



Cisco IOS XE Fuji 16.9.x (Catalyst 9300 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：2018年7月18日

最終更新：2018年8月26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

Autoconf の設定 1

Autoconf の前提条件 1

Autoconf の制約事項 1

Autoconf に関する情報 2

Autoconf の利点 2

アイデンティティセッション管理とテンプレート 2

Autoconf の動作 3

テンプレートを使用する利点 6

Autoconf の機能 7

Autoconf の設定方法 8

エンドデバイスへの組み込みテンプレートの適用 8

エンドデバイスへの変更された組み込みテンプレートの適用 12

ASP から Autoconf への移行 14

Autoconf の設定例 15

例：エンドデバイスへの組み込みテンプレートの適用 15

例：エンドデバイスへの変更された組み込みテンプレートの適用 16

例：ASP マクロから Autoconf への移行 16

Autoconf のその他の参考資料 16

Autoconf の機能履歴 17

第 2 章

Cisco プラグアンドプレイの設定 19

Cisco プラグアンドプレイの設定 19

第 3 章

Cisco Discovery Protocol の設定 21

Cisco Discovery Protocol について	21
Cisco Discovery Protocol の概要	21
Cisco Discovery Protocol のデフォルト設定	22
Cisco Discovery Protocol の設定方法	22
Cisco Discovery Protocol の特性の設定	22
Cisco Discovery Protocol のディセーブル化	24
Cisco Discovery Protocol の有効化	25
インターフェイス上で Cisco Discovery Protocol をディセーブルにします。	27
インターフェイス上での Cisco Discovery Protocol のイネーブル化	29
Cisco Discovery Protocol のモニタリングとメンテナンス	30

第 4 章

簡易ネットワーク管理プロトコルの設定 33

SNMP の前提条件	33
SNMP の制約事項	35
SNMP に関する情報	36
SNMP の概要	36
SNMP マネージャ機能	36
SNMP エージェント機能	37
SNMP コミュニティストリング	37
SNMP MIB 変数アクセス	38
SNMP 通知	38
SNMP ifIndex MIB オブジェクト値	39
SNMP のデフォルト設定	39
SNMP 設定時の注意事項	39
SNMP の設定方法	40
コミュニティストリングの設定	40
SNMP グループおよびユーザの設定	43
SNMP 通知の設定	47
エージェント コンタクトおよびロケーションの設定	51
SNMP を通して使用する TFTP サーバの制限	52
SNMP エージェントのディセーブル化	54

SNMP ステータスのモニタリング	55
SNMP の例	56
簡易ネットワーク管理プロトコルの機能の履歴と情報	57

第 5 章

サービス レベル契約の設定	59
SLA の制約事項	59
SLA について	60
Cisco IOS IP サービス レベル契約 (SLA)	60
Cisco IOS IP SLA でのネットワーク パフォーマンスの測定	61
IP SLA レスポンダおよび IP SLA 制御プロトコル	61
IP SLA の応答時間の計算	62
IP SLA 動作のスケジューリング	63
IP SLA 動作のしきい値のモニタリング	63
UDP ジッター	64
IP SLA 動作の設定方法	65
デフォルト設定	65
設定時の注意事項	65
IP SLA レスポンダの設定	66
IP SLA ネットワーク パフォーマンス測定の実装	68
UDP ジッター動作を使用した IP サービス レベルの分析	72
ICMP エコー動作を使用した IP サービス レベルの分析	76
IP SLA 動作のモニタリング	79
IP SLA 動作のモニタリングの例	80
その他の参考資料	81

第 6 章

SPAN および RSPAN の設定	83
SPAN および RSPAN の前提条件	83
SPAN および RSPAN の制約事項	83
SPAN および RSPAN について	86
SPAN および RSPAN	86
ローカル SPAN	86

リモート SPAN	87
SPAN と RSPAN の概念および用語	88
SPAN および RSPAN と他の機能の相互作用	94
SPAN と RSPAN とデバイス スタック	95
フローベースの SPAN	96
SPAN および RSPAN のデフォルト設定	97
設定時の注意事項	97
SPAN 設定時の注意事項	97
RSPAN 設定時の注意事項	97
FSPAN および FRSPAN 設定時の注意事項	98
SPAN および RSPAN の設定方法	98
ローカル SPAN セッションの作成	98
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	101
フィルタリングする VLAN の指定	104
RSPAN VLAN としての VLAN の設定	106
RSPAN 送信元セッションの作成	108
フィルタリングする VLAN の指定	110
RSPAN 宛先セッションの作成	112
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	115
FSPAN セッションの設定	117
FRSPAN セッションの設定	120
SPAN および RSPAN 動作のモニタリング	124
SPAN および RSPAN の設定例	124
例：ローカル SPAN の設定	124
例：RSPAN VLAN の作成	125
SPAN および RSPAN の機能の履歴と情報	127

第 7 章

ERSPAN の設定 129

ERSPAN の設定	129
ERSPAN の設定の前提条件	129
ERSPAN 設定時の制約事項	129

ERSPAN の設定に関する情報	130
ERSPAN の概要	130
ERSPAN 送信元	131
ERSPAN の設定方法	131
ERSPAN 送信元セッションの設定	131
ERSPAN の設定例	134
例：ERSPAN 送信元セッションの設定	134
ERSPAN の確認	134
その他の参考資料	136
ERSPAN の設定に関する機能情報	136

第 8 章

パケットキャプチャの設定	139
パケットキャプチャ設定の前提条件	139
Wireshark 設定の前提条件	139
組み込みパケットキャプチャ設定の前提条件	139
パケットキャプチャ設定の制約事項	140
Wireshark 設定の制約事項	140
組み込みパケットキャプチャの制約事項	141
パケットキャプチャについて	142
Wireshark について	142
キャプチャポイント	143
接続ポイント	143
フィルタ	143
アクション	145
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	145
.pcap ファイルにキャプチャされたパケットのストレージ	145
パケットのデコードおよび表示	146
パケットのストレージおよび表示	147
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	147
Wireshark 機能	148
Wireshark 設定のガイドライン	149

デフォルトの Wireshark の設定	152
組み込みパケットキャプチャについて	152
組み込みパケット キャプチャの利点	152
パケット データ キャプチャ	153
パケットキャプチャの設定方法	153
Wireshark の設定方法	153
キャプチャ ポイントの定義	154
キャプチャ ポイント パラメータの追加または変更	159
キャプチャ ポイント パラメータの削除	161
キャプチャ ポイントの削除	162
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	163
キャプチャ ポイント バッファのクリア	167
組み込みパケット キャプチャの実装方法	168
パケット データ キャプチャの管理	168
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	170
パケットキャプチャの設定例	171
Wireshark の設定例	171
例：.pcap ファイルからの概要出力の表示	171
例：.pcap ファイルからの詳細出力の表示	172
例：.pcap ファイルからパケット ダンプ出力の表示	173
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	174
例：.pcap ファイルにキャプチャされたパケットの数を表示	175
例：.pcap ファイルから単一パケット ダンプの表示	175
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	175
例：単純なキャプチャおよび表示	175
例：単純なキャプチャおよび保存	177
例：バッファのキャプチャの使用	179
例：出力方向のパケットの簡単なキャプチャおよび保存	185
組み込みパケット キャプチャの設定例	187
例：パケット データ キャプチャの管理	187
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	187

その他の参考資料	189
パケットキャプチャ設定の機能履歴と情報	190

第 9 章

Flexible NetFlow の設定	191
Flexible NetFlow の前提条件	191
Flexible Netflow に関する制約事項	192
Flexible NetFlow に関する情報	195
Flexible NetFlow の概要	195
以前の NetFlow と Flexible NetFlow の利点	195
Flexible NetFlow のコンポーネント	196
フロー レコード	197
フロー エクスポータ	201
フロー モニター	203
フロー サンプラー	205
サポートされている Flexible NetFlow フィールド	205
デフォルト設定	211
Flexible NetFlow : 入力 VRF サポートの概要	211
自律システム番号	211
Flexible NetFlow の設定方法	212
フロー レコードの作成	212
フロー エクスポータの作成	215
カスタマイズしたフロー モニターの作成	217
フロー サンプラーの作成	220
インターフェイスへのフローの適用	222
VLAN 上でのブリッジ型 NetFlow の設定	224
レイヤ 2 NetFlow の設定	224
Flexible NetFlow の監視	226
設定例 Flexible NetFlow	226
例 : フローの設定	226
例 : IPv4 入力トラフィックのモニタリング	227
例 : IPv4 出力トラフィックのモニタリング	228

例：入力 VRF サポート用の Flexible NetFlow の設定	229
Flexible NetFlow の機能情報	230

第 10 章

暗号化トラフィック分析の設定	231
暗号化トラフィック分析の制約事項	231
暗号化トラフィック分析について	231
概要	231
Flexible NetFlow と ETA の設定	232
非アクティブ タイマーとエクスポート	232
暗号化トラフィック分析の設定方法	233
エクスポート IP とポートの設定	233
非アクティブ タイマー値の設定	233
暗号化トラフィック分析の有効化	234
暗号化トラフィック分析の設定例	235
例：エクスポート IP とポートの設定	235
例：非アクティブ タイマーの設定	235
例：et-analytics の有効化	235
例：et-analytics 設定の確認	235
その他の参考資料	236
暗号化トラフィック分析の機能履歴と情報	237



第 1 章

Autoconf の設定

Autoconfは、ネットワークのアクセスレイヤに展開されるエンドデバイス上のデータまたは音声 VLAN、Quality of Service (QoS) パラメータ、ストーム制御、および MAC ベースのポートセキュリティのポート設定を管理するために使用できるソリューションです。

- [Autoconf の前提条件](#) (1 ページ)
- [Autoconf の制約事項](#) (1 ページ)
- [Autoconf に関する情報](#) (2 ページ)
- [Autoconf の設定方法](#) (8 ページ)
- [Autoconf の設定例](#) (15 ページ)
- [Autoconf のその他の参考資料](#) (16 ページ)
- [Autoconf の機能履歴](#) (17 ページ)

Autoconf の前提条件

- Autoconf を有効にする前に、Auto SmartPort (ASP) マクロ、デバイス分類子を無効にしてから、セッションモニターにアクセスします。

Autoconf の制約事項

- ASP マクロと Autoconf は、同じインターフェイスでは同時にサポートされません。Autoconf または ASP のいずれかをインターフェイスごとのレベルで無効にする必要があります。
- インターフェイス テンプレートは、ワイヤレスセッションには適用されません。
- **autoconf enable** コマンドを使用して Autoconf 機能を有効にすると、デフォルトの Autoconf サービスポリシーがすべてのインターフェイスに適用されます。**service-policy** コマンドを使用して他のサービスポリシーをグローバルに適用することはできません。別のサービスポリシーを適用するには、そのインターフェイスで Autoconf を無効にする必要があります。サービスポリシーをグローバルに適用する場合は、Autoconf 機能を無効にしてから有効にする必要があります。

- ローカル（インターフェイスレベル）ポリシーとグローバルサービスポリシーの両方が存在する場合、ローカルポリシーが優先されます。ローカルサービスポリシー内のイベントが処理され、グローバルサービスポリシーは適用されません。グローバルサービスポリシーは、ローカルポリシーが削除された場合にのみ有効になります。
- サービステンプレートはインターフェイスに適用できません。また、インターフェイステンプレートはサービスインスタンスに適用できません。
- インターフェイステンプレート内にネストできるサービステンプレートは1つだけです。

Autoconf に関する情報

Autoconf の利点

Autoconf機能により、エンドデバイスとインターフェイス間のハードバインドが可能になります。Autoconfは、Smart Operations ソリューションに含まれます。Smart Operationsは、LANスイッチの導入を簡素化し改善できる包括的な機能セットです。Smart Operationsは、組織が優れた運用を実現し、ネットワーク上でサービスを拡張できるように支援します。

Autoconf機能は、デバイスポートに必要な設定を自動的に適用し、インターフェイステンプレート内で設定された一連のインターフェイス設定を使用して、直接接続された各エンドデバイスの効率的なパフォーマンスを実現します。

- Autoconfは、パーサーが毎回各コマンドを解析する必要がないため、コマンドをインターフェイスに効率的に適用します。
- Autoconf機能を使用して適用された設定は、ポートの以前の設定または後続の設定に影響を与えることなく、確実にポートから削除できます。
- Autoconf機能は、インターフェイスおよびサービステンプレートを使用して、組み込みの設定およびユーザー定義の設定を提供します。テンプレートを使用して適用された設定は、1回の操作で一元的に更新できます。
- Autoconf機能を使用すると、ポートおよびアクセスセッションに設定を適用できます。
- Autoconf機能は、デバイスと接続されたエンドデバイスを直感的で自動設定可能にすることで、継続的なメンテナンスを削減します。これにより、運用コスト（OPEX）が削減され、総所有コスト（TCO）が削減されます。

アイデンティティセッション管理とテンプレート

Autoconf機能の主な利点は、コアセッション管理機能がアプリケーション固有のロジックから分離されていることです。これにより、ポリシー決定の基準や適用されるポリシーの性質に関係なく、同じフレームワークを使用できます。

アイデンティティセッション管理インフラストラクチャを使用すると、設定やポリシーをテンプレートとして適用できます。

サービステンプレートとインターフェイステンプレートは両方とも、設定とポリシーの名前付きコンテナです。サービステンプレートはアクセスセッションにのみ適用でき、インターフェイステンプレートはポートにのみ適用できます。サービステンプレートがアクセスセッションに適用されると、含まれる設定/ポリシーはターゲットセッションにのみ適用され、同じアクセスポートでホストされる可能性のある他のセッションには影響しません。同様に、インターフェイステンプレートがアクセスポートに適用されると、そのポートで交換されるすべてのトラフィックに影響します。

Autoconf機能は、一連の組み込みマップと組み込みテンプレートを使用します。組み込みテンプレートは、インターフェイス設定のベストプラクティスに基づいて設計されています。組み込みテンプレートは、カスタマイズされた設定を含めるようにユーザーが変更できるため、新しいテンプレートを作成する必要がありません。

ユーザーが作成したテンプレートは、ユーザー定義テンプレートと呼ばれます。ユーザー定義のテンプレートはデバイス上で定義でき、任意の組み込みトリガーまたはユーザー定義トリガーにマッピングできます。

Autoconfテンプレートと手動設定によって適用される全体的な適用設定を表示するには、**show derived-config** コマンドを使用します。**show running-config interface type number** コマンドの出力に表示されるインターフェイスコマンドは、必ずしも動作設定ではありません。Autoconf機能は、インターフェイスにテンプレートを動的に適用し、すでに適用されている競合する静的設定を上書きします。

Autoconf の動作

Autoconfは、デバイス分類子を使用して、ポートに接続されているエンドデバイスを識別します。

Autoconf機能は、Cisco Discovery Protocol、LLDP、DHCP、MACアドレスから収集したデバイス分類情報、およびデバイス分類子によって識別される組織固有識別子（OUI）を使用します。

デバイス分類子は、改善されたデバイス分類機能および精度、拡張デバイス可視性、および拡張設定管理を提供します。

グローバル コンフィギュレーションモードで **autoconf enable** コマンドを使用して Autoconf機能を有効にすると、デバイス分類が有効になります。

デバイス検出はイベントトリガーとして機能し、適切な自動テンプレートをインターフェイスに適用します。

Autoconf機能は、3層階層に基づいています。

- ポリシーマップは、Autoconf機能を適用するためのトリガータイプを識別します。
- パラメータマップは、エンドデバイスに基づいて、適用する必要がある適切なテンプレートを識別します。

- テンプレートには、適用する設定が含まれています。

Autoconf の組み込みテンプレートとトリガーは、これら 3 つの手順を自動的に実行します。

Autoconf 機能は、次の組み込みテンプレートを提供します。

- AP_INTERFACE_TEMPLATE
- DMP_INTERFACE_TEMPLATE
- IP_CAMERA_INTERFACE_TEMPLATE
- IP_PHONE_INTERFACE_TEMPLATE
- LAP_INTERFACE_TEMPLATE
- MSP_CAMERA_INTERFACE_TEMPLATE
- MSP_VC_INTERFACE_TEMPLATE
- PRINTER_INTERFACE_TEMPLATE
- ROUTER_INTERFACE_TEMPLATE
- SWITCH_INTERFACE_TEMPLATE
- TP_INTERFACE_TEMPLATE



(注) デフォルトでは、組み込みテンプレートは実行中の設定では表示されません。組み込みテンプレートは、編集した場合にのみ実行中のコンフィギュレーションに表示されます。

選択されるテンプレートは、インターフェイスに適用されるパラメータマップ情報に基づいています。この情報は、次の基準に基づく場合があります。

- エンドデバイスタイプ
- MAC アドレス
- OUI
- ユーザー ロール
- ユーザー名

Autoconf 機能は、次の設定の 1 つの組み込みパラメータマップ (BUILTIN_DEVICE_TO_TEMPLATE) を提供します。

```
Parameter-map name: BUILTIN_DEVICE_TO_TEMPLATE
Map: 10 map device-type regex "Cisco-IP-Phone"
Action(s):
  20 interface-template IP_PHONE_INTERFACE_TEMPLATE
Map: 20 map device-type regex "Cisco-IP-Camera"
Action(s):
  20 interface-template IP_CAMERA_INTERFACE_TEMPLATE
Map: 30 map device-type regex "Cisco-DMP"
Action(s):
```

```

20 interface-template DMP_INTERFACE_TEMPLATE
Map: 40 map oui eq "00.0f.44"
Action(s):
  20 interface-template DMP_INTERFACE_TEMPLATE
Map: 50 map oui eq "00.23.ac"
Action(s):
  20 interface-template DMP_INTERFACE_TEMPLATE
Map: 60 map device-type regex "Cisco-AIR-AP"
Action(s):
  20 interface-template AP_INTERFACE_TEMPLATE
Map: 70 map device-type regex "Cisco-AIR-LAP"
Action(s):
  20 interface-template LAP_INTERFACE_TEMPLATE
Map: 80 map device-type regex "Cisco-TelePresence"
Action(s):
  20 interface-template TP_INTERFACE_TEMPLATE
Map: 90 map device-type regex "Surveillance-Camera"
Action(s):
  10 interface-template MSP_CAMERA_INTERFACE_TEMPLATE
Map: 100 map device-type regex "Video-Conference"
Action(s):
  10 interface-template MSP_VC_INTERFACE_TEMPLATE

```



- (注) 組み込みパラメータマップの設定を表示するには、**show parameter-map type subscriber attribute-to-service All** コマンドを使用します。

Autoconf機能は、次の設定の1つの組み込みポリシーマップ (BUILTIN_AUTOCONF_POLICY) を提供します。

```

BUILTIN_AUTOCONF_POLICY
event identity-update match-all
  10 class always do-until-failure
    10 map attribute-to-service table BUILTIN_DEVICE_TO_TEMPLATE

```



- (注) 組み込みポリシーマップの設定を表示するには、**show policy-map type control subscriber BUILTIN_AUTOCONF_POLICY** コマンドを使用します。

ポリシーマップ、パラメータマップ、およびテンプレートを手動で作成することもできます。

特定のユーザー情報に基づいてトリガーが作成されると、ローカル 802.1X Cisco Identity Services Engine (ISE) サーバーがトリガーを認証し、操作のセキュリティを確保します。

インターフェイステンプレートは、次のいずれかの方法を使用して (インターフェイス上で) 動的にアクティブ化できます。

- **RADIUS CoA** : 認可変更 (CoA) コマンドは 1 つ以上のアクセスセッションを対象としていますが、参照されるテンプレートは、参照されるセッションをホストするインターフェイスに適用される必要があります。
- **RADIUS Access-Accept** (クライアントの認証または認可用) : Access-Accept で返される参照先インターフェイステンプレートは、認可されたアクセスセッションをホストしているポートに適用する必要があります。

- サービステンプレート：ローカルに定義されたサービステンプレートまたはAAAサーバーから取得されたサービステンプレートでインターフェイステンプレートが参照されている場合、そのインターフェイステンプレートは、サービステンプレートが適用されるアクセスセッションをホストするインターフェイスに適用する必要があります（ローカルに定義されたサービステンプレート内からインターフェイステンプレートを参照するための新しいコマンドを追加します）。
- 加入者制御ポリシーアクション：加入者制御ポリシーに基づくマッピングアクションは、フィルタのタイプに基づいてサービステンプレートやインターフェイステンプレート（パラメータマップで参照される）をアクティブ化し、以前のポリシーに関連付けられたテンプレートを削除します。
- デバイスからテンプレートへのパラメータマップ：サービスおよびインターフェイステンプレートへのフィルタタイプのマッピングを効率的かつ読み取り可能な方法で指定できる加入者パラメータマップ。

テンプレートを使用する利点

自動構成にテンプレートを使用すると、次の利点があります。

- テンプレートは、定義時に一度解析されます。これにより、テンプレートの動的な適用が非常に効率的になります。
- テンプレートは、エンドデバイスのタイプに基づいて、エンドデバイスに接続されているイーサネットインターフェイスに適用できます。
- サービステンプレートを使用すると、セッション指向機能をアクティブ化できます。一方、インターフェイステンプレートは、セッションをホストしているインターフェイスに設定を適用します。
- サービステンプレートはアクセスセッションに適用されるため、ポート上の単一のエンドポイントと交換されるトラフィックにのみ影響します。
- デバイスのスタートアップ設定と実行中の設定は、テンプレートの動的な適用によって変更されません。
- ポリシーの適用は、アクセスセッションのライフサイクルと同期されます。アクセスセッションのライフサイクルは、リンクアップ/リンクダウンだけを含む、利用可能なすべての手法を使用してフレームワークによって追跡されます。
- テンプレートは、1回の操作で更新できます。テンプレートの適用されたすべてのインスタンスが更新されます。
- テンプレートの構成コマンドは、実行中の設定には表示されません。
- 以前の設定や後続の設定に影響を与えることなく、テンプレートを削除できます。
- テンプレートアプリケーションが認識され、同期が可能になり、障害が発生した場合は修復アクションが実行されます。

- データ VLAN、Quality of Service (QoS) パラメータ、ストーム制御、および MAC ベースのポートセキュリティは、スイッチに接続されているエンドデバイスに基づいて自動的に設定されます。
- スイッチポートは、デバイスがポートから切断されたときに設定を削除することで完全にクリーンアップされます。
- インストールと設定のプロセスにおける人的エラーが減少します。

Autoconf の機能

Autoconf 機能は、デフォルトではグローバル コンフィギュレーション モードで無効になっています。Autoconf 機能をグローバル コンフィギュレーション モードで有効にすると、デフォルトでインターフェイスレベルで有効になります。組み込みテンプレート設定は、すべてのインターフェイスで検出されたエンドデバイスに基づいて適用されます。

Autoconf がグローバルレベルで有効になっている場合でも、インターフェイスレベルで手動で Autoconf を無効にするには、**access-session inherit disable autoconf** コマンドを使用します。

Autoconf をグローバルレベルで無効にすると、すべてのインターフェイスレベルの設定が無効になります。

グローバル	インターフェイスレベル	AutoConf ステータス
無効	無効	エンドデバイスが接続されている場合、自動設定は適用されません。
有効	デフォルトで有効	Autoconf がグローバルレベルで有効になっている場合は、デフォルトではインターフェイスレベルで有効です。組み込みテンプレート設定は、すべてのインターフェイスで検出されたエンドデバイスに基づいて適用されます。
有効	無効	グローバルレベルで有効です。インターフェイスレベルで無効です。Autoconf が無効になっているインターフェイスにエンドデバイスが接続されている場合、自動設定は適用されません。

Autoconf では、Autoconf スティッキ機能を設定することで、エンドデバイスへのリンクがダウンしている場合やエンドデバイスが切断されている場合でも、テンプレートを保持できます。**access-session interface-template sticky** コマンドを使用して、グローバル コンフィギュレーション モードで Autoconf スティッキ機能を設定します。Autoconf スティッキ機能により、エンドデバイスを検出し、リンクフラップまたはデバイスが取り外されて接続し直されるたびにテンプレートを適用する必要がなくなります。

access-session interface-template sticky コマンドは、**access-session** コマンドを含む組み込みテンプレートをインターフェイスに適用するために必須です。サービスポリシーを使用してポー

トにインターフェイス テンプレートを適用するには **access-session interface-template sticky** コマンドを設定します。

特定のインターフェイスで Autoconf 機能を無効にするには、インターフェイス コンフィギュレーションモードで **access-session inherit disable interface-template-sticky** コマンドを使用します。

Autoconf の設定方法

エンドデバイスへの組み込みテンプレートの適用

次のタスクでは、Cisco IP 電話などのエンドデバイスに接続されているインターフェイスに組み込みテンプレートを適用する方法を示します。

始める前に

Cisco IP 電話などのエンドデバイスがスイッチポートに接続されていることを確認します。

手順の概要

1. **enable**
2. **configure terminal**
3. **autoconf enable**
4. **end**
5. (任意) **show device classifier attached interface** *interface-type interface-number*
6. **show template binding target** *interface-type interface-number*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device(config)# 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	autoconf enable 例： Device(config)# autoconf enable	Autoconf 機能を有効にします。
ステップ 4	end 例：	グローバル コンフィギュレーション モードを終了し、特権 EXEC 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)# end	
ステップ 5	(任意) show device classifier attached interface <i>interface-type interface-number</i> 例： Device# show device classifier attached interface Gi3/0/26	エンドデバイスが正しい属性を持つデバイス分類子によって分類されているかどうかを表示します。
ステップ 6	show template binding target <i>interface-type</i> <i>interface-number</i> 例： Device# show template binding target gi3/0/26	インターフェイス上のテンプレートを介して適用された設定を表示します。

エンドデバイスのデバイス分類の確認

インターフェイスのインターフェイス テンプレートの確認

インターフェイス コンフィギュレーションの確認

Autoconf 適用後のグローバル設定の確認

次の例は、IP 電話が正しい属性を持つデバイス分類子によって分類されていることを示しています。

```
Device# show device classifier attached interface GigabitEthernet 3/0/26
```

```
Summary:
```

```
MAC_Address      Port_Id      Profile Name      Device Name
=====
0026.0bd9.7bbb   Gi3/0/26    Cisco-IP-Phone-7962   Cisco IP Phone 7962
```

次の例は、組み込みインターフェイス テンプレートがインターフェイスに適用されることを示しています。

```
Device# show template binding target GigabitEthernet 3/0/26
```

```
Interface Templates
=====
Interface: Gi4/0/11
Method      Source      Template-Name
-----
dynamic     Built-in    IP_PHONE_INTERFACE_TEMPLATE
```

次の例は、インターフェイス テンプレートが GigabitEthernet インターフェイス 3/0/26 に接続された IP 電話に適用された後にインターフェイス設定を確認する方法を示しています。

```
Device# show running-config interface GigabitEthernet 3/0/26
Building configuration...
```

```
Current configuration : 624 bytes
```

```

!
interface GigabitEthernet3/0/26
!
End

Device# show derived-config interface GigabitEthernet 3/0/26

Building configuration...

Derived configuration : 649 bytes
!
interface GigabitEthernet3/0/26
  switchport mode access
  switchport block unicast
  switchport port-security maximum 3
  switchport port-security maximum 2 vlan access
  switchport port-security violation restrict
  switchport port-security aging time 2
  switchport port-security aging type inactivity
  switchport port-security
  load-interval 30
  storm-control broadcast level pps 1k
  storm-control multicast level pps 2k
  storm-control action trap
  spanning-tree portfast
  spanning-tree bpduguard enable
  service-policy input AutoConf-4.0-CiscoPhone-Input-Policy
  service-policy output AutoConf-4.0-Output-Policy
  ip dhcp snooping limit rate 15
end

Device# show running config
class-map match-any AutoConf-4.0-Scavenger-Queue
  match dscp cs1
  match cos 1
  match access-group name AutoConf-4.0-ACL-Scavenger
class-map match-any AutoConf-4.0-VoIP
  match dscp ef
  match cos 5
class-map match-any AutoConf-4.0-Control-Mgmt-Queue
  match cos 3
  match dscp cs7
  match dscp cs6
  match dscp cs3
  match dscp cs2
  match access-group name AutoConf-4.0-ACL-Signaling
class-map match-any AutoConf-4.0-Multimedia-Conf
  match dscp af41
  match dscp af42
  match dscp af43
class-map match-all AutoConf-4.0-Broadcast-Vid
  match dscp cs5
class-map match-any AutoConf-4.0-Bulk-Data
  match dscp af11
  match dscp af12
  match dscp af13
class-map match-all AutoConf-4.0-Realtime-Interact
  match dscp cs4
class-map match-any AutoConf-4.0-VoIP-Signal
  match dscp cs3
  match cos 3
class-map match-any AutoConf-4.0-Trans-Data-Queue
  match cos 2
  match dscp af21

```

```
match dscp af22
match dscp af23
match access-group name AutoConf-4.0-ACL-Transactional-Data
class-map match-any AutoConf-4.0-VoIP-Data
  match dscp ef
  match cos 5
class-map match-any AutoConf-4.0-Multimedia-Stream
  match dscp af31
  match dscp af32
  match dscp af33
class-map match-all AutoConf-4.0-Internetwork-Ctrl
  match dscp cs6
class-map match-all AutoConf-4.0-VoIP-Signal-Cos
  match cos 3
class-map match-any AutoConf-4.0-Multimedia-Stream-Queue
  match dscp af31
  match dscp af32
  match dscp af33
class-map match-all AutoConf-4.0-Network-Mgmt
  match dscp cs2
class-map match-all AutoConf-4.0-VoIP-Data-Cos
  match cos 5
class-map match-any AutoConf-4.0-Priority-Queue
  match cos 5
  match dscp ef
  match dscp cs5
  match dscp cs4
class-map match-any AutoConf-4.0-Bulk-Data-Queue
  match cos 1
  match dscp af11
  match dscp af12
  match dscp af13
  match access-group name AutoConf-4.0-ACL-Bulk-Data
class-map match-any AutoConf-4.0-Transaction-Data
  match dscp af21
  match dscp af22
  match dscp af23
class-map match-any AutoConf-4.0-Multimedia-Conf-Queue
  match cos 4
  match dscp af41
  match dscp af42
  match dscp af43
  match access-group name AutoConf-4.0-ACL-Multimedia-Conf
class-map match-all AutoConf-4.0-Network-Ctrl
  match dscp cs7
class-map match-all AutoConf-4.0-Scavenger
  match dscp cs1
class-map match-any AutoConf-4.0-Signaling
  match dscp cs3
  match cos 3
!
!
policy-map AutoConf-4.0-Cisco-Phone-Input-Policy
class AutoConf-4.0-VoIP-Data-Cos
  set dscp ef
  police cir 128000 bc 8000
  exceed-action set-dscp-transmit cs1
  exceed-action set-cos-transmit 1
class AutoConf-4.0-VoIP-Signal-Cos
  set dscp cs3
  police cir 32000 bc 8000
  exceed-action set-dscp-transmit cs1
  exceed-action set-cos-transmit 1
class class-default
```

```
    set dscp default
    set cos 0
policy-map AutoConf-4.0-Output-Policy
class AutoConf-4.0-Scavenger-Queue
    bandwidth remaining percent 1
class AutoConf-4.0-Priority-Queue
    priority
    police cir percent 30 bc 33 ms
class AutoConf-4.0-Control-Mgmt-Queue
    bandwidth remaining percent 10
class AutoConf-4.0-Multimedia-Conf-Queue
    bandwidth remaining percent 10
class AutoConf-4.0-Multimedia-Stream-Queue
    bandwidth remaining percent 10
class AutoConf-4.0-Trans-Data-Queue
    bandwidth remaining percent 10
    dbl
class AutoConf-4.0-Bulk-Data-Queue
    bandwidth remaining percent 4
    dbl
class class-default
    bandwidth remaining percent 25
    dbl
policy-map AutoConf-DMP
class class-default
    set dscp cs2
policy-map AutoConf-IPVSC
class class-default
    set cos dscp table AutoConf-DscpToCos
policy-map AutoConf-4.0-Input-Policy
class AutoConf-4.0-VoIP
class AutoConf-4.0-Broadcast-Vid
class AutoConf-4.0-Realtime-Interact
class AutoConf-4.0-Network-Ctrl
class AutoConf-4.0-Internetwork-Ctrl
class AutoConf-4.0-Signaling
class AutoConf-4.0-Network-Mgmt
class AutoConf-4.0-Multimedia-Conf
class AutoConf-4.0-Multimedia-Stream
class AutoConf-4.0-Transaction-Data
class AutoConf-4.0-Bulk-Data
class AutoConf-4.0-Scavenger
```

エンドデバイスへの変更された組み込みテンプレートの適用

次のタスクは、複数のワイヤレスアクセスポイントと IP カメラがスイッチに接続されている場合に、組み込みテンプレートを変更する方法を示しています。

手順の概要

1. **enable**
2. **configure terminal**
3. **template** *template-name*
4. **switchport access vlan** *vlan-id*
5. **description** *description*
6. **exit**

7. **autoconf enable**
8. **end**
9. **show template interface binding all**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device(config)# 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	template <i>template-name</i> 例： Device(config)# template AP_INTERFACE_TEMPLATE	組み込みテンプレートのテンプレート 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	switchport access vlan <i>vlan-id</i> 例： Device(config-template)# switchport access vlan 20	インターフェイスがアクセスモードのときに VLAN を設定します。
ステップ 5	description <i>description</i> 例： Device(config-template)# description modifiedAP	組み込みテンプレートの説明を変更します。
ステップ 6	exit 例： Device(config-template)# exit	テンプレート 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 モードを開始します。
ステップ 7	autoconf enable 例： Device(config)# autoconf enable	Autoconf 機能を有効にします。
ステップ 8	end 例： Device(config)# end	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードを開始します。
ステップ 9	show template interface binding all 例： Device# show template interface binding all	テンプレートがインターフェイスに適用されているかどうかを表示します。

エンドデバイスのデバイス分類の確認

インターフェイスのインターフェイス テンプレートの確認

次の例は、IP カメラとアクセスポイントが正しい属性を持つデバイス分類子によって分類されていることを示しています。

```
Device# show device classifier attached detail
```

```
DC default profile file version supported = 1
```

```
Detail:
```

MAC_Address	Port_Id	Cert	Parent	Proto		ProfileType	Profile Name
Device_Name							
001d.a1ef.23a8	Gi1/0/7	30	3	C	M	Default	Cisco-AIR-AP-1130
cisco AIR-AP1131AG-A-K9							
001e.7a26.eb05	Gi1/0/30	70	2	C	M	Default	Cisco-IP-Camera
Cisco IP Camera							

次の例は、組み込みインターフェイステンプレートがインターフェイスに適用されることを示しています。

```
Device# show template interface binding all
```

Template-Name	Source	Method	Interface
-----	-----	-----	-----
IP_CAMERA_INTERFACE_TEMPLATE	Built-in	dynamic	Gi1/0/30
AP_INTERFACE_TEMPLATE	Modified-Built-in	dynamic	Gi1/0/7

ASP から Autoconf への移行

始める前に

show running-config | include macro auto global コマンドを使用して、AutoSmart ポート (ASP) マクロが実行されていることを確認します。

手順の概要

1. **enable**
2. **configure terminal**
3. **no macro auto global processing**
4. **exit**
5. **clear macro auto configuration all**
6. **configure terminal**
7. **autoconf enable**
8. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	no macro auto global processing 例： Device(config)# no macro auto global processing	グローバルレベルで ASP を無効にします。
ステップ 4	exit 例： Device(config)# exit	グローバル 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 5	clear macro auto configuration all 例： Device# clear macro auto configuration all	すべてのインターフェイスのマクロ設定をクリアします。
ステップ 6	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 7	autoconf enable 例： Device(config)# autoconf enable	Autoconf 機能を有効にします。
ステップ 8	end 例： Device(config)# end	グローバル 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。

Autoconf の設定例

例：エンドデバイスへの組み込みテンプレートの適用

次に、インターフェイスに接続されたエンドデバイスに組み込みテンプレートを適用する例を示します。

例：エンドデバイスへの変更された組み込みテンプレートの適用

```
Device> enable
Device(config)# configure terminal
Device(config)# autoconf enable
Device(config)# end
Device# show device classifier attached interface Gi3/0/26
Device# show template binding target GigabitEthernet 3/0/26
```

例：エンドデバイスへの変更された組み込みテンプレートの適用

次の例は、組み込みテンプレートを変更して構成を確認する方法を示しています。

```
Device> enable
Device(config)# configure terminal
Device(config)# template AP_INTERFACE_TEMPLATE
Device(config-template)# switchport access vlan 20
Device(config-template)# description modifiedAP
Device(config-template)# exit
Device(config)# autoconf enable
Device(config)# end
Device# show template interface binding all
```

例：ASP マクロから Autoconf への移行

次に、ASP から Autoconf に移行する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# no macro auto global processing
Device(config)# exit
Device# clear macro auto configuration all
Device# configure terminal
Device(config)# autoconf enable
Device(config)# end
```

Autoconf のその他の参考資料

関連資料

関連項目	マニュアル タイトル
Cisco identity-based networking services コマンド	『Cisco IOS Identity-Based Networking Services Command Reference』
インターフェイス テンプレート	『Identity-Based Networking Services Configuration Guide』の「Interface Templates」の章。

標準および RFC

標準/RFC	タイトル
IEEE 802.1X	「 <i>Port Based Network Access Control</i> 」

シスコのテクニカル サポート

説明	リンク
<p>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。</p> <p>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、Cisco Notification Service (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。</p> <p>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。</p>	http://www.cisco.com/cisco/web/support/index.html

Autoconf の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.6.1	[Autoconf]	この機能が導入されました。 Autoconf 機能では、エンドデバイスとインターフェイス間のハードバインディングが可能です。 autoconf enable 、 map attribute-to-service (autoconf) 、 map device-type (service-template) 、 parameter-map type subscriber (service-template) 、 show parameter-map type subscriber attribute-to-service all 、 show template interface の各コマンドが追加または変更されました。
Cisco IOS XE Gibraltar 16.12.1	AutoConfによるデバイス設定をシスコスイッチのPIDまで細分化	クラスマップとパラメータマップの設定用に platform type フィルタオプションが追加されました。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> [英語] からアクセスします。



第 2 章

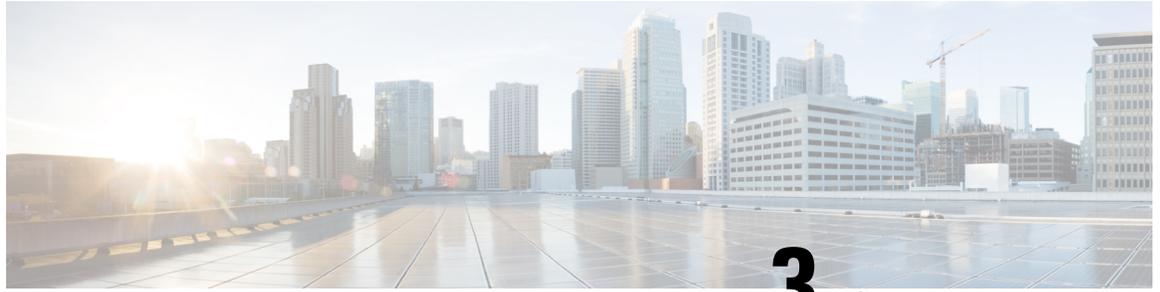
Cisco プラグ アンド プレイの設定

- [Cisco プラグ アンド プレイの設定 \(19 ページ\)](#)

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](#)



第 3 章

Cisco Discovery Protocol の設定

Cisco Discovery Protocol は、シスコデバイス上で動作し、ネットワーキングアプリケーションが直接接続された付近のデバイスに関して学習できるようにする、メディア独立型かつネットワーク独立型のレイヤ2プロトコルです。このプロトコルによってシスコデバイスが検出されてその設定状態が特定され、異なるネットワーク層プロトコルを使用するシステムが相互に学習できるようになることで、デバイスの管理が容易になります。

このモジュールでは、Cisco Discovery Protocol バージョン 2 とその SNMP での動作について説明します。

- [Cisco Discovery Protocol について \(21 ページ\)](#)
- [Cisco Discovery Protocol の設定方法 \(22 ページ\)](#)
- [Cisco Discovery Protocol のモニタリングとメンテナンス \(30 ページ\)](#)

Cisco Discovery Protocol について

Cisco Discovery Protocol の概要

Cisco Discovery Protocol 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、既知のデバイスのネイバーであるシスコデバイスを検出することができます。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、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。

Cisco Discovery Protocol 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。Cisco Discovery Protocol 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。

Cisco Discovery Protocol 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 Cisco Discovery Protocol 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。

Cisco Discovery Protocol はdevice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。deviceは Cisco Discovery Protocol 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device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。

- Cisco Discovery Protocol は、deviceと直接通信する接続されたエンドポイントを識別します。
- ネイバー デバイスのレポートが重複しないように、1つの有線deviceだけがロケーション情報をレポートします。
- 有線deviceとエンドポイントは、ロケーションの送信と受信の両方を行います。

Cisco Discovery Protocol のデフォルト設定

次の表に、Cisco Discovery Protocol のデフォルト設定を示します。

機能	デフォルト設定
Cisco Discovery Protocol グローバル状態	イネーブル
Cisco Discovery Protocol インターフェイス状態	イネーブル
Cisco Discovery Protocol タイマー（パケット更新頻度）	60 秒
Cisco Discovery Protocol 保持時間（廃棄前）	180 秒
Cisco Discovery Protocol バージョン2 アドバタイズメント	有効

Cisco Discovery Protocol の設定方法

Cisco Discovery Protocol の特性の設定

次の Cisco Discovery Protocol の特性を設定できます。

- Cisco Discovery Protocol アップデートの頻度
- 破棄するまで情報を保持する時間の長さ
- バージョン2 アドバタイズメント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。

次の手順に従って、Cisco Discovery Protocol の特性を設定します。

手順の概要

1. **enable**
2. **configure terminal**
3. **cdp timer seconds**
4. **cdp holdtime seconds**
5. **cdp advertise-v2**
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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 timer seconds 例： デバイス(config)# cdp timer 20	(任意) Cisco Discovery Protocol 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 seconds 例： デバイス(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。 指定できる範囲は 10～255 秒です。デフォルトは 180 秒です。
ステップ 5	cdp advertise-v2 例： デバイス(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 Cisco Discovery Protocol を設定します。 これは、デフォルトの状態です。

	コマンドまたはアクション	目的
ステップ 6	end 例： デバイス(config)# end	特権 EXEC モードに戻ります。
ステップ 7	show running-config 例： デバイス# show running-config	入力を確認します。
ステップ 8	copy running-config startup-config 例： デバイス# 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、Cisco Discovery Protocol コマンドの **no** 形式を使用します。

Cisco Discovery Protocol のディセーブル化

Cisco Discovery Protocol は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、Cisco Discovery Protocol メッセージを定期的に交換します。Cisco Discovery Protocol 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。

Cisco Discovery Protocol 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。

手順の概要

1. **enable**
2. **configure terminal**
3. **no cdp run**
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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： デバイス(config)# no cdp run	Cisco Discovery Protocol を無効にします。
ステップ 4	end 例： デバイス(config)# end	特権 EXEC モードに戻ります。
ステップ 5	show running-config 例： デバイス# show running-config	入力を確認します。
ステップ 6	copy running-config startup-config 例： デバイス# 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

Cisco Discovery Protocol を使用するには、再度有効にする必要があります。

Cisco Discovery Protocol の有効化

Cisco Discovery Protocol は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、Cisco Discovery Protocol メッセージを定期的に交換します。Cisco Discovery Protocol 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 Cisco Discovery Protocol 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

Cisco Discovery Protocol がディセーブルになっていないと、イネーブルにはできません。

手順の概要

1. **enable**
2. **configure terminal**
3. **cdp run**
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します (要求された場合)。
ステップ 2	configure terminal 例： デバイス# 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： デバイス (config)# cdp run	Cisco Discovery Protocol がディセーブルになっている場合にイネーブルにします。
ステップ 4	end 例： デバイス (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 : デバイス# show running-config	入力を確認します。
ステップ 6	copy running-config startup-config 例 : デバイス# 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

Cisco Discovery Protocol がイネーブルになっていることを表示するには、**show run all** コマンドを使用します。**show run** を入力しただけでは、Cisco Discovery Protocol がイネーブルになっていることが表示されない場合があります。

インターフェイス上で Cisco Discovery Protocol をディセーブルにします。

Cisco Discovery Protocol は、Cisco Discovery Protocol 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で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、Cisco Discovery Protocol メッセージを定期的に交換します。Cisco Discovery Protocol 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。



- (注) Discovery Protocol バイパスはサポートされていないため、ポートが err-disabled ステートになる場合があります。

ポートで Cisco Discovery Protocol をディセーブルにするには、次の手順に従います。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface interface-id**
4. **no cdp enable**
5. **end**

■ インターフェイス上で Cisco Discovery Protocol をディセーブルにします。

6. `show running-config`
7. `copy running-config startup-config`

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> <code>enable</code>	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# <code>configure terminal</code>	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： デバイス(config)# <code>interface gigabitethernet 1/0/1</code>	Cisco Discovery Protocol 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： デバイス(config-if)# <code>no cdp enable</code>	ステップ 3 で指定したインターフェイス上で Cisco Discovery Protocol をディセーブルにします。
ステップ 5	end 例： デバイス(config)# <code>end</code>	特権 EXEC モードに戻ります。
ステップ 6	show running-config 例： デバイス# <code>show running-config</code>	入力を確認します。
ステップ 7	copy running-config startup-config 例： デバイス# <code>copy running-config startup-config</code>	(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 Cisco Discovery Protocol のイネーブル化

Cisco Discovery Protocol は、Cisco Discovery Protocol 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で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、Cisco Discovery Protocol メッセージを定期的に交換します。Cisco Discovery Protocol 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。



- (注) Discovery Protocol バイパスはサポートされていないため、ポートが `err-disabled` ステートになる場合があります。

ポートでディセーブルになっている Cisco Discovery Protocol 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

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

手順の概要

1. `enable`
2. `configure terminal`
3. `interface interface-id`
4. `cdp enable`
5. `end`
6. `show running-config`
7. `copy running-config startup-config`

手順の詳細

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

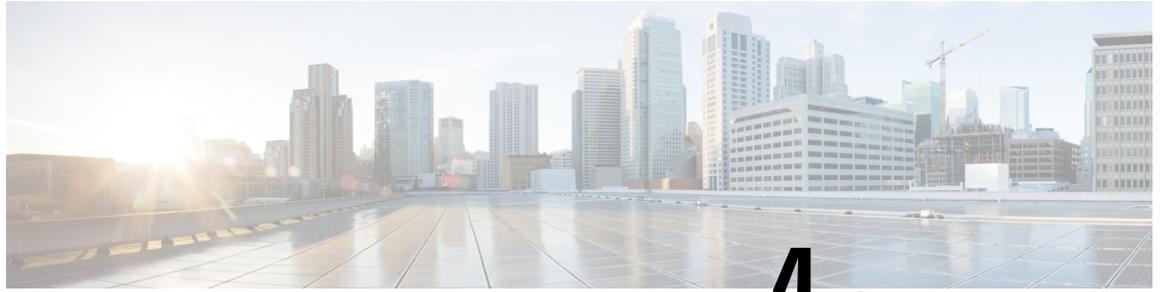
	コマンドまたはアクション	目的
	デバイス# <code>configure terminal</code>	
ステップ 3	<code>interface interface-id</code> 例： デバイス(config)# <code>interface gigabitethernet1/0/1</code>	Cisco Discovery Protocol をイネーブルにするインターフェイスを指定し、インターフェイスコンフィギュレーションモードを開始します。
ステップ 4	<code>cdp enable</code> 例： デバイス(config-if)# <code>cdp enable</code>	ディセーブルになっているインターフェイスで Cisco Discovery Protocol をイネーブルにします。
ステップ 5	<code>end</code> 例： デバイス(config)# <code>end</code>	特権 EXEC モードに戻ります。
ステップ 6	<code>show running-config</code> 例： デバイス# <code>show running-config</code>	入力を確認します。
ステップ 7	<code>copy running-config startup-config</code> 例： デバイス# <code>copy running-config startup-config</code>	(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。

Cisco Discovery Protocol のモニタリングとメンテナンス

表 1: Cisco Discovery Protocol 情報を表示するためのコマンド

コマンド	説明
<code>clear cdp counters</code>	トラフィックカウンタを 0 にリセットします。
<code>clear cdp table</code>	ネイバーに関する情報の Cisco Discovery Protocol テーブルをクリアします。
<code>show cdp</code>	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。

コマンド	説明
show cdp entry <i>entry-name</i> [version] [protocol]	<p>特定のネイバーに関する情報を表示します。</p> <p>アスタリスク (*) を入力して、すべての Cisco Discovery Protocol 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指定することもできます。</p> <p>また、指定されたネイバー上でイネーブルになっているかどうかの情報を、デバイス上で稼働しているソフトウェアのバージョンや、表示されるように、表示内容を制限することもできます。</p>
show cdp interface [<i>interface-id</i>]	<p>Cisco Discovery Protocol がイネーブルになっているインターフェイスに関する情報を表示します。</p> <p>必要なインターフェイスの情報だけを表示できます。</p>
show cdp neighbors [<i>interface-id</i>] [<i>detail</i>]	<p>装置タイプ、インターフェイスタイプ、インターフェイスのアップタイムの設定値、機能、プラットフォーム、ポート ID を表示します。</p> <p>特定のインターフェイスに関するネイバー情報だけを詳細表示にするため表示内容を拡張したりできます。</p>
show cdp traffic	<p>Cisco Discovery Protocol カウンタ（送信済み/受信済みパケット数 エラー数を含む）を表示します。</p>



第 4 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](#) (33 ページ)
- [SNMP の制約事項](#) (35 ページ)
- [SNMP に関する情報](#) (36 ページ)
- [SNMP の設定方法](#) (40 ページ)
- [SNMP ステータスのモニタリング](#) (55 ページ)
- [SNMP の例](#) (56 ページ)
- [簡易ネットワーク管理プロトコルの機能の履歴と情報](#) (57 ページ)

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	レベル	認証	暗号化	結果
SNMPv1	noAuthNoPriv	コミュニティストリング	未対応	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	未対応	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	Username	未対応	ユーザ名の照合を使用して認証します。

モデル	レベル	認証	暗号化	結果
SNMPv3	authNoPriv	Message Digest 5 (MD5) または Secure Hash Algorithm (SHA)	未対応	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)	<p>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。</p> <p>次の暗号化アルゴリズムで、User-based Security Model (USM) を指定できます。</p> <ul style="list-style-type: none"> • 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せん。

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。

SNMPv3 認証は、次のシナリオではサポートされません。

- スイッチ優先順位の変更後にスタックリロードが発生した場合。
- 低い MAC アドレスを持つデバイスがスタックに追加された場合、スタック内のすべてのスイッチの優先順位が同じであれば、そのデバイスがアクティブスイッチとして選択されます。

SNMPv3 認証の失敗を回避するには、SNMPv3 ユーザーを設定する前に、デバイスで SNMP engineID を手動で設定する必要があります。これにより、ユーザーは engineID に関連付けられているためデバイスを管理できます。

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、device に常駐します。device 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

動作	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

動作	説明
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。
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。
trap	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 この動作では、SNMP マネージャに正確な変数名を認識させ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、スパン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tring

SNMP コミュニティ スtring は、MIB オブジェクトへのアクセスを認証し、組み込みパスワードとして機能します。NMS が device にアクセスするには、NMS 上のコミュニティ スtring 定義が device 上の 3 つのコミュニティ スtring 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tring 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tring 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tring へのアクセスは許可しません。

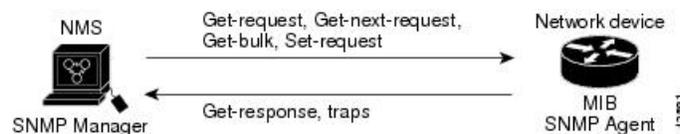
- クラスタを作成すると、コマンド `device` がメンバ `devices` と SNMP アプリケーション間のメッセージ交換を管理します。Network Assistant ソフトウェアは、コマンド `device` 上で最初に設定された RW および RO コミュニティ スtring にメンバ `device` 番号 (@esN、N は `device` 番号) を追加し、これらの String をメンバ `devices` に伝播します。

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、`device` 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、`device` 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送信側にわかりません。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、device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device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	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、device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

device が起動し、device 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 エージェントは有効になります。

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には、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、そのユーザに関連付けられてい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、 **device** 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tring 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tring 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tring を使用します。コミュニティ スtring は、device 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 String 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

device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。

手順の概要

1. **enable**
2. **configure terminal**
3. **snmp-server community string [view view-name] [ro | rw] [access-list-number]**
4. **access-list access-list-number {deny | permit} source [source-wildcard]**
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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： デバイス(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。 • <i>string</i> 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。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • (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) <i>access-list-number</i> 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 <i>access-list-number</i> { deny permit } <i>source</i> [<i>source-wildcard</i>] 例 : デバイス (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。 <ul style="list-style-type: none"> • <i>access-list-number</i> 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。 • <i>source</i> 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) <i>source-wildcard</i> には、<i>source</i> 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 : デバイス (config) # end	特権 EXEC モードに戻ります。
ステップ 6	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
	デバイス# <code>show running-config</code>	
ステップ 7	copy running-config startup-config 例 : デバイス# <code>copy running-config startup-config</code>	(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tring をヌル スtring に設定します (コミュニティ スtring に値を入力しないでください)。

特定のコミュニティ スtring を削除するには、**no snmp-server** 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。

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

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

device 上の SNMP グループとユーザーを設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **snmp-server engineID** {local *engineid-string* | remote *ip-address* [**udp-port** *port-number*]} *engineid-string*
4. **snmp-server group** *group-name* {v1 | v2c | v3 {**auth** | **noauth** | **priv**}} [**read** *readview*] [**write** *writeview*] [**notify** *notifyview*] [**access** *access-list*]
5. **snmp-server user** *username* *group-name* {remote *host* [**udp-port** *port*]} {v1 [**access** *access-list*] | v2c [**access** *access-list*] | v3 [**encrypted**] [**access** *access-list*] [**auth** {**md5** | **sha**} *auth-password*]} [**priv** {**des** | **3des** | **aes** {**128** | **192** | **256**}} *priv-password*]
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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 {local engineid-string remote ip-address [udp-port port-number] engineid-string} 例： デバイス(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。 <ul style="list-style-type: none"> 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 {v1 v2c v3 {auth noauth priv}} [read readview] [write writeview] [notify notifyview] [access access-list] 例： デバイス(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。 <ul style="list-style-type: none"> 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送信できます。 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。

	コマンドまたはアクション	目的
		<p>auth : Message Digest 5 (MD5) およびセキュアハッシュアルゴリズム (SHA) によるパケット認証を可能にします。</p> <p>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。</p> <p>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。</p> <p>(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名を表す文字列 (64 文字以内) を入力します。</p> <p>(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表示できるビュー名を表す文字列 (64 文字以内) を入力します。</p> <p>(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名を表す文字列 (64 文字以内) を入力します。</p> <p>(任意) access access-list とともに、アクセスリスト名の文字列 (64 文字以内) を入力します。</p>
<p>ステップ 5</p>	<p>snmp-server user username group-name { remote host [udp-port port] } { v1 [access access-list] v2c [access access-list] v3 [encrypted] [access access-list] [auth { md5 sha } auth-password] } [priv { des 3des aes { 128 192 256 } } priv-password]</p> <p>例 :</p> <pre>デバイス(config)# snmp-server user Pat public v2c</pre>	<p>SNMP グループに対して新規ユーザを追加します。</p> <p><i>username</i> は、エージェントに接続するホスト上のユーザ名です。</p> <p><i>group-name</i> は、ユーザが関連付けられているグループの名前です。</p> <p>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。</p> <p>SNMP バージョン番号 (v1、v2c、または v3) を入力します。v3 を入力する場合は、次のオプションを追加します。</p> <ul style="list-style-type: none"> • encrypted は、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にのみ使用できます。

	コマンドまたはアクション	目的
		<p>• 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) 認証レベルを指定できます。また、<i>auth-password</i> 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)。</p> <p>v3 を入力すると、次のキーワードを使用して (64 文字以内)、プライベート (priv) 暗号化アルゴリズムおよびパスワード文字列 <i>priv-password</i> を設定することもできます。</p> <ul style="list-style-type: none"> • priv は、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。 <p>(任意) access <i>access-list</i> とともに、アクセスリスト名の文字列 (64 文字以内) を入力します。</p>
ステップ 6	<p>end</p> <p>例 :</p> <p>デバイス (config) # end</p>	特権 EXEC モードに戻ります。
ステップ 7	<p>show running-config</p> <p>例 :</p> <p>デバイス # show running-config</p>	入力を確認します。
ステップ 8	<p>copy running-config startup-config</p> <p>例 :</p> <p>デバイス # copy running-config startup-config</p>	(任意) コンフィギュレーションファイルに設定を保存します。

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device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Devices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deviceを設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **snmp-server engineID remote ip-address engineid-string**
4. **snmp-server user username group-name {remote host [udp-port port]} {v1 [access access-list] | v2c [access access-list] | v3 [encrypted] [access access-list] [auth {md5 | sha} auth-password] }**
5. **snmp-server group group-name {v1 | v2c | v3 {auth | noauth | priv}} [read readview] [write writeview] [notify notifyview] [access access-list]**
6. **snmp-server host host-addr [informs | traps] [version {1 | 2c | 3 {auth | noauth | priv}}] community-string [notification-type]**
7. **snmp-server enable traps notification-types**
8. **snmp-server trap-source interface-id**
9. **snmp-server queue-length length**
10. **snmp-server trap-timeout seconds**
11. **end**
12. **show running-config**
13. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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： デバイス (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 {remote host [udp-port port]} {v1 [access access-list] v2c [access access-list] v3 [encrypted] [access access-list] [auth {md5 sha} auth-password]} 例： デバイス (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： デバイス (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： デバイス (config)# snmp-server host 203.0.113.1 comaccess snmp	SNMP トラップ動作の受信先を指定します。 <i>host-addr</i> 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。 (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。

	コマンドまたはアクション	目的
		<p>(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。</p> <p>(任意) バージョン3 の場合、認証レベルとして auth、noauth、または priv を選択します。</p> <p>(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指定できます。</p> <p><i>community-string</i> 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は、SNMPv3 のユーザ名を入力します。</p> <p>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。</p> <p>(任意) <i>notification-type</i> 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。</p>
<p>ステップ 7</p>	<p>snmp-server enable traps notification-types</p> <p>例 :</p> <pre>デバイス(config)# snmp-server enable traps snmp</pre>	<p>deviceでのトラップまたはインフォームの送信をイネーブルにし、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、と入力してください。 snmp-server enable traps ?</p> <p>複数のトラップタイプをイネーブルにするには、トラップタイプごとに snmp-server enable traps コマンドを個別に入力する必要があります。</p> <p>(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。</p> <ol style="list-style-type: none"> snmp-server enable traps port-security snmp-server enable traps port-security trap-rate rate

	コマンドまたはアクション	目的
ステップ 8	snmp-server trap-source interface-id 例： デバイス (config) # snmp-server trap-source gigabitethernet 1/0/1	(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： デバイス (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。指定できる値の範囲は 1～5000 です。デフォルトは 10 です。
ステップ 10	snmp-server trap-timeout seconds 例： デバイス (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： デバイス (config) # end	特権 EXEC モードに戻ります。
ステップ 12	show running-config 例： デバイス # show running-config	入力を確認します。
ステップ 13	copy running-config startup-config 例： デバイス # 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）がグローバルにイネーブルになります。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イネーブル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ディセーブルになりますが、情報はディセーブルになりません。情報をディセーブルにするには、**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ます。特定のトラップタイプをディセーブル

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。

手順の概要

1. **enable**
2. **configure terminal**
3. **snmp-server contact text**
4. **snmp-server location text**
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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 text 例： デバイス(config)# snmp-server contact Dial System Operator at beeper 21555	システムの連絡先文字列を設定します。
ステップ 4	snmp-server location text 例： デバイス(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：	特権 EXEC モードに戻ります。

SNMP を通して使用する TFTP サーバの制限

	コマンドまたはアクション	目的
	デバイス(config)# end	
ステップ 6	show running-config 例： デバイス# show running-config	入力を確認します。
ステップ 7	copy running-config startup-config 例： デバイス# 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。

手順の概要

1. **enable**
2. **configure terminal**
3. **snmp-server tftp-server-list access-list-number**
4. **access-list access-list-number {deny | permit} source [source-wildcard]**
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します (要求された場合)。
ステップ 2	configure terminal 例： デバイス# 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： デバイス(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 {deny permit} source [source-wildcard] 例： デバイス(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、 device 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： デバイス(config)# end	特権 EXEC モードに戻ります。
ステップ 6	show running-config 例： デバイス# show running-config	入力を確認します。
ステップ 7	copy running-config startup-config 例： デバイス# 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 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。

手順の概要

1. **enable**
2. **configure terminal**
3. **no snmp-server**
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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： デバイス(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：	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
	デバイス(config)# end	
ステップ 5	show running-config 例： デバイス# show running-config	入力を確認します。
ステップ 6	copy running-config startup-config 例： デバイス# 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	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 モードで実行される場合にのみ、設定情報を表示する際に使用する必要があります。また、 show running-config の出力には表示されません。

SNMP の例

次に、SNMP の全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、*device* はトラップを送信しません。

```
デバイス(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。*device* 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。

```
デバイス(config)# snmp-server community public
デバイス(config)# snmp-server enable traps vtp
デバイス(config)# snmp-server host 192.180.1.27 version 2c public
デバイス(config)# snmp-server host 192.180.1.111 version 1 public
デバイス(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。

```
デバイス(config)# snmp-server community comaccess ro 4
デバイス(config)# snmp-server enable traps snmp authentication
デバイス(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 行目で、*device* 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。

```
デバイス(config)# snmp-server enable traps entity
デバイス(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 *device* をイネーブルにする例を示します。

```
デバイス(config)# snmp-server enable traps
デバイス(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際に *auth* (*authNoPriv*) 認証レベルで情報を送信する例を示します。

```
デバイス(config)# snmp-server engineID remote 192.180.1.27 00000063000100a1c0b4011b
デバイス(config)# snmp-server group authgroup v3 auth
デバイス(config)# snmp-server user authuser authgroup remote 192.180.1.27 v3 auth md5 mypassword
デバイス(config)# snmp-server user authuser authgroup v3 auth md5 mypassword
デバイス(config)# snmp-server host 192.180.1.27 informs version 3 auth authuser config
```

```
デバイス(config)# snmp-server enable traps  
デバイス(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	変更内容
Cisco IOS XE Everest 16.5.1a	この機能が導入されました。



第 5 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 \(59 ページ\)](#)
- [SLA について \(60 ページ\)](#)
- [IP SLA 動作の設定方法 \(65 ページ\)](#)
- [IP SLA 動作のモニタリング \(79 ページ\)](#)
- [IP SLA 動作のモニタリングの例 \(80 ページ\)](#)
- [その他の参考資料 \(81 ページ\)](#)

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 of 制約事項です。

- `device` 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 サーバーのようリモート 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、ユーザー 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ー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)
- ジッター (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)
- サーバーまたは 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。

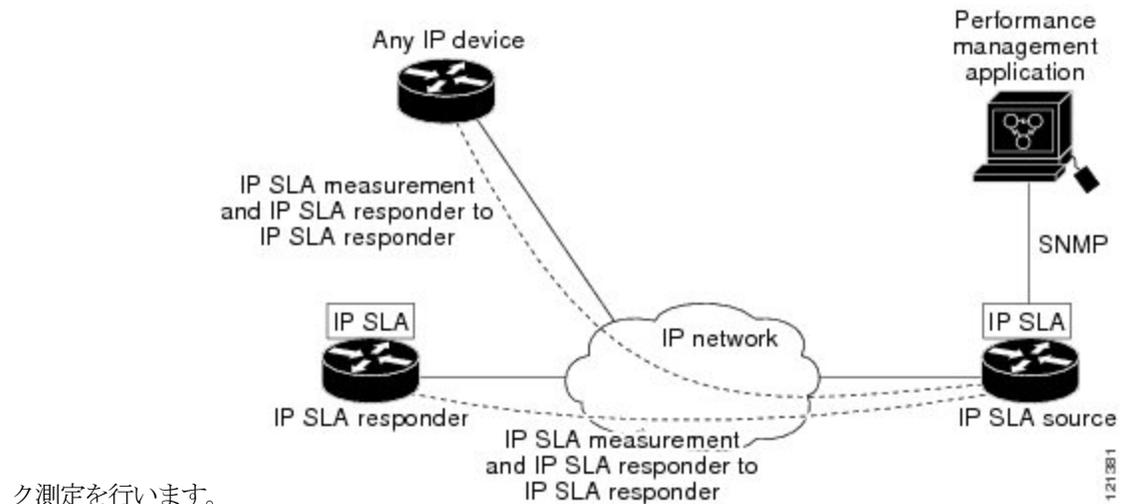
- 連続的で信頼性のある予測可能な測定。
- IP サービス ネットワーク ヘルス アセスメントにより、既存の QoS が新しい IP サービスに適していることを確認できる。
- 端末間のネットワーク アベイラビリティをモニタリングして、ネットワーク リソースをあらかじめ検証し接続をテストできる（たとえば、ビジネス上の重要なデータを保存する NFS サーバーのネットワーク アベイラビリティをリモート サイトから確認できる）。
- 問題をすぐに認識し、トラブルシューティングにかかる時間を短縮できる一貫性のある信頼性の高い測定によるネットワーク動作のトラブルシューティング。
- マルチプロトコル 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 (device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

IPSLA 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監視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



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。

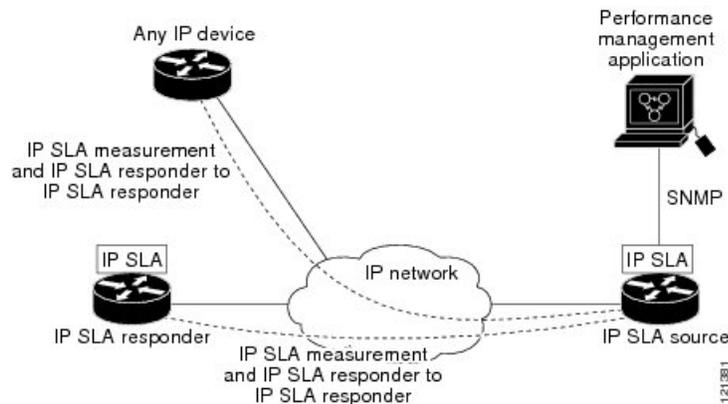
レスポンドは、受信および応答するポートが通知されるメカニズムを Cisco IOS IP SLA コントロール プロトコルを通じて実現します。



- (注) IP SLA レスポンドはレスポンド設定可能なdeviceである Cisco IOS レイヤ 2 にすることもできます。レスポンドは、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 レスポンドの配置場所を示します。レスポンドは、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有効にします。この間に、レスポンドは要求を受け付け、応答します。レスポンドは、IP SLA パケットに回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ドをイネーブ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。

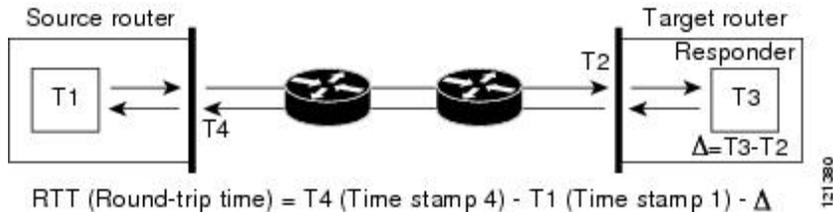
IP SLA の応答時間の計算

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 (レスポンド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 レスポンドが有効の場合、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

次の図に、レスポнда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нда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方向遅延、ジッター、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。一方方向ジッター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

- タイムアウト
- RTT しきい値
- 平均ジッターしきい値
- 一方向パケット損失
- 一方向ジッター
- 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 エコー

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、正のジッターは望ましくありません。ジッター値 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 ジッター動作では合成（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。一方向ジッター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 device でサポートされないものもあります。device 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。

```

デバイス# 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 devices 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。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip sla responder {tcp-connect | udp-echo} ipaddress ip-address port port-number**
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	configure terminal 例 : デバイス# <code>configure terminal</code>	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 : デバイス(config)# <code>ip sla responder udp-echo 172.29.139.134 5000</code>	device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。 <ul style="list-style-type: none"> • tcp-connect : Responder の TCP 接続動作をイネーブルにします。 • udp-echo : レスポンダのユーザーデータグラム プロトコル (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、IPSLA 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する必要があります。
ステップ 4	end 例 : デバイス(config)# <code>end</code>	特権 EXEC モードに戻ります。
ステップ 5	show running-config 例 : デバイス# <code>show running-config</code>	入力を確認します。
ステップ 6	copy running-config startup-config 例 : デバイス# <code>copy running-config startup-config</code>	(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。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip sla operation-number**
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*]
5. **frequency** *seconds*
6. **threshold** *milliseconds*
7. **exit**
8. **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**]
9. **end**
10. **show running-config**
11. **copy running-config startup-config**

手順の詳細

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

	コマンドまたはアクション	目的
ステップ 4	<p>udp-jitter {<i>destination-ip-address</i> <i>destination-hostname</i>} <i>destination-port</i> [source-ip {<i>ip-address</i> <i>hostname</i>}] [source-port <i>port-number</i>] [control {enable disable}] [num-packets <i>number-of-packets</i>] [interval <i>interpacket-interval</i>]</p> <p>例 :</p> <pre>Device (config-ip-sla)# udp-jitter 172.29.139.134 5000 source-ip 172.29.139.140 source-port 4000</pre>	<p>IPSLA 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) 。</p> <ul style="list-style-type: none"> • <i>destination-ip-address</i> <i>destination-hostname</i> : 宛先 IP アドレスまたはホスト名を指定します。 • <i>destination-port</i> : 宛先ポート番号を 1 ~ 65535 の範囲で指定します。 • (任意) source-ip {<i>ip-address</i> <i>hostname</i>} : 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 <i>port-number</i> : 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。 <ul style="list-style-type: none"> (注) udp-jitter コマンドで送信元ポートが設定されていない場合、UDP は制御パケット用のポートをランダムに選択します。UDP が予約済みポート 1967 を選択した場合、IP SLA レスポンダによる CPU 使用率が高くなる可能性があります。 • (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有効または無効にします。デフォルトでは、IPSLA 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 <i>number-of-packets</i> : 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 <i>inter-packet-interval</i> : パケットの送信間隔をミリ秒単位で入力します。指定できる範囲は 1 ~ 6000 です。デフォルトは 20 ミリ秒です。

	コマンドまたはアクション	目的
ステップ 5	frequency <i>seconds</i> 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 <i>milliseconds</i> 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 <i>operation-number</i> [life { forever <i>seconds</i> }] [start-time { <i>hh:mm</i> [: <i>ss</i>] [<i>month day</i> <i>day month</i>] pending now after <i>hh:mm:ss</i>] [ageout <i>seconds</i>] [recurring] 例 : Device(config)# ip sla schedule 10 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。 <ul style="list-style-type: none"> • <i>operation-number</i> : 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) に設定するか、特定の秒数 (<i>seconds</i>) を設定します。有効な範囲は 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 と入力すれば、開始時刻を指定するまでは情報を収集しません。 now と入力すれば、ただちに動作を開始します。 after <i>hh:mm:ss</i> 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 <i>seconds</i> : 情報を収集していないときに、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

	コマンドまたはアクション	目的
		<p>す。デフォルトは0秒（いつまでも保存する）です。</p> <ul style="list-style-type: none"> （任意） recurring : 毎日、動作を自動的に実行するように設定します。
ステップ 9	end 例 : Device(config)# end	特権 EXEC モードに戻ります。
ステップ 10	show running-config 例 : Device# show running-config	入力を確認します。
ステップ 11	copy running-config startup-config 例 : Device# copy running-config startup-config	（任意） コンフィギュレーション ファイルに設定を保存します。

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 source-ip 172.29.139.140 source-port
4000
Device(config-ip-sla-jitter)# frequency 30
Device(config-ip-sla-jitter)# exit
Device(config)# ip sla schedule 10 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 ジッター動作を設定するには、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。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip sla operation-number**
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*]
5. **frequency** *seconds*
6. **exit**
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**]
8. **end**
9. **show running-config**
10. **copy running-config startup-config**

手順の詳細

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

	コマンドまたはアクション	目的
	Device# configure terminal	
ステップ 3	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 { <i>destination-ip-address</i> <i>destination-hostname</i> } <i>destination-port</i> [source-ip { <i>ip-address</i> <i>hostname</i> }] [source-port <i>port-number</i>] [control { enable disable }] [num-packets <i>number-of-packets</i>] [interval <i>interpacket-interval</i>] 例 : Device(config-ip-sla)# udp-jitter 172.29.139.134 5000 source-ip 172.29.139.140 source-port 4000	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。 <ul style="list-style-type: none"> • <i>destination-ip-address</i> <i>destination-hostname</i> : 宛先 IP アドレスまたはホスト名を指定します。 • <i>destination-port</i> : 宛先ポート番号を 1 ~ 65535 の範囲で指定します。 • (任意) source-ip {<i>ip-address</i> <i>hostname</i>} : 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 <i>port-number</i> : 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。 (注) udp-jitter コマンドで送信元ポートが設定されていない場合、UDP は制御パケット用のポートをランダムに選択します。UDP が予約済みポート 1967 を選択した場合、IP SLA レスポンダによる CPU 使用率が高くなる可能性があります。 • (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有効または無効にします。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 <i>number-of-packets</i> : 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。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • (任意) interval <i>inter-packet-interval</i> : パケットの送信間隔をミリ秒単位で入力します。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 <i>seconds</i> 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 <i>operation-number</i> [life { forever <i>seconds</i> }] [start-time { <i>hh:mm[:ss]</i> [<i>month day</i> <i>day month</i>] pending now after <i>hh:mm:ss</i>] [ageout <i>seconds</i>] [recurring] 例 : Device(config) # ip sla schedule 10 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。 <ul style="list-style-type: none"> • 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) に設定するか、特定の秒数 (<i>seconds</i>) を設定します。有効な範囲は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 と入力すれば、開始時刻を指定するまでは情報を収集しません。 now と入力すれば、ただちに動作を開始します。 after <i>hh:mm:ss</i> 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 <i>seconds</i> : 情報を収集していないときに、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
		<ul style="list-style-type: none"> • (任意) recurring : 毎日、動作を自動的に実行するように設定します。
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。

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 source-ip 172.29.139.140 source-port
4000
Device(config-ip-sla-jitter)# frequency 30
Device(config-ip-sla-jitter)# exit
Device(config)# ip sla schedule 10 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 レスポンド側を有効にしておく必要はありません。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip sla operation-number**
4. **icmp-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*} | **source-interface** *interface-id*]
5. **frequency** *seconds*
6. **exit**
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**]
8. **end**
9. **show running-config**
10. **copy running-config startup-config**

手順の詳細

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

	コマンドまたはアクション	目的
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 { <i>destination-ip-address</i> <i>destination-hostname</i> } [source-ip { <i>ip-address</i> <i>hostname</i> } source-interface <i>interface-id</i>] 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。 <ul style="list-style-type: none"> • <i>destination-ip-address</i> <i>destination-hostname</i> : 宛先 IP アドレスまたはホスト名を指定します。 • (任意) source-ip {<i>ip-address</i> <i>hostname</i>} : 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 <i>interface-id</i> : 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 seconds 例 : Device (config-ip-sla-echo) # frequency 30	(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-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 <i>seconds</i> }] [start-time { <i>hh:mm</i> [: <i>ss</i>] [<i>month day</i> <i>day month</i>] pending now after <i>hh:mm:ss</i>] [ageout <i>seconds</i>] [recurring] 例 : Device (config) # ip sla schedule 10 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。 <ul style="list-style-type: none"> • <i>operation-number</i> : 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 (<i>seconds</i>) を指定します。有効な範囲は 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。

	コマンドまたはアクション	目的
		<p>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。</p> <p>pending と入力すれば、開始時刻を指定するまでは情報を収集しません。</p> <p>now と入力すれば、ただちに動作を開始します。</p> <p>after hh:mm:ss と入力すると、指定した時刻の経過後に動作を開始します。</p> <ul style="list-style-type: none"> （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	<p>end</p> <p>例 :</p> <pre>Device(config)# end</pre>	特権 EXEC モードに戻ります。
ステップ 9	<p>show running-config</p> <p>例 :</p> <pre>Device# show running-config</pre>	入力を確認します。
ステップ 10	<p>copy running-config startup-config</p> <p>例 :</p> <pre>Device# copy running-config startup-config</pre>	（任意）コンフィギュレーションファイルに設定を保存します。

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 12 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
show ip sla authentication	IP SLA 認証情報を表
show ip sla configuration [entry-number]	すべての IP SLA 動作 デフォルト値をすべ
show ip sla enhanced-history {collection-statistics distribution statistics} [entry-number]	収集した履歴バケッ での IP SLA 動作また 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

show ip sla group schedule [<i>schedule-entry-number</i>]	IP SLA グループ スケジ ます。
show ip sla history [<i>entry-number</i> full tabular]	すべての IP SLA 動作に
show ip sla mpls-lsp-monitor { collection-statistics configuration ldp operational-state scan-queue summary [<i>entry-number</i>] neighbors }	MPLS ラベル スイッチ を表示します。
show ip sla reaction-configuration [<i>entry-number</i>]	すべての IP SLA 動作ま 予防的しきい値のモニタ
show ip sla reaction-trigger [<i>entry-number</i>]	すべての IP SLA 動作ま 応トリガー情報を表示し
show ip sla responder	IP SLA レスポンダ側の
show ip sla statistics [<i>entry-number</i> aggregated details]	動作ステータスおよび給 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。

```

デバイス# 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。

```

デバイス# 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	マニュアルタイトル
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	タイトル
なし	-

MIB

MIB	MIB のリンク
本リリースでサポートするすべての MIB	<p>選択したプラットフォーム、Cisco IOS リリース、およびフィ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。</p> <p>http://www.cisco.com/go/mibs</p>

シスコのテクニカル サポート

説明	リンク
<p>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。</p> <p>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、Cisco Notification Service (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。</p> <p>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。</p>	<p>http://www.cisco.com/support</p>



第 6 章

SPAN および RSPAN の設定

- SPAN および RSPAN の前提条件 (83 ページ)
- SPAN および RSPAN の制約事項 (83 ページ)
- SPAN および RSPAN について (86 ページ)
- SPAN および RSPAN の設定方法 (98 ページ)
- SPAN および RSPAN 動作のモニタリング (124 ページ)
- SPAN および RSPAN の設定例 (124 ページ)
- SPAN および RSPAN の機能の履歴と情報 (127 ページ)

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 ポートをモニターしている場合、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ーされます。デフォルトでは、トランクポート上のすべての VLAN がモニターされます。

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 のトラフィックを監視できます。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。
- 無効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 が有効になってからです。
- 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 のポートをトラフィック監視するなど、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 が有効の場合、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。
- ディゼ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。
- デバイスで DHCP スヌーピングが有効になっている場合、SPAN セッションは Dynamic Host Configuration Protocol (DHCP) 入力パケットのみをキャプチャします。

RSPAN

RSPAN の制約事項は次のとおりです。

- RSPAN は、BPDU パケットモニタリングまたは他のレイヤ 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ン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グディングが防止されます。
- RSPAN VLAN をネイティブ VLAN として設定しないことをお勧めします。

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 device 上、またはネットワーク アナライザやその他のモニター 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 device 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 を使用してモニターできるのは、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ーできません。たとえば、着信トラフィックをモニターしている場合、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ーできません。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、モニターできます。

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 device 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 device または device 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。

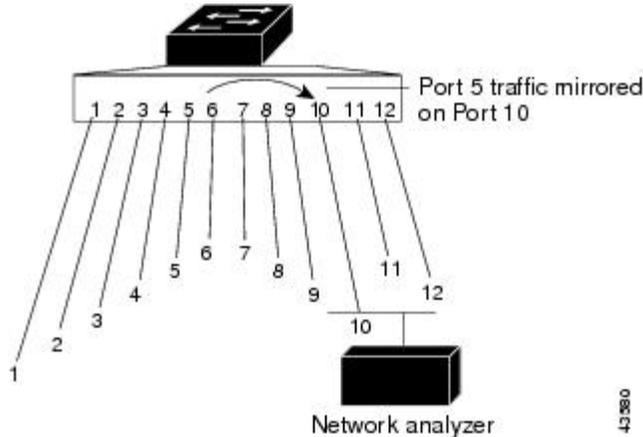
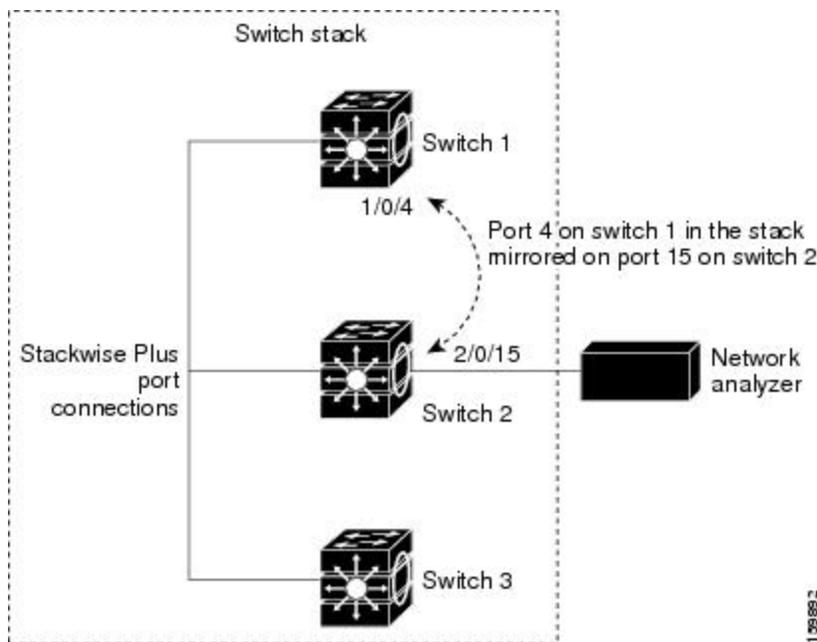


図 6: デバイス スタックでのローカル SPAN の設定例

これは、device 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。



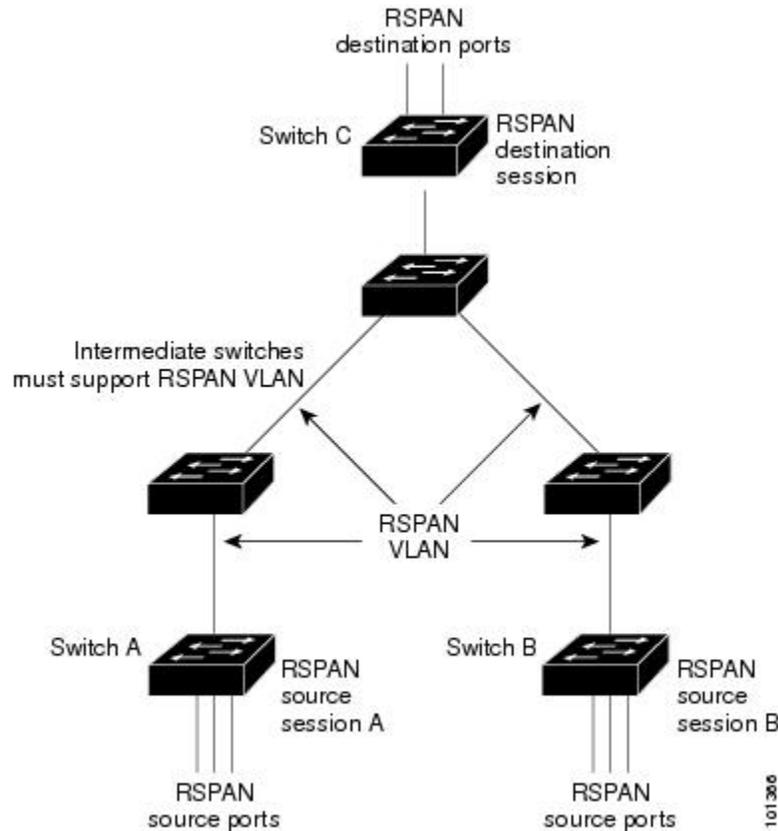
リモート SPAN

RSPAN は、異なる devices (または異なる device 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ため、ネットワーク上で複数の devices 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、ユーザーが指定した 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 devices 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 を監視する宛先セッションに転送されます。各 RSPAN 送信元 device 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

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ーし、そのモニター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ーが指定した入力および出力の packets 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 device 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 のタグングを除去し、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ーに提供します。

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

- ポートまたは VLAN を送信元にはできますが、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 device または device 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。device または device 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 device 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 device 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 のポートをトラフィック監視するなど、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 が有効の場合、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 をモニターすると、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。
- ディゼ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。
- device 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 device または device 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、device 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ーリングします。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、device 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ーリングします。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 をモニターすることもできます。これはデフォルトです。

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際に **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ーされます。

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。
- deviceの輻輳が原因でドロップされた出力パケットは、出力 SPAN からドロップされません。

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 モニター用とポート B での TX モニター用に双方向 (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 device 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 (別名モニター側ポート) 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ーできます。

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。

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ーできます。
- モニターする方向 (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ーできます。
- 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ー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ーできます。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ーされます。

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ーできます。
- 指定されたポートでは、モニター対象の 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、モニターされません。
- 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、モニター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。
- モニターできるのは、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ーする場合、デフォルトでは、トランク上でアクティブなすべての VLAN がモニターされます。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ー対象を特定の 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 のみがモニターされます。
- 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。

宛先ポート

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 パケットをユーザー（通常はネットワーク アナライザ）に送信する宛先ポート（別名モニター側ポート）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deviceまたはdevice 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device上にあります。RSPAN 送信元セッションのみを実行するdeviceまたはdevice 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。
- 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、モニターされません。
- deviceまたはdevice 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 トランキンク 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 devices 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 device 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、device 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 セッションが無効になると、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、devices 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ができます。グループが SPAN 送信元として設定されている場合、グループ全体が監視されます。

監視対象の 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。監視対象の 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監視されます。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 グループおよび監視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監視できます。出力ポートおよび入力ポートの監視では、未編集の packets 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 packets 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有効の場合、出力を監視しているポートでポートセキュリティを有効にしないでください。RSPAN 送信元セッションでは、出力を監視しているポートでポートセキュリティを有効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 を有効にできますが、SPAN 宛先としてこのポートを削除するまで、IEEE 802.1x は無効に設定されます。

SPAN セッションでは、入力転送が宛先ポートで有効の場合、出力を監視しているポートで IEEE 802.1x を有効にしないでください。RSPAN 送信元セッションでは、出力を監視しているポートで IEEE 802.1x を有効にしないでください。

SPAN と RSPAN とデバイス スタック

devices のスタックは 1 つの論理 device 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 devices である場合があります。したがって、スタック内の

devices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。device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、device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** 形式を使用すると、**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 をモニターするには、**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 packets を選択的にフィルタリングまたはモニターできます。RSPAN 送信元 devices 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 devices 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 devices 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 devices で RSPAN がサポートされている。

SPAN 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。

手順の概要

1. **enable**
2. **configure terminal**
3. **no monitor session** {*session_number* | **all** | **local** | **remote**}
4. **monitor session** *session_number* **source** { **interface** *interface-id* | **vlan** *vlan-id* } [, | -] [**both** | **rx** | **tx**]
5. **monitor session** *session_number* **destination** { **interface** *interface-id* [, | -] [**encapsulation** {**replicate** | **dot1q**}] }
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	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。 <ul style="list-style-type: none"> 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（モニター対象ポート）を指定します。 <ul style="list-style-type: none"> 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、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）。 (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。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • (任意) [, -]には、一連のインターフェイスまたはインターフェイスの範囲を指定します。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。 <ul style="list-style-type: none"> • 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モニターします。 • tx : 送信トラフィックをモニターします。 <p>(注) monitor session <i>session_number</i>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。</p>
<p>ステップ 5</p>	<p>monitor session <i>session_number</i> destination { interface <i>interface-id</i> [, -] [encapsulation {replicate dot1q}] }</p> <p>例 :</p> <pre>Device(config)# monitor session 1 destination interface gigabitethernet1/0/2 encapsulation replicate</pre>	<p>SPANセッションおよび宛先ポート（モニター側ポート）を指定します。設定変更が有効になると、ポートのLEDがオレンジ色に変わります。LEDはSPAN宛先の設定を削除した後にのみ、元の状態（緑色）に戻ります。</p> <p>(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。</p> <ul style="list-style-type: none"> • <i>session_number</i>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。 • <i>interface-id</i>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) [, -]には、一連のインターフェイスまたはインターフェイスの範囲を指定します。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。

	コマンドまたはアクション	目的
		<p>(任意) encapsulation replicate には、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。</p> <p>(任意) encapsulation dot1q は宛先インターフェイスが IEEE 802.1Q カプセル化の送信元インターフェイスの着信パケットを受け入れるように指定します。</p> <p>(注) monitor session session_number destination 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。</p>
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。

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。

手順の概要

1. **enable**
2. **configure terminal**
3. **no monitor session {session_number | all | local | remote}**
4. **monitor session session_number source { interface interface-id | vlan vlan-id} [,|-] [both | rx | tx]**

5. **monitor session** *session_number* **destination** { **interface** *interface-id* [, | -] [**encapsulation** **replicate**[**ingress** { **dot1q** **vlan** *vlan-id* | **untagged** **vlan** *vlan-id* | **vlan** *vlan-id*}]}
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例 : デバイス> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例 : デバイス# 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 { <i>session_number</i> all local remote } 例 : デバイス (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。 <ul style="list-style-type: none"> • <i>session_number</i> の範囲は、1 ~ 4 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 <i>session_number</i> source { interface <i>interface-id</i> vlan <i>vlan-id</i> } [, -] [both rx tx] 例 : デバイス (config)# monitor session 2 source gigabitethernet0/1 rx	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。
ステップ 5	monitor session <i>session_number</i> destination { interface <i>interface-id</i> [, -] [encapsulation replicate [ingress { dot1q vlan <i>vlan-id</i> untagged vlan <i>vlan-id</i> vlan <i>vlan-id</i> }]} 例 : デバイス (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。 <ul style="list-style-type: none"> • <i>session_number</i> 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。 • <i>interface-id</i> 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

	コマンドまたはアクション	目的
		<p>必要があります。EtherChannel や VLAN は指定できません。</p> <ul style="list-style-type: none"> • (任意) <code>[, -]</code> : 一連のインターフェイスまたはインターフェイスの範囲を指定します。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 : 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。 <ul style="list-style-type: none"> • 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け入れます。 • 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受け入れます。
ステップ 6	end 例 : デバイス (config) # end	特権 EXEC モードに戻ります。
ステップ 7	show running-config 例 : デバイス # show running-config	入力を確認します。
ステップ 8	copy running-config startup-config 例 : デバイス # 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。

手順の概要

1. **enable**
2. **configure terminal**
3. **no monitor session** {*session_number* | **all** | **local** | **remote**}
4. **monitor session** *session_number* **source interface** *interface-id*
5. **monitor session** *session_number* **filter vlan** *vlan-id* [, | -]
6. **monitor session** *session_number* **destination** {**interface** *interface-id* [, | -] [**encapsulation replicate**]}
7. **end**
8. **show running-config**
9. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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 { <i>session_number</i> all local remote } 例： デバイス(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。 <ul style="list-style-type: none"> • <i>session_number</i> 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 <i>session_number</i> source interface <i>interface-id</i> 例：	送信元ポート（モニター対象ポート）と SPAN セッションの特性を指定します。 <ul style="list-style-type: none"> • <i>session_number</i> の範囲は、1 ~ 66 です。

	コマンドまたはアクション	目的
	<pre>デバイス(config)# monitor session 2 source interface gigabitethernet1/0/2 rx</pre>	<ul style="list-style-type: none"> • <i>interface-id</i> 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	<p>monitor session <i>session_number</i> filter vlan <i>vlan-id</i> [, -]</p> <p>例 :</p> <pre>デバイス(config)# monitor session 2 filter vlan 1 - 5 , 9</pre>	<p>SPAN 送信元トラフィックを特定の VLAN に制限します。</p> <ul style="list-style-type: none"> • <i>session_number</i> 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。 • <i>vlan-id</i> 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	<p>monitor session <i>session_number</i> destination {interface <i>interface-id</i> [, -] [encapsulation replicate]}</p> <p>例 :</p> <pre>デバイス(config)# monitor session 2 destination interface gigabitethernet1/0/1</pre>	<p>SPANセッションおよび宛先ポート (モニター側ポート) を指定します。</p> <ul style="list-style-type: none"> • <i>session_number</i> 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。 • <i>interface-id</i> 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) [, -] には、一連のインターフェイスまたはインターフェイスの範囲を指定します。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) encapsulation replicate には、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	<p>end</p> <p>例 :</p> <pre>デバイス(config)# end</pre>	<p>特権 EXEC モードに戻ります。</p>

	コマンドまたはアクション	目的
ステップ 8	show running-config 例： デバイス# <code>show running-config</code>	入力を確認します。
ステップ 9	copy running-config startup-config 例： デバイス# <code>copy running-config startup-config</code>	(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。

手順の概要

1. `enable`
2. `configure terminal`
3. `vlan vlan-id`
4. `remote-span`
5. `end`
6. `show running-config`
7. `copy running-config startup-config`

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> <code>enable</code>	特権 EXEC モードを有効にします。 • パスワードを入力します (要求された場合)。
ステップ 2	configure terminal 例： デバイス# <code>configure terminal</code>	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 <i>vlan-id</i> 例：	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

	コマンドまたはアクション	目的
	デバイス (config) # vlan 100	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 : デバイス (config-vlan) # remote-span	VLAN を RSPAN VLAN として設定します。
ステップ 5	end 例 : デバイス (config-vlan) # end	特権 EXEC モードに戻ります。
ステップ 6	show running-config 例 : デバイス # show running-config	入力を確認します。
ステップ 7	copy running-config startup-config 例 : デバイス # 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 devices 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 device に RSPAN VLAN を作成し、VTP がこの RSPAN VLAN を VTP ドメイン内の他の devices 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 devices、および中間 devices 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

セッション 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、モニター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。

手順の概要

1. **enable**
2. **configure terminal**
3. **no monitor session {session_number | all | local | remote}**
4. **monitor session session_number source {interface interface-id | vlan vlan-id} [, | -] [both | rx | tx]**
5. **monitor session session_number destination remote vlan vlan-id**
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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： デバイス(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。 • 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。

	コマンドまたはアクション	目的
ステップ 4	<p>monitor session <i>session_number</i> source {interface <i>interface-id</i> vlan <i>vlan-id</i>} [, -] [both rx tx]</p> <p>例 :</p> <pre>デバイス(config)# monitor session 1 source interface gigabitethernet1/0/1 tx</pre>	<p>RSPAN セッションおよび送信元ポート（モニター対象ポート）を指定します。</p> <ul style="list-style-type: none"> • <i>session_number</i> 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。 <ul style="list-style-type: none"> • <i>interface-id</i> 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ンネル論理インターフェイス (port-channel <i>port-channel-number</i>) があります。有効なポートチャンネル番号は 1 ～ 48 です。 • <i>vlan-id</i> には、モニターする送信元 VLAN を指定します。指定できる範囲は 1 ～ 4094 です（RSPAN VLAN は除く）。 <p>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。</p> <ul style="list-style-type: none"> • (任意) [, -] : 一連のインターフェイスまたはインターフェイスの範囲を指定します。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。 <ul style="list-style-type: none"> • 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モニターします。 • tx : 送信トラフィックをモニターします。
ステップ 5	<p>monitor session <i>session_number</i> destination remote vlan <i>vlan-id</i></p> <p>例 :</p>	<p>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。</p>

フィルタリングする VLAN の指定

	コマンドまたはアクション	目的
	<pre>デバイス(config)# monitor session 1 destination remote vlan 100</pre>	<ul style="list-style-type: none"> • <i>session_number</i> には、ステップ 4 で指定した番号を入力します。 • <i>vlan-id</i> 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	<p>end</p> <p>例 :</p> <pre>デバイス(config)# end</pre>	特権 EXEC モードに戻ります。
ステップ 7	<p>show running-config</p> <p>例 :</p> <pre>デバイス# show running-config</pre>	入力を確認します。
ステップ 8	<p>copy running-config startup-config</p> <p>例 :</p> <pre>デバイス# copy running-config startup-config</pre>	(任意) コンフィギュレーションファイルに設定を保存します。

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。

手順の概要

1. **enable**
2. **configure terminal**
3. **no monitor session {*session_number* | all | local | remote}**
4. **monitor session *session_number* source interface *interface-id***
5. **monitor session *session_number* filter vlan *vlan-id* [, | -]**
6. **monitor session *session_number* destination remote vlan *vlan-id***
7. **end**
8. **show running-config**
9. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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： デバイス(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。 <ul style="list-style-type: none"> 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： デバイス(config)# monitor session 2 source interface gigabitethernet1/0/2 rx	送信元ポート（モニター対象ポート）と SPAN セッションの特性を指定します。 <ul style="list-style-type: none"> 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： デバイス(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。 <ul style="list-style-type: none"> 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。

	コマンドまたはアクション	目的
ステップ 6	monitor session session_number destination remote vlan vlan-id 例 : デバイス (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。 <ul style="list-style-type: none"> • <i>session_number</i> 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。 • <i>vlan-id</i> 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 : デバイス (config) # end	特権 EXEC モードに戻ります。
ステップ 8	show running-config 例 : デバイス # show running-config	入力を確認します。
ステップ 9	copy running-config startup-config 例 : デバイス # 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 device または device スタック（送信元セッションが設定されていない device または device スタック）に設定します。

この device 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。

手順の概要

1. **enable**
2. **configure terminal**
3. **vlan vlan-id**
4. **remote-span**
5. **exit**
6. **no monitor session {session_number | all | local | remote}**
7. **monitor session session_number source remote vlan vlan-id**

8. **monitor session** *session_number* **destination interface** *interface-id*
9. **end**
10. **show running-config**
11. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none">• パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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 <i>vlan-id</i> 例： デバイス(config)# vlan 901	送信元device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devices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： デバイス(config-vlan)# remote-span	VLAN を RSPAN VLAN として識別します。
ステップ 5	exit 例： デバイス(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 { <i>session_number</i> all local remote } 例： デバイス(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。 <ul style="list-style-type: none">• <i>session_number</i> 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。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • 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 source remote vlan vlan-id 例 : デバイス (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。 <ul style="list-style-type: none"> • 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 : デバイス (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。 <ul style="list-style-type: none"> • 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 : デバイス (config) # end	特権 EXEC モードに戻ります。
ステップ 10	show running-config 例 : デバイス # show running-config	入力を確認します。
ステップ 11	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
	デバイス# <code>copy running-config startup-config</code>	

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。

手順の概要

1. `enable`
2. `configure terminal`
3. `no monitor session {session_number | all | local | remote}`
4. `monitor session session_number source remote vlan vlan-id`
5. `monitor session session_number destination {interface interface-id [, | -] [ingress { dot1q vlan vlan-id | untagged vlan vlan-id | vlan vlan-id}]}`
6. `end`
7. `show running-config`
8. `copy running-config startup-config`

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<code>enable</code> 例： デバイス> <code>enable</code>	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	<code>configure terminal</code> 例： デバイス# <code>configure terminal</code>	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	<code>no monitor session {session_number all local remote}</code> 例： デバイス(config)# <code>no monitor session 2</code>	セッションに対する既存の SPAN 設定を削除します。 • <code>session_number</code> の範囲は、1 ～ 66 です。 • <code>all</code> : すべての SPAN セッションを削除します。 • <code>local</code> : すべてのローカル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
		<ul style="list-style-type: none"> • 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 <i>session_number</i> source remote vlan <i>vlan-id</i> 例 : デバイス (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。 <ul style="list-style-type: none"> • <i>session_number</i> の範囲は、1 ~ 66 です。 • <i>vlan-id</i> 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 <i>session_number</i> destination {interface <i>interface-id</i> [, -] [ingress { dot1q vlan <i>vlan-id</i> untagged vlan <i>vlan-id</i> vlan <i>vlan-id</i> }]} 例 : デバイス (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。 <ul style="list-style-type: none"> • <i>session_number</i> 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。 • <i>interface-id</i> 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) [, -] には、一連のインターフェイスまたはインターフェイスの範囲を指定します。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。 • 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するには、ingress を追加のキーワードと一緒に入力します。 <ul style="list-style-type: none"> • dot1q vlan <i>vlan-id</i> : 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。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • 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 : デバイス(config)# end	特権 EXEC モードに戻ります。
ステップ 7	show running-config 例 : デバイス# show running-config	入力を確認します。
ステップ 8	copy running-config startup-config 例 : デバイス# 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 (モニター) 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。

手順の概要

1. **enable**
2. **configure terminal**
3. **no monitor session** {*session_number* | **all** | **local** | **remote**}
4. **monitor session** *session_number* **source** {**interface** *interface-id* | **vlan** *vlan-id*} [,|-] [**both** | **rx** | **tx**]
5. **monitor session** *session_number* **destination** {**interface** *interface-id* [,|-] [**encapsulation replicate**]}
6. **monitor session** *session_number* **filter** {**ip** | **ipv6** | **mac**} **access-group** {*access-list-number* | *name*}
7. **end**
8. **show running-config**
9. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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： デバイス (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。 <ul style="list-style-type: none"> <i>session_number</i> 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： デバイス (config)# monitor session 2 source interface gigabitethernet1/0/1	SPAN セッションおよび送信元ポート（モニター対象ポート）を指定します。 <ul style="list-style-type: none"> <i>session_number</i> の範囲は、1 ~ 66 です。 <i>interface-id</i> 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。 <i>vlan-id</i> 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）。 （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。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • (任意) <code>[, -]</code> : 一連のインターフェイスまたはインターフェイスの範囲を指定します。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) <code>[both rx tx]</code> : 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ーします。 <ul style="list-style-type: none"> • both : 送信トラフィックと受信トラフィックの両方を監視します。これはデフォルトです。 • rx : 受信トラフィックをモニターします。 • tx : 送信トラフィックをモニターします。 <p>(注) monitor session <i>session_number</i>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。</p>
<p>ステップ 5</p>	<p>monitor session <i>session_number</i> destination {interface <i>interface-id</i> [, -] [encapsulation replicate]}</p> <p>例 :</p> <pre> デバイス(config)# monitor session 2 destination interface gigabitethernet1/0/2 encapsulation replicate </pre>	<p>SPANセッションおよび宛先ポート（モニター側ポート）を指定します。</p> <ul style="list-style-type: none"> • <i>session_number</i> 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。 • destination では、次のパラメータを指定します。 <ul style="list-style-type: none"> • <i>interface-id</i> 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) <code>[, -]</code> には、一連のインターフェイスまたはインターフェイスの範囲を指定します。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) encapsulation replicate には、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、

	コマンドまたはアクション	目的
		<p>ネイティブ形式 (タグなし) でのパケットの送信です。</p> <p>(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。</p> <p>monitor session <i>session_number</i> destination 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。</p>
ステップ 6	<p>monitor session <i>session_number</i> filter {ip ipv6 mac} access-group {<i>access-list-number</i> <i>name</i>}</p> <p>例 :</p> <pre>デバイス(config)# monitor session 2 filter ipv6 access-group 4</pre>	<p>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。</p> <ul style="list-style-type: none"> • <i>session_number</i> 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。 • <i>access-list-number</i> 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。 • <i>name</i> 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	<p>end</p> <p>例 :</p> <pre>デバイス(config)# end</pre>	特権 EXEC モードに戻ります。
ステップ 8	<p>show running-config</p> <p>例 :</p> <pre>デバイス# show running-config</pre>	入力を確認します。
ステップ 9	<p>copy running-config startup-config</p> <p>例 :</p> <pre>デバイス# copy running-config startup-config</pre>	(任意) コンフィギュレーションファイルに設定を保存します。

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。

手順の概要

1. **enable**
2. **configure terminal**
3. **no monitor session** {*session_number* | **all** | **local** | **remote**}
4. **monitor session** *session_number* **source** { **interface** *interface-id* | **vlan** *vlan-id* } [, | -] [**both** | **rx** | **tx**]
5. **monitor session** *session_number* **destination remote** **vlan** *vlan-id*
6. **vlan** *vlan-id*
7. **remote-span**
8. **exit**
9. **monitor session** *session_number* **filter** {**ip** | **ipv6** | **mac**} **access-group** {*access-list-number* | *name*}
10. **end**
11. **show running-config**
12. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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 { <i>session_number</i> all local remote }	セッションに対する既存の SPAN 設定を削除します。 • <i>session_number</i> 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 <i>session_number</i> source { interface <i>interface-id</i> vlan <i>vlan-id</i> } [, -] [both rx tx] 例：	SPAN セッションおよび送信元ポート（モニター対象ポート）を指定します。 • <i>session_number</i> の範囲は、1 ～ 66 です。

	コマンドまたはアクション	目的
	デバイス (config) # monitor session 2 source interface gigabitethernet1/0/1	<ul style="list-style-type: none"> • 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、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)。 <p>(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。</p> <ul style="list-style-type: none"> • (任意) [, -] : 一連のインターフェイスまたはインターフェイスの範囲を指定します。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ーします。 • 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ーします。これはデフォルトです。 • rx : 受信トラフィックをモニターします。 • tx : 送信トラフィックをモニターします。 <p>(注) monitor session session_numbersource 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。</p>
ステップ 5	monitor session session_number destination remote vlan vlan-id 例 :	RSPAN セッションと宛先 RSPAN VLAN を指定します。 <ul style="list-style-type: none"> • session_number には、ステップ 4 で指定した番号を入力します。

	コマンドまたはアクション	目的
	デバイス (config) # monitor session 2 destination remote vlan 5	<ul style="list-style-type: none"> • <i>vlan-id</i> 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 <i>vlan-id</i> 例 : デバイス (config) # vlan 10	VLAN コンフィギュレーションモードを開始します。 <i>vlan-id</i> 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 : デバイス (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 : デバイス (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 9	monitor session <i>session_number</i> filter {ip ipv6 mac} access-group {<i>access-list-number</i> <i>name</i>} 例 : デバイス (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。 <ul style="list-style-type: none"> • <i>session_number</i> 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。 • <i>access-list-number</i> 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。 • <i>name</i> 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 : デバイス (config) # end	特権 EXEC モードに戻ります。
ステップ 11	show running-config 例 : デバイス # show running-config	入力を確認します。
ステップ 12	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
	デバイス# <code>copy running-config startup-config</code>	

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モニタ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	目的
<code>show monitor</code>	現在の SPAN、RSPAN、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。

```

デバイス> enable
デバイス# configure terminal
デバイス(config)# no monitor session 1
デバイス(config)# monitor session 1 source interface gigabitethernet1/0/1
デバイス(config)# monitor session 1 destination interface gigabitethernet1/0/2
encapsulation replicate
デバイス(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。

```

デバイス> enable
デバイス# configure terminal
デバイス(config)# no monitor session 1 source interface gigabitethernet1/0/1
デバイス(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。

```

デバイス> enable

```

```
デバイス# configure terminal
デバイス(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GigabitEthernet2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。

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

次に、SPANセッション2の既存の設定を削除し、ギガビットイーサネットソース送信元ポート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、そのトラフィックを送信元ポートと同じ出力カプセル化方式の宛先ギガビットイーサネットポート2に送信し、デフォルト入力VLANとしてVLAN 6を使用した入力転送をイネーブルにする例を示します。

```
デバイス> enable
デバイス# configure terminal
デバイス(config)# no monitor session 2
デバイス(config)# monitor session 2 source gigabitethernet0/1 rx
デバイス(config)# monitor session 2 destination interface gigabitethernet0/2 encapsulation
replicate ingress vlan 6
デバイス(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GigabitEthernet2で受信されたトラフィックをモニターするように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GigabitEthernet1に送信する例を示します。

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

```
デバイス> enable
```

```

デバイス# configure terminal
デバイス(config)# vlan 901
デバイス(config-vlan)# remote span
デバイス(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。

```

デバイス> enable
デバイス# configure terminal
デバイス(config)# no monitor session 1
デバイス(config)# monitor session 1 source interface gigabitethernet1/0/1 tx
デバイス(config)# monitor session 1 source interface gigabitethernet1/0/2 rx
デバイス(config)# monitor session 1 source interface port-channel 2
デバイス(config)# monitor session 1 destination remote vlan 901
デバイス(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。

```

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

```

デバイス> enable
デバイス# configure terminal
デバイス(config)# monitor session 1 source remote vlan 901
デバイス(config)# monitor session 1 destination interface gigabitethernet2/0/1
デバイス(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、VLAN 6 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。

```

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

```

SPAN および RSPAN の機能の履歴と情報

リリース	変更内容
Cisco IOS XE Everest 16.5.1a	<p>スイッチポートアナライザ (SPAN) : スニファアアナライザまたは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。</p> <p>この機能が導入されました。</p>
Cisco IOS XE Everest 16.5.1a	<p>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。</p> <p>この機能が導入されました。</p>
Cisco IOS XE Everest 16.5.1a	<p>スイッチポートアナライザ (SPAN) - 分散型出力</p> <p>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。</p> <p>この機能が導入されました。</p>



第 7 章

ERSPAN の設定

- [ERSPAN の設定 \(129 ページ\)](#)

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

- アクセスコントロールリスト (ACL) のフィルタは、トンネルにモニター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。
- 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 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 Generic Routing Encapsulation (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。

- IPv4 接続の **ip routing** コマンドと IPv6 接続の **ipv6 unicast-routing** コマンドを無効にすると、宛先ポートへの ERSPAN トラフィックフローが停止します。

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ーし、モニターされたトラフィックを宛先ポートに送信できます。ERSPAN は、スイッチプロンプトデバイスやリモートモニタリング (RMON) プロンプト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 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。宛先デバイスでの管理トラフィックのドロップにつながる可能性のある、トラフィックのオーバーサブスクリプションを回避するには、送信元デバイスで送信元セッションを設定する前に、宛先セッションが設定され、宛先デバイスで動作していることを確認してください。

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。

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。

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。

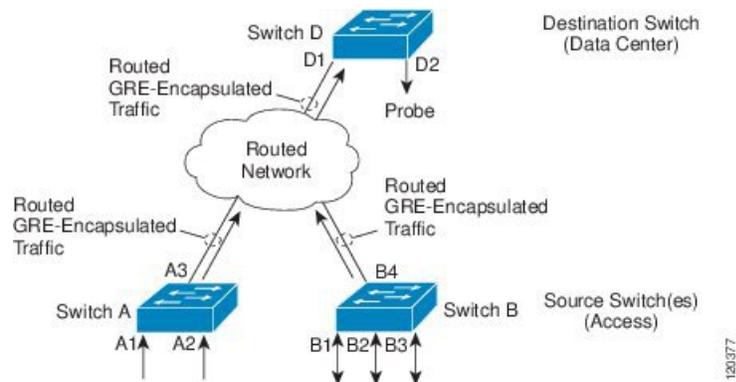
- セッション ID
- ERSPAN フロー ID
- セッションでモニターされる送信元ポートまたは送信元 VLAN の一覧
- IP タイプオブサービス (ToS) および IP 有効時間 (TTL) などの、GRE エンベロープに関連したオプション属性

- 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



- (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。
 - IPv4 配信/転送ヘッダーのみサポートされます。

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ー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ーされる VLAN です。

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、モニター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。

手順の概要

1. **enable**
2. **configure terminal**

3. **interface** *interface-type interface-number*
4. **monitor session** *span-session-number type erspan-source*
5. **description** *string*
6. **source** {**interface** *interface-type interface-number* | **vlan** *vlan-id*} [, | - | **both** | **rx** | **tx**]
7. **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* [, | -]}
8. **destination**
9. **erspan-id** *erspan-flow-id*
10. **ip address** *ip-address*
11. **ip ttl** *ttl-value*
12. **origin ip-address** *ip-address*
13. **exit**
14. **no shutdown**
15. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	interface <i>interface-type interface-number</i> 例： Device(config)# interface GigabitEthernet 1/0/1	ERSPAN 送信元セッションを設定するインターフェイスを指定します。
ステップ 4	monitor session <i>span-session-number type erspan-source</i> 例： Device(config)# monitor session 1 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 のモニター送信元セッション コンフィギュレーション モードを開始します。 • <i>span-session-number</i> 引数の範囲は 1～66 です。同じセッション番号は複数回使用できません。 • 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。 • セッション ID (<i>span-session-number</i> 引数によって設定) 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、

	コマンドまたはアクション	目的
		入力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 5	description <i>string</i> 例： Device(config-mon-erspan-src)# description source1	(任意) ERSPAN 送信元セッションの説明を入力します。 <ul style="list-style-type: none"> <i>string</i> 引数には最大 240 文字を使用できます。ただし、特殊文字またはスペースは使用できません。
ステップ 6	source { interface <i>interface-type interface-number</i> vlan <i>vlan-id</i> } [, - both rx tx] 例： Device(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、およびモニターするトラフィックの方向を設定します。
ステップ 7	filter { ip access-group { <i>standard-access-list</i> <i>expanded-access-list</i> <i>acl-name</i> } ipv6 access-group <i>acl-name</i> mac access-group <i>acl-name</i> vlan <i>vlan-ID</i> [, -]} 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 8	destination 例： Device(config-mon-erspan-src)# destination	ERSPAN 送信元セッションの宛先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	erspan-id <i>erspan-flow-id</i> 例： Device(config-mon-erspan-src-dst)# erspan-id 100	ERSPAN トラフィックを識別するため、送信元および宛先セッションで使用される ID を設定します。これは、ERSPAN 宛先セッションの設定でも入力する必要があります。
ステップ 10	ip address <i>ip-address</i> 例： Device(config-mon-erspan-src-dst)# ip address 10.1.0.2	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 11	ip ttl <i>ttl-value</i> 例： Device(config-mon-erspan-src-dst)# ip ttl 32	(任意) ERSPAN トラフィック内のパケットの IP TTL 値を設定します。
ステップ 12	origin ip-address <i>ip-address</i> 例：	ERSPAN トラフィックの送信元として使用される IP アドレスを設定します。

	コマンドまたはアクション	目的
	Device(config-mon-erspan-src-dst)# origin ip address 10.10.0.1	
ステップ 13	exit 例 : Device(config-mon-erspan-src-dst)# exit	ERSPAN 送信元セッション宛先コンフィギュレーションモードを終了し、ERSPAN 送信元セッションコンフィギュレーションモードに戻ります。
ステップ 14	no shutdown 例 : Device(config-mon-erspan-src)# no shutdown	インターフェイスで設定されたセッションをイネーブルにします。
ステップ 15	end 例 : Device(config-mon-erspan-src)# end	ERSPAN 送信元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

次に、ERSPAN 送信元セッションを設定する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# monitor session 1 type erspan-source
Device(config-mon-erspan-src)# description source1
Device(config-mon-erspan-src)# source interface fastethernet 0/1 rx
Device(config-mon-erspan-src)# filter vlan 3
Device(config-mon-erspan-src)# no shutdown
Device(config-mon-erspan-src)# destination
Device(config-mon-erspan-src-dst)# ip address 10.1.0.2
Device(config-mon-erspan-src-dst)# erspan-id 2
Device(config-mon-erspan-src-dst)# origin ip-address 203.0.113.2
Device(config-mon-erspan-src-dst)# ip ttl 32
Device(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、**show monitor session erspan-source** コマンドの出力例を示します。

```
Device# show monitor session erspan-source

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。

```
Device# 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 **show capability feature monitor 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。

```
Device# 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 and III
ACL filter Supported: true
Fragmentation Supported: true
Truncation Supported: false
Sequence number Supported: false
QOS Supported: true
```

次の **show capability feature monitor erspan-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。

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

```
ERSPAN Destination Session Supported: false
```

その他の参考資料

RFC

標準/RFC	タイトル
RFC 2784	『Generic Routing Encapsulation (GRE)』

シスコのテクニカル サポート

説明	リンク
<p>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。</p> <p>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、Cisco Notification Service (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。</p> <p>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。</p>	http://www.cisco.com/support

ERSPAN の設定に関する機能情報

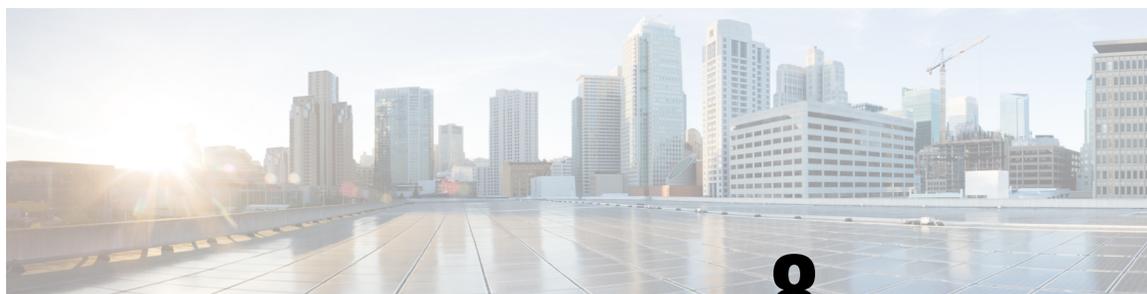
次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りがない限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、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	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。

機能名	リリース	機能情報
ERSPAN	Cisco IOS XE Gibraltar 16.11.1	<p>宛先セッションのサポートが導入されました。</p> <p>vrf コマンドと ip dscp コマンド、および sgt キーワードが導入されました。</p> <p>ERSPAN は、デバイスをタイプ III ヘッダーに設定するように拡張されました。</p> <p>header-type 3 コマンドが導入されました。</p> <p>ERSPAN 切り捨てとタイムスタンプのサポートが導入されました。</p> <p>mtu コマンドが導入されました。</p>



第 8 章

パケットキャプチャの設定

- [パケットキャプチャ設定の前提条件 \(139 ページ\)](#)
- [パケットキャプチャ設定の制約事項 \(140 ページ\)](#)
- [パケットキャプチャについて \(142 ページ\)](#)
- [パケットキャプチャの設定方法 \(153 ページ\)](#)
- [パケットキャプチャの設定例 \(171 ページ\)](#)
- [その他の参考資料 \(189 ページ\)](#)
- [パケットキャプチャ設定の機能履歴と情報 \(190 ページ\)](#)

パケットキャプチャ設定の前提条件

パケットキャプチャは Cisco Catalyst 9300 シリーズ スイッチでサポートされています。

ここでは、パケットキャプチャの設定に関する前提条件について説明します。

Wireshark 設定の前提条件

- Wireshark は、次を実行しているスイッチのみでサポートされています DNA Advantage
- 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。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。

組み込みパケットキャプチャ設定の前提条件

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	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、ユーザーが指定します。
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設定の制約事項

ここでは、パケットキャプチャの設定に関する制約事項について説明します。

Wireshark 設定の制約事項

- 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。
- 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、DNA Advantage のフラッシュのサイズに限定されます。
- 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デバイス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。
- 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。
- 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。
- 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。
- 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。
- 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。

- 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。
- 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。

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。
- 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。

組み込みパケットキャプチャの制約事項

- レイヤ2 EtherChannels はサポートされません。
- 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。
- 組み込みパケットキャプチャ（EPC）は、ポートチャンネル、スイッチ仮想インターフェイス（SVI）、およびサブインターフェイスを含む論理ポートではサポートされません。物理ポート上でのみサポートされます。
- ユーザーがスイッチポートからルーテッドポート（レイヤ2からレイヤ3）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。
- 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、デバイスの書き換えによって行われた変更（TTL、VLAN タグ CoS、チェックサム、および MAC アドレス、DSCP、プレジデント、UP など）が反映されないこともあります。
- パケットキャプチャの最小設定可能期間は1秒ですが、パケットキャプチャは少なくとも2秒間機能します。
- 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。

- 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。
- 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。
- 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。
- 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。
- Control and Provisioning of Wireless Access Points (CAPWAP) などのプロトコルのデコードは、DNA Advantage でサポートされています。
- 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。
- 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（レイヤ2スイッチポート、レイヤ3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、レイヤ3 インターフェイスとレイヤ2 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。

パケットキャプチャについて

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、DNA Advantage でサポートされています。

Wireshark について

Wireshark は、複数のプロトコルをサポートし、テキストベース ユーザー 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があります。

複数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。

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に関連する 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 9300 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デバイス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。

表示フィルタ

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

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。

- デバイス 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う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作成したファイルを作成し、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入力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グ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 機能がデバイス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 設定のガイドライン

- 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。
- 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。
- 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。
- 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。
 - 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。
- 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。
 - キャプチャ期間
 - キャプチャされたパケットの数
 - ファイル サイズ
 - パケットのセグメント サイズ
 - キャプチャセッション中に、デバイス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。
 - 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。
 - 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。
 - 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。
 - 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。
 - 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。
 - 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。
- 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。
- 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。

- 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。ユーザー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** 形式を使用します。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 GigabitEthernet1/0/1 in** と入力します。ここで、**GigabitEthernet1/0/1** は接続ポイントです。インターフェイス **GigabitEthernet1/0/2** も接続する必要がある場合は、次のように入力します **monitor capture mycap interface GigabitEthernet1/0/2 in**

- 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ののみ 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。
- 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
持続時間	制限なし
パケット	制限なし
パケット長	制限なし (フルパケット)
ファイルサイズ	制限なし
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	直線

組み込みパケットキャプチャについて

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有効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方法

ここでは、パケットキャプチャの設定について説明します。

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できません。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。

手順の概要

1. **enable**
2. **monitor capture** {*capture-name*} {**interface** *interface-type interface-id* | **control-plane**} {**in** | **out** | **both**}
3. **monitor capture** {*capture-name*} [**match** {**any** | **ipv4 any any** | **ipv6**} **any any**]
4. **show monitor capture** {*capture-name*} [**parameter**]
5. **show capwap summary**
6. **show running-config**
7. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	monitor capture { <i>capture-name</i> } { interface <i>interface-type interface-id</i> control-plane } { in out both }	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。
	例： Device# monitor capture mycap interface GigabitEthernet1/0/1 in	キーワードの意味は次のとおりです。 • <i>capture-name</i> : 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

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • (任意) interface<i>interface-type interface-id</i> : 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)。 (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。 <p><i>interface-type</i> には次のいずれかを使用します。</p> <ul style="list-style-type: none"> • 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。 <ul style="list-style-type: none"> • (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。 • in out both : キャプチャの方向を指定します。
ステップ 3	monitor capture { <i>capture-name</i> } [match { any ipv4 any any ipv6 } any any]	コア システムのフィルタを定義します。

	コマンドまたはアクション	目的
	<p>例 :</p> <pre>Device# monitor capture mycap interface GigabitEthernet1/0/1 in match any</pre>	<p>(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。</p> <p>キーワードの意味は次のとおりです。</p> <ul style="list-style-type: none"> • 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。 <p>(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。</p> <ul style="list-style-type: none"> • 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	<p>show monitor capture {<i>capture-name</i>} [parameter]</p> <p>例 :</p> <pre>Device# show monitor capture mycap parameter monitor capture mycap interface GigabitEthernet1/0/1 in monitor capture mycap match any</pre>	<p>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。</p>
ステップ 5	<p>show capwap summary</p> <p>例 :</p> <pre>Device# show capwap summary</pre>	<p>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。</p> <p>(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。</p>
ステップ 6	<p>show running-config</p> <p>例 :</p>	<p>入力を確認します。</p>

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

ユーザー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。

手順の概要

1. **enable**
2. **monitor capture** {capture-name} **match** {any | mac mac-match-string | ipv4 {any | host | protocol}{any | host} | ipv6 {any | host | protocol}{any | host}}
3. **monitor capture** {capture-name} **limit** {[duration seconds] [packet-length size] [packets num]}
4. **monitor capture** {capture-name} **file** {location filename}
5. **monitor capture** {capture-name} **file** {buffer-size size}
6. **show monitor capture** {capture-name} [parameter]
7. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	monitor capture {capture-name} match {any mac mac-match-string ipv4 {any host protocol}{any host} ipv6 {any host protocol}{any host}}	ACLまたはクラスマップで明示的に定義されたコアシステムフィルタ (ipv4 any any) を定義します。

	コマンドまたはアクション	目的
	例： Device# monitor capture mycap match ipv4 any any	
ステップ 3	monitor capture {capture-name} limit {[duration seconds] [packet-length size] [packets num]} 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-name} file {buffer-size 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 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	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。
ステップ 7	end 例： Device(config)# end	特権 EXEC モードに戻ります。

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。

手順の概要

1. **enable**
2. **no monitor capture** {capture-name} **match**
3. **no monitor capture** {capture-name} **limit** [duration] [packet-length] [packets]
4. **no monitor capture** {capture-name} **file** [location] [buffer-size]
5. **show monitor capture** {capture-name} [parameter]
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	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：	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。

	コマンドまたはアクション	目的
	<pre>Device# no monitor capture mycap limit duration packet-len Device# no monitor capture mycap limit</pre>	Wireshark のすべての制限をクリアします。
ステップ 4	<p>no monitor capture {capture-name} file [location] [buffer-size]</p> <p>例 :</p> <pre>Device# no monitor capture mycap file Device# no monitor capture mycap file location</pre>	<p>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。</p> <p>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。</p>
ステップ 5	<p>show monitor capture {capture-name} [parameter]</p> <p>例 :</p> <pre>Device# show monitor capture mycap parameter monitor capture mycap interface GigabitEthernet1/0/1 in</pre>	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	<p>end</p> <p>例 :</p> <pre>Device(config)# end</pre>	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。

手順の概要

1. **enable**
2. **no monitor capture** {capture-name}
3. **show monitor capture** {capture-name} [parameter]

4. **end**
5. **show running-config**
6. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	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： 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

削除したものと同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。

手順の概要

1. **enable**
2. **monitor capture** {capture-name} **start** [**display** [display-filter filter-string]] [**brief** | **detailed** | **dump**]
3. **monitor capture** {capture-name} **stop**
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	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。

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

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.
```

■ 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

```
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。

手順の概要

1. **enable**
2. **monitor capture** {capture-name} [**clear** | **export filename**]
3. **end**
4. **show running-config**
5. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	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、 • DNA Advantage 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。 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。

	コマンドまたはアクション	目的
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) DNA Advantage 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、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、DNA Advantage のみでサポートされています。他のすべてのタイプのライセンスでは、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。

手順の概要

1. **enable**
2. **monitor capture** *capture-name* **access-list** *access-list-name*
3. **monitor capture** *capture-name* **limit duration** *seconds*
4. **monitor capture** *capture-name* **interface** *interface-name* **both**
5. **monitor capture** *capture-name* **buffer circular size** *bytes*
6. **monitor capture** *capture-name* **start**
7. **monitor capture** *capture-name* **stop**
8. **monitor capture** *capture-name* **export** *file-location/file-name*
9. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	monitor capture <i>capture-name</i> access-list <i>access-list-name</i> 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、モニターキャプチャを設定します。
ステップ 3	monitor capture <i>capture-name</i> limit duration <i>seconds</i> 例： Device# monitor capture mycap limit duration 1000	モニターキャプチャの制限を設定します。
ステップ 4	monitor capture <i>capture-name</i> interface <i>interface-name</i> 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、モニターキャプチャを設定します。
ステップ 5	monitor capture <i>capture-name</i> buffer circular size <i>bytes</i> 例： Device# monitor capture mycap buffer circular size 10	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。
ステップ 6	monitor capture <i>capture-name</i> 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。

	コマンドまたはアクション	目的
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。

手順の概要

1. **enable**
2. **show monitor capture capture-buffer-name buffer dump**
3. **show monitor capture capture-buffer-name parameter**
4. **debug epc capture-point**
5. **debug epc provision**
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	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 <i>capture-buffer-name</i> 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有効にします。
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有効にします。
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設定例

次のセクションにパケットキャプチャの設定例を示します。

Wireshark の設定例

次のセクションに Wireshark の設定例を示します。

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
```

例 : .pcap ファイルからの詳細出力の表示

```

 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 .....
0020 ab cd .....
0030 ab cd .....
0040 ab cd ab cd ab cd ab cd .....
      Data: 0000000009c98f77abcdabcdabcdabcdabcdabcdabcd...
      [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
```

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

```

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 09 c9  ....
0030 8f 77 ab cd  .w.....
0040 ab cd  ....
0050 ab cd  ....
0060 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 09 c9  ....
0030 8f 77 ab cd  .w.....
0040 ab cd  ....
0050 ab cd  ....
0060 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 09 c9  ....
0030 8f 7a ab cd  .z.....
0040 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 .....
0040 ab cd .....
0050 ab cd .....
0060 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ー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。

```
デバイス# 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デッ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)
```

例: バッファのキャプチャの使用

```

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 .....
0020 ab cd .....
0030 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 0c.....
0040 ab cd .....
0050 ab cd .....
0060 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 0e.....
0040 ab cd .....
0050 ab cd .....
0060 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。

```
デバイス# 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、ユーザー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。

例：出力方向のパケットの簡単なキャプチャおよび保存

```
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. Reason : 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

次のセクションに EPC の設定例を示します。

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 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 00C01000 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 provisioning 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	マニュアルタイトル
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

パケットキャプチャ設定の機能履歴と情報

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だけを示しています。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。

表 10: パケットキャプチャ設定の機能情報

機能名	リリース	機能情報
パケットキャプチャの設定	Cisco IOS XE Everest 16.5.1a	この機能が導入されました。



第 9 章

Flexible NetFlow の設定

- [Flexible NetFlow の前提条件](#) (191 ページ)
- [Flexible Netflow に関する制約事項](#) (192 ページ)
- [Flexible NetFlow に関する情報](#) (195 ページ)
- [Flexible NetFlow の設定方法](#) (212 ページ)
- [Flexible NetFlow の監視](#) (226 ページ)
- [設定例 Flexible NetFlow](#) (226 ページ)
- [Flexible NetFlow の機能情報](#) (230 ページ)

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、エクスポートはディセーブル状態のままになります。
- 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 エクスポートに 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 Traffic

- The networking device must be configured for IPv4 routing.
- One of the following must be enabled on your device and on any interfaces on which you want to enable Flexible NetFlow: Cisco Express Forwarding or distributed Cisco Express Forwarding.

IPv6 Traffic

- The networking device must be configured for IPv6 routing.
- One of the following must be enabled on your device and on any interfaces on which you want to enable Flexible NetFlow: Cisco Express Forwarding IPv6 or distributed Cisco Express Forwarding.

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、レイヤ 2 ポートチャンネル インターフェイスではサポートされませんが、レイヤ 2 ポートチャンネル メンバ ポートではサポートされます。
- Flexible NetFlow は、レイヤ 3 ポートチャンネル インターフェイスとメンバポートでサポートされますが、同じトラフィックタイプと方向の両方に対して同時にサポートされることはありません。
- Traditional NetFlow 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 Application Visibility and Control (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ーは 1 つのみです。
- Flexible NetFlow および NBAR は同じインターフェイスで同時に設定できません。
- 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 モニターを、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 モニター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。
- デバイスはトンネルおよび SVI インターフェイスをサポートしていません。ただし、レイヤ 2 とレイヤ 3 の物理インターフェイスおよび VLAN コンフィギュレーションモードがサポートされています。
- 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、1 つまたは 2 つのコアをサポートできます。各オーバーフロー TCAM は、コアあたり 256 の入力エントリと 256 の出力エントリをサポートできます。
- 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 個のフローモニター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ています。

- 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、トピック「Supported Flexible NetFlow Fields」を参照してください。
- AVC フロー モニターの IPFIX エクスポート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（GigabitEthernet 0/0）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。
- Cisco TrustSec 以外のインターフェイスでは、SGT 値がゼロの場合、コマンドヘッダーがないことを意味します。Cisco TrustSec インターフェイスでは、SGT 値がゼロの場合、不明タグであることを意味します。
- IPv6 フローモニターの場合、送信元グループタグ（SGT）フィールドと宛先グループタグ（DGT）フィールドは、MAC アドレスフィールドと共存できません。
- Quality of Service（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 NetFlow 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設定された NetFlow を備えたインターフェイスで受信され、スイッチによって入力時に QoS 値が書き換えられた場合、パケットの新しい QoS 値はコレクタによってキャプチャされません。
- NetFlow レコードは、マルチプロトコル ラベル スイッチング対応（MPLS 対応）インターフェイスをサポートしません。
- MPLS ネットワーク内の MPLS ラベルに基づくデータキャプチャはサポートされていません。MPLS タグ付きパケットの IP ヘッダーフィールドのキャプチャはサポートされていません。
- 出力フローモニターは、EoMPLS モードまたは L3VPN Per-Prefix モードで出力されるフローをキャプチャしません。
- フローモニターは、レイヤ3 物理インターフェイスと論理インターフェイス（レイヤ3 ポートチャンネルインターフェイス、レイヤ3 ポートチャンネルメンバ、スイッチ仮想インターフェイス（SVI）など）間で共有することはできませんが、論理インターフェイス間またはレイヤ3 物理インターフェイス間で共有できます。

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、アカウントティング、ネットワーク モニターリング、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。

device 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。

エクスポートを使用して 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。

モニター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。モニターで、フロー レコードおよびエクスポートを 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

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

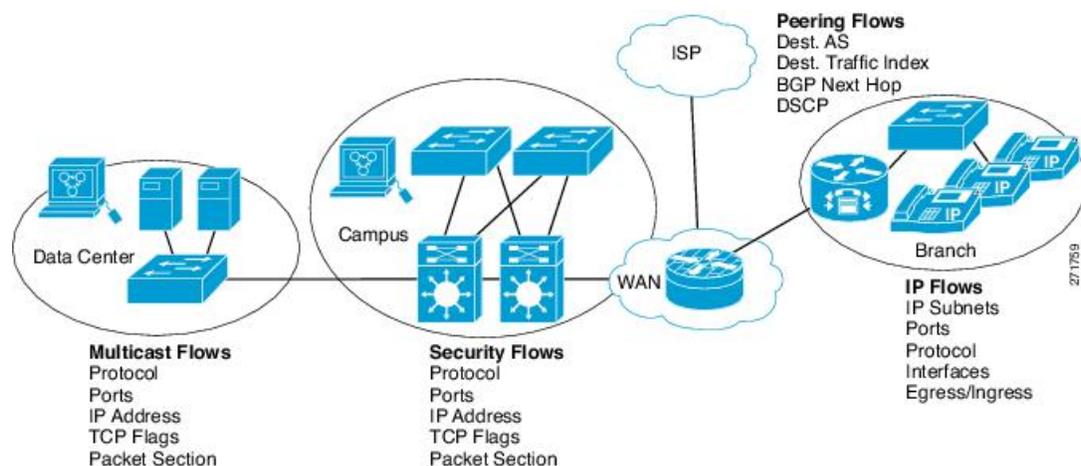
- スケーラビリティ、フロー情報の集約などの、大容量フロー認識。
- セキュリティの監視と dDoS の検出および識別のための拡張されたフロー インフラストラクチャ。
- フロー情報をネットワーク内の特定のサービスまたはオペレーションに適応させるパケットからの新しい情報。利用できるフロー情報は、Flexible NetFlow ユーザーがカスタマイズ可能。
- Cisco の柔軟で拡張可能な NetFlow Version 9 の活用。
- IP アカウンティング、ボーダー ゲートウェイ プロトコル (BGP) ポリシー アカウンティング、永続的キャッシュなどの多数のアカウントティング機能を置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。

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、ユーザー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トリング。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 のユーザー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、ネットワーク 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 モニターに、フローレコード、フロー エクスポート、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 エクスポートの宛先 IP アドレスなどのパラメータを変更する場合、フロー エクスポートを使用するすべてのフロー モニターに対して自動的に変更されます。同じフロー モニターを複数のフロー サンプラと組み合わせると、さまざまなインターフェイス上でさまざま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 フローモニターに割り当てられ、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 フィールドを有効にします。

- **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 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**— Cisco TrustSec 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、ユーザー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、一般的なユーザー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。



- (注) 事前定義されたレコードは、Cisco Catalyst 9000 シリーズスイッチの通常の Flexible NetFlow ではサポートされません。

ユーザー定義レコード

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 フローモニター キャッシュ用の独自のレコードを定義できます。Flexible NetFlow フローモニター キャッシュに対して独自のレコードを定義する場合、ユーザー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。

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 11: match パラメータ

コマンド	目的
match datalink {dot1q ethertype mac vlan }	<p>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。</p> <ul style="list-style-type: none"> • 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	<p>フローを識別するフィールドとの一致を指定します。</p>
match interface {input output}	<p>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。</p> <ul style="list-style-type: none"> • 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。

コマンド	目的
match ipv4 { destination protocol source tos ttl version }	<p>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。</p> <ul style="list-style-type: none"> • 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。
match ipv6 { destination hop-limit protocol source traffic-class version }	<p>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。</p> <ul style="list-style-type: none"> • 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。

コマンド	目的
match transport { destination-port igmp icmp source-port }	<p>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。</p> <ul style="list-style-type: none"> • 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	<p>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。</p> <ul style="list-style-type: none"> • 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 12: collect パラメータ

コマンド	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)。

コマンド	目的
collect transport tcp flags	<p>次の転送 TCP フラグを収集します。</p> <ul style="list-style-type: none"> • 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 <p>(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。</p>
collect counter bytes	フローの確認されたバイト数を非キー フィールドとして設定し、フローの合計バイト数を収集します。
collect counter packets	フローで確認されるパケット数を非キーフィールドとして設定し、フローから合計パケット数を収集します。

フロー エクスポート

フローエクスポートでは、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。フローエクスポートは、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。フローエクスポートは、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トを作成して、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トを作成し、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。

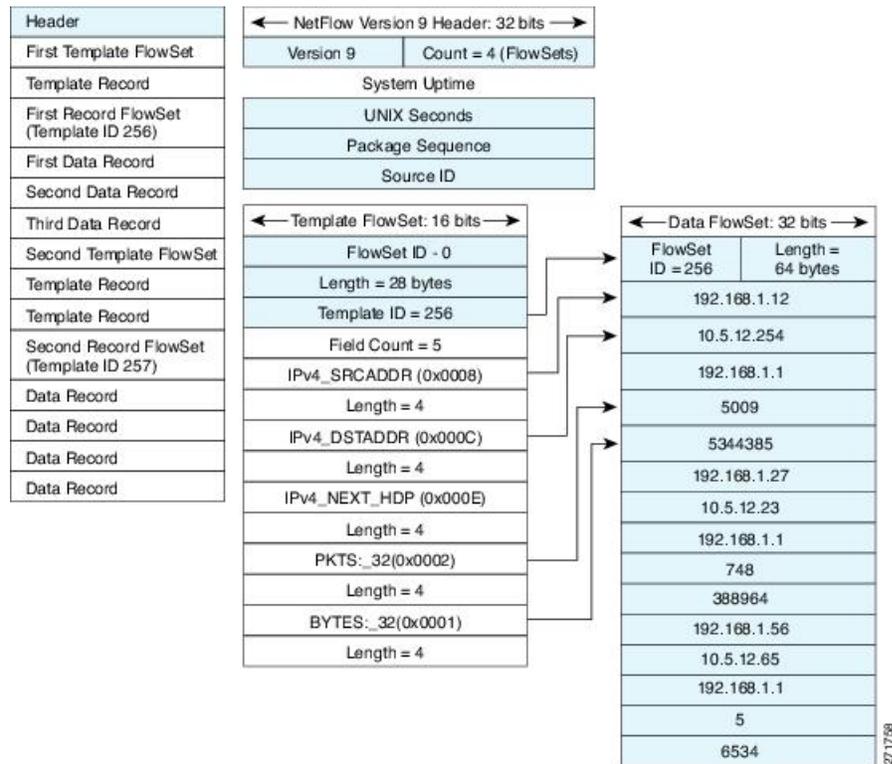
バージョン9のエクスポートフォーマットは、パケットヘッダーとそれに続く1つ以上のテンプレートフローセットまたはデータフローセットで構成されています。テンプレートフローセットでは、将来のデータフローセットに表示されるフィールドの説明が提供されます。このようなデータフローセットは、後で同じエクスポートパケットまたは後続のエクスポートパケットで発生する可能性があります。テンプレートフローセットおよびデータフローセットは、次の図に示すように、単一のエクスポートパケットに混在させることができます。

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は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、ユーザー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



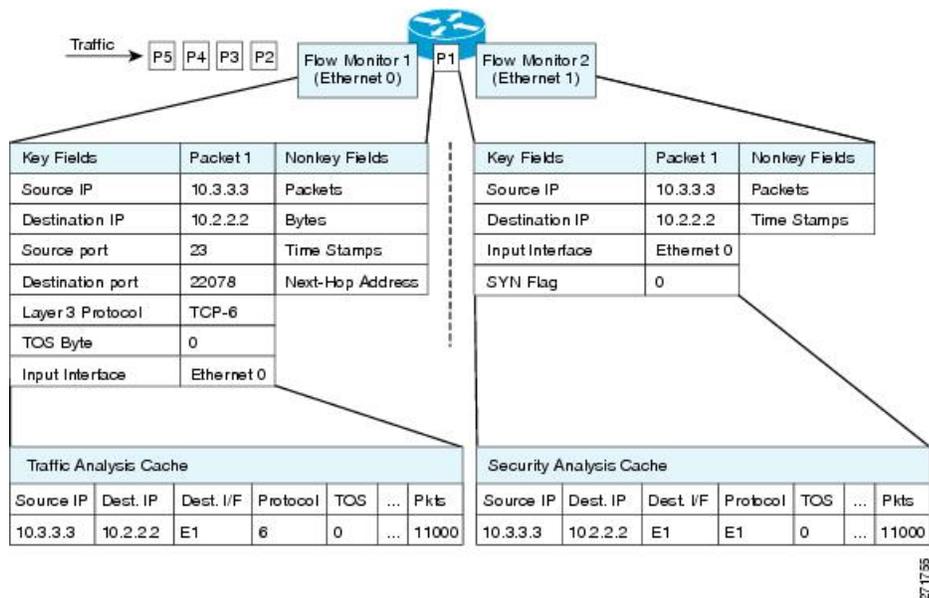
フロー モニター

フロー モニターは 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 モニター キャッシュに追加されます。

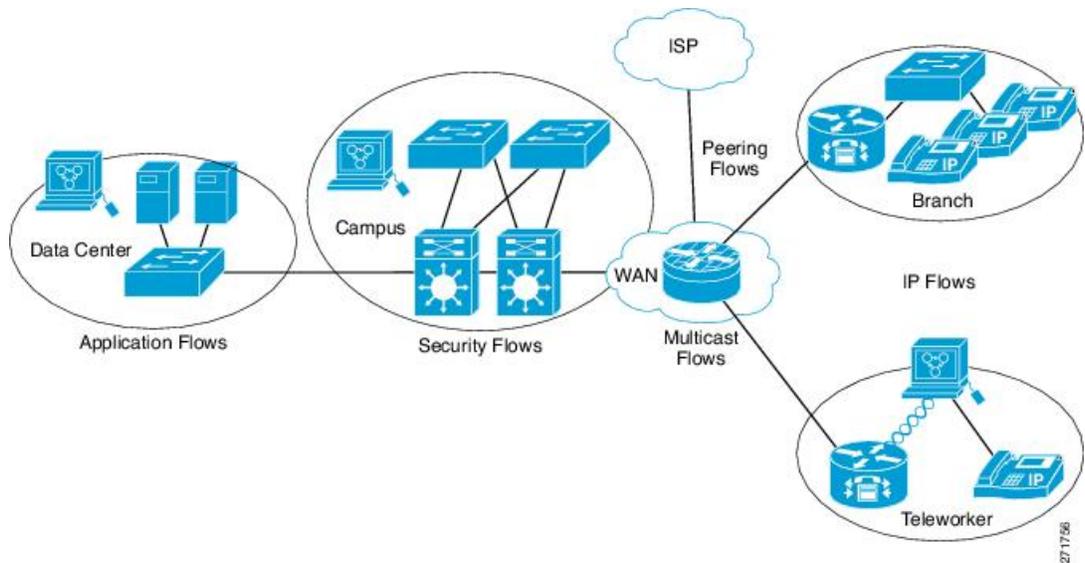
Flexible NetFlow は、同じトラフィックのさまざまなタイプの分析を実行するために使用できます。下の図では、入力インターフェイス上の標準トラフィック分析のために設計されたレコードと、出力インターフェイス上のセキュリティ分析のために設計されたレコードを使用してパケット 1 が分析されます。

図 12: 2つのフロー モニター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 モニター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 モニターの複雑な使用例



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トによってエクスポートされます。

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削減する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ーに適用すると、フローモニター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ーを実行するためのオーバーヘッド負荷が低下します。フローモニターで分析されるパケット数が減少すると、フローモニターのキャッシュに格納される情報の精度が、それに応じて低下します。

ip flow monitor コマンドを使用してインターフェイスに適用される場合、サンプラーはフローモニターと組み合わせて使用されます。

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	注意
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	注意
インターフェイス入力	あり	—	あり	—	あり	—	<p>フロー モニターを入力方向に適用する場合：</p> <ul style="list-style-type: none"> • 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	—	あり	—	あり	—	あり	<p>フロー モニターを出力方向に適用する場合：</p> <ul style="list-style-type: none"> • 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	注意
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	注意
フロー方向	対応	対応	対応	対応	対応	対応	
Ethertype	対応	対応	—	—	—	—	
VLAN 入力	あり	—	あり	—	あり	—	スイッチポートでのみサポートされています。
VLAN 出力	—	あり	—	あり	—	あり	スイッチポートでのみサポートされています。
dot1q VLAN 入力	あり	—	あり	—	あり	—	スイッチポートでのみサポートされています。
dot1q VLAN 出力	—	あり	—	あり	—	あり	スイッチポートでのみサポートされています。
dot1q 優先度	対応	対応	対応	対応	対応	対応	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	対応	対応	対応	対応	対応	対応	

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPv4 Out	IPv6 In	IPv6 Out	注意
MAC 送信元アドレス出力	—	—	—	—	—	—	
MAC 宛先アドレス入力	あり	—	あり	—	あり	—	
MAC 送信先アドレス出力	—	あり	—	あり	—	あり	
IPv4 バージョン	—	—	対応	対応	対応	対応	
IPv4 TOS	—	—	対応	対応	対応	対応	
IPv4 プロトコル	—	—	対応	対応	対応	対応	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	—	—	対応	対応	対応	対応	
IPv4 送信元アドレス	—	—	対応	対応	—	—	
IPv4 宛先アドレス	—	—	対応	対応	—	—	
ICMP IPv4 タイプ	—	—	対応	対応	—	—	

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPv4 Out	IPv6 In	IPv6 Out	注意
ICMP IPv4 コード	—	—	対応	対応	—	—	
IGMP タイプ	—	—	対応	対応	—	—	
フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPv4 Out	IPv6 In	IPv6 Out	注意
Key フィールド (続き)							
IPv6 バージョン	—	—	対応	対応	対応	対応	IP バージョンと同じです。
IPv6 プロトコル	—	—	対応	対応	対応	対応	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	—	—	—	—	対応	対応	
IPv6 宛先アドレス	—	—	—	—	対応	対応	

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPv4 Out	IPv6 In	IPv6 Out	注意
IPv6 トラフィッククラス	—	—	対応	対応	対応	対応	IP TOS と同じです。
IPv6 ホップリミット	—	—	対応	対応	対応	対応	IP TTL と同じです。
ICMP IPv6 タイプ	—	—	—	—	対応	対応	
ICMP IPv6 コード	—	—	—	—	対応	対応	
source-port	—	—	対応	対応	対応	対応	
dest-port	—	—	対応	対応	対応	対応	
フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPv4 Out	IPv6 In	IPv6 Out	注意
Collect フィールド							
バイト長	対応	対応	対応	対応	対応	対応	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。
パケット長	対応	対応	対応	対応	対応	対応	

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPv4 Out	IPv6 In	IPv6 Out	注意
Timestamp absolute first	対応	対応	対応	対応	対応	対応	
Timestamp absolute last	対応	対応	対応	対応	対応	対応	
TCP フラグ	対応	対応	対応	対応	対応	対応	すべてのフラグを収集します。
Bytes layer2 long	対応	対応	対応	対応	対応	対応	

デフォルト設定

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

表 13: 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 : 入力 VRF サポートの概要

Flexible NetFlow : 入力 VRF サポート機能では、key フィールドとして Virtual Routing and Forwarding (VRF) ID を収集するフローレコードがある入力フローモニターを適用して、デバイスで着信パケットから VRF ID を収集できるようにします。

自律システム番号

自律システム番号スペースは、4,294,967,296 個の一意の値を持つ 32 ビットのフィールドで、インターネットのパブリックドメイン間ルーティングシステムをサポートするために使用できます。

自律システム番号 (AS 番号) は、主にボーダー ゲートウェイ プロトコルで使用される IANA によって割り当てられる特別な番号です。一意のルーティングポリシーを持つ単一の技術管理下にあるネットワーク、またはパブリックインターネットにマルチホーム接続されているネットワークを一意に識別します。この自律システム番号は、ピアリングポイントのインターネット

トサービスプロバイダとインターネットエクスチェンジ (IX) の間で、BGP およびピアをインターネットサービスプロバイダと実行するために必要です。AS 番号はグローバルに一意である必要があります。これにより、BGP が検出してルーティングできる一意の場所から IP アドレスブロックが送信されるようになります。BGP は、プレフィックスと自律システムパス (AS パス) を使用して、プレフィックスが存在する宛先への最短パスを決定します。

NetFlow V9 および IPFIX エクスポートタイプは、32 ビット AS 番号をサポートします。NetFlow V5 は、固定 16 ビットの送信元および宛先 AS 形式に従うため、この 32 AS フィールドをサポートしません。

NetFlow では、次の BGP パラメータをエクスポートできます。

- BGP 送信元起源またはピア AS 番号
- BGP 宛先起源またはピア AS 番号

設定

AS 番号システムを設定するには、次のコマンドを使用します。

```
[no] collect routing { destination | source } as [[4-octet] peer] [4-octet]
```

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トを作成し、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トに基づいて、フロー モニターを作成します。
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 モニターを適用します。

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。

手順の概要

1. **enable**
2. **configure terminal**
3. **flow record** *record-name*
4. **description** *description*
5. **match** {ip | ipv6} {destination | source} address
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。
7. **match flow cts** {source | destination} group-tag
- 8.
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。
10. **end**
11. **show flow record** *record-name*
12. **show running-config flow record** *record-name*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none">パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	flow record <i>record-name</i> 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。 <ul style="list-style-type: none">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 <i>description</i> 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。

	コマンドまたはアクション	目的
ステップ 5	match {ip ipv6} {destination source} address 例 : <pre>Device(config-flow-record)# match ipv4 destination address</pre>	(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。 match ipv4 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。	—
ステップ 7	match flow cts {source destination} group-tag 例 : <pre>Device(config-flow-record)# match flow cts source group-tag</pre> <pre>Device(config-flow-record)# match flow cts destination group-tag</pre>	(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。 match ipv4/ipv6 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。 (注) <ul style="list-style-type: none"> • 入力 : <ul style="list-style-type: none"> • 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。 • 出力 : <ul style="list-style-type: none"> • 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 の強制が無効化されている場合、DGT は 0 になります。

	コマンドまたはアクション	目的
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。
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	show flow record 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	show running-config flow record 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 エクスポートの作成

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) フローエクスポートごとに、1つの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 エクスポートを設定してフロー モニター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。

手順の概要

1. **configure terminal**
2. **flow exporter name**
3. **description string**
4. **destination {ipv4-address}**
5. **dscp value**
6. **source { | }**

7. **transport udp** *number*
8. **ttl** *seconds*
9. **export-protocol** {*netflow-v9*}
10. **end**
11. **show flow exporter** [*name record-name*]
12. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： デバイス# 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 exporter <i>name</i> 例： デバイス (config)# flow exporter ExportTest	フロー エクスポートを作成し、フロー エクスポート 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 <i>string</i> 例： デバイス (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 { <i>ipv4-address</i> } 例： デバイス (config-flow-exporter)# destination 192.0.2.1 (IPv4 destination)	このエクスポートに 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 <i>value</i> 例： デバイス (config-flow-exporter)# dscp 0	(任意) DSCP (DiffServ コードポイント) 値を指定します。範囲は 0 ~ 63 です。デフォルトは 0 です。
ステップ 6	source { <i>}</i> 例： デバイス (config-flow-exporter)# source	(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。

	コマンドまたはアクション	目的
	<code>gigabitEthernet1/0/1</code>	\
ステップ 7	transport udp number 例 : デバイス (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 seconds 例 : デバイス (config-flow-exporter) # ttl 210	(任意) エクスポート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。
ステップ 9	export-protocol {netflow-v9} 例 : デバイス (config-flow-exporter) # export-protocol netflow-v9	エクスポートで使用する 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 : デバイス (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 : デバイス # show flow exporter ExportTest	(任意) NetFlow のフローエクスポート情報を表示します。
ステップ 12	copy running-config startup-config 例 : デバイス # 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 エクスポートに基づいて、フロー モニターを定義します。

カスタマイズしたフロー モニターの作成

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

各フローモニターには、専用のキャッシュが割り当てられています。フローモニターごとに、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、またはユーザー定義にすることができます。上級のユーザーであれば **flow record** 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。



- (注) フレキシブル NetFlow がレイヤ 3 ポート チャネル インターフェイスで設定されている場合、最後に適用されたフローモニター設定が、そのポートチャネルのすべてのメンバに対して有効になります。したがって、L3 ポート チャネル インターフェイスのすべてのメンバで、フローモニター設定を同じにすることを推奨します。

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 エクスポートをフロー モニターに追加する場合は、このタスクを完了する前にエクスポートを作成する必要があります。



- (注) フローモニターで **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、すべてのインターフェイスから適用済みのフローモニターを削除する必要があります。

手順の概要

1. **enable**
2. **configure terminal**
3. **flow monitor** *monitor-name*
4. **description** *description*
5. **record** {*record-name* | **netflow-original** | **netflow** {**ipv4** | **ipv6**} *record* [**peer**] }
6. **cache** {**timeout** {**active** | **inactive** | **update** | **rate-limit**} *seconds* | **type normal** }
7. 必要に応じてステップ 6 を繰り返して、このフローモニターのキャッシュパラメータの変更を完了します。
8. **statistics packet** **protocol**
9. **statistics packet** **size**
10. **exporter** *exporter-name*
11. **end**
12. **show flow monitor** [[**name**] *monitor-name* [**cache** [**format** {**csv** | **record** | **table**}]] [**statistics**]]
13. **show running-config flow monitor** *monitor-name*
14. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	flow monitor <i>monitor-name</i> 例： Device(config)# flow monitor FLOW-MONITOR-1	フロー モニターを作成し、Flexible NetFlow フロー モニター コンフィギュレーション モードを開始します。 <ul style="list-style-type: none"> このコマンドでは、既存のフロー モニターを変更することもできます。
ステップ 4	description <i>description</i> 例： Device(config-flow-monitor)# description Used for basic ipv4 traffic analysis	(任意) フロー モニターの説明を作成します。
ステップ 5	record {<i>record-name</i> netflow-original netflow {<i>ipv4</i> <i>ipv6</i>} record [<i>peer</i>]} 例： Device(config-flow-monitor)# record FLOW-RECORD-1	フロー モニターのレコードを指定します。
ステップ 6	cache {<i>timeout</i> {<i>active</i> <i>inactive</i> <i>update</i> <i>rate-limit</i>} <i>seconds</i> <i>type normal</i> } 例： Device(config-flow-monitor)# cache type normal Device(config-flow-monitor)# cache timeout active	(任意) フロー モニター キャッシュ パラメータ (タイムアウト値、キャッシュタイプなど) を変更します。指定したフロー モニターとフロー キャッシュを関連付けます。
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 モニターのキャッシュ パラメータの変更を完了します。	—
ステップ 8	statistics packet <i>protocol</i> 例： Device(config-flow-monitor)# statistics packet protocol	(任意) Flexible NetFlow モニター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。

	コマンドまたはアクション	目的
ステップ 9	statistics packet size 例： Device(config-flow-monitor)# statistics packet size	(任意) Flexible NetFlow モニター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ト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 モニター 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	show flow monitor [[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 モニターのステータスおよび統計情報が表示されます。
ステップ 13	show running-config flow monitor monitor-name 例： Device# show running-config flow monitor FLOW_MONITOR-1	(任意) 指定したフロー モニターの設定が表示されます。
ステップ 14	copy running-config startup-config 例： Device# copy running-config startup-config	(任意) コンフィギュレーション ファイルに設定を保存します。

フローサンプラーの作成

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。

手順の概要

1. **enable**
2. **configure terminal**
3. **sampler sampler-name**
4. **description description**

5. **mode** {random} 1 out-of window-size
6. **exit**
7. **interface** type number
8. {ip | ipv6} **flow monitor** monitor-name [[**sampler**] sampler-name] {input | output}
9. **end**
10. **show sampler** sampler-name

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	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。 <ul style="list-style-type: none"> 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 description 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。 <ul style="list-style-type: none"> window-size 引数の範囲は、0 ~ 1024 です。
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。

	コマンドまたはアクション	目的
ステップ 8	<p>{ip ipv6} flow monitor <i>monitor-name</i> [[<i>sampler</i> <i>sampler-name</i>] {input output}]</p> <p>例 :</p> <pre>Device(config-if)# ip flow monitor FLOW-MONITOR-1 sampler SAMPLER-1 input</pre>	作成したフロー モニターおよびフロー 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	<p>end</p> <p>例 :</p> <pre>Device(config-if)# end</pre>	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	<p>show sampler <i>sampler-name</i></p> <p>例 :</p> <pre>Device# show sampler SAMPLER-1</pre>	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 モニター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。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface *type***
4. **{ip flow monitor | ipv6 flow monitor | datalink flow monitor} *name* [*sampler name*] {input | output}**
5. **end**
6. **show flow interface [*interface-type number*]**
7. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <pre>Device> enable</pre>	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> • パスワードを入力します（要求された場合）。
ステップ 2	<p>configure terminal</p> <p>例 :</p> <pre>Device(config)# configure terminal</pre>	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	interface type 例 : Device(config)# interface GigabitEthernet1/0/1	インターフェイス コンフィギュレーション モードを開始し、インターフェイスを設定します。 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とメンバポートでサポートされますが、両方に対して同時にサポートされることはありません。
ステップ 4	{ip flow monitor ipv6 flow monitor datalink flow monitor} name [sampler name] {input output} 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、IPv6、データリンクフローモニター、およびオプションのサンプラーを関連付けます。 ip flow monitor – Flexible NetFlow で IPv4 トラフィックを監視できます。 ipv6 flow monitor – Flexible NetFlow で IPv6 トラフィックを監視できます。 datalink flow monitor – Flexible NetFlow で非 IP のトラフィックを監視できます。 (注) 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ーを関連付けることができます。
ステップ 5	end 例 : Device(config-flow-monitor)# end	特権 EXEC モードに戻ります。
ステップ 6	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 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。

VLAN 上でのブリッジ型 NetFlow の設定

フロー モニターおよびオプションのサンプラーを VLAN に適用できます。

手順の概要

1. **configure terminal**
2. **vlan [configuration] vlan-id**
3. **ip flow monitor monitor name [sampler sampler name] { input }**
4. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： デバイス# 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： デバイス (config)# vlan configuration 30 デバイス (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： デバイス (config-vlan-config)# ip flow monitor MonitorTest input	入力パケットに対応する VLAN に、フローモニター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： デバイス# 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。

手順の概要

1. **configure terminal**
2. **flow record *name***
3. **match datalink {dot1q | ethertype | mac | vlan}**
4. **end**
5. **show flow record [*name*]**
6. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： デバイス# 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 <i>name</i> 例： デバイス (config)# flow record L2_record デバイス (config-flow-record)#	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。
ステップ 3	match datalink {dot1q ethertype mac vlan} 例： デバイス (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： デバイス (config-flow-record)# end	特権 EXEC モードに戻ります。
ステップ 5	show flow record [<i>name</i>] 例： デバイス# 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： デバイス# 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 14: Flexible NetFlow のモニタリング コマンド

コマンド	目的
show flow exporter [broker export-ids name name statistics templates]	NetFlow のフロー エクスポート情報と統計情報を表示します。
show flow exporter [name <i>exporter-name</i>]	NetFlow のフロー エクスポート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。
show flow monitor [name <i>exporter-name</i>]	NetFlow のフロー モニター情報と統計情報を表示します。
show flow monitor statistics	フロー モニター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ーのキャッシュの内容を表示します。
show flow record [name <i>record-name</i>]	NetFlow のフローレコード情報を表示します。
show sampler [broker name name]	NetFlow サンプラーに関する情報を表示します。

設定例 Flexible NetFlow

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。

```

デバイス# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.

デバイス(config)# flow export export1
デバイス(config-flow-exporter)# destination 10.0.101.254
デバイス(config-flow-exporter)# transport udp 2055
デバイス(config-flow-exporter)# exit
デバイス(config)# flow record record1
デバイス(config-flow-record)# match ipv4 source address

```

```

デバイス(config-flow-record)# match ipv4 destination address
デバイス(config-flow-record)# match ipv4 protocol
デバイス(config-flow-record)# match transport source-port
デバイス(config-flow-record)# match transport destination-port
デバイス(config-flow-record)# match flow cts source group-tag
デバイス(config-flow-record)# match flow cts destination group-tag
デバイス(config-flow-record)# collect counter byte long
デバイス(config-flow-record)# collect counter packet long
デバイス(config-flow-record)# collect timestamp absolute first
デバイス(config-flow-record)# collect timestamp absolute last
デバイス(config-flow-record)# exit
デバイス(config)# flow monitor monitor1
デバイス(config-flow-monitor)# record record1
デバイス(config-flow-monitor)# exporter export1
デバイス(config-flow-monitor)# exit
デバイス(config)# interface tenGigabitEthernet 1/0/1
デバイス(config-if)# ip flow monitor monitor1 input
デバイス(config-if)# end

```

例：IPv4 入カトラフィックのモニタリング

次の例は、IPv4 入カトラフィックをモニターする方法を示しています（intg1/0/11 は、intg1/0/36 および intg3/0/11 にトラフィックを送信します）。

```

デバイス# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
デバイス(config)# flow record fr-1
デバイス(config-flow-record)# match ipv4 source address
デバイス(config-flow-record)# match ipv4 destination address
デバイス(config-flow-record)# match interface input
デバイス(config-flow-record)# collect counter bytes long
デバイス(config-flow-record)# collect counter packets long
デバイス(config-flow-record)# collect timestamp absolute first
デバイス(config-flow-record)# collect timestamp absolute last
デバイス(config-flow-record)# collect counter bytes layer2 long
デバイス(config-flow-record)# exit

デバイス(config)# flow exporter fe-ipfix6
デバイス(config-flow-exporter)# destination 2001:0:0:24::10
デバイス(config-flow-exporter)# source Vlan106
デバイス(config-flow-exporter)# transport udp 4739
デバイス(config-flow-exporter)# export-protocol ipfix
デバイス(config-flow-exporter)# template data timeout 240
デバイス(config-flow-exporter)# exit

デバイス(config)# flow exporter fe-ipfix
デバイス(config-flow-exporter)# description IPFIX format collector 100.0.0.80
デバイス(config-flow-exporter)# destination 100.0.0.80
デバイス(config-flow-exporter)# dscp 30
デバイス(config-flow-exporter)# ttl 210

```

例 : IPv4 出カトラフィックのモニタリング

```

デバイス(config-flow-exporter)# transport udp 4739
デバイス(config-flow-exporter)# export-protocol ipfix
デバイス(config-flow-exporter)# template data timeout 240
デバイス(config-flow-exporter)# exit

デバイス(config)# flow exporter fe-1
デバイス(config-flow-exporter)# destination 10.5.120.16
デバイス(config-flow-exporter)# source Vlan105
デバイス(config-flow-exporter)# dscp 32
デバイス(config-flow-exporter)# ttl 200
デバイス(config-flow-exporter)# transport udp 2055

デバイス(config-flow-exporter)# template data timeout 240
デバイス(config-flow-exporter)# exit

デバイス(config)# flow monitor fm-1
デバイス(config-flow-monitor)# exporter fe-ipfix6
デバイス(config-flow-monitor)# exporter fe-ipfix
デバイス(config-flow-monitor)# exporter fe-1
デバイス(config-flow-monitor)# cache timeout inactive 60
デバイス(config-flow-monitor)# cache timeout active 180
デバイス(config-flow-monitor)# record fr-1
デバイス(config-flow-monitor)# end

デバイス# show running-config interface g1/0/11
デバイス# show running-config interface g1/0/36
デバイス# show running-config interface g3/0/11
デバイス# show flow monitor fm-1 cache format table

```

例 : IPv4 出カトラフィックのモニタリング

```

デバイス# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
デバイス(config)# flow record fr-1 out
デバイス(config-flow-record)# match ipv4 source address
デバイス(config-flow-record)# match ipv4 destination address
デバイス(config-flow-record)# match interface output
デバイス(config-flow-record)# collect counter bytes long
デバイス(config-flow-record)# collect counter packets long
デバイス(config-flow-record)# collect timestamp absolute first
デバイス(config-flow-record)# collect timestamp absolute last
デバイス(config-flow-record)# exit

デバイス(config)# flow exporter fe-1
デバイス(config-flow-exporter)# destination 10.5.120.16
デバイス(config-flow-exporter)# source Vlan105
デバイス(config-flow-exporter)# dscp 32
デバイス(config-flow-exporter)# ttl 200
デバイス(config-flow-exporter)# transport udp 2055
デバイス(config-flow-exporter)# template data timeout 240

```

```
デバイス(config-flow-exporter)# exit

デバイス(config)# flow exporter fe-ipfix6
デバイス(config-flow-exporter)# destination 2001:0:0:24::10
デバイス(config-flow-exporter)# source Vlan106
デバイス(config-flow-exporter)# transport udp 4739
デバイス(config-flow-exporter)# export-protocol ipfix
デバイス(config-flow-exporter)# template data timeout 240
デバイス(config-flow-exporter)# exit

デバイス(config)# flow exporter fe-ipfix
デバイス(config-flow-exporter)# description IPFIX format collector 100.0.0.80
デバイス(config-flow-exporter)# destination 100.0.0.80
デバイス(config-flow-exporter)# dscp 30
デバイス(config-flow-exporter)# ttl 210
デバイス(config-flow-exporter)# transport udp 4739
デバイス(config-flow-exporter)# export-protocol ipfix
デバイス(config-flow-exporter)# template data timeout 240
デバイス(config-flow-exporter)# exit

デバイス(config)# flow monitor fm-1-output
デバイス(config-flow-monitor)# exporter fe-1
デバイス(config-flow-monitor)# exporter fe-ipfix6
デバイス(config-flow-monitor)# exporter fe-ipfix
デバイス(config-flow-monitor)# cache timeout inactive 50
デバイス(config-flow-monitor)# cache timeout active 120
デバイス(config-flow-monitor)# record fr-1-out
デバイス(config-flow-monitor)# end

デバイス# show flow monitor fm-1-output cache format table
```

例：入力 VRF サポート用の Flexible NetFlow の設定

次の例では、VRF ID を key フィールドとして収集するフローレコードを持つ入力フローモニターを適用することで、デバイスの着信パケットからの VRF ID の収集を設定します。

```
Device> enable
Device# configure terminal
Device(config)# flow record rm_1
Device(config-flow-record)# match routing vrf input
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# collect interface input
Device(config-flow-record)# collect interface output
Device(config-flow-record)# collect counter packets
Device(config-flow-record)# exit

Device(config)# flow monitor mm_1
Device(config-flow-record)# record rm_1
Device(config-flow-record)# exit

Device(config)# interface GigabitEthernet 1/0/1
Device(config-if)# ip vrf forwarding green
Device(config-if)# ip address 172.16.2.2 255.255.255.252
```

```
Device(config-if)# ip flow monitor mm_1 input  
Device(config-if)# end
```

Flexible NetFlow の機能情報

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



第 10 章

暗号化トラフィック分析の設定

- [暗号化トラフィック分析の制約事項 \(231 ページ\)](#)
- [暗号化トラフィック分析について \(231 ページ\)](#)
- [暗号化トラフィック分析の設定方法 \(233 ページ\)](#)
- [暗号化トラフィック分析の設定例 \(235 ページ\)](#)
- [その他の参考資料 \(236 ページ\)](#)
- [暗号化トラフィック分析の機能履歴と情報 \(237 ページ\)](#)

暗号化トラフィック分析の制約事項

- ETA は、アクセスポートと SDA 導入環境のワイヤレス VLAN でのみサポートされています。管理、トランク、ポートチャネル、SVI、ループバックの各インターフェイスではサポートされていません。
- ETA および Cisco Application Visibility and Control (AVC) の機能は、同じインターフェイスには適用できません。
- インターフェイスを ETA が有効の Flexible NetFlow モニターとして設定する必要がある場合は、そのインターフェイスで Flexible NetFlow を設定する前に ETA をグローバルに設定する必要があります。
- ETA と送信 (Tx) スイッチドポートアナライザ (SPAN) は、同じインターフェイスではサポートされません。

暗号化トラフィック分析について

ここでは、暗号化トラフィック分析について説明します。

概要

暗号化トラフィック分析 (ETA) はアプリケーションに機械学習を使用して、マルウェア分析や暗号監査などのフロー特性を特定します。

フローモニターに関連付けられたフローレコードに基づき、派生収集フィールドを使用して NetFlow レコードを表示するエクスポート テンプレートをスイッチが作成します。

ETA は、設定のエクスポート用の複数のテンプレートをサポートします。ETA 属性ごとに 1 つのテンプレートがあり、ETA は各テンプレートの個々の属性の詳細をエクスポート時に送信します。パケット長と時間のシーケンス (SPLT) および初期データ パッケージ (IDP) は、別個のテンプレートに格納され、NetFlow レコードの生成に使用されます。これらの NetFlow レコードの両方が、指定のアプリケーションフローに送信されます。

これらのテンプレートは、データの準備が整うたびに送信されます。これにより、NetFlow コレクタは正しい属性値でデータを解釈することができます。エクスポートの宛先とポートがすべてのインターフェイスに共通となり、この値がグローバル `et-analytics` コンフィギュレーション コマンドで提供されます。ETA のスケール数は 2000 フロー/秒です。

このテンプレート エクスポートは、ETA フローモニターで 1 つのエクスポート IP アドレスのみをサポートします。複数テンプレートのエクスポートは、以降の NetFlow v9 バージョンでサポートされます。

Flexible NetFlow と ETA の設定

Flexible NetFlow モニターは、他のフローモニターの一一致フィールドに同じ 5 タプルがある場合にのみ、ETA が有効になっている同じインターフェイスに適用できます。そのため、限定された一連の一一致属性のみを持つ Flexible NetFlow がサポートされます。Flexible NetFlow モニターと ETA に対応するフロー モニターを同じ物理インターフェイスに適用すると、ソフトウェアは論理的にマージし、収集フィールドとエクスポートの詳細を多重化します。



(注) 2 つのフローモニターを同じインターフェイスに適用し、Flexible NetFlow 設定に 5 タプルの一一致がある場合は、Flexible NetFlow モニターを最初に設定してから、`et-analytics` コマンドを設定する必要があります。

Flexible NetFlow 設定に一連の異なる一一致フィールドがある場合はエラーが表示されます。これは、フローモニターに必要なのが 5 タプルの一一致フィールドのみであるためです。

機能を無効にする場合は、最初に `et-analytics` を無効にしてから、Flexible Netflow モニターを無効にする必要があります。

非アクティブ タイマーとエクスポート

次の 2 つの条件のいずれかが満たされた場合にのみ ETA 情報がエクスポートされます。

- 必要なデータが計算され、ETA コレクタによって必要な数のパケットが確認された場合。
- 確立されたフローが非アクティブタイムアウトとして設定された期間にわたってアイドル状態のままになっていて、一部のデータがエクスポートされる場合。



- (注) 設定した非アクティブタイマーはグローバルに適用されます。異なるポートを異なる値で設定することはできません。

暗号化トラフィック分析の設定方法

ここでは、暗号化トラフィック分析の設定方法について説明します。

エクスポート IP とポートの設定

IP アドレスとポートを設定するには、次の手順に従います。

手順の概要

1. **enable**
2. **configure terminal**
3. **et-analytics**
4. **ip flow-export destination** *destination_ip_address port*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# config 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	et-analytics 例： Device(config)# et-analytics	グローバル et-analytics 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	ip flow-export destination <i>destination_ip_address port</i> 例： Device(config-et-analytics)# ip flow-export destination 10.1.1.1 2055	グローバル コレクタの宛先 IP アドレスとポートを設定します。

非アクティブ タイマー値の設定

非アクティブ タイマー値を設定するには、次の手順に従います。

手順の概要

1. **enable**
2. **configure terminal**
3. **et-analytics**
4. **inactive time** *time in seconds*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# config t	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	et-analytics 例： Device(config)# et-analytics	グローバル et-analytics 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	inactive time <i>time in seconds</i> 例： Device(config-et-analytics)# inactive time 10	非アクティブタイマー値を設定します。範囲は1～604800 で、デフォルトは 15 秒です。

暗号化トラフィック分析の有効化

脅威の可視性をイネーブルにするには、次の手順に従います。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. **et-analytics enable**

手順の詳細

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

	コマンドまたはアクション	目的
ステップ 2	configure terminal 例： Device# config t	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 gi1/0/2	インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	et-analytics enable 例： Device(config-if)# et-analytics enable	特定のインターフェイス上で et-analytics をイネーブルにします。

暗号化トラフィック分析の設定例

ここでは、暗号化トラフィック分析の設定例を示します。

例：エクスポート IP とポートの設定

次に、フローエクスポートの宛先 IP アドレスを 10.1.1.1 に、ポートを 2055 に設定する例を示します。

```
Device#config terminal
Device (config)#et-analytics
Device (config-et-analytics)#ip flow-export destination 10.1.1.1 2055
```

例：非アクティブ タイマーの設定

次に、非アクティブ タイマーを 10 秒に設定する例を示します。

```
Device#config terminal
Device (config)#et-analytics
Device (config-et-analytics)#inactive time 10
```

例：et-analytics の有効化

次に、インターフェイス GigabitEthernet1/0/2 で et-analytics を有効にする例を示します。

```
Device#config terminal
Device (config)#interface gi1/0/2
Device (config-if)#et-analytics enable
```

例：et-analytics 設定の確認

次に、グローバル et-analytics コンフィギュレーションを表示する例を示します。

```
Device#show platform software et-analytics global
ET-Analytics Global state
=====
All Interfaces : Off
IP Flow-record Destination: 172.26.202.123 : 2055
Inactive timer: 10

ET-Analytics interfaces
GigabitEthernet1/0/26
GigabitEthernet1/0/36
```

```
ET-Analytics VLANs
```

次に、インターフェイス `et-analytics` コンフィギュレーションを表示する例を示します。

```
Device#show platform software et-analytics interface
ET-Analytics interfaces
GigabitEthernet1/0/3
```

次に、ETA モニター キャッシュ出力を表示する例を示します。

```
Device#show flow monitor etta-mon cache
Cache type: Normal (Platform cache)
Cache size: 10000
Current entries: 4

Flows added: 6
Flows aged: 2
- Inactive timeout ( 15 secs) 2

IPV4 DESTINATION ADDRESS: 15.15.15.35
IPV4 SOURCE ADDRESS: 72.163.128.140
IP PROTOCOL: 17
TRNS SOURCE PORT: 53
TRNS DESTINATION PORT: 12032
counter bytes long: 128
counter packets long: 1
timestamp abs first: 06:23:24.799
timestamp abs last: 06:23:24.799
interface input: Null
interface output: Null
```

その他の参考資料

関連項目	マニュアルタイトル
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。	Cisco IOS XE Everest 16.6.x (Catalyst 9300 スイッチ) コマンドリファレンス
Flexible NetFlow	Cisco IOS XE Everest 16.6.x (Catalyst 9300 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

暗号化トラフィック分析の機能履歴と情報

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だけを示しています。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。

翻訳について

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