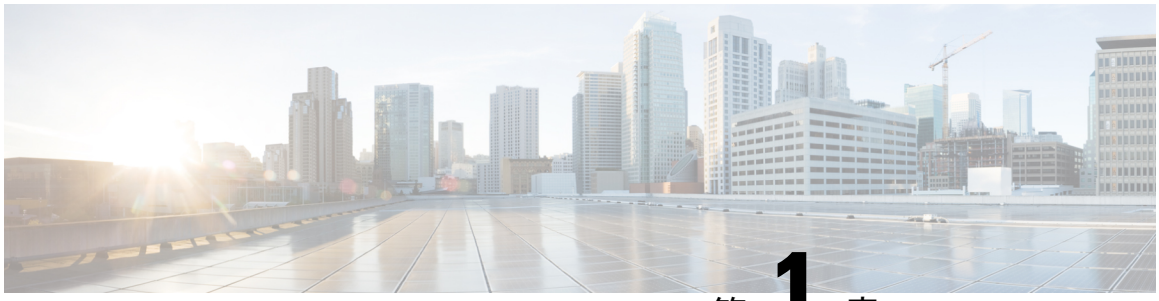
Cisco IOS XE Everest 16.6.x（Catalyst 9500 スイッチ）ネットワーク管理コンフィギュレーション ガイド

初版：2017 年 7 月 31 日

最終更新：2017 年 11 月 3 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

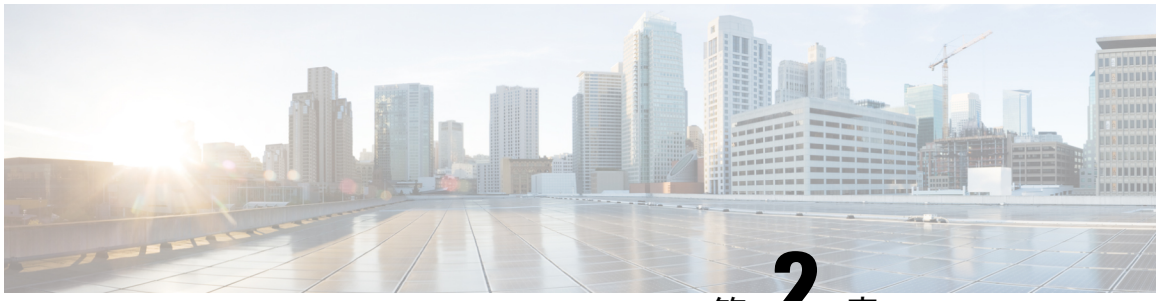
Cisco プラグ アンド プレイの設定

Cisco プラグ アンド プレイの設定

プラグ アンド プレイの設定方法については、次を参照してください。

- Cisco プラグ アンド プレイ機能ガイド
- Configuration Guide for Cisco Network Plug and Play on APIC-EM

Cisco プラグ アンド プレイの設定



第 2 章

Cisco Discovery Protocol の設定

CDP に関する情報

CDP の概要

CDP はすべてのシスコ デバイス（ルータ、ブリッジ、アクセス サーバ、コントローラ、およびスイッチ）のレイヤ 2（データリンク層）で動作するデバイス検出プロトコルです。ネットワーク管理アプリケーションは CDP を使用することにより、既知のデバイスにネイバー シスコ デバイスを検出できます。また、下位レイヤのトランスペアレント プロトコルが稼働しているネイバーデバイスのデバイス タイプや、簡易ネットワーク管理プロトコル（SNMP）エージェント アドレスを学習することもできます。この機能によって、アプリケーションからネイバー デバイスに SNMP クエリーを送信できます。

CDP は、サブネットワーク アクセス プロトコル（SNAP）をサポートしているすべてのメディアで動作します。CDP はデータリンク層でのみ動作するため、異なるネットワーク層プロトコルをサポートする 2 つのシステムで互いの情報を学習できます。

CDP が設定された各デバイスはマルチキャスト アドレスに定期的にメッセージを送信して、SNMP メッセージを受信可能なアドレスを 1 つまたは複数アドバタイズします。このアドバタイズには、受信側デバイスで CDP 情報を廃棄せずに保持する時間を表す存続可能時間、つまりホールドタイム情報も含まれます。各デバイスは他のデバイスから送信されたメッセージも待ち受けて、ネイバー デバイスについて学習します。

CDP はデバイス上で Network Assistant をイネーブルにすることで、ネットワークをグラフィカルに表示できます。デバイスは CDP を使用してクラスタ候補を検出し、クラスタ メンバ、およびコマンドデバイスから最大 3 台（デフォルト）離れたクラスタ対応の他のデバイスについての情報を維持します。

CDP のデフォルト設定

この表は、CDP のデフォルト設定を示します。

機能	デフォルト設定
CDP グローバル ステート	[有効（Enabled）]
CDP インターフェイス ステート	[有効（Enabled）]
CDP タイマー（パケット更新頻度）	60 秒
CDP ホールドタイム（廃棄までの時間）	180 秒
CDP バージョン 2 アドバタイズ	[有効（Enabled）]

CDP の設定方法

CDP 特性の設定

次の CDP 特性を設定できます。

- CDP 更新の頻度
- 破棄するまで情報を保持する時間の長さ
- バージョン 2 アドバタイズを送信するかどうか

（注） ステップ 3 ～ 5 はすべて任意であり、どの順番で実行してもかまいません。

CDP 特性を設定するには、次の手順を実行します。

手順

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

	コマンドまたはアクション	目的
ステップ 3	**cdp timer** 秒 例： `Device(config)# cdp timer 20`	（任意）CDP 更新の送信頻度を秒単位で設定します。 指定できる範囲は 5 ～ 254 です。デフォルトは 60 秒です。
ステップ 4	**cdp holdtime** 秒 例： `Device(config)# cdp holdtime 60`	（任意）受信デバイスがこのデバイスから送信された情報を破棄せずに保持する時間を指定します。 設定範囲は 10 ～ 255 秒で、デフォルトは 180 秒に設定されています。
ステップ 5	**cdp advertise-v2** 例： `Device(config)# cdp advertise-v2`	（任意）バージョン 2 アドバタイズを送信するように CDP を設定します。 これは、デフォルトの状態です。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

次のタスク

デフォルト設定に戻すには、CDP コマンドの **no** 形式を使用します。

CDP のディセーブル化

CDP はデフォルトで有効になっています。

（注） デバイス クラスタと他のシスコ デバイス（Cisco IP Phone など）は、CDP メッセージを定期的に交換します。CDP をディセーブルにすると、クラスタ検出が中断され、デバイスの接続が切断されます。

CDP デバイス検出機能をディセーブルにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**no cdp run** 例： `Device(config)# no cdp run`	CDP をディセーブルにします。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

次のタスク

CDP を使用するには、再びイネーブルにする必要があります。

Enabling CDP

CDP はデフォルトで有効になっています。

（注） デバイス クラスタと他のシスコ デバイス（Cisco IP Phone など）は、CDP メッセージを定期的に交換します。CDP をディセーブルにすると、クラスタ検出が中断され、デバイスの接続が切断されます。

ディセーブルになっている CDP をイネーブルにするには、次の手順を実行します。

始める前に

CDP をディセーブルにする必要があります。そのようにしないとイネーブルにできません。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**cdp run** 例： `Device(config)# cdp run`	ディセーブルになっている場合は、CDP をイネーブルにします。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例：	入力を確認します。

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

CDP が有効になっていることを示すには、**showrunall** コマンドを使用します。**showrun** だけを入力した場合、CDP の有効化が表示されないことがあります。

インターフェイス上での CDP のディセーブル化

CDP 情報を送受信するために、サポートされているすべてのインターフェイス上では CDP がデフォルトでイネーブルになっています。

（注） デバイス クラスタと他のシスコ デバイス（Cisco IP Phone など）は、CDP メッセージを定期的に交換します。CDP をディセーブルにすると、クラスタ検出が中断され、デバイスの接続が切断されます。

ポート上で CDP をディセーブルにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例：	CDP をディセーブルにするインターフェイスを指定し、インターフェイス コン

	コマンドまたはアクション	目的
	`Device(config)# interface gigabitethernet1/0/1`	フィギュレーション モードを開始します。
ステップ 4	**no cdp enable** 例： `Device(config-if)# no cdp enable`	ステップ 3 で指定したインターフェイスで CDP をディセーブルにします。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device# show running-config`	入力を確認します。
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。

インターフェイス上での CDP のイネーブル化

CDP 情報を送受信するために、サポートされているすべてのインターフェイス上では CDP がデフォルトでイネーブルになっています。

（注） デバイス クラスタと他のシスコ デバイス（Cisco IP Phone など）は、CDP メッセージを定期的に交換します。CDP をディセーブルにすると、クラスタ検出が中断され、デバイスの接続が切断されます。

ポート上でディセーブルになっている CDP をイネーブルにするには、次の手順を実行します。

始める前に

CDP をイネーブルにしようとしているポートで、CDP がディセーブルになっている必要があります。そうでない場合は、イネーブルにできません。

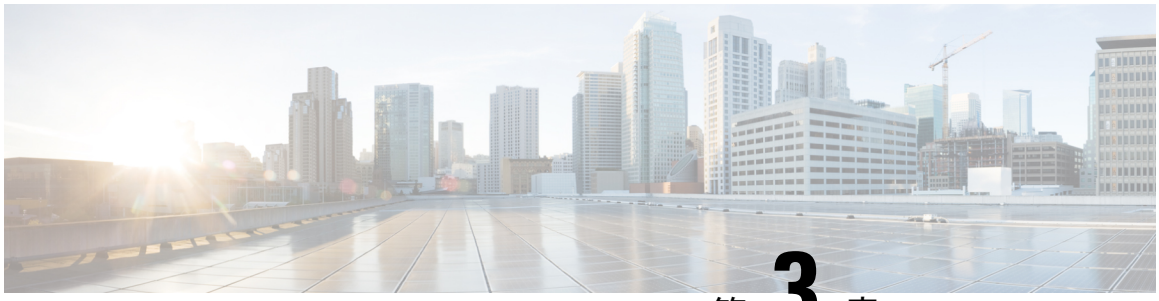
手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**interface** *interface-id* 例： `Device(config)# interface gigabitethernet1/0/1`	CDP をイネーブルにするインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	**cdp enable** 例： `Device(config-if)# cdp enable`	ディセーブルにされているインターフェイスで CDP をイネーブルにします。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device# show running-config`	入力を確認します。
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。

CDP のモニタおよびメンテナンス

表 *1 : CDP* 情報を表示するためのコマンド

コマンド（**Command**）	説明
clear cdp counters	トラフィック カウンタを0にリセットします。
clear cdp table	ネイバー デバイスに関する情報を収めた CDP テーブルを削除します。
show cdp	送信間隔、送信したパケットの保持時間などのグローバル情報を表示します。
show cdp entry *entry-name* ［**version**］［**protocol**］	特定のネイバーに関する情報を表示します。 アスタリスク（*）を入力してすべての CDP ネイバーを表示することも、情報が必要なネイバーの名前を入力することもできます。 また、指定されたネイバー上でイネーブルになっているプロトコルの情報や、デバイス上で稼働しているソフトウェアのバージョン情報が表示されるように、表示内容を制限することもできます。
show cdp interface［*interface-id*］	CDP がイネーブルに設定されているインターフェイスの情報を表示します。 必要なインターフェイスの情報だけを表示できます。
show cdp neighbors［*interface-id*］［*detail*］	装置タイプ、インターフェイス タイプ、インターフェイス番号、保持時間の設定値、機能、プラットフォーム、ポート ID を含めたネイバー情報を表示します。 特定のインターフェイスに関するネイバー情報だけを表示したり、詳細表示にするため表示内容を拡張したりできます。
show cdp traffic	CDP カウンタ（送受信されたパケット数およびチェックサムエラーを含む）を表示します。



第 3 章

簡易ネットワーク管理プロトコルの設定

SNMP の前提条件

サポートされている SNMP バージョン

このソフトウェア リリースは、次の SNMP バージョンをサポートしています。

- SNMPv1：RFC1157 に規定された SNMP（完全インターネット標準）。
- SNMPv2C は、SNMPv2Classic のバルク検索機能を残し、エラー処理を改善したうえで、SNMPv2Classic のパーティ ベースの管理およびセキュリティ フレームワークをコミュニティ ストリング ベースの管理フレームワークに置き換えたものです。次の機能があります。
  - SNMPv2：RFC 1902 ～ 1907 に規定された SNMP バージョン 2（ドラフト版インターネット標準）
  - SNMPv2C：RFC 1901 に規定された SNMPv2 のコミュニティ ストリング ベースの管理フレームワーク（試験版インターネット プロトコル）
- SNMPv3：SNMP のバージョン 3 は、RFC 2273 ～ 2275 に規定されている相互運用可能な標準ベース プロトコルです。SNMPv3 は、ネットワーク上のパケットを認証、暗号化することでデバイスへのアクセスに対するセキュリティを提供します。SNMPv3 は、次のセキュリティ機能を備えています。
  - メッセージの完全性：パケットが伝送中に改ざんされないようにします。

- 認証：有効な送信元からのメッセージであるかどうかを判別します。
- 暗号化：パッケージの内容をミキシングし、許可されていない送信元に内容が読まれることを防止します。

（注） 暗号化を選択するには、**priv** キーワードを入力します。

SNMPv1 と SNMPv2C は、ともにコミュニティベース形式のセキュリティを使用します。エージェントの MIB にアクセスできるマネージャのコミュニティが、IP アドレス アクセス コントロール リストおよびパスワードによって定義されます。

SNMPv2C にはバルク検索機能が組み込まれ、より詳細なエラー メッセージを管理ステーションに報告します。バルク検索機能は、テーブルや大量の情報を検索し、必要な往復回数を削減します。SNMPv2C ではエラー処理機能が改善され、さまざまなエラーを区別するための拡張エラー コードが使用されています。これらのエラーは、SNMPv1 では単一のエラー コードで報告されます。SNMPv2 では、エラー リターン コードでエラー タイプが報告されるようになりました。

SNMPv3 では、セキュリティ モデルとセキュリティ レベルの両方が提供されています。セキュリティ モデルは、ユーザとユーザが属しているグループ用に設定された認証方式です。セキュリティ レベルとは、セキュリティ モデル内で許可されるセキュリティのレベルです。セキュリティ レベルとセキュリティ モデルの組み合わせにより、SNMP パケットを扱うときに使用するセキュリティ方式が決まります。使用可能なセキュリティ モデルは、SNMPv1、SNMPv2C、および SNMPv3 です。

次の表では、この特性を識別し、セキュリティ モデルとセキュリティ レベルの異なる組み合わせを比較します。

表 *2: SNMP* セキュリティ モデルおよびセキュリティ レベル

モデル	水準器	認証	暗号化（Encryption）	結果
SNMPv1	noAuthNoPriv	コミュニティ ストリング（Community string）	なし	コミュニティ ストリングの照合を使用して認証します。
SNMPv2C	noAuthNoPriv	コミュニティ ストリング（Community string）	なし	コミュニティ ストリングの照合を使用して認証します。
SNMPv3	noAuthNoPriv	[ユーザ名（Username）]	なし	ユーザ名の照合を使用して認証します。

モデル	水準器	認証	暗号化（Encryption）	結果
SNMPv3	authNoPriv	Message Digest 5（MD5）または Secure Hash Algorithm（SHA）	なし	HMAC-MD5 アルゴリズムまたは HMAC-SHA アルゴリズムに基づいて認証します。
SNMPv3	authPriv	MD5 または SHA	データ暗号規格（DES）または Advanced Encryption Standard（AES）	HMAC-MD5 アルゴリズムまたは HMAC-SHA アルゴリズムに基づいて認証します。 次の暗号化アルゴリズムで、User-based Security Model（USM）を指定できます。 • CBC-DES（DES-56）規格に基づく認証に加えた DES 56 ビット暗号化 • 3DES 168 ビット暗号化 • AES 128 ビット暗号化、192 ビット暗号化、または 256 ビット暗号化

管理ステーションでサポートされているSNMPバージョンを使用するには、SNMPエージェントを設定する必要があります。エージェントは複数のマネージャと通信できるため、SNMPv1、SNMPv2C、およびSNMPv3を使用する通信をサポートするようにソフトウェアを設定できます。

SNMP の制約事項

バージョンの制約事項

- SNMPv1 は informs をサポートしていません。

SNMP に関する情報

SNMP の概要

SNMP は、マネージャとエージェント間の通信のメッセージ フォーマットを提供するアプリケーションレイヤプロトコルです。SNMP システムは、SNMP マネージャ、SNMP エージェント、および管理情報ベース（MIB）で構成されます。SNMP マネージャは、Cisco Prime Infrastructure などのネットワーク管理システム（NMS）に統合できます。エージェントおよび MIB は、デバイスに常駐します。デバイス上で SNMP を設定するには、マネージャとエージェント間の関係を定義します。

SNMP エージェントは MIB 変数を格納し、SNMP マネージャはこの変数の値を要求または変更できます。マネージャはエージェントから値を取得したり、エージェントに値を格納したりできます。エージェントは、デバイスパラメータやネットワーク データの保存場所である MIB から値を収集します。また、エージェントはマネージャのデータ取得またはデータ設定の要求に応答できます。

エージェントは非送信請求トラップをマネージャに送信できます。トラップは、ネットワーク上のある状態を SNMP マネージャに通知するメッセージです。トラップは不正なユーザ認証、再起動、リンク ステータス（アップまたはダウン）、MAC アドレス追跡、TCP 接続の終了、ネイバーとの接続の切断などの重要なイベントの発生を意味する場合があります。

SNMP マネージャ機能

SNMP マネージャは、MIB 情報を使用して、次の表に示す動作を実行します。

表 *3 : SNMP* の動作

操作	説明
get-request	特定の変数から値を取得します。
get-next-request	テーブル内の変数から値を取得します。[1]
get-bulk-request[2]	テーブルの複数の行など、通常はサイズの小さい多数のデータ ブロックに分割して送信する必要がある巨大なデータ ブロックを取得します。

操作	説明
get-response	NMS から送信される get-request、get-next-request、および set-request に対して応答します。
set-request	特定の変数に値を格納します。
トラップ	SNMP エージェントから SNMP マネージャに送られる、イベントの発生を伝える非送信請求メッセージです。

[1] この動作を使用した場合、SNMP マネージャは厳密な変数名を知る必要はありません。テーブル内を順に検索して、必要な変数を検出します。
[2] get-bulk コマンドを使用できるのは、SNMPv2 以上に限られます。

SNMP エージェント機能

SNMP エージェントは、次のようにして SNMP マネージャ要求に応答します。

- MIB 変数の取得：SNMP エージェントは NMS からの要求に応答して、この機能を開始します。エージェントは要求された MIB 変数の値を取得し、この値を使用して NMS に応答します。
- MIB 変数の設定：SNMP エージェントは NMS からのメッセージに応答して、この機能を開始します。SNMP エージェントは、MIB 変数の値を NMS から要求された値に変更します。

エージェントで重要なイベントが発生したことを NMS に通知するために、SNMP エージェントは非送信請求トラップ メッセージも送信します。トラップ条件の例には、ポートまたはモジュールがアップまたはダウン状態になった場合、スパニングツリー トポロジが変更された場合、認証に失敗した場合などがあります。

SNMP コミュニティ ストリング

SNMP コミュニティ ストリングは、MIB オブジェクトおよび関数へのアクセスを組み込みのパスワードとして認証します。NMS がデバイスにアクセスするには、NMS 上のコミュニティ ストリング定義が デバイス 上の 3 つのコミュニティ ストリング定義の少なくとも 1 つと一致しなければなりません。

コミュニティ ストリングの属性は、次のいずれかです。

- 読み取り専用（RO）：コミュニティ ストリングを除き MIB 内のすべてのオブジェクトに、許可された管理ステーションに対する読み取りアクセス権を与えますが、書き込みアクセスは許可しません。
- 読み取り-書き込み（RW）：MIB 内のすべてのオブジェクトに、許可された管理ステーションに対する読み取りおよび書き込みアクセス権を与えますが、コミュニティ ストリングへのアクセスは許可しません。
- クラスタを作成すると、コマンド デバイスがメンバ デバイスと SNMP アプリケーション間のメッセージ交換を管理します。Network Assistant ソフトウェアは、コマンド デバイス

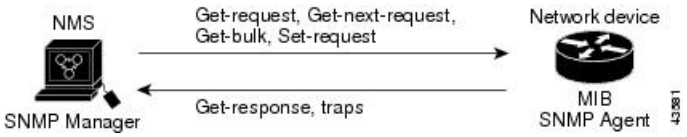
上で最初に設定された RW および RO コミュニティ ストリングにメンバ デバイス番号（@esN、N はデバイス番号）を追加し、これらのストリングをメンバ デバイスに伝播します。

SNMP MIB 変数アクセス

NMS の例として、Cisco Prime Infrastructure ネットワーク管理ソフトウェアがあります。Cisco Prime Infrastructure 3.1 ソフトウェアは、デバイス MIB 変数を使用して装置変数を設定し、ネットワーク上の装置をポーリングして特定の情報を取得します。ポーリング結果は、グラフ形式で表示されます。この結果を解析して、インターネットワーキング関連の問題のトラブルシューティング、ネットワーク パフォーマンスの改善、デバイス設定の確認、トラフィック負荷のモニタなどを行うことができます。

次の図に示すように、SNMP エージェントは MIB からデータを収集します。エージェントは SNMP マネージャに対し、トラップ（特定イベントの通知）を送信でき、SNMP マネージャはトラップを受信して処理します。トラップは、ネットワーク上で発生した不正なユーザ認証、再起動、リンク ステータス（アップまたはダウン）、MAC アドレス トラッキングなどの状況を SNMP マネージャに通知します。SNMP エージェントはさらに、SNMP マネージャから *get-request*、*get-next-request*、および *set-request* 形式で送信される MIB 関連のクエリに応答します。

図 *1: SNMP* ネットワーク



SNMP 通知

SNMP を使用すると、特定のイベントが発生した場合に、デバイスから SNMP マネージャに通知を送信できます。SNMP 通知は、トラップまたは情報要求として送信できます。コマンド構文では、トラップまたは情報を選択するオプションがコマンドにない限り、キーワード traps はトラップ、情報、またはその両方を表します。**snmp-server host** コマンドを使用して、トラップまたは情報として SNMP 通知を送信するかどうかを指定します。

（注） SNMPv1 は informs をサポートしていません。

トラップは信頼性に欠けます。受信側はトラップを受信しても確認応答を送信しないので、トラップが受信されたかどうかが送信側にわからないからです。情報要求の場合、受信した SNMP マネージャは SNMP 応答プロトコル データ ユニット（PDU）でメッセージを確認します。送信側が応答を受信しなかった場合は、再び情報要求を送信できます。再送信できるので、情報の方がトラップより意図した宛先に届く可能性が高くなります。

情報の方がトラップより信頼性が高いのは、デバイスおよびネットワークのリソースを多く消費するという特性にも理由があります。送信と同時に廃棄されるトラップと異なり、情報要求は応答を受信するまで、または要求がタイムアウトになるまで、メモリ内に保持されます。トラップの送信は1回限りですが、情報は数回にわたって再送信つまり再試行が可能です。再送信の回数が増えるとトラフィックが増加し、ネットワークのオーバーヘッドが高くなる原因にもなります。したがって、トラップにするか情報にするかは、信頼性を取るかリソースを取るかという選択になります。SNMPマネージャですべての通知を受信することが重要な場合は、情報要求を使用してください。ネットワークまたはデバイスのメモリ上のトラフィックが問題になる場合で、なおかつ通知が不要な場合は、トラップを使用してください。

SNMP ifIndex MIB オブジェクト値

SNMP エージェントの IF-MIB モジュールがリブート後すぐに起動されます。さまざまな物理インターフェイス ドライバが IF-MIB モジュールの登録を初期化されているように、「インデックス番号をください」と示します。IF-MIB モジュールが先着順で使用可能な次の ifIndex 番号を割り当てます。つまり、1 つのリブートから他のリブートへのドライバの初期化順序のマイナーな違いが、同じ物理インターフェイスにリブートを行う以前のものとは別のインデックス番号を取得する可能性があるということです（インデックス持続が有効化されていない限り）。

SNMP のデフォルト設定

機能	デフォルト設定
SNMP エージェント	ディセーブル[3]
SNMP トラップ レシーバ	未設定
SNMP トラップ	TCP接続のトラップ（tty）以外は、イネーブルではありません。
SNMP バージョン（SNMP version）	バージョン キーワードがない場合、デフォルトはバージョン 1 になります。
SNMPv3 認証	キーワードを入力しなかった場合、セキュリティ レベルはデフォルトで **noauth**（noAuthNoPriv）になります。
SNMP 通知タイプ	タイプが指定されていない場合、すべての通知が送信されます。

[3] これは、デバイスが起動し、スタートアップ コンフィギュレーションに **snmp-server** グローバル コンフィギュレーション コマンドが設定されていない場合のデフォルトです。

SNMP 設定時の注意事項

デバイスが起動し、デバイスのスタートアップ コンフィギュレーションに少なくとも 1 つの **snmp-server** グローバル コンフィギュレーション コマンドが設定されている場合、SNMP エージェントはイネーブルになります。

SNMP グループは、SNMP ユーザを SNMP ビューに対応付けるテーブルです。SNMP ユーザは、SNMP グループのメンバです。SNMP ホストは、SNMP トラップ動作の受信側です。SNMP エンジン *ID* は、ローカルまたはリモート SNMP エンジンの名前です。

SNMP グループを設定するときには、次の注意事項に従ってください。

- SNMP グループを設定するときは、通知ビューを指定しません。**snmp-server host** グローバル コンフィギュレーション コマンドがユーザの通知ビューを自動生成し、そのユーザを対応するグループに追加します。グループの通知ビューを変更すると、そのグループに対応付けられたすべてのユーザが影響を受けます。
- リモート ユーザを設定する場合は、ユーザが存在するデバイスのリモート SNMP エージェントに対応する IP アドレスまたはポート番号を指定します。
- 特定のエージェントのリモート ユーザを設定する前に、**snmp-server engineID** グローバル コンフィギュレーション コマンドを **remote** オプションとともに使用して、SNMP エンジン ID を設定してください。リモート エージェントの SNMP エンジン ID およびユーザ パスワードを使用して認証およびプライバシー ダイジェストが算出されます。先にリモート エンジン ID を設定しておかないと、コンフィギュレーション コマンドがエラーになります。
- SNMP 情報を設定するときには、プロキシ要求または情報の送信先となるリモート エージェントの SNMP エンジン ID を SNMP データベースに設定しておく必要があります。
- ローカル ユーザがリモート ホストと関連付けられていない場合、デバイスは **auth**（authNoPriv）および **priv**（authPriv）の認証レベルの情報を送信しません。
- SNMP エンジン ID の値を変更すると、重大な影響が生じます。（コマンド ラインで入力された）ユーザのパスワードは、パスワードおよびローカル エンジン ID に基づいて、MD5 または SHA セキュリティ ダイジェストに変換されます。コマンドラインのパスワードは、RFC 2274 の規定に従って廃棄されます。このようにパスワードが廃棄されるため、エンジン ID 値を変更した場合は SNMPv3 ユーザのセキュリティ ダイジェストが無効となり、**snmp-server user** *username* グローバル コンフィギュレーション コマンドを使用して、SNMP ユーザを再設定する必要があります。エンジン ID を変更した場合は、同様の制限によってコミュニティ ストリングも再設定する必要があります。

SNMP の設定方法

コミュニティ ストリングの設定

SNMP マネージャとエージェントの関係を定義するには、SNMP コミュニティ ストリングを使用します。コミュニティ ストリングは、デバイス上のエージェントへのアクセスを許可する、パスワードと同様の役割を果たします。ストリングに対応する次の特性を 1 つまたは複数指定することもできます。

- コミュニティ ストリングを使用してエージェントにアクセスできる SNMP マネージャの IP アドレスのアクセス リスト
- 指定のコミュニティにアクセスできるすべての MIB オブジェクトのサブセットを定義する MIB ビュー
- コミュニティにアクセスできる MIB オブジェクトの読み書き権限または読み取り専用権限

デバイス上でコミュニティ ストリングを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [*access-list-number*] 例： `Device(config)# snmp-server community comaccess ro 4`	コミュニティストリングを設定します。 （注） コンテキスト情報を区切るには @ 記号を使用します。このコマンドの設定時に SNMP コミュニティ ストリングの一部として @ 記号を使用しないでください。 • *string* には、パスワードと同様に機能し、SNMP プロトコルへのアクセスを許可するストリングを指定します。任意の長さのコミュニティ ストリングを 1 つまたは複数設定できます。 •（任意）**view** には、コミュニティがアクセスできるビュー レコードを指定します。 •（任意）許可された管理ステーションで MIB オブジェクトを取得する場合は読み取り専用（**ro**）、許可さ

	コマンドまたはアクション	目的
		れた管理ステーションで MIB オブジェクトを取得および変更する場合は読み書き（**rw**）を指定します。デフォルトでは、コミュニティ ストリングはすべてのオブジェクトに対する読み取り専用アクセスを許可します。 • （任意）*access-list-number* には、1 ～ 99 および 1300 ～ 1999 の標準 IP アクセスリスト番号を入力します。
ステップ 4	**access-list** *access-list-number* { **deny** | **permit** } *source* [*source-wildcard*] 例： `Device(config)# access-list 4 deny any`	（任意）ステップ 3 で標準 IP アクセスリスト番号を指定してリストを作成した場合は、必要に応じてコマンドを繰り返します。 • *access-list-number* には、ステップ 3 で指定したアクセス リスト番号を入力します。 • **deny** キーワードは、条件が一致した場合にアクセスを拒否します。**permit** キーワードは、条件が一致した場合にアクセスを許可します。 • *source* には、コミュニティ ストリングを使用してエージェントにアクセスできる SNMP マネージャの IP アドレスを入力します。 • （任意）*source-wildcard* には、source に適用されるワイルドカードビットをドット付き 10 進表記で入力します。無視するビット位置には 1 を設定します。 アクセス リストの末尾には、すべてに対する暗黙の拒否ステートメントが常に存在することに注意してください。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 6	**show running-config** 例： `Device# show running-config`	入力を確認します。
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。

次のタスク

SNMP コミュニティのアクセスをディセーブルにするには、そのコミュニティのコミュニティ ストリングをヌル ストリングに設定します（コミュニティ ストリングに値を入力しないでください）。

特定のコミュニティ ストリングを削除するには、**no snmp-server** コミュニティ ストリング 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。

デバイスのローカルまたはリモート SNMP サーバ エンジンを表す識別名（エンジン ID）を指定できます。SNMP ユーザを SNMP ビューにマッピングする、SNMP サーバ グループを設定し、新規ユーザを SNMP グループに追加できます。

SNMP グループおよびユーザの設定

デバイスのローカルまたはリモート SNMP サーバ エンジンを表す識別名（エンジン ID）を指定できます。SNMP ユーザを SNMP ビューにマッピングする、SNMP サーバ グループを設定し、新規ユーザを SNMP グループに追加できます。

デバイス上の SNMP グループとユーザを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 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	**snmp-server engineID** { **local** *engineid-string* | **remote** *ip-address* [**udp-port** *port-number*] *engineid-string* } 例： `Device(config)# snmp-server engineID local 1234`	SNMP のローカル コピーまたはリモート コピーに名前を設定します。 • *engineid-string* は、SNMP のコピー名を指定する 24 文字の ID ストリングです。後続ゼロが含まれる場合は、24 文字のエンジン ID すべてを指定する必要はありません。指定するのは、エンジン ID のうちゼロのみが続く箇所を除いた部分だけです。手順例では、123400000000000000000000 のエンジン ID を設定します。 • **remote**を指定した場合、SNMP のリモート コピーが置かれているデバイスの *ip-address* を指定し、任意でリモート デバイスのユーザ データグラム プロトコル（UDP）ポートを指定します。デフォルトは 162 です。
ステップ 4	**snmp-server group** *group-name readview writeview notifyview access* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read**] [**write**] [**notify**] [**access**] 例： `Device(config)# snmp-server group public v2c access lmnop`	リモート デバイス上で新しい SNMP グループを設定します。 *group-name* には、グループの名前を指定します。 次のいずれかのセキュリティ モデルを指定します。 • **v1** は、最も安全性の低いセキュリティ モデルです。 • **v2c** は、2 番目に安全性の低いセキュリティ モデルです。標準の 2 倍の幅で情報および整数を伝送できます。 • **v3**最も安全な場合には、次の認証レベルの 1 つを選択する必要があります。 **auth**：Message Digest 5（MD5） およびセキュア ハッシュ アルゴリズ

	コマンドまたはアクション	目的
		ム（SHA）によるパケット認証を可能にします。 **noauth**：noAuthNoPriv セキュリティレベルを可能にします。キーワードを指定しなかった場合、これがデフォルトです。 **priv**：データ暗号規格（DES）によるパケット暗号化（プライバシーともいう）を可能にします。 （任意）**read** *readview* とともに、エージェントの内容を表示できるビューの名前を表すストリング（64 文字以下）を入力します。 （任意）**write** *writeview* とともに、データを入力し、エージェントの内容を設定できるビューの名前を表すストリング（64 文字以下）を入力します。 （任意）**notify** *notifyview* とともに、通知、情報、またはトラップを指定するビューの名前を表すストリング（64 文字以下）を入力します。 （任意）**access** *access-list* とともに、アクセス リスト名のストリング（64 文字以下）を入力します。
ステップ 5	**snmp-server user** *username group-name host port access-list access-list access-list auth-passwordpriv* 例： `Device(config)# snmp-server user Pat public v2c`	SNMP グループに対して新規ユーザを追加します。 *username* は、エージェントに接続するホスト上のユーザ名です。 *group-name* は、ユーザが関連付けられているグループの名前です。 **remote** を入力して、ユーザが所属するリモート SNMP エンティティおよびそのエンティティのホスト名または IP アドレスとともに、任意で UDP ポート番号を指定します。デフォルトは 162 です。

	コマンドまたはアクション	目的
		SNMP バージョン番号（**v1**、**v2c**、または **v3**）を入力します。**v3** を入力すると、次のオプションが追加されます。 • **encrypted** パスワードを暗号化形式で表示するように指定します。このキーワードは、**v3** キーワードが指定されている場合のみ使用可能です。 • **auth** では、認証レベルを設定します。HMAC-MD5-96（**md5**）またはHMAC-SHA-96（**sha**）認証レベルのいずれかを指定できます。また、*auth-password* でパスワードの文字列を指定する必要があります（最大 64 文字）。 **v3** を入力すると、次のキーワードを使用して（64 文字以下）、プライベート（**priv**）暗号化アルゴリズムおよびパスワード ストリング *priv-password* を設定することもできます。 • **priv** ユーザベース セキュリティ モデル（USM）を指定します。 • **des** 56 ビット DES アルゴリズムを使用する場合に指定します。 • **3des** 168 ビット DES アルゴリズムを使用する場合に指定します。 • **aes** DES アルゴリズムを使用する場合に指定します。128 ビット暗号化、192 ビット暗号化、または 256 ビット暗号化のいずれかを選択する必要があります。 （任意）**access** *access-list* とともに、アクセス リスト名のストリング（64 文字以下）を入力します。
ステップ 6	**end** 例：	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	`Device(config)# end`	
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

SNMP 通知の設定

トラップ マネージャは、トラップを受信して処理する管理ステーションです。トラップは、特定のイベントが発生したときにデバイスが生成するシステム アラートです。デフォルトでは、トラップ マネージャは定義されず、トラップは送信されません。この Cisco IOS Release が稼働しているデバイスでは、トラップ マネージャを無制限に設定できます。

（注） コマンド構文で **traps** というワードを使用するコマンドは多数あります。トラップまたは情報を選択するオプションがコマンドにない限り、キーワード **traps** はトラップ、情報のいずれか、またはその両方を表します。**snmp-server host** グローバル コンフィギュレーション コマンドを使用して、トラップまたは情報として SNMP 通知を送信するかどうかを指定します。

snmp-server enable traps グローバル コンフィギュレーション コマンドを **snmp-server host** グローバル コンフィギュレーション コマンドと組み合わせて使用すると、次の表に示す通知タイプを特定のホストで受信できます。これらのトラップの一部または全部をイネーブルにして、これを受信するようにトラップ マネージャを設定できます。

（注） **snmp-server enable traps** コマンドは、デバイスのローカル認証のためのトラップをサポートしていません。

ホストにトラップまたは情報を送信するようにデバイスを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-server engineID remote** *ip-address engineid-string* 例： `Device(config)# snmp-server engineID remote 192.180.1.27 00000063000100a1c0b4011b`	リモート ホストのエンジン ID を指定します。
ステップ 4	**snmp-server user** *username group-name host port access-list access-list access-list* [**remote** *udp-port*] {**v1** | **v2c** | **v3** [**encrypted**] [**auth** {**md5** | **sha**}]} 例： `Device(config)# snmp-server user Pat public v2c`	SNMP ユーザを設定し、ステップ 3 で作成したリモートホストに関連付けます。 （注） アドレスに対応するリモートユーザを設定するには、先にリモート ホストのエンジン ID を設定しておく必要があります。このようにしないと、エラーメッセージが表示され、コマンドが実行されません。
ステップ 5	**snmp-server group** *group-name* {**v1** | **v2c** | **v3** {**auth** | **noauth** | **priv**}} [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*] 例： `Device(config)# snmp-server group public v2c access lmnop`	SNMP グループを設定します。
ステップ 6	**snmp-server host** *host-addr* [**informs** | **traps**] [**version** {**1** | **2c** | **3** {**auth** | **noauth** | **priv**}}] *community-string* [*notification-type*] 例：	SNMP トラップ動作の受信先を指定します。 *host-addr* には、ホスト（対象となる受信側）の名前またはインターネットアドレスを指定します。

	コマンドまたはアクション	目的
	`Device(config)# snmp-server host 203.0.113.1 comaccess snmp`	（任意）SNMP トラップをホストに送信するには、**traps**（デフォルト）を指定します。 （任意）SNMP 情報をホストに送信するには、**informs** を指定します。 （任意）SNMP **version**（**1**、**2c**、または **3**）を指定します。SNMPv1 は informs をサポートしていません。 （任意）バージョン 3 の場合、認証レベル **auth**、**noauth**、または **priv** を選択します。 （注） **priv** キーワードは、暗号化ソフトウェアイメージがインストールされている場合のみ使用可能です。 *community-string* には、**version 1** または **version 2c** が指定されている場合、通知動作で送信される、パスワードに類似したコミュニティストリングを入力します。**version 3** が指定されている場合、SNMPv3 ユーザ名を入力します。 コンテキスト情報を区切るには@記号を使用します。このコマンドの設定時にSNMPコミュニティストリングの一部として@記号を使用しないでください。 （任意）*notification-type* には、上の表に記載されているキーワードを使用します。タイプが指定されていない場合、すべての通知が送信されます。
ステップ 7	**snmp-server enable traps** *notification-types* 例： `Device(config)# snmp-server enable traps snmp`	デバイスでのトラップまたはインフォームの送信を有効にし、送信する通知の種類を指定します。通知タイプの一覧については、上の表を参照するか、次のコマンドを入力してください。 **snmp-server enable traps?** 複数のトラップタイプを有効にするには、トラップタイプごとに **snmp-server**

	コマンドまたはアクション	目的
		enable traps コマンドを個別に入力する必要があります。 （注） 通知タイプ **port-security** を使用してトラップを設定する際に、まずポートセキュリティトラップを設定して、次に以下のポート セキュリティ トラップ レートを設定します。 1. **snmp-server enable trapsport-security** 2. **snmp-server enable trapsport-securitytrap-rate** レート
ステップ 8	**snmp-server trap-source** *interface-id* 例： `Device(config)# snmp-server trap-source GigabitEthernet1/0/1`	（任意）送信元インターフェイスを指定します。このインターフェイスによってトラップ メッセージの IP アドレスが提供されます。情報の送信元 IP アドレスも、このコマンドで設定します。
ステップ 9	**snmp-server queue-length** *length* 例： `Device(config)# snmp-server queue-length 20`	（任意）各トラップホストのメッセージキューの長さを指定します。指定できる範囲は 1 ～ 5000 です。デフォルトは 10 です。
ステップ 10	**snmp-server trap-timeout** 秒 例： `Device(config)# snmp-server trap-timeout 60`	（任意）トラップ メッセージを再送信する頻度を指定します。指定できる範囲は 1 ～ 1000 です。デフォルトは 30 秒です。
ステップ 11	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 12	**show running-config** 例： `Device# show running-config`	入力を確認します。

	コマンドまたはアクション	目的
ステップ 13	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

次のタスク

snmp-server host コマンドでは、通知を受信するホストを指定します。**snmp-server enable traps** コマンドによって、指定された通知方式（トラップおよび情報）がグローバルで有効になります。ホストが情報を受信できるようにするには、そのホストに対応する **snmp-server host informs** コマンドを設定し**snmp-server enable traps** コマンドを使用して情報をグローバルに有効にする必要があります。

指定したホストがトラップを受信しないようにするには、**no snmp-server host** *host* 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。キーワードを指定しないで **no snmp-server host** コマンドを使用すると、ホストへのトラップは無効になりますが、情報は無効になりません。情報を無効にするには、**no snmp-server host informs** グローバル コンフィギュレーション コマンドを使用してください。特定のトラップ タイプを無効にするには、**no snmp-server enable traps** *notification-types* 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。

エージェント コンタクトおよびロケーションの設定

SNMPエージェントのシステム接点およびロケーションを設定して、コンフィギュレーション ファイルからこれらの記述にアクセスできるようにするには、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`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**snmp-server contact** テキスト 例：	システムの連絡先文字列を設定します。

	コマンドまたはアクション	目的
	`Device(config)# snmp-server contact Dial System Operator at beeper 21555`	
ステップ 4	**snmp-server location** テキスト 例： `Device(config)# snmp-server location Building 3/Room 222`	システムの場所を表す文字列を設定します。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device# show running-config`	入力を確認します。
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。

SNMP を通して使用する TFTP サーバの制限

SNMP を介したコンフィギュレーション ファイルの保存とロードに使用する TFTP サーバを、アクセス リストで指定されたサーバに限定するには、次の手順を実行します。

手順

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

	コマンドまたはアクション	目的
ステップ 3	**snmp-server tftp-server-list** *access-list-number* 例： `Device(config)# snmp-server tftp-server-list 44`	SNMP を介したコンフィギュレーション ファイルのコピーに使用する TFTP サーバを、アクセス リストのサーバに限定します。 *access-list-number* には、1 ～ 99 および 1300 ～ 1999 の標準 IP アクセス リスト番号を入力します。
ステップ 4	**access-list** *access-list-number sourcesource-wildcard*{**deny** | **permit**}[] 例： `Device(config)# access-list 44 permit 10.1.1.2`	標準アクセス リストを作成し、コマンドを必要な回数だけ実行します。 *access-list-number* には、ステップ 3 で指定したアクセス リスト番号を入力します。 **deny** キーワードは、条件が一致した場合にアクセスを拒否します。**permit** キーワードは、条件が一致した場合にアクセスを許可します。 *source* には、デバイスにアクセスできる TFTP サーバの IP アドレスを入力します。 （任意）*source-wildcard* には、source に適用されるワイルドカードビットをドット付き 10 進表記で入力します。無視するビット位置には 1 を設定します。 アクセス リストの末尾には、すべてに対する暗黙の拒否ステートメントが常に存在します。
ステップ 5	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device# show running-config`	入力を確認します。
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`	

SNMP エージェントのディセーブル化

no snmp-server グローバル コンフィギュレーション コマンドは、デバイス上で実行している SNMP エージェントのすべてのバージョン（バージョン 1、バージョン 2C、バージョン 3）をディセーブルにします。入力した最初の **snmp-server** グローバル コンフィギュレーション コマンドによって、SNMP エージェントのすべてのバージョンを再度イネーブルにします。特に SNMP をイネーブルにするために指定された Cisco IOS コマンドはありません。

SNMP エージェントをディセーブルにするには、次の手順を実行します。

始める前に

SNMP エージェントをディセーブルにする前にイネーブルにする必要があります。デバイス上で入力した最初の **snmp-server** グローバル コンフィギュレーション コマンドによって SNMP エージェントがイネーブルになります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**no snmp-server** 例： `Device(config)# no snmp-server`	SNMP エージェント動作をディセーブルにします。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

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

SNMP ステータスのモニタリング

不正なコミュニティ ストリング エントリ、エラー、要求変数の数など、SNMP の入出力統計情報を表示するには、**show snmp** 特権 EXEC コマンドを使用します。また、次の表にリストされたその他の特権 EXEC コマンドを使用して、SNMP 情報を表示することもできます。

表 *4 : SNMP* 情報を表示するためのコマンド

コマンド（Command）	目的
show snmp	SNMP 統計情報を表示します。
	デバイスに設定されているローカル SNMP エンジンおよびすべてのリモート エンジンに関する情報を表示します。
show snmp group	ネットワーク上の各 SNMP グループに関する情報を表示します。
show snmp pending	保留中の SNMP 要求の情報を表示します。
show snmp sessions	現在の SNMP セッションの情報を表示します。
show snmp user	SNMP ユーザ テーブルの各 SNMP ユーザ名に関する情報を表示します。 （注） このコマンドは、**auth** | **noauth** | **priv** モードの SNMPv3 設定情報を表示するときに使用する必要があります。この情報は、**show running-config** の出力には表示されません。

SNMP の例

次に、SNMP のすべてのバージョンをイネーブルにする例を示します。この設定では、任意の SNMP マネージャがコミュニティ ストリング *public* を使用して、読み取り専用権限ですべてのオブジェクトにアクセスできます。この設定では、デバイス はトラップを送信しません。

```
Device(config)# snmp-server community public
```

次に、任意の SNMP マネージャがコミュニティ ストリング *public* を使用して、読み取り専用権限ですべてのオブジェクトにアクセスする例を示します。デバイスはさらに、SNMPv1 を使用してホスト 192.180.1.111 および 192.180.1.33 に、SNMPv2C を使用してホスト 192.180.1.27 に VTP トラップを送信します。コミュニティ ストリング *public* は、トラップとともに送信されます。

```
Device(config)# snmp-server community public
Device(config)# snmp-server enable traps vtp
Device(config)# snmp-server host 192.180.1.27 version 2c public
Device(config)# snmp-server host 192.180.1.111 version 1 public
Device(config)# snmp-server host 192.180.1.33 public
```

次に、*comaccess* コミュニティ ストリングを使用するアクセス リスト 4 のメンバに、すべてのオブジェクトへの読み取り専用アクセスを許可する例を示します。その他の SNMP マネージャは、どのオブジェクトにもアクセスできません。SNMP 認証障害トラップは、SNMPv2C がコミュニティ ストリング *public* を使用してホスト *cisco.com* に送信します。

```
Device(config)# snmp-server community comaccess ro 4
Device(config)# snmp-server enable traps snmp authentication
Device(config)# snmp-server host cisco.com version 2c public
```

次に、エンティティ MIB トラップをホスト *cisco.com* に送信する例を示します。コミュニティ ストリングは制限されます。1 行めで、デバイスはすでにイネーブルになっているトラップ以外に、エンティティ MIB トラップを送信できるようになります。2 行目はこれらのトラップの宛先を指定し、ホスト *cisco.com* に対する以前の **snmp-server** ホスト コマンドを無効にします。

```
Device(config)# snmp-server enable traps entity
Device(config)# snmp-server host cisco.com restricted entity
```

次に、コミュニティ ストリング *public* を使用して、すべてのトラップをホスト *myhost.cisco.com* に送信するようにデバイスをイネーブルにする例を示します。

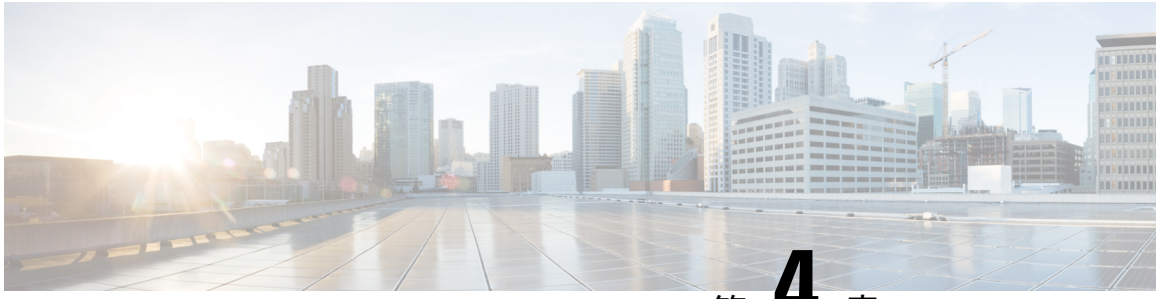
```
Device(config)# snmp-server enable traps
Device(config)# snmp-server host myhost.cisco.com public
```

次に、ユーザとリモート ホストを関連付けて、ユーザがグローバル コンフィギュレーション モードのときに **auth**（authNoPriv）認証レベルで情報を送信する例を示します。

```
Device(config)# snmp-server engineID remote 192.180.1.27 00000063000100a1c0b4011b
Device(config)# snmp-server group authgroup v3 auth
Device(config)# snmp-server user authuser authgroup remote 192.180.1.27 v3 auth md5
mypassword
Device(config)# snmp-server user authuser authgroup v3 auth md5 mypassword
Device(config)# snmp-server host 192.180.1.27 informs version 3 auth authuser config
Device(config)# snmp-server enable traps
Device(config)# snmp-server inform retries 0
```

簡易ネットワーク管理プロトコルの機能の履歴と情報

リリース	変更箇所
Cisco IOS XE Everest 16.5.1a	この機能が導入されました。



第 4 章

サービス レベル契約の設定

この章では、スイッチで Cisco IOS IP サービス レベル契約（SLA）を使用する方法について説明します。

特に明記しないかぎり、スイッチという用語はスタンドアロン スイッチまたはスイッチ スタックを意味します。

SLA の制約事項

ここでは、SLA の制約事項を示します。

次に示すのは、IP SLA ネットワーク パフォーマンス測定の制約事項です。

- デバイスは、ゲートキーパー登録遅延動作測定を使用した Voice over IP（VoIP）サービス レベルはサポートしていません。
- Cisco IOS デバイスだけが宛先 IP SLA Responder の送信元になります。
- 他社製のデバイスに IP SLA Responder を設定することはできません。また、Cisco IOS IP SLA はこれらのデバイス固有のサービスに対してだけ動作パケットを送信できます。

SLA について

Cisco IOS IP サービス レベル契約（SLA）

Cisco IOS IP SLA はネットワークにデータを送信し、複数のネットワーク ロケーション間あるいは複数のネットワーク パス内のパフォーマンスを測定します。Cisco IOS IP SLA は、ネットワーク データおよびIP サービスをシミュレーションし、ネットワーク パフォーマンス情報をリアル タイムで収集します。Cisco IOS IP SLA は、Cisco IOS デバイス間のトラフィックまたは Cisco IOS デバイスからネットワーク アプリケーション サーバのようなリモート IP デバイスへのトラフィックを生成し、分析します。さまざまな Cisco IOS IP SLA 動作で評価を実行し、トラブルシューティング、問題分析、ネットワーク トポロジの設計に使用します。

Cisco IOS IP SLA 動作に応じてシスコ デバイスのネットワーク パフォーマンス統計情報がモニタリングされ、コマンドラインインターフェイス（CLI）MIBおよび簡易ネットワーク管理プロトコル（SNMP）MIB に格納されます。IP SLA パケットには設定可能な IP レイヤおよびアプリケーション層のオプションがあります。たとえば、発信元および宛先 IP アドレス、ユーザ データグラム プロトコル（UDP）/TCP ポート番号、タイプ オブ サービス（ToS）バイト（DiffServ コードポイント（DSCP）およびIP プレフィックス ビットを含む）、VPN ルーティング/転送インスタンス（VRF）、URL Web アドレスなどが設定できます。

Cisco IP SLA はレイヤ 2 転送に依存していないので、異なるネットワーク間にエンドツーエンド動作を設定してエンドユーザが経験しそうなメトリックを最大限に反映させることができます。IP SLA は、次のパフォーマンス メトリックを収集して分析します。

- 遅延（往復および一方向）
- ジッタ（方向性あり）
- パケット損失（方向性あり）
- パケット シーケンス（パケット順序）
- パス（ホップ単位）
- 接続（方向性あり）
- サーバまたは Web サイトのダウンロード時間

Cisco IP SLA は SNMP によるアクセスが可能なので、Cisco Prime Internetwork Performance Monitor（IPM）やサードパーティ製パフォーマンス管理製品などのパフォーマンス モニタリング アプリケーションでも使用できます。

IP SLA を使用すると、次の利点が得られます。

- SLA モニタリング、評価、検証。
- ネットワーク パフォーマンス モニタリング。
    - ネットワークのジッター、遅延、パケット損失の測定。

- 連続的で信頼性のある予測可能な測定。

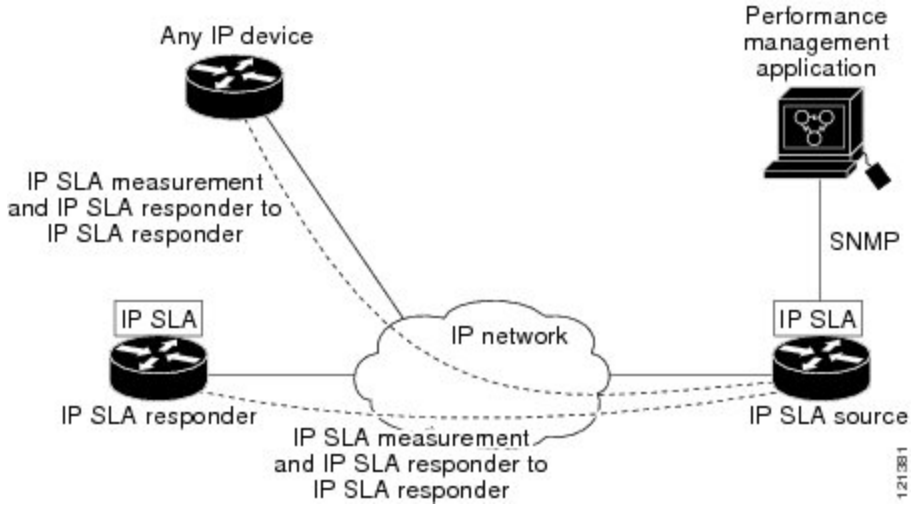
- IP サービス ネットワーク ヘルス アセスメントにより、既存の QoS が新しい IP サービスに適していることを確認できる。
- 端末間のネットワーク アベイラビリティをモニタリングして、ネットワーク リソースをあらかじめ検証し接続をテストできる（たとえば、ビジネス上の重要なデータを保存する NFS サーバのネットワーク アベイラビリティをリモート サイトから確認できる）。
- 問題をすぐに認識し、トラブルシューティングにかかる時間を短縮できる一貫性のある信頼性の高い測定によるネットワーク動作のトラブルシューティング。
- マルチプロトコル ラベル スイッチング（MPLS）パフォーマンス モニタリングとネットワークの検証を行う（デバイスが MPLS をサポートする場合）。

Cisco IOS IP SLA でのネットワーク パフォーマンスの測定

IP SLA を使用して、プローブを物理的に配置せずに、コア、分散、エッジといったネットワーク内の任意のエリア間のパフォーマンスをモニタリングすることができます。2つのネットワーク デバイス間のネットワーク パフォーマンスは、生成トラフィックで測定します。

図 *2 : Cisco IOS IP SLA* 動作

次の図に、送信元デバイスが宛先デバイスに生成パケットを送信するときに IP SLA が開始される手順を示します。宛先デバイスがパケットを受信すると、IP SLA 動作の種類によって、送信元のタイム スタンプ情報に応じてパフォーマンス メトリックを算出します。IP SLA 動作は、特定のプロトコル（UDP など）を使用してネットワークの送信元から宛先へのネットワーク測定を行います。



IP SLA レスポンダおよび IP SLA 制御プロトコル

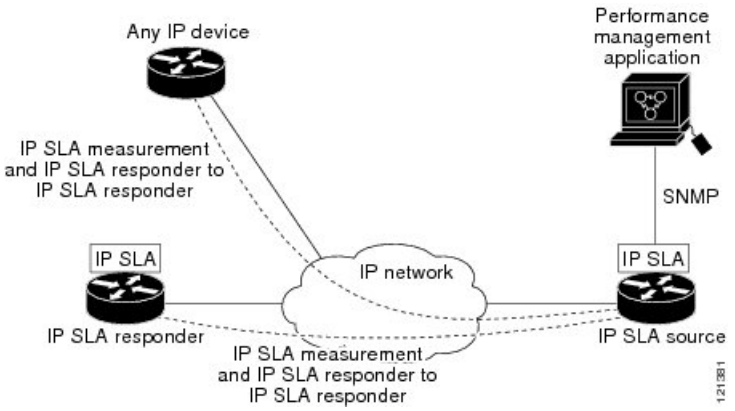
IP SLA レスポンダは宛先 Cisco デバイスに組み込まれたコンポーネントで、システムが IP SLA 要求パケットを予想して応答します。Responder は専用プローブなしで正確な測定を行います。

レスポンダは、受信および応答するポートが通知されるメカニズムを Cisco IOS IP SLA コントロール プロトコルを通じて実現します。

（注） IP SLA レスポンダはレスポンダ設定可能なデバイスである Cisco IOS レイヤ 2 にすることもできます。レスポンダは、IP SLA 機能を全面的にサポートする必要はありません。

次の図は、IP ネットワーク内での Cisco IOS IP SLA レスポンダの配置場所を示します。レスポンダは、IP SLA 動作から送信されたコントロール プロトコル メッセージを指定されたポートで受信します。コントロール メッセージを受信したら、指定された UDP または TCP ポートを指定された時間だけイネーブルにします。この間に、レスポンダは要求を受け付け、応答します。レスポンダは、IP SLA パケットに応答した後または指定の時間が経過したら ポートを無効にします。セキュリティの向上のために、コントロール メッセージでは MD5 認証が利用できます。

図 *3 : Cisco IOS IP SLA* 動作



すべての IP SLA 動作に対して宛先デバイスのレスポンダを有効にする必要はありません。たとえば、宛先ルータが提供しているサービス（Telnet や HTTP など）は Responder では必要ありません。

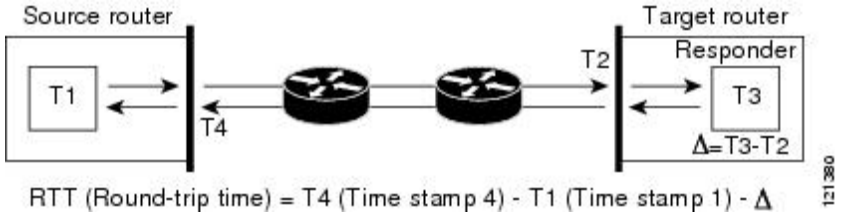
IP SLA の応答時間の計算

スイッチ、コントローラ、ルータは、他の高優先度プロセスがあるために、着信パケットの処理に数十ミリ秒かかることがあります。この遅延により応答時間が影響を受けます。テストパケットの応答が処理待ちのキューに入っていることもあるからです。この場合、応答時間は正しいネットワーク遅延を反映しません。IP SLA はソース デバイスとターゲット デバイス（レスポンダが使用されている場合）の処理遅延を最小化し、正しいラウンドトリップ時間（RTT）を識別します。IP SLA テスト パケットは、タイム スタンプによって処理遅延を最小化します。

IP SLA レスポンダが有効の場合、パケットが割り込みレベルでインターフェイスに着信したときおよびパケットが出て行くときにターゲット デバイスでタイム スタンプを付け、処理時間は含めません。タイム スタンプはサブミリ秒単位で構成されます。

図 *4 : Cisco IOS IP SLA* レスポンダ タイム スタンプ

次の図は、レスポンダの動作を示します。RTT を算出するためのタイム スタンプが 4 つ付けられます。ターゲット ルータでレスポンダ機能がイネーブルの場合、タイム スタンプ 3（TS3）からタイム スタンプ 2（TS2）を引いてテスト パケットの処理にかかった時間を求め、デルタ（Δ）で表します。次に全体の RTT からこのデルタの値を引きます。IP SLA により、この方法はソース ルータにも適用されます。その場合、着信タイム スタンプ 4（TS4）が割り込みレベルで付けられ、より正確な結果を得ることができます。



この他にも、ターゲット デバイスに 2 つのタイム スタンプがあれば一方向遅延、ジッタ、方向性を持つパケット損失がトラッキングできるという利点があります。大半のネットワーク動作は非同期なので、このような統計情報があるのは問題です。ただし一方向遅延測定を取り込むには、ソース ルータとターゲット ルータの両方にネットワーク タイム プロトコル（NTP）を設定し、両方のルータを同じくロック ソースに同期させる必要があります。一方向ジッタ測定にはクロック同期は不要です。

IP SLA 動作のスケジューリング

IP SLA 動作を設定する場合、統計情報の取り込みとエラー情報の収集から開始するように動作をスケジューリングする必要があります。スケジューリングは、すぐに動作を開始する、または特定の月、日、時刻に開始するように設定できます。また、*pending* オプションを使用して、あとで動作を開始するように設定することもできます。pending オプションは動作の内部状態に関するもので、SNMP で表示できます。トリガーを待機する反応（しきい値）動作の場合も pending オプションを使用します。1 度に 1 つの IP SLA 動作をスケジューリングしたり、グループの動作をスケジューリングすることもできます。

Cisco IOS CLI または CISCO RTTMON-MIB で 1 つのコマンドを使用して、複数の IP SLA 動作をスケジューリングできます。等間隔で動作を実行するようにスケジューリングすると、IP SLA モニタリング トラフィックの数を制御できます。IP SLA 動作をこのように分散させるとCPU 使用率を最小限に抑え、ネットワーク スケーラビリティを向上させることができます。

IP SLA 複数動作のスケジューリング機能の詳細については、『*Cisco IOS IP SLA Configuration Guide*』の「IP SLAs—Multiple Operation Scheduling」の章を参照してください。

IP SLA 動作のしきい値のモニタリング

サービス レベル契約モニタリングを正しくサポートするには、違反が発生した場合にすぐに通知されるメカニズムにする必要があります。IP SLA は次のような場合にイベントによってトリガーされる SNMP トラップを送信できます。

- 接続の損失

- Timeout
- RTT しきい値
- 平均ジッタしきい値
- 一方向パケット損失
- 一方向ジッタ
- 一方向平均オピニオン評点（MOS）
- 一方向遅延

IP SLA しきい値違反が発生した場合も、あとで分析するために別の IP SLA 動作がトリガーされます。たとえば、回数を増やしたり、Internet Control Message Protocol（ICMP）パス エコーや ICMP パス ジッター動作を開始してトラブルシューティングを行うことができます。

ICMP Echo

ICMP エコー動作は、シスコ デバイスと IP を使用するその他のデバイス間のエンドツーエンド応答時間を測定します。応答時間は、ICMP エコー要求メッセージを宛先に送信し、ICMP エコー応答を受信するのにかかる時間を測定して算出されます。多くのお客様は、IP SLA ICMP ベース動作、社内 ping テスト、またはこの応答所要時間を測定するために ping ベース専用プローブを使用します。IP SLA ICMP エコー動作は、ICMP ping テストと同じ仕様に準拠しており、どちらの方法でも同じ応答所要時間になります。

UDP ジッター

ジッターとは、パケット間遅延の差異を説明する簡単な用語です。複数のパケットが送信元から宛先まで 10 ミリ秒の間隔で継続的に送信される場合、宛先は 10 ミリ秒間隔で受信します（ネットワークが正常に動作している場合）。しかし、ネットワークに遅延がある場合（キューイングや代替ルートを通じた到着など）、パケットの着信の間隔が 10 ミリ秒を超える場合や 10 ミリ秒未満になる場合があります。正のジッター値は、パケットが 10 ミリ秒を超える間隔で到着することを示します。負のジッター値は、パケットが 10 ミリ秒未満の間隔で到着することを示します。パケットの到着が 12 ミリ秒間隔の場合、正のジッター値は 2 ミリ秒です。8 ミリ秒間隔で到着する場合、負のジッター値は 2 ミリ秒です。遅延による影響を受けやすいネットワークの場合、正のジッタ値は望ましくありません。ジッタ値 0 が理想的です。

ジッターのモニタリング以外にも、IP SLA UDP ジッター動作を多目的データ収集動作に使用できます。IP SLA によって生成されるパケットは、データを送受信するパケットを含めて、送信元および動作ターゲットからシーケンス情報とタイム スタンプを伝送します。このデータに基づいて、UDP ジッター動作は次を測定します。

- 方向別ジッター（送信元から宛先へ、宛先から送信元へ）
- 方向別パケット損失
- 方向別遅延（一方向遅延）
- ラウンドトリップ遅延（平均 RTT）

データを送受信するパスが異なる場合もあるので（非同期）、方向別データを使用すればネットワークで発生している輻輳や他の問題の場所を簡単に突き止めることができます。

UDP ジッタ動作では合成（シミュレーション）UDP トラフィックを生成し、送信元ルータからターゲット ルータに多数の UDP パケットを送信します。その際の各パケットのサイズ、パケット同士の間隔、送信間隔は決められています。デフォルトでは、10 バイトのペイロード サイズのパケット フレームを 10 ミリ秒で 10 個生成し、60 秒間隔で送信します。これらのパラメータは、提供する IP サービスを最適にシミュレートするように設定できます。

一方向遅延を正確に測定する場合、（NTP によって提供される）送信元デバイスとターゲット デバイス間のクロック同期が必要です。一方向ジッタおよびパケット損失を測定する場合は、クロック同期は不要です。送信元デバイスとターゲット デバイスの間でクロックが同期していない場合、一方向ジッターとパケット損失のデータは戻されますが、UDP ジッター動作による一方向遅延測定は 0 の値が戻ります。

IP SLA 動作の設定方法

ここでは、利用可能なすべての動作の設定情報について説明されているわけではありません。設定情報の詳細については『*Cisco IOS IP SLAs Configuration Guide*』を参照してください。ここでは、応答側の設定、UDP ジッター動作の設定（応答側が必要）、ICMP エコー動作の設定（応答側が不要）などの動作例を説明します。他の動作の設定の詳細については、『*Cisco IOS IP SLAs Configuration Guide*』を参照してください。

デフォルト設定

IP SLA 動作は設定されていません。

設定時の注意事項

IP SLA のコマンドについては、『*Cisco IOS IP SLA Command Reference, Release 12.4T*』のコマンド リファレンスを参照してください。

説明と設定手順の詳細については、『*Cisco IOS IP SLAs Configuration Guide, Release 12.4TL*』を参照してください。

ガイドに記載されている IP SLA コマンドまたは動作の中にはデバイスでサポートされないものもあります。デバイスでは、UDP ジッター、UDP エコー、HTTP、TCP 接続、ICMP エコー、ICMP パス エコー、ICMP パス ジッター、FTP、DNS、DHCP を使用する IP サービス レベル分析がサポートされます。また、複数動作スケジューリングおよび事前に設定されたしきい値のモニタリングもサポートされます。ゲートキーパー登録遅延動作測定を使用した Voice over IP（VoIP）サービス レベルはサポートしていません。

IP SLA アプリケーションを設定する前に、**show ip sla application** 特権 EXEC コマンドを使用してソフトウェアイメージで動作タイプがサポートされていることを確認してください。コマンド出力例は次のとおりです。

```
Device# show ip sla application

        IP Service Level Agreements
Version: Round Trip Time MIB 2.2.0, Infrastructure Engine-III

Supported Operation Types:
        icmpEcho, path-echo, path-jitter, udpEcho, tcpConnect, http
        dns, udpJitter, dhcp, ftp, udpApp, wspApp

Supported Features:
        IPSLAs Event Publisher

IP SLAs low memory water mark: 33299323
Estimated system max number of entries: 24389

Estimated number of configurable operations: 24389
Number of Entries configured  : 0
Number of active Entries      : 0
Number of pending Entries     : 0
Number of inactive Entries    : 0
Time of last change in whole IP SLAs: *13:04:37.668 UTC Wed Dec 19 2012
```

IP SLA レスポンダの設定

IP SLA レスポンダは、Cisco IOS ソフトウェアベース デバイスだけで利用可能です。これには、IP SLA 機能をフルにサポートしていない一部のレイヤ 2 デバイスも含まれます。

ターゲット デバイス（動作ターゲット）上の IP SLA 応答側を設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip sla responder** {**tcp-connect** | **udp-echo**} **ipaddress** *ip-address* **port** *port-number* 例： `Device(config)# ip sla responder`	デバイスを IP SLA レスポンダとして設定します。 キーワードの意味は次のとおりです。 • **tcp-connect**：レスポンダの TCP 接続動作をイネーブルにします。

	コマンドまたはアクション	目的
	`udp-echo 172.29.139.134 5000`	• **udp-echo**：レスポンダの User Datagram Protocol（UDP）エコー動作またはジッター動作をイネーブルにします。 • **ipaddress** *ip-address*：宛先 IP アドレスを入力します。 • **port** *port-number*：宛先ポート番号を入力します。 （注） IP アドレスとポート番号は、IP SLA 動作のソースデバイスに設定した IP アドレスおよびポート番号と一致している必要があります。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

IP SLA ネットワーク パフォーマンス測定の実装

デバイス上で IP SLA ネットワーク パフォーマンス測定を実施するには、次の手順を実行します。

始める前に

show ip sla application 特権 EXEC コマンドを使用して、ソフトウェア イメージで目的の動作タイプがサポートされていることを確認してください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip sla** *operation-number* 例： `Device(config)# ip sla 10`	IP SLA 動作を作成し、IP SLA 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	**udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*] 例： `Device(config-ip-sla)# udp-jitter 172.29.139.134 5000`	IP SLA 動作を目的の動作タイプとして設定して（例では UDP ジッター動作が使用されています）、そのコンフィギュレーション モードを開始します（例では UDP ジッター コンフィギュレーション モードが使用されています）。 • *destination-ip-address* | *destination-hostname*：宛先 IP アドレスまたはホスト名を指定します。 • *destination-port*：宛先ポート番号を 1 ～ 65535 の範囲で指定します。 •（任意）**source-ip** {*ip-address* | *hostname*}：送信元 IP アドレスまたはホスト名を指定します。送信元 IP アドレスまたはホスト名が指定されていない場合、IP SLA では、宛先に最も近い IP アドレスが選択されます。 •（任意）**source-port** *port-number*：送信元ポート番号を 1 ～ 65535 の範囲で指定します。ポート番号を

	コマンドまたはアクション	目的
		指定しない場合、IP SLA は利用可能なポートを選択します。 • （任意）**control**：IP SLA 制御メッセージの IP SLA レスポンダへの送信をイネーブルまたはディセーブルにします。デフォルトでは、IP SLA 制御メッセージは宛先デバイスに送信され、IP SLA レスポンダとの接続が確立されます。 • （任意）**num-packets** *number-of-packets*：生成するパケット数を入力します。指定できる範囲は 1 ～ 6000 です。デフォルトは 10 です。 • （任意）**interval** *inter-packet-interval*：パケットの送信間隔をミリ秒で入力します。指定できる範囲は 1 ～ 6000 です。デフォルトは 20 ミリ秒です。
ステップ 5	**frequency** 秒 例： `Device(config-ip-sla-jitter)# frequency 45`	（任意）SLA 動作のオプションを設定します。次の例では、指定された IP SLA 動作が繰り返されるレートを設定します。指定できる範囲は 1 ～ 604800 秒で、デフォルトは 60 秒です。
ステップ 6	**threshold** *milliseconds* 例： `Device(config-ip-sla-jitter)# threshold 200`	（任意）しきい値条件を設定します。次の例では、指定された IP SLA 動作のしきい値が 200 に設定されます。有効な範囲は 0 ～ 60000 ミリ秒です。
ステップ 7	**exit** 例： `Device(config-ip-sla-jitter)# exit`	SLA 動作コンフィギュレーションモード（この例では UDP ジッター コンフィギュレーション モード）を終了し、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 8	**ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** |	個々の IP SLA 動作のスケジューリング パラメータを設定します。

	コマンドまたはアクション	目的
	now | **after** *hh:mm:ss*] [**ageout** *seconds*] [**recurring**] 例： `Device(config)# ip sla schedule 10 start-time now life forever`	• *operation-number*：RTR エントリ番号を入力します。 • （任意）**life**：動作の実行を無制限（**forever**）に指定するか、特定の秒数（*seconds*）を指定します。指定できる範囲は 0 ～ 2147483647 です。デフォルトは 3600 秒（1 時間）です。 • （任意）**start-time**：情報の収集を開始する時刻を入力します。 特定の時刻に開始する場合は、時、分、秒（24 時間表記）、月日を入力します。月を入力しない場合、当月がデフォルト設定です。 **pending** と入力すると、開始時刻を指定するまでは情報を収集しません。 **now** と入力すると、ただちに動作を開始します。 **after** *hh:mm:ss* と入力すれば、指定した時刻の経過後に動作を開始します。 • （任意）**ageout** *seconds*：情報を収集していないとき、メモリの動作を保存する秒数を指定します。指定できる範囲は 0 ～ 2073600 秒です。デフォルトは 0 秒（いつまでも保存する）です。 • （任意）**recurring**：毎日、動作を自動的に実行します。
ステップ 9	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 10	**show running-config** 例：	入力を確認します。

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

UDP ジッター コンフィギュレーション

次に、UDP ジッター IP SLA 動作の設定例を示します。

```
Device(config)# ip sla 10
Device(config-ip-sla)# udp-jitter 172.29.139.134 5000
Device(config-ip-sla-jitter)# frequency 30
Device(config-ip-sla-jitter)# exit
Device(config)# ip sla schedule 5 start-time now life forever
Device(config)# end
Device# show ip sla configuration 10
IP SLAs, Infrastructure Engine-II.

Entry number: 10
Owner:
Tag:
Type of operation to perform: udp-jitter
Target address/Source address: 1.1.1.1/0.0.0.0
Target port/Source port: 2/0
Request size (ARR data portion): 32
Operation timeout (milliseconds): 5000
Packet Interval (milliseconds)/Number of packets: 20/10
Type Of Service parameters: 0x0
Verify data: No
Vrf Name:
Control Packets: enabled
Schedule:
    Operation frequency (seconds): 30
    Next Scheduled Start Time: Pending trigger
    Group Scheduled : FALSE
    Randomly Scheduled : FALSE
    Life (seconds): 3600
    Entry Ageout (seconds): never
    Recurring (Starting Everyday): FALSE
    Status of entry (SNMP RowStatus): notInService
Threshold (milliseconds): 5000
Distribution Statistics:
    Number of statistic hours kept: 2
    Number of statistic distribution buckets kept: 1
    Statistic distribution interval (milliseconds): 20
Enhanced History:
```

UDP ジッター動作を使用した IP サービス レベルの分析

送信元デバイス上の UDP ジッター作を設定するには、次の手順を実行します。

始める前に

送信元デバイス上で UDP ジッター動作を設定するには、ターゲット デバイス（動作ターゲット）で、IP SLA レスポンダをイネーブルにする必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip sla** *operation-number* 例： `Device(config)# ip sla 10`	IP SLA 動作を作成し、IP SLA 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	**udp-jitter** {*destination-ip-address* | *destination-hostname*} *destination-port* [**source-ip** {*ip-address* | *hostname*}] [**source-port** *port-number*] [**control** {**enable** | **disable**}] [**num-packets** *number-of-packets*] [**interval** *interpacket-interval*] 例： `Device(config-ip-sla)# udp-jitter 172.29.139.134 5000`	IP SLA 動作を UDP ジッター動作として設定し、UDP ジッター コンフィギュレーション モードを開始します。 • *destination-ip-address* | *destination-hostname*：宛先 IP アドレスまたはホスト名を指定します。 • *destination-port*：宛先ポート番号を 1 ～ 65535 の範囲で指定します。 •（任意）**source-ip** {*ip-address* | *hostname*}：送信元 IP アドレスまたはホスト名を指定します。送信元IPアドレスまたはホスト名が指定されていない場合、IP SLA で

	コマンドまたはアクション	目的
		は、宛先に最も近いIPアドレスが選択されます。 • （任意）**source-port** *port-number*：送信元ポート番号を 1 ～ 65535 の範囲で指定します。ポート番号を指定しない場合、IP SLA は利用可能なポートを選択します。 • （任意）**control**：IP SLA 制御メッセージの IP SLA レスポンダへの送信をイネーブルまたはディセーブルにします。デフォルトでは、IP SLA 制御メッセージは宛先デバイスに送信され、IP SLA レスポンダとの接続が確立されます。 • （任意）**num-packets** *number-of-packets*：生成するパケット数を入力します。指定できる範囲は 1 ～ 6000 です。デフォルトは 10 です。 • （任意）**interval** *inter-packet-interval*：パケットの送信間隔をミリ秒で入力します。指定できる範囲は 1 ～ 6000 です。デフォルトは 20 ミリ秒です。
ステップ 5	**frequency** 秒 例： `Device(config-ip-sla-jitter)# frequency 45`	（任意）指定した IP SLA 動作を繰り返す間隔を設定します。指定できる範囲は 1 ～ 604800 秒で、デフォルトは 60 秒です。
ステップ 6	**exit** 例： `Device(config-ip-sla-jitter)# exit`	UDP ジッター コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*] [**ageout** *seconds*] [**recurring**]	個々の IP SLA 動作のスケジューリング パラメータを設定します。 • *operation-number*：RTR エントリ番号を入力します。

	コマンドまたはアクション	目的
	例： `Device(config)# ip sla schedule 10 start-time now life forever`	• （任意）**life**：動作の実行を無制限（**forever**）に指定するか、特定の秒数（*seconds*）を指定します。指定できる範囲は 0 ～ 2147483647 です。デフォルトは 3600 秒（1 時間）です。 • （任意）**start-time**：情報の収集を開始する時刻を入力します。 特定の時刻に開始する場合は、時、分、秒（24 時間表記）、月日を入力します。月を入力しない場合、当月がデフォルト設定です。 **pending** と入力すると、開始時刻を指定するまでは情報を収集しません。 **now** と入力すると、ただちに動作を開始します。 **after** *hh:mm:ss* と入力すれば、指定した時刻の経過後に動作を開始します。 • （任意）**ageout** *seconds*：情報を収集していないとき、メモリの動作を保存する秒数を指定します。指定できる範囲は 0 ～ 2073600 秒です。デフォルトは 0 秒（いつまでも保存する）です。 • （任意）**recurring**：毎日、動作を自動的に実行します。
ステップ 8	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 9	**show running-config** 例： `Device# show running-config`	入力を確認します。

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

UDP ジッター IP SLA 動作の設定

次に、UDP ジッター IP SLA 動作の設定例を示します。

```
Device(config)# ip sla 10
Device(config-ip-sla)# udp-jitter 172.29.139.134 5000
Device(config-ip-sla-jitter)# frequency 30
Device(config-ip-sla-jitter)# exit
Device(config)# ip sla schedule 5 start-time now life forever
Device(config)# end
Device# show ip sla configuration 10
IP SLAs, Infrastructure Engine-II.

Entry number: 10
Owner:
Tag:
Type of operation to perform: udp-jitter
Target address/Source address: 1.1.1.1/0.0.0.0
Target port/Source port: 2/0
Request size (ARR data portion): 32
Operation timeout (milliseconds): 5000
Packet Interval (milliseconds)/Number of packets: 20/10
Type Of Service parameters: 0x0
Verify data: No
Vrf Name:
Control Packets: enabled
Schedule:
    Operation frequency (seconds): 30
    Next Scheduled Start Time: Pending trigger
    Group Scheduled : FALSE
    Randomly Scheduled : FALSE
    Life (seconds): 3600
    Entry Ageout (seconds): never
    Recurring (Starting Everyday): FALSE
    Status of entry (SNMP RowStatus): notInService
Threshold (milliseconds): 5000
Distribution Statistics:
    Number of statistic hours kept: 2
    Number of statistic distribution buckets kept: 1
    Statistic distribution interval (milliseconds): 20
Enhanced History:
```

ICMP エコー動作を使用した IP サービス レベルの分析

送信元デバイス上の ICMP エコー動作を設定するには、次の手順を実行します。

始める前に

この動作では、IP SLA レスポンダ側を有効にしておく必要はありません。

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 モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**ip sla** *operation-number* 例： `Device(config)# ip sla 10`	IP SLA 動作を作成し、IP SLA 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	**icmp-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*} | **source-interface** *interface-id*] 例： `Device(config-ip-sla)# icmp-echo 172.29.139.134`	IP SLA 動作を ICMP エコー動作として設定し、ICMP エコー コンフィギュレーション モードを開始します。 • *destination-ip-address* | *destination-hostname*：宛先 IP アドレスまたはホスト名を指定します。 •（任意）**source-ip** {*ip-address* | *hostname*}：送信元 IP アドレスまたはホスト名を指定します。送信元 IP アドレスまたはホスト名が指定されていない場合、IP SLA では、宛先に最も近い IP アドレスが選択されます。 •（任意）**source-interface** *interface-id*：動作に対する送信元インターフェイスを指定します。
ステップ 5	**frequency** 秒 例：	（任意）指定した IP SLA 動作を繰り返す間隔を設定します。指定できる範囲

	コマンドまたはアクション	目的
	`Device(config-ip-sla-echo)# frequency 30`	は 1 ～ 604800 秒で、デフォルトは 60 秒です。
ステップ 6	**exit** 例： `Device(config-ip-sla-echo)# exit`	UDP エコー コンフィギュレーション モードを終了します。続いて、グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh:mm* [*:ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh:mm:ss*] [**ageout** *seconds*] [**recurring**] 例： `Device(config)# ip sla schedule 5 start-time now life forever`	個々の IP SLA 動作のスケジューリング パラメータを設定します。 • *operation-number*：RTR エントリ番号を入力します。 • （任意）**life**：動作の実行を無制限（**forever**）に指定するか、特定の秒数（*seconds*）を指定します。指定できる範囲は 0 ～ 2147483647 です。デフォルトは 3600 秒（1 時間）です。 • （任意）**start-time**：情報の収集を開始する時刻を入力します。 特定の時刻に開始する場合は、時、分、秒（24 時間表記）、月日を入力します。月を入力しない場合、当月がデフォルト設定です。 **pending** と入力すると、開始時刻を指定するまでは情報を収集しません。 **now** と入力すると、ただちに動作を開始します。 **after** *hh:mm:ss* と入力すれば、指定した時刻の経過後に動作を開始します。 • （任意）**ageout** *seconds*：情報を収集していないとき、メモリの動作を保存する秒数を指定します。指定できる範囲は 0 ～ 2073600 秒です。デフォルトは 0 秒（いつまでも保存する）です。

	コマンドまたはアクション	目的
		• （任意）**recurring**：毎日、動作を自動的に実行します。
ステップ 8	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 9	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 10	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーションファイルに設定を保存します。

ICMP エコー IP SLA 動作の設定

次に、ICMP エコー IP SLA 動作の設定例を示します。

```
Device(config)# ip sla 12
Device(config-ip-sla)# icmp-echo 172.29.139.134
Device(config-ip-sla-echo)# frequency 30
Device(config-ip-sla-echo)# exit
Device(config)# ip sla schedule 5 start-time now life forever
Device(config)# end
Device# show ip sla configuration 22
IP SLAs, Infrastructure Engine-II.

Entry number: 12
Owner:
Tag:
Type of operation to perform: echo
Target address: 2.2.2.2
Source address: 0.0.0.0
Request size (ARR data portion): 28
Operation timeout (milliseconds): 5000
Type Of Service parameters: 0x0
Verify data: No
Vrf Name:
Schedule:
    Operation frequency (seconds): 60
    Next Scheduled Start Time: Pending trigger
    Group Scheduled : FALSE
    Randomly Scheduled : FALSE
    Life (seconds): 3600
```

```
    Entry Ageout (seconds): never
    Recurring (Starting Everyday): FALSE
    Status of entry (SNMP RowStatus): notInService
Threshold (milliseconds): 5000
Distribution Statistics:
    Number of statistic hours kept: 2
    Number of statistic distribution buckets kept: 1
    Statistic distribution interval (milliseconds): 20
History Statistics:
    Number of history Lives kept: 0
    Number of history Buckets kept: 15
    History Filter Type: None
Enhanced History:
```

IP SLA 動作のモニタリング

次の表で、IP SLA 動作の設定と結果を表示するために使用するコマンドについて説明します。

表 *5 : IP SLA* 動作のモニタリング

show ip sla application	Cisco IOS IP SLA のグローバル情報を表示します。
show ip sla authentication	IP SLA 認証情報を表示します。
show ip sla configuration [*entry-number*]	すべての IP SLA 動作または特定の IP SLA 動作に関する、デフォルト値をすべて含めた設定値を表示します。
show ip sla enhanced-history {**collection-statistics** | **distribution statistics**} [*entry-number*]	収集した履歴バケットの拡張履歴統計情報、あるいはすべての IP SLA 動作または特定の IP SLA 動作に関する分散統計情報を表示します。
show ip sla ethernet-monitor configuration [*entry-number*]	IP SLA 自動イーサネット設定を表示します。
show ip sla group schedule [*schedule-entry-number*]	IP SLA グループ スケジューリング設定と個別情報を表示します。
show ip sla history [*entry-number* | **full** | **tabular**]	すべての IP SLA 動作について収集した履歴を表示します。
show ip sla mpls-lsp-monitor {**collection-statistics** | **configuration** | **ldp operational-state** | **scan-queue** | **summary** [*entry-number*] | **neighbors**}	MPLS ラベル スイッチド パス（LSP）ヘルス モニタ動作を表示します。
show ip sla reaction-configuration [*entry-number*]	すべての IP SLA 動作または特定の IP SLA 動作に関する、予防的しきい値のモニタリングの設定を表示します。

show ip sla reaction-trigger [*entry-number*]	すべての IP SLA 動作または特定の IP SLA 動作に関する反応トリガー情報を表示します。
show ip sla responder	IP SLA レスポンダ側の情報を表示します。
show ip sla statistics [*entry-number* | **aggregated** | **details**]	動作ステータスおよび統計情報の現在値または合計値を表示します。

IP SLA 動作のモニタリングの例

次の例は、アプリケーションごとのすべての IP SLA を示しています。

```
Device# show ip sla application

        IP Service Level Agreements
Version: Round Trip Time MIB 2.2.0, Infrastructure Engine-III

Supported Operation Types:
        icmpEcho, path-echo, path-jitter, udpEcho, tcpConnect, http
        dns, udpJitter, dhcp, ftp, udpApp, wspApp

Supported Features:
        IPSLAs Event Publisher

IP SLAs low memory water mark: 33299323
Estimated system max number of entries: 24389

Estimated number of configurable operations: 24389
Number of Entries configured  : 0
Number of active Entries      : 0
Number of pending Entries     : 0
Number of inactive Entries    : 0
Time of last change in whole IP SLAs: *13:04:37.668 UTC Wed Dec 19 2012
```

次の例は、すべての IP SLA ディストリビューション統計情報を示しています。

```
Device# show ip sla enhanced-history distribution-statistics

Point by point Enhanced History
Entry    = Entry Number
Int      = Aggregation Interval
BucI     = Bucket Index
StartT   = Aggregation Start Time
Pth      = Path index
Hop      = Hop in path index
Comps    = Operations completed
OvrTh    = Operations completed over thresholds
SumCmp   = Sum of RTT (milliseconds)
SumCmp2L = Sum of RTT squared low 32 bits (milliseconds)
SumCmp2H = Sum of RTT squared high 32 bits (milliseconds)
TMax     = RTT maximum (milliseconds)
TMin     = RTT minimum (milliseconds)

Entry Int BucI StartT     Pth Hop Comps OvrTh SumCmp     SumCmp2L   SumCmp2H   T
Max    TMin
```

その他の参考資料

関連資料

関連項目	参照先
Cisco Medianet Metadata Guide	http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/mdata/configuration/15-sy/mdata-15sy-book/metadata-framework.pdf
Cisco Media Services Proxy Configuration Guide	http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/msp/configuration/15-mt/msp-15-mt-book.pdf
Cisco Mediatrace and Cisco Performance Monitor Configuration Guide	http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/media_monitoring/configuration/15-mt/mm-15-mt-book/mm-mediatrace.html

エラー メッセージ デコーダ

説明	リンク
このリリースのシステムエラーメッセージを調査し解決するために、エラー メッセージ デコーダ ツールを使用します。	https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi

標準および RFC

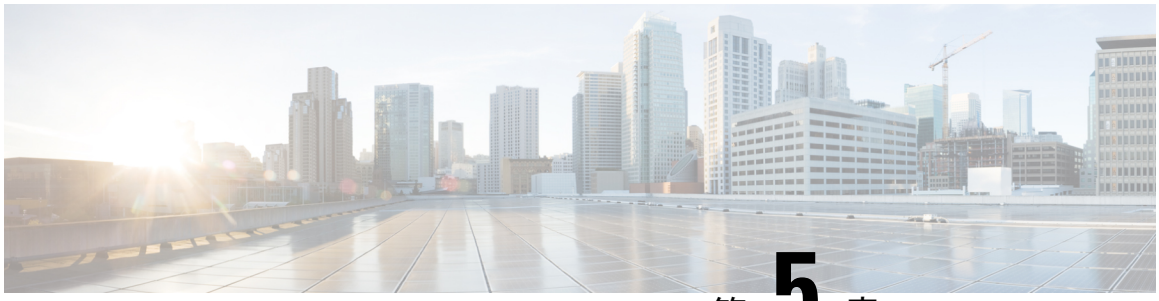
標準/RFC	役職（Title）
なし	-

MIB

MIB	MIB リンク
本リリースでサポートするすべての MIB	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

テクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Product Alert Tool（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support



第 5 章

SPAN および RSPAN の設定

SPAN および RSPAN の前提条件

SPAN

- SPAN トラフィックを特定の VLAN に制限するには、**filter vlan** キーワードを使用します。トランク ポートをモニタしている場合、このキーワードで指定された VLAN 上のトラフィックのみがモニタされます。デフォルトでは、トランク ポート上のすべての VLAN がモニタされます。

RSPAN

- RSPAN VLAN を設定してから、RSPAN 送信元または宛先セッションを設定することを推奨します。

SPAN および RSPAN の制約事項

SPAN

SPAN の制約事項は次のとおりです。

- 各 デバイス で 66 のセッションを設定できます。最大 8 つの送信元セッションを設定できます。残りのセッションは、RSPAN宛先セッションとして設定できます。送信元セッショ

ンは、ローカル SPAN セッションまたは RSPAN 送信元セッションのどちらかになります。

- SPAN 送信元の場合は、セッションごとに、単一のポートまたは VLAN、一連のポートまたは VLAN、一定範囲のポートまたは VLAN のトラフィックをモニタできます。1 つの SPAN セッションに、送信元ポートおよび送信元 VLAN を混在させることはできません。
- 宛先ポートを送信元ポートにすることはできません。同様に、送信元ポートを宛先ポートにすることもできません。
- 同じ宛先ポートで 2 つの SPAN セッションを設定することはできません。
- デバイス ポートを SPAN 宛先ポートとして設定すると、通常のデバイス ポートではなくなります。SPAN 宛先ポートを通過するのは、監視対象トラフィックのみになります。
- SPAN コンフィギュレーション コマンドを入力しても、前に設定した SPAN パラメータは削除されません。設定されている SPAN パラメータを削除するには、**no monitor session** {*session_number* | **all** | **local** | **remote**} グローバル 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。
- ローカル SPAN では、**encapsulation replicate** キーワードが指定されている場合、SPAN 宛先ポートを経由する発信パケットは元のカプセル化ヘッダー（タグなし、ISL、または IEEE 802.1Q）を伝送します。このキーワードが指定されていない場合、パケットはネイティブ形式で送信されます。
- ディセーブルのポートを送信元ポートまたは宛先ポートとして設定することはできますが、SPAN 機能が開始されるのは、宛先ポートと少なくとも 1 つの送信元ポートまたは送信元 VLAN がイネーブルになってからです。
- 単一の SPAN セッションに、送信元 VLAN とフィルタ VLAN を混在させることはできません。

SPAN セッションのトラフィック監視には次の制約事項があります。

- ポートまたは VLAN を送信元にできますが、同じセッション内に送信元ポートと送信元 VLAN を混在させることはできません。
- Wireshark は、出力スパンがアクティブな場合は出力パケットをキャプチャしません。
- 同じデバイスまたはデバイス スタック内で、ローカル SPAN と RSPAN の送信元セッションの両方を実行できます。デバイスまたはデバイス スタックは合計 66 の送信元および RSPAN 宛先セッションをサポートします。
- 別個のまたは重複する SPAN 送信元ポートと VLAN のセットによって、SPAN または RSPAN 送信元セッションを 2 つ個別に設定できます。スイッチド ポートおよびルーテッド ポートはいずれも SPAN 送信元および宛先として設定できます。
- 1 つの SPAN セッションに複数の宛先ポートを設定できますが、1 つのデバイス スタックあたりに設定できる宛先ポートは最大で 64 個です。

- SPAN セッションがデバイスの通常の動作を妨げることはありません。ただし、10 Mbps のポートで 100 Mbps のポートをモニタするなど、オーバーサブスクライブの SPAN 宛先は、パケットのドロップまたは消失を招くことがあります。
- SPAN または RSPAN がイネーブルの場合、監視中の各パケットは 2 回送信されます（1 回は標準トラフィックとして、もう 1 回は監視されたパケットとして）。多数のポートまたは VLAN を監視すると、大量のネットワーク トラフィックが生成されることがあります。
- ディセーブルのポート上に SPAN セッションを設定することはできますが、そのセッション用に宛先ポートと少なくとも 1 つの送信元ポートまたは VLAN をイネーブルにしない限り、SPAN セッションはアクティブになりません。
- デバイスは、単一セッション内でのローカル SPAN と RSPAN の併用をサポートしません。
  - RSPAN 送信元セッションにローカル宛先ポートを設定できません。
  - RSPAN 宛先セッションにローカル送信元ポートを設定できません。
  - 同じデバイスまたはデバイス スタック上で、同じ RSPAN VLAN を使用する RSPAN 宛先セッションおよび RSPAN 送信元セッションを実行できません。

RSPAN

RSPAN の制約事項は次のとおりです。

- RSPAN は、BPDU パケット監視または他のレイヤ 2 デバイス プロトコルをサポートしません。
- RSPAN VLAN はトランク ポートにのみ設定されており、アクセス ポートには設定されていません。不要なトラフィックが RSPAN VLAN に発生しないようにするために、参加しているすべてのデバイスで VLAN RSPAN 機能がサポートされていることを確認してください。
- 送信元トランク ポートにアクティブな RSPAN VLAN が設定されている場合、RSPAN VLAN はポートベース RSPAN セッションの送信元として含まれます。また、RSPAN VLAN を SPAN セッションの送信元に設定することもできます。ただし、デバイスはスパンされたトラフィックを監視しないため、デバイスの RSPAN 送信元セッションの宛先として識別された RSPAN VLAN では、パケットの出力スパニングがサポートされません。
- VTP および VTP プルーニングをイネーブルにすると、トランク内で RSPAN トラフィックがプルーニングされ、1005 以下の VLAN ID に関して、ネットワークで不必要な RSPAN トラフィックのフラッディングが防止されます。

SPAN および RSPAN について

SPAN および RSPAN

ポートまたは VLAN を通過するネットワーク トラフィックを解析するには、SPAN または RSPAN を使用して、そのデバイス上、またはネットワーク アナライザやその他のモニタ デバイス、あるいはセキュリティデバイスに接続されている別のデバイス上のポートにトラフィックのコピーを送信します。SPAN は送信元ポート上または送信元 VLAN 上で受信、送信、または送受信されたトラフィックを宛先ポートにコピー（ミラーリング）して、解析します。SPAN は送信元ポートまたは VLAN 上のネットワーク トラフィックのスイッチングには影響しません。宛先ポートは SPAN 専用にする必要があります。SPAN または RSPAN セッションに必要なトラフィック以外、宛先ポートがトラフィックを受信したり転送したりすることはありません。

SPAN を使用してモニタできるのは、送信元ポートを出入りするトラフィックまたは送信元 VLAN に出入りするトラフィックだけです。送信元 VLAN にルーティングされたトラフィックはモニタできません。たとえば、着信トラフィックをモニタしている場合、別の VLAN から送信元 VLAN にルーティングされているトラフィックはモニタできません。ただし、送信元 VLAN で受信し、別の VLAN にルーティングされるトラフィックは、モニタできます。

ネットワーク セキュリティ デバイスからトラフィックを注入する場合、SPAN または RSPAN 宛先ポートを使用できます。たとえば、Cisco 侵入検知システム（IDS）センサー装置を宛先ポートに接続した場合、IDS デバイスは TCP リセット パケットを送信して、疑わしい攻撃者の TCP セッションを停止させることができます。

ローカル SPAN

ローカル SPAN は 1 つのデバイス内の SPAN セッション全体をサポートします。すべての送信元ポートまたは送信元 VLAN、および宛先ポートは、同じデバイスまたはデバイス スタック内にあります。ローカル SPAN は、任意の VLAN 上の 1 つまたは複数の送信元ポートからのトラフィック、あるいは 1 つまたは複数の VLAN からのトラフィックを解析するために宛先ポートへコピーします。

図 *5:* 単一デバイスでのローカル *SPAN* の設定例

ポート 5（送信元ポート）上のすべてのトラフィックがポート 10（宛先ポート）にミラーリングされます。ポート 10 のネットワーク アナライザは、ポート 5 に物理的には接続されていま

せんが、ポート 5 からのすべてのネットワーク トラフィックを受信します。

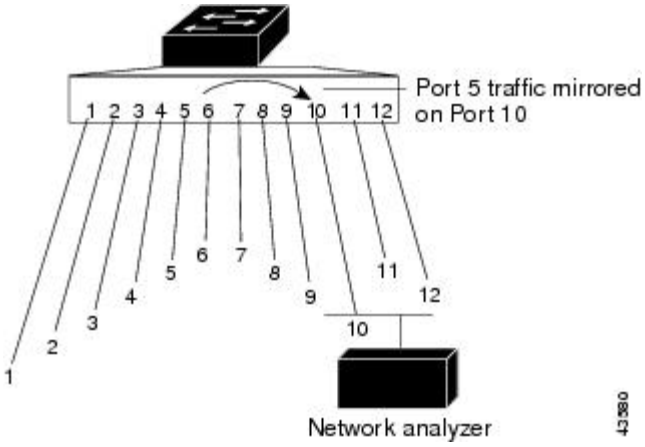
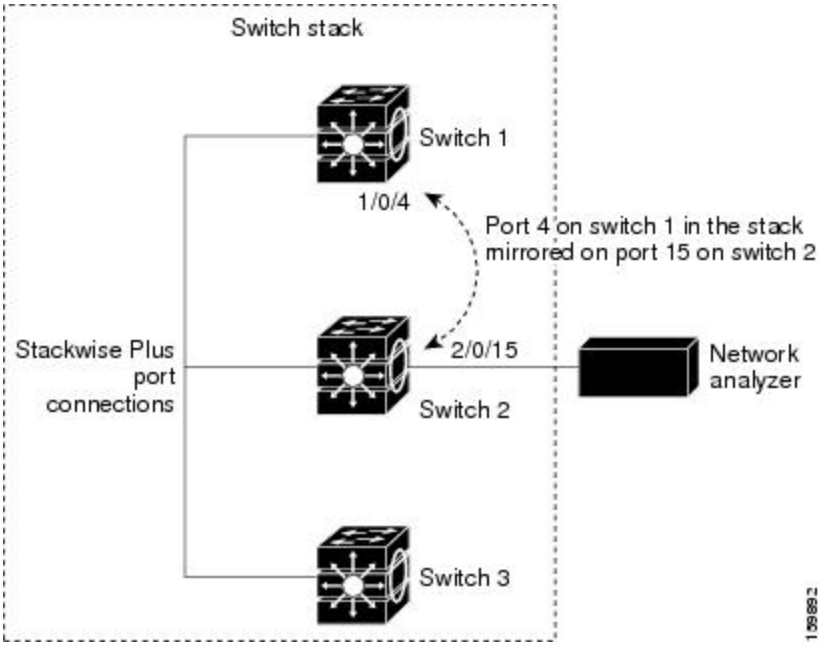


図 *6 :* デバイス スタックでのローカル *SPAN* の設定例

これは、デバイス スタック内のローカル SPAN の例です。送信元ポートと宛先ポートは異なるスタック メンバにあります。



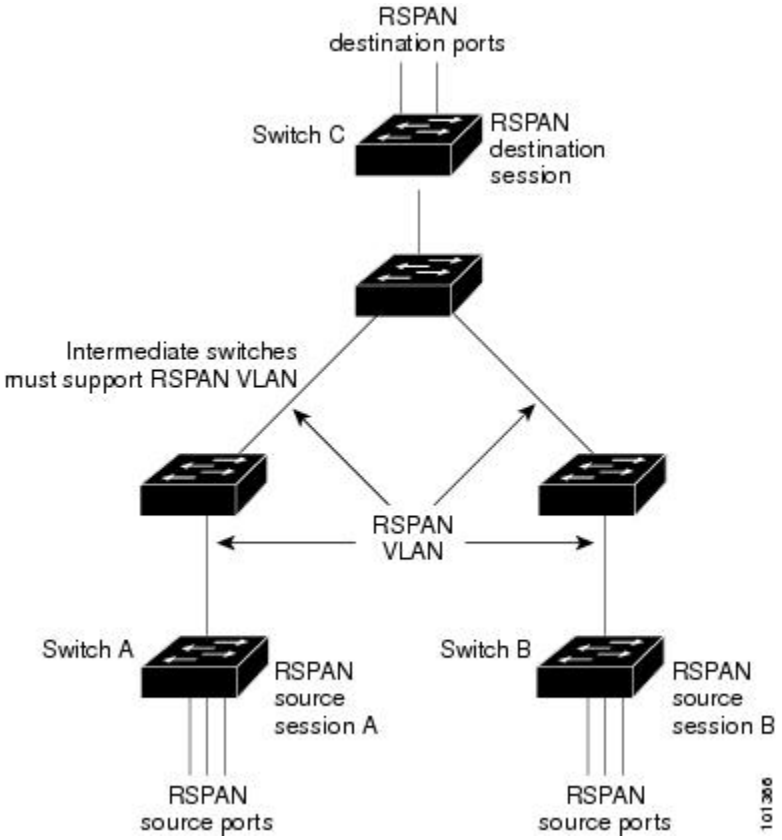
リモート SPAN

RSPAN は、異なるデバイス（または異なるデバイス スタック）上の送信元ポート、送信元 VLAN、および宛先ポートをサポートしているので、ネットワーク上で複数のデバイスをリモート監視できます。

図 *7 : RSPAN* の設定例

下の図にデバイス A とデバイス B の送信元ポートを示します。各 RSPAN セッションのトラフィックは、ユーザが指定した RSPAN VLAN 上で伝送されます。この RSPAN VLAN は、参

加しているすべてのデバイスの RSPAN セッション専用です。送信元ポートまたは VLAN からの RSPAN トラフィックは RSPAN VLAN にコピーされ、RSPAN VLAN を伝送するトランクポートを介して、RSPAN VLAN をモニタする宛先セッションに転送されます。各 RSPAN 送信元デバイスには、ポートまたは VLAN のいずれかが RSPAN 送信元として必要です。図中のデバイス C のように、宛先は常に物理ポートになります。



SPAN と RSPAN の概念および用語

- SPAN セッション
- モニタ対象トラフィック
- 送信元ポート
- 送信元 VLAN
- VLAN フィルタリング
- 接続先ポート
- RSPAN VLAN

SPAN セッション

SPAN セッション（ローカルまたはリモート）を使用すると、1 つまたは複数のポート上、あるいは 1 つまたは複数の VLAN 上でトラフィックをモニタし、そのモニタしたトラフィックを 1 つまたは複数の宛先ポートに送信できます。

ローカル SPAN セッションは、宛先ポートと送信元ポートまたは送信元 VLAN（すべて単一のネットワーク デバイス上にある）を結び付けたものです。ローカル SPAN には、個別の送信元および宛先のセッションはありません。ローカル SPAN セッションはユーザが指定した入力および出力のパケット セットを収集し、SPAN データ ストリームを形成して、宛先ポートに転送します。

RSPAN は少なくとも 1 つの RSPAN 送信元セッション、1 つの RSPAN VLAN、および少なくとも 1 つの RSPAN 宛先セッションで構成されています。RSPAN 送信元セッションと RSPAN 宛先セッションは、異なるネットワーク デバイス上に別々に設定します。デバイスに RSPAN 送信元セッションを設定するには、一連の送信元ポートまたは送信元 VLAN を RSPAN VLAN に関連付けます。このセッションの出力は、RSPAN VLAN に送信される SPAN パケットのストリームです。別のデバイスに RSPAN 宛先セッションを設定するには、宛先ポートを RSPAN VLAN に関連付けます。宛先セッションは RSPAN VLAN トラフィックをすべて収集し、RSPAN 宛先ポートに送信します。

RSPAN 送信元セッションは、パケット ストリームが転送される点を除き、ローカル SPAN セッションに非常に似ています。RSPAN 送信元セッションでは、SPAN パケットに RSPAN VLAN ID ラベルが再設定され、通常のトランク ポートを介して宛先デバイスに転送されます。

RSPAN 宛先セッションは RSPAN VLAN 上で受信されたすべてのパケットを取得し、VLAN のタギングを除去し、宛先ポートに送ります。セッションは、（レイヤ 2 制御パケットを除く）すべての RSPAN VLAN パケットのコピーを分析のためにユーザに提供します。

SPAN セッションでのトラフィックのモニタには、次のような制約があります。

- ポートまたは VLAN を送信元にできますが、同じセッション内に送信元ポートと送信元 VLAN を混在させることはできません。
- 同じデバイスまたはデバイス スタック内で、ローカル SPAN と RSPAN の送信元セッションの両方を実行できます。デバイスまたはデバイス スタックは合計 66 の送信元および RSPAN 宛先セッションをサポートします。
- 別個のまたは重複する SPAN 送信元ポートと VLAN のセットによって、SPAN または RSPAN 送信元セッションを 2 つ個別に設定できます。スイッチド ポートおよびルーテッド ポートはいずれも SPAN 送信元および宛先として設定できます。
- 1 つの SPAN セッションに複数の宛先ポートを設定できますが、1 つのデバイス スタックあたりに設定できる宛先ポートは最大で 64 個です。
- SPAN セッションがデバイスの通常の動作を妨げることはありません。ただし、10 Mbps のポートで 100 Mbps のポートをモニタするなど、オーバーサブスクライブの SPAN 宛先は、パケットのドロップまたは消失を招くことがあります。
- SPAN または RSPAN がイネーブルの場合、監視中の各パケットは 2 回送信されます（1 回は標準トラフィックとして、もう 1 回は監視されたパケットとして）。したがって、多

数のポートまたは VLAN をモニタすると、大量のネットワーク トラフィックが生成されることがあります。

- ディセーブルのポート上に SPAN セッションを設定することはできますが、そのセッション用に宛先ポートと少なくとも 1 つの送信元ポートまたは VLAN をイネーブルにしない限り、SPAN セッションはアクティブになりません。
- デバイスは、単一セッション内でのローカル SPAN と RSPAN の併用をサポートしません。
  - RSPAN 送信元セッションにローカル宛先ポートを設定できません。
  - RSPAN 宛先セッションにローカル送信元ポートを設定できません。
  - 同じデバイスまたはデバイス スタック上で、同じ RSPAN VLAN を使用する RSPAN 宛先セッションおよび RSPAN 送信元セッションを実行できません。

モニタ対象トラフィック

SPAN セッションは、次のトラフィック タイプを監視できます。

- 受信（Rx）SPAN：受信（または入力）SPAN は、デバイスが変更または処理を行う前に、送信元インターフェイスまたは VLAN が受信したすべてのパケットをできるだけ多くモニタリングします。送信元が受信した各パケットのコピーがその SPAN セッションに対応する宛先ポートに送られます。

  Diffserv コード ポイント（DSCP）の変更など、ルーティングや Quality of Service（QoS）が原因で変更されたパケットは、変更される前にコピーされます。

  受信処理中にパケットをドロップする可能性のある機能は、入力 SPAN には影響を与えません。宛先ポートは、実際の着信パケットがドロップされた場合でも、パケットのコピーを受信します。パケットをドロップする可能性のある機能は、標準および拡張 IP 入力アクセス コントロール リスト（ACL）、入力 QoS ポリシング、VLAN ACL、および出力 QoS ポリシングです。

- 送信（Tx）SPAN：送信（または出力）SPAN は、デバイスによる変更または処理がすべて実行されたあとに、送信元インターフェイスから送信されたすべてのパケットをできる限り多くモニタリングします。送信元が送信した各パケットのコピーがその SPAN セッションに対応する宛先ポートに送られます。コピーはパケットの変更後に用意されます。

  ルーティングが原因で変更されたパケット（存続可能時間（TTL）、MAC アドレス、QoS 値の変更など）は、宛先ポートで（変更されて）コピーされます。

  送信処理中にパケットをドロップする可能性のある機能は、SPAN 用の複製コピーにも影響します。これらの機能には、標準および拡張 IP 出力 ACL、出力 QoS ポリシングがあります。

- 両方：SPAN セッションで、受信パケットと送信パケットの両方について、ポートまたは VLAN をモニタすることもできます。これがデフォルトです。

ローカル SPAN セッション ポートのデフォルト設定では、すべてのタグなしパケットが送信されます。ただし、宛先ポートを設定するときに **encapsulation replicate** キーワードを入力すると、次の変更が発生します。

- 送信元ポートと同じカプセル化設定（タグなし、またはIEEE 802.1Q）を使用して、パケットが宛先ポートに送信されます。
- BPDU やレイヤ 2 プロトコル パケットを含むすべてのタイプのパケットがモニタされます。

したがって、カプセル化レプリケーションがイネーブルにされたローカルSPANセッションでは、タグなし、および IEEE 802.1Q タグ付きパケットが宛先ポートに混在することがあります。

デバイスの輻輳により、入力送信元ポート、出力送信元ポート、またはSPAN宛先ポートでパケットがドロップされることがあります。一般に、これらの特性は互いに無関係です。次に例を示します。

- パケットは通常どおり転送されますが、SPAN 宛先ポートのオーバーサブスクライブが原因でモニタされないことがあります。
- 入力パケットが標準転送されないにもかかわらず、SPAN 宛先ポートに着信することがあります。
- デバイスの輻輳が原因でドロップされた出力パケットは、出力 SPAN からもドロップされます。

SPAN の設定によっては、同一送信元のパケットのコピーが複数、SPAN 宛先ポートに送信されます。たとえば、ポート A での RX モニタ用とポート B での TX モニタ用に双方向（RX と TX）SPAN セッションが設定されているとします。パケットがポート A からデバイスに入ってポートBにスイッチされると、着信パケットも発信パケットも宛先ポートに送信されます。このため、両方のパケットは同じものになります。レイヤ3書き換えが行われた場合には、パケット変更のため異なるパケットになります。

送信元ポート

送信元ポート（別名モニタ側ポート）は、ネットワーク トラフィック分析のために監視するスイッチド ポートまたはルーテッド ポートです。

1 つのローカル SPAN セッションまたは RSPAN 送信元セッションでは、送信元ポートまたは VLAN のトラフィックを単一方向または双方向でモニタできます。

デバイスは、任意の数の送信元ポート（デバイスで利用可能なポートの最大数まで）と任意の数の送信元 VLAN（サポートされている VLAN の最大数まで）をサポートしています。

ただし、デバイスが送信元ポートまたはVLANでサポートするセッション数には上限（2つ）（ローカルまたは RSPAN）があります。単一のセッションにポートおよび VLAN を混在させることはできません。

送信元ポートの特性は、次のとおりです。

- 複数の SPAN セッションでモニタできます。

- モニタする方向（入力、出力、または両方）を指定して、各送信元ポートを設定できます。
- すべてのポート タイプ（EtherChannel、ギガビット イーサネットなど）が可能です。
- EtherChannel 送信元の場合は、EtherChannel 全体で、または物理ポートがポート チャネルに含まれている場合は物理ポート上で個別に、トラフィックをモニタできます。
- アクセス ポート、トランク ポート、ルーテッド ポート、または音声 VLAN ポートに指定できます。
- 宛先ポートにすることはできません。
- 送信元ポートは同じ VLAN にあっても異なる VLAN にあってもかまいません。
- 単一セッション内で複数の送信元ポートをモニタすることが可能です。

送信元 VLAN

VLAN ベースの SPAN（VSPAN）では、1 つまたは複数の VLAN のネットワーク トラフィックをモニタできます。VSPAN 内の SPAN または RSPAN 送信元インターフェイスが VLAN ID となり、トラフィックはその VLAN のすべてのポートでモニタされます。

VSPAN には次の特性があります。

- 送信元 VLAN 内のすべてのアクティブ ポートは送信元ポートとして含まれ、単一方向または双方向でモニタできます。
- 指定されたポートでは、モニタ対象の VLAN 上のトラフィックのみが宛先ポートに送信されます。
- 宛先ポートが送信元 VLAN に所属する場合は、送信元リストから除外され、モニタされません。
- ポートが送信元 VLAN に追加または削除されると、これらのポートで受信された送信元 VLAN のトラフィックは、モニタ中の送信元に追加または削除されます。
- VLAN 送信元と同じセッション内のフィルタ VLAN を使用することはできません。
- モニタできるのは、イーサネット VLAN だけです。

VLAN フィルタリング

トランク ポートを送信元ポートとしてモニタする場合、デフォルトでは、トランク上でアクティブなすべての VLAN がモニタされます。VLAN フィルタリングを使用して、トランク送信元ポートでの SPAN トラフィックのモニタ対象を特定の VLAN に制限できます。

- VLAN フィルタリングが適用されるのは、トランク ポートまたは音声 VLAN ポートのみです。
- VLAN フィルタリングはポートベース セッションにのみ適用され、VLAN 送信元によるセッションでは使用できません。

- VLAN フィルタ リストが指定されている場合、トランク ポートまたは音声 VLAN アクセス ポートではリスト内の該当 VLAN のみがモニタされます。
- 他のポート タイプから着信する SPAN トラフィックは、VLAN フィルタリングの影響を受けません。つまり、すべての VLAN を他のポートで使用できます。
- VLAN フィルタリング機能は、宛先 SPAN ポートに転送されたトラフィックにのみ作用し、通常のトラフィックのスイッチングには影響を与えません。

接続先ポート

各ローカル SPAN セッションまたは RSPAN 宛先セッションには、送信元ポートおよび VLAN からのトラフィックのコピーを受信し、SPAN パケットをユーザ（通常はネットワーク アナライザ）に送信する宛先ポート（別名モニタ側ポート）が必要です。

宛先ポートの特性は、次のとおりです。

- ローカル SPAN セッションの場合、宛先ポートは送信元ポートと同じデバイスまたはデバイス スタックに存在している必要があります。RSPAN セッションの場合は、RSPAN 宛先セッションを含むデバイス上にあります。RSPAN 送信元セッションのみを実行するデバイスまたはデバイス スタックには、宛先ポートはありません。
- ポートを SPAN 宛先ポートとして設定すると、元のポート設定が上書きされます。SPAN 宛先設定を削除すると、ポートは以前の設定に戻ります。ポートが SPAN 宛先ポートとして機能している間にポートの設定が変更されると、SPAN 宛先設定が削除されるまで、変更は有効になりません。

（注） SPAN の宛先ポートに QoS が設定されている場合、QoS はただちに有効になります。

- ポートが EtherChannel グループに含まれていた場合、そのポートが宛先ポートとして設定されている間、グループから削除されます。削除されたポートがルーテッド ポートであった場合、このポートはルーテッド ポートでなくなります。
- 任意のイーサネット物理ポートにできます。
- セキュア ポートにすることはできません。
- 送信元ポートにすることはできません。
- 一度に 1 つの SPAN セッションにしか参加できません（ある SPAN セッションの宛先ポートは、別の SPAN セッションの宛先ポートになることはできません）。
- アクティブな場合、着信トラフィックはディセーブルになります。ポートは SPAN セッションに必要なトラフィック以外は送信しません。宛先ポートでは着信トラフィックを学習したり、転送したりしません。
- 入力トラフィック転送がネットワーク セキュリティ デバイスでイネーブルの場合、宛先ポートはレイヤ 2 でトラフィックを転送します。

- レイヤ 2 プロトコル（STP、VTP、CDP、DTP、PAgP）のいずれにも参加しません。
- 任意の SPAN セッションの送信元 VLAN に所属する宛先ポートは、送信元リストから除外され、モニタされません。
- デバイスまたはデバイス スタックの宛先ポートの最大数は 64 です。

ローカル SPAN および RSPAN 宛先ポートは、VLAN タギングおよびカプセル化で次のように動作が異なります。

- ローカル SPAN では、宛先ポートに **encapsulation replicate** キーワードが指定されている場合、各パケットに元のカプセル化が使用されます（タグなし、ISL、または IEEE 802.1Q）。これらのキーワードが指定されていない場合、パケットはタグなしフォーマットになります。したがって、**encapsulation replicate** がイネーブルになっているローカル SPAN セッションの出力に、タグなし、ISL、または IEEE 802.1Q タグ付きパケットが混在することがあります。
- RSPAN の場合は、元の VLAN ID は RSPAN VLAN ID で上書きされるため失われます。したがって、宛先ポート上のすべてのパケットはタグなしになります。

RSPAN VLAN

RSPAN VLAN は、RSPAN の送信元セッションと宛先セッション間で SPAN トラフィックを伝送します。RSPAN VLAN には、次の特性があります。

- RSPAN VLAN 内のすべてのトラフィックは、常にフラッディングされます。
- RSPAN VLAN では MAC アドレスは学習されません。
- RSPAN VLAN トラフィックが流れるのは、トランク ポート上のみです。
- RSPAN VLAN は、**remote-span** VLAN コンフィギュレーション モード コマンドを使用して、VLAN コンフィギュレーション モードで設定する必要があります。
- STP は RSPAN VLAN トランク上で実行できますが、SPAN 宛先ポート上では実行できません。
- RSPAN VLAN を、プライベート VLAN のプライマリまたはセカンダリ VLAN にはできません。

VLAN トランキング プロトコル（VTP）に対して可視である VLAN 1 ～ 1005 の場合、VLAN ID および対応する RSPAN 特性は VTP によって伝播されます。拡張 VLAN 範囲（1006 ～ 4094）内の RSPAN VLAN ID を割り当てる場合は、すべての中間デバイスを手動で設定する必要があります。

通常は、ネットワークに複数の RSPAN VLAN を配置し、それぞれの RSPAN VLAN でネットワーク全体の RSPAN セッションを定義します。つまり、ネットワーク内の任意の場所にある複数の RSPAN 送信元セッションで、パケットを RSPAN セッションに送信できます。また、ネットワーク全体に対して複数の RSPAN 宛先セッションを設定し、同じ RSPAN VLAN をモニタしたり、ユーザにトラフィックを送信したりできます。セッションは RSPAN VLAN ID によって区別されます。

SPAN および RSPAN と他の機能の相互作用

SPAN は次の機能と相互に作用します。

- ルーティング：SPAN はルーテッド トラフィックを監視しません。VSPAN が監視するのはデバイスに出入りするトラフィックに限られ、VLAN間でルーティングされるトラフィックは監視しません。たとえば、VLAN が受信モニタされ、デバイスが別の VLAN から監視対象VLANにトラフィックをルーティングする場合、そのトラフィックは監視されず、SPAN 宛先ポートで受信されません。
- STP：SPAN または RSPAN セッションがアクティブな間、宛先ポートは STP に参加しません。SPAN または RSPAN セッションがディセーブルになると、宛先ポートは STP に参加できます。送信元ポートでは、SPAN は STP ステータスに影響を与えません。STP は RSPAN VLAN を伝送するトランク ポート上でアクティブにできます。
- CDP：SPAN セッションがアクティブな間、SPAN 宛先ポートは CDP に参加しません。SPAN セッションがディセーブルになると、ポートは再び CDP に参加します。
- VTP：VTP を使用すると、デバイス間で RSPAN VLAN のプルーニングが可能です。
- VLAN およびトランキング：送信元ポート、または宛先ポートの VLAN メンバーシップまたはトランクの設定値を、いつでも変更できます。ただし、宛先ポートの VLAN メンバーシップまたはトランクの設定値に対する変更が有効になるのは、SPAN 宛先設定を削除してからです。送信元ポートの VLAN メンバーシップまたはトランクの設定値に対する変更は、ただちに有効になり、対応する SPAN セッションが変更に応じて自動的に調整されます。
- EtherChannel：EtherChannelグループを送信元ポートとして設定することはできますが、SPAN宛先ポートとして設定することはできません。グループが SPAN 送信元として設定されている場合、グループ全体がモニタされます。

  モニタ対象の EtherChannel グループに物理ポートを追加すると、SPAN 送信元ポート リストに新しいポートが追加されます。モニタ対象の EtherChannel グループからポートを削除すると、送信元ポート リストからそのポートが自動的に削除されます。

  EtherChannel グループに所属する物理ポートを SPAN 送信元ポートとして設定し、引き続き EtherChannel の一部とすることができます。この場合、この物理ポートは EtherChannel に参加しているため、そのポートからのデータはモニタされます。ただし、EtherChannel グループに含まれる物理ポートを SPAN 宛先として設定した場合、その物理ポートはグループから削除されます。SPANセッションからそのポートが削除されると、EtherChannel グループに再加入します。EtherChannel グループから削除されたポートは、グループ メンバのままですが、inactive または suspended ステートになります。

  EtherChannel グループに含まれる物理ポートが宛先ポートであり、その EtherChannel グループが送信元の場合、ポートは EtherChannel グループおよびモニタ対象ポート リストから削除されます。
- マルチキャスト トラフィックをモニタできます。出力ポートおよび入力ポートのモニタでは、未編集のパケットが 1 つだけ SPAN 宛先ポートに送信されます。マルチキャスト パケットの送信回数は反映されません。

- プライベート VLAN ポートは、SPAN 宛先ポートには設定できません。
- セキュア ポートを SPAN 宛先ポートにすることはできません。

  SPAN セッションでは、入力転送が宛先ポートでイネーブルの場合、出力をモニタしているポートでポートセキュリティをイネーブルにしないでください。RSPAN 送信元セッションでは、出力をモニタしているポートでポートセキュリティをイネーブルにしないでください。
- IEEE 802.1x ポートは SPAN 送信元ポートにできます。SPAN 宛先ポート上で IEEE 802.1x をイネーブルにできますが、SPAN 宛先としてこのポートを削除するまで、IEEE 802.1x はディセーブルに設定されます。

  SPAN セッションでは、入力転送が宛先ポートでイネーブルの場合、出力をモニタしているポートで IEEE 802.1x をイネーブルにしないでください。RSPAN 送信元セッションでは、出力をモニタしているポートで IEEE 802.1x をイネーブルにしないでください。

SPAN と RSPAN とデバイス スタック

デバイスのスタックは 1 つの論理デバイスを表すため、ローカル SPAN の送信元ポートおよび宛先ポートは、スタック内の異なるデバイスである場合があります。したがって、スタック内でのデバイスの追加または削除は、RSPAN の送信元セッションまたは宛先セッションだけではなく、ローカル SPAN セッションにも影響を及ぼします。デバイスがスタックから削除されると、アクティブセッションが非アクティブになります。また、デバイスがスタックに追加されると、非アクティブ セッションがアクティブになります。

フローベースの SPAN

送信元ポートで監視されるトラフィックにアクセス コントロール リスト（ACL）を適用するフローベース SPAN（FSPAN）またはフローベース RSPAN（FRSPAN）を使用して、SPAN または RSPAN で監視するネットワーク トラフィックのタイプを制御できます。FSPAN ACL は、IPv4、IPv6、および監視される非 IP トラフィックをフィルタリングするように設定できます。

インターフェイスを通して ACL を SPAN セッションに適用します。ACL は SPAN セッション内のすべてのインターフェイスで監視されるすべてのトラフィックに適用されます。この ACL によって許可されるパケットは、SPAN 宛先ポートにコピーされます。ほかのパケットは SPAN 宛先ポートにコピーされません。

元のトラフィックは継続して転送され、接続している任意のポート、VLAN、およびルータ ACL が適用されます。FSPAN ACL は転送の決定に影響を与えることはありません。同様に、ポート、VLAN、およびルータ ACL は、トラフィックのモニタリングに影響を与えません。セキュリティ入力 ACL がパケットを拒否したために転送されない場合でも、FSPAN ACL が許可すると、パケットは SPAN 宛先ポートにコピーされます。しかし、セキュリティ出力 ACL がパケットを拒否したために転送されない場合、パケットは SPAN 宛先ポートにコピーされません。ただし、セキュリティ出力 ACL がパケットの送信を許可した場合だけ、パケットは、FSPAN ACL が許可した場合 SPAN 宛先ポートにコピーされます。これは RSPAN セッションについてもあてはまります。

SPAN セッションには、次の 3 つのタイプの FSPAN ACL を接続できます。

- IPv4 FSPAN ACL：IPv4 パケットだけをフィルタリングします。
- IPv6 FSPAN ACL：IPv6 パケットだけをフィルタリングします。
- MAC FSPAN ACL：IP パケットだけをフィルタリングします。

スタックに設定された VLAN ベースの FSPAN セッションが 1 つまたは複数のデバイス上のハードウェア メモリに収まらない場合、セッションはこれらのデバイス上でアンロードされたものとして処理され、デバイスでの FSPAN ACL およびソーシングのためのトラフィックは、SPAN 宛先ポートにコピーされません。FSPAN ACL は継続して正しく適用され、トラフィックは FSPAN ACL がハードウェア メモリに収まるデバイスの SPAN 宛先ポートにコピーされます。

空の FSPAN ACL が接続されると、一部のハードウェア機能により、その ACL の SPAN 宛先ポートにすべてのトラフィックがコピーされます。十分なハードウェア リソースが使用できない場合、空の FSPAN ACL もアンロードされる可能性があります。

SPAN および RSPAN のデフォルト設定

表 *6 : SPAN* および *RSPAN* のデフォルト設定

機能	デフォルト設定
SPAN のステート（SPAN および RSPAN）	ディセーブル。
モニタする送信元ポート トラフィック	受信トラフィックと送信トラフィックの両方（**both**）
カプセル化タイプ（宛先ポート）	ネイティブ形式（タグなしパケット）
入力転送（宛先ポート）	ディセーブル。
VLAN フィルタリング	送信元ポートとして使用されるトランク インターフェイス上では、すべての VLAN がモニタリングされます。
RSPAN VLAN	未設定

設定時の注意事項

SPAN 設定時の注意事項

- SPAN セッションから送信元ポート、宛先ポート、または VLAN を削除する場合は、**no monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} グローバル コンフィギュレーション コマンドまたは **no monitor session** *session_number* **destination interface**

interface-id 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。宛先インターフェイスの場合、このコマンドの **no** 形式では、**encapsulation** オプションは無視されます。

- トランク ポート上のすべての VLAN をモニタするには、**no monitor session** *session_number* **filter** 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。

RSPAN 設定時の注意事項

- すべての SPAN 設定時の注意事項が RSPAN に適用されます。
- RSPAN VLAN には特性があるので、RSPAN VLAN として使用するためにネットワーク上の VLAN をいくつか確保し、それらの VLAN にはアクセス ポートを割り当てないでおく必要があります。
- RSPAN トラフィックに出力 ACL を適用して、特定のパケットを選択的にフィルタリングまたはモニタできます。RSPAN 送信元デバイス内の RSPAN VLAN 上で、これらの ACL を指定します。
- RSPAN を設定する場合は、送信元ポートおよび宛先ポートをネットワーク内の複数のデバイスに分散させることができます。
- RSPAN VLAN 上のアクセス ポート（音声 VLAN ポートを含む）は、非アクティブ ステートになります。
- 次の条件を満たす限り、任意の VLAN を RSPAN VLAN として設定できます。
  - すべてのデバイスで、RSPAN セッションに同じ RSPAN VLAN が使用されている。
  - 参加しているすべてのデバイスで RSPAN がサポートされている。

FSPAN および FRSPAN 設定時の注意事項

- 少なくとも 1 つの FSPAN ACL が接続されている場合、FSPAN はイネーブルになります。
- SPAN セッションに空ではない FSPAN ACL を少なくとも 1 つ接続し、ほかの 1 つまたは複数の FSPAN ACL を接続しなかった場合（たとえば、空ではない IPv4 ACL を接続し、IPv6 と MAC ACL を接続しなかった場合）、FSPAN は、接続されていない ACL によってフィルタリングされたと思われるトラフィックをブロックします。したがって、このトラフィックは監視されません。

SPAN および RSPAN の設定方法

ローカル SPAN セッションの作成

SPAN セッションを作成し、送信元（監視対象）ポートまたは VLAN、および宛先（監視側）ポートを指定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} 例： `Device(config)# no monitor session all`	セッションに対する既存の SPAN 設定を削除します。 • *session_number* の範囲は、1 ～ 66 です。 • **all**：すべての SPAN セッションを削除します。 • **local**：すべてのローカル セッションを削除します。 • **remote**：すべてのリモート SPAN セッションを削除します。
ステップ 4	**monitor session** *session_number***source** {**interface** *interface-id* |**vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**] 例： `Device(config)# monitor session 1 source interface gigabitethernet1/0/1`	SPAN セッションおよび送信元ポート（監視対象ポート）を指定します。 • *session_number* の範囲は、1 ～ 66 です。 • *interface-id* には、監視する送信元ポートを指定します。有効なインターフェイスには、物理インターフェイスおよびポート チャネル論理インターフェイス（**port-channel** *port-channel-number*）があります。有効なポートチャネル番号は 1 ～ 48 です。 • *vlan-id* には、監視する送信元 VLAN を指定します。指定できる範囲は 1 ～ 4094 です（RSPAN VLAN は除く）。

	コマンドまたはアクション	目的
		（注） 1つのセッションに、一連のコマンドで定義された複数の送信元（ポートまたは VLAN）を含めることができます。ただし、1つのセッション内では送信元ポートと送信元 VLAN を併用できません。 • （任意）**[,** | **-]** は、一連または一定範囲のインターフェイスを指定します。カンマの前後およびハイフンの前後にスペースを1つずつ入力します。 • （任意）**both** | **rx** | **tx**：監視するトラフィックの方向を指定します。トラフィックの方向を指定しなかった場合、送信元インターフェイスは送信トラフィックと受信トラフィックの両方を送信します。 • **both**：受信トラフィックと送信トラフィックの両方を監視します。 • **rx**：受信トラフィックを監視します。 • **tx**：送信トラフィックを監視します。 （注） **monitor session** *session_number***source** コマンドを複数回使用すると、複数の送信元ポートを設定できます。
ステップ 5	**monitor session** *session_number***destination** {**interface** *interface-id* **[,** | **-] [encapsulation replicate**]}	SPANセッションおよび宛先ポート（監視側ポート）を指定します。

	コマンドまたはアクション	目的
	例： `Device(config)# monitor session 1 destination interface gigabitethernet1/0/2 encapsulation replicate`	（注） ローカル SPAN の場合は、送信元および宛先インターフェイスに同じセッション番号を使用する必要があります。 • *session_number* には、ステップ 4 で入力したセッション番号を指定します。 • *interface-id* には、宛先ポートを指定します。宛先インターフェイスには物理ポートを指定する必要があります。EtherChannel や VLAN は指定できません。 • （任意）[**,** | **-**] は、一連または一定範囲のインターフェイスを指定します。カンマの前後およびハイフンの前後にスペースを 1 つずつ入力します。 （任意）**encapsulation replicate** は、宛先インターフェイスが送信元インターフェイスのカプセル化方式を複製することを指定します。選択しない場合のデフォルトは、ネイティブ形式（タグなし）でのパケットの送信です。 （注） **monitor session** *session_number* **destination** コマンドを複数回使用すると、複数の宛先ポートを設定できます。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。

	コマンドまたはアクション	目的
ステップ 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。

ローカル SPAN セッションの作成および着信トラフィックの設定

SPAN セッションを作成し、さらに送信元ポートまたは VLAN および宛先ポートを指定した後、宛先ポートでネットワーク セキュリティ デバイス（Cisco IDS センサー装置等）用に着信トラフィックをイネーブルにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} 例： `Device(config)# no monitor session all`	セッションに対する既存の SPAN 設定を削除します。 • *session_number* の範囲は、1 ～ 66 です。 • **all**：すべての SPAN セッションを削除します。 • **local**：すべてのローカル セッションを削除します。 • **remote**：すべてのリモート SPAN セッションを削除します。
ステップ 4	**monitor session** *session_number***source** {**interface** *interface-id* |**vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**]	SPAN セッションおよび送信元ポート（モニタ対象ポート）を指定します。

	コマンドまたはアクション	目的
	例： `Device(config)# monitor session 2 source gigabitethernet1/0/1 rx`	

ステップ 5	**monitor session** *session_number***destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**] [**ingress** {**dot1q vlan** *vlan-id* | **untagged vlan** *vlan-id* | **vlan** *vlan-id*}]}
	
	例：
	
	`Device(config)# monitor session 2 destination interface gigabitethernet1/0/2 encapsulation replicate ingress dot1q vlan 6`
		SPANセッション、宛先ポート、パケットカプセル化、および入力 VLAN とカプセル化を指定します。
		
		• *session_number* には、ステップ 4 で入力したセッション番号を指定します。
		
		• *interface-id* には、宛先ポートを指定します。宛先インターフェイスには物理ポートを指定する必要があります。EtherChannel や VLAN は指定できません。
		
		•（任意）**[,|-]**：一連のインターフェイスまたはインターフェイス範囲を指定します。カンマまたはハイフンの前後にスペースを 1 つずつ入力します。
		
		•（任意）**encapsulation replicate** は、宛先インターフェイスが送信元インターフェイスのカプセル化方式を複製することを指定します。選択しない場合のデフォルトは、ネイティブ形式（タグなし）でのパケットの送信です。
		
		• **ingress** 宛先ポートでの着信トラフィックの転送をイネーブルにして、カプセル化タイプを指定します。
		
		  • **dot1q vlan** *vlan-id*：デフォルトの VLAN として指定した VLAN で、IEEE 802.1Q でカプセル化された着信パケットを受信します。
		
		  • **untagged vlan** *vlan-id* または **vlan** *vlan-id*：デフォルトの VLAN として指定した VLAN

	コマンドまたはアクション	目的
		で、タグなしでカプセル化された着信パケットを受信します。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

フィルタリングする VLAN の指定

SPAN 送信元トラフィックを特定の VLAN に制限するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} 例：	セッションに対する既存の SPAN 設定を削除します。

	コマンドまたはアクション	目的
	`Device(config)# no monitor session all`	• *session_number* の範囲は、1 ～ 66 です。 • **all**：すべての SPAN セッションを削除します。 • **local**：すべてのローカル セッションを削除します。 • **remote**：すべてのリモート SPAN セッションを削除します。
ステップ 4	**monitor session** *session_number* **source interface** *interface-id* 例： `Device(config)# monitor session 2 source interface gigabitethernet1/0/2 rx`	送信元ポート（モニタ対象ポート）と SPAN セッションの特性を指定します。 • *session_number* の範囲は、1 ～ 66 です。 • *interface-id* には、モニタリングする送信元ポートを指定します。指定したインターフェイスは、あらかじめトランク ポートとして設定しておく必要があります。
ステップ 5	**monitor session** *session_number***filter vlan** *vlan-id* [**,**	**-**] 例： `Device(config)# monitor session 2 filter vlan 1 - 5 , 9`	SPAN 送信元トラフィックを特定の VLAN に制限します。 • *session_number* には、ステップ 4 で指定したセッション番号を入力します。 • *vlan-id* に指定できる範囲は 1 ～ 4094 です。 •（任意）カンマ（**,**）を使用して一連の VLAN を指定するか、ハイフン（**-**）を使用して VLAN 範囲を指定します。カンマの前後およびハイフンの前後にスペースを 1 つずつ入力します。
ステップ 6	**monitor session** *session_number***destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**]} 例： `Device(config)# monitor session 2`	SPAN セッションおよび宛先ポート（モニタ側ポート）を指定します。 • *session_number* には、ステップ 4 で入力したセッション番号を指定します。

	コマンドまたはアクション	目的
	destination interface gigabitethernet1/0/1	• *interface-id* には、宛先ポートを指定します。宛先インターフェイスには物理ポートを指定する必要があります。EtherChannel や VLAN は指定できません。 • （任意）**[,| -]** は、一連または一定範囲のインターフェイスを指定します。カンマの前後およびハイフンの前後にスペースを 1 つずつ入力します。 • （任意）**encapsulation replicate** は、宛先インターフェイスが送信元インターフェイスのカプセル化方式を複製することを指定します。選択しない場合のデフォルトは、ネイティブ形式（タグなし）でのパケットの送信です。
ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

RSPAN VLAN としての VLAN の設定

新しい VLAN を作成し、RSPAN セッション用の RSPAN VLAN になるように設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**vlan** *vlan-id* 例： `Device(config)# vlan 100`	VLAN ID を入力して VLAN を作成するか、または既存の VLAN の VLAN ID を入力して、VLAN コンフィギュレーション モードを開始します。指定できる範囲は 2 ～ 1001 または 1006 ～ 4094 です。 RSPAN VLAN を VLAN 1（デフォルト VLAN）または VLAN ID 1002 ～ 1005（トークンリングおよび FDDI VLAN 専用）にすることはできません。
ステップ 4	**remote-span** 例： `Device(config-vlan)# remote-span`	VLAN を RSPAN VLAN として設定します。
ステップ 5	**end** 例： `Device(config-vlan)# end`	特権 EXEC モードに戻ります。
ステップ 6	**show running-config** 例： `Device# show running-config`	入力を確認します。
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。

次のタスク

RSPAN に参加するすべてのデバイスに RSPAN VLAN を作成する必要があります。RSPAN VLAN ID が標準範囲（1005 未満）であり、VTP がネットワーク内でイネーブルである場合は、1 つのデバイスに RSPAN VLAN を作成し、VTP がこの RSPAN VLAN を VTP ドメイン内の他のデバイスに伝播するように設定できます。拡張範囲 VLAN（1005 を超える ID）の場合、送信元と宛先の両方のデバイス、および中間デバイスに RSPAN VLAN を設定する必要があります。

VTP プルーニングを使用して、RSPAN トラフィックが効率的に流れるようにするか、または RSPAN トラフィックの伝送が不要なすべてのトランクから、RSPAN VLAN を手動で削除します。

VLAN からリモート SPAN 特性を削除して、標準 VLAN に戻すように変換するには、**no remote-span** VLAN コンフィギュレーション コマンドを使用します。

SPAN セッションから送信元ポートまたは VLAN を削除するには、**no monitor session** *session_number***source** {**interface** *interface-id* |**vlan** *vlan-id*} 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。セッションから RSPAN VLAN を削除するには、**no monitor session** *session_number***destination remote vlan** *vlan-id* コマンドを使用します。

RSPAN 送信元セッションの作成

RSPAN 送信元セッションを作成および開始し、モニタ対象の送信元および宛先 RSPAN VLAN を指定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} 例： `Device(config)# no monitor session 1`	セッションに対する既存の SPAN 設定を削除します。 • *session_number* の範囲は、1 ～ 66 です。 • **all**：すべての SPAN セッションを削除します。

	コマンドまたはアクション	目的
		• **local**：すべてのローカル セッションを削除します。 • **remote**：すべてのリモート SPAN セッションを削除します。
ステップ 4	**monitor session** *session_number***source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**] 例： `Device(config)# monitor session 1 source interface gigabitethernet1/0/1 tx`	RSPAN セッションおよび送信元ポート（モニタ対象ポート）を指定します。 • *session_number* の範囲は、1 ～ 66 です。 • RSPAN セッションの送信元ポートまたは送信元 VLAN を入力します。 • *interface-id* には、モニタリングする送信元ポートを指定します。有効なインターフェイスには、物理インターフェイスおよびポートチャネル論理インターフェイス（**port-channel** *port-channel-number*）があります。有効なポートチャネル番号は 1 ～ 48 です。 • *vlan-id* には、モニタする送信元 VLAN を指定します。指定できる範囲は 1 ～ 4094 です（RSPAN VLAN は除く）。 1 つのセッションに、一連のコマンドで定義された複数の送信元（ポートまたは VLAN）を含めることができます。ただし、1 つのセッション内で送信元ポートと送信元 VLAN を併用することはできません。 • （任意）**[,|-]**：一連のインターフェイスまたはインターフェイス範囲を指定します。カンマの前後およびハイフンの前後にスペースを 1 つずつ入力します。 • （任意）**both** | **rx** | **tx**：監視するトラフィックの方向を指定します。トラフィックの方向を指定しなかった

	コマンドまたはアクション	目的
		場合、送信元インターフェイスは送信トラフィックと受信トラフィックの両方を送信します。 • **both**：受信トラフィックと送信トラフィックの両方を監視します。 • **rx**：受信トラフィックを監視します。 • **tx**：送信トラフィックをモニタします。
ステップ 5	**monitor session** *session_number***destinationremote vlan** *vlan-id* 例： `Device(config)# monitor session 1 destination remote vlan 100`	RSPAN セッション、宛先 RSPAN VLAN、および宛先ポートグループを指定します。 • *session_number* には、ステップ 4 で指定した番号を入力します。 • *vlan-id* には、モニタリングする送信元 RSPAN VLAN を指定します。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

フィルタリングする VLAN の指定

RSPAN 送信元トラフィックを特定の VLAN に制限するように RSPAN 送信元セッションを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} 例： `Device(config)# no monitor session 2`	セッションに対する既存の SPAN 設定を削除します。 • *session_number* の範囲は、1 ～ 66 です。 • **all**：すべての SPAN セッションを削除します。 • **local**：すべてのローカル セッションを削除します。 • **remote**：すべてのリモート SPAN セッションを削除します。
ステップ 4	**monitor session** *session_number* **source interface** *interface-id* 例： `Device(config)# monitor session 2 source interface gigabitethernet1/0/2 rx`	送信元ポート（モニタ対象ポート）と SPAN セッションの特性を指定します。 • *session_number* の範囲は、1 ～ 66 です。 • *interface-id* には、モニタリングする送信元ポートを指定します。指定したインターフェイスは、あらかじめトランク ポートとして設定しておく必要があります。

	コマンドまたはアクション	目的
ステップ 5	**monitor session** *session_number***filter vlan** *vlan-id* [, | -] 例： `Device(config)# monitor session 2 filter vlan 1 - 5 , 9`	SPAN 送信元トラフィックを特定の VLAN に制限します。 • *session_number* には、ステップ 4 で指定したセッション番号を入力します。 • *vlan-id* に指定できる範囲は 1 ～ 4094 です。 • （任意）**,** | **-**：カンマ（**,**）を使用して一連の VLAN を指定するか、ハイフン（**-**）を使用して VLAN 範囲を指定します。カンマの前後およびハイフンの前後にスペースを 1 つずつ入力します。
ステップ 6	**monitor session** *session_number***destinationremote vlan** *vlan-id* 例： `Device(config)# monitor session 2 destination remote vlan 902`	RSPAN セッションおよび宛先リモート VLAN（RSPAN VLAN）を指定します。 • *session_number* には、ステップ 4 で指定したセッション番号を入力します。 • *vlan-id* には、宛先ポートにモニタ対象トラフィックを伝送する RSPAN VLAN を指定します。
ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

RSPAN 宛先セッションの作成

RSPAN 宛先セッションは、別のデバイスまたはデバイス スタック（送信元セッションが設定されていないデバイスまたはデバイス スタック）に設定します。

このデバイス上で RSPAN VLAN を定義し、RSPAN 宛先セッションを作成し、送信元 RSPAN VLAN および宛先ポートを指定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**vlan** *vlan-id* 例： `Device(config)# vlan 901`	送信元デバイスで作成された RSPAN VLAN の VLAN ID を指定し、VLAN コンフィギュレーション モードを開始します。 両方のデバイスが VTP に参加し、RSPAN VLAN ID が 2 ～ 1005 である場合は、VTP ネットワークを介して RSPAN VLAN ID が伝播されるため、ステップ 3 ～ 5 は不要です。
ステップ 4	**remote-span** 例： `Device(config-vlan)# remote-span`	VLAN を RSPAN VLAN として識別します。
ステップ 5	**exit** 例： `Device(config-vlan)# exit`	グローバル コンフィギュレーション モードに戻ります。
ステップ 6	**no monitor session** {*session_number* | **all** | **local** | **remote**} 例：	セッションに対する既存の SPAN 設定を削除します。

	コマンドまたはアクション	目的
	`Device(config)# no monitor session 1`	• *session_number* の範囲は、1 ～ 66 です。 • **all**：すべての SPAN セッションを削除します。 • **local**：すべてのローカルセッションを削除します。 • **remote**：すべてのリモート SPAN セッションを削除します。
ステップ 7	**monitor session** *session_number* **sourceremote vlan** *vlan-id* 例： `Device(config)# monitor session 1 source remote vlan 901`	RSPAN セッションと送信元 RSPAN VLAN を指定します。 • *session_number* の範囲は、1 ～ 66 です。 • *vlan-id* には、モニタリングする送信元 RSPAN VLAN を指定します。
ステップ 8	**monitor session** *session_number* **destination interface** *interface-id* 例： `Device(config)# monitor session 1 destination interface gigabitethernet2/0/1`	RSPAN セッションと宛先インターフェイスを指定します。 • *session_number* には、ステップ 7 で指定した番号を入力します。 RSPAN 宛先セッションでは、送信元 RSPAN VLAN および宛先ポートに同じセッション番号を使用する必要があります。 • *interface-id* には、宛先インターフェイスを指定します。宛先インターフェイスは物理インターフェイスでなければなりません。 • **encapsulation replicate** はコマンドラインのヘルプストリングに表示されますが、RSPAN ではサポートされていません。元の VLAN ID は RSPAN VLAN ID によって上書きされ、宛先ポート上のすべてのパケットはタグなしになります。
ステップ 9	**end** 例：	特権 EXEC モードに戻ります。

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

RSPAN 宛先セッションの作成および着信トラフィックの設定

RSPAN 宛先セッションを作成し、送信元 RSPAN VLAN および宛先ポートを指定し、宛先ポートでネットワーク セキュリティ デバイス（Cisco IDS センサー装置等）用に着信トラフィックをイネーブルにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 3	**no monitor session** {*session_number* | **all** | **local** | **remote**} 例： `Device(config)# no monitor session 2`	セッションに対する既存の SPAN 設定を削除します。 • *session_number* の範囲は、1 ～ 66 です。 • **all**：すべての SPAN セッションを削除します。

	コマンドまたはアクション	目的
		• **local**：すべてのローカル セッションを削除します。 • **remote**：すべてのリモート SPAN セッションを削除します。
ステップ 4	**monitor session** *session_number***sourceremote vlan** *vlan-id* 例： `Device(config)# monitor session 2 source remote vlan 901`	RSPAN セッションと送信元 RSPAN VLAN を指定します。 • *session_number* の範囲は、1 ～ 66 です。 • *vlan-id* には、モニタリングする送信元 RSPAN VLAN を指定します。
ステップ 5	**monitor session** *session_number* **destination** {**interface** *interface-id* [**,** | **-**] [**ingress** {**dot1q vlan** *vlan-id* | **untagged vlan** *vlan-id*| **vlan** *vlan-id*}]} 例： `Device(config)# monitor session 2 destination interface gigabitethernet1/0/2 ingress vlan 6`	SPAN セッション、宛先ポート、パケット カプセル化、および着信 VLAN とカプセル化を指定します。 • *session_number* には、ステップ 5 で指定した番号を入力します。 RSPAN 宛先セッションでは、送信元 RSPAN VLAN および宛先ポートに同じセッション番号を使用する必要があります。 • *interface-id* には、宛先インターフェイスを指定します。宛先インターフェイスは物理インターフェイスでなければなりません。 • **encapsulation replicate** はコマンドラインのヘルプ ストリングに表示されますが、RSPAN ではサポートされていません。元の VLAN ID は RSPAN VLAN ID によって上書きされ、宛先ポート上のすべてのパケットはタグなしになります。 • （任意）**[,** | **-]** は、一連または一定範囲のインターフェイスを指定します。カンマの前後およびハイフンの前後にスペースを 1 つずつ入力します。 • 宛先ポートでの着信トラフィックの転送をイネーブルにして、カプセル

	コマンドまたはアクション	目的
		化タイプを指定するには、**ingress** を追加のキーワードと一緒に入力します。 • **dot1q vlan** *vlan-id*：デフォルトの VLAN として指定した VLAN で、IEEE 802.1Q でカプセル化された着信パケットを転送します。 • **untagged vlan** *vlan-id* または **vlan** *vlan-id*：デフォルトの VLAN として指定した VLAN で、タグなしでカプセル化された着信パケットを転送します。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 7	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 8	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

FSPAN セッションの設定

SPAN セッションを作成し、送信元（監視対象）ポートまたは VLAN、および宛先（モニタ側）ポートを指定し、セッションに FSPAN を設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
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	**no monitor session** {*session_number* | **all** | **local** | **remote**} 例： `Device(config)# no monitor session 2`	セッションに対する既存の SPAN 設定を削除します。 • *session_number* の範囲は、1 ～ 66 です。 • **all**：すべての SPAN セッションを削除します。 • **local**：すべてのローカル セッションを削除します。 • **remote**：すべてのリモート SPAN セッションを削除します。
ステップ 4	**monitor session** *session_number***source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**] 例： `Device(config)# monitor session 2 source interface gigabitethernet1/0/1`	SPAN セッションおよび送信元ポート（監視対象ポート）を指定します。 • *session_number* の範囲は、1 ～ 66 です。 • *interface-id* には、モニタリングする送信元ポートを指定します。有効なインターフェイスには、物理インターフェイスおよびポート チャネル論理インターフェイス（**port-channel** *port-channel-number*）があります。有効なポートチャネル番号は 1 ～ 48 です。 • *vlan-id* には、モニタリングする送信元 VLAN を指定します。指定できる範囲は 1 ～ 4094 です（RSPAN VLAN は除く）。

	コマンドまたはアクション	目的
		（注） 1つのセッションに、一連のコマンドで定義された複数の送信元（ポートまたは VLAN）を含めることができます。ただし、1つのセッション内では送信元ポートと送信元 VLAN を併用できません。 • （任意）[**,**|**-**]：一連のインターフェイスまたはインターフェイス範囲を指定します。カンマの前後およびハイフンの前後にスペースを1つずつ入力します。 • （任意）[**both**|**rx**|**tx**]：モニタリングするトラフィックの方向を指定します。トラフィックの方向を指定しなかった場合、SPAN は送信トラフィックと受信トラフィックの両方をモニタします。 • **both**：送信トラフィックと受信トラフィックの両方をモニタします。これがデフォルトです。 • **rx**：受信トラフィックをモニタします。 • **tx**：送信トラフィックを監視します。 （注） **monitor session** *session_number***source** コマンドを複数回使用すると、複数の送信元ポートを設定できます。
ステップ 5	**monitor session** *session_number***destination** {**interface** *interface-id* [**,** | **-**] [**encapsulation replicate**]} 例：	SPANセッションおよび宛先ポート（モニタ側ポート）を指定します。 • *session_number* には、ステップ 4 で入力したセッション番号を指定します。

	コマンドまたはアクション	目的
	Device(config)# **monitor session 2 destination interface gigabitethernet1/0/2 encapsulation replicate**	• **destination** には、次のパラメータを指定します。 • *interface-id* には、宛先ポートを指定します。宛先インターフェイスには物理ポートを指定する必要があります。EtherChannel や VLAN は指定できません。 • （任意）**[,**	**-]** は、一連または一定範囲のインターフェイスを指定します。カンマの前後およびハイフンの前後にスペースを 1 つずつ入力します。 • （任意）**encapsulation replicate** は、宛先インターフェイスが送信元インターフェイスのカプセル化方式を複製することを指定します。選択しない場合のデフォルトは、ネイティブ形式（タグなし）でのパケットの送信です。 （注） ローカル SPAN の場合は、送信元および宛先インターフェイスに同じセッション番号を使用する必要があります。**monitor session** *session_number***destination** コマンドを複数回使用すると、複数の宛先ポートを設定できます。
ステップ 6	**monitor session** *session_number***filter** {**ip** | **ipv6** | **mac**} **access-group** {*access-list-number* | *name*} 例： Device(config)# **monitor session 2 filter ipv6 access-group 4**	SPAN セッション、フィルタリングするパケットのタイプ、および FSPAN セッションで使用する ACL を指定します。 • *session_number* には、ステップ 4 で入力したセッション番号を指定します。 • *access-list-number* には、トラフィックのフィルタリングに使用したい ACL 番号を指定します。

	コマンドまたはアクション	目的
		• *name* には、トラフィックのフィルタリングに使用する ACL の名前を指定します。
ステップ 7	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 8	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 9	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

FRSPAN セッションの設定

RSPAN 送信元セッションを開始し、監視対象の送信元および宛先 RSPAN VLAN を指定し、セッションに FRSPAN を設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
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	**no monitor session** {*session_number* | **all** | **local** | **remote**} 例：	セッションに対する既存の SPAN 設定を削除します。

	コマンドまたはアクション	目的
	`Device(config)# no monitor session 2`	• *session_number* の範囲は、1 ～ 66 です。 • **all**：すべての SPAN セッションを削除します。 • **local**：すべてのローカルセッションを削除します。 • **remote**：すべてのリモート SPAN セッションを削除します。
ステップ 4	**monitor session** *session_number***source** {**interface** *interface-id* | **vlan** *vlan-id*} [**,** | **-**] [**both** | **rx** | **tx**] 例： `Device(config)# monitor session 2 source interface gigabitethernet1/0/1`	SPAN セッションおよび送信元ポート（監視対象ポート）を指定します。 • *session_number* の範囲は、1 ～ 66 です。 • *interface-id* には、モニタリングする送信元ポートを指定します。有効なインターフェイスには、物理インターフェイスおよびポートチャネル論理インターフェイス（**port-channel** *port-channel-number*）があります。有効なポートチャネル番号は 1 ～ 48 です。 • *vlan-id* には、モニタリングする送信元 VLAN を指定します。指定できる範囲は 1 ～ 4094 です（RSPAN VLAN は除く）。 （注） 1 つのセッションに、一連のコマンドで定義された複数の送信元（ポートまたは VLAN）を含めることができます。ただし、1 つのセッション内では送信元ポートと送信元 VLAN を併用できません。 • （任意）[**,** | **-**]：一連のインターフェイスまたはインターフェイス範囲を指定します。カンマの前後

	コマンドまたはアクション	目的
		およびハイフンの前後にスペースを 1 つずつ入力します。 • （任意） **[both | rx | tx]**：モニタリングするトラフィックの方向を指定します。トラフィックの方向を指定しなかった場合、SPAN は送信トラフィックと受信トラフィックの両方をモニタします。 • **both**：送信トラフィックと受信トラフィックの両方をモニタします。これがデフォルトです。 • **rx**：受信トラフィックをモニタします。 • **tx**：送信トラフィックを監視します。 （注） **monitor session** *session_number***source** コマンドを複数回使用すると、複数の送信元ポートを設定できます。
ステップ 5	**monitor session** *session_number***destinationremote vlan** *vlan-id* 例： `Device(config)# monitor session 2 destination remote vlan 5`	RSPAN セッションと宛先 RSPAN VLAN を指定します。 • *session_number* には、ステップ 4 で指定した番号を入力します。 • *vlan-id* には、モニタリングする宛先 RSPAN VLAN を指定します。
ステップ 6	**vlan** *vlan-id* 例： `Device(config)# vlan 10`	VLAN コンフィギュレーション モードを開始します。*vlan-id* には、モニタリングする送信元 RSPAN VLAN を指定します。
ステップ 7	**remote-span** 例： `Device(config-vlan)# remote-span`	ステップ 5 で指定した VLAN が RSPAN VLAN の一部であることを指定します。
ステップ 8	**exit** 例：	グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
	`Device(config-vlan)# exit`	
ステップ 9	**monitor session** *session_number***filter** {**ip** | **ipv6** | **mac**} **access-group** {*access-list-number* | *name*} 例： `Device(config)# monitor session 2 filter ip access-group 7`	RSPANセッション、フィルタリングするパケットのタイプ、およびFRSPANセッションで使用するACLを指定します。 • *session_number* には、ステップ 4 で入力したセッション番号を指定します。 • *access-list-number* には、トラフィックのフィルタリングに使用したいACL 番号を指定します。 • *name* には、トラフィックのフィルタリングに使用するACLの名前を指定します。
ステップ 10	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 11	**show running-config** 例： `Device# show running-config`	入力を確認します。
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。

SPAN および RSPAN 動作のモニタリング

次の表で、SPAN および RSPAN 動作の設定と結果を表示して動作を監視するために使用するコマンドについて説明します。

表 *7: SPAN* および *RSPAN* 動作のモニタリング

コマンド（Command）	目的
show monitor	現在の SPAN、RSPAN、FSPAN、または FRSPAN 設定を表示します。

SPAN および RSPAN の設定例

例：ローカル SPAN の設定

次に、SPAN セッション 1 を設定し、宛先ポートへ向けた送信元ポートのトラフィックをモニタする例を示します。最初に、セッション 1 の既存の SPAN 設定を削除し、カプセル化方式を維持しながら、双方向トラフィックを送信元ポート GigabitEthernet 1 から宛先ポート GigabitEthernet 2 にミラーリングします。

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1
Device(config)# monitor session 1 source interface gigabitethernet1/0/1
Device(config)# monitor session 1 destination interface gigabitethernet1/0/2
encapsulation replicate
Device(config)# end
```

次に、SPAN セッション 1 の SPAN 送信元としてのポート 1 を削除する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1 source interface gigabitethernet1/0/1
Device(config)# end
```

次に、双方向モニタが設定されていたポート 1 で、受信トラフィックのモニタをディセーブルにする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1 source interface gigabitethernet1/0/1 rx
```

ポート 1 で受信するトラフィックのモニタはディセーブルになりますが、このポートから送信されるトラフィックは引き続きモニタされます。

次に、SPAN セッション 2 内の既存の設定を削除し、VLAN 1 ～ 3 に属するすべてのポートで受信トラフィックをモニタするように SPAN セッション 2 を設定し、モニタされたトラフィッ

クを宛先ポート GigabitEthernet 2 に送信する例を示します。さらに、この設定は VLAN 10 に属するすべてのポートですべてのトラフィックをモニタするよう変更されます。

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source vlan 1 - 3 rx
Device(config)# monitor session 2 destination interface gigabitethernet1/0/2
Device(config)# monitor session 2 source vlan 10
Device(config)# end
```

次に、SPAN セッション 2 の既存の設定を削除し、送信元ポート GigabitEthernet 1 上で受信されるトラフィックをモニタするように SPAN セッション 2 を設定し、送信元ポートと同じ出力カプセル化方式を使用してそれを宛先ポート GigabitEthernet 2 に送信し、VLAN 6 をデフォルトの入力 VLAN として IEEE 802.1Q カプセル化を使用する入力転送をイネーブルにする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source gigabitethernet1/0/1 rx
Device(config)# monitor session 2 destination interface gigabitethernet1/0/2 encapsulation
 replicate ingress dot1q vlan 6
Device(config)# end
```

次に、SPAN セッション 2 の既存の設定を削除し、トランク ポート GigabitEthernet 2 で受信されたトラフィックをモニタするように SPAN セッション 2 を設定し、VLAN 1 ～ 5 および 9 に対してのみトラフィックを宛先ポート GigabitEthernet 1 に送信する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source interface gigabitethernet1/0/2 rx
Device(config)# monitor session 2 filter vlan 1 - 5 , 9
Device(config)# monitor session 2 destination interface gigabitethernet1/0/1
Device(config)# end
```

例：RSPAN VLAN の作成

この例は、RSPAN VLAN 901 の作成方法を示しています。

```
Device> enable
Device# configure terminal
Device(config)# vlan 901
Device(config-vlan)# remote span
Device(config-vlan)# end
```

次に、セッション 1 に対応する既存の RSPAN 設定を削除し、複数の送信元インターフェイスをモニタするように RSPAN セッション 1 を設定し、さらに宛先を RSPAN VLAN 901 に設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1
```

```
Device(config)# monitor session 1 source interface gigabitethernet1/0/1 tx
Device(config)# monitor session 1 source interface gigabitethernet1/0/2 rx
Device(config)# monitor session 1 source interface port-channel 2
Device(config)# monitor session 1 destination remote vlan 901
Device(config)# end
```

次に、RSPANセッション2の既存の設定を削除し、トランク ポート2で受信されるトラフィックをモニタするようにRSPANセッション2を設定し、VLAN 1 ～ 5および9に対してのみトラフィックを宛先RSPAN VLAN 902に送信する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source interface gigabitethernet1/0/2 rx
Device(config)# monitor session 2 filter vlan 1 - 5 , 9
Device(config)# monitor session 2 destination remote vlan 902
Device(config)# end
```

次に、送信元リモートVLANとしてVLAN 901、宛先インターフェイスとしてポート1を設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# monitor session 1 source remote vlan 901
Device(config)# monitor session 1 destination interface gigabitethernet2/0/1
Device(config)# end
```

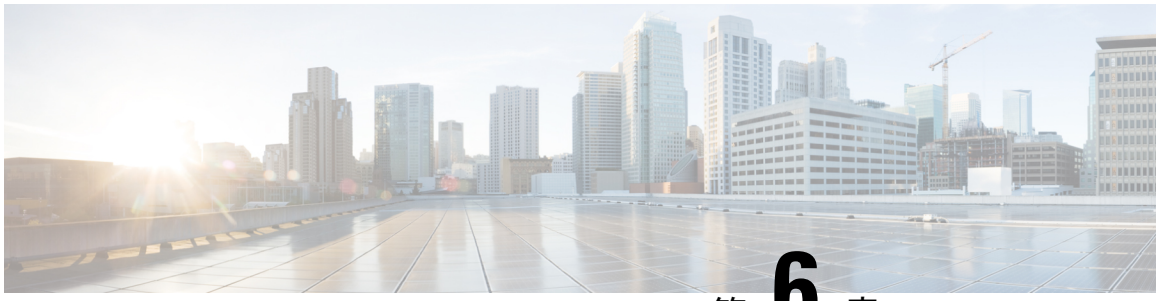
次に、RSPANセッション2で送信元リモートVLANとしてVLAN 901を設定し、送信元ポートGigabitEthernet2を宛先インターフェイスとして設定し、VLAN6をデフォルトの受信VLANとして着信トラフィックの転送をイネーブルにする例を示します。

```
Device> enable
Device# configure terminal
Device(config)# monitor session 2 source remote vlan 901
Device(config)# monitor session 2 destination interface gigabitethernet1/0/2 ingress
vlan 6
Device(config)# end
```

SPAN および RSPAN の機能の履歴と情報

リリース	変更箇所
Cisco IOS XE Everest 16.5.1a	スイッチ ポート アナライザ（SPAN）：スニファやアナライザまたはRMONプローブを使用してポートまたはVLANのデバイスのトラフィックを監視できます。 この機能が導入されました。

リリース	変更箇所
Cisco IOS XE Everest 16.5.1a	フローベースのスイッチ ポート アナライザ（SPAN）：指定されたフィルタを使用してエンド ホスト間の必要なデータのみをキャプチャする手段を提供します。フィルタは、IPv4、IPv6 または IPv4 と IPv6、あるいは指定された送信元と宛先アドレス間の IP トラフィック（MAC）以外を制限するアクセス リストの観点から定義されます。 この機能が導入されました。
Cisco IOS XE Everest 16.5.1a	EtherChannel での SPAN 宛先ポートのサポート：EtherChannel で SPAN 宛先ポートを設定できるようにします。 この機能が導入されました。
Cisco IOS XE Everest 16.5.1a	スイッチ ポート アナライザ（SPAN） - 分散型出力 SPAN：ラインカードにすでに分散された入力 SPAN とともにラインカードに出力 SPAN 機能を分散させます。出力 SPAN 機能をラインカードに分散させることで、システムのパフォーマンスが向上します。 この機能が導入されました。



第 6 章

ERSPAN の設定

ERSPAN の設定

このモジュールは、Encapsulated Remote Switched Port Analyzer（ERSPAN）を設定する方法について説明します。Cisco ERSPAN 機能を使用すると、ポートまたは VLAN のトラフィックをモニタし、モニタされたトラフィックを宛先ポートに送信できます。

ERSPAN の設定の前提条件

- IPv4 配信/転送ヘッダーのみサポートされます。
- アクセス コントロール リスト（ACL）のフィルタは、トンネルにモニタ対象トラフィックを送信する前に適用されます。
- タイプ II ERSPAN ヘッダーのみサポートします。

ERSPAN 設定時の制約事項

この機能には、次の制限があります。

- 宛先セッションはサポートされません。
- デバイスは、最大 66 のセッションをサポートします。最大 8 つの送信元セッションを設定できます。残りのセッションは、RSPAN 宛先セッションとして設定できます。送信元セッションは、ローカル SPAN 送信元セッションまたは RSPAN 送信元セッションあるいは ERSPAN 送信元セッションのいずれかになります。
- 送信元としてポートのリストまたは VLAN のリストを設定できますが、特定のセッションに両方を設定することはできません。

- ERSPAN CLI を介してセッションが設定されると、セッション ID とセッション タイプは変更できません。これらを変更するには、コンフィギュレーション コマンドの no 形式を使用してセッションを削除してから、セッションを再設定する必要があります。
- ERSPAN 送信元セッションは、RSPAN VLAN を伝送する送信元トランク ポートからローカルに送信されたリモート SPAN（RSPAN）VLAN トラフィックをコピーしません。
- ERSPAN 送信元セッションは、ローカルに送信された ERSPAN GRE でカプセル化されたトラフィックを送信元ポートからコピーしません。

ERSPAN の設定に関する情報

ERSPAN の概要

Cisco ERSPAN 機能を使用すると、ポートまたは VLAN のトラフィックをモニタし、モニタされたトラフィックを宛先ポートに送信できます。ERSPAN は、スイッチ プローブ デバイスやリモート モニタリング（RMON）プローブなどのネットワーク アナライザにトラフィックを送信します。ERSPANは、異なるデバイス上のソース ポート、ソースVLAN、および宛先ポートをサポートして、ネットワーク上での複数のデバイスのリモート モニタリングを支援します。

ERSPAN は、最大 9180 バイトのカプセル化されたパケットをサポートします。ERSPAN は、ERSPAN 送信元セッション、ルーティング可能な ERSPAN GRE カプセル化トラフィック、および ERSPAN 宛先セッションで構成されています。

ERSPAN は、ERSPAN 送信元セッション、ルーティング可能な ERSPAN GRE カプセル化トラフィック、およびERSPAN宛先セッションで構成されています。ERSPAN送信元セッション、ERSPAN 宛先セッション、またはその両方をデバイスで設定できます。ERSPAN 送信元セッションだけが設定されたデバイスは、ERSPAN 送信元デバイスと呼ばれ、ERSPAN 宛先セッションだけが設定されたデバイスはERSPAN終端デバイスと呼ばれます。デバイスは、ERSPAN 送信元デバイスと終端デバイスの両方として機能できます。

送信元ポートまたは送信元 VLAN については、ERSPAN は、入力トラフィック、出力トラフィック、または入出力トラフィックを監視できます。デフォルトでは、ERSPANは、マルチキャストおよびブリッジ プロトコル データ ユニット（BPDU）フレームを含む、すべてのトラフィックを監視します。

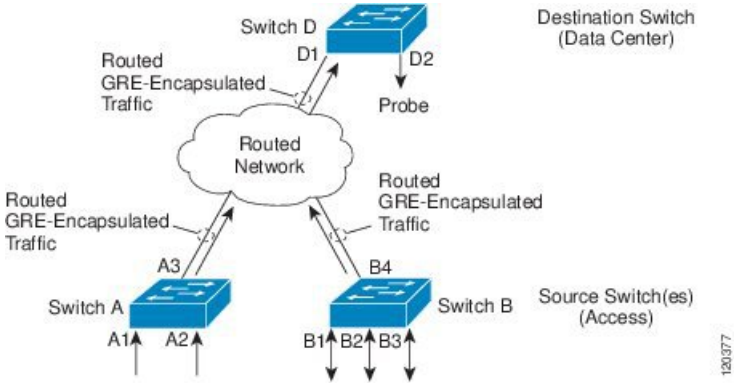
ERSPAN 送信元セッションは、次のパラメータによって定義されます。

- セッション ID
- セッションでモニタされる送信元ポートまたは送信元 VLAN の一覧
- キャプチャされたトラフィックの Generic Routing Encapsulation（GRE）エンベロープの宛先 IP アドレスおよび送信元 IP アドレスとしてそれぞれ使用される、宛先および元の IP アドレス
- ERSPAN フロー ID
- IP 有効時間（TTL）などの、GRE エンベロープに関連したオプション属性

（注） ERSPAN 送信元セッションは、ERSPAN GRE カプセル化されたトラフィックを送信元ポートからコピーしません。ERSPAN 送信元セッションごとに、送信元としてポートまたは VLAN を使用することはできますが、両方は使用できません。

（注） カプセル化はハードウェアで実行されるため、CPU パフォーマンスは影響を受けません。

図 *8 : ERSPAN* の設定



ERSPAN 送信元

Cisco ERSPAN 機能は次の送信元をサポートします。

- 送信元ポート：トラフィック分析のためにモニタされる送信元ポートです。任意の VLAN の送信元ポートを設定することができ、トランク ポートは、非トランク送信元ポートとともに送信元ポートとして設定できます。
- 送信元 VLAN：トラフィック分析のためにモニタされる VLAN です。

次のインターフェイスが送信元ポートとしてサポートされています。

- GigabitEthernet
- PortChannel
- TenGigabitEthernet

ERSPAN の設定方法

ERSPAN 送信元セッションの設定

ERSPAN送信元セッションは、モニタするセッション設定パラメータおよびポートまたはVLANを定義します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Switch> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**configure terminal** 例： `Switch# 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	**monitor session** *span-session-number***type erspan-source** 例： `Switch(config)# monitor session span-session-number type erspan-source`	セッション ID とセッション タイプを使用して ERSPAN 送信元セッションを定義し、ERSPAN のモニタ送信元セッション コンフィギュレーション モードを開始します。 • 送信元セッションまたは宛先セッションのセッション ID は同じグローバルな ID スペース内にあるため、各セッション ID は両方のセッション タイプに対してグローバルに一意です。 • *span-session-number* およびセッション タイプ（**erspan-source** キーワードによって設定）は、設定後は変更できません。セッションを削除するには、このコマンドの no 形式を使用し、新しいセッション ID または新しいセッション タイプでセッションを再作成します。
ステップ 4	**description** 説明 例： `Switch(config-mon-erspan-src)# description source1`	ERSPAN 送信元セッションの説明を入力します。
ステップ 5	**source** {**interface** *type number* | **vlan** *vlan-ID*} [**,** | **-**| **both** | **rx** | **tx**] 例： `Switch(config-mon-erspan-src)# source interface fastethernet 0/1 rx`	送信元インターフェイスまたは VLAN、およびモニタするトラフィックの方向を設定します。

	コマンドまたはアクション	目的
ステップ 6	**filter** {**ip access-group** {*standard-access-list* | *expanded-access-list* | *acl-name* } | **ipv6 access-group** *acl-name* | **mac access-group** *acl-name* | **vlan** *vlan-ID* [**,** **-**]} 例： `Switch(config-mon-erspan-src)# filter vlan 3`	（任意）ERSPAN 送信元がトランクポートである場合、送信元 VLAN フィルタリングを設定します。 • （注） 送信元 VLAN とフィルタ VLAN を同じセッションに含めることはできません。
ステップ 7	**no shutdown** 例： `Switch(config-mon-erspan-src)# no shutdown`	設定されたセッションのシャットダウンを無効にします。
ステップ 8	**destination** 例： `Switch(config-mon-erspan-src)# destination`	ERSPAN 宛先セッションを定義し、ERSPAN モニタ宛先セッション コンフィギュレーションモードを開始します。
ステップ 9	**ip address** *ip-address* 例： `Switch(config-mon-erspan-src-dst)# ip address 192.0.2.9`	ERSPAN 宛先セッションの IP アドレスを設定します。
ステップ 10	**erspan-id** *erspan-ID* 例： `Switch(config-mon-erspan-src-dst)# erspan-id 2`	ERSPAN トラフィックを識別するため、宛先セッションで使用される ID を設定します。
ステップ 11	**origin** *ip-address* 例： `Switch(config-mon-erspan-src-dst)# origin ip-address 203.0.113.2`	ERSPAN トラフィックの宛先として使用される IP アドレスを設定します。
ステップ 12	**ip ttl** *ttl-value* 例： `Switch(config-mon-erspan-src-dst)# erspan ttl 32`	ERSPAN トラフィックのパケットの存続可能時間（TTL）値を設定します。
ステップ 13	**end** 例： `Switch(config-mon-erspan-src-dst)# end`	ERSPAN モニタ宛先セッション 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。

ERSPAN の設定例

例：ERSPAN 送信元セッションの設定

```
Switch> enable
Switch# configure terminal
Switch(config)# monitor session 1 type erspan-source
Switch(config-mon-erspan-src)# description source1
Switch(config-mon-erspan-src)# source interface fastethernet 0/1 rx
Switch(config-mon-erspan-src)# filter vlan 3
Switch(config-mon-erspan-src)# no shutdown
Switch(config-mon-erspan-src)# destination
Switch(config-mon-erspan-src-dst)# ip address 192.0.2.9
Switch(config-mon-erspan-src-dst)# erspan-id 2
Switch(config-mon-erspan-src-dst)# origin ip-address 203.0.113.2
Switch(config-mon-erspan-src-dst)# ip ttl 32
Switch(config-mon-erspan-src-dst)# end
```

ERSPAN の確認

ERSPAN 設定を確認するには、次のコマンドを使用します。

次に、**showmonitorsessionerspan-source** コマンドの出力例を示します。

```
Switch# show monitor session erspan-source session

Type : ERSPAN Source Session
Status : Admin Enabled
Source Ports :
RX Only : Gi1/4/33
Destination IP Address : 192.0.2.1
Destination ERSPAN ID : 110
Origin IP Address : 10.10.10.216
IPv6 Flow Label : None
```

次に、**show monitor session erspan-source detail** コマンドの出力例を示します。

```
Switch# show monitor session erspan-source detail

Type : ERSPAN Source Session
Status : Admin Enabled
Description : -
Source Ports :
RX Only : Gi1/4/33
TX Only : None
Both : None
Source VLANs :
RX Only : None
TX Only : None
Both : None
Source RSPAN VLAN : None
Destination Ports : None
Filter VLANs : None
Filter Addr Type :
RX Only : None
TX Only : None
```

```
Both : None
Filter Pkt Type :
RX Only : None
Dest RSPAN VLAN : None
IP Access-group : None
IPv6 Access-group : None
Destination IP Address : 192.0.2.1
Destination IPv6 Address : None
Destination IP VRF : None
Destination ERSPAN ID : 110
Origin IP Address : 10.10.10.216
IP QOS PREC : 0
IP TTL : 255
```

次の **showcapabilityfeaturemonitor erspan-source** コマンドの出力は、設定された ERSPAN 送信元セッションに関する情報を表示しています。

```
Switch# show capability feature monitor erspan-source

ERSPAN Source Session Supported: true
No of Rx ERSPAN source session: 8
No of Tx ERSPAN source session: 8
ERSPAN Header Type supported: II
ACL filter Supported: true
Fragmentation Supported: true
Truncation Supported: false
Sequence number Supported: false
QOS Supported: true
```

次の **showcapabilityfeaturemonitorerspan-destination** コマンドの出力は、設定されたすべてのグローバル組み込みテンプレートを表示しています。

```
Switch# show capability feature monitor erspan-destination

ERSPAN Destination Session Supported: false
```

その他の参考資料

関連資料

関連項目	参照先
Cisco IOS コマンド	『Cisco IOS Master Commands List, All Releases』

RFC

標準/RFC	役職（Title）
RFC 2784	『Generic Routing Encapsulation (GRE)』

テクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Product Alert Tool（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

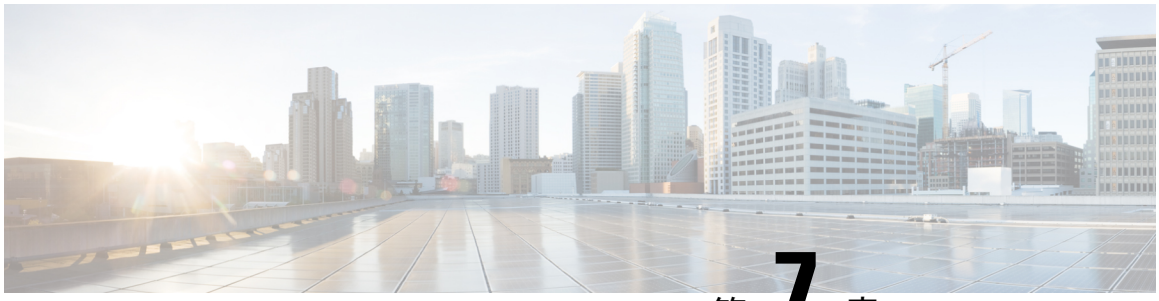
ERSPAN の設定に関する機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースのみを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

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

表 *8 : ERSPAN* の設定に関する機能情報

機能名（Feature Name）	リリース	機能情報
ERSPAN	Cisco IOS XE Everest 16.5.1a	この機能が導入されました



第 7 章

パケット キャプチャの設定

パケット キャプチャの前提条件

- パケット キャプチャは Cisco Catalyst 9500 シリーズ スイッチでサポートされています。
- Wireshark は Network Essentials または Network Advantage を実行しているスイッチのみでサポートされています。

組み込みパケット キャプチャ（EPC）のソフトウェア サブシステムは、その動作で CPU とメモリ リソースを消費します。さまざまなタイプの操作を行うために十分なシステム リソースを準備する必要があります。システム リソースを使用するためのガイドラインを以下の表に示します。

表 *9 : EPC* サブシステムのシステム要件

システム リソース	要件
ハードウェア（Hardware）	CPU 利用率の要件は、プラットフォームによって異なります。
メモリ	パケット バッファは DRAM に保存されます。パケット バッファのサイズは、ユーザが指定します。
ディスクスペース	パケットは外部のデバイスにエクスポートできます。フラッシュ ディスクでの中間保管は必要ありません。

パケット キャプチャの制約事項

- Wireshark でのグローバル パケット キャプチャはサポートされていません。
- 表示フィルタは、Wireshark でサポートされています。
- Wireshark を設定するための CLI では、機能を EXEC モードからのみ実行する必要があります。通常は設定サブモードで発生するアクション（キャプチャ ポイントの定義など）は、代わりに EXEC モードから処理されます。すべての主要コマンドは NVGEN の対象ではなく、NSF と SSO のシナリオではスタンバイ スーパーバイザに同期されません。
- インターフェイスの出力方向にキャプチャされたパケットは、 rewrite（TTL、VLAN タグ、CoS、チェックサム、MAC アドレス、DSCP、precedent、UP などを含む）によって加えられる変更を反映しない場合があります。
- 入力および出力の両方のパケットの書き換え情報はキャプチャされません。
- ファイル サイズによる循環ファイル保存の制限はサポートされません。
- ファイル制限は、Network Essentials と Network Advantage のフラッシュのサイズに限定されます。
- Control and Provisioning of Wireless Access Points（CAPWAP）などのプロトコルのデコードは、Network Essentials と Network Advantage でサポートされています。
- Network Essentials および Network Advantage では、ファイル モードにおいて、パケットはエクスポートされずにファイルに書き込まれます。
- 組み込み型の Wireshark はサポートされていますが、次の制限があります。
  - キャプチャ フィルタと表示フィルタはサポートされません。
  - アクティブなキャプチャの復号化は使用できません。
  - 出力形式は、以前のリリースとは異なります。
- 組み込みパケット キャプチャ（EPC）は、入力のマルチキャスト パケットのみをキャプチャし、出力の複製パケットはキャプチャしません。

設定の制限

- 最大 8 つのキャプチャ ポイントを定義できますが、一度にアクティブにできるのは 1 つだけです。1 つ開始するには 1 つ停止する必要があります。
- VRF、管理ポート、プライベート VLAN はいずれも接続ポイントとして使用することはできません。
- Wireshark クラス マップでは、1 つの ACL（IPv4、IPv6、MAC）のみが許可されます。
- Wireshark は宛先 SPAN ポートでパケットをキャプチャできません。

- Wireshark は、キャプチャ ポイントにアタッチされる接続ポイント（インターフェイス）のいずれかが動作を停止するとキャプチャを停止します。たとえば、接続ポイントに関連付けられているデバイスがから切断された場合です。キャプチャを再開するには、手動で再起動する必要があります。
- CPU 注入されたパケットは、コントロール プレーン パケットと見なされます。したがって、これらのタイプのパケットはインターフェイスの出力キャプチャではキャプチャされません。
- MAC フィルタは、MAC アドレスに一致しても IP パケットをキャプチャしません。これは、すべてのインターフェイス（L2 スイッチ ポート、L3 ルーテッド ポート）に適用されます。
- MAC ACL は、ARP などの非 IP パケットだけに使用されます。レイヤ 3 ポートまたは SVI ではサポートされません。
- MAC フィルタは、L3 インターフェイスと L2 パケット（ARP）をキャプチャすることはできません。
- IPv6 ベースの ACL は VACL ではサポートされません。
- レイヤ 2 EtherChannels はサポートされません。
- レイヤ 3 ポートチャネル サポートが使用できます。
- キャプチャがすでにアクティブである、または開始されている場合、キャプチャ ポイント パラメータを変更することはできません。
- ACL ロギングおよび Wireshark には互換性がありません。Wireshark はアクティブになると優先されます。任意のポートにロギング中の ACL にキャプチャされているものも含めたすべてのトラフィックが Wireshark にリダイレクトされます。Wireshark を開始する前に、ACL ロギングを非アクティブにすることをお勧めします。これを実行しないと、Wireshark のトラフィックは ACL ロギング トラフィックに汚染されます。
- Wireshark は floodblock によってドロップされるパケットをキャプチャしません。
- 同じポートの PACL および RACL の両方をキャプチャすると、1 つのコピーだけが CPU に送信されます。DTLS 暗号化 CAPWAP インターフェイスをキャプチャすると、暗号化されたものと復号化されたものの 2 つのコピーが Wireshark に送信されます。DTLS 暗号化 CAPWAP トラフィックを運ぶレイヤ 2 インターフェイスをキャプチャすると同じ動作が発生します。コア フィルタは外部 CAPWAP ヘッダーに基づいています。
- Cisco IOS XE Everest 16.5.1a 以降：
  - L3 ポート チャネルのサポートが追加されます。
  - 表示形式がマイナーチェンジされました。
  - cap ファイルのパケット数を表示する機能
  - キャプチャされたバッファをクリアすると、その内容とともにバッファも削除されます。パケット キャプチャがアクティブなときに実行することはできません。

- 追加の警告メッセージが、コントロール プレーンのキャプチャで表示されます。
- バッファ モードでは、パケットの表示は停止後のみに許可されます。
- Network Essentials および Network Advantage では停止時にパケット統計情報が表示されます。
- pcap ファイルでキャプチャされたパケット数を問い合わせる機能。
- 表示が cap ファイルからの場合、packet-number を使用して指定されたパケットの詳細を表示できます。
- 表示フィルタは、ファイル モードで使用可能です。
- パケット キャプチャの統計情報（受信またはドロップされたパケットおよびバイト）は、キャプチャ中またはキャプチャ停止後のいずれかに表示できます。
- システムは、Wireshark でサポートされるように、pcap/cap ファイルの内容に関する統計情報について問い合わせることができます。
- パケット キャプチャ セッションは、バッファのサイズに関係なく常にストリーミング モードです。ロックステップ モードは使用できません。

警告 コントロール プレーン パケットは、レート制限とパフォーマンスへの影響はありません。コントロールプレーン パケットのキャプチャを制限するフィルタを使用してください。

- ユーザがスイッチ ポートからルーテッド ポート（L2 > L3）へ、またはその逆へインターフェイスを変更した場合、インターフェイスが再び起動したときに、そのキャプチャ ポイントを削除し、新しいファイルを作成する必要があります。キャプチャ ポイントの停止/開始が機能しません。
- ユーザがアクティブなキャプチャ セッションで使用されたファイルを削除した場合、そのキャプチャ セッションは新しいファイルを作成できないため、キャプチャされたすべてのパケットが失われます。ユーザは、キャプチャ ポイントを再起動する必要があります。

パケット キャプチャの概要

パケット キャプチャ ツールの概要

パケットキャプチャ機能は、オンボードのパケットキャプチャ ファシリティです。ネットワーク管理者がデバイスを出入りするかデバイスを通るパケットをキャプチャすることで、パケットをローカルで分析したり、Wireshark や Embedded Packet Capture（EPC）のようなツールを使用するオフライン分析に向けてパケットを保存してエクスポートしたりできるようにするものです。この機能は、デバイスがネットワークの管理と操作にアクティブに参加できるようにす

ることによって、ネットワーク操作を簡略化します。この機能は、パケットの形式に関する情報を収集することによって、トラブルシューティングを容易にします。また、アプリケーションの分析とセキュリティも容易にします。

Wireshark を使用する Embedded Packet Capture は、Network Essentials および Network Advantage ライセンスでサポートされています。

Wireshark について

Wireshark の概要

Wireshark は、複数のプロトコルをサポートし、テキストベース ユーザ インターフェイスで情報を提供するパケット アナライザ プログラムです。

Wireshark は、.pcap と呼ばれる既知の形式を使用してファイルへパケットをダンプし、個々のインターフェイスに対して適用されイネーブルになります。EXEC モードでインターフェイスを指定し、フィルタおよび他のパラメータも指定します。Wireshark アプリケーションは、**start** コマンドを入力した場合にだけ適用され、Wireshark が自動または手動でキャプチャを停止した場合にだけ削除されます。

キャプチャ ポイント

キャプチャ ポイントとは、Wireshark 機能の一元的なポリシー定義です。キャプチャ ポイントは、どのパケットをキャプチャするか、どこからキャプチャするか、キャプチャ パケットに何を実行するか、およびいつ停止するかなど、Wireshark の特定のインスタンスに関連付けられたすべての特徴を説明します。キャプチャ ポイントは作成後に変更される場合があり、**start** コマンドを使用して明示的にアクティブ化しない限り、アクティブになりません。このプロセスは、キャプチャ ポイントのアクティブ化またはキャプチャ ポイントの開始といいます。キャプチャ ポイントは名前で識別され、手動または自動で非アクティブ化または停止する場合もあります。

複数のキャプチャ ポイントを定義してできますが、一度にアクティブにできるのは1つだけです。1 つ開始するには 1 つ停止する必要があります。

スタック構成のシステムの場合、キャプチャ ポイントはアクティブなメンバーによりアクティブ化されます。スイッチオーバーが発生すると、アクティブなすべてのパケット キャプチャ セッションが終了し、再起動する必要があります。

接続ポイント

接続ポイントは、キャプチャ ポイントに関連付けられた論理パケットのプロセス パスのポイントです。接続ポイントはキャプチャ ポイントの属性です。接続ポイントに影響するパケットはキャプチャ ポイント フィルタに対してテストされます。一致するパケットはキャプチャ ポイントの関連する Wireshark インスタンスにコピーされ、送信されます。特定のキャプチャ ポイントを複数の接続ポイントに関連付けることができます。異なるタイプ接続ポイントの混合に制限はありません。一部の制限は、異なるタイプの添付ポイントを指定すると適用されま

す。接続ポイントは、常に双方向であるレイヤ 2 VLAN の接続ポイントを除き、方向性あり（入力/出力/両方）です。

スタック型システムの場合では、すべてのスタック メンバの接続ポイントに有効です。EPC は定義されたすべての接続ポイントからパケットをキャプチャします。ただし、これらのパケットはアクティブ メンバーでのみに処理されます。

フィルタ

フィルタは、Wireshark にコピーされ、渡されるキャプチャ ポイントの接続ポイントを通過するトラフィックのサブセットを識別し制限するキャプチャ ポイントの属性です。Wireshark で表示されるためには、パケットは接続ポイントと、キャプチャ ポイントに関連付けられたすべてのフィルタも通過する必要があります。

キャプチャ ポイントには以下のタイプのフィルタがあります。

- コア システム フィルタ：コア システム フィルタはハードウェアによって適用され、一致基準はハードウェアによって制限されます。このフィルタは、ハードウェア転送トラフィックが Wireshark の目的でソフトウェアにコピーするかどうかを決定します。
- キャプチャ フィルタ：キャプチャ フィルタは、Wireshark によって適用されます。一致基準は、コア システム フィルタによってサポートされるものよりも詳細に表示されます。コア フィルタを通過するが、キャプチャ フィルタに失敗するパケットは CPU/ソフトウェアにコピーされ、送信されますが、Wireshark プロセスによって廃棄されます。キャプチャ フィルタの構文は、表示フィルタの構文と同じです。

（注） Cisco Catalyst 9500 シリーズ スイッチ の Wireshark はキャプチャ フィルタの構文を使用しません。

- 表示フィルタ：表示フィルタは、Wireshark によって適用されます。その一致基準はキャプチャ フィルタと似ています。表示フィルタに失敗したパケットは表示されません。

コア システム フィルタ

クラス マップまたは ACL を使用して、または CLI を使用して明示的にコア システム フィルタの一致基準を指定できます。

（注） CAPWAP を接続ポイントとして指定すると、コア システム フィルタは使用されません。

一部のインストール済み環境では、承認プロセスが長い場合さらに遅延を引き起こす可能性がある の設定を変更する権限を取得する必要があります。これにより、ネットワーク管理者の機能がトラフィックの監視および分析に制限される場合があります。この状況に対処するため、Wireshark は、EXEC モード CLI から、コア システム フィルタ一致基準の明示的な仕様をサポートします。この対処方法の欠点は、指定できる一致基準が、クラスマップがサポートす

る対象の限定的なサブセットである（MAC、IP 送信元アドレスおよび宛先アドレス、イーサネット タイプ、IP プロトコル、および TCP/UDP の発信元および宛先ポートなど）ことです。

コンフィギュレーション モードを使用する場合は ACL を定義するか、クラス マップでそこへキャプチャ ポイントを参照させることができます。明示的かつ ACL ベースの一致基準がクラス マップとポリシー マップの作成に内部的に使用されます。

注：ACL およびクラス マップの設定はシステムの一部であり、Wireshark 機能の側面ではありません。

Display Filter

表示フィルタを使用すると、.pcap ファイルからデコードして表示するときに表示するパケットの集合をさらに絞り込むように Wireshark に指示できます。

アクション（Actions）

Wireshark はライブ トラフィックまたは前の既存 .pcap ファイルで呼び出すことができます。ライブ トラフィックに対して起動されたとき、その表示フィルタを通過するパケットに対して次の 4 種類の処理を実行できます。

- デコード、分析、保存のためにメモリ内バッファへキャプチャ
- .pcap ファイルへ保存
- デコードおよび表示
- 保存および表示

.pcap ファイルのみに対して起動された場合は、デコードと表示の処理だけが適用できます。

キャプチャ パケットのメモリ内のバッファへのストレージ

パケットは、メモリ内のキャプチャ バッファに格納して、後でデコード、分析、または .pcap ファイルへ保存できます。

キャプチャ バッファは線形モードまたは循環モードを選択できます。線形モードでは、バッファが上限に達すると、新しいパケットが廃棄されます。循環モードでは、バッファが上限に達すると、新しいパケットを格納するために最も古いパケットが廃棄されます。必要に応じてバッファをクリアすることもできますが、このモードは、ネットワーク トラフィックのデバッグに主に使用されます。ただし、これを削除せずに、バッファの内容をクリアだけすることはできません。これを有効にするためには、現行のキャプチャを停止し、キャプチャをもう一度再起動します。

（注） パケットをバッファ内に保存する複数のキャプチャがある場合、メモリ ロスを避けるため、新しいキャプチャを開始する前にバッファをクリアしてください。

.pcap ファイルにキャプチャされたパケットのストレージ

（注） WireShark がスタック内のスイッチで使用される場合は、パケット キャプチャをアクティブ スイッチに接続されたフラッシュまたは USB フラッシュ デバイスにのみ保存できます。

たとえば、flash1 がアクティブなスイッチに接続されており、flash2 がセカンダリ スイッチに接続されている場合、flash1 にのみパケット キャプチャを保存できます。

アクティブ スイッチに接続されたフラッシュまたは USB フラッシュ デバイス以外のデバイスにパケット キャプチャを保存しようとすると、エラーが発生する場合があります。

Wireshark は .pcap ファイルにキャプチャされたパケットを保存できます。キャプチャ ファイルは次のストレージ デバイスに配置可能です。

- オンボード フラッシュ ストレージ（flash:）
- USB ドライブ(usbflash0:)

（注） サポートされていないデバイスまたはアクティブなスイッチに接続されていないデバイスにパケット キャプチャを保存しようとするとエラーが発生する可能性があります。

Wireshark のキャプチャ ポイントを設定する場合は、ファイル名を関連付けることができます。キャプチャ ポイントをアクティブにすると、Wireshark は指定された名前でファイルを作成し、パケットを書き込みます。キャプチャ ポイントの作成時にファイルがすでに存在する場合、Wireshark はファイルを上書きできるかどうかについて問い合わせます。キャプチャポイントの有効化時にファイルがすでに存在する場合、Wireshark は既存のファイルを上書きします。特定のファイル名には 1 つのキャプチャ ポイントのみ関連付けることができます。

Wireshark が書き込んでいるファイル システムが一杯になると、Wireshark はファイルの一部のデータで失敗します。そのため、キャプチャ セッションを開始する前に、ファイル システムに十分な領域があることを確認する必要があります。

パケット全体ではなくセグメントのみを保持して、必要な記憶域を減らすことができます。通常、最初の 64 または 128 バイトを超える詳細は不要です。デフォルトの動作は、パケット全体の保存です。

ファイルシステムを処理し、ファイルシステムへの書き込みを行う際、パケットのドロップの発生を避けるため、Wireshark ではオプションでメモリ バッファを使用してパケットの到着時に一時的に保持できます。メモリ バッファのサイズは、キャプチャ ポイントが .pcap ファイルに関連付けられる際に指定できます。

パケットのデコードおよび表示

Wireshark はコンソールにパケットをデコードして表示できます。この機能は、ライブ トラフィックに適用されるキャプチャ ポイントと前の既存 .pcap ファイルに適用されるキャプチャ ポイントで使用可能です。

（注） パケットをデコードして表示すると、CPU への負荷が高くなる場合があります。

Wireshark は、幅広い種類のパケット形式に対してパケット詳細をデコードおよび表示できます。詳細は、**monitor capture name start** コマンドを以下のキーワード オプション付きですることにより表示されます。これにより、表示およびデコード モードが開始します。

- brief : パケットごとに 1 行表示します（デフォルト）。
- detailed：プロトコルがサポートされているすべてのパケットのすべてのフィールドをデコードして表示します。詳細モードでは、他の 2 種類のモードよりも多くの CPU が必要です。
- (hexadecimal) dump：パケット データの 16 進ダンプおよび各パケットの印刷可能文字としてパケットごとに 1 行表示します。

capture コマンドをデコードおよび表示オプション付きで入力すると、Wireshark 出力が Cisco IOS に返され、変更なしでコンソールに表示されます。

ライブ トラフィックの表示

Wireshark はコア システムからパケットのコピーを受信します。Wireshark は、表示フィルタを適用して、不要なパケットを破棄し、残りのパケットをデコードおよび表示します。

.pcap ファイルの表示

Wireshark は、以前に保存された .pcap ファイルからのパケットをデコードして表示し、選択的にパケットを表示するように表示フィルタに指示できます。

パケットのストレージおよび表示

機能的には、このモードは以前の 2 種類のモードの組み合わせです。Wireshark は指定された .pcap ファイルにパケットを保存し、これらをコンソールにデコードおよび表示します。ここではコア フィルタだけが該当します。

Wireshark キャプチャ ポイントのアクティブ化および非アクティブ化

Wireshark のキャプチャ ポイントが、接続ポイント、フィルタ、アクション、およびその他のオプションで定義された場合、Wireshark をアクティブにする必要があります。キャプチャ ポイントがアクティブになるまで、実際にパケットをキャプチャしません。

キャプチャ ポイントがアクティブになる前に、一部の機能性チェックが実行されます。キャプチャ ポイントは、コア システム フィルタと接続ポイントのどちらも定義されていない場合はアクティブにできません。これらの要件を満たしていないキャプチャ ポイントをアクティブ化しようとすると、エラーが生成されます。

表示フィルタを、必要に応じて指定します。

Wireshark のキャプチャ ポイントはアクティブになると、複数の方法で非アクティブにできます。.pcapファイルにパケットを格納するだけのキャプチャ ポイントは手動で停止することも、また時間制限またはパケット制限付きで設定することもでき、その後でキャプチャ ポイントは自動的に停止します。

Wireshark のキャプチャ ポイントがアクティブになると、固定レート ポリサーがハードウェアに自動的に適用され、CPU が Wireshark によって指示されたパケットでフラッディングしないようになります。レートポリサーの短所は、リソースが使用可能な場合でも、確立されたレートを超えて連続するパケットをキャプチャできないことです。

パケット キャプチャ設定レートは、1 秒あたり 1000 パケット（pps）です。1000 pps の制限は、すべての接続ポイントの合計に適用されます。たとえば、3つの接続ポイントにキャプチャ セッションがあれば、3 つの接続ポイントすべてのレートの合計が 1000 pps にポリシングされます。

（注） ポリサーは、コントロール プレーン パケット キャプチャではサポートされていません。コントロール プレーン キャプチャ ポイントを有効化するときは、CPU があふれないよう慎重に行う必要があります。

Wireshark 機能

ここでは、Wireshark 機能が 環境でどのように動作するかについて説明します。

- ポート セキュリティおよびWiresharkが入力キャプチャに適用された場合でも、ポート セキュリティによってドロップされたパケットは Wireshark でキャプチャされます。ポート セキュリティが入力キャプチャに適用され、Wireshark が出力キャプチャに適用された場合、ポート セキュリティによってドロップされたパケットは Wireshark ではキャプチャされません。
- ダイナミック ARPインスペクション（DAI）によってドロップされたパケットはWireshark ではキャプチャされません。
- STP ブロック ステートのポートが接続ポイントとして使用され、コア フィルタが一致する場合、Wireshark は、パケットがスイッチにドロップされる場合でもポートに入ってくるパケットをキャプチャします。
- 分類ベースのセキュリティ機能：入力分類ベースのセキュリティ機能によってドロップされたパケット（ACL およびIPSG など）は同じ層の接続ポイントに接続する Wireshark キャプチャ ポイントでは検出されません。一方、出力分類ベースのセキュリティ機能によってドロップされたパケットは、同じ層の接続ポイントに接続されている Wireshark のキャプチャ ポイントでキャッチされます。論理モデルは、Wireshark の接続ポイントが、入力側のセキュリティ機能のルックアップ後、および出力側のセキュリティ機能のルックアップ前に発生することです。

  入力では、パケットはレイヤ 2 ポート、VLAN、およびレイヤ 3 ポート/SVI を介して送信されます。出力では、パケットはレイヤ 3 ポート/SVI、VLAN、およびレイヤ 2 ポートを介して送信されます。接続ポイントがパケットがドロップされるポイントの前にある場

合、Wireshark はパケットをキャプチャします。これ以外の場合は、Wireshark はパケットをキャプチャしません。たとえば、入力方向のレイヤ2接続ポイントに接続されるWiresharkのキャプチャ ポリシーはレイヤ3分類ベースのセキュリティ機能によってドロップされたパケットをキャプチャします。対照的に、出力方向のレイヤ 3 接続ポイントに接続するWiresharkのキャプチャ ポリシーは、レイヤ2分類ベースのセキュリティ機能によりドロップされたパケットをキャプチャします。

- ルーテッドポートおよびスイッチ仮想インターフェイス（SVIs）：SVIの出力から送信されるパケットは CPU で生成されるため、Wireshark は SVI の出力をキャプチャできません。これらのパケットをキャプチャするには、コントロールプレーンを接続ポイントとして含めます。
- VLAN：Cisco IOS リリース 16.1 以降、VLAN が Wireshark の接続ポイントとして使用されている場合、パケット キャプチャは、入力方向と出力方向の両方の L2 と L3 でサポートされます。
- リダイレクション機能：入力方向では、レイヤ3（PBRおよびWCCPなど）でリダイレクトされる機能トラフィックは、レイヤ 3 の Wireshark の接続ポイントよりも論理的に後です。Wireshark は、後で別のレイヤ 3 インターフェイスにリダイレクトされる可能性がある場合でも、これらのパケットをキャプチャします。対照的に、レイヤ3によってリダイレクトされる出力機能（出力WCCPなど）は論理的にレイヤ3接続ポイントの前にあり、Wireshark ではキャプチャされません。
- SPAN：Wireshark は、SPAN 宛先として設定されたインターフェイスでパケットをキャプチャできません。
- SPAN：Wireshark は、入力方向の SPAN 送信元として設定されたインターフェイスでパケットをキャプチャできます。出力方向でも使用できる可能性があります。
- ACL が適用されていない場合、最大 1000 の VLAN からパケットを一度にキャプチャできます。ACL が適用されている場合、Wireshark の使用できるハードウェア領域はより少なくなります。結果として、パケット キャプチャに一度に使用できる VLAN の最大数は低くなります。1000 以上の VLAN トンネルを一度に使用したり、ACL を多数使用すると予測されない結果が生じる可能性があります。たとえば、モビリティがダウンする可能性があります。

（注） 過剰なCPU使用につながり、予測されないハードウェア動作の原因となる可能性があるため、過剰な数の接続ポイントを一度にキャプチャしないことを強くお勧めします。

Wireshark のガイドライン

- Wireshark でのパケット キャプチャ中に、ハードウェア転送が同時に発生します。
- Wireshark のキャプチャ プロセスを開始する前に、CPU 使用率が妥当であり、十分なメモリ（少なくとも 200 MB）が使用可能であることを確認します。

- ストレージ ファイルにパケットを保存する予定の場合、Wireshark キャプチャ プロセスを開始する前に十分なスペースが利用可能であることを確認してください。
- Wireshark のキャプチャ中の CPU 使用率は、設定された基準に一致するパケットの数と、一致したパケット用のアクション（ストア、デコードして表示、あるいはこの両方）によって異なります。
- 高 CPU 使用率および他の不要な条件を避けるため、可能な限りキャプチャを最小限に抑えてください（パケット、期間による制限）。
- パケット転送はハードウェアで通常実行されるため、パケットは、ソフトウェア処理のために CPU にコピーされません。Wireshark のパケット キャプチャの場合、パケットは CPU にコピーされ、配信されて、これが CPU 使用率の増加につながります。

  CPU 使用率を高くしないようにするには、次の手順を実行します。

  - 関連ポートだけに接続します。
  - 一致条件を表すにはクラス マップを使用し、二次的にアクセス リストを使用してください。いずれも実行可能でない場合は、明示的な、インライン フィルタを使用します。
  - フィルタ規則に正しく準拠させます。緩和されたのではなく制限的な ACL で、トラフィック タイプを（IPv4 のみなどに）制限して、不要なトラフィックを引き出します。

- パケット キャプチャを短い期間または小さなパケット番号に常に制限します。capture コマンドのパラメータにより、次を指定することができます。
  - Capture duration
  - キャプチャされたパケットの数
  - ファイルサイズ（File size）
  - パケットのセグメント サイズ
- コア フィルタと一致するトラフィックが非常に少ないことが判明している場合は、制限なしでキャプチャ セッションを実行します。
- 次の場合に高い CPU（またはメモリ）使用率になる可能性があります。
  - キャプチャ セッションをイネーブルにし長期間不在のままにして、予期しないトラフィックのバーストが起きた場合。
  - リング ファイルまたはキャプチャ バッファを使用してキャプチャ セッションを起動して、長期間不在のままにすると、パフォーマンスまたはシステム ヘルスの問題が引き起こされます。
- キャプチャ セッション中に、のパフォーマンスやヘルスに影響する可能性のある Wireshark による高い CPU 使用率およびメモリ消費がないか監視します。こうした状況が発生した場合、Wireshark セッションをすぐに停止します。

- 大きなファイルの .pcap ファイルからのパケットをデコードして表示することは避けてください。代わりに、PC に .pcap ファイルを転送し PC 上で Wireshark を実行します。
- Wireshark インスタンスは最大 8 個まで定義できます。 .pcap ファイルまたはキャプチャ バッファからパケットをデコードして表示するアクティブな **show** コマンドは、1 個のインスタンスとしてカウントされます。ただし、アクティブにできるインスタンスは1つだけです。
- 実行中のキャプチャに関連付けられた ACL が変更された場合は常に、ACL 変更を有効にするにはキャプチャを再起動する必要があります。キャプチャを再起動しないと、変更前の元の ACL が継続して使用されます。
- パケット損失を防ぐには、次の点を考慮します。
  - ライブ パケットをキャプチャしている間は、CPU に負荷のかかる操作であるデコードと表示ではなく（特に detailed モードの場合）、保存のみを使用します（display オプションを指定しない場合）。
  - パケットをバッファ内に保存する複数のキャプチャがある場合、メモリ ロスを避けるため、新しいキャプチャを開始する前にバッファをクリアしてください。
  - デフォルト バッファ サイズを使用し、パケットが失われている場合、バッファ サイズを増加してパケットの喪失を防ぐことができます。
  - フラッシュ ディスクへの書き込みは、CPU に負荷のかかる操作であるため、キャプチャ レートが不十分な場合、バッファ キャプチャの使用をお勧めします。
  - Wireshark キャプチャ セッションは 1000 pps のレートで常にストリーミング モードで動作します。
  - ストリーミング キャプチャ モードは約 1000 pps をサポートし、ロックステップ モードは約2 Mbps（256バイト パケットで測定）をサポートします。一致するトラフィック レートがこの値を超えると、パケット損失が発生する可能性があります。
- ストリーミング キャプチャ モードのレートは 1000 pps です。
- コンソール ウィンドウのライブ パケットをデコードして表示する場合は、Wireshark セッションが短いキャプチャ期間によって抑制されていることを確認します。

警告 期間制限がより長いまたはキャプチャ期間がない（**term len 0** コマンドを使用して auto-more サポートのない端末を使用した）Wireshark セッションでは、コンソールまたは端末が使用できなくなる場合があります。

- 高 CPU 使用率につながるライブ トラフィックのキャプチャに Wireshark を使用している場合、QoS ポリシーを一時的に適用して、キャプチャ プロセスが終了するまで実際のトラフィックを制限することを考慮してください。
- すべての Wireshark 関連のコマンドは EXEC モードで、コンフィギュレーション コマンドは、Wireshark にありません。

Wireshark CLI でアクセス リストまたはクラス マップを使用する必要がある場合は、コンフィギュレーション コマンドでアクセス リストおよびクラス マップを定義する必要があります。

- 特定の順序はキャプチャ ポイントを定義する場合には適用されません。CLI で許可されている任意の順序でキャプチャ ポイント パラメータを定義できます。Wireshark CLI では、単一行のパラメータ数に制限はありません。これはキャプチャ ポイントを定義するために必要なコマンドの数を制限します。
- 接続ポイントを除くすべてのパラメータは、単一の値を取ります。通常、コマンドを再入力することにより、値を新しいものに置き換えることができます。ユーザの確認後にシステムが新しい値を受け入れ、古い値を上書きします。コマンドの **no** 形式は、新しい値の入力には必要はありませんが、パラメータの削除には必要です。
- Wireshark では 1 つ以上の接続ポイントを指定することができます。複数の接続ポイントを追加するには、新しい接続ポイントでコマンドを再入力します。接続ポイントを削除するには、コマンドの **no** 使用します。接続ポイントとしてインターフェイス範囲を指定できます。たとえば、monitor capture mycap interface gi 3/1 in**monitor capture mycap interface GigabitEthernet1/0/1 in** を入力します。ここで、interface gi 3/1GigabitEthernet1/0/1 は接続ポイントです。

  またインターフェイス GigabitEthernet1/0/2 にも接続する必要がある場合、次のように、別の行で指定します。

  monitor capture mycap interface GigabitEthernet1/0/2 in
- キャプチャがアクティブなときは、キャプチャに対する変更を行うことはできません。
- 実行する処理は、いずれのパラメータが必須であるかを決定します。Wireshark CLI では、**start** コマンドを入力する前に任意のパラメータを指定または変更することができます。**start** コマンドを入力すると、すべての必須パラメータが入力されたと判断した後にのみ Wireshark が開始します。
- キャプチャ ポイントの作成時にファイルがすでに存在する場合、Wireshark はファイルを上書きできるかどうかについて問い合わせます。キャプチャポイントの有効化時にファイルがすでに存在する場合、Wireshark は既存のファイルを上書きします。
- コア フィルタは明示的なフィルタ、アクセス リスト、またはクラス マップにできます。これらのタイプの新しいフィルタを指定すると、既存のものを置き換えます。

  （注） コア フィルタは、CAPWAP トンネル インターフェイスをキャプチャ ポイントの接続ポイントとして使用している場合を除き、必須です。
- 明示的な **stop** コマンドを使用するか、automore モードに **q** を入力して、Wireshark のセッションを終了します。セッションは、期間やパケット キャプチャの制限などの停止の条件が満たされたときに、自動的に終了します。

- ドロップされたパケットはキャプチャの最後に表示されません。ただし、ドロップされたサイズ超過のパケット数のみが表示されます。

デフォルトの Wireshark の設定

次の表は、デフォルトの Wireshark の設定を示しています。

機能	デフォルト設定
時間（Duration）	No limit
パケット（Packets）	No limit
パケット長	制限なし（フル パケット）
ファイルサイズ（File size）	No limit
リング ファイル ストレージ	なし
バッファのストレージ モード	線形

組み込みパケット キャプチャについて

組み込みパケット キャプチャの概要

組み込みパケットキャプチャ（EPC）は、パケットのトレースとトラブルシューティングに役立つ組み込みシステム管理機能を提供します。この機能を使用すると、ネットワーク管理者は、シスコ デバイスを出入りするか通過するデータ パケットをキャプチャできます。ネットワーク管理者は、キャプチャ バッファ サイズとタイプ（循環またはリニア）およびキャプチャする各パケットの最大バイト数を定義する場合があります。パケット キャプチャ レートは、詳細な管理制御を使用してスロットリングできます。たとえば、アクセス コントロール リストを使用してキャプチャ対象パケットをフィルタリングするオプションや、最大パケットキャプチャ レートまたはサンプリング間隔の指定などの詳細な定義を行うオプションが利用できます。

組み込みパケット キャプチャの利点

- デバイスで IPv4 および IPv6 パケットをキャプチャでき、MAC フィルタを使用したり、MAC アドレスをマッチさせたりして、非 IP パケットもキャプチャ可能。
- パケット キャプチャ ポイントをイネーブルにする拡張可能なインフラストラクチャキャプチャ ポイントは、パケットがキャプチャされ、バッファと関連付けられるトラフィック トランジット ポイントです。
- 外部ツールを使用した分析に適したパケット キャプチャ ファイル（PCAP）形式でパケット キャプチャをエクスポートする機能。
- さまざまな詳細レベルでキャプチャされたデータ パケットをデコードする方法。

パケット データ キャプチャ

パケット データ キャプチャは、バッファに格納されるデータ パケットのキャプチャです。パケット データ キャプチャは、一意の名前とパラメータを入力することによって定義します。

こうしたキャプチャでは、次のアクションを実行できます。

- インターフェイスでのキャプチャのアクティブ化。
- キャプチャ ポイントへのアクセス コントロール リスト（ACL）やクラス マップの適用。

（注） Network Based Application Recognition（NBAR）と MAC スタイルのクラス マップは、サポートされていません。

- キャプチャの破棄。
- サイズやタイプなどのバッファ ストレージ パラメータの指定。サイズの範囲は 1 ～ 100 MB です。デフォルトのバッファは線形です。もう 1 つのバッファ オプションは循環です。
- プロトコル、IP アドレス、ポート アドレスに関する情報を含む一致基準の指定。

パケット キャプチャの設定

Wireshark の設定方法

Wireshark を設定するには、次の基本的な手順を実行します。

1. キャプチャ ポイントを定義します。
2. キャプチャ ポイントのパラメータを追加または変更します。
3. キャプチャ ポイントをアクティブ化または非アクティブ化します。
4. キャプチャ ポイントを今後使用しない場合は削除します。

キャプチャ ポイントの定義

この手順の例では、非常にシンプルなキャプチャ ポイントを定義します。必要に応じて、**monitor capture** コマンドの 1 つのインスタンスを使用してキャプチャ ポイントとそのすべてのパラメータを定義できます。

（注） 接続ポイント、キャプチャの方向、およびコア フィルタが機能するキャプチャ ポイントを持つよう定義する必要があります。

コア フィルタを定義する必要がないのは、CAPWAP トンネリング インターフェイスを使用してワイヤレス キャプチャ ポイントを定義する場合です。この場合、コア フィルタは定義しません。これは使用できません。

キャプチャ ポイントを定義するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**monitor capture** {*capture-nameinterface-typeinterface-id*}{**interface** | **control-plane**}{**in** | **out** | **both**} 例： `Device# monitor capture mycap interface GigabitEthernet1/0/1 in`	キャプチャ ポイントを定義し、キャプチャ ポイントが関連付けられている接続ポイントを指定し、キャプチャの方向を指定します。 キーワードの意味は次のとおりです。 • *capture-name*：定義するキャプチャ ポイントの名前を指定します（例では mycap が使用されています）。キャプチャ名の長さは 8 文字以下にしてください。英数字、アンダースコア（_）のみが許可されます •（任意）**interface** *interface-type interface-id*：キャプチャ ポイントが関連付けられる接続ポイントを指定します（例では GigabitEthernet1/0/1 が使用されています）。

	コマンドまたはアクション	目的
		（注） オプションで、このコマンド インスタンス 1 つでこのキャプチャ ポイントの複数の接続ポイントおよびパラメータすべてを定義できます。これらのパラメータについては、キャプチャ ポイント パラメータの変更に関する手順で説明されています。範囲のサポートは、接続ポイントを追加および削除するためにも使用できます。 *interface-type* には次のいずれかを使用します。 • **GigabitEthernet**：接続ポイントを GigabitEthernet として指定します。 • **vlan**：接続ポイントを VLAN として指定します。 （注） このインターフェイスを接続ポイントとして使用する場合は、入力キャプチャのみが可能です。 • **capwap**：接続ポイントを CAPWAP トンネルとして指定します。 （注） このインターフェイスを接続ポイントとして使用すると、コア フィルタは使用できません。 • （任意）**control-plane**：接続ポイントとしてコントロール プレーンを指定します。

	コマンドまたはアクション	目的
		• **in**	**out**	**both**：キャプチャの方向を指定します。
ステップ 3	**monitor capture** {*capture-name*}[**match** {**any**	**ipv4 any any**	**ipv6**} **any any**}] 例： `Device# monitor capture mycap interface GigabitEthernet1/0/1 in match any`	コア システムのフィルタを定義します。 （注） コア フィルタが使用できなくなるため、CAPWAP のトンネリング インターフェイスを接続ポイントとして使用する場合はこの手順を実行しないでください。 キーワードの意味は次のとおりです。 • *capture-name*：定義するキャプチャ ポイントの名前を指定します（例では mycap が使用されています）。 • **match**：フィルタを指定します。定義されている最初のフィルタはコア フィルタです。 （注） キャプチャ ポイントは、コア システム フィルタと接続ポイントのどちらも定義されていない場合はアクティブにできません。これらの要件を満たしていないキャプチャ ポイントをアクティブ化しようとすると、エラーが生成されます。 • **ipv4**：IP バージョン 4 のフィルタを指定します。 • **ipv6**：IP バージョン 6 のフィルタを指定します。

ステップ 4	**show monitor capture** {*capture-name*}[**parameter**]
	例：
	`Device# show monitor capture mycap parameter`
	`monitor capture mycap interface GigabitEthernet1/0/1 in`
	`monitor capture mycap match any`
		ステップ 2 で定義したキャプチャ ポイント パラメータを表示し、キャプチャ ポイントを定義したことを確認します。

	コマンドまたはアクション	目的
ステップ 5	**show capwap summary** 例： `Device# show capwap summary`	ワイヤレス キャプチャの接続ポイントとして使用できる CAPWAP トンネルを表示します。 （注） このコマンドは、ワイヤレス キャプチャを実行するために CAPWAP トンネルを接続ポイントとして使用している場合にのみ使用します。例の項の CAPWAP の例を参照してください。
ステップ 6	**show running-config** 例： `Device# show running-config`	入力を確認します。
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。

例

CAPWAP 接続ポイントでキャプチャ ポイントを定義するには次を実行します。

```
Device# show capwap summary

CAPWAP Tunnels General Statistics:
  Number of Capwap Data Tunnels       = 1
  Number of Capwap Mobility Tunnels   = 0
  Number of Capwap Multicast Tunnels  = 0


Name   APName                           Type PhyPortIf Mode      McastIf
------ -------------------------------- ---- --------- --------- -------
Ca0    AP442b.03a9.6715                 data Gi3/0/6   unicast   -


Name   SrcIP           SrcPort DestIP          DstPort DtlsEn MTU   Xact
------ --------------- ------- --------------- ------- ------ ----- ----
Ca0    10.10.14.32     5247    10.10.14.2      38514   No     1449  0

Device# monitor capture mycap interface capwap 0 both
Device# monitor capture mycap file location flash:mycap.pcap
Device# monitor capture mycap file buffer-size 1
Device# monitor capture mycap start

*Aug 20 11:02:21.983: %BUFCAP-6-ENABLE: Capture Point mycap enabled.on
```

```
Device# show monitor capture mycap parameter
   monitor capture mycap interface capwap  0 in
   monitor capture mycap interface capwap  0 out
   monitor capture mycap file location flash:mycap.pcap buffer-size 1
Device#
Device# show monitor capture mycap

Status Information for Capture mycap
  Target Type:
  Interface: CAPWAP,
    Ingress:
 0
    Egress:
 0
   Status : Active
  Filter Details:
    Capture all packets
  Buffer Details:
   Buffer Type: LINEAR (default)
  File Details:
   Associated file name: flash:mycap.pcap
   Size of buffer(in MB): 1
  Limit Details:
   Number of Packets to capture: 0 (no limit)
   Packet Capture duration: 0 (no limit)
   Packet Size to capture: 0 (no limit)
   Packets per second: 0 (no limit)
   Packet sampling rate: 0 (no sampling)
Device#
Device# show monitor capture file flash:mycap.pcap
  1   0.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
  2   0.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
  3   2.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
  4   2.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
  5   3.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
  6   4.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
  7   4.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
  8   5.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
  9   5.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
 10   6.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
 11   8.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
 12   9.225986   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 13   9.225986   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 14   9.225986   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 15   9.231998   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 16   9.231998   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 17   9.231998   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 18   9.236987   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 19  10.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
 20  10.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
```

```
 21  12.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
 22  12.239993   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 23  12.244997   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 24  12.244997   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 25  12.250994   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 26  12.256990   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 27  12.262987   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 28  12.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
 29  12.802012   10.10.14.3 -> 10.10.14.255 NBNS Name query NB WPAD.<00>
 30  13.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
```

次のタスク

さらなる接続ポイントを追加して、キャプチャ ポイントのパラメータを変更し、アクティブ化できます。または、キャプチャ ポイントをそのまま使用したい場合はすぐにアクティブ化することもできます。

（注） このトピックで説明されているメソッドを使用してキャプチャ ポイントのパラメータを変更することはできません。

ユーザが誤ったキャプチャ名、または無効/存在しない接続ポイントを入力すると、スイッチは、「Capture Name should be less than or equal to 8 characters.Only alphanumeric characters and underscore (_) is permitted」および「% Invalid input detected at '^' marker」のようなエラーを表示します。

キャプチャ ポイント パラメータの追加または変更

パラメータの値を指定する手順は、順番にリストされますが、任意の順序で実行できます。1行、2行、または複数行で指定できます。複数指定が可能な接続ポイントを除き、同じオプションを再定義することで、任意の値をより最近の値に置き換えることができます。すでに指定された特定のパラメータが変更されている場合は、インタラクティブに確認する必要があります。

キャプチャ ポイントのパラメータを変更するには、次の手順を実行します。

始める前に

以下の手順を実行する前にキャプチャ ポイントを定義する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**monitor capture** {*capture-namemac-match-string*} **match** {**any** | **mac** | **ipv4** {**any** | **host** | **protocol**}{**any** | **host**} | **ipv6** {**any** | **host** | **protocol**}{**any** | **host**}} 例： `Device# monitor capture mycap match ipv4 any any`	明示的に、または ACL を介して、またはクラス マップを介して定義されたコア システム フィルタ（**ipv4 any any**）を定義します。
ステップ 3	**monitor capture** {*capture-namesecondssizenum*} **limit** {[**duration**][**packet-length**][**packets**]} 例： `Device# monitor capture mycap limit duration 60 packet-len 400`	秒単位のセッション制限（60）、キャプチャされたパケット、または Wireshark によって保持されるパケット セグメント長（400）を指定します。
ステップ 4	**monitor capture** {*capture-name*} **file** {**location** *filename*} 例： `Device# monitor capture mycap file location flash:mycap.pcap`	キャプチャ ポイントがパケットを表示するだけでなくキャプチャできるようにする場合は、ファイルのアソシエーションを指定します。 （注） すでにファイルが存在する場合、それが上書きが可能かどうかを確認する必要があります。
ステップ 5	**monitor capture** {*capture-namesize*} **file** {**buffer-size** } 例： `Device# monitor capture mycap file buffer-size 100`	トラフィック バーストの処理に Wireshark で使用されるメモリ バッファのサイズを指定します。

ステップ 6	**show monitor capture** {*capture-name*}[**parameter**]
	例：
	`Device# show monitor capture mycap parameter`
	`monitor capture mycap interface`
		以前に定義したキャプチャ ポイント パラメータを表示します。

	コマンドまたはアクション	目的
	GigabitEthernet1/0/1 in monitor capture mycap match ipv4 any any monitor capture mycap limit duration 60 packet-len 400 monitor capture point mycap file location bootdisk:mycap.pcap monitor capture mycap file buffer-size 100	
ステップ 7	**end** 例： Device(config)# **end**	特権 EXEC モードに戻ります。

例

パラメータの変更

キャプチャ ファイルの関連付けまたは関連付け解除

```
Device# monitor capture point mycap file location flash:mycap.pcap
Device# no monitor capture mycap file
```

パケット バーストの処理にメモリ バッファ サイズを指定する

```
Device# monitor capture mycap buffer size 100
```

IPv4 と IPv6 の両方に一致するように、明示的なコア システム フィルタを定義する

```
Device# monitor capture mycap match any
```

次のタスク

キャプチャ ポイントに必要なパラメータがすべて含まれている場合はアクティブ化します。

キャプチャ ポイント パラメータの削除

順番に表示されていますが、パラメータを削除する手順は任意の順序で実行できます。1 行、2 行、または複数行で削除できます。複数が可能な接続ポイントを除いて、任意のパラメータを削除できます。

キャプチャ ポイントのパラメータを削除するには、次の手順を実行します。

始める前に

キャプチャ ポイント パラメータは、以下の手順を使用して削除する前に定義する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**no monitor capture** {*capture-name*} **match** 例： `Device# no monitor capture mycap match`	キャプチャ ポイント（mycap）で定義されているすべてのフィルタを削除します。

ステップ 3	**no monitor capture** {*capture-name*} **limit** [**duration**] [**packet-length**] [**packets**]
	例：
	`Device# no monitor capture mycap limit duration packet-len`
	`Device# no monitor capture mycap limit`
		Wireshark によって保持されるセッション タイム制限およびパケット セグメント長を削除します。その他の指定された制限はそのままになります。
		Wireshark のすべての制限をクリアします。

ステップ 4	**no monitor capture** {*capture-name*} **file** [**location**] [**buffer-size**]
	例：
	`Device# no monitor capture mycap file`
	`Device# no monitor capture mycap file location`
		ファイルの関連付けを削除します。キャプチャポイントはパケットをキャプチャしなくなります。表示だけが実行されます。
		ファイル位置の関連付けを削除します。ファイル位置はキャプチャ ポイントとは関連付けられなくなります。ただし、他の定義されたファイル関連付けはこのアクションによっては影響を受けません。

ステップ 5	**show monitor capture** {*capture-name*} [**parameter**]
	例：
	`Device# show monitor capture mycap parameter`
	`monitor capture mycap interface GigabitEthernet1/0/1 in`
		パラメータの削除操作後にまだ定義されているキャプチャ ポイント パラメータを表示します。このコマンドは、キャプチャポイントと関連付けられるパラメータを確認するために手順の任意の地点で実行できます。

ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

次のタスク

キャプチャ ポイントに必要なパラメータがすべて含まれている場合はアクティブ化します。

（注） キャプチャ ポイントがアクティブなときにパラメータが削除されると、スイッチは「キャプチャがアクティブです（Capture is active）」というエラーを表示します。

キャプチャ ポイントの削除

キャプチャ ポイントを削除するには、次の手順を実行します。

始める前に

キャプチャ ポイントは、以下の手順を使用して削除する前に定義する必要があります。削除する前に、キャプチャ ポイントを停止する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**no monitor capture** {*capture-name*} 例： `Device# no monitor capture mycap`	指定されたキャプチャ ポイント（mycap）を削除します。

ステップ 3	**show monitor capture** {*capture-name*}[**parameter**]
	例：
	`Device# show monitor capture mycap parameter`
	`Capture mycap does not exist`
		指定されたキャプチャ ポイントは削除されたため存在しないことを示すメッセージを表示します。

ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例：	入力を確認します。

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

削除したものと同じ名前の新規キャプチャ ポイントを定義できます。これらの手順は通常、キャプチャ ポイントの定義をやり直したい場合に実行します。

キャプチャ ポイントをアクティブまたは非アクティブにする

キャプチャ ポイントをアクティブまたは非アクティブにするには、次の手順を実行します。

始める前に

接続ポイントおよびコア システム フィルタが定義され、関連付けられたファイル名がすでに存在する場合でも、キャプチャ ポイントはアクティブ化することができます。このようなケースでは、既存のファイルは上書きされます。

関連するファイル名のないキャプチャ ポイントは、表示するためだけにアクティブにできます。ファイル名が指定されていないと、パケットはバッファに保管されます。ライブ表示（キャプチャ時の表示）は、ファイルおよびバッファ モードの両方で使用できます。

表示フィルタを指定しない場合、パケットはライブ表示されず、コア システム フィルタによってキャプチャされたすべてのパケットが表示されます。デフォルトの表示モードはbriefです。

（注） CAPWAP のトンネリング インターフェイスを接続ポイントとして使用すると、コア フィルタは使用されないため、この場合は定義する必要はありません。

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 モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。

	コマンドまたはアクション	目的
ステップ 2	**monitor capture** {*capture-name*} **start**[**display** [**display-filter** *filter-string*]][**brief** | **detailed** | **dump**] 例： `Device# monitor capture mycap start display display-filter "stp"`	キャプチャポイントをアクティブ化し、「stp」を含むパケットだけが表示されるように表示をフィルタします。
ステップ 3	**monitor capture** {*capture-name*} **stop** 例： `Device# monitor capture name stop`	キャプチャ ポイントを非アクティブにします。
ステップ 4	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show running-config** 例： `Device# show running-config`	入力を確認します。
ステップ 6	**copy running-config startup-config** 例： `Device# copy running-config startup-config`	（任意）コンフィギュレーション ファイルに設定を保存します。

次のタスク

キャプチャポイントをアクティブおよび非アクティブにする際に、いくつかのエラーが発生する可能性があります。次に、発生する可能性のあるエラーのいくつかの例を示します。

アクティブ化する際に接続ポイントが不明

```
Switch#monitor capture mycap match any
Switch#monitor capture mycap start
No Target is attached to capture failed to disable provision featurefailed to remove
policyfailed to disable provision featurefailed to remove policyfailed to disable provision
 featurefailed to remove policy
Capture statistics collected at software (Buffer):
 Capture duration - 0 seconds
 Packets received - 0
 Packets dropped - 0
 Packets oversized - 0

Unable to activate Capture.
```

```
Switch# unable to get action unable to get action unable to get action
Switch#monitor capture mycap interface g1/0/1 both
Switch#monitor capture mycap start
Switch#
*Nov 5 12:33:43.906: %BUFCAP-6-ENABLE: Capture Point mycap enabled.
```

アクティブ化する際にフィルタが不明

```
Switch#monitor capture mycap int g1/0/1 both
Switch#monitor capture mycap start
Filter not attached to capture
Capture statistics collected at software (Buffer):
 Capture duration - 0 seconds
 Packets received - 0
 Packets dropped - 0
 Packets oversized - 0

Unable to activate Capture.
Switch#monitor capture mycap match any
Switch#monitor capture mycap start
Switch#
*Nov 5 12:35:37.200: %BUFCAP-6-ENABLE: Capture Point mycap enabled.
```

キャプチャ ポイントがすでにアクティブ化されているのに、別のキャプチャ ポイントをアクティブ化しようとする

```
Switch#monitor capture mycap start
PD start invoked while previous run is active Failed to start capture : Wireshark operation
 failure
Unable to activate Capture.
Switch#show monitor capture

Status Information for Capture test
 Target Type:
 Interface: GigabitEthernet1/0/13, Direction: both
 Interface: GigabitEthernet1/0/14, Direction: both
 Status : Active
 Filter Details:
 Capture all packets
 Buffer Details:
 Buffer Type: LINEAR (default)
 Buffer Size (in MB): 10
 File Details:
 Associated file name: flash:cchh.pcap
 Limit Details:
 Number of Packets to capture: 0 (no limit)
 Packet Capture duration: 0 (no limit)
 Packet Size to capture: 0 (no limit)
 Maximum number of packets to capture per second: 1000
 Packet sampling rate: 0 (no sampling)

Status Information for Capture mycap
 Target Type:
 Interface: GigabitEthernet1/0/1, Direction: both
 Status : Inactive
 Filter Details:
 Capture all packets
 Buffer Details:
 Buffer Type: LINEAR (default)
 Buffer Size (in MB): 10
 File Details:
 File not associated
 Limit Details:
 Number of Packets to capture: 0 (no limit)
```

```
 Packet Capture duration: 0 (no limit)
 Packet Size to capture: 0 (no limit)
 Maximum number of packets to capture per second: 1000
 Packet sampling rate: 0 (no sampling)
Switch#monitor capture test stop
Capture statistics collected at software (Buffer & Wireshark):
 Capture duration - 157 seconds
 Packets received - 0
 Packets dropped - 0
 Packets oversized - 0

Switch#
*Nov 5 13:18:17.406: %BUFCAP-6-DISABLE: Capture Point test disabled.
Switch#monitor capture mycap start
Switch#
*Nov 5 13:18:22.664: %BUFCAP-6-ENABLE: Capture Point mycap enabled.
Switch#
```

キャプチャ ポイント バッファのクリア

次の手順に従ってバッファ コンテンツをクリアするか、外部ファイルにストレージとして保存します。

（注） パケットをバッファ内に保存する複数のキャプチャがある場合、メモリロスを避けるため、新しいキャプチャを開始する前にバッファをクリアしてください。アクティブなキャプチャ ポイントのバッファをクリアしないでください。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** **例：** `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**monitor capture** {*capture-name*} [**clear** | **export** *filename*] **例：** `Device# monitor capture mycap clear`	clear：完全にバッファを削除します。 （注） clear コマンドを実行すると、 • Network Essentials ライセンスでは、このコマンドはバッファを削除せずにバッファの内容をクリアします。 • 他のすべてのライセンスでは、このコマンドはバッファ自体を削除します。

	コマンドまたはアクション	目的
		export：バッファでキャプチャされたパケットを保存し、バッファを削除します。
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 running-config** 例： `Device# show running-config`	入力を確認します。
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。

例：キャプチャ ポイント バッファの処理

キャプチャのファイルへのエクスポート

```
Device# monitor capture mycap export flash:mycap.pcap

Storage configured as File for this capture
```

キャプチャ ポイント バッファのクリア

```
Device# monitor capture mycap clear

 Capture configured with file options
```

次のタスク

（注） Network Essentials 以外のライセンスでキャプチャ ポイントのバッファをクリアしようとすると、スイッチは「*Failed to clear capture buffer : Capture Buffer BUSY*」エラーを表示します。

組み込みパケット キャプチャの実装方法

パケット データ キャプチャの管理

（注） アクティブなキャプチャ ポイントのエクスポートは、Network Essentials のみでサポートされています。他のすべてのライセンスでは、まずキャプチャを停止してからエクスポートをする必要があります。

バッファ モードでパケット データ キャプチャを管理するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**monitor capture** *capture-name* **access-list** *access-list-name* 例： `Device# monitor capture mycap access-➥ list v4acl`	アクセス リストをパケット キャプチャのコア フィルタとして指定し、モニタ キャプチャを設定します。
ステップ 3	**monitor capture** *capture-name* **limit duration** *seconds* 例： `Device# monitor capture mycap limit duration 1000`	モニタ キャプチャの制限を設定します。
ステップ 4	**monitor capture** *capture-name* **interface** *interface-name* **both** 例： `Device# monitor capture mycap interface GigabitEthernet 0/0/1 both`	接続ポイントおよびパケット フロー方向を指定して、モニタ キャプチャを設定します。
ステップ 5	**monitor capture** *capture-name* **buffer circular size** *bytes* 例：	パケット データをキャプチャするようにバッファを設定します。

	コマンドまたはアクション	目的
	`Device# monitor capture mycap buffer circular size 10`	
ステップ 6	**monitor capture** *capture-name* **start** 例： `Device# monitor capture mycap start`	トラフィック トレース ポイントでパケットデータのバッファへのキャプチャを開始します。
ステップ 7	**monitor capture** *capture-name* **stop** 例： `Device# monitor capture mycap stop`	トラフィック トレース ポイントでパケット データのキャプチャを停止します。
ステップ 8	**monitor capture** *capture-name* **export** *file-location/file-name* 例： `Device# monitor capture mycap export tftp://10.1.88.9/mycap.pcap`	分析のためにキャプチャされたデータをエクスポートします。
ステップ 9	**end** 例： `Device# end`	特権 EXEC モードに戻ります。

キャプチャされたデータのモニタリングとメンテナンス

キャプチャされたパケットデータのモニタリングとメンテナンスを行うには、次の作業を実行します。キャプチャ バッファの詳細とキャプチャ ポイントの詳細を表示します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。プロンプトが表示されたら、パスワードを入力します。
ステップ 2	**show monitor capture** *capture-buffer-name* **buffer dump** 例： `Device# show monitor capture mycap buffer dump`	（任意）キャプチャ パケットの 16 進数ダンプおよびそのメタデータを表示します。

	コマンドまたはアクション	目的
ステップ 3	**show monitor capture** *capture-buffer-name* **parameter** 例： `Device# show monitor capture mycap parameter`	（任意）キャプチャを指定するために使用されたコマンドのリストを表示します。
ステップ 4	**debug epc capture-point** 例： `Device# debug epc capture-point`	（任意）パケット キャプチャ ポイントのデバッグをイネーブルにします。
ステップ 5	**debug epc provision** 例： `Device# debug epc provision`	（任意）パケット キャプチャ プロビジョニングのデバッグをイネーブルにします。
ステップ 6	**end** 例： `Device(config)# end`	特権 EXEC モードに戻ります。

パケット キャプチャのモニタリング

Wireshark の設定例

例：.pcap ファイルからの概要出力の表示

次のように入力して、.pcap ファイルからの出力を表示できます。

```
Device# show monitor capture file flash:mycap.pcap brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

  1 0.000000000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=0/0, ttl=254
  2 0.000051000   10.10.10.1 -> 10.10.10.2   ICMP 114 Echo (ping) reply    id=0x002e,
seq=0/0, ttl=255 (request in 1)
  3 0.000908000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=1/256, ttl=254
  4 0.001782000   10.10.10.1 -> 10.10.10.2   ICMP 114 Echo (ping) reply    id=0x002e,
seq=1/256, ttl=255 (request in 3)
  5 0.002961000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=2/512, ttl=254
  6 0.003676000   10.10.10.1 -> 10.10.10.2   ICMP 114 Echo (ping) reply    id=0x002e,
seq=2/512, ttl=255 (request in 5)
```

```
  7 0.004835000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=3/768, ttl=254
  8 0.005579000   10.10.10.1 -> 10.10.10.2   ICMP 114 Echo (ping) reply    id=0x002e,
seq=3/768, ttl=255 (request in 7)
  9 0.006850000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=4/1024, ttl=254
 10 0.007586000   10.10.10.1 -> 10.10.10.2   ICMP 114 Echo (ping) reply    id=0x002e,
seq=4/1024, ttl=255 (request in 9)
 11 0.008768000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=5/1280, ttl=254
 12 0.009497000   10.10.10.1 -> 10.10.10.2   ICMP 114 Echo (ping) reply    id=0x002e,
seq=5/1280, ttl=255 (request in 11)
 13 0.010695000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=6/1536, ttl=254
 14 0.011427000   10.10.10.1 -> 10.10.10.2   ICMP 114 Echo (ping) reply    id=0x002e,
seq=6/1536, ttl=255 (request in 13)
 15 0.012728000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=7/1792, ttl=254
 16 0.013458000   10.10.10.1 -> 10.10.10.2   ICMP 114 Echo (ping) reply    id=0x002e,
seq=7/1792, ttl=255 (request in 15)
 17 0.014652000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=8/2048, ttl=254
 18 0.015394000   10.10.10.1 -> 10.10.10.2   ICMP 114 Echo (ping) reply    id=0x002e,
seq=8/2048, ttl=255 (request in 17)
 19 0.016682000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=9/2304, ttl=254
 20 0.017439000   10.10.10.1 -> 10.10.10.2   ICMP 114 Echo (ping) reply    id=0x002e,
seq=9/2304, ttl=255 (request in 19)
 21 0.018655000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=10/2560, ttl=254
 22 0.019385000   10.10.10.1 -> 10.10.10.2   ICMP 114 Echo (ping) reply    id=0x002e,
seq=10/2560, ttl=255 (request in 21)
 23 0.020575000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=11/2816, ttl=254
 --More‹
```

例：.pcap ファイルからの詳細出力の表示

次のように入力して、.pcap ファイルの出力詳細を表示できます。

```
Device# show monitor capture file flash:mycap.pcap detailed
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

Frame 1: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0
    Interface id: 0
    Encapsulation type: Ethernet (1)
    Arrival Time: Nov  6, 2015 11:44:48.322497000 UTC
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1446810288.322497000 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 114 bytes (912 bits)
    Capture Length: 114 bytes (912 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:icmp:data]
Ethernet II, Src: Cisco_f3:63:46 (00:e1:6d:f3:63:46), Dst: Cisco_31:f1:c6
(00:e1:6d:31:f1:c6)
    Destination: Cisco_31:f1:c6 (00:e1:6d:31:f1:c6)
```

```
        Address: Cisco_31:f1:c6 (00:e1:6d:31:f1:c6)
       .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: Cisco_f3:63:46 (00:e1:6d:f3:63:46)
        Address: Cisco_f3:63:46 (00:e1:6d:f3:63:46)
       .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IP (0x0800)
Internet Protocol Version 4, Src: 10.10.10.2 (10.10.10.2), Dst: 10.10.10.1 (10.10.10.1)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not
ECN-Capable Transport))
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
       .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport)
 (0x00)
    Total Length: 100
    Identification: 0x04ba (1210)
    Flags: 0x00
        0... .... = Reserved bit: Not set
        .0.. .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 254
    Protocol: ICMP (1)
    Header checksum: 0x8fc8 [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 10.10.10.2 (10.10.10.2)
    Destination: 10.10.10.1 (10.10.10.1)
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xe4db [correct]
    Identifier (BE): 46 (0x002e)
    Identifier (LE): 11776 (0x2e00)
    Sequence number (BE): 0 (0x0000)
    Sequence number (LE): 0 (0x0000)
    Data (72 bytes)

0000  00 00 00 00 09 c9 8f 77 ab cd ab cd ab cd ab cd   .......w........
0010  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0020  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0030  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0040  ab cd ab cd ab cd ab cd                           ........
        Data: 0000000009c98f77abcdabcdabcdabcdabcdabcdabcdabcd...
        [Length: 72]

Frame 2: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0
    Interface id: 0
```

例：.pcap ファイルからパケット ダンプ出力の表示

次のように入力して、パケット ダンプの出力を表示できます。

```
Device# show monitor capture file flash:mycap.pcap dump
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

0000  00 e1 6d 31 f1 c6 00 e1 6d f3 63 46 08 00 45 00   ..m1....m.cF..E.
0010  00 64 04 ba 00 00 fe 01 8f c8 0a 0a 0a 02 0a 0a   .d..............
```

```
0020  0a 01 08 00 e4 db 00 2e 00 00 00 00 00 00 09 c9   ................
0030  8f 77 ab cd ab cd ab cd ab cd ab cd ab cd ab cd   .w..............
0040  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0050  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0060  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0070  ab cd                                            ..

0000  00 e1 6d 31 f1 80 00 e1 6d 31 f1 80 08 00 45 00   ..m1....m1....E.
0010  00 64 04 ba 00 00 ff 01 8e c8 0a 0a 0a 01 0a 0a   .d..............
0020  0a 02 00 00 ec db 00 2e 00 00 00 00 00 00 09 c9   ................
0030  8f 77 ab cd ab cd ab cd ab cd ab cd ab cd ab cd   .w..............
0040  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0050  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0060  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0070  ab cd                                            ..

0000  00 e1 6d 31 f1 c6 00 e1 6d f3 63 46 08 00 45 00   ..m1....m.cF..E.
0010  00 64 04 bb 00 00 fe 01 8f c7 0a 0a 0a 02 0a 0a   .d..............
0020  0a 01 08 00 e4 d7 00 2e 00 01 00 00 00 00 09 c9   ................
0030  8f 7a ab cd ab cd ab cd ab cd ab cd ab cd ab cd   .z..............
0040  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
```

例：表示フィルタを使用した .pcap ファイルからのパケットの表示

次のように入力して、出力された .pcap ファイルのパケットを表示できます。

```
Device# show monitor capture file flash:mycap.pcap display-filter "ip.src == 10.10.10.2"
 brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

  1 0.000000000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=0/0, ttl=254
  3 0.000908000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=1/256, ttl=254
  5 0.002961000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=2/512, ttl=254
  7 0.004835000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=3/768, ttl=254
  9 0.006850000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=4/1024, ttl=254
 11 0.008768000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=5/1280, ttl=254
 13 0.010695000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=6/1536, ttl=254
 15 0.012728000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=7/1792, ttl=254
 17 0.014652000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=8/2048, ttl=254
 19 0.016682000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=9/2304, ttl=254
 21 0.018655000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=10/2560, ttl=254
 23 0.020575000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x002e,
seq=11/2816, ttl=254
```

例：.pcap ファイルにキャプチャされたパケットの数を表示

次のように入力して、.pcap ファイルにキャプチャされたパケットの数を表示できます。

```
Device# show monitor capture file flash:mycap.pcap packet-count
File name:           /flash/mycap.pcap
Number of packets:   50
```

例：.pcap ファイルから単一パケット ダンプの表示

次のように入力して、.pcap ファイルから単一のパケット ダンプを表示できます。

```
Device#  show monitor capture file flash:mycap.pcap packet-number 10 dump
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

0000  00 e1 6d 31 f1 80 00 e1 6d 31 f1 80 08 00 45 00   ..m1....m1....E.
0010  00 64 04 be 00 00 ff 01 8e c4 0a 0a 0a 01 0a 0a   .d..............
0020  0a 02 00 00 ec ce 00 2e 00 04 00 00 00 00 09 c9   ................
0030  8f 80 ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0040  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0050  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0060  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0070  ab cd
```

例：.pcap ファイルにキャプチャされたパケットの統計情報を表示

次のように入力して、.pcapファイルにキャプチャされたパケットの統計情報を表示できます。

```
Device# show monitor capture file flash:mycap.pcap statistics "h225,counter"
================== H225 Message and Reason Counter ==================
RAS-Messages:
Call Signalling:
=====================================================================
```

例：単純なキャプチャおよび表示

次の例は、レイヤ 3 インターフェイス ギガビット イーサネット 1/0/1 でトラフィックをモニタする方法を示しています。

ステップ 1：次のように入力して関連トラフィックで一致するキャプチャ ポイントを定義します。

```
Device# monitor capture mycap interface GigabitEthernet1/0/3 in
Device# monitor capture mycap match ipv4 any any
Device# monitor capture mycap limit duration 60 packets 50
Device# monitor capture mycap buffer size 100
```

CPU 使用率の上昇を避けるため、制限として最も低いパケット数および時間が設定されています。

ステップ 2：次のように入力してキャプチャ ポイントが正確に定義されていることを確認します。

```
Device# show monitor capture mycap parameter
      monitor capture mycap interface GigabitEthernet1/0/3 in
   monitor capture mycap match ipv4  any any
   monitor capture mycap buffer size 100
   monitor capture mycap limit packets 50 duration 60

Device# show monitor capture mycap
Status Information for Capture mycap
  Target Type:
```

```
   Interface: GigabitEthernet1/0/3, Direction: in
   Status : Inactive
  Filter Details:
   IPv4
    Source IP:  any
    Destination IP:  any
   Protocol: any
  Buffer Details:
   Buffer Type: LINEAR (default)
   Buffer Size (in MB): 100
  File Details:
   File not associated
  Limit Details:
   Number of Packets to capture: 50
   Packet Capture duration: 60
   Packet Size to capture: 0 (no limit)
   Packet sampling rate: 0 (no sampling)
```

ステップ 3：キャプチャ プロセスを開始し、結果を表示します。

```
Device# monitor capture mycap start display
 Starting the packet display ........ Press Ctrl + Shift + 6 to exit

  1   0.000000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0030,
seq=0/0, ttl=254
  2   0.003682   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0030,
seq=1/256, ttl=254
  3   0.006586   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0030,
seq=2/512, ttl=254
  4   0.008941   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0030,
seq=3/768, ttl=254
  5   0.011138   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0030,
seq=4/1024, ttl=254
  6   0.014099   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0030,
seq=5/1280, ttl=254
  7   0.016868   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0030,
seq=6/1536, ttl=254
  8   0.019210   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0030,
seq=7/1792, ttl=254
  9   0.024785   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0030,
seq=8/2048, ttl=254
 --More--
```

ステップ 4：次のように入力して、キャプチャ ポイントを削除します。

```
Device# no monitor capture mycap
```

（注） 制限が設定してあり、その制限に達するとキャプチャは自動的に停止するため、この特定のケースでは、**stop** コマンドは必要ありません。

pcap の統計情報に使用する構文の詳細については、「その他の参考資料」セクションを参照してください。

例：単純なキャプチャおよび保存

次の例は、フィルタにパケットをキャプチャする方法を示しています。

ステップ 1：次のように入力して、関連トラフィックで一致するキャプチャ ポイントを定義し、それをファイルに関連付けます。

```
Device# monitor capture mycap interface GigabitEthernet1/0/3 in
Device# monitor capture mycap match ipv4 any any
Device# monitor capture mycap limit duration 60 packets 50
Device# monitor capture mycap file location flash:mycap.pcap
```

ステップ 2：次のように入力してキャプチャ ポイントが正確に定義されていることを確認します。

```
Device# show monitor capture mycap parameter
   monitor capture mycap interface GigabitEthernet1/0/3 in
   monitor capture mycap match ipv4  any any
   monitor capture mycap file location flash:mycap.pcap
   monitor capture mycap limit packets 50 duration 60

Device# show monitor capture mycap

Status Information for Capture mycap
  Target Type:
   Interface: GigabitEthernet1/0/3, Direction: in
   Status : Inactive
  Filter Details:
   IPv4
    Source IP:  any
    Destination IP:  any
   Protocol: any
  Buffer Details:
   Buffer Type: LINEAR (default)
  File Details:
   Associated file name: flash:mycap.pcap
  Limit Details:
   Number of Packets to capture: 50
   Packet Capture duration: 60
   Packet Size to capture: 0 (no limit)
   Packet sampling rate: 0 (no sampling)
```

ステップ 3：次のように入力してパケットを開始します。

```
Device# monitor capture mycap start
```

ステップ 4: 次のように入力して実行中のエクステンドキャプチャ統計情報を表示します。

```
Device#  show monitor capture mycap capture-statistics
Capture statistics collected at software:
        Capture duration - 15 seconds
        Packets received - 40
        Packets dropped - 0
        Packets oversized - 0
        Packets errored - 0
        Packets sent - 40
        Bytes received - 7280
        Bytes dropped - 0
        Bytes oversized - 0
        Bytes errored - 0
```

```
        Bytes sent ¬ 4560
```

ステップ 5：十分な時間の経過後に、次のように入力してキャプチャを停止します。

```
Device# monitor capture mycap stop
Capture statistics collected at software (Buffer & Wireshark):
        Capture duration - 20 seconds
        Packets received - 50
        Packets dropped - 0
        Packets oversized - 0
```

（注） あるいは、時間の経過またはパケット カウントが一致した後に、キャプチャ操作を自動的に停止させることもできます。

mycap.pcap ファイルには、キャプチャしたパケットが含まれます。

ステップ 6: 次のように入力して停止後のエクステンドキャプチャの統計情報を表示します。

```
Device#  show monitor capture mycap capture-statistics
Capture statistics collected at software:
        Capture duration - 20 seconds
        Packets received - 50
        Packets dropped - 0
        Packets oversized - 0
        Packets errored - 0
        Packets sent - 50
        Bytes received - 8190
        Bytes dropped - 0
        Bytes oversized - 0
        Bytes errored - 0
        Bytes sent ¬ 5130
```

ステップ 7：次のように入力してパケットを表示します。

```
Device# show monitor capture file flash:mycap.pcap
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

  1 0.000000000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=0/0, ttl=254
  2 0.002555000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=1/256, ttl=254
  3 0.006199000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=2/512, ttl=254
  4 0.009199000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=3/768, ttl=254
  5 0.011647000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=4/1024, ttl=254
  6 0.014168000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=5/1280, ttl=254
  7 0.016737000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=6/1536, ttl=254
  8 0.019403000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=7/1792, ttl=254
  9 0.022151000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=8/2048, ttl=254
```

```
 10 0.024722000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=9/2304, ttl=254
 11 0.026890000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=10/2560, ttl=254
 12 0.028862000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=11/2816, ttl=254
 --More--
```

pcap の統計情報に使用する構文の詳細については、「その他の参考資料」セクションを参照してください。

ステップ 8：次のように入力して、キャプチャ ポイントを削除します。

```
Device# no monitor capture mycap
```

例：バッファのキャプチャの使用

次に、バッファのキャプチャを使用する例を示します。

ステップ 1：次のように入力してバッファ キャプチャ オプションでキャプチャ セッションを起動します。

```
Device# monitor capture mycap interface GigabitEthernet1/0/3 in
Device# monitor capture mycap match ipv4 any any
Device# monitor capture mycap buffer circular size 1
Device# monitor capture mycap start
```

ステップ 2：次のように入力してキャプチャがアクティブであるかどうかを決定します。

```
Device# show monitor capture mycap
Status Information for Capture mycap
  Target Type:
   Interface: GigabitEthernet1/0/3, Direction: in
   Status : Active
  Filter Details:
   IPv4
    Source IP:  any
    Destination IP:  any
   Protocol: any
  Buffer Details:
   Buffer Type: CIRCULAR
   Buffer Size (in MB): 1
  File Details:
   File not associated
  Limit Details:
   Number of Packets to capture: 0 (no limit)
   Packet Capture duration: 0 (no limit)
   Packet Size to capture: 0 (no limit)
   Maximum number of packets to capture per second: 1000
   Packet sampling rate: 0 (no sampling)
```

ステップ 3：次のように入力してランタイム時に拡張キャプチャの統計情報を表示します。

```
Device# show monitor capture mycap capture-statistics
Capture statistics collected at software:
        Capture duration - 88 seconds
        Packets received - 1000
```

```
        Packets dropped - 0
        Packets oversized - 0
        Packets errored - 0
        Packets sent - 1000
        Bytes received - 182000
        Bytes dropped - 0
        Bytes oversized - 0
        Bytes errored - 0
        Bytes sent - 114000
```

ステップ 4：次のように入力してキャプチャを停止します。

```
Device# monitor capture mycap stop
Capture statistics collected at software (Buffer):
        Capture duration - 2185 seconds
        Packets received - 51500
        Packets dropped - 0
        Packets oversized - 0
```

ステップ 5：次のように入力して停止後の拡張キャプチャの統計情報を表示します。

```
Device# show monitor capture mycap capture-statistics
Capture statistics collected at software:
        Capture duration - 156 seconds
        Packets received - 2000
        Packets dropped - 0
        Packets oversized - 0
        Packets errored - 0
        Packets sent - 2000
        Bytes received - 364000
        Bytes dropped - 0
        Bytes oversized - 0
        Bytes errored - 0
        Bytes sent - 228000
```

ステップ 6：次のように入力してキャプチャがアクティブであるかどうかを決定します。

```
Device# show monitor capture mycap
Status Information for Capture mycap
  Target Type:
   Interface: GigabitEthernet1/0/3, Direction: in
   Status : Inactive
  Filter Details:
   IPv4
    Source IP:  any
    Destination IP:  any
   Protocol: any
  Buffer Details:
   Buffer Type: CIRCULAR
   Buffer Size (in MB): 1
  File Details:
   File not associated
  Limit Details:
   Number of Packets to capture: 0 (no limit)
   Packet Capture duration: 0 (no limit)
   Packet Size to capture: 0 (no limit)
   Maximum number of packets to capture per second: 1000
   Packet sampling rate: 0 (no sampling)
```

ステップ 7：次のように入力してバッファのパケットを表示します。

```
Device# show monitor capture mycap buffer brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

  1   0.000000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0038,
seq=40057/31132, ttl=254
  2   0.000030   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0038,
seq=40058/31388, ttl=254
  3   0.000052   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0038,
seq=40059/31644, ttl=254
  4   0.000073   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0038,
seq=40060/31900, ttl=254
  5   0.000094   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0038,
seq=40061/32156, ttl=254
  6   0.000115   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0038,
seq=40062/32412, ttl=254
  7   0.000137   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0038,
seq=40063/32668, ttl=254
  8   0.000158   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0038,
seq=40064/32924, ttl=254
  9   0.000179   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0038,
seq=40065/33180, ttl=254
 10   0.000200   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0038,
seq=40066/33436, ttl=254
 11   0.000221   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0038,
seq=40067/33692, ttl=254
 12   0.000243   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0038,
seq=40068/33948, ttl=254
 --More--
```

パケットがバッファに入ったことに注意してください。

ステップ 8：他の表示モードでパケットを表示します。

```
Device# show monitor capture mycap buffer detailed
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

Frame 1: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0
    Interface id: 0
    Encapsulation type: Ethernet (1)
    Arrival Time: Nov  6, 2015 18:10:06.297972000 UTC
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1446833406.297972000 seconds
    [Time delta from previous captured frame: 0.000000000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 0.000000000 seconds]
    Frame Number: 1
    Frame Length: 114 bytes (912 bits)
    Capture Length: 114 bytes (912 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:icmp:data]
Ethernet II, Src: Cisco_f3:63:46 (00:e1:6d:f3:63:46), Dst: Cisco_31:f1:c6
(00:e1:6d:31:f1:c6)
    Destination: Cisco_31:f1:c6 (00:e1:6d:31:f1:c6)
        Address: Cisco_31:f1:c6 (00:e1:6d:31:f1:c6)
       .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Source: Cisco_f3:63:46 (00:e1:6d:f3:63:46)
        Address: Cisco_f3:63:46 (00:e1:6d:f3:63:46)
       .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
```

```
    Type: IP (0x0800)
Internet Protocol Version 4, Src: 10.10.10.2 (10.10.10.2), Dst: 10.10.10.1 (10.10.10.1)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not
ECN-Capable Transport))
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
       .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport)
 (0x00)
    Total Length: 100
    Identification: 0xabdd (43997)
    Flags: 0x00
        0... .... = Reserved bit: Not set
        .0.. .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 254
    Protocol: ICMP (1)
    Header checksum: 0xe8a4 [validation disabled]
        [Good: False]
        [Bad: False]
    Source: 10.10.10.2 (10.10.10.2)
    Destination: 10.10.10.1 (10.10.10.1)
Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xa620 [correct]
    Identifier (BE): 56 (0x0038)
    Identifier (LE): 14336 (0x3800)
    Sequence number (BE): 40057 (0x9c79)
    Sequence number (LE): 31132 (0x799c)
    Data (72 bytes)

0000  00 00 00 00 0b 15 30 63 ab cd ab cd ab cd ab cd   ......0c........
0010  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0020  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0030  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0040  ab cd ab cd ab cd ab cd                           ........
        Data: 000000000b153063abcdabcdabcdabcdabcdabcdabcdabcd...
        [Length: 72]

Frame 2: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0


Device# show monitor capture mycap buffer dump
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

0000  00 e1 6d 31 f1 c6 00 e1 6d f3 63 46 08 00 45 00   ..m1....m.cF..E.
0010  00 64 ab dd 00 00 fe 01 e8 a4 0a 0a 0a 02 0a 0a   .d..............
0020  0a 01 08 00 a6 20 00 38 9c 79 00 00 00 00 0b 15   ..... .8.y......
0030  30 63 ab cd ab cd ab cd ab cd ab cd ab cd ab cd   0c..............
0040  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0050  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0060  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0070  ab cd                                            ..

0000  00 e1 6d 31 f1 c6 00 e1 6d f3 63 46 08 00 45 00   ..m1....m.cF..E.
0010  00 64 ab de 00 00 fe 01 e8 a3 0a 0a 0a 02 0a 0a   .d..............
0020  0a 01 08 00 a6 1d 00 38 9c 7a 00 00 00 00 0b 15   .......8.z......
0030  30 65 ab cd ab cd ab cd ab cd ab cd ab cd ab cd   0e..............
0040  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0050  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
0060  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd   ................
```

```
0070  ab cd
```

ステップ 9：次のように入力してバッファをクリアします。

```
Device# monitor capture mycap clear
```

（注） 注：バッファをクリアすると、その内容とともにバッファが削除されます。

（注） バッファの内容を表示する必要がある場合は、show コマンドの後に clear コマンドを実行します。

ステップ 10：トラフィックを再開し、10 秒待ってから次のように入力してバッファコンテンツを表示します。

（注） キャプチャがアクティブなときに、バッファから show の実行をすることはできません。バッファから showを実行する前に、キャプチャを停止する必要があります。しかし、ファイルおよびバッファ モードの両方においてキャプチャがアクティブなときにpcap ファイルで showの実行ができます。ファイル モードでは、キャプチャがアクティブなときに、現在のキャプチャ セッションの pcap ファイルでパケットを表示することもできます。

```
Device# monitor capture mycap start
Switch#  show monitor capture mycap

Status Information for Capture mycap
  Target Type:
   Interface: GigabitEthernet1/0/3, Direction: in
   Status : Active
  Filter Details:
   IPv4
    Source IP:  any
    Destination IP:  any
   Protocol: any
  Buffer Details:
   Buffer Type: CIRCULAR
   Buffer Size (in MB): 1
  File Details:
   File not associated
  Limit Details:
   Number of Packets to capture: 0 (no limit)
   Packet Capture duration: 0 (no limit)
   Packet Size to capture: 0 (no limit)
   Maximum number of packets to capture per second: 1000
   Packet sampling rate: 0 (no sampling)
```

ステップ 11：次のように入力して、パケットキャプチャを停止し、バッファの内容を表示します。

```
Device# monitor capture mycap stop
Capture statistics collected at software (Buffer):
 Capture duration - 111 seconds
 Packets received - 5000
 Packets dropped - 0
 Packets oversized - 0
```

ステップ 12：次のように入力してキャプチャがアクティブであるかどうかを決定します。

```
Device# show monitor capture mycap
Status Information for Capture mycap
 Target Type:
  Interface: GigabitEthernet1/0/3, Direction: in
 Status : Inactive
Filter Details:
 IPv4
  Source IP: any
  Destination IP: any
 Protocol: any
Buffer Details:
 Buffer Type: CIRCULAR
 Buffer Size (in MB): 1
File Details:
 File not associated
Limit Details:
 Number of Packets to capture: 0 (no limit)
 Packet Capture duration: 0 (no limit)
 Packet Size to capture: 0 (no limit)
 Maximum number of packets to capture per second: 1000
 Packet sampling rate: 0 (no sampling)
```

ステップ 13：次のように入力してバッファのパケットを表示します。

```
Device# show monitor capture mycap buffer brief
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

  1 0.000000000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=0/0, ttl=254
  2 0.000030000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=1/256, ttl=254
  3 0.000051000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=2/512, ttl=254
  4 0.000072000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=3/768, ttl=254
  5 0.000093000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=4/1024, ttl=254
  6 0.000114000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=5/1280, ttl=254
  7 0.000136000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=6/1536, ttl=254
  8 0.000157000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=7/1792, ttl=254
  9 0.000178000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=8/2048, ttl=254
 10 0.000199000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=9/2304, ttl=254
 11 0.000220000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=10/2560, ttl=254
 12 0.000241000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=11/2816, ttl=254
 --More‹
```

ステップ 14：次のように入力して、内部 flash: storage デバイス内の mycap1.pcap ファイルにバッファ コンテンツを保存します。

```
Device# monitor capture mycap export flash:mycap.pcap
Exported Successfully
```

（注） 現在のエクスポート実装では、コマンドを実行すると、エクスポートは「開始」されますが、ユーザにプロンプトを返す場合には完了しません。そこで、ファイルでパケットの表示を実行する前に、Wireshark からコンソールにメッセージが表示されるのを待機する必要があります。

ステップ 15：次のように入力してファイルからキャプチャ パケットを表示します。

```
Device#  show monitor capture file flash:mycap.pcap
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

  1 0.000000000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=0/0, ttl=254
  2 0.000030000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=1/256, ttl=254
  3 0.000051000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=2/512, ttl=254
  4 0.000072000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=3/768, ttl=254
  5 0.000093000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=4/1024, ttl=254
  6 0.000114000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=5/1280, ttl=254
  7 0.000136000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=6/1536, ttl=254
  8 0.000157000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=7/1792, ttl=254
  9 0.000178000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=8/2048, ttl=254
 10 0.000199000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=9/2304, ttl=254
 11 0.000220000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=10/2560, ttl=254
 12 0.000241000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0039,
seq=11/2816, ttl=254
 --More--
```

ステップ 16：次のように入力して、キャプチャ ポイントを削除します。

```
Device# no monitor capture mycap
```

例：出力方向のパケットの簡単なキャプチャおよび保存

次の例は、フィルタにパケットをキャプチャする方法を示しています。

ステップ 1：次のように入力して、関連トラフィックで一致するキャプチャ ポイントを定義し、それをファイルに関連付けます。

```
Device# monitor capture mycap interface Gigabit 1/0/1 out match ipv4 any any
Device# monitor capture mycap limit duration 60 packets 100
Device# monitor capture mycap file location flash:mycap.pcap buffer-size 90
```

ステップ 2：次のように入力してキャプチャ ポイントが正確に定義されていることを確認します。

```
Device# show monitor capture mycap parameter
   monitor capture mycap interface GigabitEthernet1/0/1 out
   monitor capture mycap match ipv4  any any
   monitor capture mycap file location flash:mycap.pcap buffer-size 90
   monitor capture mycap limit packets 100 duration 60

Device# show monitor capture mycap

Status Information for Capture mycap
  Target Type:
   Interface: GigabitEthernet1/0/1, Direction: out
   Status : Inactive
  Filter Details:
   IPv4
    Source IP:  any
    Destination IP:  any
   Protocol: any
  Buffer Details:
   Buffer Type: LINEAR (default)
  File Details:
   Associated file name: flash:mycap.pcap
   Size of buffer(in MB): 90
  Limit Details:
   Number of Packets to capture: 100
   Packet Capture duration: 60
   Packet Size to capture: 0 (no limit)
   Packets per second: 0 (no limit)
   Packet sampling rate: 0 (no sampling)
```

ステップ 3：次のように入力してパケットを開始します。

```
Device# monitor capture mycap start
A file by the same capture file name already exists, overwrite?[confirm]
Turning on lock-step mode

Device#
*Oct 14 09:35:32.661: %BUFCAP-6-ENABLE: Capture Point mycap enabled.
```

（注） 時間の経過またはパケット カウントが一致した後に、キャプチャ操作を自動的に停止させてください。出力に次のメッセージが表示された場合は、キャプチャ処理が停止していることを意味します。

```
*Oct 14 09:36:34.632: %BUFCAP-6-DISABLE_ASYNC: Capture Point mycap disabled. Rea
son : Wireshark Session Ended
```

mycap.pcap ファイルには、キャプチャしたパケットが含まれます。

ステップ 4：次のように入力してパケットを表示します。

```
Device# show monitor capture file flash:mycap.pcap
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

  0.000000    10.1.1.30 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

```
  1.000000    10.1.1.31 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000    10.1.1.32 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000    10.1.1.33 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000    10.1.1.34 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000    10.1.1.35 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.000000    10.1.1.36 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.000000    10.1.1.37 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.000000    10.1.1.38 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.000000    10.1.1.39 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

ステップ 5：次のように入力してキャプチャ ポイントを削除します。

```
Device# no monitor capture mycap
```

組み込みパケット キャプチャの設定例

例：パケット データ キャプチャの管理

次の例では、パケット データ キャプチャを管理する方法を示します。

```
Device> enable
Device# monitor capture mycap start
Device# monitor capture mycap access-list v4acl
Device# monitor capture mycap limit duration 1000
Device# monitor capture mycap interface GigabitEthernet 0/0/1 both
Device# monitor capture mycap buffer circular size 10
Device# monitor capture mycap start
Device# monitor capture mycap export tftp://10.1.88.9/mycap.pcap
Device# monitor capture mycap stop
Device# end
```

例：キャプチャされたデータのモニタリングとメンテナンス

次の例は、ASCII 形式でパケットをダンプする方法を示しています。

```
Device# show monitor capture mycap buffer dump
Starting the packet display ........ Press Ctrl + Shift + 6 to exit

0
0000: 01005E00 00020000 0C07AC1D 080045C0 ..^...........E.
0010: 00300000 00000111 CFDC091D 0002E000 .0..............
0020: 000207C1 07C1001C 802A0000 10030AFA .........*......
0030: 1D006369 73636F00 0000091D 0001 ..example.......
1
0000: 01005E00 0002001B 2BF69280 080046C0 ..^.....+.....F.
0010: 00200000 00000102 44170000 0000E000 . ......D.......
0020: 00019404 00001700 E8FF0000 0000 ..............
2
0000: 01005E00 0002001B 2BF68680 080045C0 ..^.....+.....E.
0010: 00300000 00000111 CFDB091D 0003E000 .0..............
0020: 000207C1 07C1001C 88B50000 08030A6E ...............n
0030: 1D006369 73636F00 0000091D 0001 ..example.......
3
0000: 01005E00 000A001C 0F2EDC00 080045C0 ..^...........E.
0010: 003C0000 00000258 CE7F091D 0004E000 .<.....X........
0020: 000A0205 F3000000 00000000 00000000 ................
0030: 00000000 00D10001 000C0100 01000000 ................
0040: 000F0004 00080501 0300
```

次の例は、mycapという名前のキャプチャの設定に使用するコマンドのリストを表示する方法を示しています。

```
Device# show monitor capture mycap parameter
monitor capture mycap interface GigabitEthernet 1/0/1 both
monitor capture mycap match any
monitor capture mycap buffer size 10
monitor capture mycap limit pps 1000
```

次の例は、キャプチャ ポイントをデバッグする方法を示しています。

```
Device# debug epc capture-point
EPC capture point operations debugging is on

Device# monitor capture mycap start
*Jun 4 14:17:15.463: EPC CP: Starting the capture cap1
*Jun 4 14:17:15.463: EPC CP: (brief=3, detailed=4, dump=5) = 0
*Jun 4 14:17:15.463: EPC CP: final check before activation
*Jun 4 14:17:15.463: EPC CP: setting up c3pl infra
*Jun 4 14:17:15.463: EPC CP: Setup c3pl acl-class-policy
*Jun 4 14:17:15.463: EPC CP: Creating a class
*Jun 4 14:17:15.464: EPC CP: Creating a class : Successful
*Jun 4 14:17:15.464: EPC CP: class-map Created
*Jun 4 14:17:15.464: EPC CP: creating policy-name epc_policy_cap1
*Jun 4 14:17:15.464: EPC CP: Creating Policy epc_policy_cap1 of type 49 and client type
 21
*Jun 4 14:17:15.464: EPC CP: Storing a Policy
*Jun 4 14:17:15.464: EPC CP: calling ppm_store_policy with epc_policy
*Jun 4 14:17:15.464: EPC CP: Creating Policy : Successful
*Jun 4 14:17:15.464: EPC CP: policy-map created
*Jun 4 14:17:15.464: EPC CP: creating filter for ANY
*Jun 4 14:17:15.464: EPC CP: Adding acl to class : Successful
*Jun 4 14:17:15.464: EPC CP: Setup c3pl class to policy
*Jun 4 14:17:15.464: EPC CP: Attaching Class to Policy
*Jun 4 14:17:15.464: EPC CP: Attaching epc_class_cap1 to epc_policy_cap1
*Jun 4 14:17:15.464: EPC CP: Attaching Class to Policy : Successful
*Jun 4 14:17:15.464: EPC CP: setting up c3pl qos
*Jun 4 14:17:15.464: EPC CP: DBG> Set packet rate limit to 1000
*Jun 4 14:17:15.464: EPC CP: creating action for policy_map epc_policy_cap1 class_map
epc_class_cap1
*Jun 4 14:17:15.464: EPC CP: DBG> Set packet rate limit to 1000
*Jun 4 14:17:15.464: EPC CP: Activating Interface GigabitEthernet1/0/1 direction both
*Jun 4 14:17:15.464: EPC CP: Id attached 0
*Jun 4 14:17:15.464: EPC CP: inserting into active lists
*Jun 4 14:17:15.464: EPC CP: Id attached 0
*Jun 4 14:17:15.465: EPC CP: inserting into active lists
*Jun 4 14:17:15.465: EPC CP: Activating Vlan
*Jun 4 14:17:15.465: EPC CP: Deleting all temp interfaces
*Jun 4 14:17:15.465: %BUFCAP-6-ENABLE: Capture Point cap1 enabled.
*Jun 4 14:17:15.465: EPC CP: Active Capture 1

Device# monitor capture mycap1 stop
*Jun 4 14:17:31.963: EPC CP: Stopping the capture cap1
*Jun 4 14:17:31.963: EPC CP: Warning: unable to unbind capture cap1
*Jun 4 14:17:31.963: EPC CP: Deactivating policy-map
*Jun 4 14:17:31.963: EPC CP: Policy epc_policy_cap1
*Jun 4 14:17:31.964: EPC CP: Deactivating policy-map Successful
*Jun 4 14:17:31.964: EPC CP: removing povision feature
*Jun 4 14:17:31.964: EPC CP: Found action for policy-map epc_policy_cap1 class-map
epc_class_cap1
*Jun 4 14:17:31.964: EPC CP: cleanning up c3pl infra
*Jun 4 14:17:31.964: EPC CP: Removing Class epc_class_cap1 from Policy
*Jun 4 14:17:31.964: EPC CP: Removing Class from epc_policy_cap1
*Jun 4 14:17:31.964: EPC CP: Successfully removed
```

```
*Jun 4 14:17:31.964: EPC CP: Removing acl mac from class
*Jun 4 14:17:31.964: EPC CP: Removing acl from class : Successful
*Jun 4 14:17:31.964: EPC CP: Removing all policies
*Jun 4 14:17:31.964: EPC CP: Removing Policy epc_policy_cap1
*Jun 4 14:17:31.964: EPC CP: Removing Policy : Successful
*Jun 4 14:17:31.964: EPC CP: Removing class epc_class_cap1
*Jun 4 14:17:31.965: EPC CP: Removing class : Successful
*Jun 4 14:17:31.965: %BUFCAP-6-DISABLE: Capture Point cap1 disabled.
*Jun 4 14:17:31.965: EPC CP: Active Capture 0
```

次の例は、組み込みパケット キャプチャ（EPC）のプロビジョニングをデバッグする方法を示しています。

```
Device# debug epc provision
EPC provisionioning debugging is on

Device# monitor capture mycap start
*Jun 4 14:17:54.991: EPC PROV: No action found for policy-map epc_policy_cap1 class-map
 epc_class_cap1
*Jun 4 14:17:54.991: EPC PROV:
*Jun 4 14:17:54.991: Attempting to install service policy epc_policy_cap1
*Jun 4 14:17:54.992: EPC PROV: Attached service policy to epc idb subblock
*Jun 4 14:17:54.992: EPC PROV: Successful. Create feature object
*Jun 4 14:17:54.992: EPC PROV:
*Jun 4 14:17:54.992: Attempting to install service policy epc_policy_cap1
*Jun 4 14:17:54.992: EPC PROV: Successful. Create feature object
*Jun 4 14:17:54.992: %BUFCAP-6-ENABLE: Capture Point cap1 enabled.

Device# monitor capture mycap stop
*Jun 4 14:18:02.503: EPC PROV: Successful. Remove feature object
*Jun 4 14:18:02.504: EPC PROV: Successful. Remove feature object
*Jun 4 14:18:02.504: EPC PROV: Destroyed epc idb subblock
*Jun 4 14:18:02.504: EPC PROV: Found action for policy-map epc_policy_cap1 class-map
epc_class_cap1
*Jun 4 14:18:02.504: EPC PROV: Deleting EPC action
*Jun 4 14:18:02.504: EPC PROV: Successful. CLASS_REMOVE, policy-map epc_policy_cap1,
class epc_class_cap1
*Jun 4 14:18:02.504: %BUFCAP-6-DISABLE: Capture Point cap1 disabled.
```

その他の参考資料

関連資料

関連項目	参照先
表示フィルタ	表示フィルタの構文については、以下を参照して下さい。 『Display Filter Reference』
pcap ファイル統計情報	pcap ファイル統計情報の表示に使用する構文については、以下で「-z」オプションの詳細を参照してください。 『Tshark Command Reference』

エラー メッセージ デコーダ

説明	リンク
このリリースのシステムエラーメッセージを調査し解決するために、エラー メッセージ デコーダ ツールを使用します。	https://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi

標準および RFC

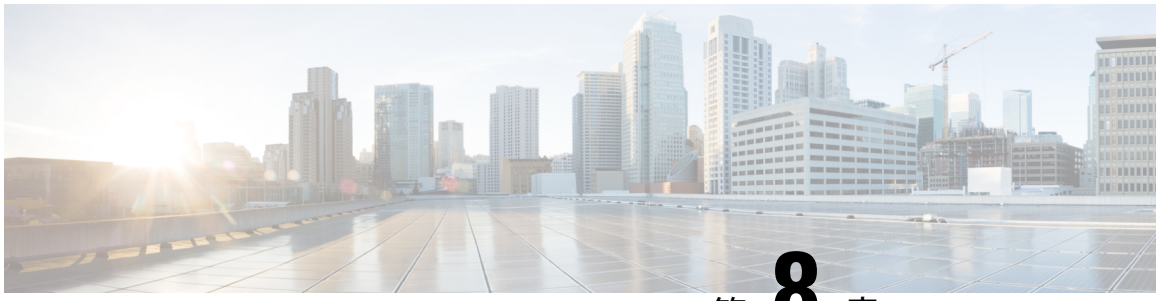
標準/RFC	役職（Title）
なし	-

MIB

MIB	MIB リンク
本リリースでサポートするすべての MIB	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

テクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Product Alert Tool（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support



第 8 章

Flexible NetFlow の設定

Flexible NetFlow の前提条件

次に、Flexible NetFlow コンフィギュレーションの前提条件を示します。

- 送信元インターフェイスを設定する必要があります。送信元インターフェイスを設定しない場合、エクスポータはディセーブル ステートのままになります。
- フロー モニタごとに、有効なレコード名を設定する必要があります。
- IPv6 宛先サーバにフロー レコードをエクスポートするには、IPv6 ルーティングをイネーブルにする必要があります。
- IPFIX 形式の NetFlow レコードをエクスポートするには、フロー エクスポータに IPFIX エクスポート プロトコルを設定する必要があります。
- 『Cisco IOS Flexible NetFlow Command Reference』で、次のコマンドで定義する Flexible NetFlow の key フィールドについてよく理解してください。
  - **match datalink**：データリンク（レイヤ 2）フィールド
  - **match flow**：フィールド識別フロー
  - **match interface**：インターフェイス フィールド
  - **match ipv4**：IPv4 フィールド
  - **match ipv6**：IPv6 フィールド

- **match transport**：トランスポート層フィールド
- **match flow cts**：CTS フィールド

- 『Cisco IOS Flexible NetFlow Command Reference』で、次のコマンドで定義する Flexible NetFlow の nonkey フィールドについてよく理解してください。
  - **collect counter**：カウンタ フィールド
  - **collect flow**：フィールド識別フロー
  - **collect interface**：インターフェイス フィールド
  - **collect timestamp**：タイムスタンプ フィールド
  - **collect transport**：トランスポート層フィールド

IPv4 トラフィック

- ネットワーキング デバイスが IPv4 ルーティング用に設定されていること。
- Cisco Express Forwarding またはdistributed Cisco Express Forwarding のいずれかが、デバイスおよびFlexible NetFlow を有効化するすべてのインターフェイスで有効化されていること。

IPv6 トラフィック

- ネットワーキング デバイスが、IPv6 ルーティング用に設定されていること。
- Cisco Express Forwarding IPv6 または分散型 Cisco Express Forwarding のいずれかが、デバイスおよび Flexible NetFlow を有効化するすべてのインターフェイスで有効化されていること。

Flexible Netflow に関する制約事項

次に、Flexible NetFlow に関する制約事項を示します。

- Flexible NetFlow は、L2 ポートチャネル インターフェイスではサポートされませんが、L2 ポートチャネル メンバー ポートではサポートされます。
- Flexible NetFlow は、L3 ポートチャネル インターフェイスではサポートされませんが、L3 ポートチャネル メンバー ポートではサポートされます。
- Traditional NetFlow（TNF）のアカウンティングはサポートされていません。
- Flexible NetFlow バージョン 9 およびバージョン 10 のエクスポート フォーマットがサポートされています。ただし、エクスポートプロトコルが設定されていない場合は、バージョン 9 のエクスポート フォーマットがデフォルトで適用されます。

- 有線 AVC トラフィックの場合、システム上の 1 つ以上のレイヤ 2 またはレイヤ 3 の物理インターフェイスに設定できるフロー モニタは 1 つのみです。
- レイヤ 2、IPv4、および IPv6 のトラフィック タイプがサポートされています。異なるトラフィック タイプの複数のフロー モニタを、指定したインターフェイスと方向に適用できます。同じトラフィック タイプの複数のフロー モニタを指定したインターフェイスと方向には適用できません。
- レイヤ 2、VLAN、およびレイヤ 3 のインターフェイスをサポートしていますが、デバイスは SVI およびトンネルをサポートしていません。
- 次のサイズの NetFlow テーブルがサポートされています。

トリム レベル	入力 NetFlow テーブル	出力 NetFlow テーブル
Network Essentials	32 K	32 K
Network Advantage	32 K	32 K

- スイッチのタイプに応じて、スイッチには 1 個または 2 個の転送 ASIC があります。上の表に示されている容量は、コア単位または ASIC 単位です。
- スイッチは最大 4 つの ASIC をサポートします。各 ASIC には 2 つのコアがあります。各 TCAM は最大 1024 の入力エントリと 2048 の出力エントリを処理できますが、各コアには 32K の入力と 32K の出力エントリがあります。
- NetFlow テーブルは個別のコンパートメントにあり、組み合わせることはできません。パケットを処理したコアに応じて、対応したコアのテーブルにフローが作成されます。
- NetFlow ハードウェアの実装では、4 台のハードウェア サンプラーがサポートされています。1/2 ～ 1/1024 のサンプラー レートを選択できます。ランダム サンプリングと確定的サンプリングの両方のモードがサポートされています。
- NetFlow ハードウェアの内部では、ハッシュ テーブルが使用されています。ハードウェア内でハッシュ衝突が発生する場合があります。したがって、内部の連想メモリ（CAM）でオーバーフローが発生しても、実際の NetFlow テーブルの使用率は約 80 % しかない場合があります。
- フローに使用されるフィールドによって異なりますが、単一のフローは 2 個の連続したエントリを取得できます。IPv6 フローとデータリンク フローも 2 個のエントリを取得します。この場合、NetFlow エントリを効果的に使用すれば、テーブル サイズの半分で済みます。これは、上記のハッシュ衝突の制限とは別です。
- デバイスは、最大 15 個のフロー モニタをサポートしています。
- NetFlow ソフトウェアの実装では、分散 NetFlow エクスポートがサポートされるため、フローが作成された同じデバイスからフローがエクスポートされます。
- 入力フローは最初にフローのパケットを受信した ASIC にあります。出力フローは、パケットが実際に デバイス セットアップを残した ASIC にあります。

- バイト カウント フィールドのレポート値（「bytes long」と呼ばれる）は、レイヤ 2 パケット サイズの 18 バイトです。従来のイーサネット トラフィック（802.3）の場合、これは正確です。他のすべてのイーサネット タイプの場合、このフィールドは正確ではありません。「bytes layer2」フィールドを使用すると、常に正確なレイヤ 2 パケット サイズが報告されます。サポートされる Flexible NetFlow フィールドについては、サポートされている Flexible NetFlow フィールド （186 ページ） を参照してください。
- AVC フロー モニタの IPFIX エクスポータの設定はサポートされていません。
- Flexible NetFlow エクスポートは、イーサネット管理ポート（Gi0/0）ではサポートされていません。
- フロー レコードに送信元グループ タグ（SGT）と宛先グループ タグ（DGT）のフィールド（またはこの 2 つのいずれかのフィールド）だけが含まれる場合、両方の値を適用できないとしても、SGT と DGT に値ゼロを設定したフローが作成されます。フロー レコードには、SGT および DGT フィールドと一緒に、送信元および宛先 IP アドレスが含まれる必要があります。
- QoS のマークが付けられたパケットが入力方向に NetFlow が設定されているインターフェイスで受信されると、パケットの QoS 値がコレクタによってキャプチャされます。ただし、パケットが出力方向に NetFlow が設定されているインターフェイスで受信されると、パケットの QoS 値はコレクタによってキャプチャされません。

Flexible NetFlow に関する情報

Flexible NetFlow の概要

Flexible NetFlow ではフローを使用して、アカウンティング、ネットワーク モニタリング、およびネットワーク プランニングに関連する統計情報を提供します。

フローは送信元インターフェイスに届く単方向のパケット ストリームで、キーの値は同じです。キーは、パケット内のフィールドを識別する値です。フローを作成するには、フロー レコードを使用して、フロー固有のキーを定義します。

デバイス は、ネットワーク異常とセキュリティ問題の高度な検出をイネーブルにする Flexible NetFlow 機能をサポートします。Flexible NetFlow により、大量の定義済みフィールドの集合からキーを選択して、特定のアプリケーションに最適なフロー レコードを定義できます。

1 つのフローと見なされるパケットでは、すべてのキー値が一致している必要があります。フローは、設定したエクスポート レコード バージョンに基づいて、関係のある他のフィールドを集めることもあります。フローはFlexible NetFlow キャッシュに格納されます。

エクスポータを使用してFlexible NetFlowがフローのために収集するデータをエクスポートし、Flexible NetFlow コレクタなどのリモート システムにこのデータをエクスポートできます。Flexible NetFlow コレクタは、IPv4 アドレスを使用できます。

モニタを使用してフローのために収集するデータのサイズを定義します。モニタで、フローレコードおよびエクスポータを Flexible NetFlow キャッシュ情報と結合します。

以前の NetFlow と Flexible NetFlow の利点

以前の NetFlow では、フローの判定に固定の 7 タプルの IP 情報を使用していました。

Flexible NetFlow ではフローをユーザが定義できます。次に、Flexible NetFlow の利点を示します。

- スケーラビリティ、フロー情報の集約などの、大容量フロー認識。
- セキュリティの監視と dDoS の検出および識別のための拡張されたフローインフラストラクチャ。
- フロー情報をネットワーク内の特定のサービスまたはオペレーションに適応させるパケットからの新しい情報。利用できるフロー情報は、Flexible NetFlow ユーザがカスタマイズ可能。
- Cisco の柔軟で拡張可能な NetFlow Version 9 および Version 10 エクスポート フォーマットの活用。Version 10 エクスポート フォーマットでは、ワイヤレス クライアントの SSID の可変長フィールドをサポート。
- IP アカウンティング、ボーダー ゲートウェイ プロトコル（BGP）ポリシー アカウンティング、永続的キャッシュなどの多数のアカウンティング機能を置換するために使用できる包括的な IP アカウンティング機能。

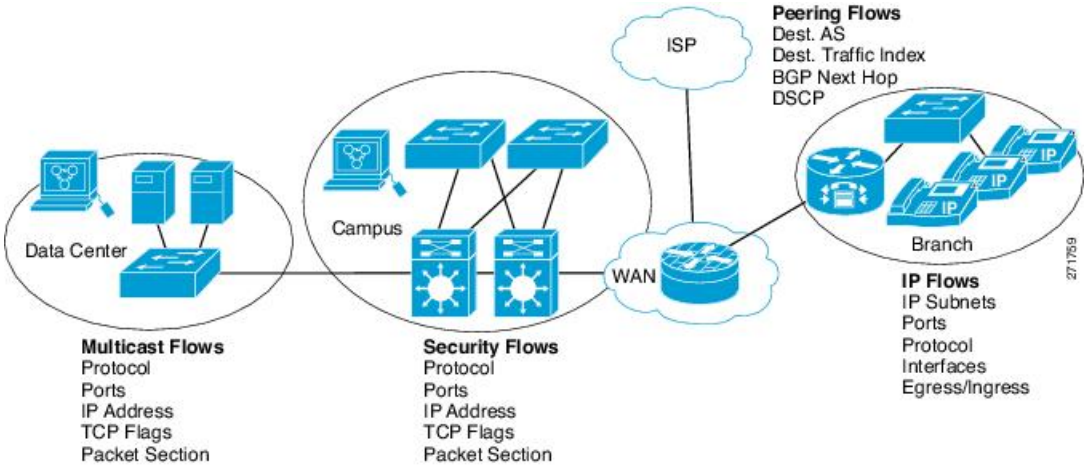
以前の NetFlow では、ネットワーク内のアクティビティを理解して、ネットワーク設計を最適化し、稼働コストを削減できます。

Flexible NetFlow では、ネットワークの動作を、ネットワーク内で使用されるさまざまなサービスに合わせた特定のフロー情報とともに、より効率的に理解できます。次に、Flexible NetFlow 機能用の適用例を示します。

- Flexible NetFlow は Cisco NetFlow をセキュリティ監視ツールとして拡張します。たとえば、ユーザがネットワーク内で特定のタイプの攻撃を検索できるように、パケット長や MAC アドレスのために新しいフロー キーを定義することができます。
- Flexible NetFlow を使用すると、TCP アプリケーションまたは UDP アプリケーションをパケット内のサービス クラス（CoS）ごとに明確に追跡することによって、ホスト間で送信されるアプリケーション トラフィックの量を迅速に識別できます。
- サービス クラスごとに各ネクストホップのマルチプロトコル ラベル スイッチング（MPLS）か IP コア ネットワーク、およびその宛先を入力するトラフィックのアカウンティング。この機能では、エッジ間のトラフィック マトリクスを構築できます。

次の表に、Flexible NetFlow をネットワークに導入する方法の例を示します。

図 *9 : Flexible NetFlow* の通常の導入



Flexible NetFlow のコンポーネント

Flexible NetFlow は、いくつかのバリエーションで一緒に使用して、トラフィック分析およびデータ エクスポートに使用できるコンポーネントで構成されます。Flexible NetFlow のユーザ定義のフロー レコードおよびコンポーネントの構造では、最小限の数のコンフィギュレーション コマンドで、ネットワーキング デバイスでのトラフィック分析およびデータ エクスポートのためのさまざまなコンフィギュレーションの作成が容易になります。各フローモニタに、フロー レコード、フロー エクスポータ、およびキャッシュ タイプの固有の組み合わせを設定できます。フロー エクスポータの宛先 IP アドレスなどのパラメータを変更する場合、フロー エクスポータを使用するすべてのフロー モニタに対して自動的に変更されます。同じフロー モニタを複数のフローサンプラと組み合わると、さまざまなインターフェイス上でさまざな速度の同じタイプのネットワーク トラフィックをサンプリングできます。ここでは、Flexible NetFlow コンポーネントのその他の情報を提供します。

フロー レコード

Flexible NetFlow では、キー フィールドと非キー フィールドの組み合わせをレコードと呼びます。Flexible NetFlow のレコードは Flexible NetFlow フローモニタに割り当てられ、フローデータの格納に使用されるキャッシュが定義されます。Flexible NetFlow には、Flexible NetFlow の使用を開始する際に役立ついくつかの事前定義済みのレコードが含まれています。

フロー レコードでは、フロー内のパケットを識別するために Flexible NetFlow で使用するキーとともに、Flexible NetFlow がフローについて収集する他の関連フィールドを定義します。キーと関連フィールドを任意の組み合わせで指定して、フロー レコードを定義できます。デバイスは、幅広いキー セットをサポートします。フロー レコードでは、フロー単位で収集するカウンタのタイプも定義します。64 ビットのパケットまたはバイト カウンタを設定できます。デバイスは、フロー レコードの作成時に、デフォルトとして次の match フィールドをイネーブルにします。

- match datalink： レイヤ 2 属性

- match flow direction：フローの方向を識別するフィールドとの一致を指定します。
- match interface：インターフェイス属性
- match interface：インターフェイス属性
- match ipv4：IPv4 属性
- match ipv6：IPv6 属性
- match transport：トランスポート層フィールド
- match flow cts：CTS フィールド

NetFlow の事前定義済みのレコード

Flexible NetFlow には事前定義済みのレコードがいくつか含まれ、それを使用してネットワーク トラフィックの監視を開始できます。事前定義済みのレコードは、Flexible NetFlow を迅速に導入するために役立ち、ユーザ定義のフロー レコードよりも簡単に使用できます。ネットワーク モニタリングのニーズを満たす定義済みのレコードのリストから選択できます。Flexible NetFlow が改良されると、一般的なユーザ定義のフロー レコードを事前定義済みレコードとして使用でき、簡単に導入できるようになります。

事前定義済みレコードにより、エクスポートされるデータのために既存の NetFlow コレクタ コンフィギュレーションとの下位互換性が確保されます。事前定義済みレコードは、それぞれ固有の key および nonkey フィールドの組み合わせを持ち、ルータで Flexible NetFlow をカスタマイズしなくても、ネットワーク内のさまざまなタイプのトラフィックを監視する、内蔵機能を提供します。

2 つの事前定義済みレコード（NetFlow original と NetFlow IPv4/IPv6 original output）は機能的に同等で、以前の（入力）NetFlow、および以前の NetFlow の 出力 NetFlow アカウンティング機能をそれぞれエミュレートします。その他の Flexible NetFlow の事前定義済みレコードのいくつかは、以前の NetFlow で利用できる集約キャッシュ方式に基づきます。以前の NetFlow で利用できる集約キャッシュ方式に基づく Flexible NetFlow の事前定義済みレコードでは、集約を実行しません。代わりに、事前定義済みレコードによって各フローが個別に追跡されます。

ユーザ定義レコード

Flexible NetFlow では、key および nonkey フィールドを指定し、実際の要件に合わせてデータ収集をカスタマイズすることで、Flexible NetFlow フロー モニタ キャッシュ用の独自のレコードを定義できます。Flexible NetFlow フロー モニタ キャッシュに対して独自のレコードを定義する場合、ユーザ定義レコードと呼ばれます。nonkeyフィールドの値は、フロー内のトラフィックに関する追加情報を提供するためにフローに追加されます。nonkeyフィールドの値の変更によって新しいフローが作成されることはありません。ほとんどの場合、nonkeyフィールドの値はフロー内の最初のパケットからのみ取得されます。Flexible NetFlow を使用すると、nonkeyフィールドとして、フロー内のバイト数やパケット数などのカウンター値をキャプチャできます。

ユーザ定義レコードは、QoS および帯域幅監視、アプリケーションとユーザのトラフィック プロファイリング、dDoS 攻撃に対するセキュリティ監視などのアプリケーション用に作成で

きます。また、Flexible NetFlow には以前の NetFlow をエミュレートするいくつかの事前定義済みレコードも含まれています。Flexible NetFlow のユーザ定義レコードでは、ユーザが設定可能なサイズのパケットの連続するセクションを監視する機能を利用でき、key フィールドまたは nonkey フィールドとしてパケットのその他のフィールドや属性とともにフロー レコード内で使用します。セクションにはパケットのレイヤ 3 データが含まれる場合があります。パケットのセクション フィールドでは、ユーザが Flexible NetFlow の事前定義済みレコードの対象外のパケット フィールドを監視できます。事前定義済みキーで収集されないパケット フィールドの分析機能によって、さらに詳細なトラフィック モニタリングが可能になるため、dDoS 攻撃の調査に役立ち、URL モニタリングなど他のセキュリティ アプリケーションの実装が可能になります。

Flexible NetFlow では、事前定義済みタイプのユーザが設定可能なサイズのパケット セクションが提供されます。次の Flexible NetFlow コマンド（Flexible NetFlow フロー レコード コンフィギュレーション モードで使用される）をパケット セクションの事前定義済みタイプの設定に使用できます。

- **collectipv4sectionheadersize** *bytes*：各パケットの IPv4 ヘッダーの先頭から *bytes* 引数で指定されたバイト数のキャプチャを開始します。
- **collectipv4sectionpayloadsize** *bytes*：各パケットの IPv4 ヘッダーの直後からバイトのキャプチャを開始します。キャプチャされるバイト数は *bytes* 引数で指定されます。
- **collectipv6sectionheadersize** *bytes*：各パケットの IPv6 ヘッダーの先頭から *bytes* 引数で指定されたバイト数のキャプチャを開始します。
- **collectipv6sectionpayloadsize** *bytes*：各パケットの IPv6 ヘッダーの直後からバイトのキャプチャを開始します。キャプチャされるバイト数は *bytes* 引数で指定されます。

bytes 値は、フロー レコードのこれらのフィールドのサイズ（バイト単位）です。パケットの対応フラグメントが要求されたセクション サイズよりも小さい場合、Flexible NetFlow はフロー レコード内の残りのセクション フィールドを 0 で埋めます。パケット タイプが要求されたセクション タイプと一致しなかった場合、Flexible NetFlow はフロー レコード内のセクション フィールド全体を 0 で埋めます。

Flexible NetFlow では、ヘッダーおよびパケット セクションのタイプに新しいバージョン 9 エクスポート フォーマット フィールド タイプが追加されます。Flexible NetFlow は NetFlow コレクタに、対応するバージョン 9 エクスポート テンプレート フィールドで設定されたセクション サイズを通知します。ペイロード セクションには、対応する長さフィールドがあり、収集されるセクションの実際のサイズを収集するために使用できます。

Flexible NetFlow の match パラメータ

次の表で、Flexible NetFlow の match パラメータについて説明します。フロー レコードごとに、次の match パラメータを 1 つ以上設定する必要があります。

表 *10 : match* パラメータ

コマンド（Command）	目的
match datalink {**dot1q** | **ethertype** | **mac** | **vlan** }	データ リンクまたはレイヤ 2 フィールドとの一致を指定します。次のコマンド オプションが使用可能です。 • **dot1q**：dot1q フィールドと一致します。 • **ethertype**：パケットの ethertype と一致します。 • **mac**：送信元または宛先の MAC フィールドと一致します。 • **vlan**：パケットが配置される VLAN と一致します（入力または出力）。
match flow direction	フローを識別するフィールドとの一致を指定します。
match interface {**input** | **output**}	インターフェイス フィールドとの一致を指定します。次のコマンド オプションが使用可能です。 • **input**：入力インターフェイスと一致します。 • **output**：出力インターフェイスと一致します。
match ipv4 {**destination** | **protocol** | **source** | **tos** | **ttl** | **version**}	IPv4 フィールドとの一致を指定します。次のコマンド オプションが使用可能です。 • **destination**：IPv4 宛先アドレス ベースのフィールドと一致します。 • **protocol**：IPv4 プロトコルと一致します。 • **source**：IPv4 送信元アドレス ベースのフィールドと一致します。 • **tos**：IPv4 タイプ オブ サービス フィールドと一致します。 • **ttl**：IPv4 存続時間フィールドと一致します。 • **version**：IPv4 ヘッダーの IP バージョンと一致します。

コマンド（Command）	目的
match ipv6 {**destination** | **hop-limit** | **protocol** | **source** | **traffic-class** | **version** }	IPv6 フィールドとの一致を指定します。次のコマンド オプションが使用可能です。 • **destination**：IPv6 宛先アドレス ベースのフィールドと一致します。 • **hop-limit**：IPv6 ホップ リミット フィールドと一致します。 • **protocol**：IPv6 ペイロード プロトコル フィールドと一致します。 • **source**：IPv6 送信元アドレス ベースのフィールドと一致します。 • **traffic-class**：IPv6 トラフィック クラスと一致します。 • **version**：IPv6 ヘッダーの IP バージョンと一致します。
match transport {**destination-port** | **igmp** | **icmp** | **source-port**}	トランスポート層フィールドとの一致を指定します。次のコマンド オプションが使用可能です。 • **destination-port**：転送先ポートと一致します。 • **icmp**：ICMP IPv4 および IPv6 フィールドを含む ICMP フィールドと一致します。 • **igmp**：IGMP フィールドと一致します。 • **source-port**：転送元ポートと一致します。
match flow cts {source | destination} group-tag	FNF レコードの CTS フィールドのサポートとの一致を指定します。次のコマンド オプションが使用可能です。 • **source**：ドメインを入力する CTS の送信元と一致します。 • **destination**：ドメインを脱退する CTS の宛先と一致します。

Flexible NetFlow の collect パラメータ

次の表で、Flexible NetFlow の collect パラメータについて説明します。

表 *11 : collect* パラメータ

コマンド（Command）	目的
collect counter { **bytes** { **layer2** { **long** } | **long** } | **packets** { **long** } }	カウンタ フィールドの合計バイト数と合計パケット数を収集します。
collect interface {**input** | **output**}	入力または出力インターフェイスからフィールドを収集します。
collect timestamp absolute {**first** | **last**}	最初のパケットが確認された絶対時間、または最新のパケットが最後に確認された絶対時間のフィールドを収集します（ミリ秒）。
collect transport tcp flags	次の転送 TCP フラグを収集します。 • **ack**：TCP 確認応答フラグ • **cwr**：TCP 輻輳ウィンドウ縮小フラグ • **ece**：TCP ECN エコー フラグ • **fin**：TCP 終了フラグ • **psh**：TCP プッシュ フラグ • **rst**：TCP リセット フラグ • **syn**：TCP 同期フラグ • **urg**：TCP 緊急フラグ （注） デバイスでは、収集する TCP フラグを指定できません。転送 TCP フラグの収集のみ指定できます。すべての TCP フラグはこのコマンドで収集されます。

フロー エクスポータ

フロー エクスポータでは、フロー モニタ キャッシュ内のデータをリモート システム（たとえば、分析および保管のために NetFlow コレクタを実行するサーバ）にエクスポートします。フロー エクスポータは、コンフィギュレーションで別のエンティティとして作成されます。フロー エクスポータは、フロー モニタにデータ エクスポート機能を提供するためにフロー モニタに割り当てられます。複数のフロー エクスポータを作成して、1 つまたは複数のフロー モニタに適用すると、いくつかのエクスポート先を指定することができます。1 つのフロー エクスポータを作成し、いくつかのフロー モニタに適用することができます。

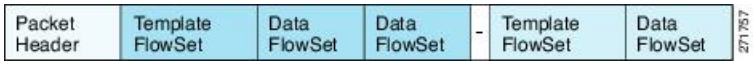
NetFlow データ エクスポート フォーマットのバージョン 9

NetFlow の基本出力はフロー レコードです。NetFlow が改良され、フロー レコードのいくつかのフォーマットが向上しました。NetFlow エクスポート フォーマットの最新の進化は、バージョン 9 と呼ばれます。NetFlow Version 9 エクスポート フォーマットの識別機能は、テンプレートがベースとなります。テンプレートは、レコードフォーマットの設計を拡張可能なものにします。NetFlow サービスが将来拡張されても、基本フロー レコード フォーマットを変更し続ける必要がありません。テンプレートを使用すると、次のいくつかの利点があります。

- NetFlow のコレクタを提供したり、サービスを表示したりするアプリケーションを作成するサードパーティ ビジネス パートナーは、新規の NetFlow 機能が追加されるたびにアプリケーションを再コンパイルする必要はありません。代わりに、既知のテンプレート フォーマットを記述する外部のデータ ファイルを使用することができます。
- 新規機能は、現在の導入環境を損ねることなく、NetFlow に迅速に追加できます。
- バージョン 9 フォーマットは新しいプロトコルや開発中のプロトコルに適応できるため、NetFlow はこれらのプロトコルに対して「将来的に対応」します。

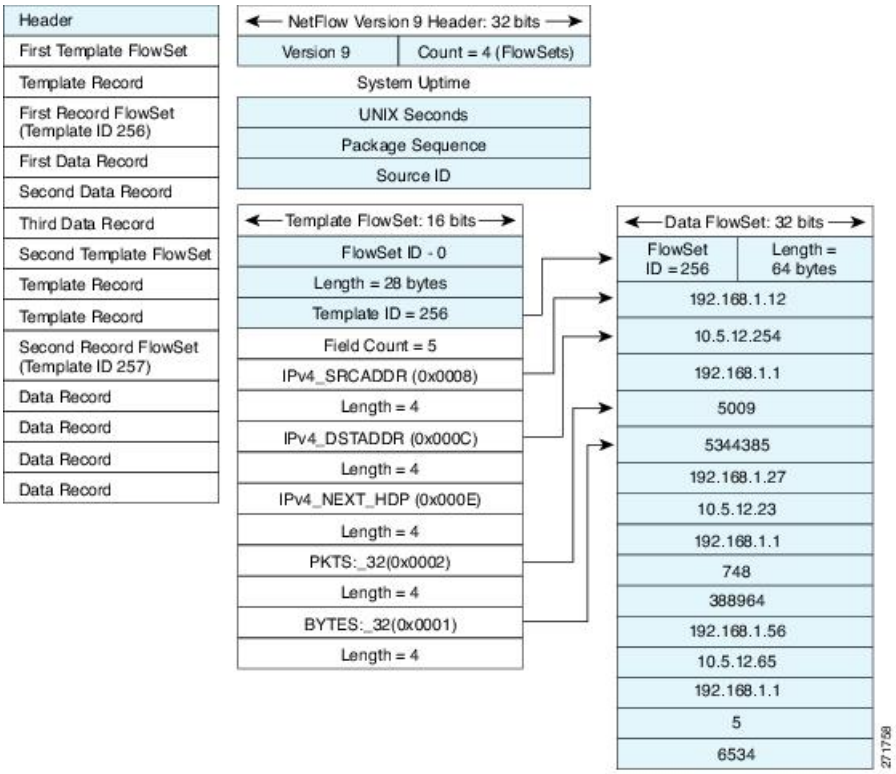
バージョン 9 のエクスポート フォーマットは、パケット ヘッダーとそれに続く 1 つ以上のテンプレート フロー セットまたはデータ フロー セットで構成されています。テンプレート フローセットでは、将来のデータ フローセットに表示されるフィールドの説明が提供されます。このようなデータ フロー セットは、後で同じエクスポート パケットまたは後続のエクスポート パケットで発生する可能性があります。テンプレート フロー セットおよびデータ フロー セットは、次の図に示すように、単一のエクスポート パケットに混在させることができます。

図 *10* : バージョン *9* エクスポート パケット



NetFlow Version 9 では、送信されるデータを NetFlow コレクタが理解できるように、テンプレート データを定期的にエクスポートします。また、テンプレートのデータ フロー セットもエクスポートします。Flexible NetFlow の主な利点は、ユーザがフロー レコードを設定すると、バージョン 9 テンプレートに効率的に変換され、コレクタに転送されることです。下の図に、ヘッダー、テンプレート フロー セットおよびデータ フロー セットを含めて、NetFlow Version 9 エクスポート フォーマットの詳細な例を示します。

図 *11* : *NetFlow* バージョン *9* エクスポート フォーマットの詳細例



バージョン 9 エクスポート フォーマットの詳細については、ホワイト ペーパー『Cisco IOS NetFlow Version 9 Flow-Record Format』を参照してください。次の URL から入手できます。http://www.cisco.com/en/US/tech/tk648/tk362/technologies_white_paper09186a00800a3db9.shtml

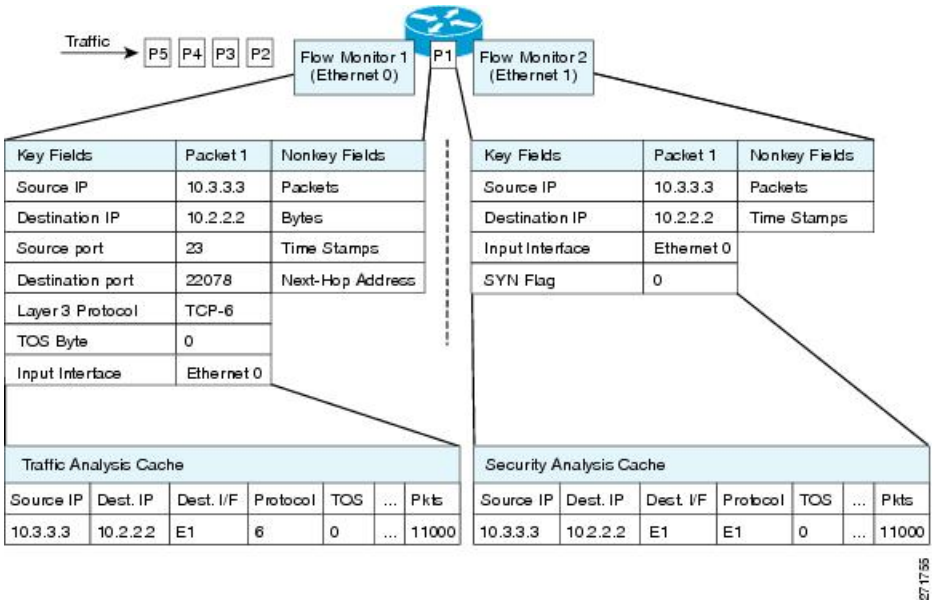
フロー モニタ

フロー モニタは Flexible NetFlow のネットワーク トラフィックの監視を実行するコンポーネントで、インターフェイスに適用されます。

フロー データはネットワーク トラフィックから収集され、フロー レコードの key フィールドおよび nonkey フィールドに基づいて監視プロセス中にフロー モニタ キャッシュに追加されます。

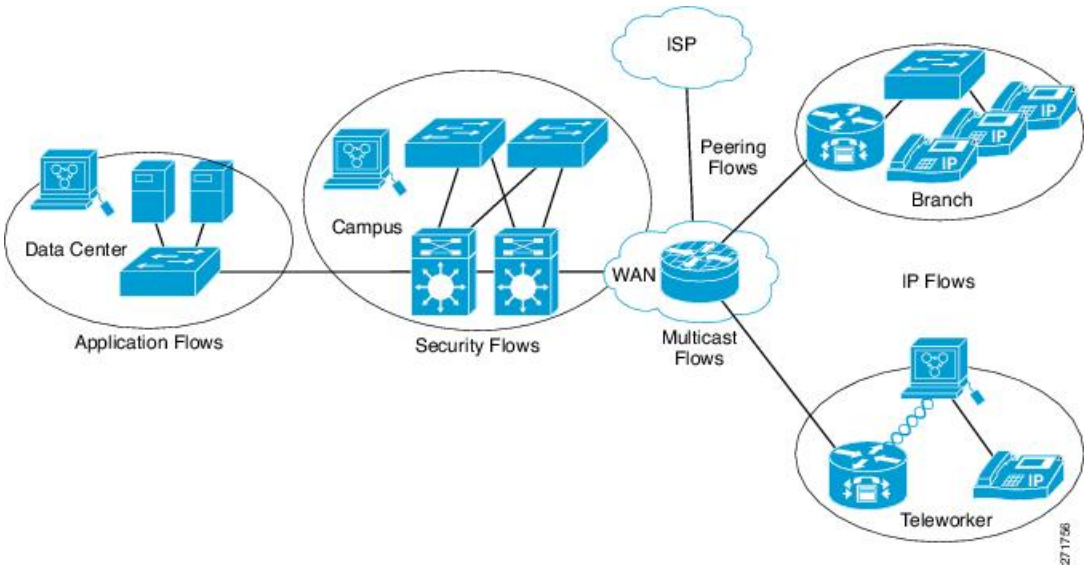
Flexible NetFlow は、同じトラフィックのさまざまなタイプの分析を実行するために使用できます。下の図では、入力インターフェイス上の標準トラフィック分析のために設計されたレコードと、出力インターフェイス上のセキュリティ分析のために設計されたレコードを使用してパケット 1 が分析されます。

図 *12 : 2* つのフロー モニタを使用した同じトラフィックの分析例



下の図に、カスタム レコードを使用して複数のタイプのフロー モニタを適用するより複雑な方法の例を示します。

図 *13 :* カスタム レコードでの複数のタイプのフロー モニタの複雑な使用例



3 つのタイプのフロー モニタ キャッシュがあります。フロー モニタの作成後に、そのフロー モニタで使用するキャッシュ タイプを変更します。3 タイプのフロー モニタ キャッシュについては、次の各項に説明があります。

標準

デフォルトのキャッシュ タイプは「normal」です。このモードでは、キャッシュ内のエントリが timeout active 設定と timeout inactive 設定に従って期限切れになります。キャッシュ エントリは、期限切れになるとキャッシュから削除され、設定されている何らかのエクスポータによってエクスポートされます。

緊急

「immediate」タイプのキャッシュは、作成されるとすぐにレコードを期限切れにします。その結果、どのフローにも 1 パケットしか含まれません。キャッシュ内容を表示するコマンドでは、パケットの履歴が表示されます。

予想されるフローが非常に少なく、パケットが検出されてからレポートがエクスポートされるまでの遅延を最小限にする場合は、このモードが適しています。

注意 このモードでは大量のエクスポートデータが生じて、低速のリンクが過負荷状態になり、エクスポート先のシステムに著しく影響する可能性があります。処理するパケット数を削減するようにサンプリングを設定することをお勧めします。

（注） キャッシュ タイムアウト設定は、このモードでは何の効果もありません。

永続的

タイプが「permanent」のキャッシュでは、フローが期限切れになることはありません。permanent キャッシュは、検出が予想されるフローの数が少なく、ルータに長期間の統計情報を保存する必要がある場合に便利です。たとえば、フロー レコード内の key フィールドが 8 ビット IP ToS フィールドだけで、256 フローだけを監視する場合があります。ネットワーク トラフィックの IP ToS フィールドの使用状況を長期間に渡って監視するには、permanent キャッシュを使用します。permanent キャッシュは、課金アプリケーション、および追跡対象が固定セットのフローに対する、全域におよぶトラフィック マトリクスに役立ちます。アップデート メッセージは、「timeout update」設定に従って設定されたすべてのフロー エクスポータに、定期的に送信されます。

（注） permanent モードでキャッシュがいっぱいになった場合は、新しいフローが監視されなくなります。そうなった場合は、キャッシュの統計情報に「Flows not added」というメッセージが示されます。

（注） permanent キャッシュでは、デルタ カウンタではなくアップデート カウンタが使用されます。そのため、フローがエクスポートされると、カウンタにはフローのライフタイム全体の総検出数が示され、最後のエクスポート送信後に検出された追加パケットは示されません。

フロー サンプラー

フローサンプラーは、ルータのコンフィギュレーションで別のコンポーネントとして作成されます。フロー サンプラーは、分析用に選択されるパケットの数を制限することで、 Flexible NetFlow を実行しているデバイス上の負荷を減らすために使用されます。

フロー サンプリングでは、ルータのパフォーマンスに対するモニタリング精度が交換されます。サンプラーをフロー モニタに適用すると、フロー モニタが分析する必要のあるパケット数が減少するため、ルータでフロー モニタを実行するためのオーバーヘッド負荷が低下します。フロー モニタで分析されるパケット数が減少すると、それに応じて、フロー モニタのキャッシュに格納される情報の精度が低下します。

ip flow monitor コマンドを使用してインターフェイスに適用する場合、サンプラーとフローモニタを組み合わせます。

サポートされている Flexible NetFlow フィールド

次の表では、さまざまなトラフィック タイプおよびトラフィック方向について、Flexible NetFlow（FNF）でサポートされるフィールドの統合リストを提供しています。

（注） パケットに VLAN フィールドがある場合、その長さは考慮されません。

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注記（Notes）
Key または Collect フィールド							

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注記（Notes）
インターフェイス入力	○	—	○	—	○	—	フロー モニタを入力方向に適用する場合： • **match** キーワードを使用し、入力インターフェイスを key フィールドとして使用します。 • **collect** キーワードを使用し、出力インターフェイスを collect フィールドとして使用します。このフィールドはエクスポートされるレコードに含まれますが、値は 0 になります。
インターフェイス出力	—	○	—	○	—	○	フロー モニタを出力方向に適用する場合： • **match** キーワードを使用し、出力インターフェイスを key フィールドとして使用します。 • **collect** キーワードを使用し、入力インターフェイスを collect フィールドとして使用します。このフィールドはエクスポートされるレコードに含まれますが、値は 0 になります。

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注記（Notes）
Key フィールド							

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注記（Notes）
フロー方向	○	○	○	○	○	○	
Ethertype	○	○	—	—	—	—	
VLAN 入力	○	—	○	—	○	—	スイッチポートでのみサポートされています。
VLAN 出力	—	○	—	○	—	○	スイッチポートでのみサポートされています。
dot1q VLAN 入力	○	—	○	—	○	—	スイッチポートでのみサポートされています。
dot1q VLAN 出力	—	○	—	○	—	○	スイッチポートでのみサポートされています。
dot1q 優先度	○	○	○	○	○	○	スイッチポートでのみサポートされています。
MAC 送信元アドレス入力	○	○	○	○	○	○	

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注記（Notes）
MAC 送信元アドレス出力	—	—	—	—	—	—	
MAC 宛先アドレス入力	○	—	○	—	○	—	
MAC 送信先アドレス出力	—	○	—	○	—	○	
IPv4 バージョン	—	—	○	○	○	○	
IPv4 TOS	—	—	○	○	○	○	
IPv4 プロトコル	—	—	○	○	○	○	送信元/宛先ポート、ICMP コード/タイプ、IGMP タイプ、TCP フラグのいずれかが使用されている場合に使用する必要があります。
IPv4 TTL	—	—	○	○	○	○	
IPv4 送信元アドレス（IPv4 source address）	—	—	○	○	—	—	

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注記（Notes）
IPv4 宛先アドレス（IPv4 destination address）	—	—	○	○	—	—	
ICMP IPv4 タイプ	—	—	○	○	—	—	
ICMP IPv4 コード	—	—	○	○	—	—	
IGMP タイプ	—	—	○	○	—	—	

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注記（Notes）
Key フィールド（続き）							
IPv6 バージョン	—	—	○	○	○	○	IP バージョンと同じです。

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注記（Notes）
IPv6 プロトコル	—	—	○	○	○	○	IP プロトコルと同じです。送信元/宛先ポート、ICMP コード/タイプ、IGMP タイプ、TCP フラグのいずれかが使用されている場合に使用する必要があります。
IPv6 送信元アドレス（IPv6 source address）	—	—	—	—	○	○	
IPv6 宛先アドレス（IPv6 destination address）	—	—	—	—	○	○	
IPv6 トラフィック クラス	—	—	○	○	○	○	IP TOS と同じです。
IPv6 ホップ リミット	—	—	○	○	○	○	IP TTL と同じです。
ICMP IPv6 タイプ	—	—	—	—	○	○	
ICMP IPv6 コード	—	—	—	—	○	○	

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注記（Notes）
source-port	—	—	○	○	○	○	
dest-port	—	—	○	○	○	○	

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注記（Notes）
Collect フィールド							
Bytes long	○	○	○	○	○	○	パケットサイズ=（FCS を含むイーサネットフレームサイズ-18 バイト） **推奨**： このフィールドを回避し、Bytes layer2 long を使用します。
Packets long	○	○	○	○	○	○	
Timestamp absolute first	○	○	○	○	○	○	
Timestamp absolute last	○	○	○	○	○	○	
TCP フラグ	○	○	○	○	○	○	すべてのフラグを収集します。

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPV4 Out	IPv6 In	IPv6 Out	注記（Notes）
Bytes layer2 long	○	○	○	○	○	○	

デフォルト設定

次の表は、デバイスに対する Flexible NetFlow のデフォルト設定を示します。

表 *12* : デフォルトの *Flexible NetFlow* 設定

設定	デフォルト
フロー アクティブ タイムアウト	1800 秒
フロー タイムアウトの非アクティブ化	15 秒

Flexible NetFlow の設定方法

Flexible Netflow を設定するには、次の一般的な手順に従います。

1. フローにキー フィールドおよび非キー フィールドを指定して、フロー レコードを作成します。
2. プロトコルを指定して任意のフローエクスポータを作成し、宛先ポート、宛先、およびその他のパラメータを転送します。
3. フロー レコードおよびフロー エクスポータに基づいて、フロー モニタを作成します。
4. 任意のサンプラーを作成します。
5. レイヤ 2 ポート、レイヤ 3 ポート、または VLAN にフロー モニタを適用します。

カスタマイズしたフロー レコードの作成

カスタマイズしたフロー レコードを設定するには、次のタスクを実行します。

カスタマイズしたフロー レコードは、特定の目的でトラフィック データを分析するために使用します。カスタマイズしたフロー レコードには、key フィールドとして使用する **match** 基準が 1 つ以上必要です。通常は nonkey フィールドとして使用する **collect** 基準が 1 つ以上あります。

カスタマイズしたフロー レコードの順列は、数百もの可能性があります。このタスクでは、可能性のある順列の 1 つを作成するための手順について説明します。必要に応じて当該タスクの手順を変更し、要件に合わせてカスタマイズしたフロー レコードを作成します。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
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	**flowrecord** *record-name* 例： `Device(config)# flow record FLOW-RECORD-1`	フロー レコードを作成し、Flexible NetFlow フロー レコード コンフィギュレーション モードを開始します。 • このコマンドでは、既存のフロー レコードを変更することもできます。
ステップ 4	**description** 説明 例： `Device(config-flow-record)# description Used for basic traffic analysis`	（任意）フロー レコードの説明を作成します。
ステップ 5	**match** {**ip** | **ipv6**} {**destination** | **source**} **address** 例： `Device(config-flow-record)# match ipv4 destination address`	フロー レコードの key フィールドを設定します。 （注） この例では、IPv4宛先アドレスをレコードの key フィールドとして設定します。**matchipv4** コマンドで利用できるその他の key フィールド、および key フィールドの設定に利用できる他の **match** コマンドの詳細について。
ステップ 6	必要に応じてステップ 5 を繰り返し、レコードの追加 key フィールドを設定します。	—

	コマンドまたはアクション	目的

ステップ 7	**match flow cts** {**source** | **destination**} **group-tag**
	例：
	`Device(config-flow-record)# match flow cts source group-tag`
	`Device(config-flow-record)# match flow cts destination group-tag`
		（注） この例では、CTS の送信元グループ タグと宛先グループ タグをレコードのキーフィールドとして設定します。**matchipv4** コマンドで利用できるその他の key フィールド、および key フィールドの設定に利用できる他の **match** コマンドの詳細について。

	コマンドまたはアクション	目的
		（注） • Ingress: • 着信パケットでは、ヘッダーがある場合、SGT にはヘッダーと同じ値が反映されます。値がない場合は、0 が示されます。 • DGT 値は入力ポートの SGACL 設定に依存しません。 • Egress: • SGT または CTS のいずれかの伝播が出力インターフェイス上で無効化されていると、SGT は 0 になります。 • 発信パケットで、（SGT、DGT）に対応する SGACL 設定が存在すれば、DGT はゼロ以外になります。 • SGACL が出力ポート/VLANで無効化されているか、またはグローバル SGACL の強制を無効化されている場合、DGT は 0 になります。
ステップ 8	**例**：	入力インターフェイスをレコードの nonkey フィールドとして設定します。 （注） この例では、入力インターフェイスをレコードの nonkey フィールドとして設定します。

	コマンドまたはアクション	目的
ステップ 9	必要に応じて上記のステップを繰り返し、レコードの追加 nonkey フィールドを設定します。	—
ステップ 10	**end** 例： `Device(config-flow-record)# end`	Flexible NetFlow フロー レコード コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 11	**showflowrecord** *record-name* 例： `Device# show flow record FLOW_RECORD-1`	（任意）指定したフロー レコードの現在のステータスが表示されます。
ステップ 12	**showrunning-configflowrecord** *record-name* 例： `Device# show running-config flow record FLOW_RECORD-1`	（任意）指定したフロー レコードの設定が表示されます。

フロー エクスポータの作成

フロー エクスポートを作成して、フローのエクスポート パラメータを定義できます。

（注） フローエクスポータごとに、1 つ宛先のみがサポートされます。複数の宛先にデータをエクスポートする場合は、複数のフロー エクスポータを設定してフロー モニタに割り当てる必要があります。

IPv4 アドレスを使用して宛先にエクスポートできます。

手順

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

	コマンドまたはアクション	目的
ステップ 2	**flow exporter** 名前 例： `Device(config)# flow exporter ExportTest`	フローエクスポータを作成し、フローエクスポータ 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	**description** *string* 例： `Device(config-flow-exporter)# description ExportV9`	（任意）最大 63 文字で、このフローの説明を指定します。
ステップ 4	**destination** {*ipv4-address*} 例： `Device(config-flow-exporter)# destination 192.0.2.1 (IPv4 destination)`	このエクスポータに IPv4 宛先アドレスまたはホスト名を設定します。
ステップ 5	**dscp** *value* 例： `Device(config-flow-exporter)# dscp 0`	（任意）DSCP（DiffServ コードポイント）値を指定します。指定できる範囲は 0 ～ 63 です。デフォルトは 0 です。
ステップ 6	**source** { |} 例： `Device(config-flow-exporter)# source gigabitEthernet1/0/1`	（任意）設定された宛先で NetFlow コネクタに到達するために使用するインターフェイスを指定します。送信元として次のインターフェイスを設定できます。 \
ステップ 7	**transportudp** *number* 例： `Device(config-flow-exporter)# transport udp 200`	（任意）NetFlow コレクタに到達するために使用する UDP ポートを指定します。
ステップ 8	**ttl** 秒 例： `Device(config-flow-exporter)# ttl 210`	（任意）エクスポータによって送信されるデータグラムの存続可能時間（TTL）値を設定します。範囲は 1 ～ 255 秒です。デフォルトは 255 です。

	コマンドまたはアクション	目的
ステップ 9	**export-protocol netflow-v9** 例： `Device(config-flow-exporter)# export-protocol netflow-v9`	エクスポータで使用される NetFlow エクスポート プロトコルのバージョンを指定します。
ステップ 10	**end** 例： `Device(config-flow-record)# end`	特権 EXEC モードに戻ります。
ステップ 11	**show flow exporter** [**name** *record-name*] 例： `Device# show flow exporter ExportTest`	（任意）NetFlow のフロー エクスポータ情報を表示します。
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。

次のタスク

フロー レコードおよびフロー エクスポータに基づいて、フロー モニタを定義します。

カスタマイズしたフロー モニタの作成

カスタマイズしたフロー モニタを作成するには、この必須のタスクを実行します。

各フロー モニタには、専用のキャッシュが割り当てられています。フロー モニタごとに、キャッシュ エントリの内容およびレイアウトを定義するレコードが必要です。これらのレコード フォーマットは、事前定義済みのレコード フォーマットのいずれか、またはユーザ定義にすることができます。上級のユーザであれば **flowrecord** コマンドを使用して、カスタマイズしたフォーマットを作成することもできます。

始める前に

Flexible NetFlow の事前定義済みレコードの代わりにカスタマイズしたレコードを使用する場合は、このタスクを実行する前に、カスタマイズしたレコードを作成する必要があります。データをエクスポートするためにフロー エクスポータをフロー モニタに追加する場合は、このタスクを完了する前にエクスポータを作成する必要があります。

（注） フロー モニタで **record** コマンドのパラメータを変更する前に、**no ip flow monitor** コマンドを使用して、フロー モニタを適用したすべてのインターフェイスから、フロー モニタを削除しておく必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
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	**flow monitor** *monitor-name* 例： `Device(config)# flow monitor FLOW-MONITOR-1`	フロー モニタを作成し、Flexible NetFlow フロー モニタ コンフィギュレーション モードを開始します。 • このコマンドでは、既存のフロー モニタを変更することもできます。
ステップ 4	**description** 説明 例： `Device(config-flow-monitor)# description Used for basic ipv4 traffic analysis`	（任意）フローモニタの説明を作成します。
ステップ 5	**record** {*record-name* | **netflow-original** | **netflow** {**ipv4** | **ipv6**} *record* [**peer**]} 例： `Device(config-flow-monitor)# record FLOW-RECORD-1`	フロー モニタのレコードを指定します。
ステップ 6	**cache** {**entries** *number* | **timeout** {**active** | **inactive** | **update**} *seconds* | {**immediate** | **normal** | **permanent**}} 例：	**timeout** キーワードに関連するキーワードの値は、キャッシュ タイプが **immediate** に設定されている場合には反映されません。

	コマンドまたはアクション	目的
ステップ 7	必要に応じてステップ 6 を繰り返して、このフローモニタのキャッシュパラメータの変更を完了します。	—
ステップ 8	**statisticspacket protocol** 例： `Device(config-flow-monitor)# statistics packet protocol`	（任意）Flexible NetFlow モニタのプロトコル分散統計情報の収集をイネーブルにします。
ステップ 9	**statisticspacket size** 例： `Device(config-flow-monitor)# statistics packet size`	（任意）Flexible NetFlow モニタのサイズ分散統計情報の収集をイネーブルにします。
ステップ 10	**exporter** *exporter-name* 例： `Device(config-flow-monitor)# exporter EXPORTER-1`	（任意）事前に作成されたエクスポータの名前を指定します。
ステップ 11	**end** 例： `Device(config-flow-monitor)# end`	Flexible NetFlow フローモニタ コンフィギュレーションモードを終了して、特権 EXEC モードに戻ります。
ステップ 12	**showflowmonitor**[[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]] [**statistics**]] 例： `Device# show flow monitor FLOW-MONITOR-2 cache`	（任意）Flexible NetFlow フローモニタのステータスおよび統計情報を表示します。
ステップ 13	**showrunning-configflowmonitor** *monitor-name* 例： `Device# show running-config flow monitor FLOW_MONITOR-1`	（任意）指定したフローモニタの設定が表示されます。

フロー サンプリングの設定および有効化

フロー サンプラーを設定して有効化するには、この必須のタスクを実行します。

（注） 「NetFlow original」/「NetFlow IPv4 original input」/「NetFlow IPv6 original input」事前定義済みレコードをフロー モニタに指定して、以前の NetFlow をエミュレートする場合は、フロー モニタを入力（受信）トラフィックの分析だけに使用できます。

「NetFlow IPv4 original output」/「NetFlow IPv6 original output」事前定義済みレコードをフロー モニタに指定して、出力NetFlowアカウンティング機能をエミュレートする場合は、フロー モニタを出力（発信）トラフィックの分析だけに使用できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**enable** 例： `Device> enable`	特権 EXEC モードをイネーブルにします。 • プロンプトが表示されたら、パスワードを入力します。
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	**sampler** *sampler-name* 例： `Device(config)# sampler SAMPLER-1`	サンプラーを作成し、サンプラー コンフィギュレーション モードを開始します。 • このコマンドでは、既存のサンプラーを変更することもできます。
ステップ 4	**description** 説明 例： `Device(config-sampler)# description Sample at 50%`	（任意）フローサンプラーの説明を作成します。
ステップ 5	**mode** {**random**} **1 out-of** *window-size* 例： `Device(config-sampler)# mode random 1 out-of 2`	サンプラー モードおよびフローサンプラーのウィンドウ サイズを指定します。 • *window-size* 引数の範囲は、2 ～ 32768 です。

	コマンドまたはアクション	目的
ステップ 6	**exit** 例： `Device(config-sampler)# exit`	サンプラー コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードに戻ります。
ステップ 7	**interface** *type number* 例： `Device(config)# interface GigabitEthernet 0/0/0`	インターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 8	{**ip** | **ipv6**} **flowmonitor** *monitor-name* [[**sampler**] *sampler-name*] {**input** | **output**} 例： `Device(config-if)# ip flow monitor FLOW-MONITOR-1 sampler SAMPLER-1 input`	作成したフロー モニタおよびフロー サンプラーをインターフェイスに割り当てて、サンプリングをイネーブルにします。
ステップ 9	**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。
ステップ 10	**showsamplersampler-name** 例： `Device# show sampler SAMPLER-1`	設定し有効化したフロー サンプラーのステータスおよび統計情報を表示します。

インターフェイスへのフローの適用

フロー モニタおよびオプションのサンプラーをインターフェイスに適用できます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**interface** *type* 例： `Device(config)# interface`	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、インターフェイスを設定します。

	コマンドまたはアクション	目的
	GigabitEthernet1/0/1	Flexible NetFlow は、L2 ポートチャネル インターフェイスではサポートされませんが、L2 ポートチャネル メンバー ポートではサポートされます。 Flexible NetFlow は、L3 ポートチャネル インターフェイスではサポートされませんが、L3 ポートチャネル メンバー ポートではサポートされます。 インターフェイス コンフィギュレーション のコマンド パラメータは次のとおりです。
ステップ 3	{**ip flow monitor** | **ipv6 flow monitor**}*name* [| **sampler** *name*] {**input**} 例： `Device(config-if)# ip flow monitor MonitorTest input`	入力または出力パケットに対応するインターフェイスに、IPv4 または IPv6 フロー モニタ、およびオプションのサンプラーを関連付けます。 入力と出力の両方向でインターフェイスに複数のモニタを関連付けることができます。
ステップ 4	**end** 例： `Device(config-flow-monitor)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show flow interface** [*interface-type number*] 例： `Device# show flow interface`	（任意）インターフェイスの NetFlow 情報を表示します。
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。

VLAN 上でのブリッジ型 NetFlow の設定

フロー モニタおよびオプションのサンプラーを VLAN に適用できます。

手順

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

ステップ 2	**vlan** [**configuration**] *vlan-id*
	例：
	`Device(config)# vlan configuration 30`
	`Device(config-vlan-config)#`
		VLAN または VLAN 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 flow monitor** *monitor name* [**sampler** *sampler name*] {**input** } 例： `Device(config-vlan-config)# ip flow monitor MonitorTest input`	入力パケットに対応する VLAN に、フロー モニタおよびオプションのサンプラーを関連付けます。
ステップ 4	**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。

レイヤ 2 NetFlow の設定

Flexible NetFlow レコード内でレイヤ 2 キーを定義できます。このレコードを使用して、レイヤ 2 インターフェイスのフローをキャプチャできます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configureterminal** 例： `Device# configure terminal`	グローバル コンフィギュレーション モードを開始します。
ステップ 2	**flow record** 名前 例：	フロー レコード 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(config)# flow record L2_record` `Device(config-flow-record)#`	
ステップ 3	**match datalink** {**dot1q** |**ethertype** | **mac** | **vlan**} 例： `Device(config-flow-record)# match datalink ethertype`	レイヤ2属性をキーとして指定します。
ステップ 4	**end** 例： `Device(config-flow-record)# end`	特権 EXEC モードに戻ります。
ステップ 5	**show flow record** [*name*] 例： `Device# show flow record`	（任意）インターフェイスの NetFlow 情報を表示します。
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。

Flexible NetFlow の監視

次の表にあるコマンドを使用して、Flexible NetFlow をモニタリングできます。

表 *13 : Flexible NetFlow* のモニタリング コマンド

コマンド（**Command**）	目的
show flow exporter [**broker** | **export-ids** | **name** | *name* | **statistics** | **templates**]	NetFlow のフロー エクスポータ情報と統計情報を表示します。
show flow exporter [**name** *exporter-name*]	NetFlow のフロー エクスポータ情報と統計情報を表示します。
show flow interface	NetFlowインターフェイスに関する情報を表示します。

コマンド（Command）	目的
show flow monitor [**name** *exporter-name*]	NetFlow のフロー モニタ情報と統計情報を表示します。
show flow monitor statistics	フロー モニタの統計情報を表示します。
show flow monitor cache format {**table** | **record** | **csv**}	指定された形式でフロー モニタのキャッシュの内容を表示します。
show flow record [**name** *record-name*]	NetFlowのフロー レコード情報を表示します。
show sampler [**broker** | **name** | *name*]	NetFlowサンプラに関する情報を表示します。

Flexible NetFlow の設定例

例：フローの設定

フローを作成し、そのフローをインターフェイスに適用する例を示します。

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.

Device(config)# flow export export1
Device(config-flow-exporter)# destination 10.0.101.254
Device(config-flow-exporter)# transport udp 2055
Device(config-flow-exporter)# exit
Device(config)# flow record record1
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match ipv4 protocol
Device(config-flow-record)# match transport source-port
Device(config-flow-record)# match transport destination-port
Device(config-flow-record)# match flow cts source group-tag
Device(config-flow-record)# match flow cts destination group-tag
Device(config-flow-record)# collect counter byte long
Device(config-flow-record)# collect counter packet long
Device(config-flow-record)# collect timestamp absolute first
Device(config-flow-record)# collect timestamp absolute last
Device(config-flow-record)# exit
Device(config)# flow monitor monitor1
Device(config-flow-monitor)# record record1
Device(config-flow-monitor)# exporter export1
Device(config-flow-monitor)# exit
Device(config)# interface tenGigabitEthernet 1/0/1
Device(config-if)# ip flow monitor monitor1 input
Device(config-if)# end
```

例：IPv4 入力トラフィックのモニタリング

次の例は、IPv4 入力トラフィックをモニタする方法を示しています（int g1/0/11 は、int g1/0/36 および int g3/0/11 にトラフィックを送信します）。

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# flow record  fr-1
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match interface input
Device(config-flow-record)# collect counter bytes long
Device(config-flow-record)# collect counter packets long
Device(config-flow-record)# collect timestamp absolute first
Device(config-flow-record)# collect timestamp absolute last
Device(config-flow-record)# collect counter bytes layer2 long
Device(config-flow-record)# exit

Device(config)# flow exporter fe-ipfix6
Device(config-flow-exporter)# destination 2001:0:0:24::10
Device(config-flow-exporter)# source Vlan106
Device(config-flow-exporter)# transport udp 4739
Device(config-flow-exporter)# export-protocol ipfix
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow exporter fe-ipfix
Device(config-flow-exporter)# description IPFIX format collector 100.0.0.80
Device(config-flow-exporter)# destination 100.0.0.80
Device(config-flow-exporter)# dscp 30
Device(config-flow-exporter)# ttl 210
Device(config-flow-exporter)# transport udp 4739
Device(config-flow-exporter)# export-protocol ipfix
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow exporter fe-1
Device(config-flow-exporter)# destination 10.5.120.16
Device(config-flow-exporter)# source Vlan105
Device(config-flow-exporter)# dscp 32
Device(config-flow-exporter)# ttl 200
Device(config-flow-exporter)# transport udp 2055

Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow monitor fm-1
Device(config-flow-monitor)# exporter fe-ipfix6
Device(config-flow-monitor)# exporter fe-ipfix
Device(config-flow-monitor)# exporter fe-1
Device(config-flow-monitor)# cache timeout inactive 60
Device(config-flow-monitor)#  cache timeout active 180
Device(config-flow-monitor)# record fr-1
Device(config-flow-monitor)# end

Device# show running-config interface g1/0/11
Device# show running-config interface g1/0/36
Device# show running-config interface g3/0/11
```

```
Device# show flow monitor fm-1 cache format table
```

例：IPv4 出力トラフィックのモニタリング

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# flow record  fr-1 out
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match interface output
Device(config-flow-record)# collect counter bytes long
Device(config-flow-record)# collect counter packets long
Device(config-flow-record)# collect timestamp absolute first
Device(config-flow-record)# collect timestamp absolute last
Device(config-flow-record)# exit

Device(config)# flow exporter fe-1
Device(config-flow-exporter)# destination 10.5.120.16
Device(config-flow-exporter)# source Vlan105
Device(config-flow-exporter)# dscp 32
Device(config-flow-exporter)# ttl 200
Device(config-flow-exporter)# transport udp 2055
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow exporter fe-ipfix6
Device(config-flow-exporter)# destination 2001:0:0:24::10
Device(config-flow-exporter)# source Vlan106
Device(config-flow-exporter)# transport udp 4739
Device(config-flow-exporter)# export-protocol ipfix
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow exporter fe-ipfix
Device(config-flow-exporter)# description IPFIX format collector 100.0.0.80
Device(config-flow-exporter)# destination 100.0.0.80
Device(config-flow-exporter)# dscp 30
Device(config-flow-exporter)# ttl 210
Device(config-flow-exporter)# transport udp 4739
Device(config-flow-exporter)# export-protocol ipfix
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

Device(config)# flow monitor fm-1-output
Device(config-flow-monitor)# exporter fe-1
Device(config-flow-monitor)# exporter fe-ipfix6
Device(config-flow-monitor)# exporter fe-ipfix
Device(config-flow-monitor)# cache timeout inactive 50
Device(config-flow-monitor)#  cache timeout active 120
Device(config-flow-monitor)# record fr-1-out
Device(config-flow-monitor)# end

Device# show flow monitor fm-1-output cache format table
```

NetFlow に関する追加情報

関連資料

関連項目	参照先
この章で使用するコマンドの完全な構文および使用方法の詳細。	*Command Reference (Catalyst 9500 Series Switches)*

標準および RFC

標準/RFC	役職（Title）
RFC 3954	『Cisco Systems NetFlow Services Export Version 9』

MIB

MIB	MIB リンク
本リリースでサポートするすべての MIB	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャ セットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

テクニカル サポート

説明	リンク
シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンライン リソースを提供しています。 お使いの製品のセキュリティ情報や技術情報を入手するために、Product Alert Tool（Field Notice からアクセス）、Cisco Technical Services Newsletter、Really Simple Syndication（RSS）フィードなどの各種サービスに加入できます。 シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。	http://www.cisco.com/support

Flexible NetFlow の機能情報

リリース	変更箇所
Cisco IOS XE Everest 16.5.1a	この機能が導入されました。