACL を設定し、適切なアクションを実行できます。次の設定例ではさまざまなフラグメント オプションで使用できます。

```
/* Enter the global configuraton mode and configure an IPv4 access list */
Router# config
Router(config)# ipv4 access-list TEST
Router(config-ipv4-acl)# 10 permit tcp any any

/* Configure an ACE to match on the dont-fragment flag (indicates a non-fragmented packet)

 and forward the packet to the default (pre-configured) next hop  */
Router(config-ipv4-acl)# 20 permit tcp any any fragment-type dont-fragment default

/* Configure an ACE to match on the is-fragment flag (indicates a fragmented packet)
 and forward the packet to a next hop of 10.10.10.1  */
Router(config-ipv4-acl)# 30 permit udp any any fragment-type is-fragment nexthop1 ipv4
10.10.10.1

/* Configure an ACE to match on the first-fragment flag (indicates the first fragment
of a fragmented packet)
```

```
 and forward the packet to a next hop of 20.20.20.1  */
Router(config-ipv4-acl)# 40 permit ospf any any fragment-type first-fragment nexthop1
ipv4 20.20.20.1


/* Configure an ACE to match on the last-fragment flag (indicates the last fragment of
a fragmented packet)
 and forward the packet to a next hop of 30.30.30.1  */
Router(config-ipv4-acl)# 50 permit icmp any any fragment-type last-fragment nexthop1
ipv4 30.30.30.1
Router(config-ipv4-acl)# commit
```

使用例：最初のフラグメントと最後のフラグメントを照合する IPv4 ACL の設定

この項では、パケットの最初のフラグメントの場合はそのフラグメントを転送し、パケットの最後のフラグメントの場合はそのフラグメントを破棄するように ACL を設定する使用例について説明します。

この設定では、ACLがフラグメント オフセット値（最初のフラグメントの場合は「0」）を確認します。そのフラグメントがパケットの最初のフラグメントの場合は、パケットが転送されます。そのフラグメントがパケットの最後のフラグメントの場合は、インターフェイスでドロップされます。

```
/* Enter the global configuraton mode and configure an IPv4 access list */
Router# config
Thu Jan 11 11:56:27.221 IST
Router(config)# ipv4 access-list ACLFIRSTFRAG

/* Configure an ACE to match on the first fragment.
If the fragment offset value equals 0, the fragment is forwarded to the 192.168.1.2 next
 hop */
Router(config-ipv4-acl)# 10 permit tcp any any fragment-type first-fragment nexthop1
ipv4 192.168.1.2

/* Configure an ACE to match on the last fragment, and drop the fragment at the interface.
 */
Router(config-ipv4-acl)# 20 deny tcp any any fragment-type last-fragment
Router(config-ipv4-acl)# commit
Thu Jan 11 12:01:33.297 IST

/* Validate the configuration */
Router(config-ipv4-acl)# do show access-lists
Thu Jan 11 12:05:23.646 IST
ipv4 access-list ACLFIRSTFRAG
 10 permit tcp any any fragment-type first-fragment nexthop1 ipv4 192.168.1.20
 20 deny tcp any any fragment-type last-fragment
```

フラグメント タイプを照合する IPv4 ACL を正常に設定しました。

ACL でのフラグメント オフセットによる一致

フラグメント オフセット値でパケットをフィルタリングするようにアクセス コントロール リスト（ACL）のルールを作成できます。パケットが permit ステートメントまたは deny ステートメントの条件に一致するかどうかにより、パケットはインターフェイスでそれぞれ処理され

るか、またはドロップされます。フラグメント オフセット フィルタリングは、圧縮モードの ACL を使用して入力方向でのみサポートされています。

この機能の詳細については、『*IP Addresses and Services Configuration Guide for Cisco NCS 540 Series Routers*』の「*Implementing Access Lists and Prefix Lists*」の章を参照してください。詳細なコマンド リファレンスについては、『*IP Addresses and Services Command Reference for Cisco NCS 5500 Series and NCS 540 Series Routers*』の「*Access List Commands*」の章を参照してください。

フラグメント オフセットによる ACL 照合の設定

ACL でフラグメント オフセット一致を設定するには、IPv4 または IPv6 アクセスリスト コンフィギュレーション モードの **permit** コマンドまたは **deny** コマンドで **fragment-offset** オプションを使用します。

（注） フラグメント オフセット フィルタリングの場合は、特定の ACL を圧縮レベルが 3 のインターフェイスに付加する必要があります。そうしないと、設定が拒否されます。

設定

次に、IPv4 ヘッダーごとのフラグメント オフセットに基づいて ACL ルールを設定する例を示します。ここでは、パケットの IPv4 ヘッダー内のフラグメント オフセットが 300 ～ 400 の範囲内にある場合にのみ、パケットが許可されます。値 *300* ～ *400* は 8 バイトの単位に基づいており、これは *2400* ～ *3200* バイトのフラグメント オフセットと同じです。

```
/* Configure ACL */
Router# configure
Router(config)# ipv4 access-list fragment-offset-acl
Router(config-ipv4-acl)# 10 permit ipv4 any any fragment-offset range 300 400
Router# commit

/* Attach the ACL to the interface */
Router# configure
Router(config)# interface Bundle-Ether70
Router(config-if)# ipv4 access-group fragment-offset-acl ingress compress level 3
Router# commit
```

実行コンフィギュレーション

```
ipv4 access-list fragment-offset-acl
 10 permit ipv4 any any fragment-offset range 300 400
!

interface Bundle-Ether70
 ipv4 address 192.0.2.1 255.255.255.0
 ipv6 address 2001:DB8::1:1::1/48
 ipv4 access-group fragment-offset-acl ingress compress level 3
!
```

ACL でのフラグメント オフセットの一致の確認

```
Router#
show access-lists ipv4 fragment-offset-acl usage pfilter loc 0/0/CPU0

Wed Apr 12 19:49:54.457 UTC
Interface : Bundle-Ether70
    Input  ACL : Common-ACL : N/A  ACL : fragment-offset-acl  (comp-lvl 3)
    Output ACL : N/A



Router#
show access-lists ipv4 fragment-offset-acl hardware ing int Bundle-Ether70 loc 0/0/CPU0

Wed Apr 12 19:51:07.837 UTC
ipv4 access-list fragment-offset-acl
 10 permit ipv4 any any fragment-offset range 300 400
```

関連コマンド

- ipv4 access-list
- ipv6 access-list
- deny（IPv4）
- deny（IPv6）
- fragment-offset
- permit（IPv4）
- permit（IPv6）

IP パケット長による ACL フィルタリングの設定

入力インターフェイスでパケット長を使用してパケットをフィルタリングするようにアクセス コントロール リストを設定できます。パケットが permit ステートメントまたは deny ステートメントのパケット長条件と一致するかどうかに応じて、パケットはインターフェイスでそれぞれ処理またはドロップされます。

ACL でパケット長フィルタリングを設定するには、IPv4 または IPv6 アクセスリスト コンフィギュレーション モードの **permit** コマンドまたは **deny** コマンドで **packet-length** オプションを使用します。

制約事項

ACL でのパケット長フィルタ リング機能には次の制約事項があります。

- パケット長フィルタリングは、シンプル（非圧縮）ACL とハイブリッド（圧縮）ACL の両方で入力方向のみがサポートされています。

- IPv6 のパケット長フィルタ リングは、ハイブリッド ACL でのみサポートされています。シンプル ACL ではサポートされていません。
- IPv4 のシンプル ACL でサポートされているのは、量子化された（16 で割り切れる値）パケット長フィルタリングのみです。

パケット長を使用してフィルタリングするためのシンプルな IPv4 ACL の設定

シンプルな ACL を設定して IPv4 ネットワーク内のパケット長でフィルタリングするには、次のステップを実行します。

1. グローバル コンフィギュレーション モードを開始し、パケット長値でパケットをフィルタリングするシンプルな IPv4 アクセス リストを設定します。

   次の特定の例では、指定したパケット長の条件に一致するパケットのみを処理する一連のステートメントを設定します。その他のパケットは、この ACL が入力インターフェイスに適用された時点でドロップされます。

```
RP/0/RP0/CPU0:router# config
RP/0/RP0/CPU0:router(config)# ipv4 access-list pktlen-v4
RP/0/RP0/CPU0:router(config-ipv4-acl)# 10 permit tcp any any packet-length eq 1664
RP/0/RP0/CPU0:router(config-ipv4-acl)# 20 permit udp any any packet-length range 1600
 2000
RP/0/RP0/CPU0:router(config-ipv4-acl)# 30 deny ipv4 any any
```

2. ACL をコミットし、IPv4 ACL コンフィギュレーション モードを終了します。

```
RP/0/RP0/CPU0:router(config-ipv4-acl)# commit
RP/0/RP0/CPU0:router(config-ipv4-acl)# end
```

3. 必要なイーサネット インターフェイスに ACL を適用します。

```
RP/0/RP0/CPU0:router(config)# interface Te0/0/0/0
RP/0/RP0/CPU0:router(config-if)# ipv4 access-group pktlen-v4 ingress
```

4. 設定をコミットし、インターフェイス コンフィギュレーション モードを終了します。

```
RP/0/RP0/CPU0:router(config-if)# commit
RP/0/RP0/CPU0:router(config-if)# end
```

5. 設定を確認します。

```
RP/0/RP0/CPU0:router# show access-lists pktlen-v4


ipv4 access-list pktlen-v4
10 permit ipv4 host 10.0.0.10 any packet-length lt 1008
20 permit ipv4 host 10.0.0.9 any packet-length gt 992
```

6. ハードウェアで ACL の一致を確認します。

```
RP/0/RP0/CPU0:router# show access-lists pktlen-v4 hardware ingress location 0/0/CPU0
```

```
ipv4 access-list pklen-v4
10 permit ipv4 host 10.0.0.10 any packet-length lt 1008
20 permit ipv4 host 10.0.0.9 any packet-length gt 992
```

パケット長でフィルタリングするためのシンプルな IPv4 ACL を正常に設定しました。

パケット長を使用してフィルタリングするための拡張 IPv4 ACL の設定

拡張 ACL を設定して IPv4 ネットワーク内のパケット長でフィルタリングするには、次のステップを実行します。

1. グローバル コンフィギュレーション モードを開始し、拡張 ACL を設定するオブジェクト グループを作成します。

```
RP/0/RP0/CPU0:router(config)# object-group network ipv4 netobject1
RP/0/RP0/CPU0:router(config-object-group-ipv4)# 50.0.0.0/24
RP/0/RP0/CPU0:router(config-object-group-ipv4)# commit
```

2. グローバル コンフィギュレーション モードから、パケット長値でパケットをフィルタリングする IPv4 アクセス リストを設定します。

   次の特定の例では、指定したパケット長の条件に一致するパケットのみを処理するステートメントを設定します。その他のパケットは、この ACL が入力インターフェイスに適用された時点でドロップされます。

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ipv4 access-list scaled_acl1
RP/0/RP0/CPU0:router(config-ipv4-acl)# 10 permit ipv4 net-group netobject1 any
packet-length eq 1000
```

3. ACL をコミットし、IPv4 ACL コンフィギュレーション モードを終了します。

```
RP/0/RP0/CPU0:router(config-ipv4-acl)# commit
RP/0/RP0/CPU0:router(config-ipv4-acl)# end
```

4. 必要なギガビット イーサネット インターフェイスに ACL を適用します。

```
RP/0/RP0/CPU0:router(config)# interface Te0/0/0/3
RP/0/RP0/CPU0:router(config-if)#ipv4 access-group scaled_acl1 ingress
```

5. 設定をコミットし、インターフェイス コンフィギュレーション モードを終了します。

```
RP/0/RP0/CPU0:router(config-if)# commit
RP/0/RP0/CPU0:router(config-if)# end
```

6. 設定を確認します。

```
RP/0/RP0/CPU0:router# show access-lists scaled_acl1
ipv4 access-list scaled_acl1
10 permit ipv4 net-group netobject1 any packet-length eq 1000
```

7. ハードウェアで ACL の一致を確認します。

```
RP/0/RP0/CPU0:router# show access-lists scaled_acl1 hardware ingress location 0/0/CPU0
ipv4 access-list scaled_acl1
10 permit ipv4 net-group netobject1 any packet-length eq 1000 (1500 hw matches)
```

パケット長でフィルタリングするための拡張 IPv4 ACL を正常に設定しました。

パケット長を使用してフィルタリングするための拡張 IPv6 ACL の設定

拡張 ACL を設定して IPv6 ネットワーク内のパケット長でフィルタリングするには、次のステップを実行します。

1. グローバル コンフィギュレーション モードを開始し、拡張 ACL を設定するオブジェクト グループを作成します。

```
RP/0/RP0/CPU0:router(config)# object-group  network ipv6 netobject2
RP/0/RP0/CPU0:router(config-object-group-ipv6)# 2001::0/128
RP/0/RP0/CPU0:router(config-object-group-ipv6)# commit
```

2. グローバル コンフィギュレーション モードから、パケット長値でパケットをフィルタリングする拡張 IPv6 アクセス リストを設定します。

   次の特定の例では、指定したパケット長の条件に一致するパケットのみを処理するステートメントを設定します。その他のパケットは、この ACL が入力インターフェイスに適用された時点でドロップされます。

```
RP/0/RP0/CPU0:router(config)# ipv6 access-list scaled_acl2
RP/0/RP0/CPU0:router(config-ipv6-acl)# 10 permit ipv6 net-group netobject2 any
packet-length eq 1000
RP/0/RP0/CPU0:router(config-ipv6-acl)# commit
```

3. ACL をコミットし、IPv6 ACL コンフィギュレーション モードを終了します。

```
RP/0/RP0/CPU0:router(config-ipv6-acl)# commit
RP/0/RP0/CPU0:router(config-ipv6-acl)# end
```

4. 必要なギガビット イーサネット インターフェイスに ACL を適用します。

```
RP/0/RP0/CPU0:router# config
RP/0/RP0/CPU0:router(config)# interface Te/0/0/0/3
RP/0/RP0/CPU0:router(config-if)# ipv6 access-group scaled_acl2 ingress
```

5. 設定をコミットし、インターフェイス コンフィギュレーション モードを終了します。

```
RP/0/RP0/CPU0:router(config-if)# commit
RP/0/RP0/CPU0:router(config-if)# end
```

6. 設定を確認します。

```
RP/0/RP0/CPU0:router# show access-lists ipv6 scaled_acl2
ipv6 access-list scaled_acl2
10 permit ipv6 net-group netobject2 any packet-length eq 1000
```

7. ハードウェアで ACL の一致を確認します。

```
RP/0/RP0/CPU0:router# show access-lists ipv6 scaled_acl2 hardware ingress location
0/0/CPU0
```

```
ipv6 access-list scaled_acl2
10 permit ipv6 net-group netobject2 any packet-length eq 1000 (2000 hw matches)
```

パケット長でフィルタリングするための拡張 IPv6 ACL を正常に設定しました。

オブジェクトグループ ACL の概要

オブジェクトグループ ACL を使用すると、ユーザ、デバイス、またはプロトコルをグループに分類できるため、グループ レベルのアクセス コントロール ポリシーを設定できます。個別の IP アドレス、プロトコル、およびポート番号を複数の ACE に指定する代わりに、単一の ACL にオブジェクト グループを指定できます。

この機能は、現在、数百ものACLが含まれている大規模なネットワークには非常に有益です。オブジェクトグループACL機能を使用することで、ACLあたりのACEの数が大幅に削減します。また、オブジェクトグループACLは判読性が高く、従来のACLよりも簡単に管理できます。従来の ACL の代わりに オブジェクトグループ ACL を使用することで、TCAM に必要なストレージが最適化されます。

オブジェクトグループ ACL のタイプ

Cisco IOS XR では 2 つのタイプの オブジェクトグループ ACL を作成できます。

- **ネットワーク オブジェクトグループ ACL**：ホスト IP アドレスとネットワーク IP アドレスから構成されます。
- **ポート オブジェクトグループ ACL**：ポートとサポートしているレイヤ 3/レイヤ 4 プロトコルのグループから構成されます。

ACL の圧縮

オブジェクトグループACLは圧縮を使用して多数のACEに対応します。圧縮は、ACEの次の3つのフィールドを圧縮することで実行されます。

- 送信元 IP プレフィックス
- 送信先 IP プレフィックス
- 送信元ポート番号

Cisco NCS 540 シリーズ ルータは、入力インターフェイス上の ACL のアクセスグループ設定で 2 つの圧縮レベルのみをサポートしています。

- **圧縮レベル 0**：圧縮は ACE フィールドでは実行されません。

  このモードでは、オブジェクトグループ ACL は従来の ACL のように動作します。内部 TCAM リソースを利用するため、ACL の処理に必要なシステム リソースと時間に膨大な影響を与えます。

- **圧縮レベル 3**：ACE の 3 つフィールド（送信元 IP、送信先 IP、および送信元ポート）すべてが圧縮されます。

このモードでは、プレフィックスのルックアップには外部 TCAM 、ACE のルックアップには内部 TCAM を使用します。このモードは、16 ビット ベースのパケット長フィルタリングとフラグメント オフセット フィルタリングをサポートしています。

オブジェクトグループ ACL の設定

はじめる前に

オブジェクトグループ ACL に適用される次の情報を把握しておく必要があります。

- 従来の ACL とオブジェクトグループ ACL の両方を含む ACL を設定できます。
- オブジェクト グループや、そのオブジェクト グループを参照する ACE を定義し直すことなく、オブジェクト グループ内のオブジェクトをダイナミックに変更できます。
- オブジェクトグループ ACL は、送信元グループまたは送信先グループ、あるいは送信元と送信先のグループの両方を使用して複数回設定できます。

制約事項

オブジェクトグループ ACL の設定には、次の制約事項があります。

- オブジェクトグループ ACL は、インターフェイスにのみ設定できます。SSH、SNMP、NTP などのアプリケーションで使用または参照することはできません。
- オブジェクトグループを削除するには、最初にすべての ACL から削除する必要があります。
- QoS ポリシーとともにオブジェクトグループ ACL を設定することはできません。
- オブジェクトグループ ACL は、ポリシー ベースの設定ではサポートされません。
- リリース 6.2.1 以降、ネストされたオブジェクトグループはサポートされていません。

ネットワーク オブジェクトグループ ACL の設定

ネットワーク オブジェクト グループには単一または複数のネットワーク オブジェクトを含めることができます。

設定

次の一連の設定ステートメントを使用して、IPv4アドレスのネットワーク オブジェクトグループ ACL を設定します。

```
/* From the global configuration mode, create a network object group. */
Router(config)# object-group network ipv4 netobj1
Router(config-object-group-ipv4)# description my-network-object
Router(config-object-group-ipv4)# host 10.1.1.1
Router(config-object-group-ipv4)# 10.2.1.0 255.255.255.0
Router(config-object-group-ipv4)# range 10.3.1.10 10.3.1.50
```

```
/* Create an access list referencing the object group. */
Router(config)# ipv4 access-list network-object-acl permit ipv4 net-group netobj1 any

/* Apply the access list containing the object group to the desired interface and commit
 your configuration. */
Router(config)# interface TenGigE0/0/0/10/3
Router(config-if)# ipv4 address 1.1.1.1/24
Router(config-if)# no shut
Router(config-if)# ipv4 access-group network-object-acl ingress compress level 3
Router(config-if)# commit
Tue Mar 28 10:23:34.106 IST

RP/0/0/CPU0:Mar 28 10:37:48.570 : ifmgr[397]: %PKT_INFRA-LINK-3-UPDOWN : Interface
TenGigE0/0/0/10/3 , changed state to Down
RP/0/0/CPU0:Mar 28 10:37:48.608 : ifmgr[397]: %PKT_INFRA-LINK-3-UPDOWN : Interface
TenGigE0/0/0/10/3 , changed state to Up

Router(config-if)# exit
```

次の一連の設定ステートメントを使用して、IPv6アドレスのネットワーク オブジェクトグループ ACL を設定します。

```
/* From the global configuration mode, create a network object group. */
Router(config)# object-group network ipv6 netobj1
Router(config-object-group-ipv6)# description my-network-object
Router(config-object-group-ipv6)# host 2001:DB8:1::1
Router(config-object-group-ipv6)# 2001:DB8::1 2001:DB8:0:ABCD::1
Router(config-object-group-ipv6)# range 2001:DB8::2 2001:DB8::5

/* Create an access list referencing the object group. */
Router(config)# ipv6 access-list network-object-acl permit ipv6 net-group netobj1 any

/* Apply the access list containing the object group to the desired interface and commit
 your configuration. */
Router(config)# interface TenGigE0/0/0/10/3
Router(config-if)# ipv6 address 2001:DB8::1/32
Router(config-if)# no shut
Router(config-if)# ipv6 access-group network-object-acl ingress compress level 3
Router(config-if)# commit
Tue Mar 28 10:23:34.106 IST

RP/0/0/CPU0:Mar 28 10:37:48.570 : ifmgr[397]: %PKT_INFRA-LINK-3-UPDOWN : Interface
TenGigE0/0/0/10/3 , changed state to Down
RP/0/0/CPU0:Mar 28 10:37:48.608 : ifmgr[397]: %PKT_INFRA-LINK-3-UPDOWN : Interface
TenGigE0/0/0/10/3 , changed state to Up

Router(config-if)# exit
```

実行コンフィギュレーション

設定を確認します。

```
Router(config)# show run
Tue Mar 28 10:37:55.737 IST

Building configuration...
!! IOS XR Configuration 0.0.0
...
```

```
!
object-group network ipv4 netobj1
 10.2.1.0/24
 host 10.1.1.1
 range  10.3.1.10 10.3.1.50
 description my-network-object
!
!
ipv4 access-list network-object-acl
 10 permit ipv4 net-group netobj1 any
!
interface Te0/0/0/0/3
 ipv4 address 1.1.1.1 255.255.255.0
 ipv4 access-group network-object-acl ingress compress level 3
!
```

ネットワーク オブジェクトグループ ACL は正常に設定されました。

ポート オブジェクトグループ ACL の設定

ポート オブジェクト グループには単一または複数のポート オブジェクトを含めることができます。

設定

次の一連の設定ステートメントを使用して、ポートオブジェクトグループACLを設定します。

```
/* From the global configuration mode, create a port object group, and commit your
configuration. */
RP/0/RP0/CPU0:router(config)# object-group port portobj1
RP/0/RP0/CPU0:router(config-object-group-ipv4)# description my-port-object
RP/0/RP0/CPU0:router(config-object-group-ipv4)# eq bgp
RP/0/RP0/CPU0:router(config-object-group-ipv4)# range 100 200
RP/0/RP0/CPU0:router(config-object-group-ipv4)# commit
RP/0/RP0/CPU0:router(config-object-group-ipv4)# exit

/* Create an access list referencing the object group. */
RP/0/RP0/CPU0:router(config)# ipv4 access-list port-object-acl permit ipv4 net-group
portobj1

/* Apply the access list containing the object group to the desired interface and commit
 your configuration. */
RP/0/RP0/CPU0:router(config)# interface Te0/0/0/3
RP/0/RP0/CPU0:router(config-if)# ipv4 address 2.2.2.2/24
RP/0/RP0/CPU0:router(config-if)# ipv4 access-group port-obj-acl ingress compress level
3
RP/0/RP0/CPU0:router(config-if)# no shut
RP/0/RP0/CPU0:router(config-if)# commit
Tue Mar 28 10:23:34.106 IST

RP/0/0/CPU0:Mar 28 10:37:48.570 : ifmgr[397]: %PKT_INFRA-LINK-3-UPDOWN : Interface
TenGigE0/0/0/10/3 , changed state to Down
RP/0/0/CPU0:Mar 28 10:37:48.608 : ifmgr[397]: %PKT_INFRA-LINK-3-UPDOWN : Interface
TenGigE0/0/0/10/3 , changed state to Up

RP/0/RP0/CPU0:router(config-if)# exit
```

実行コンフィギュレーション

設定を確認します。

```
RP/0/RP0/CPU0:router(config)# show run
Tue Mar 28 10:37:55.737 IST

Building configuration...
!! IOS XR Configuration 0.0.0
...
object-group port portobj1
 eq bgp
 range 100 200
!

ipv4 access-list port-object-acl
 10 permit tcp net-group portobj1
!
interface Te/0/0/0/3
 ipv4 access-group port-obj-acl ingress compress level 3
!
end
!
```

ポート オブジェクトグループ ACL が正常に設定されました。

オブジェクトグループ ACL 圧縮の確認

動作中の設定済みオブジェクトグループ ACL と ACL 内の AEC の圧縮を確認するには、この項で説明するコマンドを使用します。

（注） この項に示す出力はこの項のみのサンプルであり、前の項で示した設定に関連するものではありません。

確認

オブジェクトグループ ACL の圧縮を確認するには、次の一連の verification コマンドを使用します。

```
/* Verify the entries of the ACL in operation. */

RP/0/RP0/CPU0:router# show access-lists ipv4 network-object-acl hardware ingress location
 0/0/CPU0
ipv4 access-list network-object-acl
40 permit ospf net-group n_192.168.0.0_16 any (20898463272 matches)
70 permit tcp any net-group CORP_ALL_V4 established
100 permit udp net-group INTERNAL port-group KERBEROS_UDP net-group CORP_ALL_V4
130 permit udp net-group INTERNAL port-group DNS_UDP net-group CORP_ALL_V4
160 permit udp net-group INTERNAL port-group NTP net-group CORP_ALL_V4
190 permit udp net-group INTERNAL port-group LDAP_UDP net-group CORP_ALL_V4
...
1500 permit udp net-group VLAN60_SECURITY net-group h_192.168.77.242 port-group
UDP_50000-50100
1530 deny ipv4 net-group VLAN60_SECURITY any log (20891956640 matches)
```

```
...

/* Verify the ACE compression in the ACL. */
RP/0/RP0/CPU0:router# show access-lists ipv4 network-object-acl hardware ingress verify
 location 0/0/CPU0
Verifying TCAM entries for network-object-acl
Please wait...


    INTF    NPU lookup  ACL # intf Total  compression Total   result failed(Entry) TCAM
 entries
                type    ID  shared ACES   prefix-type Entries        ACE SEQ #
verified
---------- --- ------- --- ------ ------ ----------- ------- ------ -------------
------------

TenGigE0_0_0_10_3 (ifhandle: 0x1c8)

             1 IPV4      2      1    247 COMPRESSED      810 passed
     810
                                         SRC IP         2746 passed
    2746
                                         DEST IP        3413 passed
    3413
                                         SRC PORT        340 passed
     340
```

ACL 内の ACE の圧縮を正常に確認しました。

IPv4 ACL での TTL の照合および書き換えの設定

IPv4 ヘッダーに指定されている TTL 値で照合するように ACL を設定できます。TTL 一致条件を単一の値か、または複数の値に基づくように指定できます。また、**set ttl** コマンドを使用して、IPv4 ヘッダー内の TTL 値を書き換えることもできます。

IPv4 ACL での TTL の照合および書き換えの使用に関する制限事項

IPv4 ACL での TTL の照合および書き換えの使用には、次の制限事項があります。

- TTL の照合は、入力 ACL でのみサポートされています。
- ユーザ定義の TCAM キー（UDK）に **enable set ttl** オプションを設定すると、入力 ACL で ACL ロギングがサポートされません。
- TTL の書き換えを IP-in-IP ヘッダーの外側の IPv4 ヘッダーに適用する場合は、GRE カプセル化解除により外側の IPv4 ヘッダーのカプセル化が解除されると、内部 IPv4 ヘッダーにも TTL の書き換えが適用されます。
- TTL の照合はデフォルトの TCAM キーではサポートされていないため、「設定」の項で説明されているように、**hw-module profile tcam format** コマンドを使用するユーザ定義の TCAM キー（UDK）が必要です。

設定

IPv4 ACL に TTL の照合および書き換えを設定するには、次のステップを実行します。

```
/* Enable TTL matching and rewriting in the global configuration mode by using the
hw-module command */
Router(config)# hw-module profile tcam format access-list ipv4 dst-addr dst-port proto
port-range enable-set-ttl ttl-match

/* Configure an IPv4 ACL with the TTL parameters */
Router(config)# ipv4 access-list acl-v4
Router(config-ipv4-acl)# 10 deny tcp any any ttl eq 100
Router(config-ipv4-acl)# 20 permit tcp any any ttl range 1 50 set ttl 200
Router(config-ipv4-acl)# 30 permit tcp any any ttl neq 100 set ttl 255
Router(config-ipv4-acl)# commit
Thu Nov  2 12:22:58.948 IST

/* Attach the IPv4 ACL to the GigE interface */
Router(config)# interface Te0/0/0/0
Router(config-if)# ipv4 address 15.1.1.1 255.255.255.0
Router(config-if)# ipv4 access-group acl-v4 ingress
Router(config-if)# commit
```

実行コンフィギュレーション

show run コマンドを使用して設定を検証します。

```
Router(config)# show run
Thu Nov  2 14:01:53.376 IST
Building configuration...
!! IOS XR Configuration 0.0.0
!! Last configuration change at Thu Nov  2 12:22:59 2017 by annseque
!
hw-module profile tcam format access-list ipv4 dst-addr dst-port proto port-range
enable-set-ttl ttl-match
!
ipv4 access-list acl-v4
 10 deny tcp any any ttl eq 100
 20 permit tcp any any ttl range 1 50 set ttl 200
 30 permit tcp any any ttl neq 100 set ttl 255
!
interface Te0/0/0/0
 ipv4 address 15.1.1.1 255.255.255.0
 ipv4 access-group acl-v4 ingress
!
```

IPv4 ACL に TTL の照合および書き換えが正常に設定されています。

インターフェイス ベースの一意の IPv4 ACL の設定

インターフェイス間で共有され、同じ TCAM スペースを使用する ACL は共有 ACL と呼ばれています。ただし、設定できる一意の共有 ACL は 31 個のみです。それ以上の ACL を設定するには、**interface-based** コマンドを使用して ACL 共有を無効にする必要があります。ACL をインターフェイスに一意にすることによって、31 個を超える ACL を作成できます。

設定

一意のインターフェイスベースの IPv4 ACL を作成するには、次の設定を使用します。

（注）
- **hw-module profile** コマンドを入力した後は、コマンドをアクティブにするためにラインカードを再起動する必要があります。
- TCAM はインターフェイス上の各 ACL に割り当てられ、ACL 間で共有されることはありません。したがって、たとえば 10 の TCAM エントリを使用する ACL が 100 のインターフェイスに適用されている場合、TCAM エントリの合計数は 1000 になります。

```
/* Enable interface-based, unique IPv4 ACLs */
Router(config)# hw-module profile tcam format access-list ipv4 src-addr src-port dst-addr
 dst-port interface-based

/* Configure an IPv4 ACL with the TTL parameters */
Router(config)# ipv4 access-list acl-v4
Router(config-ipv4-acl)# 10 deny tcp any any ttl eq 100
Router(config-ipv4-acl)# 20 permit tcp any any ttl range 1 50 set ttl 200
Router(config-ipv4-acl)# 30 permit tcp any any ttl neq 100 set ttl 255
Router(config-ipv4-acl)# commit
Thu Nov  2 12:22:58.948 IST

/* Attach the IPv4 ACL to the GigE interface */
Router(config)# interface Te0/0/0/0
Router(config-if)# ipv4 address 15.1.1.1 255.255.255.0
Router(config-if)# ipv4 access-group acl-v4 ingress
Router(config-if)# commit
```

実行コンフィギュレーション

show run コマンドを使用して設定を検証します。

```
Router(config)# show run
Thu Nov  2 14:01:53.376 IST
Building configuration...
!! IOS XR Configuration 0.0.0
!! Last configuration change at Thu Nov  2 12:22:59 2017 by annseque
!
hw-module profile tcam format access-list ipv4 src-addr src-port dst-addr dst-port
interface-based
!
ipv4 access-list acl-v4
 10 deny tcp any any ttl eq 100
 20 permit tcp any any ttl range 1 50 set ttl 200
 30 permit tcp any any ttl neq 100 set ttl 255
!
interface Te0/0/0/0
 ipv4 address 15.1.1.1 255.255.255.0
 ipv4 access-group acl-v4 ingress
!
```

一意のインターフェイスベースの IPv4 ACL が正常に設定されています。

IPv6 ACL での TTL の照合および書き換えの設定

IPv6 ヘッダーに指定されている TTL 値で照合するように ACL を設定できます。TTL 一致条件を単一の値か、または複数の値に基づくように指定できます。また、**set ttl** コマンドを使用して、IPv6 ヘッダー内の TTL 値を書き換えることもできます。

（注） **hw-module profile** コマンドを入力した後は、コマンドをアクティブにするためにラインカードを再起動する必要があります。

IPv6 ACL での TTL の照合および書き換えの使用に関する制限

IPv6 ACL での TTL の照合および書き換えの使用には、次の制限事項があります。

- TTL の照合は、入力 ACL でのみサポートされています。
- ユーザ定義の TCAM キー（UDK）に **enable set ttl** オプションを設定すると、入力 ACL で ACL ロギングがサポートされません。
- TTL の書き換えを IP-in-IP ヘッダーの外側の IPv6 ヘッダーに適用する場合は、GRE カプセル化解除により外側の IPv6 ヘッダーのカプセル化が解除されると、内部 IPv6 ヘッダーにも TTL の書き換えが適用されます。
- TTL の照合はデフォルトの TCAM キーではサポートされていないため、「設定」の項で説明されているように、**hw-module profile tcam format** コマンドを使用するユーザ定義の TCAM キー（UDK）が必要です。

設定

IPv6 ACL に TTL の照合および書き換えを設定するには、次のステップを実行します。

```
/* Enable TTL matching and rewriting in the global configuration mode by using the
hw-module command */
Router(config)# hw-module profile tcam format access-list ipv6 dst-addr dst-port src-port
 next-hdr enable-set-ttl ttl-match

/* Configure an IPv6 ACL with the TTL parameters */
Router(config)# ipv6 access-list acl-v6
Router(config-ipv6-acl)# 10 deny tcp any any ttl eq 50
Router(config-ipv6-acl)# 20 permit tcp any any ttl lt 50 set ttl 255
Router(config-ipv6-acl)# 30 permit tcp any any ttl gt 50 set ttl 200
Router(config-ipv6-acl)# commit
Thu Nov  2 12:22:58.948 IST

/* Attach the IPv6 ACL to the GigE interface */
Router(config)# interface Te0/0/0/0
Router(config-if)# ipv6 address 2001:2:1::1/64
Router(config-if)# ipv6 access-group acl-v6 ingress
Router(config-if)# commit
```

実行コンフィギュレーション

show run コマンドを使用して設定を検証します。

```
Router(config)# show run
Thu Nov  2 14:01:53.376 IST
Building configuration...
!! IOS XR Configuration 0.0.0
!! Last configuration change at Thu Nov  2 12:22:59 2017 by annseque
!hw-module profile tcam format access-list ipv6 dst-addr dst-port src-port next-hdr
enable-set-ttl ttl-match
!
ipv6 access-list acl-v6
 10 deny tcp any any ttl eq 50
20 permit tcp any any ttl lt 50 set ttl 255
30 permit tcp any any ttl gt 50 set ttl 200
!
interface Te0/0/0/0
 ipv6 address 2001:2:1::1/64
 ipv6 access-group acl-v6 ingress
!
```

IPv6 ACL に TTL の照合および書き換えが正常に設定されています。

インターフェイス ベースの一意の IPv6 ACL の設定

インターフェイス間で共有され、同じ TCAM スペースを使用する ACL は共有 ACL と呼ばれています。ただし、設定できる一意の共有 ACL は 31 個のみです。それ以上の ACL を設定するには、**interface-based** コマンドを使用して ACL 共有を無効にする必要があります。ACL をインターフェイスに一意にすることによって、31 個を超える ACL を作成できます。

設定

一意のインターフェイスベースの IPv6 ACL を作成するには、次の設定を使用します。

（注）

- hw-module profile コマンドを入力した後は、コマンドをアクティブにするためにラインカードを再起動する必要があります。
- TCAM はインターフェイス上の各 ACL に割り当てられ、ACL 間で共有されることはありません。したがって、たとえば 10 の TCAM エントリを使用する ACL が 100 のインターフェイスに適用されている場合、TCAM エントリの合計数は 1000 になります。

```
/* Enable interface-based, unique IPv6 ACLs */
Router(config)# hw-module profile tcam format access-list ipv6 src-addr src-port dst-addr
 dst-port next-hdr interface-based

/* Configure an IPv6 ACL with the TTL parameters */
Router(config)# ipv6 access-list acl-v6
Router(config-ipv6-acl)# 10 deny tcp any any ttl eq 100
Router(config-ipv6-acl)# 20 permit tcp any any ttl range 1 50 set ttl 200
Router(config-ipv6-acl)# 30 permit tcp any any ttl neq 100 set ttl 255
Router(config-ipv6-acl)# commit
Thu Nov  2 12:22:58.948 IST
```

```
/* Attach the IPv6 ACL to the GigE interface */
Router(config)# interface Te0/0/0/0
Router(config-if)# ipv6 address 2001:2:1::1/64
Router(config-if)# ipv6 access-group acl-v6 ingress
Router(config-if)# commit
```

実行コンフィギュレーション

show run コマンドを使用して設定を検証します。

```
Router(config)# show run
Thu Nov  2 14:01:53.376 IST
Building configuration...
!! IOS XR Configuration 0.0.0
!! Last configuration change at Thu Nov  2 12:22:59 2017 by annseque
!
hw-module profile tcam format access-list ipv6 src-addr src-port dst-addr dst-port
next-hdr interface-based
!
ipv6 access-list acl-v6
 10 deny tcp any any ttl eq 100
 20 permit tcp any any ttl range 1 50 set ttl 200
 30 permit tcp any any ttl neq 100 set ttl 255
!
interface Te0/0/0/0
 ipv6 address 2001:2:1::1/64
 ipv6 access-group acl-v6 ingress
!
```

一意のインターフェイスベースの IPv6 ACL が正常に設定されています。

IP アクセス リスト ロギング メッセージの概要

Cisco IOS XR ソフトウェアでは、標準 IP アクセス リストで許可または拒否されたパケットに関するログ メッセージが表示されます。つまり、パケットがアクセス リストに一致すると、そのパケットに関するログメッセージ情報がコンソールに送信されます。ログをコンソールに送信するメッセージのレベルは、グローバル コンフィギュレーション モードの **logging console** コマンドで制御します。

最初にパケットがアクセスリストをトリガーすると、すぐにログメッセージが生成されます。その後、5分間隔でパケットが収集されて表示または記録されます。ログメッセージにはアクセス リスト番号、パケットの許可または拒否に関する状況、パケットの送信元 IP アドレス、および直前の 5 分間に許可または拒否された送信元からのパケット数が示されます。

ただし、{ **ipv4 | ipv6** } **access-list log-update threshold** コマンドを使用すれば、アクセス リストに一致する（許可または拒否される）場合にシステムでログ メッセージを生成するパケットの数を設定できます。この手順は、5分間隔よりも短い頻度でログ メッセージを受信する場合に実行することを推奨します。

注意 *number-of-matches* 引数を 1 に設定すると、ログ メッセージはキャッシュされずにただちに送信されます。この場合、アクセス リストに一致するすべてのパケットについてログ メッセージが生成されます。大量のログ メッセージでシステムが過負荷になる可能性があるため、1 に設定することは推奨されません。

{ ipv4 | ipv6} access-list log-update threshold コマンドを使用する場合も 5 分タイマーは有効なままなので、各キャッシュのメッセージ数に関係なく、5 分が経過すると各キャッシュは空になります。ログ メッセージを送信するタイミングに関係なく、しきい値が指定されていない場合と同様に、ログ メッセージのキャッシュは消去され、カウントは 0 にリセットされます。

（注） ログ メッセージが多すぎて処理できない場合や、1 秒以内に 2 つ以上のログ メッセージを処理した場合には、ログ メッセージ パケットの一部がドロップされることがあります。この動作により、ログを生成するパケットの数が多くなっても、ルータが CPU サイクルを過度に使用することはありません。したがって、ロギング機能は課金ツールや、アクセス リストとの一致数を正確に把握するための情報源として使用しないでください。

プレフィックス リストの概要

プレフィックス リストはルート マップおよびルート フィルタリング操作に使用されるほか、ボーダー ゲートウェイ プロトコル（BGP）の多くのルート フィルタリング コマンドではアクセス リストの代わりに使用できます。プレフィックスは IP アドレスの一部であり、左端のオクテットの左端のビットから始まります。アドレスの何ビットがプレフィックスに属するかを正確に指定すると、プレフィックスを使用してアドレスを集約し、そのアドレスに対して再配布（フィルタ ルーティング アップデート）などの機能を実行できるようになります。

プレフィックス リストを使用した BGP フィルタリング

プレフィックス リストは、BGP ルート フィルタリング コマンドの多くでアクセス リストの代わりに使用できます。これは BGP プロトコルのグローバル コンフィギュレーションで設定されます。プレフィックス リストを使用した場合の利点は次のとおりです。

- サイズの大きなリストをロードしてルート ルックアップを実施する場合のパフォーマンスが大幅に向上します。
- 差分更新がサポートされます。
- CLI の使い勝手が向上します。アクセス リストを使用して BGP 更新をフィルタリングするための CLI は、パケット フィルタリング形式を使用しているため、わかりにくく使い勝手もよくありません。
- 柔軟性が高まります。

コマンドでプレフィックス リストを使用するには、あらかじめプレフィックス リストをセットアップしておく必要があります。プレフィックス リストのエントリには、シーケンス番号を割り当ててください。

プレフィックス リストでトラフィックをフィルタリングする仕組み

プレフィックス リストによるフィルタリングでは、ルートのプレフィックスが、プレフィックス リストに記載されているプレフィックスと照合されます。一致すると、一致したルートが使用されます。具体的には、プレフィックスを許可するか、拒否するかは次のルールに基づきます。

- 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、暗黙の deny が適用されます。
- プレフィックス リストの複数のエントリが特定のプレフィックスと一致したときは、最も長く、最も具体的な一致が選択されます。

シーケンス番号は自動的に生成されます。ただし、この自動生成をディセーブルにしている場合を除きます。シーケンス番号の自動生成をディセーブルにしている場合は、IPv4 のプレフィックス リスト コンフィギュレーション コマンドで、**permit** と **deny** コマンドの *sequence-number* 引数を使用して、各エントリのシーケンス番号を指定する必要があります。プレフィックス リストのエントリを削除するには、permit または deny コマンドの **no** 形式を使用して、*sequence-number* 引数を指定します。

show コマンドの出力には、シーケンス番号が含まれます。

プレフィックス リストの設定

設定例

「Deny all routes with a prefix of 10/8」というコメントが付いたプレフィックス リスト「pfx_2」を作成します。このプレフィックス リストは、128.0.0.0/8 の /24 に一致するプレフィックスをすべて拒否します。

```
Router#configure
Router(config)#ipv4 prefix-list pfx_2


Router(config-ipv4_pfx)#10 remark Deny all routes with a prefix of 10/8
Router(config-ipv4_pfx)#20 deny 128.0.0.0/8 eq 24
/* Repeat the above step as necessary. Use the no sequence-number command to delete an
entry. */

Router(config-ipv4_pfx)#commit
```

実行コンフィギュレーション

```
Router#show running-config ipv4 prefix-list pfx_2
ipv4 prefix-list pfx_2
```

```
 10 remark Deny all routes with a prefix of 10/8
 20 deny 128.0.0.0/8 eq 24
!
```

確認

許可とコメントの設定が設定されているコンフィギュレーションに合致していることを確認します。

```
Router# show prefix-list pfx_2
ipv4 prefix-list pfx_2
 10 remark Deny all routes with a prefix of 10/8
 20 deny 128.0.0.0/8 eq 24
RP/0/RP0/CPU0:ios#
```

関連コマンド

- ipv4 prefix-list
- ipv6 prefix-list
- show prefix-list ipv4
- show prefix-list ipv6

プレフィックスリストエントリの順序付けとプレフィックス リストの変更

設定例

名前付きプレフィックス リストのエントリにシーケンス番号を割り当て、プレフィックス リストに対してエントリを追加または削除する方法を示します。プレフィックス リストを変更することを前提に説明します。プレフィックス リストの並べ替えは任意です。

```
Router#config
Router(config)#ipv4 prefix-list cl_1


Router(config)#10 permit 172.16.0.0 0.0.255.255
/* Repeat the above step as necessary adding statements by sequence number where you
planned; use the no sequence-number command to delete an entry */

Router(config)#commit
end
Router#resequence prefix-list ipv4 cl_1 20 15
```

実行コンフィギュレーション

```
/*Before resequencing/*
Router#show running-config ipv4 prefix-list cl_1
ipv4 prefix-list cl_1
 10 permit 172.16.0.0/16
```

```
!
/* After resequencing using the resequence prefix-list ipv4 cl_1 20 15 command: */
Router#show running-config ipv4 prefix-list cl_1
ipv4 prefix-list cl_1
 35 permit 172.16.0.0/16
!
```

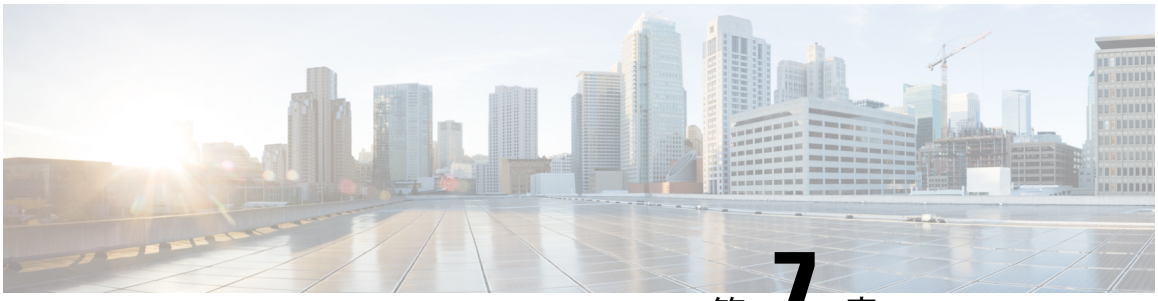
確認

プレフィックス リストが並べ替えられたことを確認します。

```
Router#show prefix-list cl_1
ipv4 prefix-list cl_1
 35 permit 172.16.0.0/16
```

関連コマンド

- resequence prefix-list ipv4
- resequence prefix-list ipv6
- ipv4 prefix-list
- ipv6 prefix-list
- show prefix-lists ipv4
- show prefix-lists ipv6



第 7 章

シスコ エクスプレス フォワーディングの実装

シスコ エクスプレス フォワーディングの実装

Cisco Express Forwarding（CEF）は、拡張レイヤ 3 IP スイッチング テクノロジーです。CEF によって、インターネットや、Webベースのアプリケーションまたは対話型セッションが集中的に使用されるネットワークなどの、大規模でダイナミックなトラフィック パターンを持つネットワークのパフォーマンスおよびスケーラビリティが最適化されます。CEF は内蔵されている機能であり、イネーブルにするために設定を行う必要はありません。必要に応じて、デフォルトのルート パージ遅延とスタティック ルートを変更できます。

コンポーネント

Cisco IOS XR ソフトウェアの CEF は常に、次の 2 種類のコンポーネントとともに CEF モードで動作します。

- 転送情報ベース（FIB）データベース：プロトコル依存の FIB プロセスは、ルート プロセッサに IPv4 および IPv6 ユニキャストの転送テーブルを保持します。各ノード上の FIB はルーティング情報ベース（RIB）を更新し、ルート解決を実行してルートプロセッサに個別に FIB テーブルを保持します。各ノード上の FIB テーブルに格納されている情報は、テーブルによって若干異なることがあります。
- 隣接関係テーブル：プロトコルに依存しない隣接情報ベース（AIB）

CEF は、Cisco IOS XR ソフトウェアのプライマリ IP パケット転送データベースです。CEF の役割は次の機能を果たすことです。

- ソフトウェア スイッチング パス
- ソフトウェアおよびハードウェア転送エンジンの転送テーブルおよび隣接関係テーブルのメンテナンス（AIB によるメンテナンス）

Cisco IOS XR ソフトウェア上の CEF では、次の機能をサポートしています。

- バンドル インターフェイスのサポート
- マルチパス サポート
- ルート整合性
- パッケージング、再起動性、リソース不足（OOR）処理などのハイ アベイラビリティ機能
- OSPFv2 SPF プレフィックス優先順位付け
- BGP 属性ダウンロード

CEF の利点

- パフォーマンス向上：CEF は、高速スイッチング ルート キャッシングよりも CPU を消費しません。より多くの CPU 処理能力を Quality of Service（QoS）や暗号化などのレイヤ 3 サービスに向けることができます。
- スケーラビリティ：CEF では、各ライン カードでスイッチング機能を最大限に活用できます。
- 復元力：CEF では、大規模な動的ネットワーク上で比類ないレベルのスイッチング一貫性と安定性を実現します。動的ネットワークでは、ルーティング変更のために、高速にスイッチングされるキャッシュ エントリが頻繁に無効化されます。ルーティング変更により、ルート キャッシュを使用した高速スイッチングではなく、ルーティング テーブルを使用したトラフィックのプロセス スイッチングが行われることがあります。転送情報ベース（FIB）ルックアップテーブルにはルーティングテーブルに存在する既知のルートがすべて含まれているため、ルートキャッシュのメンテナンスが不要になるほか、高速スイッチングまたはプロセス スイッチング フォワーディングのシナリオも必要ありません。CEF では、一般的なデマンド キャッシング スキームよりも効率よくトラフィックを切り替えることができます。

Cisco IOS XR ソフトウェアでは、次の CEF 転送テーブルがメンテナンスされます。

- IPv4 CEF データベース：IPv4 ユニキャスト パケット転送用の IPv4 ユニキャスト ルートが保存されます。
- IPv6 CEF データベース：IPv6 ユニキャスト パケット転送用の IPv6 ユニキャスト ルートが保存されます。
- MPLS LFD データベース：MPLS パケット転送用の MPLS ラベル テーブルが保存されます。

CEF の確認

IPv4 または IPv6 CEF テーブルの詳細を表示するには、次のコマンドを使用します。

- show cef {*ipv4 address* | *ipv6 address*} hardware egress

IPv4 または IPv6 CEF テーブルを表示します。ネクスト ホップおよび転送インターフェイスがプレフィックスごとに表示されます。**show cef** コマンドの出力は、場所によって異なります。

```
Router# show cef 203.0.1.2 hardware egress
  203.0.1.2/32, version 0, internal 0x1020001 0x0 (ptr 0x8d7db7f0) [1], 0x0 (0x8daeedf0),
 0x0 (0x0)
 Updated Nov 20 13:33:23.557
 local adjacency 203.0.1.2
 Prefix Len 32, traffic index 0, Adjacency-prefix, precedence n/a, priority 15
   via 203.0.1.2/32, HundredGigE0/9/0/0, 3 dependencies, weight 0, class 0 [flags 0x0]
    path-idx 0 NHID 0x0 [0x8cfc81a0 0x0]
    next hop 203.0.1.2/32
    local adjacency
```

- show cef {ipv4 | ipv6} summary

IPv4 または IPv6 CEF テーブルのサマリーを表示します。

```
Router#show cef ipv4 summary
Fri Nov 20 13:50:45.239 UTC

Router ID is 216.1.1.1

IP CEF with switching (Table Version 0) for node0_RP0_CPU0

  Load balancing: L4
  Tableid 0xe0000000 (0x8cf5b368), Vrfid 0x60000000, Vrid 0x20000000, Flags 0x1019
  Vrfname default, Refcount 4129
  56 routes, 0 protected, 0 reresolve, 0 unresolved (0 old, 0 new), 7616 bytes
  13 rib, 0 lsd, 0:27 aib, 1 internal, 10 interface, 4 special, 1 default routes
  56 load sharing elements, 24304 bytes, 1 references
  1 shared load sharing elements, 432 bytes
  55 exclusive load sharing elements, 23872 bytes
  0 route delete cache elements
  13 local route bufs received, 1 remote route bufs received,  0 mix bufs received
  13 local routes, 0 remote routes
  13 total local route updates processed
  0 total remote route updates processed
  0 pkts pre-routed to cust card
  0 pkts pre-routed to rp card
  0 pkts received from core card
  0 CEF route update drops, 0 revisions of existing leaves
  0 CEF route update drops due to version mis-match
  Resolution Timer: 15s
  0 prefixes modified in place
  0 deleted stale prefixes
  0 prefixes with label imposition, 0 prefixes with label information
  0 LISP EID prefixes, 0 merged, via 0 rlocs
 28 next hops
  1 incomplete next hop

 0 PD backwalks on LDIs with backup path
```

- show cef { ipv4 address | ipv6 address } detail

IPv4 または IPv6 CEF テーブルの詳細を表示します。

```
Router#show cef 203.0.1.2 detail
203.0.1.2/32, version 0, internal 0x1020001 0x0 (ptr 0x8d7db7f0) [1], 0x0 (0x8daeedf0),
 0x0 (0x0)
 Updated Nov 20 13:33:23.556
 local adjacency 203.0.1.2
```

```
 Prefix Len 32, traffic index 0, Adjacency-prefix, precedence n/a, priority 15
  gateway array (0x8d84beb0) reference count 1, flags 0x0, source aib (10), 0 backups
                [2 type 3 flags 0x8401 (0x8d99a598) ext 0x0 (0x0)]
  LW-LDI[type=3, refc=1, ptr=0x8daeedf0, sh-ldi=0x8d99a598]
  gateway array update type-time 1 Nov 20 13:33:23.556
 LDI Update time Nov 20 13:33:23.556
 LW-LDI-TS Nov 20 13:33:23.556
   via 203.0.1.2/32, HundredGigE0/9/0/0, 3 dependencies, weight 0, class 0 [flags 0x0]
    path-idx 0 NHID 0x0 [0x8cfc81a0 0x0]
    next hop 203.0.1.2/32
    local adjacency
        Load distribution: 0 (refcount 2)

    Hash  OK  Interface                 Address
    0     Y   HundredGigE0/9/0/0        203.0.1.2
```

- show adjacency detail

インターフェイスごとのレイヤ 2 情報など詳細な隣接情報を表示します。show adjacency コマンドの出力は、場所によって異なります。

```
Router#show adjacency detail

------------------------------------------------------------------------------
0/RP0/CPU0
------------------------------------------------------------------------------
Interface                  Address                 Version  Refcount Protocol
Hu0/9/0/0                 (interface)                 13        1(    0)
                          (interface entry)
                          mtu: 1500, flags 1 4


Hu0/9/0/1                 (interface)                 31        1(    0)
                          (interface entry)
                          mtu: 1500, flags 1 4
```

フロー単位ロード バランシング

システムは基本的に 7 タプル ハッシュ アルゴリズムをサポートしています。ロード バランシングでは、レイヤ 3（ネットワーク層）およびレイヤ 4（トランスポート層）のルーティング情報に基づいてパケットを複数のリンクに分散させるルータ機能について説明します。ルータが宛先に至るパスを複数検出した場合は、その宛先の複数のエントリでルーティング テーブルが更新されます。

フロー単位のロード バランシングでは、以下の機能が実行されます。

- 着信データ トラフィックは、複数の等コスト接続に均等に分散されます。
- 着信データ トラフィックは、バンドル インターフェイス内の複数の等コスト接続メンバ リンクに均等に分散されます。
- レイヤ 2 バンドルとレイヤ 3（ネットワーク レイヤ）のロード バランシングの決定は、IPv4、IPv6、および MPLS フローで行われます。IPv4 または IPv6 のペイロードの場合は、7 タプル ハッシングが実行されます。3 つ以下のラベルが設定された MPLS ペイロードの場合は、ハードウェアが下部のペイロードを解析し、そのペイロードに IPv4 または IPv6 ヘッダーがあるかどうかを特定します。IPv4 または IPv6 ヘッダーの場合は、IP 送信元、

IP 送信先、ルータ ID、およびラベル スタックの基づいて 4 タプル ハッシングが実行されます。それ以外の場合は、MPLS ラベル ベースのハッシングが実行されます。MPLS ラベル ベースのハッシングの場合は、上位 4 ラベルがハッシュの計算に使用されます。.

- 7 タプル ハッシュ アルゴリズムはより細かなロード バランシングを実現し、複数の等コスト レイヤ3（ネットワーク層）パス全体でのロードバランシングに使用されます。レイヤ3（ネットワーク層）パスは、物理インターフェイスまたはバンドルインターフェイス上にあります。また、メンバ リンクに対するロード バランシングが、レイヤ 2 バンドル インターフェイス内で行われることがあります。
- 7 タプル ロードバランス ハッシュ計算には以下が含まれます。
  - 送信元 IP アドレス
  - 宛先 IP アドレス
  - IP プロトコル タイプ
  - ルータ ID
  - 送信元ポート
  - 宛先ポート
  - 入力インターフェイス

宛先別ロード バランシング

再帰 MPLS パス（BGP 3107 を通じて学習したパスなど）を介して遷移するパケットには、宛先別ロード バランシングが使用されます。宛先別ロード バランシングとは、ルートの宛先に基づいてルータがパケットを分散します。同じネットワークへのパスが 2 つある場合、そのネットワーク上の宛先 1 へのすべてのパケットが最初のパスを介して移動し、そのネットワーク上の宛先 2 へのすべてのパケットが 2 番目のパスを介して移動するという具合になります。これにより、パケットの順序は保持されますが、リンクの使用が不等になる可能性があります。1 つのホストがトラフィックの大部分を受け取る場合は、すべてのパケットが 1 つのリンクを使用し、他のリンクの帯域幅は使用されないままとなります。宛先アドレスが多数ある場合は、より均等にリンクが使用されます。

スタティック ルートの設定

ルータは、ユーザが手動で設定したルート テーブル エントリのルート情報を使用するか、またはダイナミック ルーティング アルゴリズムで計算されたルート情報を使用して、パケットを転送します。スタティック ルートは、2つのルータ間の明示パスを定義するものであり、自動的にはアップデートされません。ネットワークに変更があった場合は、ユーザが手動でスタティック ルートを再設定する必要があります。スタティック ルートは、ダイナミック ルートに比べて使用する帯域幅が少なくなります。スタティック ルートは、ネットワーク トラフィックが予測可能で、ネットワーク設計がシンプルな環境で使用します。スタティック ルートはネットワークの変化に対応できないので、大規模でたえず変化しているネットワークでは、スタティック ルートを使用すべきではありません。大部分のネットワークは、ルータ間の通信に

ダイナミック ルートを使用しますが、特殊な状況でスタティック ルートを 1 つか 2 つ設定する場合があります。スタティック ルートは、最終手段としてのゲートウェイ（ルーティング不能なすべてのパケットの送信先となるデフォルト ルータ）を指定する場合にも便利です。

設定例

HundredGigE インターフェイスを介してルータ A とルータ B 間にスタティック ルートを作成します。宛先 IP アドレスは 203.0.1.2/32、ネクストホップ アドレスは 1.0.0.2 です。


```
Router(config)#router static address-family ipv4 unicast
Router(config-static-afi)#203.0.1.2/32 HundredGigE 0/9/0/0 1.0.0.2
Router(config-static-afi)#commit
```

実行コンフィギュレーション

```
Router#show running-config router static address-family ipv4 unicast
router static
 address-family ipv4 unicast
 203.0.1.2/32 HundredGigE 0/9/0/0 1.0.0.2
 !
!
```

確認

Next Hop Flags のフィールドに COMPLETE と表示され、設定が適切に機能していることを確認してください。

```
Router#show cef  203.0.1.2/32 hardware egress detail location 0/RP0/CPU0
203.0.1.2/32, version 0, internal 0x1020001 0x0 (ptr 0x8d7db7f0) [1], 0x0 (0x8daeedf0),
 0x0 (0x0)
 Updated Nov 20 13:33:23.557
 local adjacency 203.0.1.2
 Prefix Len 32, traffic index 0, Adjacency-prefix, precedence n/a, priority 15
   via 203.0.1.2/32, HundredGigE0/9/0/0, 3 dependencies, weight 0, class 0 [flags 0x0]
    path-idx 0 NHID 0x0 [0x8cfc81a0 0x0]
    next hop 1.0.0.2
    local adjacency
```

関連コマンド

- router static
- show cef

BGP 属性ダウンロード

BGP 属性ダウンロード機能を使用すると、CEF にインストールした BGP 属性を表示できます。

- **show cef bgp-attribute** コマンドは、CEF にインストールした BGP 属性を表示します。

- 属性 ID とローカル属性 ID で特定の BGP 属性を表示するには、**show cef bgp-attribute** *attribute-id* コマンドと **show cef bgp-attribute** *local-attribute-id* コマンドを使用します。

確認

```
Router# show cef bgp-attribute
Router ID is 216.1.1.1

IP CEF with switching (Table Version 0) for node0_RP0_CPU0

  Load balancing: L4
  Tableid 0xe0000000 (0x8cf5b368), Vrfid 0x60000000, Vrid 0x20000000, Flags 0x1019
  Vrfname default, Refcount 4129
  56 routes, 0 protected, 0 reresolve, 0 unresolved (0 old, 0 new), 7616 bytes
  13 rib, 0 lsd, 0:27 aib, 1 internal, 10 interface, 4 special, 1 default routes
  56 load sharing elements, 24304 bytes, 1 references
  1 shared load sharing elements, 432 bytes
  55 exclusive load sharing elements, 23872 bytes
  0 route delete cache elements
  13 local route bufs received, 1 remote route bufs received,  0 mix bufs received
  13 local routes, 0 remote routes
  13 total local route updates processed
  0 total remote route updates processed
  0 pkts pre-routed to cust card
  0 pkts pre-routed to rp card
  0 pkts received from core card
  0 CEF route update drops, 0 revisions of existing leaves
  0 CEF route update drops due to version mis-match
  Resolution Timer: 15s
  0 prefixes modified in place
  0 deleted stale prefixes
  0 prefixes with label imposition, 0 prefixes with label information
  0 LISP EID prefixes, 0 merged, via 0 rlocs
 28 next hops
  1 incomplete next hop

 0 PD backwalks on LDIs with backup path


VRF: default

_____________
Table ID: 0xe0000000. Total number of entries: 0
OOR state: GREEN. Number of OOR attributes: 0
```

関連コマンド

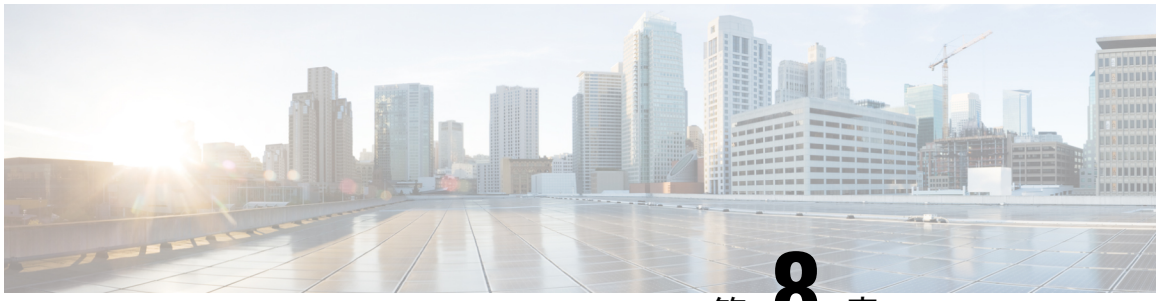
- show cef bgp-attribute

プロアクティブなアドレス解決プロトコルおよびネイバー探索

CEF は、レイヤ 2 隣接関係情報がないルートをインストールすると、不完全なレイヤ 3 ネクストホップを作成してハードウェアにプログラムします。この不完全なプログラミングが原因となり、最初のパケットはソフトウェア転送パスに転送されます。次に、ソフトウェア転送ではレイヤ 2 隣接関係情報を解決するために、パケットからレイヤ 2 ヘッダーを除去して、ARP（アドレス解決プロトコル）または ND（ネイバー探索）に転送します。このようなパケット

でレイヤ 2 ヘッダー内に機能固有の情報が含まれていると、ソフトウェア転送パスがレイヤ 2 ヘッダーを完全に除去できないため、ARP または ND は欠落しているレイヤ 2 隣接関係情報を解決できずに、トラフィックがドロップされる結果になります。

プロアクティブな ARP および ND 機能は、欠落しているレイヤ 2 隣接関係情報を解決するために CEF が ARP または ND をプロアクティブにトリガーするようにし、ネクストホップ情報が解決されるまで 15 秒ごとに再試行することで上記の問題を解決します。したがって、不完全なネクストホップ情報を含むスタティック ルートを設定すると、この機能によって ARP または ND の解決が自動的にトリガーされます。



第 8 章

LPTS の実装

LPTS の概要

Local Packet Transport Services（LPTS）では、セキュア ドメイン ルータ（SDR）宛てのすべてのパケット フローを記述するテーブルを保持し、これにより、意図した宛先に確実にパケットが配信されます。

LPTS では、ポート アービトレータおよびフロー マネージャという 2 つのコンポーネントを使用して、このタスクを実行します。ポート アービトレータおよびフローマネージャは、Internal Forwarding Information Base（IFIB）と呼ばれる、論理ルータ用のパケット フローを記述するテーブルを保持するプロセスです。IFIBは、受信したパケットを適切なルートプロセッサにルーティングして処理するために使用します。

LPTSは、ルータ外からパケットを受信するすべてのアプリケーションと内部的にインターフェイスします。LPTS は、カスタマー設定の必要なく機能します。ただし、ポリサー値は、必要に応じてカスタマイズできます。カスタマーがLPTSフローマネージャとポートアービトレータのアクティビティやパフォーマンスをモニタリングできるように、LPTSのshowコマンドが用意されています。

LPTS ポリサー

Cisco IOS XR では、ルート プロセッサ（RP）宛ての制御パケットは、着信ポートで一連の入力ポリサーを使用してポリシングされます。これらのポリサーは、ブートアップ時に LPTS コンポーネントによって静的にプログラミングされます。これらのポリサーは、着信制御トラフィックのフロー タイプに基づいて適用されます。フロー タイプは、パケット ヘッダーを調べることで決定されます。これらの静的入力ポリサーのポリサー レートは、コンフィギュレーション ファイルで定義され、ブートアップ時にルートプロセッサにプログラミングされます。

これらの一連の入力ポリサーのフロータイプに基づいて、ポリサー値を変更できます。各ノードのポリサーごとにレートを設定できます。

（注） デフォルトのポリサー値とフロー タイプの現在のレートは、次の show コマンドの出力で確認できます。

```
show lpts pifib hardware police
```

設定例

次の値をすべてのノードにグローバルに使用して、OSPF および BGP フロータイプに LPTS ポリサーを設定します。

- ospf unicast default rate 3000
- bgp default rate 4000

```
Router#configure
Router(config)#lpts pifib hardware police
Router(config-pifib-policer-global)#flow ospf unicast default rate 3000
Router(config-pifib-policer-global)#flow bgp default rate 4000
Router (config-pifib-policer-global)#commit
```

実行コンフィギュレーション

```
lpts pifib hardware police
flow ospf unicast default rate 3000
flow bgp default rate 4000
!
```

確認

```
Router#show run lpts pifib hardware police
lpts pifib hardware police
flow ospf unicast default rate 3000
flow bgp default rate 4000
```

設定例

次の値を個々のノード 0/RP0/CPU0 に使用して、OSPF および BGP フロー タイプに LPTS ポリサーを設定します。

- ospf unicast default rate 3000
- flow bgp default rate 4000

```
Router#configure
Router(config)#lpts pifib hardware police location 0/RP0/CPU0
Router(config-pifib-policer-per-node)#flow ospf unicast default rate 3000
Router(config-pifib-policer-per-node)#flow bgp default rate 4000
Router(config-pifib-policer-per-node)#commit
```

実行コンフィギュレーション

```
lpts pifib hardware police location 0/RP0/CPU0
flow ospf unicast default rate 3000
flow bgp default rate 4000
```

確認

show lpts pifib hardware police location 0/RP0/CPU0 コマンドは、指定したノードの Pre-Internal Forwarding Information Base（IFIB）情報を表示します。

```
Router#show lpts pifib hardware police location 0/RP0/CPU0
-------------------------------------------------------------
               Node 0/RP0/CPU0:
-------------------------------------------------------------
 Burst = 100ms for all flow types
-------------------------------------------------------------
FlowType               Policer Type    Cur. Rate Burst     npu
---------------------- ------- ------- --------- --------- ---------
OSPF-uc-default        32106   np      3000      1000      0
BGP-default            32118   np      4000      1250      0
```

確認

show controllers npu stats traps-all instance all location 0/RP0/CPU0 コマンドは、ローカルで処理されたパケットと、CPU によってドロップされたパケットを表示します。

```
Router# show controllers npu stats traps-all instance all location 0/RP0/CPU0

Trap Type                                    NPU  Trap TrapStats   Policer Packet
Packet
                                             ID    ID      ID              Accepted
Dropped
=============================================================================================
RxTrapMimSaMove(CFM_DOWM_MEP_DMM)            0    6    0x6         32037   0          0

RxTrapMimSaUnknown(RCY_CFM_DOWN_MEP_DMM)     0    7    0x7         32037   0          0

RxTrapAuthSaLookupFail (IPMC default)        0    8    0x8         32033   0          0

RxTrapSaMulticast                            0    11   0xb         32018   0          0

RxTrapArpMyIp                                0    13   0xd         32001   0          0

RxTrapArp                                    0    14   0xe         32001   11         0

RxTrapDhcpv4Server                           0    18   0x12        32022   0          0

RxTrapDhcpv4Client                           0    19   0x13        32022   0          0

RxTrapDhcpv6Server                           0    20   0x14        32022   0          0

RxTrapDhcpv6Client                           0    21   0x15        32022   0          0

RxTrapL2Cache_LACP                           0    23   0x17        32003   0          0

RxTrapL2Cache_LLDP1                          0    24   0x18        32004   0          0

RxTrapL2Cache_LLDP2                          0    25   0x19        32004   1205548    0

RxTrapL2Cache_LLDP3                          0    26   0x1a        32004   0          0
```

RxTrapL2Cache_ELMI	0	27	0x1b	32005	0	0
RxTrapL2Cache_BPDU	0	28	0x1c	32027	0	0
RxTrapL2Cache_BUNDLE_BPDU	0	29	0x1d	32027	0	0
RxTrapL2Cache_CDP	0	30	0x1e	32002	0	0
RxTrapHeaderSizeErr	0	32	0x20	32018	0	0
RxTrapIpCompMcInvalidIp	0	35	0x23	32018	0	0
RxTrapMyMacAndIpDisabled	0	36	0x24	32018	0	0
RxTrapMyMacAndMplsDisable	0	37	0x25	32018	0	0
RxTrapArpReply	0	38	0x26	32001	2693	0
RxTrapFibDrop	0	41	0x29	32018	0	0
RxTrapMTU	0	42	0x2a	32020	0	0
RxTrapMiscDrop	0	43	0x2b	32018	0	0
RxTrapL2AclDeny	0	44	0x2c	32034	0	0
Rx_UNKNOWN_PACKET	0	46	0x2e	32018	0	0
RxTrapL3AclDeny	0	47	0x2f	32034	0	0
RxTrapOamY1731MplsTp(OAM_SWOFF_DN_CCM)	0	57	0x39	32029	0	0
RxTrapOamY1731Pwe(OAM_SWOFF_DN_CCM)	0	58	0x3a	32030	0	0
RxTrapOamLevel	0	64	0x40	32023	0	0
RxTrapRedirectToCpuOamPacket	0	65	0x41	32025	0	0
RxTrapOamPassive	0	66	0x42	32024	0	0
RxTrap1588	0	67	0x43	32038	0	0
RxTrapExternalLookupError	0	72	0x48	32018	0	0
RxTrapArplookupFail	0	73	0x49	32001	0	0
RxTrapUcLooseRpfFail	0	84	0x54	32035	0	0
RxTrapMplsControlWordTrap	0	88	0x58	32015	0	0
RxTrapMplsControlWordDrop	0	89	0x59	32015	0	0
RxTrapMplsUnknownLabel	0	90	0x5a	32018	0	0
RxTrapIpv4VersionError	0	98	0x62	32018	0	0
RxTrapIpv4ChecksumError	0	99	0x63	32018	0	0
RxTrapIpv4HeaderLengthError	0	100	0x64	32018	0	0
RxTrapIpv4TotalLengthError	0	101	0x65	32018	0	0
RxTrapIpv4Ttl0	0	102	0x66	32008	0	0

RxTrapIpv4Ttl1	0	104	0x68	32008	0	0
RxTrapIpv4DipZero	0	106	0x6a	32018	0	0
RxTrapIpv4SipIsMc	0	107	0x6b	32018	0	0
RxTrapIpv6VersionError	0	109	0x6d	32018	0	0
RxTrapIpv6HopCount0	0	110	0x6e	32011	0	0
RxTrapIpv6LoopbackAddress	0	113	0x71	32018	0	0
RxTrapIpv6MulticastSource	0	114	0x72	32018	0	0
RxTrapIpv6NextHeaderNull	0	115	0x73	32010	0	0
RxTrapIpv6Ipv4CompatibleDestination	0	121	0x79	32018	0	0
RxTrapMplsTtl1	0	125	0x7d	32012	316278	2249
RxTrapUcStrictRpfFail	0	137	0x89	32035	0	0
RxTrapMcExplicitRpfFail	0	138	0x8a	32033	0	0
RxTrapOamp(OAM_BDL_DN_NON_CCM)	0	141	0x8d	32031	0	0
RxTrapOamEthUpAccelerated(OAM_BDL_UP_NON_CCM)	0	145	0x91	32032	0	0
RxTrapReceive	0	150	0x96	32017	125266112	0
RxTrapUserDefine_FIB_IPV4_NULL0	0	151	0x97	32018	0	0
RxTrapUserDefine_FIB_IPV6_NULL0	0	152	0x98	32018	0	0
RxTrapUserDefine_FIB_IPV4_GLEAN	0	153	0x99	32016	0	0
RxTrapUserDefine_FIB_IPV6_GLEAN	0	154	0x9a	32016	0	0
RxTrapUserDefine_IPV4_OPTIONS	0	155	0x9b	32006	0	0
RxTrapUserDefine_IPV4_RSVP_OPTIONS	0	156	0x9c	32007	0	0
RxTrapUserDefine	0	157	0x9d	32026	0	0
RxTrapUserDefine_BFD	0	163	0xa3	32028	0	0
RxTrapMC	0	181	0xb5	32033	0	0
RxNetflowSnoopTrap0	0	182	0xb6	32018	0	0
RxNetflowSnoopTrap1	0	183	0xb7	32018	0	0
RxTrapMimSaMove(CFM_DOWM_MEP_DMM)	1	6	0x6	32037	0	0
RxTrapMimSaUnknown(RCY_CFM_DOWN_MEP_DMM)	1	7	0x7	32037	0	0
RxTrapAuthSaLookupFail (IPMC default)	1	8	0x8	32033	0	0
RxTrapSaMulticast	1	11	0xb	32018	0	0
RxTrapArpMyIp	1	13	0xd	32001	0	0

関連コマンド

- lpts pifib hardware police
- flow ospf
- flow bgp
- show lpts pifib hardware police

ダイナミック LPTS フロー タイプの定義

ダイナミック LPTS フロー タイプ機能を使用すると、LPTS フロー タイプを設定できるとともに、TCAM 内の各フロー タイプの最大 LPTS エントリ数を定義できます。ダイナミック LPTS フロー タイプの設定はライン カードごとに行うため、複数のライン カードにわたって複数のプロファイルを設定できます。

ルータが起動すると、デフォルトのLPTSフロータイプがTCAMでプログラミングされます。フロー タイプそれぞれに、最大フロー エントリ数が事前に定義されています。後で、実行時にネットワーク要件に基づいてフロー タイプを選択し、最大フロー エントリ値も設定するオプションがあります。最大フロー エントリ値ゼロは、フロー タイプが設定されていないことを示します。

（注） 設定可能なフローと設定不能なフローの両方のデフォルト最大フロー値は、次の show コマンドの出力で確認できます。

```
show lpts pifib dynamic-flows statistics location <location specification>
```

設定可能なフロー タイプと設定不能なフロー タイプのリストを次の表に示します。また、設定可能なフロー タイプと設定不能なフロー タイプのリストを表示するには、**show lpts pifib dynamic-flows statistics location** コマンドも使用できます。

（注） すべてのフロー タイプに設定される最大 LPTS エントリの総数は、ライン カードあたり 8,000 エントリを超えないものとします。

設定例

次の例では、TCAM に BGP-known と ISIS-known の LPTS フロー タイプを設定し、ノード ロケーション 0/1/CPU0 に最大フロー エントリ 1800 と 500 を定義します。新しい最大値がデフォルト値を超えているため、他のフロー タイプを無効にして TCAM 内にスペースを作成し、ラインカードあたりのすべてのフロータイプの最大エントリ総数を8,000エントリを超えないようにする必要があります。そのため、次の例では RSVP-known フロー タイプがゼロに設定されています。

```
Router#configure
Router(config)#lpts pifib hardware dynamic-flows location 0/1/CPU0
Router(config-pifib-flows-per-node)#flow bgp-known max 1800
Router(config-pifib-flows-per-node)#flow ISIS-known max 500
Router(config-pifib-flows-per-node)#flow RSVP-known max 0
Router(config-pifib-flows-per-node)#commit
```

実行コンフィギュレーション

```
Router#show run lpts pifib hardware dynamic-flows location 0/1/CPU0
flow bgp known max 1800
flow isis-known 500
flow RSVP-known 0
```

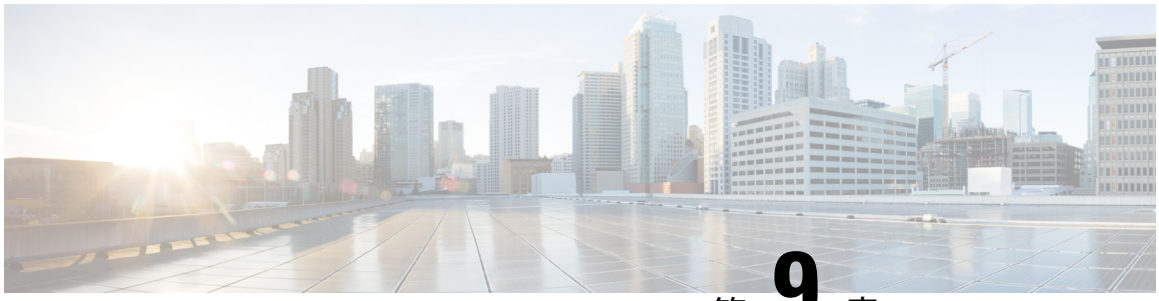
確認

次の show コマンドは、ダイナミック フローの統計情報を表示します。フロー タイプの BGP-known と ISIS-known が新たに設定した最大フロー エントリ値で TCAM に設定されていることを確認できます。また、RSVP-known フロータイプが無効になっていることも確認できます。

```
Router#show lpts pifib dynamic-flows statistics location 0/1/CPU0

 Dynamic-flows Statistics:
 -------------------------
 (C - Configurable, T - TRUE, F - FALSE, * - Configured)
 Def_Max  - Default Max Limit
 Conf_Max - Configured Max Limit
 HWCnt    - Hardware Entries Count
 ActLimit - Actual Max Limit
 SWCnt    - Software Entries Count
 P, (+)   - Pending Software Entries


  FLOW-TYPE           C  Def_Max Conf_Max     HWCnt/ActLimit      SWCnt P
 -------------------- -- ------- --------   -------/--------    ------- -
 Fragment             F        2        --         2/2                2
 OSPF-mc-known        T      600        --         2/600              2
 OSPF-mc-default      F        4        --         4/4                4
 OSPF-uc-known        T      300        --         1/300              1
 OSPF-uc-default      F        2        --         2/2                2
 ISIS-known           T      300       500       500/300             0
 ISIS-default         F        1        --         1/1                1
 BGP-known            T      900      1800      1800/900             0
 BGP-cfg-peer         T      900        --         0/900              0
 BGP-default          F        4        --         4/4                4
 PIM-mcast-default    F       40        --         0/40               0
 PIM-mcast-known      T      300        --         0/300              0
 PIM-ucast            F       40        --         2/40               2
 IGMP                 T     1200        --         0/1200             0
 ICMP-local           F        4        --         4/4                4
 ICMP-control         F        5        --         5/5                5
 ICMP-default         F        9        --         9/9                9
 ICMP-app-default     F        2        --         2/2                2
 LDP-TCP-known        T      300        --         0/300              0
 LDP-TCP-cfg-peer     T      300        --         0/300              0
 LDP-TCP-default      F       40        --         0/40               0
 LDP-UDP              T      300        --         0/300              0
 All-routers          T      300        --         0/300              0
 RSVP-default         F        4        --         1/4                1
 RSVP-known           T      300         0         0/300              0
```

```
 SNMP                 T      300       --         0/300               0
 SSH-known            T      150       --         0/150               0
 SSH-default          F       40       --         0/40                0
 TELNET-known         T      150       --         0/150               0
 TELNET-default       F        4       --         0/4                 0
 UDP-default          F        2       --         2/2                 2
 TCP-default          F        2       --         2/2                 2
 Raw-default          F        2       --         2/2                 2
 GRE                  F        4       --         0/4                 0
 VRRP                 T      150       --         150/150             0
 DNS                  T       40       --         0/40                0
 NTP-default          F        4       --         0/4                 0
 NTP-known            T      150       --         0/150               0
 TPA                  T        5       --         0/5                 0
----------------------------
 Local Limit : 7960/8000 /*The sum of maximum flow entries configured for all flow types

                                  per line card is less than 8000*/
 HWCnt/SWCnt : 45/51
----------------------------
```

上記の show コマンド出力では、最後の列の **P** でそのフロー タイプの保留中ソフトウェア フロー エントリを指定します。



第 9 章

VRRP の実装

VRRP の設定

仮想ルータ冗長プロトコル（VRRP）機能を使用すると、ファーストホップ IP ルータでの透過的なフェールオーバーが可能になり、ルータ グループが単一の仮想ルータを形成できるようになります。VRRP と関連する概念の詳細については、VRRP の概要 （131 ページ）を参照してください。

VRRP 設定の制約事項

- Cisco NCS 5500、Cisco NCS 560、および Cisco NCS 540 シリーズ ルータでは、最大 64 の VRRP グループがサポートされています。
- BFD を BVI とともに設定し、すべての BVI が同じシャーシ（デフォルト）MAC を共有している場合、VRRP スケールは 13 に減少します。VRRP スケールが 11 に減少するまで、このモードでカスタム BVI MAC を使用することはできません。
- BFD を BVI とともに設定し、すべての BVI に 1 つのカスタム MAC、または 1 つのカスタム MAC とシャーシ（デフォルト）MAC の組み合わせが設定されている場合は、VRRP スケールが 11 に減少します。
- BVI インターフェイスを介した VRRP がサポートされています。
- ICMP リダイレクトはサポートされていません。

VRRP の概要

仮想ルータ冗長プロトコル（VRRP）機能を使用すると、ファーストホップ IP ルータでの透過的なフェールオーバーが可能になり、ルータ グループが単一の仮想ルータを形成できるようになります。

VRRP の概要

LAN クライアントは、動的プロセスまたは静的設定を使用して、特定のリモート宛先への最初のホップとなるルータを決定します。次に、ダイナミック ルータ ディスカバリのクライアント例を示します。

- プロキシ ARP：クライアントはアドレス解決プロトコル（ARP）を使用して到達すべき宛先を取得します。ルータは独自の MAC アドレスで ARP 要求に応答します。
- ルーティング プロトコル：クライアントはダイナミック ルーティング プロトコルのアップデートを（ルーティング情報プロトコル（RIP）などから）受信し、独自のルーティング テーブルを形成します。
- IRDP（ICMP Router Discovery Protocol）クライアント：クライアントはインターネット制御メッセージ プロトコル（ICMP）ルータ ディスカバリ クライアントを実行します。

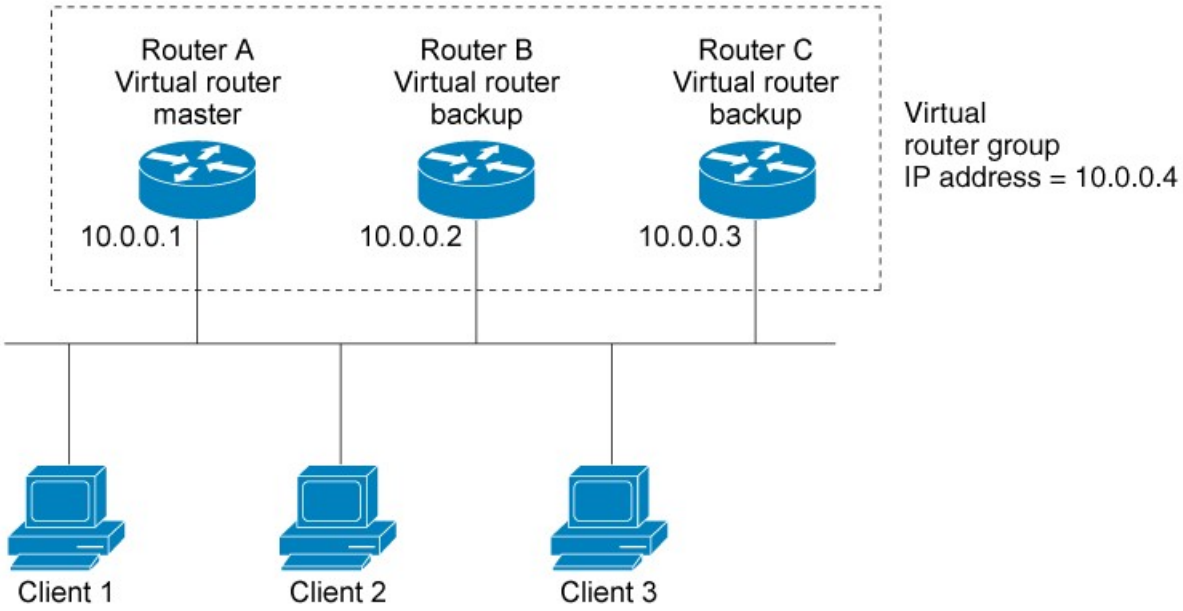
ダイナミック ディスカバリ プロトコルには、LAN クライアントにおいて、設定および処理のオーバーヘッドが発生するという短所があります。また、ルータが機能を停止したときに、別のルータへの切り替え処理が遅くなる可能性があります。

ダイナミック ディスカバリ プロトコルの代わりに、クライアント上でデフォルト ルータをスタティックに設定することもできます。このアプローチでは、クライアントの設定と処理は簡略化されますが、単一障害点が生じます。デフォルト ゲートウェイで障害が発生した場合、LAN クライアントの通信はローカル IP ネットワーク セグメントに限定され、ネットワークの他の部分から切り離されます。

仮想ルータ冗長プロトコル（VRRP）機能により、この静的設定の問題を解決できます。VRRP は、ファーストホップ IP ルータの透過的なフェールオーバーを可能にするように設計された IP ルーティング冗長プロトコルです。VRRP を使用すると、ルータのグループで 1 つの仮想ルータを形成できます。 これにより、仮想ルータをデフォルト ゲートウェイとして使用するように、LAN クライアントを設定できます。ルータのグループを表す仮想ルータは、*VRRP* グループとも呼ばれます。

例として、図 12 : 基本的な VRRP トポロジ （133 ページ）に、VRRP が設定された LAN トポロジを示します。この例では、ルータ A、B、および C は仮想ルータで構成される *VRRP* ルータ（VRRP を実行するルータ）です。仮想ルータの IP アドレスは、ルータ A のインターフェイスに設定されたアドレス（10.0.0.1）と同じです。

図 *12* : 基本的な *VRRP* トポロジ



仮想ルータはルータ A の物理インターフェイスの IP アドレスを使用するため、ルータ A はマスター仮想ルータのロールを担い、*IP* アドレス所有者とも呼ばれます。ルータ A は、マスター仮想ルータとして、仮想ルータの IP アドレスを管理し、この IP アドレスに送信されたパケットの転送を行います。クライアント 1 ～ 3 には、デフォルト ゲートウェイの IP アドレス 10.0.0.1 が設定されています。

ルータ B および C は、バックアップ仮想ルータとして機能します。マスター仮想ルータに障害が発生すると、高いプライオリティが設定されているルータがマスター仮想ルータになり、LAN ホストに対して中断なくサービスが提供されます。ルータ A は、回復すると、再びマスター仮想ルータになります。

（注） 仮想ルータが接続されているスイッチ ポートでは、スパニング ツリー プロトコル（STP）を無効にすることをお勧めします。スイッチがこれらのプロトコルをサポートしている場合に、RSTP または rapid-PVST を有効にします。

複数の仮想ルータのサポート

ルータ インターフェイスには、最大 100 の仮想ルータを設定できます。ルータ インターフェイスには、最大 256 の仮想ルータを設定できます。ルータ インターフェイスがサポートできる実際の仮想ルータの数は、次の要因によって異なります。

- ルータの処理能力
- ルータのメモリの能力
- 複数の MAC アドレスのルータ インターフェイス サポート

1つのルータインターフェイス上に複数の仮想ルータが設定されているトポロジでは、そのインターフェイスは 1 つ以上の仮想ルータのマスター、および 1 つ以上の仮想ルータのバックアップとして動作することができます。

VRRP ルータ プライオリティ

VRRP 冗長性スキームの重要な一面に、VRRP ルータ プライオリティがあります。プライオリティにより、各 VRRP ルータが果たすロールと、マスター仮想ルータが機能を停止したときにどのようなことが起こるかが決定されます。

VRRP ルータが仮想ルータの IP アドレスと物理インターフェイスの IP アドレスのオーナーである場合には、このルータがマスター仮想ルータとして機能します。

IP アドレスのオーナーである VRRP ルータが存在しない場合は、VRRP ルータのプライオリティおよびプリエンプション設定の組み合わせにより、VRRP ルータがマスターとして機能するか、またはバックアップ仮想ルータとして機能するかが決まります。デフォルトでは、最高のプライオリティを持つVRRP ルータがマスターとして機能し、その他のすべてがバックアップとして機能します。プライオリティにより、マスター仮想ルータが機能を停止した場合にマスター仮想ルータになる優先順位も決まります。vrrp priority コマンドを使用して 1 ～ 254 の値を設定し、各バックアップ仮想ルータのプライオリティを設定できます。

たとえば、LAN トポロジのマスター仮想ルータであるルータ A が機能を停止した場合、選択プロセスが実行されて、バックアップ仮想ルータ B または C が引き継ぐかどうかが決定されます。ルータ B とルータ C がそれぞれプライオリティ 101 と 100 に設定されている場合、プライオリティの高いルータ B がマスター仮想ルータになります。ルータ B とルータ C が両方ともプライオリティ 100 に設定されている場合、IP アドレスがより高いバックアップ仮想ルータが選択されてマスター仮想ルータになります。

デフォルトでは、プリエンプティブ スキームが有効になっており、使用可能になった高いプライオリティのバックアップ仮想ルータが、現在のマスター仮想ルータから引き継ぎます。このプリエンプティブ スキームを無効にするには、vrrp preempt disable コマンドを使用します。プリエンプションが無効になっている場合、元のプライオリティがより高いマスターの障害時にマスターになるように選択されたバックアップ仮想ルータは、元のマスター仮想ルータが回復して再び使用可能になっても、マスターのままとなります。

VRRP のアドバタイズメント

マスター仮想ルータは、同じグループ内の他の VRRP ルータに VRRP アドバタイズメントを送信します。アドバタイズメントでは、マスター仮想ルータのプライオリティと状態を伝えます。VRRP アドバタイズメントは IP パケットにカプセル化され、VRRP グループに割り当てられた IP バージョン 4 マルチキャスト アドレスに送信されます。アドバタイズメントは、デフォルトで 1 秒に 1 回送信されますが、この間隔は設定可能です。

VRRP の利点

VRRP の利点は、次のとおりです。

- 冗長性：VRRP により、複数のルータをデフォルト ゲートウェイ ルータとして設定できるようになるため、ネットワークに単一障害点が生じる可能性を低減できます。

- ロード シェアリング：LAN クライアントとの間のトラフィックを複数のルータで共有するようにVRRPを設定できるため、利用可能なルータ間でより均等にトラフィックの負荷を分散できます。
- 複数の仮想ルータ：プラットフォームが複数のMACアドレスをサポートする場合、VRRPは、ルータのインターフェイス上で最大 100 の仮想ルータ（VRRP グループ）をサポートします。ルータインターフェイスには、最大256の仮想ルータを設定できます。複数の仮想ルータをサポートすることで、LAN トポロジ内で冗長化とロード シェアリングを実装できます。
- 複数の IP アドレス：仮想ルータは、セカンダリ IP アドレスを含む、複数の IP アドレスを管理できます。そのため、イーサネットインターフェイスに複数のサブネットを設定した場合、サブネットごとに VRRP を設定できます。
- プリエンプション：VRRP の冗長性スキームにより、障害が発生したマスター仮想ルータを引き継いだバックアップ仮想ルータを、使用可能になった高いプライオリティのバックアップ仮想ルータに切り替えることができます。
- テキスト認証：簡易テキスト パスワードを設定して、仮想ルータを構成している VRRP ルータから受信した VRRP メッセージが認証されたことを確認できます。
- アドバタイズメント プロトコル：VRRP では、VRRP アドバタイズメントに、専用のインターネット割り当て番号局（IANA）規格マルチキャスト アドレス（224.0.0.18）を使用します。このアドレッシング方式によって、マルチキャストを提供するルータ数が最小限になり、テスト機器でセグメント上のVRRPパケットを正確に識別できるようになります。IANA では VRRP に IP プロトコル番号 112 を割り当てています。

VRRP のホット リスタート

1 つのグループで VRRP プロセスの障害が発生した場合には、ピア VRRP マスター ルータ グループで強制的にフェールオーバーが行われないようにする必要があります。ホットリスタートはウォームRPフェールオーバーをサポートしており、ピアVRRPルータへの強制的なフェールオーバーは発生しません。

BVI を介した VRRP の概要

Virtual Router Redundancy Protocol（VRRP）プロトコルは、デフォルト ゲートウェイの冗長性を提供します。これにより、ルータのグループを、1 台のルータがマスター ルータ、他のルータがバックアップ ルータとして機能する単一の仮想デフォルト ゲートウェイ ルータとして動作させることができます。

BVI（ブリッジグループ仮想インターフェイス）は、ブリッジ グループに L3 またはルーテッド機能を提供する仮想インターフェイスです。L2機能はブリッジグループに含まれるインターフェイスに適用され、BVI はそのブリッジ グループのルーテッド インターフェイスになります。

通常、VRRPセッションは同じホーム ネットワーク内にある複数のルータのインターフェイス上で実行されます。ただし、BVIを介したVRRPセッションを設定することは可能です。した

がって、物理インターフェイスではなく、複数のルータの BVI インターフェイス間で VRRP セッションを実行できます。

IPv4 ネットワーク用 VRRP の設定

ここでは、IPv4 ネットワークの VRRP を設定および確認する手順を説明します。

設定

IPv4 ネットワークに VRRP を設定するには、次の設定を使用します。

（注） ルータ上で VRRP 設定をコミットする際、VRRP グループの動作を制御する特定のカスタマイズ（後述）が推奨されます。以下のカスタマイズを設定しない場合、ルータが VRRP グループを制御してただちにマスター仮想ルータの役割を担います。

```
/* Enter the interface configuration mode and configure an IPv4 address for the interface.
 */
Router(config)# interface gigabitEthernet 0/0/0/1
Router(config-if)# ipv4 address 10.10.10.1 255.255.255.0
Router(config-if)# no shut
Router(config-if)# commit
Fri Dec  8 13:49:24.142 IST
Router:Dec  8 13:49:24.285 : ifmgr[402]: %PKT_INFRA-LINK-3-UPDOWN : Interface
GigabitEthernet0/0/0/1, changed state to Down
Router:Dec  8 13:49:24.711 : ifmgr[402]: %PKT_INFRA-LINK-3-UPDOWN : Interface
GigabitEthernet0/0/0/1, changed state to Up

Router(config-if)# exit
Router(config)# do show ip int brief
Fri Dec  8 13:50:05.505 IST

Interface                      IP-Address      Status          Protocol Vrf-Name
GigabitEthernet0/0/0/0         unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/1         10.10.10.1      Up              Up       default
GigabitEthernet0/0/0/2         unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/3         unassigned      Shutdown        Down     default
GigabitEthernet0/0/0/4         unassigned      Shutdown        Down     default


/* Enter the VRRP configuration mode and add the configured interface. */
Router(config)# router vrrp
Router(config-vrrp)# interface GigabitEthernet 0/0/0/1

/* CUSTOMIZATION: Configure a delay for the startup of the state machine when the interface
 comes up. */
Router(config-vrrp)# delay minimum 2 reload 10 */

/* Configure VRRP version 3 for IPv4 */
Router(config-vrrp-if)# address-family ipv4 vrrp 100 version 3
Router(config-vrrp-virtual-router)# address 10.10.10.1

/* CUSTOMIZATION: Disable the installation of routes for the VRRP virtual addresses. */
Router(config-vrrp-virtual-Router)# accept-mode disable

/* CUSTOMIZATION: Set a priority for the virtual Router. */
Router(config-vrrp-virtual-Router)# priority 254
```

```
/* CUSTOMIZATION: Configure a preempt delay value that controls the selection of the
master virtual Router. */
Router(config-vrrp-virtual-Router)# preempt delay 15

/* CUSTOMIZATION: Configure the interval between successive advertisements by the master
 virtual Router. */
Router(config-vrrp-virtual-Router)#timer 4

/* CUSTOMIZATION: Configure VRRP to track an interface. */
Router(config-vrrp-virtual-Router)# track interface GigabitEthernet0/0/0/1 30

 /* Commit the configuration */
Router(config-vrrp-virtual-Router)# commit
```

IPv4 ネットワークに VRRP が正常に設定されました。

Validation

次のコマンドを使用して設定を検証します。

```
/* Validate the configuration */
Router(config-vrrp-virtual-router)# do show run interface GigabitEthernet 0/0/0/1
Fri Dec  8 15:04:38.140 IST
interface GigabitEthernet0/0/0/1
 ipv4 address 10.10.10.1 255.255.255.0
!
```

```
Router(config)# show running-config router vrrp
Fri Dec  8 13:50:18.959 IST
router vrrp
 interface GigabitEthernet0/0/0/1
   delay minimum 2 reload 10
  address-family ipv4
   vrrp 100 version 3
    priority 254
    preempt delay 15
    timer 4
    track interface GigabitEthernet0/0/0/2 30
    address 10.10.10.1
    accept-mode disable
   !
  !
 !
```

```
Router(config-vrrp-virtual-router)# do show vrrp ipv4 interface gigabitEthernet 0/0/0/1
Fri Dec  8 15:02:56.952 IST
IPv4 Virtual Routers:
                     A indicates IP address owner
                     | P indicates configured to preempt
                     | |
Interface   vrID Prio A P State    Master addr     VRouter addr
Gi0/0/0/1    100  255 A P Master   local           10.10.10.1
```

```
Router(config-vrrp-virtual-router)# end
Router# show vrrp detail
Fri Dec  8 15:08:36.469 IST
GigabitEthernet0/0/0/1 - IPv4 vrID 100
  State is Master, IP address owner
```

```
    1 state changes, last state change 01:19:06
    State change history:
    Dec  8 13:49:30.147 IST  Init     -> Master   Delay timer expired
  Last resign sent:     Never
  Last resign received: Never
  Virtual IP address is 10.10.10.1
  Virtual MAC address is 0000.5E00.0164, state is active
  Master router is local
  Version is 3
  Advertise time 1 secs
    Master Down Timer 3.003 (3 x 1 + (1 x 1/256))
  Minimum delay 1 sec, reload delay 5 sec
  Current priority 255
    Configured priority 100, may preempt
      minimum delay 0 secs
```

IPv4 ネットワークの VRRP が正常に検証されました。

IPv6 ネットワーク用 VRRP の設定

ここでは、IPv6 ネットワークの VRRP を設定および確認する手順を説明します。

設定

次のサンプルには、IPv6 ネットワークの VRRP の設定とカスタマイズが含まれています。

（注） ルータ上で VRRP 設定をコミットする際、VRRP グループの動作を制御する特定のカスタマイズ（後述）が推奨されます。以下のカスタマイズを設定しない場合、ルータが VRRP グループを制御してただちにマスター仮想ルータの役割を担います。

```
/* Enter the interface configuration mode and configure an IPv6 address */
Router# interface GigabitEthernet 0/0/0/2
Router(config-if)# ipv6 address 10::1/64
Router(config-if)# no shut

/* Exit the interface configuration mode and enter the vrrp configuration mode */
Router(config-if)# exit
Router(config)# Router vrrp

/* Add the configured interface for VRRP */
Router(config-vrrp)# interface GigabitEthernet 0/0/0/2

/* CUSTOMIZATION: Configure a delay for the startup of the state machine when the interface
 comes up. */
Router(config-vrrp)# delay minimum 2 reload 10 */

/* Enable the IPv6 global and link local address family on the interface  */
Router(config-vrrp-if)# address-family ipv6 vrrp 50
Router(config-vrrp-virtual-Router)# address linklocal autoconfig

/* CUSTOMIZATION: Disable the installation of routes for the VRRP virtual addresses. */
Router(config-vrrp-virtual-Router)# accept-mode disable

/* CUSTOMIZATION: Set a priority for the virtual Router. */
```

```
Router(config-vrrp-virtual-Router)# priority 254

/* CUSTOMIZATION: Configure a preempt delay value that controls the selection of the
master virtual Router. */
Router(config-vrrp-virtual-Router)# preempt delay 15

/* CUSTOMIZATION: Configure the interval between successive advertisements by the master
 virtual Router. */
Router(config-vrrp-virtual-Router)#timer 4

/* CUSTOMIZATION: Configure VRRP to track an interface. */
Router(config-vrrp-virtual-Router)# track interface GigabitEthernet0/0/0/2 30

/* Commit the configuration */
Router(config-vrrp-virtual-Router)# commit
```

IPv6 ネットワークに VRRP が正常に設定されました。

Validation

次のコマンドを使用して設定を検証します。

```
/* Validate the configuration */
Router(config-vrrp-virtual-router)# do show run interface GigabitEthernet 0/0/0/2
Fri Dec  8 14:55:48.378 IST
interface GigabitEthernet0/0/0/2
 ipv6 address 10::1/64
!
```

```
Router(config-vrrp-virtual-router)# do show running-config router vrrp
...
router vrrp
 interface GigabitEthernet0/0/0/2
  delay minimum 2 reload 10
  address-family ipv6
   vrrp 50
    priority 254
    preempt delay 15
    timer 4
    track interface GigabitEthernet0/0/0/2 30
    address linklocal autoconfig
    accept-mode disable
   !
  !
 !
!
```

```
Router(config-vrrp-virtual-router)# do show vrrp ipv6 interface gigabitEthernet 0/0/0/2
Fri Dec  8 14:59:25.547 IST
IPv6 Virtual Routers:
                     A indicates IP address owner
                     | P indicates configured to preempt
                     | |
Interface   vrID Prio A P State    Master addr     VRouter addr
Gi0/0/0/2      50  254   P Master   local
                                                fe80::200:5eff:fe00:203
```

```
Router(config-vrrp-virtual-router)# end
Router# show vrrp detail
Fri Dec  8 15:08:36.469 IST
GigabitEthernet0/0/0/2 - IPv6 vrID 50
```

```
State is Master
  2 state changes, last state change 00:18:01
  State change history:
  Dec  8 14:50:23.326 IST  Init     -> Backup   Virtual IP configured
  Dec  8 14:50:35.365 IST  Backup   -> Master   Master down timer expired
Last resign sent:     Never
Last resign received: Never
Virtual IP address is fe80::200:5eff:fe00:203
Virtual MAC address is 0000.5E00.0203, state is active
Master router is local

Advertise time 4 secs
  Master Down Timer 12.031 (3 x 4 + (2 x 4/256))
Minimum delay 2 sec, reload delay 10 sec
Current priority 254
  Configured priority 254, may preempt
    minimum delay 15 secs
  Tracked items: 1/1 up: 0 decrement
    Object name               State     Decrement
    GigabitEthernet0/0/0/2        Up            30
```

IPv6 ネットワークの VRRP が正常に検証されました。

BVI を介した VRRP の設定

BVI を介した VRRP セッションを設定するには、次の設定を完了する必要があります。

1. 一連のインターフェイスを L2 インターフェイスとして設定します。
2. ブリッジ グループを設定します。
3. BVI を設定します。
4. BVI を介した VRRP を設定します。

設定例

```
/* Enter the global configuration mode and Configure a set of interfaces as L2 interfaces
 */
Router# configure
Router(config)# interface HundredGigE0/0/0/0.1 l2transport
Router(config-subif)# exit
Router(config)# interface HundredGigE0/0/0/1.1 l2transport
Router(config-subif)# commit
Router(config-subif)# exit

/* Enter the Layer 2 VPN configuration mode and Configure a bridge group */
Router(config)# l2vpn
Router(config-l2vpn)# bridge group 5
Router(config-l2vpn-bg)# bridge-domain 5
Router(config-l2vpn-bg-bd)# interface HundredGigE 0/0/0/0.1
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# interface HundredGigE 0/0/0/1.1
Router(config-l2vpn-bg-bd-ac)# exit
Router(config-l2vpn-bg-bd)# routed interface BVI 10
Router(config-l2vpn-bg-bd-bvi)# commit
Router(config-l2vpn-bg-bd-bvi)# exit
```

```
/* Configure a BVI in the global configuration mode*/
Router(config-l2vpn-bg-bd)# interface BVI 10

Router(config-if)# ipv4 address 209.165.200.225 255.255.255.0
Router(config-if)# ipv6 address 2001:DB8:A:B::1/64
Router(config-if)# commit

/* Configure VRRP over BVI in the global configuration mode for IPv4 address*/
Router(config)# router VRRP
Router(config-vrrp)# interface BVI 10
Router(config-vrrp-if)# address-family ipv4
Router(config-vrrp-address-family)# VRRP 10
Router config-vrrp-virtual-router)# priority 101
Router config-vrrp-virtual-router)# 209.165.200.226
Router (config-vrrp-virtual-router)# commit


/* Configure VRRP over BVI in the global configuration mode for IPv6 address*/
Router(config)# router VRRP
Router(config-vrrp)# interface BVI 10
Router(config-vrrp-if)# address-family ipv6
Router(config-vrrp-address-family)# VRRP 11
Router config-vrrp-virtual-router)# address global 2001:DB8:A:B::2/64
Router config-vrrp-virtual-router)#  address linklocal autoconfig
Router (config-vrrp-virtual-router)# commit
```

確認

ブリッジ ドメインの詳細を確認するには、次のコマンドを使用します。

```
Router# show l2vpn bridge-domain detail

Legend: pp = Partially Programmed.
Bridge group: 10, bridge-domain: 10, id: 0, state: up, ShgId: 0, MSTi: 0
  Coupled state: disabled
  VINE state: BVI Resolved
  MAC learning: enabled
  MAC withdraw: enabled
    MAC withdraw for Access PW: enabled
    MAC withdraw sent on: bridge port up
    MAC withdraw relaying (access to access): disabled
  Flooding:
    Broadcast & Multicast: enabled
    Unknown unicast: enabled
  MAC aging time: 300 s, Type: inactivity
  MAC limit: 64000, Action: none, Notification: syslog
  MAC limit reached: no, threshold: 75%
  MAC port down flush: enabled
  MAC Secure: disabled, Logging: disabled
  Split Horizon Group: none
  Dynamic ARP Inspection: disabled, Logging: disabled
  IP Source Guard: disabled, Logging: disabled
  DHCPv4 Snooping: disabled
  DHCPv4 Snooping profile: none
  IGMP Snooping: disabled
  IGMP Snooping profile: none
  MLD Snooping profile: none
  Storm Control: disabled
  Bridge MTU: 1500
  MIB cvplsConfigIndex: 1
  Filter MAC addresses:
  P2MP PW: disabled
  Create time: 25/07/2018 19:04:41 (3w5d ago)
```

```
  No status change since creation
  ACs: 2 (2 up), VFIs: 0, PWs: 0 (0 up), PBBs: 0 (0 up), VNIs: 0 (0 up)
  List of ACs:
    AC: BVI10, state is up
      Type Routed-Interface
      MTU 1514; XC ID 0x80000002; interworking none
      BVI MAC address:
        008a.9651.e0e0
      Split Horizon Group: Access
    AC: Bundle-Ether10.1, state is up
      Type VLAN; Num Ranges: 1
      Rewrite Tags: []
      VLAN ranges: [10, 10]
      MTU 1500; XC ID 0xa0000001; interworking none
      MAC learning: enabled
      Flooding:
        Broadcast & Multicast: enabled
        Unknown unicast: enabled
      MAC aging time: 300 s, Type: inactivity
      MAC limit: 64000, Action: none, Notification: syslog
      MAC limit reached: no, threshold: 75%
      MAC port down flush: enabled
      MAC Secure: disabled, Logging: disabled
      Split Horizon Group: none
      Dynamic ARP Inspection: disabled, Logging: disabled
      IP Source Guard: disabled, Logging: disabled
      DHCPv4 Snooping: disabled
      DHCPv4 Snooping profile: none
      IGMP Snooping: disabled
      IGMP Snooping profile: none
      MLD Snooping profile: none
      Storm Control: bridge-domain policer
      Static MAC addresses:
      Statistics:
        packets: received 0 (multicast 0, broadcast 0, unknown unicast 0, unicast 0),
sent 4429828
         bytes: received 0 (multicast 0, broadcast 0, unknown unicast 0, unicast 0), sent
 344854904
        MAC move: 0
      Storm control drop counters:
        packets: broadcast 0, multicast 0, unknown unicast 0
        bytes: broadcast 0, multicast 0, unknown unicast 0
      Dynamic ARP inspection drop counters:
        packets: 0, bytes: 0
      IP source guard drop counters:
        packets: 0, bytes: 0
  List of Access PWs:
  List of VFIs:
  List of Access VFIs:
```

VRRP の詳細を表示するには、次のコマンドを使用します。

```
Router# show vrrp ipv4 detail

BVI10 - IPv4 vrID 10
  State is Master
    63 state changes, last state change 5d20h
    State change history:
    Aug 15 20:30:47.986 UTC  Backup   -> Init     Interface Down update
    Aug 15 20:30:50.642 UTC  Init     -> Backup   Delay timer expired
    Aug 15 20:30:51.304 UTC  Backup   -> Init     Interface Down update
    Aug 15 20:31:44.957 UTC  Init     -> Backup   Delay timer expired
    Aug 15 20:31:48.562 UTC  Backup   -> Master   Master down timer expired
  Last resign sent:     Aug 15 20:30:20.700 UTC
```

```
  Last resign received: Jul 25 19:04:21.466 UTC
  Virtual IP address is 209.165.200.226
  Virtual MAC address is 0000.5E00.010a, state is active
  Master router is local
  Version is 2
  Advertise time 1 secs
    Master Down Timer 3.605 (3 x 1 + (155 x 1/256))
  Minimum delay 1 sec, reload delay 5 sec
  Current priority 101
    Configured priority 101, may preempt
      minimum delay 0 secs

Router# show vrrp ipv6 detail

BVI10 - IPv6 vrID 11
  State is Master
    63 state changes, last state change 5d20h
    State change history:
    Aug 15 20:30:48.032 UTC  Backup   -> Init     Interface Down update
    Aug 15 20:30:50.517 UTC  Init     -> Backup   Delay timer expired
    Aug 15 20:30:51.348 UTC  Backup   -> Init     Interface Down update
    Aug 15 20:31:44.996 UTC  Init     -> Backup   Delay timer expired
    Aug 15 20:31:48.605 UTC  Backup   -> Master   Master down timer expired
  Last resign sent:     Aug 15 20:30:20.702 UTC
  Last resign received: Never
  Virtual IP address is fe80::200:5eff:fe00:20b
    Secondary Virtual IP address is 2001:DB8:A:B::2/64
  Virtual MAC address is 0000.5E00.020b, state is active
  Master router is local
  Version is 3
  Advertise time 1 secs
    Master Down Timer 3.609 (3 x 1 + (156 x 1/256))
  Minimum delay 1 sec, reload delay 5 sec
  Current priority 100
    Configured priority 100, may preempt
      minimum delay 0 secs
```

状態変更ロギングのディセーブル化

設定例

syslog を介して VRRP 状態変更イベントをロギングするタスクをディセーブルにします。

```
Router#configure
Router(config)#router vrrp
router(config-vrrp)#message state disable
router(config-vrrp)#commit
```

VRRPのマルチグループオプティマイゼーション（MGO）の有効化

設定例

Virtual Router Redundancy Protocol（VRRP）のマルチ グループ オプティマイゼーションは、多くのサブインターフェイスから構成される導入環境で制御トラフィックを削減するためのソリューションです。VRRP 制御トラフィックの実行を 1 つのセッションに制限することにより、冗長性要求が同一のサブインターフェイスに対して制御トラフィックが減少します。他のすべてのセッションはこのプライマリ セッションのスレーブになり、プライマリ セッションから状態を継承します。

VRRP セッション名

```
Router#configure
Router(config)#router vrrp
router(config-vrrp)#interface TenGigE 0/0/0/2
router(config-vrrp-if)#address-family ipv4
router(config-vrrp-address-family)#vrrp 1
/* Enables VRRP group configuration mode on a specific interface. */

router(config-vrrp-vritual-router)#name s1
/* Specifies the VRRP session name. */

router(config-vrrp-gp)#commit
```

Slave Follow

```
Router#configure
Router(config)#router vrrp
router(config-vrrp)#interface TenGigE 0/0/0/2
router(config-vrrp-if)#address-family ipv4

router(config-vrrp-address-family)#vrrp 2 slave
/* Enables VRRP slave configuration mode on a specific interface. */

router(config-vrrp-slave)#follow m1
/* Configures a slave follow. Instructs the slave group to inherit its state from the
specified group, m1 (MGO session name). */

router(config-vrrp-slave)#address 10.2.3.2
/* Specifies the primary virtual IPv4 address for slave group. */

router(config-vrrp-slave)#address 10.2.3.3 secondary
/* Specifies the secondary virtual IPv4 address for slave group. */

router(config-vrrp-gp)#commit
```

スレーブ グループのライマリおよびセカンダリの仮想 IPv4アドレス

```
Router#configure
Router(config)#router vrrp
router(config-vrrp)#interface TenGigE 0/0/0/2
router(config-vrrp-if)#address-family ipv4

router(config-vrrp-address-family)#vrrp 2 slave
```

```
/* Enables VRRP slave configuration mode on a specific interface. */

router(config-vrrp-slave)#address 10.2.3.2
/* Specifies the primary virtual IPv4 address for slave group. */

router(config-vrrp-slave)#address 10.2.3.3 secondary
/* Specifies the secondary virtual IPv4 address for slave group. */

router(config-vrrp-slave)#commit
```

実行コンフィギュレーション

```
Router#show running-config router vrrp 1
router vrrp
interface TenGigE 0/0/0/2
address-family ipv4
vrrp 1
name s1
!

/* Slave group */
Router#show running-config router vrrp 2
router vrrp
interface TenGigE 0/0/0/2
address-family ipv4
vrrp 2 slave
follow m1
address 10.2.3.2
address 10.2.3.3 secondary
!
```

VRRP イベントに関する SNMP サーバ通知の設定

MIB の VRRP サポート

VRRP を使用すると、障害が発生したとき、ルータが 1 つ以上の IP アドレスを引き継ぐことができます。たとえば、障害の発生したルータがデフォルト ゲートウェイであったために、ホストからの IP トラフィックがそのルータに到達した場合、そのトラフィックは制御を引き継いだ VRRP ルータによって透過的に転送されます。VRRP を使用する場合、ダイナミック ルーティングやルータ ディスカバリ プロトコルの設定を各エンドホストで行う必要はありません。仮想ルータに割り当てる IP アドレスを制御する VRRP ルータはマスターと呼ばれ、送信されたパケットをそれらの IP アドレスに転送します。この選択プロセスにより、マスターが使用不可になった場合の転送責任のダイナミック フェールオーバー（スタンバイ）が提供されます。これにより、LAN 上の仮想ルータ IP アドレスをデフォルトの最初のホップ ルータとしてエンドホストが使用するようにできます。

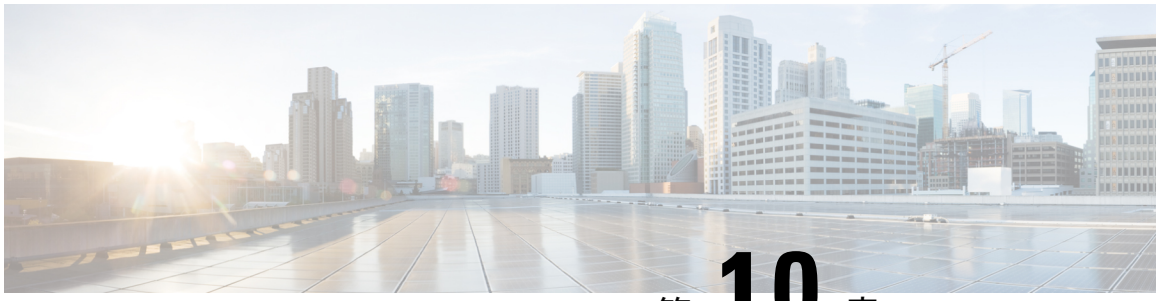
VRRP を使用することで得られるメリットは、ダイナミック ルーティングや Router Discovery Protocol をエンドホストごとに設定する必要なく、デフォルト パスの可用性が向上することです。Simple Network Management Protocol（SNMP）トラップは、仮想ルータ（スタンバイ）がマスター状態に移行した場合、またはスタンバイ ルータがマスターになった場合に、状態変更に関する情報を提供します。

設定例

VRRP に対して SNMP サーバ通知（トラップ）を有効にします。

```
Router#configure
Router(config)#snmp-server traps vrrp events
router(config)#commit
```

SNMP サーバ通知の詳細を表示するには、**show snmp traps details** コマンドを使用します。



第 10 章

TCP 転送、UDP 転送の設定に関する情報

TCP 転送、UDP 転送、および RAW 転送を設定するには、次の概念を理解しておく必要があります。

グレースフル リスタート

BGP ルーティング プロトコル情報がフェールオーバー後に復元されている間に、転送情報ベース（FIB）内の既知のルートでデータ パケットを転送するように、BGP の無停止フォワーディング（NSF）を使用できます。NSF では、BGP ピアはルーティング フラップと無縁です。フェールオーバー時に、データ トラフィックはインテリジェント モジュール経由で転送され、スタンバイ スーパーバイザがアクティブになります。

シスコ ルータでコールド リブートが発生した場合、ネットワークはルータへのトラフィック転送を中止し、ネットワーク トポロジからルータを削除します。この状況では、BGP は非グレースフル リスタートになり、すべてのルートが削除されます。シスコ オペレーティング システムがスタートアップ コンフィギュレーションを適用すると、BGP はピアリング セッションを再確立して、ルートを再学習します。

デュアル スーパーバイザ構成のシスコ ルータでは、ステートフル スーパーバイザ スイッチオーバーが実行されます。スイッチオーバーの間、BGP は無停止フォワーディングを使用し、FIB の情報に基づいてトラフィックを転送します。システムがネットワーク トポロジから取り除かれることはありません。ネイバーが再起動しているルータは、「ヘルパー」と呼ばれます。スイッチオーバー後、グレースフル リスタート動作が開始されます。この処理が進行中の際、2 つのルータはネイバー関係を再確立し、これらの BGP ルートを交換します。このネイバー関係が再起動中でも、ヘルパーは再起動中のピアを指すプレフィックスの転送を続け、再起動中のルータはピアにトラフィックを転送し続けます。再起動中のルータがグレースフル リスタート可能なすべての BGP ピアを持つ場合、グレースフル リスタートが完了し、BGP は再び動作可能なネイバーを通知します。

TCP の概要

TCPは、2つのコンピュータ システムがデータを転送するために交換する、データおよび確認応答の形式が指定されたコネクション型プロトコルです。また、TCPでは、データを正しく到達させるために、コンピュータが使用する手順も指定されています。TCP では、アプリケーション プログラム間の着信トラフィックのすべての逆多重化を処理するため、TCP を使用すると、1 つのシステム上の複数のアプリケーションが同時に通信できます。

UDP の概要

ユーザ データグラム プロトコル（UDP）は、IP ファミリに属するコネクションレス型トランスポートレイヤ プロトコルです。UDP は、ネットワーク ファイル システム（NFS）、簡易ネットワーク管理プロトコル（SNMP）、ドメイン ネーム システム（DNS）、TFTP などの一般的なアプリケーション層プロトコルのための、トランスポート プロトコルです。

TCP および UDP 以外のすべての IP プロトコルは、RAW プロトコルと考えられています。

ほとんどのサイトでは、TCP、UDP、および RAW トランスポートのデフォルト設定を変更する必要はありません。