



プロトコル独立機能

- [プロトコル独立機能 \(1 ページ\)](#)

プロトコル独立機能

分散型シスコ エクスプレス フォワーディング

シスコ エクスプレス フォワーディングに関する情報

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

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

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

ング情報を付加します。隣接テーブルには、すべての FIB エントリに対する、レイヤ 2 のネクストホップのアドレスが保持されます。

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

シスコ エクスプレス フォワーディングの設定方法

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

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 2	ip cef 例 : Device(config)# ip cef	非スタッキングスイッチで CEF の動作をイネーブルにします。 ステップ 4 に進みます。
ステップ 3	ip cef distributed 例 : Device(config)# ip cef distributed	アクティブスイッチで CEF の動作をイネーブルにします。
ステップ 4	interface interface-id 例 : Device(config)# interface gigabitethernet 1/0/1	インターフェイス コンフィギュレーションモードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 5	ip route-cache cef 例 : Device(config-if)# ip route-cache cef	ソフトウェア転送トラフィック用のインターフェイスで CEF をイネーブルにします。 (注) ip route-cache cef コマンドはデフォルトで有効になっており、無効にはできません。

	コマンドまたはアクション	目的
ステップ 6	end 例 : Device (config-if) # end	特権 EXEC モードに戻ります。
ステップ 7	show ip cef 例 : Device# show ip cef	すべてのインターフェイスの CEF ステータスを表示します。
ステップ 8	show cef linecard [detail] 例 : Device# show cef linecard detail	(任意) 非スタッキング スイッチの CEF 関連インターフェイス情報を表示します。
ステップ 9	show cef linecard [slot-number] [detail] 例 : Device# show cef linecard 5 detail	(任意) スタック内のすべてのスイッチ、または指定されたスイッチに対して、スイッチの CEF 関連インターフェイス情報をスタックメンバ別に表示します。 (任意) <i>slot-number</i> には、スタックメンバーのスイッチ番号を入力します。
ステップ 10	show cef interface [interface-id] 例 : Device# show cef interface gigabitethernet 1/0/1	すべてのインターフェイスまたは指定されたインターフェイスの詳細な CEF 情報を表示します。
ステップ 11	show adjacency 例 : Device# show adjacency	CEF の隣接テーブル情報を表示します。
ステップ 12	copy running-config startup-config 例 : Device# copy running-config startup-config	(任意) コンフィギュレーションファイルに設定を保存します。

CEF トラフィック用のロードバランシングスキーム

CEF トラフィック用のロードバランシングスキームの設定に関する制約事項

- デバイスまたはデバイススタックメンバのロードバランシングを同じように、グローバルに設定する必要があります。
- CEF トラフィックのパケットごとのロードバランシングはサポートされていません。

CEF ロードバランシングの概要

CEF のロードバランシングを行うと、トラフィックを複数のパスに分散することにより、リソースを最適化することができます。CEF のロードバランシングは、送信元と宛先のパケット情報の組み合わせに基づいて動作します。

ロードバランシングは宛先単位で設定できます。ロードバランシングの判断はアウトバウンドインターフェイス上で行われるため、ロードバランシングは、アウトバウンドインターフェイスで設定する必要があります。

CEF トラフィックに対する宛先別ロードバランシング

宛先単位のロードバランシングにより、デバイスは、複数のパスを使用して、複数の発信元と宛先ホストのペアにわたって負荷を共有することができます。指定された発信元と宛先ホストのペアは、複数のパスを使用可能な場合であっても、同じパスを使用することが保証されています。異なるペアを宛先とするトラフィックストリームは、異なるパスを使用します。

CEF がイネーブルの場合、宛先別ロードバランシングはデフォルトでイネーブルです。CEF をイネーブルにした場合、宛先単位のロードバランシングを使用するための追加タスクはありません。多くの状況では、ロードバランシングの方法として宛先単位を使用します。

宛先単位のロードバランシングはトラフィックの統計的な分散に依存しているため、発信元と宛先ホストのペア数が増大すると、ロードシェアリングがさらに有効になります。

宛先単位のロードバランシングを使用することにより、個々のホストペアのパケットが順に到達することが保証されます。特定のホストペアに宛てられたすべてのパケットは、（複数の場合も）同じリンクを介して転送されます。

CEF トラフィックに対するロードバランシングアルゴリズム

CEF トラフィックで使用するために、次のロードバランシングアルゴリズムが用意されています。ロードバランシングアルゴリズムは、`ip cef load-sharing algorithm` コマンドで選択します。

- オリジナルアルゴリズム：オリジナルのロードバランシングアルゴリズムでは、すべてのデバイスで同じアルゴリズムが使用されるため、複数のデバイスにわたるロードシェアリングで歪みが発生します。ネットワーク環境に応じて、アルゴリズムを選択する必要があります。

- ユニバーサルアルゴリズム：ユニバーサル ロード バランシング アルゴリズムでは、ネットワーク上の各デバイスは、発信元と宛先の各アドレスペアに対して異なるロードシェアリングの判断を行うことができます。これにより、ロードシェアリングの不均衡が解決されます。デバイスは、デフォルトではユニバーサル ロード シェアリングを実行するように設定されています。

CEF トラフィックに対するロードバランシングの設定方法

ここでは、CEF トラフィックに対するロードバランシングの設定について説明します。

CEF の宛先別ロードバランシングの有効化または無効化

CEF の宛先単位のロードバランシングを有効または無効にするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device# enable	グローバル コンフィギュレーション モードを開始します。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	interface interface-id 例： Device(config-if)# interface gigabitethernet 1/0/1	インターフェイスコンフィギュレーション モードを開始し、設定するレイヤ 3 インターフェイスを指定します。
ステップ 4	[no] ip load-sharing per-destination 例： Device(config-if)# ip load-sharing per-destination	インターフェイスで CEF の宛先別ロードバランシングを有効にします。 no ip load-sharing per-destination コマンドを使用すると、インターフェイスで CEF の宛先別ロードバランシングが無効になります。
ステップ 5	end 例： Device(config-if)# end	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、特権 EXEC モードに戻ります。

CEF トラフィックに対するトンネル ロードバランシング アルゴリズムの選択

ネットワーク環境に少数の発信元と宛先のペアしか存在しない場合には、トンネルアルゴリズムを選択します。デバイスは、デフォルトではユニバーサルロードシェアリングを実行するように設定されています。

CEF トラフィック用にトンネルロードバランシングアルゴリズムを選択するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例 : Device# enable	グローバル コンフィギュレーション モードを開始します。
ステップ 2	configure terminal 例 : Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	ip cef load-sharing algorithm {original universal [id]} 例 : Device(config)# ip cef load-sharing algorithm universal	CEF のロードバランシング アルゴリズムを選択します。 <ul style="list-style-type: none"> • original キーワードは、送信元 IP と宛先 IP のハッシュに基づいて、ロードバランシングアルゴリズムとしてオリジナルアルゴリズムを設定します。 • universal キーワードは、送信元 IP、宛先 IP、レイヤ 3 プロトコル、レイヤ 4 送信元ポート、レイヤ 4 宛先ポート、および IPv6 トラフィックラベル (IPv6 トラフィック用) を使用するロードバランシングアルゴリズムを設定します。 • <i>id</i> 引数は、固定 ID です。
ステップ 4	end 例 : Device(config)# end	特権 EXEC モードに戻ります。

CEF トラフィックのロードバランシングの設定例

ここでは、CEF トラフィックのロードバランシングの設定例を示します。

例：CEF の宛先別ロードバランシングの有効化または無効化

CEF がイネーブルの場合、宛先別ロードバランシングはデフォルトでイネーブルです。次の例は、宛先単位のロードバランシングをディセーブルにする方法を示しています。

```
Device> enable
Device# configure terminal
Device(config)# interface Ethernet1/0/1
Device(config-if)# no ip load-sharing per-destination
Device(config-if)# end
```

等コスト ルーティング パスの個数

等コスト ルーティング パスに関する情報

同じネットワークへ向かう同じメトリックのルートが複数ルータに格納されている場合、これらのルートは等価コストを保有していると思なされます。ルーティングテーブルに複数の等コストルートが含まれる場合は、これらをパラレルパスと呼ぶこともあります。ネットワークへの等コストパスがルータに複数格納されている場合、ルータはこれらを同時に使用できます。パラレルパスを使用すると、パスに障害が発生した場合に冗長性を確保できます。また、使用可能なパスにパケットの負荷を分散し、使用可能な帯域幅を有効利用することもできます。等コストルートは、スタック内の各スイッチでサポートされます。

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

等コスト ルーティング パスの設定方法

手順

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

	コマンドまたはアクション	目的
	Device# configure terminal	
ステップ 3	router {rip ospf eigrp} 例： Device(config)# router eigrp	ルータ コンフィギュレーション モードを開始します。
ステップ 4	maximum-paths maximum 例： Device(config-router)# maximum-paths 2	プロトコルルーティング テーブルの平行パスの最大数を設定します。指定できる範囲は 1～16 です。ほとんどの IP ルーティング プロトコルでデフォルトは 4 ですが、BGP の場合だけ 1 です。
ステップ 5	end 例： Device(config-router)# end	特権 EXEC モードに戻ります。
ステップ 6	show ip protocols 例： Device# show ip protocols	<i>Maximum path</i> フィールドの設定を確認します。
ステップ 7	copy running-config startup-config 例： Device# copy running-config startup-config	(任意) コンフィギュレーション ファイルに設定を保存します。

スタティックユニキャストルート

スタティックユニキャストルートに関する情報

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

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

ルートのアドミニストレーティブディスタンスがダイナミックプロトコルのアドミニストレーティブディスタンスよりも大きな値になるように設定します。

表 1: ダイナミックルーティングプロトコルのデフォルトのアドミニストレーティブディスタンス

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

インターフェイスを指し示すスタティックルートは、RIP、IGRP、およびその他のダイナミックルーティングプロトコルを通してアドバタイズされます。**redistribute** スタティックルータコンフィギュレーションコマンドが、これらのルーティングプロトコルに対して指定されているかどうかは関係ありません。これらのスタティックルートがアドバタイズされるのは、インターフェイスを指し示すスタティックルートが接続された結果、静的な性質を失ったとルーティングテーブルで見なされるためです。ただし、**network** コマンドで定義されたネットワーク以外のインターフェイスに対してスタティックルートを定義する場合は、ダイナミックルーティングプロトコルに **redistribute** スタティックコマンドを指定しない限り、ルートはアドバタイズされません。

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

スタティックユニキャストルートの設定

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

スタティックルートを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	ip route prefix mask {address interface} [distance] 例： Device(config)# ip route prefix mask gigabitethernet 1/0/4	スタティック ルートを確立します。
ステップ 4	end 例： Device(config)# end	特権 EXEC モードに戻ります。
ステップ 5	show ip route 例： Device# show ip route	設定を確認するため、ルーティングテーブルの現在の状態を表示します。
ステップ 6	copy running-config startup-config 例： Device# copy running-config startup-config	（任意）コンフィギュレーション ファイルに設定を保存します。

次のタスク

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

デフォルトのルートおよびネットワーク

デフォルトのルートおよびネットワークに関する情報

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

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

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

ダイナミックルーティングプロトコルによってデフォルト情報を送信するときは、特に設定する必要はありません。ルーティングテーブルは定期的にスキャンされ、デフォルトルートとして最適なデフォルトネットワークが選択されます。IGRP ネットワークでは、システムのデフォルトネットワークの候補が複数存在する場合があります。Cisco ルータでは、デフォルトルートまたは最終ゲートウェイを設定するため、アドミニストレーティブディスタンスおよびメトリック情報を使用します。

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

デフォルトのルートおよびネットワークの設定方法

デフォルトルートおよびネットワークを設定するには、次の手順を実行します。

手順

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

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

ルーティング情報を再配信するためのルートマップ

ルートマップの概要

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

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

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



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

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

ルートマップの設定方法

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



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

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 2	route-map <i>map-tag</i> [permit deny] [<i>sequence number</i>] 例： Device (config)# route-map rip-to-ospf permit 4	再配信を制御するために使用するルートマップを定義し、ルートマップ コンフィギュレーションモードを開始します。 <i>map-tag</i> : ルートマップ用のわかりやすい名前を指定します。 redistribute ルータ コンフィギュレーション コマンドはこの名前を使用して、このルートマップを参照します。複数のルートマップで同じマップタグ名を共有できます。 (任意) permit が指定され、このルートマップの一致条件が満たされている場合は、 set アクションの制御に従ってルートが再配信されます。 deny が指定されている場合、ルートは再配信されません。

	コマンドまたはアクション	目的
		<i>sequence number</i> (任意) : 同じ名前によってすでに設定されているルートマップのリスト内で、新しいルートマップの位置を指定する番号です。
ステップ 3	match as-path <i>path-list-number</i> 例 : Device(config-route-map)# match as-path 10	BGP AS パス アクセスリストと照合します。
ステップ 4	match community-list <i>community-list-number</i> [exact] 例 : Device(config-route-map)# match community-list 150	BGP コミュニティリストのマッチングを行います。
ステップ 5	match ip address { <i>access-list-number</i> <i>access-list-name</i> } [... <i>access-list-number</i> ... <i>access-list-name</i>] 例 : Device(config-route-map)# match ip address 5 80	名前または番号を指定し、標準アクセスリストと照合します。1 ~ 199 の整数を指定できます。
ステップ 6	match metric <i>metric-value</i> 例 : Device(config-route-map)# match metric 2000	指定されたルートメトリックと一致させます。 <i>metric-value</i> には、0 ~ 4294967295 の値が指定された、EIGRP のメトリックを指定できます。
ステップ 7	match ip next-hop { <i>access-list-number</i> <i>access-list-name</i> } [... <i>access-list-number</i> ... <i>access-list-name</i>] 例 : Device(config-route-map)# match ip next-hop 8 45	指定されたアクセスリスト (番号 1 ~ 199) のいずれかで送信される、ネクストホップのルータアドレスと一致させます。
ステップ 8	match tag <i>tag value</i> [... <i>tag-value</i>] 例 : Device(config-route-map)# match tag 3500	1 つまたは複数のルートタグ値からなるリスト内の指定されたタグ値と一致させます。0 ~ 4294967295 の整数を指定できます。

	コマンドまたはアクション	目的
ステップ 9	match interface <i>type number</i> [... <i>type-number</i>] 例 : Device (config-route-map) # match interface gigabitethernet 1/0/1	指定されたインターフェイスの1つから、指定されたネクストホップへのルートと一致させます。
ステップ 10	match ip route-source { <i>access-list-number</i> <i>access-list-name</i> } [... <i>access-list-number</i> ... <i>access-list-name</i>] 例 : Device (config-route-map) # match ip route-source 10 30	アドバタイズされた指定のアクセスリストによって指定したアドレスに一致します。
ステップ 11	match route-type { <i>local</i> <i>internal</i> <i>external</i> [<i>type-1</i> <i>type-2</i>]} 例 : Device (config-route-map) # match route-type local	指定された route-type と一致させます。 <ul style="list-style-type: none"> • local : ローカルに生成された BGP ルート。 • internal : OSPF エリア内およびエリア間ルート、または EIGRP 内部ルート。 • external : OSPF 外部ルート (タイプ 1 またはタイプ 2) または EIGRP 外部ルート。
ステップ 12	set dampening <i>half-life reuse suppress max-suppress-time</i> 例 : Device (config-route-map) # set dampening 30 1500 10000 120	BGP ルート ダンプニング係数を設定します。
ステップ 13	set local-preference <i>value</i> 例 : Device (config-route-map) # set local-preference 100	ローカル BGP パスに値を割り当てます。
ステップ 14	set origin { <i>igp</i> <i>egp as</i> <i>incomplete</i> } 例 : Device (config-route-map) # set origin igp	BGP 送信元コードを設定します。

	コマンドまたはアクション	目的
ステップ 15	set as-path {tag prepend <i>as-path-string</i>} 例 : Device(config-route-map)# set as-path tag	BGP の自律システム パスを変更します。
ステップ 16	set level {level-1 level-2 level-1-2 stub-area backbone} 例 : Device(config-route-map)# set level level-1-2	ルーティングドメインの指定エリアにアダプタイズされるルートレベルを設定します。 stub-area および backbone は、OSPF NSSA およびバックボーンエリアです。
ステップ 17	set metric <i>metric value</i> 例 : Device(config-route-map)# set metric 100	再配布されるルート指定のためのメトリック値を設定します (EIGRP のみ)。 <i>metric value</i> は -294967295 ~ 294967295 の整数です。
ステップ 18	set metric bandwidth delay reliability loading mtu 例 : Device(config-route-map)# set metric 10000 10 255 1 1500	再配布されるルート指定のためのメトリック値を設定します (EIGRP のみ)。 <ul style="list-style-type: none"> • <i>bandwidth</i> : 0 ~ 4294967295 の範囲のルートのメトリック値または IGRP 帯域幅 (キロビット/秒単位)。 • <i>delay</i> : 0 ~ 4294967295 の範囲のルート遅延 (10 マイクロ秒単位)。 • <i>reliability</i> : 0 ~ 255 の数値で表されるパケット伝送の成功可能性。255 は信頼性が 100% であること、0 は信頼性がないことを意味します。 • <i>loading</i> : 0 ~ 255 の数値で表されるルートの有効帯域幅 (255 は 100% の負荷)。 • <i>mtu</i> : ルートの MTU の最小サイズ (バイト単位)。範囲は 0 ~ 4294967295 です。

	コマンドまたはアクション	目的
ステップ 19	set metric-type {type-1 type-2} 例： Device(config-route-map)# set metric-type type-2	再配信されるルートに OSPF 外部メトリック タイプを設定します。
ステップ 20	set metric-type internal 例： Device(config-route-map)# set metric-type internal	ネクストホップの IGP メトリックと一致するように、EBGP ネイバーにアドバタイズされるプレフィックスの Multi-Exit 識別子 (MED) 値を設定します。
ステップ 21	set weight number 例： Device(config-route-map)# set weight 100	ルーティング テーブルの BGP 重みを設定します。指定できる値は 1 ~ 65535 です。
ステップ 22	end 例： Device(config-route-map)# end	特権 EXEC モードに戻ります。
ステップ 23	show route-map 例： Device# show route-map	設定を確認するため、設定されたすべてのルートマップ、または指定されたルートマップだけを表示します。
ステップ 24	copy running-config startup-config 例： Device# copy running-config startup-config	(任意) コンフィギュレーションファイルに設定を保存します。

ルート配信の制御方法

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



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

ルーティング プロトコルのメトリックを、必ずしも別のルーティング プロトコルのメトリックに変換する必要はありません。たとえば、RIP メトリックはホップカウントで、IGRP メ

リックは5つの特性の組み合わせです。このような場合は、メトリックを独自に設定し、再配信されたルートに割り当てます。ルーティング情報を制御せずにさまざまなルーティングプロトコル間で交換するとルーティングループが発生し、ネットワーク動作が著しく低下することがあります。

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーションモードを開始します。
ステップ 2	router {rip ospf eigrp} 例： Device(config)# router eigrp 10	ルータ コンフィギュレーションモードを開始します。
ステップ 3	redistribute protocol [process-id] {level-1 level-1-2 level-2} [metric metric-value] [metric-type type-value] [match internal external type-value] [tag tag-value] [route-map map-tag] [weight weight] [subnets] 例： Device(config-router)# redistribute eigrp 1	ルーティングプロトコル間でルートを再配信します。route-map を指定しないと、すべてのルートが再配信されます。キーワード route-map に <i>map-tag</i> を指定しないと、ルートは配信されません。
ステップ 4	default-metric number 例： Device(config-router)# default-metric 1024	現在のルーティングプロトコルが、再配信されたすべてのルートに対して同じメトリック値を使用するように設定します（RIP と OSPF）。
ステップ 5	default-metric bandwidth delay reliability loading mtu 例：	EIGRP ルーティングプロトコルが、EIGRP 以外で再配信されたすべてのルートに対して同じメトリック値を使用するように設定します。

	コマンドまたはアクション	目的
	Device(config-router)# default-metric 1000 100 250 100 1500	
ステップ6	end 例： Device(config-router)# end	特権 EXEC モードに戻ります。
ステップ7	show route-map 例： Device# show route-map	設定を確認するため、設定されたすべてのルートマップ、または指定されたルートマップだけを表示します。
ステップ8	copy running-config startup-config 例： Device# copy running-config startup-config	(任意) コンフィギュレーションファイルに設定を保存します。

ポリシーベースルーティング

PBR の設定に関する制約事項

- ポリシーベースルーティング (PBR) は、トラフィックの GRE トンネルへの転送ではサポートされません。これは、任意のインターフェイスに適用される PBR と、トラフィックの GRE トンネルへの転送 (PBR ネクストホップもしくはデフォルトネクストホップまたは設定済みのインターフェイスを使用) に適用される PBR に適用されます。
- PBR は、GRE トンネル自体ではサポートされていません (GRE トンネル自体のもので適用されます)。

ポリシーベースルーティングの概要

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

- 特定のエンドシステムの ID
- アプリケーション
- プロトコル

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

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

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

次に例を示します。

```
match length A B
match ip address acl1 acl2
match ip address acl3
```

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

- 決定が許可の場合は、`set` コマンドで指定されたアクションがパケットで適用されます。
- 下された決定が拒否の場合は、PBR アクション (`set` コマンドで指定された) が適用されません。代わりに、処理ロジックが、シーケンス内の次のルートマップ文 (シーケンス番号が次に高い文) に移動します。次の文が存在しない場合は、PBR 処理が終了し、パケットがデフォルトの IP ルーティング テーブルを使用してルーティングされます。
- PBR では、拒否としてマークされているルートマップ ステートメントはサポートされません。

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

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

PBR の設定方法

- PBR を使用するには、スタンドアロンスイッチまたはアクティブスイッチ上で Network Essentials ライセンスをイネーブルにしておく必要があります。
- マルチキャスト トラフィックには、ポリシーによるルーティングが行われません。PBR が適用されるのはユニキャスト トラフィックだけです。
- ルーテッド ポートまたは SVI 上で、PBR を有効にできます。
- スイッチは一致長に基づき PBR をサポートします。
- レイヤ 3 モードの EtherChannel ポート チャンネルにはポリシー ルート マップを適用できませんが、EtherChannel のメンバーである物理インターフェイスには適用できません。適用しようとする、コマンドが拒否されます。ポリシー ルート マップが適用されている物理インターフェイスは、EtherChannel のメンバーになることができません。
- スイッチまたはスイッチ スタックには最大 128 個の IP ポリシー ルート マップを定義できます。
- スイッチまたはスイッチ スタックには、PBR 用として最大 512 個のアクセス コントロール エントリ (ACE) を定義できます。
- ルート マップに一致基準を設定する場合は、次の注意事項に従ってください。
 - ローカル アドレス宛ての packets を許可する ACL と照合させないでください。
- WCCP と PBR は、スイッチ インターフェイスで相互に排他的です。PBR がインターフェイスで有効になっているときは、WCCP を有効にできません。その反対の場合も同じで、WCCP がインターフェイスで有効になっているときは、PBR を有効にできません。
- PBR で使用されるハードウェア エントリ数は、ルート マップ自体、使用される ACL、ACL およびルート マップ エントリの順序によって異なります。
- TOS、DSCP、および IP Precedence に基づく PBR はサポートされません。
- set interface、set default next-hop、および set default interface はサポートされません。
- **ip next-hop recursive** および **ip next-hop verify availability** 機能は使用できません。next-hop は、直接接続される必要があります。
- set アクションのないポリシー マップはサポートされます。一致パケットは通常どおりにルーティングされます。
- match 句のないポリシー マップはサポートされます。set アクションはすべてのパケットに適用されます。

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

スイッチ（CPU）で生成されたパケットまたはローカルパケットは、通常どおりにポリシールーティングされません。スイッチ上でローカル PBR をグローバルに有効にすると、そのスイッチから送信されたすべてのユニキャストパケットがローカル PBR の影響を受けます。ローカル PBR に関してサポートされているプロトコルは、NTP、DNS、MSDP、SYSLOG、および TFTP です。ローカル PBR は、デフォルトで無効に設定されています。

手順

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <pre>Device> enable</pre>	<p>特権 EXEC モードを有効にします。</p> <p>パスワードを入力します（要求された場合）。</p>
ステップ 2	<p>configure terminal</p> <p>例 :</p> <pre>Device# configure terminal</pre>	<p>グローバル コンフィギュレーション モードを開始します。</p>
ステップ 3	<p>route-map map-tag [permit] [sequence number]</p> <p>例 :</p> <pre>Device(config)# route-map pbr-map permit</pre>	<p>パケットの出力場所を制御するために使用するルートマップを定義し、ルートマップのコンフィギュレーションモードを開始します。</p> <ul style="list-style-type: none"> • <i>map-tag</i> - : ルートマップ用のわかりやすい名前を指定します。 ip policy route-map インターフェイスコンフィギュレーションコマンドは、この名前を使用して、このルートマップを参照します。同じ <i>map-tag</i> がある複数の <i>route-map</i> 文は、1 つの <i>route-map</i> を定義します。 • (任意) permit - : permit が指定され、このルートマップの一致条件が満たされている場合は、set アクションの制御に従ってルートがポリシールーティングされます。 • (任意) <i>sequence number</i> - : シーケンス番号は、特定のルートマップ内の <i>route-map</i> ステートメントの位置を示します。

	コマンドまたはアクション	目的
ステップ 4	<p>match ip address {<i>access-list-number</i> <i>access-list-name</i>} [<i>access-list-number</i> ...<i>access-list-name</i>]</p> <p>例 :</p> <pre>Device(config-route-map)# match ip address 110 140</pre>	<p>1 つ以上の標準または拡張アクセスリストで許可されている送信元および宛先 IP アドレスを照合します。ACL は、複数の送信元および宛先 IP アドレスでも照合できます。</p> <p>match コマンドを指定しない場合、ルートマップはすべてのパケットに適用されます。</p>
ステップ 5	<p>match length min max</p> <p>例 :</p> <pre>Device(config-route-map)# match length 64 1500</pre>	<p>パケット長と照合します。</p>
ステップ 6	<p>set ip next-hop ip-address [...<i>ip-address</i>]</p> <p>例 :</p> <pre>Device(config-route-map)# set ip next-hop 10.1.6.2</pre>	<p>基準と一致するパケットの動作を指定します。パケットのルーティング先となるネクストホップを設定します（ネクストホップは隣接している必要があります）。</p>
ステップ 7	<p>exit</p> <p>例 :</p> <pre>Device(config-route-map)# exit</pre>	<p>グローバル コンフィギュレーションモードに戻ります。</p>
ステップ 8	<p>interface interface-id</p> <p>例 :</p> <pre>Device(config)# interface gigabitethernet 1/0/1</pre>	<p>インターフェイス コンフィギュレーションモードを開始し、設定するインターフェイスを指定します。</p>
ステップ 9	<p>ip policy route-map map-tag</p> <p>例 :</p> <pre>Device(config-if)# ip policy route-map pbr-map</pre>	<p>レイヤ 3 インターフェイス上で PBR を有効にし、使用するルートマップを識別します。1 つのインターフェイスに設定できるルートマップは、1 つだけです。ただし、異なるシーケンス番号を持つ複数のルートマップエントリを設定できます。これらのエントリは、最初の一致が見つかるまで、シーケンス番号順に評価されます。一致が見つからない場合、パケットは通常どおりにルーティングされます。</p>
ステップ 10	<p>ip route-cache policy</p> <p>例 :</p>	<p>(任意) PBR の高速スイッチングを有効にします。PBR の高速スイッチング</p>

	コマンドまたはアクション	目的
	Device(config-if)# ip route-cache policy	を有効にするには、PBR を有効にする必要があります。
ステップ 11	exit 例： Device(config-if)# exit	グローバル コンフィギュレーション モードに戻ります。
ステップ 12	ip local policy route-map map-tag 例： Device(config)# ip local policy route-map local-pbr	(任意) ローカル PBR を有効にして、スイッチから送信されるパケットに PBR を実行します。ローカル PBR は、スイッチによって生成されるパケットに適用されます。着信パケットには適用されません。
ステップ 13	end 例： Device(config)# end	特権 EXEC モードに戻ります。
ステップ 14	show route-map [map-name] 例： Device# show route-map	(任意) 設定を確認するため、設定されたすべてのルートマップ、または指定されたルートマップだけを表示します。
ステップ 15	show ip policy 例： Device# show ip policy	(任意) インターフェイスに付加されたポリシー ルート マップを表示します。
ステップ 16	show ip local policy 例： Device# show ip local policy	(任意) ローカル PBR が有効であるかどうか、および有効である場合は使用されているルート マップを表示します。

ルーティング情報のフィルタリング

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



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

受動インターフェイスの設定

ローカルネットワーク上の他のルータが動的にルートを取得しないようにするには、**passive-interface** ルータ コンフィギュレーション コマンドを使用し、ルーティング アップデー

トメッセージがルータインターフェイスから送信されないようにします。OSPFプロトコルでこのコマンドを使用すると、パッシブに指定したインターフェイスアドレスがOSPFドメインのスタブネットワークとして表示されます。OSPFルーティング情報は、指定されたルータインターフェイスから送受信されません。

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 2	router {rip ospf eigrp} 例： Device(config)# router ospf	ルータ コンフィギュレーション モードを開始します。
ステップ 3	passive-interface interface-id 例： Device(config-router)# passive-interface gigabitethernet 1/0/1	指定されたレイヤ3インターフェイス経由のルーティング アップデートの送信を抑制します。
ステップ 4	passive-interface default 例： Device(config-router)# passive-interface default	(任意) すべてのインターフェイスを、デフォルトでパッシブとなるように設定します。
ステップ 5	no passive-interface interface type 例： Device(config-router)# no passive-interface gigabitethernet1/0/3 gigabitethernet 1/0/5	(任意) 隣接関係を送信する必要があるインターフェイスだけをアクティブにします。

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

ルーティングアップデートのアドバタイズおよび処理の制御

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 パスワードを入力します (要求された場合)。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	router {rip eigrp} 例： Device (config) # router eigrp 10	ルータ コンフィギュレーション モードを開始します。

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

ルーティング情報の送信元のフィルタリング

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	router {rip ospf eigrp} 例： Device(config)# router eigrp 10	ルータ コンフィギュレーション モードを開始します。
ステップ 4	distance weight {ip-address {ip-address mask}} [ip access list] 例： Device(config-router)# distance 50 10.1.5.1	アドミニストレーティブ ディスタンスを定義します。 <i>weight</i> : アドミニストレーティブ ディスタンスは 10 ~ 255 の整数です。単独で使用した場合、 <i>weight</i> はデフォルトのアドミニストレーティブ ディスタンスを指定します。ルーティング情報の送信元に他の指定がない場合に使用されます。アドミニストレーティブ ディスタンスが 255 のルートはルーティングテーブルに格納されません。 (任意) <i>ip access list</i> : 着信ルーティングアップデートに適用される IP 標準または IP 拡張アクセス リストです。
ステップ 5	end 例： Device(config-router)# end	特権 EXEC モードに戻ります。
ステップ 6	show ip protocols 例： Device# show ip protocols	指定されたルーティング プロセス用のデフォルトのアドミニストレーティブ ディスタンスを表示します。
ステップ 7	copy running-config startup-config 例：	(任意) コンフィギュレーション ファイルに設定を保存します。

	コマンドまたはアクション	目的
	Device# copy running-config startup-config	

認証キーの管理

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

前提条件

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

認証キーの設定方法

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

手順

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

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

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。