



Cisco IOS XE Bengaluru 17.6.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

初版：2021年7月30日

最終更新：2023年7月24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/>

【注意】 シスコ製品をご使用になる前に、安全上の注意（www.cisco.com/jp/go/safety_warning/）をご確認ください。本書は、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。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2020 Cisco Systems, Inc. All rights reserved.



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

プラットフォーム タイプ フィルタの設定 15

クラスマップのプラットフォーム タイプ フィルタの設定 15

パラメータマップのプラットフォーム タイプ フィルタの設定 16

クラスマップのデバイスタイプフィルタの設定 17

Autoconf の設定例 18

例：エンドデバイスへの組み込みテンプレートの適用 18

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 18

例：ASP マクロから Autoconf への移行 19

例：プラットフォーム タイプ フィルタの設定 19

Autoconf のその他の参考資料 20

Autoconf の機能履歴 20

第 2 章	Cisco プラグ アンド プレイの設定 23
	Cisco プラグ アンド プレイの設定 23
	シスコプラグアンドプレイの機能履歴 23

第 3 章	Cisco Discovery Protocol の設定 25
	Cisco Discovery Protocol について 25
	Cisco Discovery Protocol のデフォルト設定 25
	Cisco Discovery Protocol の概要 26
	Cisco Discovery Protocol の設定方法 26
	Cisco Discovery Protocol の特性の設定 26
	Cisco Discovery Protocol のディセーブル化 28
	Cisco Discovery Protocol の有効化 29
	インターフェイス上で Cisco Discovery Protocol をディセーブルにします。 30
	インターフェイス上での Cisco Discovery Protocol のイネーブル化 32
	Cisco Discovery Protocol のモニタリングとメンテナンス 33
	Cisco Discovery Protocol の機能の履歴 34

第 4 章	簡易ネットワーク管理プロトコルの設定 37
	SNMP の前提条件 37
	SNMP の制約事項 39
	SNMP に関する情報 40
	SNMP の概要 40
	SNMP マネージャ機能 40
	SNMP エージェント機能 41
	SNMP コミュニティストリング 42
	SNMP MIB 変数アクセス 42
	SNMP フラッシュ MIB 43
	SNMP 通知 43
	SNMP ifIndex MIB オブジェクト値 43
	SNMP ENTITY-MIB 識別子 44

SNMP and Syslog Over IPv6	44
SNMP のデフォルト設定	45
SNMP 設定時の注意事項	45
SNMP の設定方法	46
SNMP コミュニティストリング	46
SNMP グループおよびユーザの設定	47
SNMP UDP ポートの開閉	51
SNMP 通知	52
エージェント コンタクトおよびロケーションの設定	52
SNMP を通して使用する TFTP サーバの制限	53
SNMP エージェントのディセーブル化	55
SNMP の例	56
SNMP ステータスのモニタリング	57
簡易ネットワーク管理プロトコルの機能の履歴	58

 第 5 章

サービス レベル契約の設定	61
SLA の制約事項	61
サービスレベル契約に関する情報	61
Cisco IOS IP サービス レベル契約 (SLA)	62
Cisco IOS IP SLA でのネットワーク パフォーマンスの測定	63
IP SLA レスポンダおよび IP SLA 制御プロトコル	63
IP SLA の応答時間の計算	64
IP SLA 動作のスケジューリング	65
IP SLA 動作のしきい値のモニタリング	65
UDP ジッター	66
IP SLA 動作の設定方法	67
デフォルト設定	67
設定時の注意事項	67
IP SLA レスポンダの設定	68
IP SLA ネットワーク パフォーマンス測定の実装	70
UDP ジッター動作を使用した IP サービス レベルの分析	74

ICMP エコー動作を使用した IP サービス レベルの分析	78
IP SLA 動作のモニタリング	82
IP SLA 動作のモニタリングの例	82
その他の参考資料	83
サービスレベル契約の機能履歴	84

第 6 章

SPAN および RSPAN の設定 87

SPAN および RSPAN の前提条件	87
SPAN および RSPAN の制約事項	87
SPAN および RSPAN について	90
SPAN および RSPAN	90
ローカル SPAN	90
リモート SPAN	91
SPAN と RSPAN の概念および用語	92
SPAN および RSPAN と他の機能の相互作用	98
SPAN と RSPAN とデバイス スタック	99
フローベースの SPAN	100
SPAN および RSPAN のデフォルト設定	101
SPAN および RSPAN の設定	101
SPAN 設定時の注意事項	101
RSPAN 設定時の注意事項	101
FSPAN および FRSPAN 設定時の注意事項	102
SPAN および RSPAN の設定方法	102
ローカル SPAN セッションの作成	102
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	106
フィルタリングする VLAN の指定	108
RSPAN VLAN としての VLAN の設定	111
RSPAN 送信元セッションの作成	112
フィルタリングする VLAN の指定	115
RSPAN 宛先セッションの作成	117
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	119

FSPAN セッションの設定	121
FRSPAN セッションの設定	125
SPAN および RSPAN 動作のモニタリング	129
SPAN および RSPAN の設定例	129
例：ローカル SPAN の設定	129
例：RSPAN VLAN の作成	130
SPAN および RSPAN の機能履歴	132

 第 7 章

ERSPAN の設定 133

ERSPAN の設定の前提条件	133
ERSPAN 設定時の制約事項	133
ERSPAN の設定に関する情報	134
ERSPAN の概要	134
ERSPAN 送信元	136
ERSPAN 宛先ポート	136
SGT ベースの ERSPAN	136
ERSPAN タイムスタンプ	137
ERSPAN over MPLS VPN	137
ERSPAN の設定方法	137
ERSPAN 送信元セッションの設定 (IPv4)	137
ERSPAN 宛先セッションの設定 (IPv4)	140
ERSPAN 送信元セッションの設定 (IPv6)	142
ERSPAN 宛先セッションの設定 (IPv6)	145
ERSPAN の設定例	147
例：ERSPAN 送信元セッションの設定	148
例：ERSPAN 宛先セッションの設定	148
ERSPAN の確認	148
その他の参考資料	151
ERSPAN の機能の履歴	151

 第 8 章

パケットキャプチャの設定 153

	153
	153
	153
	154
	154
	155
	157
	157
	157
	157
	158
	159
	159
	160
	161
	161
	161
	162
	163
	166
	166
	167
	167
	168
	168
	168
	174
	176
	178
	179
	182

組み込みパケット キャプチャの実装方法	184
パケット データ キャプチャの管理	184
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	185
パケットキャプチャの設定例	186
Wireshark の設定例	186
例：.pcap ファイルからの概要出力の表示	186
例：.pcap ファイルからの詳細出力の表示	187
例：.pcap ファイルからパケット ダンプ出力の表示	188
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	189
例：.pcap ファイルにキャプチャされたパケットの数を表示	189
例：.pcap ファイルから単一パケット ダンプの表示	190
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	190
例：単純なキャプチャおよび表示	190
例：単純なキャプチャおよび保存	192
例：バッファのキャプチャの使用	194
例：出力方向のパケットの簡単なキャプチャおよび保存	200
組み込みパケット キャプチャの設定例	202
例：パケット データ キャプチャの管理	202
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	202
その他の参考資料	204
パケットキャプチャの機能履歴	205

 第 9 章

Flexible NetFlow の設定	207
Flexible NetFlow の前提条件	207
Flexible Netflow に関する制約事項	208
Flexible NetFlow に関する情報	210
Flexible NetFlow の概要	210
以前の NetFlow と Flexible NetFlow の利点	211
Flexible NetFlow のコンポーネント	212
フロー レコード	212
フロー エクスポータ	218

フロー モニター	219
フロー サンプラー	221
サポートされている Flexible NetFlow フィールド	221
デフォルト設定	228
Flexible NetFlow : 入力 VRF サポートの概要	229
Flexible NetFlow : 出力 VRF サポートの概要	229
自律システム番号	229
MPLS での入出力 Flexible NetFlow の概要	230
Flexible NetFlow の VPN ID の設定	230
Flexible NetFlow の設定方法	231
フロー レコードの作成	231
フロー エクスポートの作成	234
カスタマイズしたフロー モニターの作成	237
フロー サンプラーの作成	239
インターフェイスへのフローの適用	241
VLAN 上でのブリッジ型 NetFlow の設定	242
レイヤ 2 NetFlow の設定	243
Flexible NetFlow の監視	244
Flexible NetFlow の設定例	245
例 : フローの設定	245
例 : IPv4 入力トラフィックのモニタリング	245
例 : IPv4 出力トラフィックのモニタリング	246
例 : 入力 VRF サポート用の Flexible NetFlow の設定	247
例 : 出力 VRF サポート用の Flexible NetFlow の設定	248
Flexible NetFlow の機能履歴	248

第 10 章

暗号化トラフィック分析の設定	251
暗号化トラフィック分析の制約事項	251
暗号化トラフィック分析について	251
概要	251
暗号化トラフィック分析のエクスポート	252

暗号化トラフィック分析の設定方法	252
エクスポート IP とポートの設定	252
Active-Timeout 値の設定	253
非アクティブ タイマー値の設定	253
暗号化トラフィック分析の有効化	254
暗号化トラフィック分析の設定例	255
例：エクスポート IP とポートの設定	255
例：Active-Timeout 値の設定	255
例：非アクティブ タイマーの設定	255
例：et-analytics の有効化	255
例：et-analytics 設定の確認	255
その他の参考資料	256
暗号化トラフィック分析の機能履歴	256

第 11 章

ネットワークの検出と対応の設定	259
ネットワークの検出と対応の前提条件	259
ネットワークの検出と対応の制約事項	259
ネットワークの検出と対応について	260
ネットワークの検出と対応の設定方法	260
登録に向けた証明書の設定	260
ファイルアップロードに向けた証明書の設定	262
デバイスでの Stealthwatch Cloud の設定	263
Flexible NetFlow を Stealthwatch Cloud ポータルと統合する方法	265
フロー レコードの作成	265
フロー エクスポートの作成	267
フロー モニタの設定	267
インターフェイスへのフローの適用	268
ネットワークの検出と対応の設定の確認	269
ネットワークの検出と対応の設定例	270
例：デバイスでの Stealthwatch Cloud の設定と統合	270
例：Stealthwatch Cloud の設定の確認	270

ネットワークの検出と対応の機能履歴 273



第 1 章

Autoconf の設定

ここでは、Autoconf に関する情報と Autoconf の設定方法について説明します。

- [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](#) (18 ページ)
- [Autoconf のその他の参考資料](#) (20 ページ)
- [Autoconf の機能履歴](#) (20 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。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 はスイッチオーバー機能をサポートしていません。

Autoconf に関する情報

ここでは、Autoconf について説明します。

Autoconf の利点

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 は、Cisco 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および改善し、組織がネットワーク上で優れた運用と拡張サービスを提供できるようにする包括的な機能セットです。

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では、設定またはポリシー、またはその両方をテンプレートとして適用できます。

サービスとインターフェイスの両方のテンプレートは、設定とポリシーのコンテナとして名前が付けられます。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、含まれている設定と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てい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。これらのテンプレートはデバイスで定義でき、組み込みまたはユーザー定義の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 組み込みテンプレートおよびトリガーは、上記のタスク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設定に表示されます。

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
- プラットフォームタイプ
- 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、参照先テンプレートは、参照先セッションをホストしているインターフェイスに適用する必要があります。
- **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：インターフェイステンプレートが、ローカルに定義されているサービステンプレートか AAA サーバーから送信されたサービステンプレートで参照されている場合、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しているインターフェイスにインターフェイステンプレートを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参照用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の下の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または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）または両方をアクティブ化し、以前のポリシーに関連付けられているテンプレートがある場合は削除します。
- デバイスからテンプレートパラメータへのマップ：フィルタタイプからサービスまたはインターフェイステンプレートへのマッピング、またはその両方を効率的で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

自動設定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。これは、リンクアップまたはリンクダウンを含む使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。この操作中に、適用されたテンプレートのすべてのインスタンスも更新されます。
- 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。
- テンプレートの適用が確認され、障害発生時に修復アクションの同期と実行が可能になります。

- 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。

表 1: Autoconf の機能

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	有効	Autoconf がグローバルレベルで有効になっている場合は、デフォルトでインターフェイスレベルでも有効です。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、グローバル コンフィギュレーション モードで Autoconf スティッキ機能の **access-session interface-template sticky** コマンドを設定することで、エンドデバイスへのリンクがダウンした場合やエンドデバイスが切断された場合でも、テンプレートを保持できます。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

ここでは、Autoconf の設定方法について説明します。

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

	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： 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 5	show device classifier attached interface interface-type interface-number 例： Device# show device classifier attached interface Gi3/0/26	（任意）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。

	コマンドまたはアクション	目的
ステップ 6	show template binding target <i>interface-type 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。

例

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

```

次の例は、Autoconfを設定した後にグローバルコンフィギュレーションを確認する方法を示しています。

```

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

	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。

	コマンドまたはアクション	目的
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 説明 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。

例

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
001d.a1ef.23a8	Gi1/0/7	30	3	C	M	Default
cisco AIR-AP1131AG-A-K9						
001e.7a26.eb05	Gi1/0/30	70	2	C	M	Default
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

	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：	Autoconf 機能を有効にします。

	コマンドまたはアクション	目的
	<code>Device(config)# autoconf enable</code>	
ステップ 8	end 例： <code>Device(config)# end</code>	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。

プラットフォーム タイプ フィルタの設定

次のタスクでは、クラスマップおよびパラメータマップのプラットフォーム タイプ フィルタを設定する方法を示します。

クラスマップのプラットフォーム タイプ フィルタの設定

制御クラスは、制御ポリシーのアクションが実行される条件を定義します。制御ポリシーのアクションを実行するためには、すべての条件を評価する必要があるか、いずれかの条件を評価する必要があるか、または評価しないかを定義する必要があります。プラットフォームタイプは、制御ポリシーで指定されたプラットフォーム名に基づいて評価されます。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： <code>Device> 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： <code>Device# 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	class-map type control subscriber {match-all match-any match-none} <i>control-class-name</i> 例： <code>Device (config)# class-map type control subscriber match-all DOT1X_NO_AGENT</code>	制御クラスを作成し、制御クラスマップ フィルタ モードを開始します。 <ul style="list-style-type: none"> • match-all : 制御クラスのすべての条件に一致する必要があります。 • match-any : 制御クラスの少なくとも 1 つの条件に一致する必要があります。 • match-none : 制御クラスのどの条件にも一致しない必要があります。

	コマンドまたはアクション	目的
ステップ 4	match platform-type <i>platform-name</i> 例： Device (config-filter-control-classmap) # match platform-type C3850	指定されたプラットフォームタイプに基づいて制御クラスを評価する条件を作成します。
ステップ 5	end 例： Device (config-filter-control-classmap) # end	制御クラスマップフィルタモードを終了し、特権 EXEC モードに戻ります。
ステップ 6	show class-map type control subscriber { all name control-class-name } 例： Device# show class-map type control subscriber all	(任意) すべてのクラスマップまたは特定のクラスマップの制御ポリシーに関する情報を表示します。

パラメータマップのプラットフォームタイプフィルタの設定

パラメータマップを使用することを推奨します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	parameter-map type subscriber attribute-to-service <i>parameter-map-name</i> 例： Device (config) # parameter-map type subscriber attribute-to-service Aironet-Policy-para	パラメータマップのタイプと名前を指定し、パラメータマップ フィルタ モードを開始します。
ステップ 4	map-index map platform-type {{ eq not-eq regex } <i>filter-name</i> } 例： Device (config-parameter-map-filter) # 10 map platform-type eq C9xxx	プラットフォームタイプにパラメータマップ属性フィルタ基準を指定します。

	コマンドまたはアクション	目的
ステップ 5	end 例： Device (config-parameter-map-filter-submode) # end	パラメータマップ フィルタ モードを終了し、特権 EXEC モードに戻ります。
ステップ 6	show parameter-map type subscriber attribute-to-service {all name parameter-map-name} 例： Device# show parameter-map type subscriber attribute-to-service	(任意) パラメータマップ属性を表示します。

クラスマップのデバイスタイプフィルタの設定

制御クラスは、制御ポリシーのアクションが実行される条件を定義します。制御ポリシーのアクションを実行するためには、すべての条件を評価する必要があるか、いずれかの条件を評価する必要があるか、または評価しないかを定義する必要があります。デバイスタイプは、制御ポリシーで指定されたデバイス名に基づいて評価されます。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	class-map type control subscriber {match-all match-any match-none} control-class-name 例： Device (config)# class-map type control subscriber match-all Device-Type-Match	制御クラスを作成し、制御クラスマップ フィルタ モードを開始します。 <ul style="list-style-type: none"> • match-all : 制御クラスのすべての条件に一致する必要があります。 • match-any : 制御クラスの少なくとも1つの条件に一致する必要があります。 • match-none : 制御クラスのどの条件にも一致しない必要があります。

	コマンドまたはアクション	目的
ステップ 4	match device-type {device-name regex regular-expression} 例 : <pre>Device (config-filter-control-classmap) # match device-type laptop Device (config-filter-control-classmap) # match device-type regex cis*</pre>	指定されたデバイスタイプに基づいてコントロールクラスを評価するための条件を作成します。 <ul style="list-style-type: none"> • device-name : クラスマップ属性フィルタ条件のデバイス名を入力します。 • regexregular-expression : フィルタタイプを指定する正規表現を入力します。
ステップ 5	end 例 : <pre>Device (config-filter-control-classmap) # end</pre>	制御クラスマップフィルタモードを終了し、特権 EXEC モードに戻ります。
ステップ 6	show class-map type control subscriber {all name control-class-name} 例 : <pre>Device# show class-map type control subscriber all</pre>	(任意) すべてのクラスマップまたは特定のクラスマップの制御ポリシーに関する情報を表示します。

Autoconf の設定例

次のセクションに Autoconf 機能の設定例を示します。

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。

```
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

次に、変更された組み込みテンプレートをエンドデバイスに適用し、設定を確認する例を示します。

```
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
```

例 : プラットフォーム タイプ フィルタの設定

次の例では、クラスマップのプラットフォームタイプフィルタを設定する方法を示します。

```
Device> enable
Device# configure terminal
Device(config)# class-map type control subscriber match-all DOT1X_NO_AGENT
Device(config-filter-control-classmap)# match platform-type C9xxx
Device(config-filter-control-classmap)# end
Device#
```

次の例では、パラメータマップのプラットフォームタイプフィルタを設定する方法を示します。

```
Device> enable
Device# configure terminal
Device(config)# parameter-map type subscriber attribute-to-service Aironet-Policy-para
Device(config-parameter-map-filter)# 10 map platform-type eq C9xxx
Device(config-parameter-map-filter-submode)# end
Device#
```

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	「Port Based Network Access Control」

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 16.5.1a	[Autoconf]	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 機能では、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ます。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。
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	プラットフォーム フィルタ処理オプションはクラスマップとパラメータマップの設定用に導入されました。

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 にアクセスするには、<https://cfng.cisco.com/> に進みます。



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 (23 ページ)
- シスコ プラグ アンド プレイ の機能履歴 (23 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

シスコ プラグ アンド プレイ 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 16.5.1a	シスコ プラグ アンド プレイ	Cisco® プラグ アンド プレイ ソリューションは、安全性が高くスケーラブルかつシームレスな、統一されたゼロタッチ導入エクスペリエンスを提供する統合ソリューションです。

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 にアクセスするには、<https://cfmng.cisco.com/> に進みます。



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 \(25 ページ\)](#)
- [Cisco Discovery Protocol の設定方法 \(26 ページ\)](#)
- [Cisco Discovery Protocol のモニタリングとメンテナンス \(33 ページ\)](#)
- [Cisco Discovery Protocol の機能の履歴 \(34 ページ\)](#)

Cisco Discovery Protocol について

ここでは、Cisco Discovery Protocol について説明します

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

次の内容は、デバイスおよび接続されたエンドポイントデバイスに当てはまります。

- Cisco Discovery Protocol は、デバイス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 つの有線デバイスだけがロケーション情報をレポートします。
- 有線デバイス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 の設定方法

ここでは、Cisco Discovery Protocol の設定方法について説明します。

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

	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	configureterminal 例： Device# configure terminal	
ステップ 3	cdp timer seconds 例： Device(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： Device(config)# cdp holdtime 60	(任意) 受信デバイスがこのデバイスから送信された情報を破棄せずに保持する時間を指定します。 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： Device(config)# cdp advertise-v2	(任意) バージョン 2 アドバタイズを送信するように Cisco Discovery Protocol を設定します。 これは、デフォルトの状態です。
ステップ 6	end 例： Device(config)# end	特権 EXEC モードに戻ります。
ステップ 7	show running-config 例：	入力を確認します。

	コマンドまたはアクション	目的
	Device# <code>show running-config</code>	
ステップ 8	copy running-config startup-config 例： Device# <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

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

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> <code>enable</code>	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	configureterminal 例： Device# <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	no cdp run 例： Device(config)# <code>no cdp run</code>	Cisco Discovery Protocol を無効にします。

	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

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

	コマンドまたはアクション	目的
ステップ 1	enable 例 :	特権 EXEC モードを有効にします。

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

	コマンドまたはアクション	目的
	Device> enable	<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	configureterminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	cdp run 例： Device(config)# cdp run	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： 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

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

	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	configureterminal 例 : Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	interface interface-id 例 : Device(config)# interface gigabitethernet 1/0/1	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 : Device(config-if)# no cdp enable	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 : Device(config)# end	特権 EXEC モードに戻ります。
ステップ 6	show running-config 例 : Device# show running-config	入力を確認します。

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

インターフェイス上での 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

	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	configureterminal 例 : Device# configure terminal	グローバル コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	interface interface-id 例 : Device(config)# interface gigabitethernet1/0/1	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	cdp enable 例 : Device(config-if)# cdp enable	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	end 例 : Device(config)# end	特権 EXEC モードに戻ります。
ステップ 6	show running-config 例 : Device# show running-config	入力を確認します。
ステップ 7	copy running-config startup-config 例 : Device# copy running-config startup-config	(任意) コンフィギュレーション ファイルに設定を保存します。

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

表 2: Cisco Discovery Protocol 情報を表示するためのコマンド

コマンド	説明
clear cdp counters	トラフィック カウンタを 0 にリセットします。
clear cdp table	ネイバーに関する情報の Cisco Discovery Protocol テーブルをクリアします。
show cdp	送信間隔、送信したパケットの保持時間などのグローバル設定を表示します。

コマンド	説明
show cdp entry <i>entry-name</i> [version] [protocol]	特定のネイバーに関する情報を表示します。 アスタリスク (*) を入力して、すべての 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入力することもできます。 また、指定されたネイバー上でイネーブルになっているフラグの情報や、デバイス上で稼働しているソフトウェアのバージョンが表示されるように、表示内容を制限することもできます。
show cdp interface [<i>interface-id</i>]	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。 必要なインターフェイスの情報だけを表示できます。
show cdp neighbors [<i>interface-id</i>] [<i>detail</i>]	装置タイプ、インターフェイスタイプ、インターフェイスタイプ時間の設定値、機能、プラットフォーム、ポートIDを含めた情報を表示します。 特定のインターフェイスに関するネイバー情報だけを表示し、詳細表示にするため表示内容を拡張したりできます。
show cdp traffic	Cisco Discovery Protocol カウンタ（送信済み/受信済みパケットサム エラー数を含む）を表示します。
show ap cdp neighbors	アクセス ポイントの Cisco Discovery Protocol ネイバーに関する情報を表示します。
show ap cdp neighbors detail	アクセス ポイントの Cisco Discovery Protocol ネイバーに関する詳細情報を表示します。
show ap name ap-name cdp neighbors	アクセス ポイントの Cisco Discovery Protocol 情報を表示します。
show ap name ap-name cdp neighbors detail	Cisco Discovery Protocol を使用している特定のアクセスポイントに関する詳細情報を表示します。

Cisco Discovery Protocol の機能の履歴

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 16.5.1a	Cisco Discovery Protocol	Cisco Discovery Protocol は、シスコデバイス上で動作し、ネットワーク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。

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 にアクセスするには、<https://cfmng.cisco.com> に進みます。



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 \(37 ページ\)](#)
- [SNMP の制約事項 \(39 ページ\)](#)
- [SNMP に関する情報 \(40 ページ\)](#)
- [SNMP の設定方法 \(46 ページ\)](#)
- [SNMP の例 \(56 ページ\)](#)
- [SNMP ステータスのモニタリング \(57 ページ\)](#)
- [簡易ネットワーク管理プロトコルの機能の履歴 \(58 ページ\)](#)

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 3: 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す。

SNMP の制約事項

バージョンの制約事項

- SNMPv1 は informs をサポートしていません。

- 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 ENTITY-MIB は、イーサネット管理ポートではサポートされていません。

SNMP に関する情報

ここでは、SNMP の概要について説明します。

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。エージェントと MIB はネットワークデバイス上に存在します。デバイスに SNMP を設定するには、マネージャとエージェントの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 4: 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
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。



(注) パフォーマンスに関連する問題を回避するために、SNMP マネージャで **ciscoFlashFileDate** MIB オブジェクトをクエリから除外することを推奨します。これは、**ciscoFlashFileDate** オブジェクトが MIB で公開されていても、製品ではサポートされていないためです。

SNMP エージェント機能

SNMP エージェントは、1つ以上の SNMP マネージャから要求を受信できます。すべての要求に、NMS の IP アドレス、NMS がエージェントをポーリングした回数、およびポーリングのタイムスタンプが含まれます。この情報は、IPv4 サーバーと IPv6 サーバーの両方で追跡できます。

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。

show snmp stats hosts コマンドを使用してキュー内の SNMP マネージャ要求のリストを表示し、**clear snmp stats hosts** コマンドを使用してキューをクリアします。

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、ポートまたはホ

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 コミュニティストリング

SNMP コミュニティストリングは、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 がデバイスにアクセスするには、NMS 上のコミュニティストリング定義がデバイス上の3つのコミュニティストリング定義の少なくとも1つと一致しなければなりません。

コミュニティストリングの属性は、次のいずれかです。

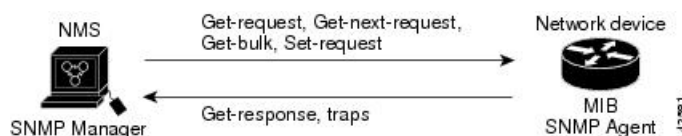
- 読み取り専用 (RO) : コミュニティストリングを除き MIB 内のすべてのオブジェクトに、許可された管理ステーションに対する読み取りアクセス権を与えますが、書き込みアクセスは許可しません。
- 読み取り-書き込み (RW) : MIB 内のすべてのオブジェクトに、許可された管理ステーションに対する読み取りおよび書き込みアクセス権を与えますが、コミュニティストリングへのアクセスは許可しません。
- クラスタを作成すると、コマンドデバイスがメンバデバイスと SNMP アプリケーション間のメッセージ交換を管理します。Network Assistant ソフトウェアは、コマンドデバイス上で最初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

SNMP MIB 変数アクセス

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

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

図 1: SNMP ネットワーク



SNMP フラッシュ MIB

Cisco フラッシュ MIB は、シスコ製のデバイスからフラッシュファイルデータを照会するために使用されます。以前は、フラッシュ MIB でデバイスごとにパーティションあたりのリストされるファイルの数は 100 に制限されていました。Cisco IOS XE 17.1.1 リリース以降、デバイスごとにパーティションあたり 100 ファイルをリストするフラッシュ MIB の制限が削除されました。現在フラッシュ MIB では、フラッシュファイルシステムからすべてのファイルを取得するようになりました。フラッシュ MIB ウォークを実行するには、**snmp mib flash cache** コマンドを使用する必要があります。**snmp mib flash cache** コマンドは、すべてのファイルをローカルフラッシュ MIB キャッシュにプリフェッチします。100 ファイルの制限が削除されたため、ファイルを取得するためのフラッシュ MIB ウォークには時間がかかります。

SNMP 通知

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



(注) SNMPv1 は **informs** をサポートしていません。

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

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

SNMP ifIndex MIB オブジェクト値

SNMP エージェントの IF-MIB モジュールがリブート後すぐに起動されます。さまざまな物理インターフェイス ドライバが IF-MIB モジュールの登録を初期化されているように、「インデックス番号をください」と示します。IF-MIB モジュールが先着順で使用可能な次の ifIndex

番号を割り当てます。つまり、1つのリポートから他のリポ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ポ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 ENTITY-MIB 識別子

ENTITY-MIBには、現場交換可能ユニット（FRU）、ファン、デバイスの電源装置などの物理エンティティを管理するための情報が含まれています。各エンティティは、現在のMIBや他のMIB内のエンティティに関する情報へのアクセスに使用する一意のインデックス番号（*entPhysicalIndex*）によって識別されます。エンティティの活性挿抜（OIR）により、新しいエンティティが挿入されたか、既存のエンティティが再挿入されたかに関係なく、エンティティには次に使用可能な *entPhysicalIndex* 番号が割り当てられます。

SNMP and Syslog Over IPv6

IPv4 と IPv6 の両方をサポートするには、IPv6 のネットワーク管理で IPv4 および IPv6 のトランスポートが必要になります。Syslog over IPv6 は、このトランスポートのアドレスデータタイプをサポートします。

Simple Network Management Protocol (SNMP) と syslog over IPv6 は、次の機能を提供します。

- IPv4 と IPv6 両方のサポート
- SNMP に対する IPv6 トランスポート、および SNMP 変更による IPv6 ホストのトラップのサポート
- IPv6 アドレス指定をサポートするための SNMP および syslog に関連する MIB
- IPv6 ホストをトラップ レシーバとして設定

Over IPv6 をサポートするため、SNMP は既存の IP トランスポート マッピングを変更して、IPv4 と IPv6 を同時にサポートします。次の SNMP 動作は、IPv6 トランスポート管理をサポートします。

- デフォルト設定のユーザー データグラム プロトコル (UDP) SNMP ソケットを開く
- *SR_IPV6_TRANSPORT* と呼ばれる新しいトランスポート メカニズムを提供
- IPv6 トランスポートによる SNMP 通知の送信
- IPv6 トランスポートの SNMP 名のアクセス リストのサポート
- IPv6 トランスポートを使用した SNMP プロキシ転送のサポート
- SNMP マネージャ機能と IPv6 トランスポートの連動確認

設定手順を含む、SNMP over IPv6 については、Cisco.com で『*Cisco IOS IPv6 Configuration Library*』の「Managing Cisco IOS Applications over IPv6」の章を参照してください。

設定手順を含む、syslog over IPv6 については、Cisco.com で『Cisco IOS IPv6 Configuration Library』の「Implementing IPv6 Addressing and Basic Connectivity」の章を参照してください。

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

SNMP 設定時の注意事項

デバイスでは、SNMP UDP ポート 161 および 162 を開き、SNMP エージェントを有効にするために、**snmp-server host**、**snmp-server user**、**snmp-server community**、**snmp-server manager** のいずれかのグローバル コンフィギュレーション コマンドを設定する必要があります。

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

SNMP グループを設定するときには、次の注意事項に従ってください。

- SNMP グループを設定するときは、通知ビューを指定しません。**snmp-server host** グローバル コンフィギュレーション コマンドがユーザの通知ビューを自動生成し、そのユーザ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、デバイスは **auth** (authNoPriv) および **priv** (authPriv) の認証レベルの情報を送信しません。
- SNMP エンジン ID の値を変更すると、重大な影響が生じます。(コマンドラインで入力された) ユーザのパスワードは、パスワードおよびローカルエンジン ID に基づいて、MD5 または SHA セキュリティダイジェストに変換されます。コマンドラインのパスワードは、RFC2274 の規定に従って廃棄されます。このようにパスワードが廃棄されるため、エンジン ID 値を変更した場合は、SNMPv3 ユーザのセキュリティダイジェストが無効となり、**snmp-server user username** グローバル コンフィギュレーション コマンドを使用して、SNMP ユーザを再設定する必要があります。エンジン ID を変更した場合は、同様の制限によってコミュニティストリングも再設定する必要があります。
- SNMP サーバーホストをデフォルトの UDP ポート (162) で設定すると、**show running-config** コマンドの出力に UDP ポート値が表示されません。**snmp-server host {host-addr} community-string udp-port value** コマンドを使用してデフォルト以外の UDP ポート値を指定すると、UDP ポート番号が **show running-config** コマンド出力に表示されます。デフォルトの UDP ポート 162 を使用しても使用しなくても **snmp-server host** コマンドを設定できます。ただし、両方を同時に設定することはできません。

正しい例を次に示します。

```
Device(config)# snmp-server host 10.10.10.10 community udp-port 163
Device(config)# snmp-server host 10.10.10.10 community
```

```
Device(config)# snmp-server host 10.10.10.10 community udp-port 163
Device(config)# snmp-server host 10.10.10.10 community udp-port 162
```

次の例は正しくありません。

```
Device(config)# snmp-server host 10.10.10.10 community udp-port 163
Device(config)# snmp-server host 10.10.10.10 community
Device(config)# snmp-server host 10.10.10.10 community udp-port 162
```

```
Device(config)# snmp-server host 10.10.10.10 community udp-port 163
Device(config)# snmp-server host 10.10.10.10 community udp-port 162
Device(config)# snmp-server host 10.10.10.10 community
```

SNMP の設定方法

ここでは、SNMP の設定方法について説明します。

SNMP コミュニティストリング

SNMP コミュニティストリングは、MIB オブジェクト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 がデバイスにアクセスするには、NMS 上のコミュニティストリング定義がデバイス上の 3 つのコミュニティストリング定義の少なくとも 1 つと一致しなければなりません。

コミュニティストリングの属性は、次のいずれかです。

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

SNMP グループおよびユーザの設定

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例 : Device> enable	特権 EXEC モードを有効にします。 • 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	snmp-server engineID { local engineid-string remote ip-address [udp-port port-number] engineid-string } 例 : Device(config)# snmp-server engineID local 1234	SNMP のローカル コピーまたはリモート コピーに名前を設定します。 • engineid-string は、SNMP のコピー名を指定する 24 文字の ID ストリングです。後続ゼロが含まれる場合は、24 文字のエンジン ID すべてを指定する必要はありません。指定するのは、エンジン ID のうちゼロのみが続く箇所を除いた部分だけです。手順例では、

	コマンドまたはアクション	目的
		<p>12340000000000000000000000000000 のエンジン ID を設定します。</p> <ul style="list-style-type: none"> • 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 <i>ip-address</i> 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。
<p>ステップ 4</p>	<pre>snmp-server group group-name {v1 v2c v3 {auth noauth priv}} [read readview] [write writeview] [notify notifyview] [access access-list]</pre> <p>例 :</p> <pre>Device(config)# snmp-server group public v2c access lmnop</pre>	<p>リモート デバイス上で新しい SNMP グループを設定します。</p> <p><i>group-name</i> には、グループの名前を指定します。</p> <p>次のいずれかのセキュリティ モデルを指定します。</p> <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送できます。 • v3最も安全な場合には、次の認証レベルの 1 つを選択する必要があります。 <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</p>

	コマンドまたはアクション	目的
		<p>表す文字列 (64 文字以内) を入力します。</p> <p>(任意) write <i>writeview</i> 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 <i>notifyview</i> 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 <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>
<p>ステップ 5</p>	<p>snmp-server user <i>username group-name</i> { remote <i>host</i> [udp-port <i>port</i>] } { v1 [access <i>access-list</i>] v2c [access <i>access-list</i>] v3 [encrypted] [access <i>access-list</i>] [auth { md5 sha } <i>auth-password</i>] } [priv { des 3des aes { 128 192 256 } } <i>priv-password</i>]</p> <p>例 :</p> <pre>Device(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。 • 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

	コマンドまたはアクション	目的
		<p>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> <pre>Device(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>Device# 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>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。

SNMP UDP ポートの開閉

SNMP プロセスはポート 161 および 162 を使用します。ポート 161 はデバイスのポーリングに使用され、ポート 162 はエージェントからサーバーへの通知の送信に使用されます。必要なコマンドのいずれかが設定されていない場合、SNMP UDP ポートは閉じたままになります。この設計により、必要な場合にのみポートが開くため、セキュリティが向上し、デバイスは不必要にポートをリスンしなくなります。

SNMP UDP ポートを開くには、ユーザー EXEC モードで次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーションモードを開始します。
ステップ 3	snmp-server {host user community manager} 例： Device(config)# snmp-server host	SNMP UDP ポート 161 および 162 を開きます。オプション (host 、 user 、 community 、 manager) のいずれかを設定すると、両方のポートが開きます。ポートを閉じるには、設定したすべてのオプションの no 形式を入力します。キーワードが 1 つでも設定されていると、ポートは開いたままになります。 キーワードを指定せずに no snmp-server コマンドを入力すると、SNMP UDP ポートだけでなく、SNMP プロセスもシャットダウンされます。
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 udp 例： Device# show udp	SNMP UDP ポートを表示します。必要なコマンドのいずれかが設定されている場合、ポート 161 および 162 は、リモートフィールドの下に値 listen を表示します。

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

SNMP 通知

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



(注) SNMPv1 は informs をサポートしていません。

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

エージェント コンタクトおよびロケーションの設定

SNMPエージェントのシステム接点およびロケーションを設定して、コンフィギュレーションファイルからこれらの記述にアクセスできるようにするには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例：	特権 EXEC モードを有効にします。

	コマンドまたはアクション	目的
	Device> enable	<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	snmp-server contact text 例： Device(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： Device(config)# snmp-server location Building 3/Room 222	システムの場所を表す文字列を設定します。
ステップ 5	end 例： Device(config)# end	特権 EXEC モードに戻ります。
ステップ 6	show running-config 例： Device# show running-config	入力を確認します。
ステップ 7	copy running-config startup-config 例： Device# copy running-config startup-config	(任意) コンフィギュレーション ファイルに設定を保存します。

SNMP を通して使用する TFTP サーバの制限

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • 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	snmp-server tftp-server-list access-list-number 例： Device(config)# snmp-server tftp-server-list 44	SNMP を介したコンフィギュレーション ファイルのコピーに使用する TFTP サーバを、アクセス リストのサーバに限定します。 <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 access-list-number {deny permit} source [source-wildcard] 例： Device(config)# access-list 44 permit 10.1.1.2	標準アクセス リストを作成し、コマンドを必要な回数だけ実行します。 <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、デバイスにアクセスできる TFTP サーバの IP アドレスを入力します。 (任意) <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します。
ステップ 5	end 例：	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	Device(config)# end	
ステップ 6	show running-config 例： Device# show running-config	入力を確認します。
ステップ 7	copy running-config startup-config 例： Device# copy running-config startup-config	(任意) コンフィギュレーションファイルに設定を保存します。

SNMP 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て、SNMP プロセスをシャットダウンします。グローバル コンフィギュレーション モードで、**snmp-server host**、**snmp-server user**、**snmp-server community**、**snmp-server manager** のいずれかのコマンドを入力して、SNMP エージェントのすべてのバージョンを再度イネーブルにします。特に SNMP をイネーブルにするために指定された Cisco IOS コマンドはありません。

SNMP エージェントをディセーブルにするには、次の手順を実行します。

始める前に

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 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	no snmp-server 例： Device(config)# no snmp-server	SNMP エージェント動作をディセーブルにします。
ステップ 4	end 例： Device(config)# end	特権 EXEC モードに戻ります。
ステップ 5	show running-config 例： Device# show running-config	入力を確認します。
ステップ 6	copy running-config startup-config 例： Device# copy running-config startup-config	(任意) コンフィギュレーション ファイルに設定を保存します。

SNMP 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、デバイスはトラップを送信しません。

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

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

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

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

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

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

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

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

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

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

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

次に、SNMP エージェントにポーリングされた SNMP マネージャのエントリを表示する例を示します。

```
Device# show snmp stats host
Request Count          Last Timestamp          Address
2                    00:00:01 ago           3.3.3.3
1                    1w2d ago               2.2.2.2
```

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 5: 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。

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

簡易ネットワーク管理プロトコルの機能の履歴

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 16.5.1a	SNMP	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。

リリース	機能	機能情報
Cisco IOS XE Amsterdam 17.1.1	SNMP UDP ポートの開閉	SNMP UDP ポートは、必要なコマンドのいずれかが設定されている場合にのみ開かれます。この設計により、必要な場合にのみポートが開くため、セキュリティが向上し、デバイスは不必要にポートをリッスンしなくなります。
	NMS ポーリングの追跡	SNMP エージェントにポーリングされた SNMP マネージャの詳細を表示できます。show snmp server stats hosts コマンドは、NMS の IP アドレス、NMS がエージェントをポーリングした回数、およびポーリングのタイムスタンプを表示するために使用します。

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 にアクセスするには、<https://cfmng.cisco.com/> に進みます。



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 \(61 ページ\)](#)
- [サービスレベル契約に関する情報 \(61 ページ\)](#)
- [IP SLA 動作の設定方法 \(67 ページ\)](#)
- [IP SLA 動作のモニタリング \(82 ページ\)](#)
- [IP SLA 動作のモニタリングの例 \(82 ページ\)](#)
- [その他の参考資料 \(83 ページ\)](#)
- [サービスレベル契約の機能履歴 \(84 ページ\)](#)

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。

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

サービスレベル契約に関する情報

ここでは、サービスレベル契約について説明します。

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なりモート 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。

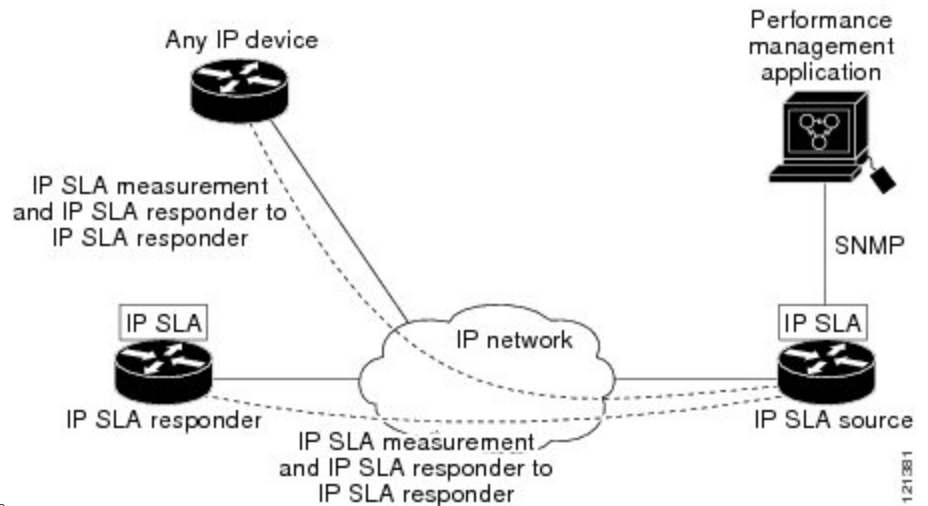
- 端末間のネットワーク アベイラビリティをモニタリングして、ネットワーク リソースをあらかじめ検証し接続をテストできる（たとえば、ビジネス上の重要なデータを保存する NFS サーバーのネットワーク アベイラビリティをリモート サイトから確認できる）。
- 問題をすぐに認識し、トラブルシューティングにかかる時間を短縮できる一貫性のある信頼性の高い測定によるネットワーク動作のトラブルシューティング。
- マルチプロトコル ラベル スイッチング (MPLS) パフォーマンス モニタリングとネットワークの検証を行う（デバイスが MPLS をサポートする場合）。

Cisco IOS IP SLA でのネットワーク パフォーマンスの測定

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



ク測定を行います。

IP SLA レスポンダおよび IP SLA 制御プロトコル

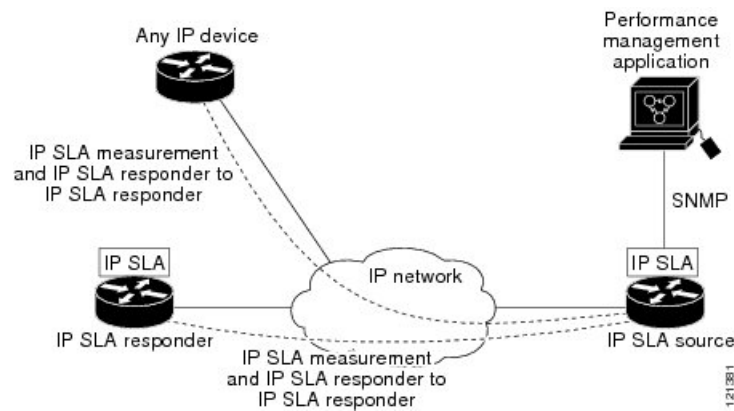
IP SLA レスポンダは宛先 Cisco デバイスに組み込まれたコンポーネントで、システムが IP SLA 要求パケットを予想して応答します。Responder は専用プローブなしで正確な測定を行います。レスポンダは、受信および応答するポートが通知されるメカニズムを Cisco IOS IP SLA コントロール プロトコルを通じて実現します。



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

次の図は、IP ネットワーク内での Cisco IOS IP SLA レスポンダの配置場所を示します。レスポンダは、IP SLA 動作から送信されたコントロールプロトコルメッセージを指定されたポートで受信します。コントロールメッセージを受信したら、指定された UDP または TCP ポートを指定された時間だけ有効にします。この間に、レスポンダは要求を受け付け、応答します。レスポンダは、IP SLA パケットに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 認証が利用できません。

図 3: Cisco IOS IP SLA 動作



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

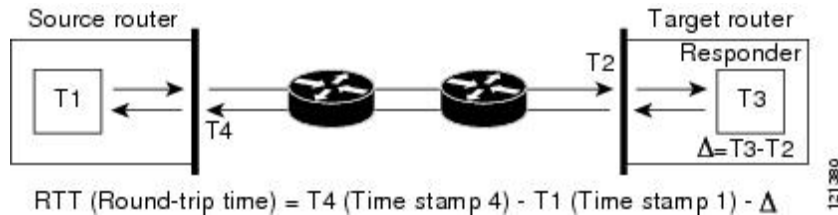
IP SLA の応答時間の計算

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

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

図 4: Cisco IOS IP SLA レスポンダ タイムスタンプ

次の図に、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](#)』を参照してください。

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

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

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

```
Device# show ip sla application

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

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

Supported Features:
      IPSLAs Event Publisher

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

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

IP SLA レスポンダの設定

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。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	ip sla responder {tcp-connect udp-echo} ipaddress ip-address port port-number 例：	デバイスを IP SLA レスポンダとして設定します。 キーワードの意味は次のとおりです。

	コマンドまたはアクション	目的
	<pre>Device(config)# ip sla responder udp-echo 172.29.139.134 5000</pre>	<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。 <p>(注) IP アドレスとポート番号は、IP SLA 動作のソース デバイスに設定した IP アドレスおよびポート番号と一致している必要があります。</p>
ステップ 4	<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 5	<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 6	<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 7	<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。

IP SLA ネットワーク パフォーマンス測定の実装

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

始める前に

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 <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# 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	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）。 <ul style="list-style-type: none"> destination-ip-address destination-hostname : 宛先 IP アドレスまたはホスト名を指定します。 destination-port : 宛先ポート番号を 1 ~ 65535 の範囲で指定します。 (任意) source-ip {<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

	コマンドまたはアクション	目的
		<p>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。</p> <ul style="list-style-type: none"> • (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。 <p>(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。</p> <ul style="list-style-type: none"> • (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。 • (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 :	(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

	コマンドまたはアクション	目的
	Device(config-ip-sla-jitter)# frequency 45	します。指定できる範囲は1～604800秒で、デフォルトは60秒です。
ステップ6	threshold milliseconds 例： Device(config-ip-sla-jitter)# threshold 200	(任意) しきい値条件を設定します。次の例では、指定されたIP SLA動作のしきい値が200に設定されます。有効な範囲は0～60000ミリ秒です。
ステップ7	exit 例： Device(config-ip-sla-jitter)# exit	SLA動作コンフィギュレーションモード(この例ではUDPジッターコンフィギュレーションモード)を終了し、グローバルコンフィギュレーションモードに戻ります。
ステップ8	ip sla schedule operation-number [life {forever seconds}] [start-time {hh:mm[:ss] [month day day month] pending now after hh:mm:ss} [ageout seconds] [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(seconds)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 hh:mm:ss と入力すれば、指定した時刻の経過後に動作を開始します。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • (任意) ageout seconds : 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。 • (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

	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	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>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。</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。 <p>(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 レスポンドによる CPU 使用率が高くなる可能性があります。</p> <ul style="list-style-type: none"> • (任意) control : IP SLA 制御メッセージの IP SLA レスポンド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 レスポンドとの接続が確立されます。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • (任意) 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。
ステップ 5	frequency seconds 例 : Device(config-ip-sla-jitter) # frequency 45	(任意) 指定した IP SLA 動作を繰り返す間隔を設定します。指定できる範囲は1～604800秒で、デフォルトは60秒です。
ステップ 6	exit 例 : Device(config-ip-sla-jitter) # exit	UDPジッターコンフィギュレーションモードを終了し、グローバルコンフィギュレーションモードに戻ります。
ステップ 7	ip sla schedule operation-number [life {forever seconds}] [start-time {hh:mm[:ss]} [month day day month] pending now after hh:mm:ss] [ageout seconds] [recurring] 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。

	コマンドまたはアクション	目的
		<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 と入力すれば、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。 （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	<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。

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 レスポンダ側を有効にしておく必要はありません。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • 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 terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	ip sla operation-number 例 : Device (config) # ip sla 10	IP SLA 動作を作成し、IP SLA コンフィギュレーションモードを開始します。
ステップ 4	icmp-echo { <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	個々の IP SLA 動作のスケジューリングパラメータを設定します。

	コマンドまたはアクション	目的
	<p>now after hh:mm:ss] [ageout seconds] [recurring]</p> <p>例 :</p> <pre>Device(config)# ip sla schedule 10 start-time now life forever</pre>	<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) に指定するか、特定の秒数 (seconds) 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。 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 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。 • (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>	入力を確認します。

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

ICMP エコー IP SLA 動作の設定

次に、ICMP エコー IP SLA 動作の設定例を示します。

```

Device(config)# ip sla 10
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 10 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 6: IP SLA 動作のモニタリング

show ip sla application	Cisco IOS IP SLA のグローバル設定
show ip sla authentication	IP SLA 認証情報を表示します。
show ip sla configuration [entry-number]	すべての IP SLA 動作または指定された entry-number の IP SLA 動作のデフォルト値をすべて表示します。
show ip sla enhanced-history {collection-statistics distribution statistics} [entry-number]	収集した履歴バケットの IP SLA 動作または指定された entry-number の IP SLA 動作の統計情報を表示します。
show ip sla ethernet-monitor configuration [entry-number]	IP SLA 自動イーサネット監視の設定
show ip sla group schedule [schedule-entry-number]	IP SLA グループ スケジュールの IP SLA 動作を表示します。
show ip sla history [entry-number full tabular]	すべての IP SLA 動作の履歴を表示します。
show ip sla mpls-lsp-monitor {collection-statistics configuration ldp operational-state scan-queue summary [entry-number] neighbors}	MPLS ラベル スイッチの IP SLA 動作を表示します。
show ip sla reaction-configuration [entry-number]	すべての IP SLA 動作または指定された entry-number の IP SLA 動作の予防的しきい値のモニタリング設定
show ip sla reaction-trigger [entry-number]	すべての IP SLA 動作または指定された entry-number の IP SLA 動作の応トリガー情報を表示します。
show ip sla responder	IP SLA レスポнда側の IP SLA 動作
show ip sla statistics [entry-number aggregated details]	動作ステータスおよび IP SLA 動作の結果を表示します。

IP SLA 動作のモニタリングの例

次の例は、アプリケーションごとのすべての IP SLA を示しています。

```
Device# show ip sla application

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

Supported Operation Types:
```

```
icmpEcho, path-echo, path-jitter, udpEcho, tcpConnect, http
dns, udpJitter, dhcp, ftp, udpApp, wspApp
```

```
Supported Features:
  IPSLAs Event Publisher
```

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

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

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

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

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

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

その他の参考資料

関連資料

関連項目	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

MIB

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

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

サービスレベル契約の機能履歴

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 16.5.1a	サービス レベル契約	Cisco IOS IP サービスレベル契約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 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 にアクセスするには、<https://cfng.cisco.com/> に進みます。



第 6 章

SPAN および RSPAN の設定

- [SPAN および RSPAN の前提条件](#) (87 ページ)
- [SPAN および RSPAN の制約事項](#) (87 ページ)
- [SPAN および RSPAN について](#) (90 ページ)
- [SPAN および RSPAN の設定](#) (101 ページ)
- [SPAN および RSPAN の設定方法](#) (102 ページ)
- [SPAN および RSPAN 動作のモニタリング](#) (129 ページ)
- [SPAN および RSPAN の設定例](#) (129 ページ)
- [SPAN および RSPAN の機能履歴](#) (132 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。
- RSPAN セッションでは、DHCP Inject パケットはキャプチャされません。

SPAN および RSPAN について

ここでは、SPAN および RSPAN について説明します。

SPAN および RSPAN

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

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

ローカル SPAN は 1 つのスイッチ内の 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および宛先ポートは、同じスイッチ内にあります。ローカル SPAN は、1 つ以上の送信元ポートからのトラフィックを、解析のため宛先ポートにコピーします。

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。

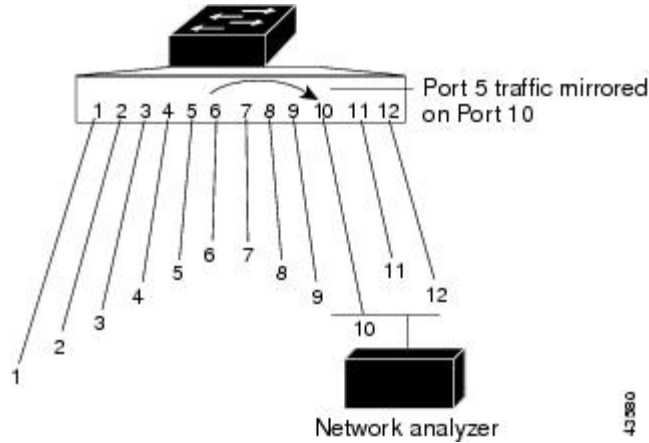
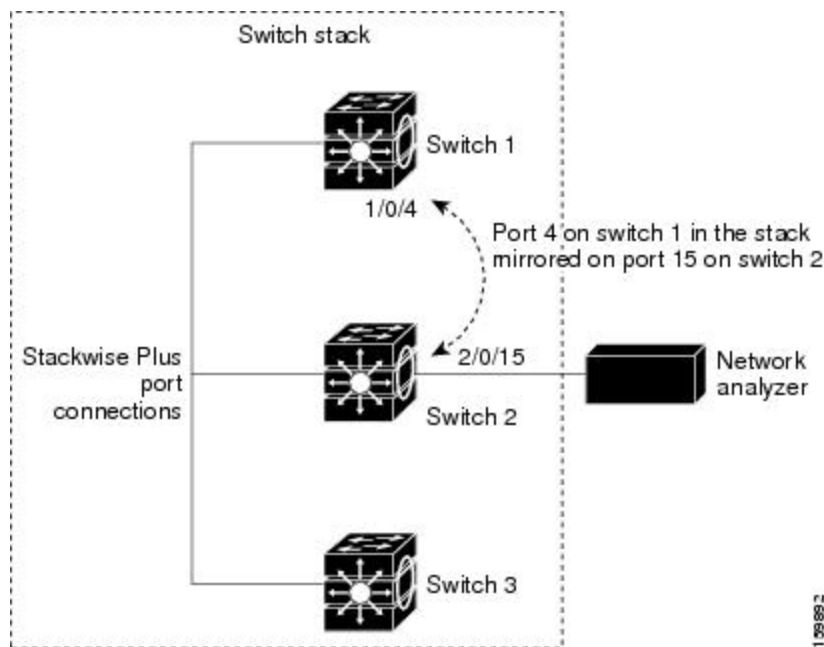


図 6: デバイスタックでのローカル SPAN の設定例

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



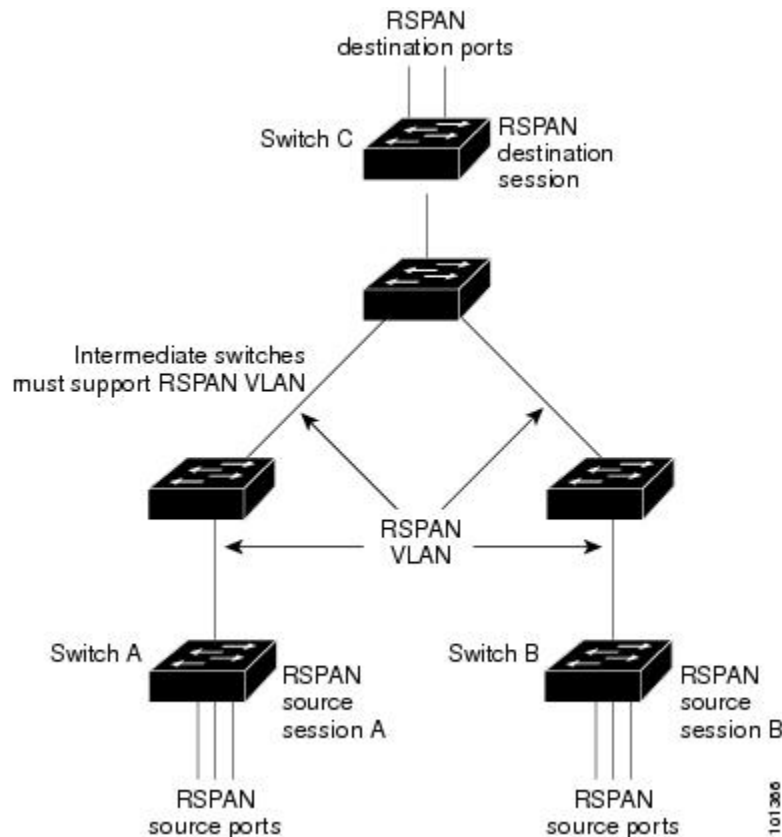
リモート SPAN

RSPAN は、異なるデバイス（または異なるデバイスタック）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デバイスをリモート監視できます。

図 7: RSPAN の設定例

下の図にデバイス A とデバイス B の送信元ポートを示します。各 RSPAN セッションのトラフィックは、ユーザーが指定した 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デバイスの RSPAN セッション専用です。送信元ポートまたは VLAN からの RSPAN トラフィックは RSPAN VLAN にコピーされ、RSPAN VLAN を伝送するトランクポートを介して、RSPAN VLAN を監視する宛先セッションに転送されます。各 RSPAN 送信元デバイスには、ポートまたは VLAN のいずれかが RSPAN 送信元として必要です。図中のデバイス C のように、宛先は常に物理ポートになります。



SPAN と RSPAN の概念および用語

SPAN セッション

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 VLANID ラベルが再設定され、通常のトランクポートを介して宛先デバイスに転送されます。

RSPAN 宛先セッションは RSPAN VLAN 上で受信されたすべてのパケットを取得し、VLAN のタグングを除去し、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)。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。
- デバイスは、単一セッション内でのローカル SPAN と RSPAN の併用をサポートしません。
 - RSPAN 送信元セッションにローカル宛先ポートを設定できません。
 - RSPAN 宛先セッションにローカル送信元ポートを設定できません。
 - 同じデバイスまたはデバイススタック上で、同じ RSPAN VLAN を使用する RSPAN 宛先セッションおよび RSPAN 送信元セッションを実行できません。

モニタ対象トラフィック

SPAN セッションは、次のトラフィック タイプを監視できます。

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

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

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

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

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

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

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

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されません。

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

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

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

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

送信元ポート

送信元ポート (別名モニター側ポート) 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ン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デバイスまたはデバイススタックに存在している必要があります。RSPAN セッションの場合は、RSPAN 宛先セッションを含むデバイス上にあります。RSPAN 送信元セッションのみを実行するデバイスまたはデバイススタックには、宛先ポートはありません。
- ポートを SPAN 宛先ポートとして設定すると、元のポート設定が上書きされます。SPAN 宛先設定を削除すると、ポートは以前の設定に戻ります。ポートが SPAN 宛先ポートとして機能している間にポートの設定が変更されると、SPAN 宛先設定が削除されるまで、変更は有効になりません。
- 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。
- デバイスまたはデバイススタックの宛先ポートの最大数は 64 です。

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

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

RSPAN VLAN

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

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

VLAN トランキンク 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、すべての中間デバイスを手動で設定する必要があります。

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

SPAN および RSPAN と他の機能の相互作用

SPAN は次の機能と相互に作用します。

- ルーティング：SPAN はルーテッドトラフィックを監視しません。VSPAN が監視するのはデバイスに出入りするトラフィックに限られ、VLAN間でルーティングされるトラフィックは監視しません。たとえば、VLAN が受信モニターされ、デバイスが別の VLAN から監視対象 VLAN にトラフィックをルーティングする場合、そのトラフィックは監視されず、SPAN 宛先ポートで受信されません。
- STP：SPAN または RSPAN セッションがアクティブな間、宛先ポートは STP に参加しません。SPAN または RSPAN セッションが無効になると、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、デバイス間で RSPAN VLAN のプルーニングが可能です。

- VLAN および トランキング：送信元ポート、または宛先ポートの VLAN メンバーシップまたは トランクの設定値を、いつでも変更できます。ただし、宛先ポートの VLAN メンバーシップまたは トランクの設定値に対する変更が有効になるのは、SPAN 宛先設定を削除してからです。送信元ポートの VLAN メンバーシップまたは トランクの設定値に対する変更は、ただちに有効になり、対応する SPAN セッションが変更に応じて自動的に調整されます。

- EtherChannel：EtherChannel グループを送信元ポートとして設定できます。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

スイッチのスタックは 1 つの論理スイッチを表すため、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スイッチである場合があります。したがって、スタック内

でのスイッチ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。スイッチ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、スイッチ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 7: 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

SPAN および RSPAN の設定

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 {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で RSPAN がサポートされている。

FSPAN および FRSPAN 設定時の注意事項

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

SPAN および RSPAN の設定方法

ここでは、SPAN および RSPAN の設定方法について説明します。

ローカル SPAN セッションの作成

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • 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	<p>no monitor session {<i>session_number</i> all local remote}</p> <p>例 :</p> <pre>Device(config)# no monitor session all</pre>	<p>セッションに対する既存の SPAN 設定を削除します。</p> <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	<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>Device(config)# monitor session 1 source interface gigabitethernet1/0/1</pre>	<p>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。</p> <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 <i>port-channel-number</i>) 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)。 <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>

	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>
ステップ 5	monitor session <i>session_number</i> destination { interface <i>interface-id</i> [, -] [encapsulation { replicate dot1q }] } 例 : <pre>Device(config)# monitor session 1 destination interface gigabitethernet1/0/2 encapsulation replicate</pre>	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。 (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。 • <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。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • <i>interface-id</i> には、宛先ポートを指定します。 <p>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。</p> <ul style="list-style-type: none"> • (任意) [<i> </i>-] 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。 <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 <i>session_number destination</i> 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 :	(任意) コンフィギュレーションファイルに設定を保存します。

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

ローカル SPAN セッションの作成および着信トラフィックの設定

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> <code>enable</code>	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# <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	no monitor session { <i>session_number</i> all local remote } 例： Device(config)# <code>no monitor session all</code>	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> vlan <i>vlan-id</i> } [, -] [both rx tx] 例： Device(config)# <code>monitor session 2 source gigabitethernet1/0/1 rx</code>	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。

	コマンドまたはアクション	目的
ステップ 5	<p>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> }] }</p> <p>例 :</p> <pre>Device(config)# monitor session 2 destination interface gigabitethernet1/0/2 encapsulation replicate ingress dot1q vlan 6</pre>	<p>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。</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。 <p>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。</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 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。 • (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。 • ingress 宛先ポートでの着信トラフィックの転送をイネーブルにして、カプセル化タイプを指定します。 • dot1q vlan <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受け入れます。

	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 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 : Device(config)# end	特権 EXEC モードに戻ります。
ステップ 7	show running-config 例 : Device# show running-config	入力を確認します。
ステップ 8	copy running-config startup-config 例 : Device# copy running-config startup-config	(任意) コンフィギュレーションファイルに設定を保存します。

フィルタリングする VLAN の指定

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例 : Device> enable	特権 EXEC モードを有効にします。 <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。

	コマンドまたはアクション	目的
ステップ 3	<p>no monitor session {<i>session_number</i> all local remote}</p> <p>例 :</p> <pre>Device(config)# no monitor session all</pre>	<p>セッションに対する既存の SPAN 設定を削除します。</p> <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	<p>monitor session <i>session_number</i> source interface <i>interface-id</i></p> <p>例 :</p> <pre>Device(config)# monitor session 2 source interface gigabitethernet1/0/2 rx</pre>	<p>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。</p> <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。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>Device(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>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>

	コマンドまたはアクション	目的
	<p>例 :</p> <pre>Device(config)# monitor session 2 destination interface gigabitethernet1/0/1</pre>	<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) <i>[, -]</i> 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>Device(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>Device# 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>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。

RSPAN VLAN としての VLAN の設定

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • 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	vlan vlan-id 例： Device(config)# vlan 100	VLAN ID を入力して VLAN を作成するか、または既存の VLAN の VLAN ID を入力して、VLAN コンフィギュレーション モードを開始します。指定できる範囲は 2 ~ 1001 または 1006 ~ 4094 です。 RSPAN VLAN を VLAN 1（デフォルト VLAN）または VLAN ID 1002 ~ 1005（トークンリングおよび FDDI VLAN 専用）にすることはできません。
ステップ 4	remote-span 例： Device(config-vlan)# remote-span	VLAN を RSPAN VLAN として設定します。
ステップ 5	end 例： Device(config-vlan)# end	特権 EXEC モードに戻ります。
ステップ 6	show running-config 例： Device# show running-config	入力を確認します。

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

次のタスク

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

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

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

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

RSPAN 送信元セッションの作成

RSPAN 送信元セッションを作成および開始し、モニタ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

	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。

	コマンドまたはアクション	目的
ステップ 3	<p>no monitor session {<i>session_number</i> all local remote}</p> <p>例 :</p> <pre>Device(config)# no monitor session 1</pre>	<p>セッションに対する既存の SPAN 設定を削除します。</p> <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	<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>Device(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>

	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> : 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"> • <code>both</code> : 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。 • <code>rx</code> : 受信トラフィックをモニターします。 • <code>tx</code> : 送信トラフィックをモニターします。
ステップ 5	monitor session <i>session_number</i> destination remote vlan <i>vlan-id</i> 例 : <pre>Device(config)# monitor session 1 destination remote vlan 100</pre>	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。 <ul style="list-style-type: none"> • <code>session_number</code> には、ステップ 4 で指定した番号を入力します。 • <code>vlan-id</code> には、モニタリングする送信元 RSPAN VLAN を指定します。
ステップ 6	end 例 : <pre>Device(config)# end</pre>	特権 EXEC モードに戻ります。
ステップ 7	show running-config 例 : <pre>Device# show running-config</pre>	入力を確認します。
ステップ 8	copy running-config startup-config 例 :	(任意) コンフィギュレーションファイルに設定を保存します。

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

フィルタリングする VLAN の指定

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> <code>enable</code>	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# <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	no monitor session {<i>session_number</i> all local remote} 例： Device(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。 <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： Device(config)# <code>monitor session 2 source interface gigabitethernet1/0/2 rx</code>	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。<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

	コマンドまたはアクション	目的
		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 <i>session_number</i> filter vlan vlan-id [, -] 例 : <pre>Device(config)# monitor session 2 filter vlan 1 - 5 , 9</pre>	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"> • <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	monitor session <i>session_number</i> destination remote vlan <i>vlan-id</i> 例 : <pre>Device(config)# monitor session 2 destination remote vlan 902</pre>	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 : <pre>Device(config)# end</pre>	特権 EXEC モードに戻ります。
ステップ 8	show running-config 例 : <pre>Device# show running-config</pre>	入力を確認します。
ステップ 9	copy running-config startup-config 例 :	(任意) コンフィギュレーション ファ イルに設定を保存します。

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

RSPAN 宛先セッションの作成

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

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • 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	vlan vlan-id 例： Device(config)# vlan 901	送信元デバイスで作成された RSPAN VLAN の VLAN ID を指定し、VLAN コンフィギュレーションモードを開始します。 両方のデバイスが VTP に参加し、RSPAN VLAN ID が 2 ～ 1005 である場合は、VTP ネットワークを介して RSPAN VLAN ID が伝播されるため、ステップ 3 ～ 5 は不要です。
ステップ 4	remote-span 例： Device(config-vlan)# remote-span	VLAN を RSPAN VLAN として識別します。
ステップ 5	exit 例：	グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
	Device(config-vlan)# exit	
ステップ 6	<p>no monitor session {<i>session_number</i> all local remote}</p> <p>例 :</p> <pre>Device(config)# no monitor session 1</pre>	<p>セッションに対する既存の SPAN 設定を削除します。</p> <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 7	<p>monitor session <i>session_number</i> source remote vlan <i>vlan-id</i></p> <p>例 :</p> <pre>Device(config)# monitor session 1 source remote vlan 901</pre>	<p>RSPAN セッションと送信元 RSPAN VLAN 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。 • <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 8	<p>monitor session <i>session_number</i> destination interface <i>interface-id</i></p> <p>例 :</p> <pre>Device(config)# monitor session 1 destination interface gigabitethernet2/0/1</pre>	<p>RSPAN セッションと宛先インターフェイスを指定します。</p> <ul style="list-style-type: none"> • <i>session_number</i> には、ステップ 7 で指定した番号を入力します。 <p>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。</p> <ul style="list-style-type: none"> • <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

	コマンドまたはアクション	目的
		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： 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。

RSPAN 宛先セッションの作成および着信トラフィックの設定

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • 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 monitor session {session_number all local remote} 例：	セッションに対する既存の SPAN 設定を削除します。

	コマンドまたはアクション	目的
	<pre>Device(config)# no monitor session 2</pre>	<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	<p>monitor session <i>session_number</i> source remote vlan <i>vlan-id</i></p> <p>例 :</p> <pre>Device(config)# monitor session 2 source remote vlan 901</pre>	<p>RSPAN セッションと送信元 RSPAN VLAN 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。 • <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	<p>monitor session <i>session_number</i> destination {interface <i>interface-id</i> [, -] [ingress { dot1q <i>vlan</i> <i>vlan-id</i> untagged <i>vlan</i> <i>vlan-id</i> vlan <i>vlan-id</i> }]}</p> <p>例 :</p> <pre>Device(config)# monitor session 2 destination interface gigabitethernet1/0/2 ingress vlan 6</pre>	<p>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。</p> <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

	コマンドまたはアクション	目的
		<p>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。</p> <ul style="list-style-type: none"> 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 vlan-id : デフォルトの VLAN として指定した VLAN で、IEEE 802.1Q でカプセル化された着信パケットを転送します。 untagged vlan vlan-id または vlan vlan-id : デフォルトの VLAN として指定した VLAN で、タグなしでカプセル化された着信パケットを転送します。
ステップ 6	end 例 : Device(config)# end	特権 EXEC モードに戻ります。
ステップ 7	show running-config 例 : Device# show running-config	入力を確認します。
ステップ 8	copy running-config startup-config 例 : Device# copy running-config startup-config	(任意) コンフィギュレーションファイルに設定を保存します。

FSPAN セッションの設定

SPAN セッションを作成し、送信元 (監視対象) ポートまたは VLAN、および宛先 (モニター) 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

	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 { <i>session_number</i> all local remote } 例： Device(config)# no monitor session 2	セッションに対する既存の SPAN 設定を削除します。 <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> vlan <i>vlan-id</i> } [, -] [both rx tx] 例： Device(config)# monitor session 2 source interface gigabitethernet1/0/1	SPAN セッションおよび送信元ポート（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 <i>port-channel-number</i>）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）。

	コマンドまたはアクション	目的
		<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。 <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>
ステップ 5	monitor session <i>session_number</i> destination {interface <i>interface-id</i> [-] [encapsulation replicate]}	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。

	コマンドまたはアクション	目的
	<p>例 :</p> <pre>Device(config)# monitor session 2 destination interface gigabitethernet1/0/2 encapsulation replicate</pre>	<ul style="list-style-type: none"> • session_number には、ステップ 4 で入力したセッション番号を指定します。 • destination では、次のパラメータを指定します。 <ul style="list-style-type: none"> • interface-id には、宛先ポートを指定します。 <p>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。</p> • (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。 <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 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>

	コマンドまたはアクション	目的
ステップ 6	monitor session <i>session_number</i> filter {ip ipv6 mac} access-group {<i>access-list-number</i> <i>name</i>} 例 : Device(config)# monitor session 2 filter ipv6 access-group 4	SPANセッション、フィルタリングするパケットのタイプ、および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 7	end 例 : Device(config)# end	特権 EXEC モードに戻ります。
ステップ 8	show running-config 例 : Device# show running-config	入力を確認します。
ステップ 9	copy running-config startup-config 例 : Device# copy running-config startup-config	(任意) コンフィギュレーションファイルに設定を保存します。

FRSPAN セッションの設定

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例 : Device> enable	特権 EXEC モードを有効にします。 <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 : Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	no monitor session {session_number all local remote} 例 : Device(config)# no monitor session 2	セッションに対する既存の SPAN 設定を削除します。 <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 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"> • 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)。

	コマンドまたはアクション	目的
		<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_number source コマンドを複数回使用すると、複数の送信元ポートを設定できます。</p>
ステップ 5	<p>monitor session session_number destination remote vlan vlan-id</p> <p>例 :</p>	<p>RSPAN セッションと宛先 RSPAN VLAN 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
	Device(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 : Device(config)# vlan 10	VLAN コンフィギュレーションモードを開始します。 <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 : Device(config-vlan)# remote-span	ステップ 5 で指定した VLAN が RSPAN VLAN の一部であることを指定します。
ステップ 8	exit 例 : Device(config-vlan)# exit	グローバル コンフィギュレーションモードに戻ります。
ステップ 9	monitor session <i>session_number</i> filter {ip ipv6 mac} access-group {<i>access-list-number</i> <i>name</i>} 例 : Device(config)# monitor session 2 filter ip access-group 7	RSPAN セッション、フィルタリングするパケットのタイプ、および FRSPAN セッションで使用する ACL を指定します。 <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 : Device(config)# end	特権 EXEC モードに戻ります。
ステップ 11	show running-config 例 : Device# show running-config	入力を確認します。

	コマンドまたはアクション	目的
ステップ 12	copy running-config startup-config 例 : Device# copy running-config startup-config	(任意) コンフィギュレーションファイルに設定を保存します。

SPAN および RSPAN 動作のモニタリング

次の表で、SPAN および RSPAN 動作の設定と結果を表示して動作を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 8: SPAN および RSPAN 動作のモニタリング

コマンド	目的
show monitor	現在の SPAN、RSPAN を示します。

SPAN および RSPAN の設定例

次のセクションに SPAN および RSPAN の設定例を示します

例 : ローカル SPAN の設定

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

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

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

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

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

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

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

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

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source vlan 1 - 3 rx

Device(config)# monitor session 2 destination interface gigabitethernet1/0/2
Device(config)# monitor session 2 source vlan 10
Device(config)# end
```

次に、SPANセッション2の既存の設定を削除し、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。

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 2
Device(config)# monitor session 2 source gigabitethernet0/1 rx
Device(config)# monitor session 2 destination interface gigabitethernet0/2 encapsulation
  replicate ingress vlan 6
Device(config)# end
```

次に、SPANセッション2の既存の設定を削除し、トランクポート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。

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

例 : RSPAN VLAN の作成

この例は、RSPAN VLAN 901 の作成方法を示しています。

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

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

```
Device> enable
Device# configure terminal
Device(config)# no monitor session 1
Device(config)# monitor session 1 source interface gigabitethernet1/0/1 tx
Device(config)# monitor session 1 source interface gigabitethernet1/0/2 rx
Device(config)# monitor session 1 source interface port-channel 2
Device(config)# monitor session 1 destination remote vlan 901
Device(config)# end
```

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

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

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

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

次に、RSPAN セッション 2 で送信元リモート VLAN として VLAN 901 を設定し、送信元ポート GigabitEthernet2 を宛先インターフェイスとして設定し、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。

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

SPAN および RSPAN の機能履歴

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 16.5.1a	スイッチポートアナライザ (SPAN)	スイッチポートアナライザ：スニファヤアナライザまたは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。
	Flow-Based スイッチポートアナライザ (SPAN)	フローベースの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。
	スイッチポートアナライザ (SPAN)：分散イーグレスSPAN	この機能は、ラインカードにすでに分散されている入力SPANと組み合わせて、ラインカードに分散された出力SPAN機能を提供します。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。

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 にアクセスするには、<https://cfng.cisco.com/> に進みます。



第 7 章

ERSPAN の設定

- [ERSPAN の設定の前提条件](#) (133 ページ)
- [ERSPAN 設定時の制約事項](#) (133 ページ)
- [ERSPAN の設定に関する情報](#) (134 ページ)
- [ERSPAN の設定方法](#) (137 ページ)
- [ERSPAN の設定例](#) (147 ページ)
- [ERSPAN の確認](#) (148 ページ)
- [その他の参考資料](#) (151 ページ)
- [ERSPAN の機能の履歴](#) (151 ページ)

ERSPAN の設定の前提条件

- アクセスコントロールリスト (ACL) のフィルタ処理は、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。

ERSPAN 設定時の制約事項

この機能には、次の制限があります。

- 切り捨ては、IPv4 および IPv6 のパケットでのみサポートされ、IP ヘッダーのないレイヤ 2 パケットではサポートされません。
- ERSPAN 宛先インターフェイスは、1 つのセッションだけに使用することができます。同じ宛先インターフェイスを、複数の ERSPAN/SPAN セッションに設定することはできません。
- 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。
- filter IP/IPv6/MAC/VLAN access-group と filter SGT を同時に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 over MPLS VPN は、レイヤ 3 VPN、セグメントルーティング、およびシームレス MPLS でサポートされます。
- ERSPAN over MPLS VPN は、L2VPN、6PE、6VPE、MPLS over GRE、および InterAS ではサポートされません。
- MPLS コアスイッチを ERSPAN 宛先として設定することはできません。ERSPAN トラフィックは、1つのプロバイダーエッジ (PE) から別のプロバイダーエッジ (PE) に転送できます。2つのプロバイダーエッジ間のコアスイッチに転送できません。
- ERSPAN セッションでは、DHCP Inject パケットはキャプチャされません。
- ERSPAN セッションが有効になっているバックアップ設定が実行中の設定に復元された場合、ERSPAN セッションは無効な状態で自動的に作成されます。作成された ERSPAN セッションは手動で有効にする必要があります。

ERSPAN の設定に関する情報

ここでは、ERSPAN の設定について説明します。

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 送信元セッションのみが設定されているデバイスは、ERSPAN 送信元デバイス

と呼ばれます。ERSPAN宛先セッションだけが設定されているデバイスは、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。送信元セッションの数は、設定された ERSPAN 宛先セッションの数だけ減少します。

デバイスは、セッションごとに最大 50 のセキュリティグループタグ (SGT) フィルタをサポートできます。

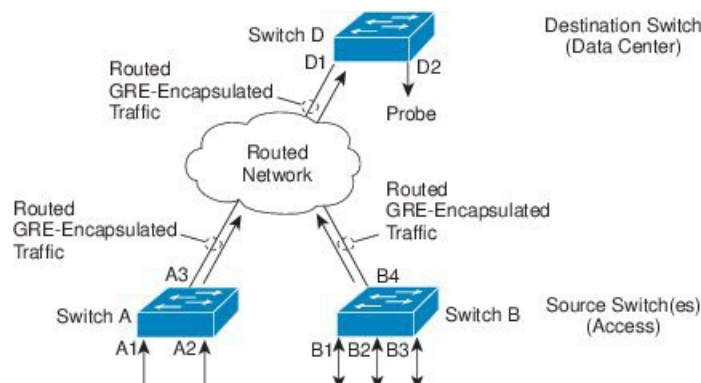
ERSPAN 送信元セッションは、次のパラメータによって定義されます。

- セッション ID。
- ERSPAN フロー ID。
- セッションによって監視される送信元ポートまたは送信元 VLAN の一覧。
- Generic Routing Encapsulation (GRE) エンベロープに関連する、IP Type of Service (ToS) や IP Time to Live (TTL) などのオプションの属性。
- 宛先および送信元 IP アドレス。これらは、キャプチャされたトラフィックの GRE エンベロープの宛先 IP アドレスと送信元 IP アドレスとしてそれぞれ使用されます。



- (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。
 - カプセル化およびカプセル化解除はハードウェアで実行されるため、CPU パフォーマンスは影響を受けません。
 - IPv4 および IPv6 の送信およびトランスポートヘッダーがサポートされています。Type-II および Type-III ヘッダーを含みます。

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 宛先ポート

宛先ポートは、ERSPAN 送信元が分析用のトラフィックを送信するレイヤ 2 LAN ポートまたはレイヤ 3 LAN ポートです。

ポートを宛先ポートとして設定すると、そのポートはトラフィックを受信することができなくなります。ポートは ERSpan 機能専用です。ERSPAN 宛先ポートでは、ERSPAN セッションに必要なトラフィック以外の転送は行われません。トランクポートを宛先ポートとして設定することができます。これによって、宛先トランクポートがカプセル化したトラフィックを転送することができます。

SGT ベースの ERSpan

セキュリティグループタグ (SGT) は、ログイン時に Cisco Identity Services Engine (ISE) がユーザーまたはエンドポイントセッションに割り当てる 16 ビット値です。ネットワーク インフラストラクチャでは、セッションに割り当てる別の属性として SGT が認識され、そのセッションからのすべてのトラフィックにレイヤ 2 タグが挿入されます。プラットフォームは、セッションあたり最大 50 の SGT ポリシーをサポートできます。

既存のフローベース SPAN (FSPAN) または VLAN フィルタセッションでは、SGT フィルタリング設定は許可されていません。

ERSPAN タイムスタンプ

ERSPAN ヘッダーがタイプ III に設定されている場合、ERSPAN タイムスタンプは自動的に有効になります。タイムスタンプフィールドは、デバイスのパケット遅延を計算するために使用されます。ERSPAN 送信元セッションでは、パケットを受信するとタイムスタンプフィールドにローカル時間情報を入力されます。宛先セッションは、このタイムスタンプをアプリケーションに渡すことができます。ERSPAN は、32 ビット形式のすべてのタイムスタンプをサポートします。100 ナノ秒 (ns) の粒度をサポートし、タイムスタンプフィールドのラップアラウンド時間は約 7 分です。

ERSPAN over MPLS VPN

Cisco IOS XE Bengaluru 17.5.x リリース以降、マルチプロトコルラベルスイッチング (MPLS) バーチャルプライベートネットワーク (VPN) を介して ERSPAN トラフィックを転送できるようになりました。ERSPAN over MPLS VPN を有効にするには、ネットワークでマルチプロトコルラベルスイッチング (MPLS)、ラベル配布プロトコル (LDP)、および Cisco Express Forwarding を有効にする必要があります。

ERSPAN 宛先を設定して、MPLS VPN を介した ERSPAN トラフィックの送信元 VRF を選択できます。ERSPAN destination session source コマンドで **vrf** キーワードを使用して、送信元 VRF を設定できます。

ERSPAN の設定方法

ここでは、ERSPAN の設定方法について説明します。

ERSPAN 送信元セッションの設定 (IPv4)

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。IPv4 ERSPAN 送信元セッションを定義するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 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	<p>monitor session <i>span-session-number</i> type erspan-source</p> <p>例 :</p> <pre>Device(config)# monitor session 1 type erspan-source</pre>	<p>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。</p> <ul style="list-style-type: none"> • <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、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 4	<p>description <i>string</i></p> <p>例 :</p> <pre>Device(config-mon-erspan-src)# description source1</pre>	<p>(任意) ERSPAN 送信元セッションの説明を入力します。</p> <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 5	<p>[no] header-type 3</p> <p>例 :</p> <pre>Device(config-mon-erspan-src)# header-type 3</pre>	<p>(任意) スイッチをタイプ III ERSPAN ヘッダーに設定します。デフォルトタイプはタイプ II ERSPAN ヘッダーです。</p>
ステップ 6	<p>source {interface interface-type interface-number vlan vlan-id} [, - both rx tx]</p> <p>例 :</p> <pre>Device(config-mon-erspan-src)# source interface fastethernet 0/1 rx</pre>	<p>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。</p>

	コマンドまたはアクション	目的
ステップ 7	<p>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> sgt <i>sgt-ID</i> [, -] vlan <i>vlan-ID</i> [, -]}</p> <p>例 :</p> <pre>Switch(config-mon-erspan-src)# filter vlan 3</pre>	<p>(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。 filter sgt sgt-ID コマンドは、ERSPAN 送信元セッションで SGT フィルタリングを設定します。</p> <p>(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。</p>
ステップ 8	<p>destination</p> <p>例 :</p> <pre>Device(config-mon-erspan-src)# destination</pre>	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	<p>erspan-id <i>erspan-flow-id</i></p> <p>例 :</p> <pre>Device(config-mon-erspan-src-dst)# erspan-id 100</pre>	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	<p>ip address <i>ip-address</i></p> <p>例 :</p> <pre>Device(config-mon-erspan-src-dst)# ip address 10.1.0.2</pre>	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	<p>ip dscp <i>dscp-value</i></p> <p>例 :</p> <pre>Device(config-mon-erspan-src-dst)# ip dscp 10</pre>	(任意) 回線エミュレーション (CEM) チャンネルからのパケットに対して IP DiffServ コードポイント (DSCP) の使用をイネーブルにします。
ステップ 12	<p>ip ttl <i>ttl-value</i></p> <p>例 :</p> <pre>Device(config-mon-erspan-src-dst)# ip ttl 32</pre>	(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 13	<p>mtu <i>mtu-size</i></p> <p>例 :</p> <pre>Device(config-mon-erspan-src-dst)# mtu 512</pre>	MTU の切り捨てサイズを設定します。設定された MTU サイズよりも大きい ERSPAN パケットはすべて、設定されたサイズに切り捨てられます。MTU サイズの範囲は、176 ~ 9000 バイトです。デフォルト値は 9000 バイトです。

	コマンドまたはアクション	目的
ステップ 14	origin ip-address ip-address 例： Device(config-mon-erspan-src-dst)# origin ip address 10.10.0.1	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。
ステップ 15	vrf vrf-id 例： Device(config-mon-erspan-src-dst)# vrf 1	(任意) グローバルルーティングテーブルの代わりに使用する VRF 名を設定します。
ステップ 16	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 17	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 18	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 宛先セッションの設定 (IPv4)

ERSPAN 宛先セッションは、セッション設定パラメータとモニター対象トラフィックを受信するポートを定義します。IPv4 ERSPAN 宛先セッションを定義するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 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	<p>monitor session session-number type erspan-destination</p> <p>例 :</p> <pre>Device(config)# monitor session 1 type erspan-destination</pre>	<p>セッションIDとセッションタイプを使用してERSPAN宛先セッションを定義し、ERSPANのモニター宛先セッションコンフィギュレーションモードを開始します。</p> <ul style="list-style-type: none"> • <i>session-number</i> 引数の範囲は1～66です。セッション番号は一意である必要があり、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>session-number</i> 引数によって設定) およびセッションタイプ (<i>erspan-destination</i> によって設定) は、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 4	<p>description string</p> <p>例 :</p> <pre>Device(config-mon-erspan-dst)# description source1</pre>	<p>(任意) ERSPAN 宛先セッションの説明を入力します。</p> <ul style="list-style-type: none"> • <i>string</i> 引数には最大240文字まで入力できますが、特殊文字やスペースを含めることはできません。
ステップ 5	<p>destination interface interface-type interface-number</p> <p>例 :</p> <pre>Device(config-mon-erspan-dst)# destination interface GigabitEthernet1/0/1</pre>	ERSPAN 宛先セッション番号を送信元ポートに関連付け、モニターするトラフィックの方向を選択します。
ステップ 6	<p>source</p> <p>例 :</p> <pre>Device(config-mon-erspan-dst)# source</pre>	ERSPAN 宛先セッションの送信元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 7	erspan-id <i>erspan-flow-id</i> 例 : Device(config-mon-erspan-dst-src) # 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 送信元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 8	ip address <i>ip-address</i> [force] 例 : Device(config-mon-erspan-dst-src) # 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。 <ul style="list-style-type: none"> • この IP アドレスは、ローカルインターフェイスまたはループバックインターフェイスのアドレスであり、宛先スイッチのアドレスと一致する必要があります。 • ip address ip-address force コマンドは、すべての ERSPAN 宛先セッションの宛先 IP アドレスを変更します。
ステップ 9	vrf <i>vrf-id</i> 例 : Device(config-mon-erspan-dst-src) # vrf 1	(任意) グローバルルーティングテーブルの代わりに使用する VRF 名を設定します。
ステップ 10	no shutdown 例 : Device(config-mon-erspan-dst-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 11	end 例 : Device(config-mon-erspan-dst-src) # end	ERSPAN 宛先セッション送信元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 送信元セッションの設定 (IPv6)

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。IPv6 ERSPAN 送信元セッションを定義するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例 : Device> enable	特権 EXEC モードを有効にします。 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	monitor session <i>session-number</i> type erspan-source 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。 <ul style="list-style-type: none"> • <i>span-session-number</i> 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、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 4	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。

	コマンドまたはアクション	目的
ステップ 5	[no] header-type 3 例： Device(config-mon-erspan-src)# header-type 3	(任意) スイッチをタイプ III ERSPAN ヘッダーに設定します。デフォルトタイプはタイプ II ERSPAN ヘッダーです。
ステップ 6	source {interface interface-type interface-number vlan vlan-id} [, - both rx tx] 例： Device(config-mon-erspan-src)# source interface fortygigabitethernet 1/0/3	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 {standard-access-list expanded-access-list acl-name } ipv6 access-group acl-name mac access-group acl-name sgt sgt-ID [, -] vlan vlan-ID [, -]} 例： Switch(config-mon-erspan-src)# filter ipv6 access-group exampleacl	(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。 filter sgt sgt-ID コマンドは、ERSPAN 送信元セッションで SGT 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 erspan-flow-id 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	ipv6 address ipv6-address 例： Device(config-mon-erspan-src-dst)# ipv6 address 2001:DB8::1	ERSPAN トラフィックの宛先として使用される IPv6 アドレスを設定します。
ステップ 11	ipv6 dscp dscp-value 例： Device(config-mon-erspan-src-dst)# ipv6 dscp 2	(任意) 回線エミュレーション (CEM) チャネルからのパケットに対して IPv6 DiffServ コードポイント (DSCP) の使用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 12	ipv6 ttl <i>ttl-value</i> 例 : Device (config-mon-erspan-src-dst) # ipv6 ttl 4	(任意) ERSPAN トラフィック内のパケットの IPv6 TTL 値を設定します。
ステップ 13	mtu <i>mtu-size</i> 例 : Device (config-mon-erspan-src-dst) # mtu 512	MTU の切り捨てサイズを設定します。設定された MTU サイズよりも大きい ERSPAN パケットはすべて、設定されたサイズに切り捨てられます。MTU サイズの範囲は、176 ~ 9000 バイトです。デフォルト値は 9000 バイトです。
ステップ 14	origin ipv6-address <i>ipv6-address</i> 例 : Device (config-mon-erspan-src-dst) # origin ipv6 address 2001:DB8:1::1	ERSPAN トラフィックの送信元として使用される IPv6 アドレスを設定します。
ステップ 15	vrf <i>vrf-id</i> 例 : Device (config-mon-erspan-src-dst) # vrf 1	(任意) グローバルルーティングテーブルの代わりに使用する VRF 名を設定します。
ステップ 16	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 17	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 18	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 宛先セッションの設定 (IPv6)

ERSPAN 宛先セッションは、セッション設定パラメータとモニター対象トラフィックを受信するポートを定義します。IPv6 ERSPAN 宛先セッションを定義するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 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	monitor session session-number type erspan-destination 例： Device(config)# monitor session 3 type erspan-destination	セッション ID とセッションタイプを使用して ERSPAN 宛先セッションを定義し、ERSPAN のモニター宛先セッションコンフィギュレーションモードを開始します。 <ul style="list-style-type: none"> • <i>session-number</i> 引数の範囲は 1 ～ 66 です。セッション番号は一意である必要があります、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>session-number</i> 引数によって設定) およびセッションタイプ (erspan-destination に よって設定) は、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 4	description string 例： Device(config-mon-erspan-dst)# description source 1	(任意) ERSPAN 宛先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 文字まで入力できますが、特殊文字やスペースを含めることはできません。

	コマンドまたはアクション	目的
ステップ 5	destination interface <i>interface-type</i> <i>interface-number</i> 例 : Device (config-mon-erspan-dst) # destination interface fortygigabitethernet 1/0/3	ERSPAN 宛先セッション番号を送信元ポートに関連付け、モニターするトラフィックの方向を選択します。
ステップ 6	source 例 : Device (config-mon-erspan-dst) # source	ERSPAN 宛先セッションの送信元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	erspan-id <i>erspan-flow-id</i> 例 : Device (config-mon-erspan-dst-src) # 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 送信元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 8	ipv6 address <i>ipv6-address</i> 例 : Device (config-mon-erspan-dst-src) # ip address 2001:DB8::1	ERSPAN トラフィックの宛先として使用される IPv6 アドレスを設定します。 この IPv6 アドレスは、ローカルインターフェイスまたはループバックインターフェイスのアドレスであり、宛先スイッチのアドレスと一致する必要があります。
ステップ 9	exit 例 : Switch (config-mon-erspan-dst-src) #exit	ERSPAN 宛先セッション送信元コンフィギュレーションモードを終了し、ERSPAN 宛先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 10	no shutdown 例 : Device (config-mon-erspan-dst) # 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 11	end 例 : Device (config-mon-erspan-dst) # end	ERSPAN 宛先セッション送信元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

次のセクションに ERSPAN の設定例を示します。

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 GigabitEthernet 1/0/1 rx
Device(config-mon-erspan-src)# source interface GigabitEthernet 1/0/4 - 8 tx
Device(config-mon-erspan-src)# source interface GigabitEthernet 1/0/3
Device(config-mon-erspan-src)# destination
Device(config-mon-erspan-src-dst)# erspan-id 100
Device(config-mon-erspan-src-dst)# ip address 10.1.0.2
Device(config-mon-erspan-src-dst)# ip dscp 10
Device(config-mon-erspan-src-dst)# ip ttl 32
Device(config-mon-erspan-src-dst)# mtu 512
Device(config-mon-erspan-src-dst)# origin ip address 10.10.0.1
Device(config-mon-erspan-src-dst)# vrf monitoring
Device(config-mon-erspan-src-dst)# exit
Device(config-mon-erspan-src)# no shutdown
Device(config-mon-erspan-src)# end
```

例 : ERSpan 宛先セッションの設定

次に、ERSpan 宛先セッションを設定する例を示します。

```
Device(config)# monitor session 2 type erspan-destination
Device(config-mon-erspan-dst)# destination interface GigabitEthernet1/3/2
Device(config-mon-erspan-dst)# destination interface GigabitEthernet2/2/0
Device(config-mon-erspan-dst)# source
Device(config-mon-erspan-dst-src)# erspan-id 100
Device(config-mon-erspan-dst-src)# ip address 10.1.0.2
```

次の例は、ERSpan 宛先セッションの送信元 VRF を設定する方法を示しています。

```
Device(config)# monitor session 2 type erspan-destination
Device(config-mon-erspan-dst)# destination interface GigabitEthernet1/3/2
Device(config-mon-erspan-dst)# destination interface GigabitEthernet2/2/0
Device(config-mon-erspan-dst)# source
Device(config-mon-erspan-dst-src)# erspan-id 100
Device(config-mon-erspan-dst-src)# ip address 10.1.0.2
Device(config-mon-erspan-dst-src)# vrf 1
```

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** コマンドの出力例を示します。

```
Device# show monitor session 53
```



```

Session 53
-----
Type                : ERSPAN Source Session
Status              : Admin Enabled
Source Ports        :
MTU                 : Fo1/0/2

```

次に、**show platform software monitor session** コマンドの出力例を示します。

```

Device# show platform software monitor session 53

Span Session 53 (FED Session 0):
Type: ERSPAN Source
Prev type: Unknown
Ingress Src Ports:
Egress Src Ports:
Ingress Local Src Ports: (null)
Egress Local Src Ports: (null)
Destination Ports:
Ingress Src Vlans:
Egress Src Vlans:
Ingress Up Src Vlans: (null)
Egress Up Src Vlans: (null)
Src Trunk filter Vlans:
RSPAN dst vlan: 0
RSPAN src vlan: 0
RSPAN src vlan sav: 0
Dest port encap = 0x0000
Dest port ingress encap = 0x0000
Dest port ingress vlan = 0x0
SrcSess: 1 DstSess: 0 DstPortCfgr: 0 RspnDstCfgr: 0 RspnSrcVld: 0
DstCliCfgr: 0 DstPrtInit: 0 PsLclCfgr: 0
Flags: 0x00000000
Remote dest port: 0 Dest port group: 0
FSPAN disabled
FSPAN not notified
ERSPAN Id : 0
ERSPAN Org Ip: 0.0.0.0
ERSPAN Dst Ip: 0.0.0.0
ERSPAN Ip Ttl: 255
ERSPAN DSCP : 0
ERSPAN MTU : 1500 >>>>
ERSPAN VRFID : 0
ERSPAN State : Disabled
ERSPAN Tun id: 61
ERSPAN header-type: 2
ERSPAN SGT :

```

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         : None
    TX Only         : None
    Both            : None
Source Subinterfaces :
    RX Only         : None
    TX Only         : None
    Both            : None

```

```

Source VLANs          :
  RX Only              : None
  TX Only              : None
  Both                 : None
Source Drop-cause     : None
Source EFPs           :
  RX Only              : None
  TX Only              : None
  Both                 : None
Source RSPAN VLAN     : None
Destination Ports     : None
Filter VLANs          : None
Filter SGT             : None
Dest RSPAN VLAN       : None
IP Access-group       : None
MAC Access-group      : None
IPv6 Access-group     : None
Filter access-group   :None
smac for wan interface : None
dmac for wan interface : None
Destination IP Address : 192.0.2.1
Destination IPv6 Address : None
Destination IP VRF    : None
MTU                   : 1500
Destination ERSPAN ID : 251
Origin IP Address     : 10.10.10.216
Origin IPv6 Address   : None
IP QOS PREC           : 0
IPv6 Flow Label       : None
IP TTL                : 255
ERSPAN header-type    : 3

```

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: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
SGT 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:ERSPAN Destination Session Supported: TRUE
Maximum No of ERSPAN destination session: 8
ERSPAN Header Type supported: II and III

```

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 の機能の履歴

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 16.5.1a	ERSPAN	<p>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。</p> <p>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。</p> <p>この機能のサポートは、Cisco Catalyst 9500 シリーズスイッチの C9500-12Q、C9500-16X、C9500-24Q、C9500-40X モデルでのみ導入されました。</p>
Cisco IOS XE Gibraltar 16.11.1	ERSPAN : 宛先セッションのサポート	ERSPAN 宛先セッションは、セッション設定パラメータとモニター対象トラフィックを受信するポートを定義します。
Cisco IOS XE Amsterdam 17.1.1	ERSPAN IPv6	ERSPAN に IPv6 サポートが導入されました。これにより、IPv6 ERSPAN 送信元および宛先セッションの設定が可能になります。
Cisco IOS XE Bengaluru 17.5.1	ERSPAN over MPLS VPN	<p>MPLS VPN サポートが ERSPAN に導入されました。</p> <p>ERSPAN トラフィックは MPLS VPN 経由で転送できます。</p> <p>ERSPAN のサポートは、L3VPN IPv4 MPLS に限定されています。</p>

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 にアクセスするには、<https://cfngn.cisco.com/> に進みます。



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](#) (153 ページ)
- [パケットキャプチャ設定の制約事項](#) (154 ページ)
- [パケットキャプチャについて](#) (157 ページ)
- [パケットキャプチャの設定方法](#) (168 ページ)
- [パケットキャプチャの設定例](#) (186 ページ)
- [その他の参考資料](#) (204 ページ)
- [パケットキャプチャの機能履歴](#) (205 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 キャプチャ中の CPU 使用率は、指定された条件に一致するパケットの数によって異なります。また、一致したパケットの意図されたアクション (保存、複合化、表示、またはその両方) によっても異なります。

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。次の表は、システムリソースを使用するためのガイドラインを示しています。

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	DRAMはパケットバッファを保存します。パケットバッファのサイズは、ユーザーが指定します。
ディスク容量	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 は、グローバルパケットキャプチャをサポートしていません。
- Wireshark は、ファイルサイズによる循環ファイルストレージの制限をサポートしていません。
- アクティブなキャプチャセッションで使用されているファイルを削除すると、キャプチャセッションで新しいファイルを作成できなくなります。キャプチャされた以降のパケットはすべて失われます。キャプチャポイントを再起動する必要があります。
- 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、手動で再起動します。
- ストリーミングキャプチャモードは約 1000 pps をサポートし、ロックステップモードは約 2 Mbps（256 バイトパケットで測定）をサポートします。一致するトラフィックレートがこの値を超えると、パケット損失が発生する可能性があります。
- キャプチャがアクティブな場合、キャプチャポイントを変更することはできません。
- 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、または MAC）のみが許可されません。

- ACL ロギングおよび Wireshark には互換性がありません。Wireshark を有効にすると、それが優先されます。ポート上の ACL ロギングによってキャプチャされたトラフィックを含むすべてのトラフィックは、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。
- パケットキャプチャのフィルタとしてのパケット長の範囲は、非 IPv4/IPv6 パケットおよびフラグメント化されたパケットではサポートされません。
- フィルタとしてのパケット長の範囲は、他のフィルタとともに使用できません。

組み込みパケットキャプチャの設定の制約事項

- レイヤ2 EtherChannels はサポートされません。
- VRF、管理ポート、およびプライベート VLAN を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。
- シャットダウン状態の VLAN インターフェイスは EPC をサポートしていません。
- インターフェイスを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）に、またはその逆へ変更する場合、インターフェイスが復旧したら、キャプチャポイントを削

除して新しいキャプチャポイント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で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加えられた変更を反映しない場合があります。これには、TTL、VLAN タグ、CoS、チェックサム、MAC アドレス、DSCP、優先順位、UP などが含まれます。
- 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キャプチャを制限するには、フィルタを使用してください。
- DNA Advantage は、ワイヤレスアクセスポイントの制御とプロビジョニング (CAPWAP) などのプロトコルの複合化をサポートしています。
- 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。もう一方を開始する前に、一方を停止してください。
- 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インターフェイスでレイヤ2パケット (ARP) をキャプチャできません。
- VACL は IPv6 ベースの ACL をサポートしません。
- EPCは、MPLS パケットの基礎となるルーティングプロトコルに基づいてキャプチャすることはできません。

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、ネットワーク管理者がデバイスに出入りするパケットをキャプチャできるオンボードパケットキャプチャ機能です。Wireshark や Embedded Packet Capture (EPC) などのツールを使用して、ローカルで分析したり、オフライン分析用に保存してエクスポートしたりできます。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、Wireshark の特定インスタンスに関連付けられているすべての特性を示します。これには、キャプチャするパケット、キャプチャする場所、キャプチャされたパケットの処理方法、停止するタイミングが含まれます。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も識別され、手動または自動で非アクティブ化または停止する場合もあります。

複数のキャプチャポイントを定義できますが、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。スイッチオーバーにより、アクティブなパケットキャプチャセッションが終了するため、セッションを再開する必要があります。

接続ポイント

接続ポイントは、キャプチャポイントに関連付けられた論理パケットのプロセスパスのポイントです。接続ポイントはキャプチャポイントの属性です。接続ポイントに影響するパケットはキャプチャポイントフィルタに対してテストされます。一致するパケットはキャプチャポ

イントの関連する Wireshark インスタンスにコピーされ、送信されます。特定のキャプチャポイントに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せん。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、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て制限するキャプチャポイントの属性です。これらはコピーされ、Wireshark にパスされます。Wireshark では、パケットが接続ポイントを通過する場合、パケットと、キャプチャポイントに関連付けられているすべてのフィルタが表示されます。

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を 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：Wireshark ではキャプチャフィルタが適用されます。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したものの、キャプチャフィルタを通過しなかったパケットはコピーされません。それらは CPU/ソフトウェアに送信されますが、Wireshark プロセスによって破棄されます。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：Wireshark では表示フィルタが適用されます。その一致基準は、キャプチャフィルタの基準に似ています。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 機能の一部ではありません。

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

ライブトラフィックまたは既存の .pcap ファイルで Wireshark を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にキャプチャします。
- .pcap ファイルへ保存
- デコードおよび表示
- 保存および表示

デコードおよび表示アクションは、.pcap ファイルで呼び出された場合にのみ適用されます。

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

パケットをメモリのキャプチャバッファに保存できます。パケットは、後続の複合化、分析、または .pcap ファイルへの保存に使用できます。

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より古いほうの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使用されます。ただし、これを削除せずに、バッファのコンテンツをクリア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) スタック内のスイッチで WireShark を使用する場合、アクティブなスイッチに接続されているフラッシュまたは USB フラッシュデバイスにのみパケットキャプチャを保存できます。
- 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witch 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あります。

- 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スイッチに接続されていないデバイスにパケットキャプチャを保存しようとすると、エラーが発生します。

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、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存在する場合、Wireshark はファイルを上書きできるかどうかを確認します。キャプチャポイントの有効化時にファイルが存在する場合、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関連付けることができるキャプチャポイントは1つだけです。

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。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スペースがあることを確認してください。

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 バイトまたは 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。キャプチャポイントが .pcap ファイルに関連付けられている場合は、メモリバッファサイズを指定できます。

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。

- 要約：パケット（デフォルト）ごとに 1 行を表示します。
- 詳細：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

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を使用して Wireshark キャプチャポイントを定義したら、アクティブにする必要があります。キャプチャポイントをアクティブにするまで、実際にはパケットをキャプチャしません。

キャプチャポイントをアクティブにする前に、一部の機能性チェックが実行されます。コアシステムフィルタまたは接続ポイントが定義されていない場合、キャプチャポイントをアクティ

ブにすることはできません。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、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 キャプチャポイントをアクティブにした後、複数の方法で非アクティブにすることができます。パケットのみを .pcap ファイルに保存しているキャプチャポイントを手動で停止できます。また、時間またはパケットの制限を設定して設定することもできます。その後、キャプチャポイントは自動的に停止します。

Wireshark キャプチャポイントをアクティブにすると、固定レートポリサーがハードウェアで自動的に適用されます。これにより、CPU が Wireshark に指示されたパケットでフラッディングされなくなります。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る場合、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と Wireshark を入力キャプチャに適用した場合、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 によって引き続きキャプチャされます。入力キャプチャにポートセキュリティを適用し、出力キャプチャに Wireshark を適用した場合、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 によってキャプチャされません。
- Wireshark は、Dynamic ARP Inspection (DAI) によってドロップされたパケットをキャプチャしません。
- STP ブロックステートにあるポートを接続ポイントとして使用し、コアフィルタが一致する場合、パケットがスイッチによってドロップされても、Wireshark はポートに着信す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 を変更する場合は常に、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場合、変更されていないかのように元の ACL が引き続き使用されます。
- フラッシュディスクへの書き込みは、CPU を集中的に使用する操作です。キャプチャレートが不十分な場合は、バッファキャプチャを使用することをお勧めします。
- 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。
 - ライブパケットのキャプチャ中には、ストアのみ（表示オプションを指定しない場合）を使用します。CPU に負荷がかかる操作（特に詳細モード）である複合化と表示には使用しないでください。
 - 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複合化して表示する場合は、短いキャプチャ期間で Wireshark セッションをバインドしていることを確認してください。
- 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。

- キャプチャポイントを定義する場合、特定の順序は適用されません。CLI で許可されている場合は、キャプチャ ポイント パラメータを任意の順序で定義できます。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にのみ Wireshark が開始します。
- キャプチャポイントの作成時にファイルが存在する場合、Wireshark はファイルを上書きできるかどうかを確認します。キャプチャポイントのアクティブ化時にファイルが存在する場合、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、自身を自動的に終了することができます。内部エラーが発生した場合、またはリソースがフルになった場合（特に、ディスクがファイルモードでフルの場合）に終了することがあります。
- 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が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パケットをフィルタ処理できます。さらに、最大パケットキャプチャレートを指定するか、サンプリング間隔を指定することで、制御を定義できます。

Cisco IOS XE Amsterdam 17.2.1 以前では、EPC はシャットダウン状態のインターフェイスではサポートされていません。Cisco IOS XE Amsterdam 17.2.1 以降は、EPC はシャットダウン状態のインターフェイスでサポートされます。これは、インターフェイスの起動時にパケットをキャプチャする場合に便利です。

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を、機能するキャプチャポイントを持つよう定義します。

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。使用できません。

キャプチャ ポイントを定義するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 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</i> <i>interface-id</i> control-plane } { in out both } 例：	キャプチャ ポイントを定義し、キャプチャ ポイントが関連付けられている接続ポイントを指定し、キャプチャの方向を指定します。 キーワードの意味は次のとおりです。

	コマンドまたはアクション	目的
	<pre>Device# monitor capture mycap interface GigabitEthernet1/0/1 in</pre>	<ul style="list-style-type: none"> • <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 <i>mycap</i> 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 (<code>_</code>) のみが許可されます • (任意) <i>interfaceinterface-type</i> <i>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 <i>GigabitEthernet1/0/1</i> が使用されています)。 <p>(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> <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 <i>GigabitEthernet</i> 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。 <p>(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。</p>

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • capwap : 接続ポイントを CAPWAP トンネルとして指定します。 <li style="margin-left: 2em;">(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、コアフィルタを使用することはできません。 • (任意) control-plane : 接続ポイントとしてコントロールプレーンを指定します。 • in out both : キャプチャの方向を指定します。
<p>ステップ 3</p>	<p>monitor capture {<i>capture-name</i>} [match {<i>any ipv4 any any ipv6</i>} <i>any any</i>}]</p> <p>例 :</p> <pre>Device# monitor capture mycap interface GigabitEthernet1/0/1 in match any</pre>	<p>コアシステムのフィルタを定義します。</p> <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。

	コマンドまたはアクション	目的
		<p>(注) キャプチャポイントにコアシステムフィルタまたは接続ポイントが定義されていない場合、それをアクティブにすることはできません。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。
<p>ステップ 4</p>	<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>
<p>ステップ 5</p>	<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>
<p>ステップ 6</p>	<p>show running-config</p> <p>例 :</p> <pre>Device# show running-config</pre>	<p>入力を確認します。</p>

	コマンドまたはアクション	目的
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

接続ポイントをさらに追加し、キャプチャポイントのパラメータを変更してから、アクティブ化できます。キャプチャポイントをそのまま使用する場合は、有効化できます。



(注) このトピックで説明されている方法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。

ユーザーが間違ったキャプチャ名を入力した場合、または無効または存在しない接続ポイントを入力した場合、スイッチはエラーを表示します。たとえば、「キャプチャ名は8文字以下である必要があります。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

順番にリストされていますが、ステップを実行してパラメータの値を任意の順序で指定できます。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。すでに指定されている特定のパラメータが変更された場合は、インタラクティブに確認する必要があります。

Cisco IOS XE Amsterdam 17.3.x リリース以降では、パケット長の範囲とイーサタイプをパケットキャプチャのパラメータとして使用できます。

キャプチャポイントのパラメータを変更するには、次の手順に従います。

始める前に

これらの手順を使用する前に、キャプチャポイントを定義する必要があります。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • 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}} 例： Device# monitor capture mycap match ipv4 any any	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。 パケットキャプチャのパケット長の範囲を指定するには、EXEC コンフィギュレーションモードで monitor capture capture-name interface interface-id {in out both} match pktlen-range max packet-length-in-bytes min packet-length-in-bytes コマンドを使用します。

	コマンドまたはアクション	目的
		ACL を使用してコアシステムフィルタを定義できます。ACL でプロトコルの Ethertype を設定できます。Wireshark で同じ ACL を設定して、特定の Ethertype を持つパケットのキャプチャを有効にすることができます。
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) ファイルが存在する場合、上書き可能か確認してください。
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 :	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	Device(config)# end	

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
```

パケットキャプチャのパケット長の範囲の指定

```
Device(config)# monitor capture cap1 interface FortyGigabitEthernet 1/0/1 in match
pktlen-range max 100 min 50
```

パケットのイーサタイプの指定

```
MAC ACL:
Device(config)#mac access-list extended macl
Device(config-ext-macl)#permit any any 0x806 0x0
Device(config-ext-macl)exit
Device(config)#monitor capture mycap access-list macl

IP ACL:
Device#ip access-list extended ip1
Device(config-ext-nacl)#permit 1 any any icmp-message-type
Device(config-ext-nacl)# exit
Device#monitor capture mycap access-list ip1
```

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

順番にリストされていますが、パラメータを削除する手順は任意の順序で実行できます。1行、2行、または複数行で削除できます。複数が必要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、次の手順に従います。

始める前に

これらの手順を使用してキャプチャポイントを削除する前に、キャプチャポイントのパラメータを定義してください。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • 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： Device# no monitor capture mycap limit duration packet-len Device# no monitor capture mycap limit	Wireshark が保持しているセッション時間制限とパケットセグメント長を削除します。その他の指定された制限はそのままになります。 Wireshark のすべての制限をクリアします。
ステップ 4	no monitor capture {capture-name} file [location] [buffer-size] 例： Device# no monitor capture mycap file location Device# no monitor capture mycap file location	ファイルの関連付けを削除します。キャプチャポイントはパケットをキャプチャしなくなります。それらは表示されるのみです。 ファイル位置の関連付けを削除します。ファイルの場所は、キャプチャポイントに関連付けられなくなりました。ただし、他の定義されたファイルの関連付けは、このアクションの影響を受けません。
ステップ 5	show monitor capture {capture-name} [parameter] 例： Device# show monitor capture mycap parameter monitor capture mycap interface GigabitEthernet1/0/1 in	パラメータの削除操作後にまだ定義されているキャプチャポイントパラメータを表示します。手順のどの段階でもこのコマンドを実行して、キャプチャポイントに関連付けられているパラメータを確認できます。
ステップ 6	end 例： Device(config)# end	特権 EXEC モードに戻ります。

次のタスク

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



(注) キャプチャポイントがアクティブなときにパラメータを削除すると、スイッチに「*Capture is active*」というエラーが表示されます。

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

これらの手順を使用して削除する前に、キャプチャポイントを定義します。キャプチャポイントを削除する前に停止してください。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • 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が削除されたため、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。

	コマンドまたはアクション	目的
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。キャプチャポイントの定義を最初からやり直す場合は、これらの手順を実行できます。

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

接続ポイントとコアシステムフィルタが定義されていて、関連付けられたファイル名が存在する場合でも、キャプチャポイントをアクティブ化できます。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場合、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。

表示フィルタが指定されていない場合、パケットはライブで表示されません。コアシステムフィルタによってキャプチャされたすべてのパケットが表示されます。デフォルトの表示モードは **brief** です。



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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例 : Device> enable	特権 EXEC モードを有効にします。 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]	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処理します。

■ 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

	コマンドまたはアクション	目的
	例： Device# monitor capture mycap start display display-filter "stp"	
ステップ 3	monitor capture {capture-name} stop 例： Device# monitor capture name stop	キャプチャポイントを非アクティブにします。
ステップ 4	end 例： Device(config)# end	特権 EXEC モードに戻ります。
ステップ 5	show running-config 例： Device# show running-config	入力を確認します。
ステップ 6	copy running-config startup-config 例： Device# copy running-config startup-config	(任意) コンフィギュレーションファイルに設定を保存します。

次のタスク

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

アクティブ化する際に接続ポイントが不明

```
Device# monitor capture mycap match any
Device# 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.
Device# unable to get action unable to get action unable to get action
Device# monitor capture mycap interface g1/0/1 both
Device#monitor capture mycap start
Device#
*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

```
Device# monitor capture mycap int g1/0/1 both
Device# 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.

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

*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

```
Device# monitor capture mycap start
PD start invoked while previous run is active Failed to start capture : Wireshark operation failure
Unable to activate Capture.
Device# 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)
Device# monitor capture test stop
Capture statistics collected at software (Buffer & Wireshark):
```

```
Capture duration - 157 seconds
Packets received - 0
Packets dropped - 0
Packets oversized - 0

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

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ようにしてください。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • 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> } [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。

	コマンドまたはアクション	目的
ステップ 3	end 例： Device(config)# end	特権 EXEC モードに戻ります。
ステップ 4	show running-config 例： Device# show running-config	入力を確認します。
ステップ 5	copy running-config startup-config 例： Device# copy running-config startup-config	(任意) コンフィギュレーションファイルに設定を保存します。

例：キャプチャポイントバッファの処理

キャプチャのファイルへのエクスポート

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

```
Storage configured as File for this capture
```

キャプチャポイントバッファのクリア

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

```
Capture configured with file options
```

次のタスク



(注) 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と、スイッチは「*Failed to clear capture buffer : Capture Buffer BUSY*」エラーを表示します。

組み込みパケットキャプチャの実装方法

パケットデータキャプチャの管理



(注) アクティブなキャプチャを停止した後にのみ、アクティブなキャプチャポイントをエクスポートできます。

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 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 access-list access-list-name 例： Device# monitor capture mycap access-list v4acl	アクセスリストをパケットキャプチャのコアフィルタとして指定し、モニターキャプチャを設定します。
ステップ 3	monitor capture capture-name limit duration seconds 例： Device# monitor capture mycap limit duration 1000	モニターキャプチャの制限を設定します。
ステップ 4	monitor capture capture-name interface interface-name both 例： Device# monitor capture mycap interface GigabitEthernet 0/0/1 both	接続ポイントおよびパケットフロー方向を指定して、モニターキャプチャを設定します。
ステップ 5	monitor capture capture-name buffer circular size bytes 例： 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。

	コマンドまたはアクション	目的
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。
ステップ 7	monitor capture <i>capture-name</i> 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 <i>capture-name</i> export <i>file-location/file-name</i> 例 : Device# monitor capture mycap export tftp://10.1.88.9/mycap.pcap	分析のためにキャプチャされたデータをエクスポートします。
ステップ 9	end 例 : Device# end	特権 EXEC モードに戻ります。

キャプチャされたデータのモニタリングとメンテナンス

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

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例 : Device> enable	特権 EXEC モードを有効にします。 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 <i>capture-buffer-name</i> 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。
ステップ 3	show monitor capture <i>capture-buffer-name</i> parameter 例 :	(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。

	コマンドまたはアクション	目的
	Device# show monitor capture mycap parameter	
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
  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)

```

例: .pcap ファイルからパケット ダンプ出力の表示

```

.....0. .... = LG bit: Globally unique address (factory default)

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

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

0000 00 00 00 00 09 c9 8f 77 ab cd ab cd ab cd ab cd .....W.....
0010 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0020 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0030 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0040 ab cd ab cd ab cd ab cd .....
      Data: 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

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 ab cd ab cd ab cd ab cd ab cd ab cd  .w.....
0040  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd  .....
0050  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd  .....
0060  ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd  .....
0070  ab cd  ..

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

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

```

例：表示フィルタを使用した.pcap ファイルからのパケットの表示

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

```

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

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

```

例：.pcap ファイルにキャプチャされたパケットの数を表示

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

例：.pcap ファイルから単一パケット ダンプの表示

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

例：.pcap ファイルから単一パケット ダンプの表示

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

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

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

例：.pcap ファイルにキャプチャされたパケットの統計情報を表示

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

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

例：単純なキャプチャおよび表示

次の例は、レイヤ3 インターフェイス ギガビットイーサネット 1/0/1 でトラフィックをモニタ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) この特定のケースでは、制限を設定しており、その制限に達するとキャプチャが停止するため **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。

```
# 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) または、時間が経過した後、またはパケット数に達した後、キャプチャ操作が自動的に停止するようにすることもできます。

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,
```

例：バッファのキャプチャの使用

```
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。

例：バッファのキャプチャの使用

```

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リングされていることに注意してください。

ステップ 8：他の表示モードでパケットを表示します。

```

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

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

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

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

```



```

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

```

```

0000 00 00 00 00 0b 15 30 63 ab cd ab cd ab cd ab cd .....0c.....
0010 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0020 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0030 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
0040 ab cd ab cd ab cd ab cd ab cd ab cd ab cd ab cd .....
      Data: 0000000000b153063abcdabcdabcdabcdabcdabcdabcd...
      [Length: 72]

```

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

Device# **show monitor capture mycap buffer dump**

Starting the packet display Press Ctrl + Shift + 6 to exit

```

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

```

```

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

```

```
0070 ab cd
```

ステップ9：次のように入力してバッファをクリアします。

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



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



(注) バッファにそ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。show from buffer を実行する前にキャプチャを停止します。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 ファイル内のパケットも表示できます。

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

```
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<
```

例：出力方向のパケットの簡単なキャプチャおよび保存

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の実装では、コマンドを実行するとエクスポートが「開始」されますが、プロンプトが表示されたときには完了していません。そこで、ファイルでパケットの表示を実行する前に、Wireshark からコンソールにメッセージが表示されるのを待機する必要があります。

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

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

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

ステップ 16：次のように入力して、キャプチャ ポイントを削除します。

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

例：出力方向のパケットの簡単なキャプチャおよび保存

次の例は、フィルタにパケットをキャプチャする方法を示しています。

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

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

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

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

```
Device# show monitor capture mycap
```

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

ステップ 3：次のように入力してパケットを開始します。

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

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



(注) 時間が経過するか、パケットカウントに達した後、キャプチャ操作が自動的に停止することを許可します。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 stop
Device# monitor capture mycap export tftp://10.1.88.9/mycap.pcap
Device# end

```

例: キャプチャされたデータのモニタリングとメンテナンス

次の例は、ASCII形式でパケットをダンプする方法を示しています。

```

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

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

```

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

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

次の例は、キャプチャポイントをデバッグする方法を示しています。

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

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

Device# monitor capture mycap 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 provision 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: cleaning 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 16.5.1a	Packet Capture	パケットキャプチャ機能は、ネットワーク管理者がデバイスに出入りするパケットをキャプチャできるオンボードパケットキャプチャ機能です。
Cisco IOS XE Amsterdam 17.2.1	ダウン状態または管理状態のいずれかのインターフェイス上の組み込みパケットキャプチャ (EPC)。	この機能を使用すると、ダウン状態または管理ダウン状態のいずれかのインターフェイスで Embedded Packet Capture (EPC) を設定できます。これは、インターフェイスがアップ状態に変更された後のパケットキャプチャには影響しません。
Cisco IOS XE Amsterdam 17.3.1	組み込みパケットキャプチャ (EPC) のパケットフィルタの機能拡張：パケット長と Ether タイプ	組み込みパケットキャプチャ (EPC) では、パケット長または Ether タイプに基づいてパケットをキャプチャできます。

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 にアクセスするには、<https://cfng.cisco.com/> に進みます。



第 9 章

Flexible NetFlow の設定

- [Flexible NetFlow の前提条件](#) (207 ページ)
- [Flexible Netflow に関する制約事項](#) (208 ページ)
- [Flexible NetFlow に関する情報](#) (210 ページ)
- [Flexible NetFlow の設定方法](#) (231 ページ)
- [Flexible NetFlow の監視](#) (244 ページ)
- [Flexible NetFlow の設定例](#) (245 ページ)
- [Flexible NetFlow の機能履歴](#) (248 ページ)

Flexible NetFlow の前提条件

- 次のコマンドで定義される Flexible NetFlow の key フィールドについてよく理解していること。
 - **match flow**
 - **match interface**
 - **match {ipv4 | ipv6}**
 - **match routing**
 - **match transport**
- 次のコマンドで定義される Flexible NetFlow の nonkey フィールドについてよく理解していること。
 - **collect counter**
 - **collect flow**
 - **collect interface**
 - **collect {ipv4 | ipv6}**
 - **collect routing**
 - **collect timestamp sys-uptime**

- **collect transport**

- ネットワーキング デバイスで、Flexible NetFlow がサポートされた Cisco リリースが稼働していること。

IPv4 トラフィック

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

IPv6 トラフィック

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

Flexible Netflow に関する制約事項

次に、Flexible NetFlow に関する制約事項を示します。

- Flexible NetFlow は、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。
- 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。
- レイヤ 2、VLAN、レイヤ 3 および SVI インターフェイスがサポートされています。ただし、デバイスはトンネルをサポートしていません。
- 次のサイズの NetFlow テーブルがサポートされています。

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

- スイッチのタイプに応じて、スイッチには 1 個または 2 個の転送 ASIC があります。上の表に示されている容量は、コア単位または ASIC 単位です。
- スイッチは、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。
- 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。
- 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。
- 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。
- フローエクスポートは、テンプレートデータのタイムアウト期間が終了した後のみ、フローデータをエクスポートします。VPN ID の変更や VRF の削除などの設定変更は、タイムアウト期間の終了後に有効になります。
- 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

ここでは、Flexible Netflow について説明します。

Flexible NetFlow の概要

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

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

デバイスは、ネットワークの変則性とセキュリティの高度な検出を可能にする Flexible NetFlow 機能をサポートします。フレキシブル NetFlow 機能を使用すると、大量の定義済みフィールドの集合からキーを選択することで、そ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。

Cisco IOS XE 16.12.1 リリース以降、Flexible NetFlow 上の送信元グループタグ (SGT) および宛先グループタグ (DGT) フィールドは、IPv6 トラフィックでサポートされます。

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ティング機能を置換するために利用できる包括的な IP アカウンティング機能。

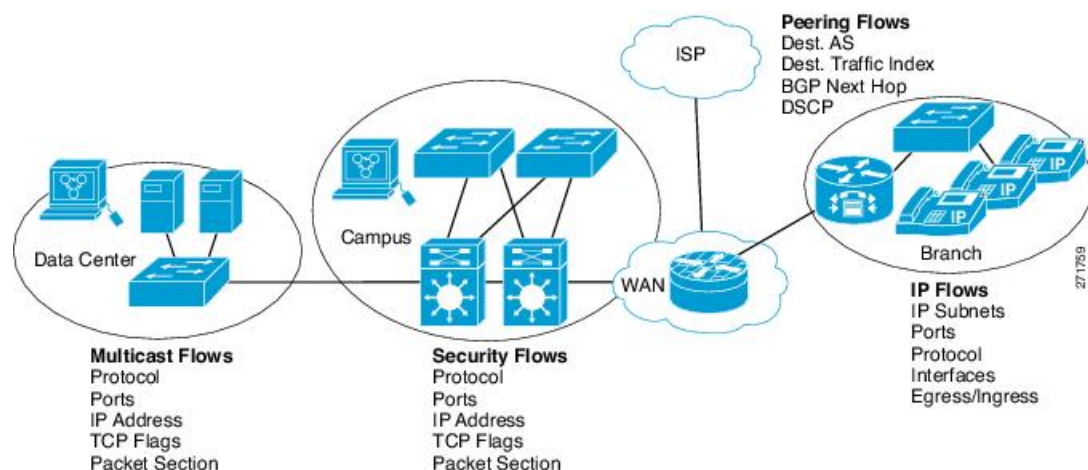
Flexible NetFlow では、ネットワークの動作を、ネットワーク内で使用されるさまざまなサービスに合わせた特定のフロー情報とともに、より効率的に理解できます。次に、Flexible NetFlow 機能用の適用例を示します。

- 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、フローエクスポーター、およびキャッシュタイプの固有の組み合わせを設定できます。フローエクスポーターの宛先 IP アドレスなどのパラメータを変更する場合、フローエクスポーター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。

ユーザー定義レコードは、QoS および帯域幅監視、アプリケーションとユーザーのトラフィックプロファイリング、DDoS 攻撃に対するセキュリティ監視などのアプリケーション用に作成できます。また、Flexible NetFlow には以前の NetFlow をエミュレートするいくつかの事前定

義済みレコードも含まれています。Flexible NetFlow のユーザー定義レコードでは、ユーザーが設定可能なサイズのパケットの連続するセクションを監視する機能を利用でき、key フィールドまたは nonkey フィールドとしてパケットのその他のフィールドや属性とともにフローレコード内で使用します。セクションにはパケットのレイヤ 3 データが含まれる場合があります。パケットのセクションフィールドでは、ユーザーが Flexible NetFlow の事前定義済みレコードの対象外のパケットフィールドを監視できます。パケットフィールドの分析機能によって、さらに詳細なトラフィック監視が可能になるため、dDoS 攻撃の調査に役立ち、URL 監視など他のセキュリティアプリケーションの実装が可能になります。

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

- **collect ipv4 section header size bytes** : 各パケットの IPv4 ヘッダーの先頭から *bytes* 引数で指定されたバイト数のキャプチャを開始します。
- **collect ipv4 section payload size bytes** : 各パケットの IPv4 ヘッダーの直後からバイトのキャプチャを開始します。キャプチャされるバイト数は *bytes* 引数で指定されます。
- **collect ipv6 section header size bytes** : 各パケットの IPv6 ヘッダーの先頭から *bytes* 引数で指定されたバイト数のキャプチャを開始します。

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

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

Flexible NetFlow の match パラメータ

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

表 10: match パラメータ

コマンド	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。
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。

コマンド	目的
<code>match ipv6 {destination hop-limit protocol source traffic-class version }</code>	<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。
<code>match transport {destination-port igmp icmp source-port}</code>	<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。
<code>match flow cts {source destination} group-tag</code>	<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 11: 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)。
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。
collect flow sampler	フローサンプラー ID をレコードの非キーフィールドとして設定します。このフィールドには、フローのモニターに使用されるフローサンプラーの ID が含まれます。

コマンド	目的
collect ipv4 destination	IPv4 宛先をフローレコードの非キーフィールドとして設定します。
collect ipv4 source	IPv4 送信元をフローレコードの非キーフィールドとして設定します
collect routing next-hop address	ネクストホップアドレス値を非キーフィールドとして設定し、フローからネクストホップ情報を収集します

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。

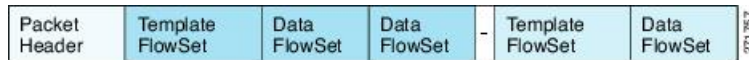
NetFlow データ エクスポート フォーマットのバージョン 9

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。

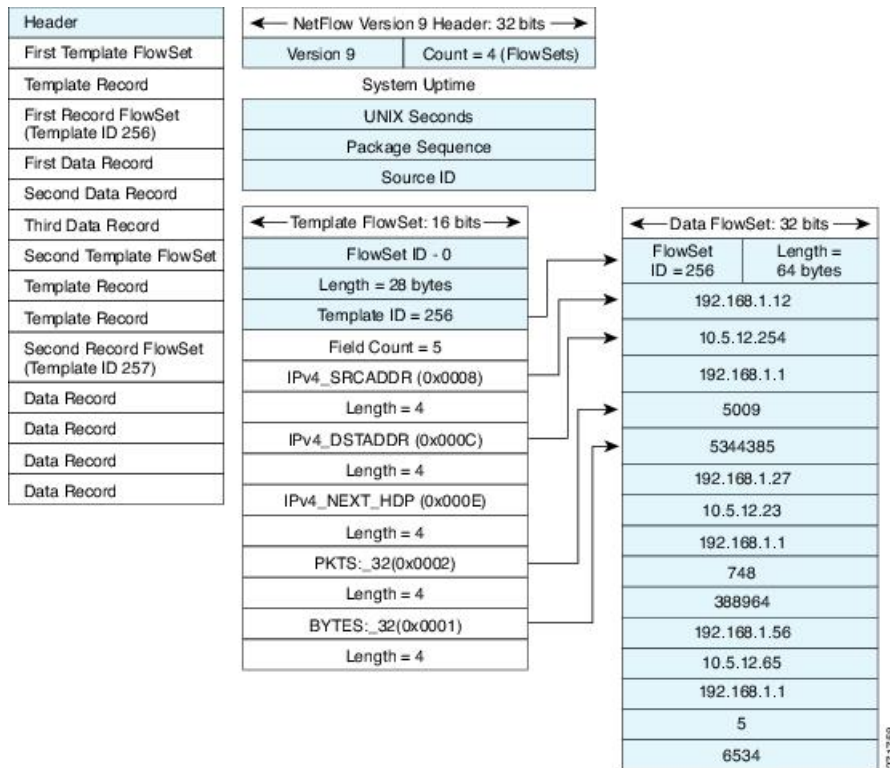
バージョン 9 のエクスポート フォーマットは、パケット ヘッダーとそれに続く 1つ以上のテンプレート フローセットまたはデータ フローセットで構成されています。テンプレート フローセットでは、将来のデータフローセットに表示されるフィールドの説明が提供されます。このようなデータ フローセットは、後で同じエクスポート パケットまたは後続のエクスポート パケットで発生する可能性があります。テンプレート フローセットおよびデータ フローセットは、次の図に示すように、単一のエクスポート パケットに混在させることができます。

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



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

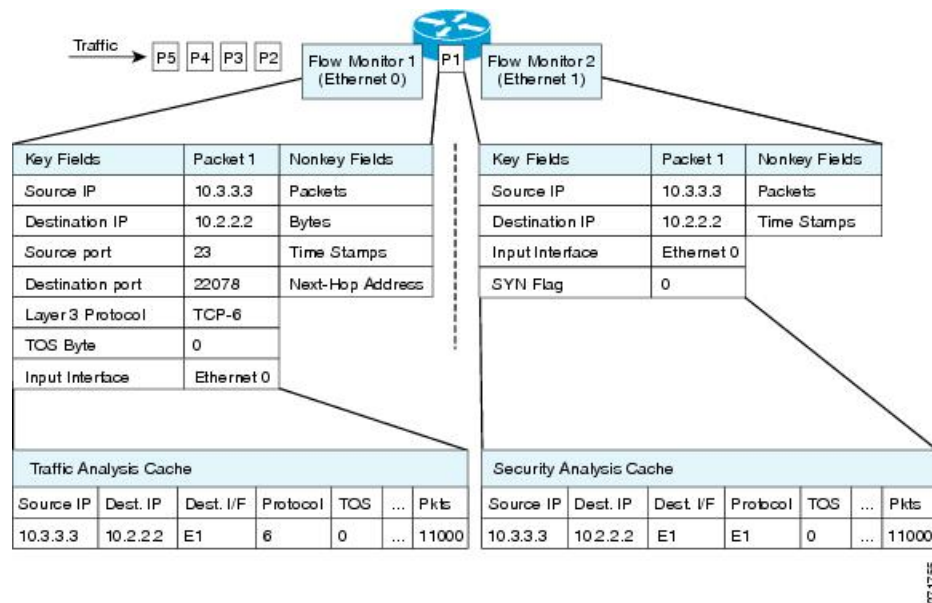
フロー モニター

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。

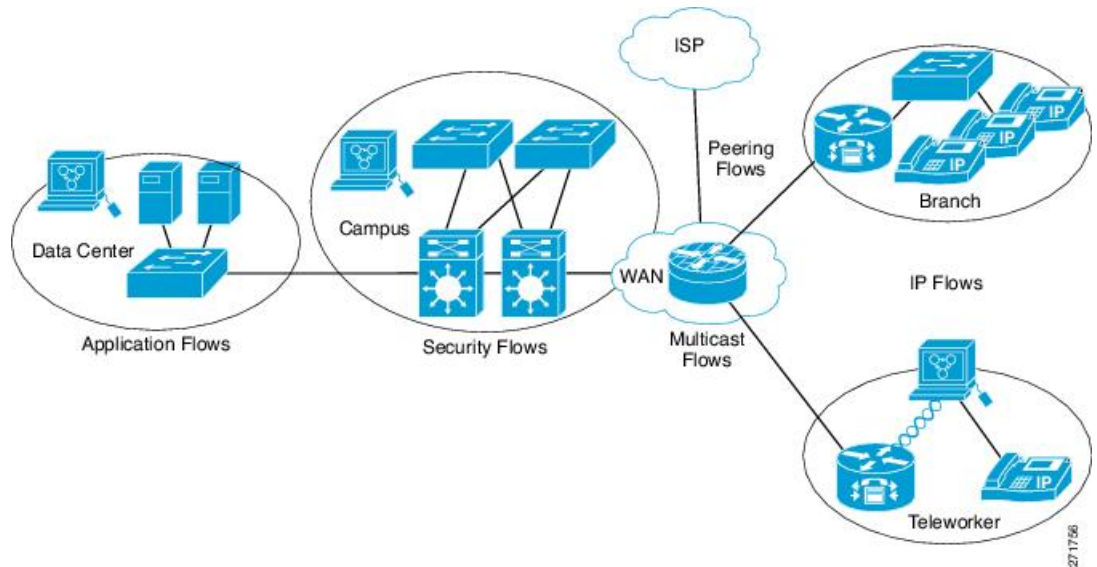
Flexible NetFlow は、同じトラフィックのさまざまなタイプの分析を実行するために使用できます。下の図では、入力インターフェイス上の標準トラフィック分析のために設計されたレコードと、出力インターフェイス上のセキュリティ分析のために設計されたレコードを使用してパケット 1 が分析されます。

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の数を制限することで、Flexible NetFlow を実行しているデバイス上の負荷を減らすために使用されます。

フロー サンプリングでは、ルータのパフォーマンスに対するモニタリング精度が交換されます。サンプラーをフロー モニターに適用すると、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							
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。

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。

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
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	
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	

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
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 TTL	—	—	対応	対応	対応	対応	IPv4 TTL と同じです。
IPv4 プロトコル	—	—	対応	対応	対応	対応	IPv4 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。

サポートされている Flexible NetFlow 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
IPv4 発信元アドレス	—	—	対応	対応	—	—	
IPv4 宛先アドレス	—	—	対応	対応	—	—	
ICMP IPv4 タイプ	—	—	対応	対応	—	—	
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。

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

表 12: デフォルトの Flexible NetFlow 設定

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

Flexible NetFlow : 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。

Flexible NetFlow : 出力 VRF サポートの概要

Flexible NetFlow : 出力 VRF サポート機能では、key フィールドとして VRF ID を収集するフローレコードがある出力フローモニターを適用して、ルータで発信パケットから 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 個の一意的な値を持つ 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 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的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]
```

MPLS での入出力 Flexible NetFlow の概要

- MPLS での入力 Flexible NetFlow (IP レベル) : この機能を使用すると、MPLS ラベルインポジションを経て MPLS ネットワークに入るパケットのインターネットプロトコル (IP) フロー情報をキャプチャできます。これらのパケットは、IP パケットとしてルータに到着し、MPLS パケットとして送信されます。PE ノードの CE 側に IPv4 および IPv6 トラフィックの入力フローモニターを設定することにより、この機能を有効にできます。
- MPLS での出力 Flexible NetFlow (IP レベル) : この機能を使用すると、MPLS ラベルインポジションを経て MPLS ネットワークから出るパケットのインターネットプロトコル (IP) フロー情報をキャプチャできます。これらのパケットは、MPLS パケットとしてルータに到着し、IP パケットとして送信されます。PE ノードの CE 側に IPv4 および IPv6 トラフィックの出力フローモニターを設定することにより、この機能を有効にできます。

Flexible NetFlow の VPN ID の設定

同じプライベートネットワークからの複数の VPN は、データトラフィックで同じプライベート送信元および宛先 IP を使用できます。これにより、データが属する IP アドレスを特定することが困難になる可能性があります。VPN-ID を使用してこの問題を解決できます。VPN-ID は、グローバルに一意の仮想プライベートネットワーク識別子です。自律システム (AS) 全体で VPN を識別するために使用されます。VPN-ID が NetFlow エクスポートパケットでエクスポートされる場合、別の AS のコレクタは、データが属する VPN に基づいてフローを関連付けて分離できます。VPN-ID は VRF-ID と同様のシステムレベルプロパティであり、同様の方法でエクスポートできます。

VPN ID の構成要素

各 VPN ID は次の要素で構成されています。

- 3 オクテットの 16 進数である組織固有識別子 (OUI)。IEEE 登録局は、ISO/IEC 8802 規格の下でコンポーネントを製造するあらゆる企業に OUI を割り当てます。OUI は、ローカルエリアネットワークアプリケーションとメトロポリタンエリアネットワークアプリケーションで使用するための、汎用的な LAN MAC アドレスとプロトコル ID を生成するために使用されます。たとえば、Cisco Systems の OUI は 00-03-6B (16 進数) です。
- 企業内での VPN を示す 4 オクテットの 16 進数である VPN インデックス。

VRF 定義コンフィギュレーションモードで **vpn id** コマンドを使用して、VPN ID を設定できます。VPN ID を次の形式で指定します。

```
vpn id oui:vpn-index
```

VPN ID を設定したら、flow-exporter コンフィギュレーションモードで **option vrf-attributes** コマンドを使用して VPN ID を設定できます。

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

	コマンドまたはアクション	目的
ステップ 1	enable 例： <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	configure terminal 例： <pre>Device# 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。
ステップ 3	flow record record-name 例： <pre>Device(config)# flow record FLOW-RECORD-1</pre>	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。

	コマンドまたはアクション	目的
		<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。
ステップ 5	match {ip ipv6} {destination source} address 例 : Device(config-flow-record)# match ipv4 destination address	(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。
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 : Device(config-flow-record)# match flow cts source group-tag Device(config-flow-record)# match flow cts destination group-tag	(注) この例では、CTSの送信元グループタグと宛先グループタグをレコードのキーフィールドとして設定します。 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。

	コマンドまたはアクション	目的
		<p>(注)</p> <ul style="list-style-type: none"> • 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 の強制を無効化されている場合、DGT は 0 になります。 • 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。
ステップ 8	<p>end</p> <p>例 :</p> <pre>Device(config-flow-record)# end</pre>	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。

	コマンドまたはアクション	目的
ステップ 9	show flow record <i>record-name</i> 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 10	show running-config flow record <i>record-name</i> 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

	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	flow exporter <i>name</i> 例： Device(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 4	description <i>string</i> 例：	(任意) 最大63文字で、このフローの説明を指定します。

	コマンドまたはアクション	目的
	Device (config-flow-exporter) # description ExportV9	
ステップ 5	destination { <i>ipv4-address</i> } 例 : Device (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 6	dscp <i>value</i> 例 : Device (config-flow-exporter) # dscp 0	(任意) DSCP (DiffServ コードポイン ト) 値を指定します。範囲は 0～63 で す。デフォルトは 0 です。
ステップ 7	source <i>interface type interface number</i> 例 : Device (config-flow-exporter) # source gigabitEthernet1/0/1	(任意) 設定された宛先で NetFlow コ ネクタに到達するために使用するイン ターフェイスを指定します。 (注) フローエクスポートは、送 信元インターフェイスとし てアンナンバード IP イン ターフェイスをサポートし ていません。 送信元として次のインターフェイスを 設定できます。 <ul style="list-style-type: none"> • Auto Template : 自動テンプレート インターフェイス • Capwap : Capwap トンネル イン ターフェイス • GigabitEthernet : Gigabit Ethernet IEEE 802 • GroupVI : グループ仮想インター フェイス • Internal Interface : 内部インター フェイス • Loopback : ループバック インター フェイス • Null : スル インターフェイス

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • Port-channel : インターフェイスのイーサネット チャンネル • TenGigabitEthernet : 10 ギガビットイーサネット • Tunnel : トンネルインターフェイス • Vlan : Catalyst VLAN
ステップ 8	transport udp number 例 : Device(config-flow-exporter) # transport udp 200	(任意) NetFlow コレクタに到達するために使用する UDP ポートを指定します。
ステップ 9	ttl seconds 例 : Device(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 10	export-protocol {netflow-v9} 例 : Device(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 11	end 例 : Device(config-flow-record) # end	特権 EXEC モードに戻ります。
ステップ 12	show flow exporter [name record-name] 例 : Device# 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 13	copy running-config startup-config 例 : Device# copy running-config startup-config	(任意) コンフィギュレーションファイルに設定を保存します。

次のタスク

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。

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

	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	flow monitor monitor-name 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。 • 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。

	コマンドまたはアクション	目的
ステップ 4	description <i>description</i> 例 : <pre>Device(config-flow-monitor)# description Used for basic ipv4 traffic analysis</pre>	(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 { ipv4 ipv6 } <i>record</i> [peer]} 例 : <pre>Device(config-flow-monitor)# record FLOW-RECORD-1</pre>	フロー モニターのレコードを指定します。
ステップ 6	cache { timeout { active inactive update rate-limit } <i>seconds</i> type normal } 例 : <pre>Device(config-flow-monitor)# cache type normal Device(config-flow-monitor)# cache timeout active</pre>	(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 protocol 例 : <pre>Device(config-flow-monitor)# statistics packet protocol</pre>	(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。
ステップ 9	statistics packet size 例 : <pre>Device(config-flow-monitor)# statistics packet size</pre>	(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 <i>exporter-name</i> 例 : <pre>Device(config-flow-monitor)# exporter EXPORTER-1</pre>	(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。

	コマンドまたはアクション	目的
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] <i>monitor-name</i> [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 <i>monitor-name</i> 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

	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	sampler <i>sampler-name</i> 例 : Device(config)# sampler SAMPLER-1	サンプラーを作成し、サンプラーコンフィギュレーションモードを開始します。 • このコマンドでは、既存のサンプラーを変更することもできます。
ステップ 4	description <i>description</i> 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。 • <i>window-size</i> 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 <i>type number</i> 例 : Device(config)# interface GigabitEthernet 0/0/0	インターフェイスを指定し、インターフェイスコンフィギュレーションモードを開始します。
ステップ 8	{ip ipv6} flow monitor monitor-name [[sampler] sampler-name] {input output} 例 : Device(config-if)# ip flow monitor FLOW-MONITOR-1 sampler SAMPLER-1 input	作成したフローモニタ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、サンプリングをイネーブルにします。
ステップ 9	end 例 : Device(config-if)# end	インターフェイス コンフィギュレーションモードを終了し、特権 EXEC モードに戻ります。
ステップ 10	show sampler sampler-name 例 : Device# show sampler SAMPLER-1	設定し有効化したフローサンプラーのステータスおよび統計情報を表示します。

インターフェイスへのフローの適用

フロー モニタ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

	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(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	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。

	コマンドまたはアクション	目的
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

	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	vlan [configuration] <i>vlan-id</i> 例： Device(config)# vlan configuration 30 Device(config-vlan-config)#	VLAN または VLAN コンフィギュレーションモードを開始します。
ステップ 4	ip flow monitor <i>monitor name</i> [sampler <i>sampler name</i>] { input } 例：	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。

	コマンドまたはアクション	目的
	Device(config-vlan-config)# ip flow monitor MonitorTest input	
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。

レイヤ 2 NetFlow の設定

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

手順

	コマンドまたはアクション	目的
ステップ 1	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(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	flow record name 例 : Device(config)# flow record L2_record Device(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 4	match datalink {dot1q ethertype mac vlan} 例 : Device(config-flow-record)# match datalink ethertype	レイヤ 2 属性をキーとして指定します。
ステップ 5	end 例 :	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	Device (config-flow-record) # end	
ステップ 6	show flow record [name] 例 : Device# show flow record	(任意) インターフェイスの NetFlow 情報を表示します。
ステップ 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。

Flexible NetFlow の監視

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

表 13: Flexible NetFlow のモニタリング コマンド

コマンド	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 exporter-name]	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 exporter-name]	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 record-name]	NetFlow のフロー レコード情報を表示します。
show sampler [broker name name]	NetFlow サンプラーに関する情報を表示します。

Flexible NetFlow の設定例

例：フローの設定

フローを作成し、そのフローをインターフェイスに適用する例を示します。

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

Device(config)# flow export export1
Device(config-flow-exporter)# destination 10.0.101.254
Device(config-flow-exporter)# transport udp 2055
Device(config-flow-exporter)# exit
Device(config)# flow record record1
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match ipv4 protocol
Device(config-flow-record)# match transport source-port
Device(config-flow-record)# match transport destination-port
Device(config-flow-record)# match flow cts source group-tag
Device(config-flow-record)# match flow cts destination group-tag
Device(config-flow-record)# collect counter byte long
Device(config-flow-record)# collect counter packet long
Device(config-flow-record)# collect timestamp absolute first
Device(config-flow-record)# collect timestamp absolute last
Device(config-flow-record)# exit
Device(config)# flow monitor monitor1
Device(config-flow-monitor)# record record1
Device(config-flow-monitor)# exporter export1
Device(config-flow-monitor)# exit
Device(config)# interface tenGigabitEthernet 1/0/1
Device(config-if)# ip flow monitor monitor1 input
Device(config-if)# end
```

例：IPv4 入力トラフィックのモニタリング

次の例は、IPv4 入力トラフィックをモニタ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）。

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

例 : IPv4 出カトラフィックのモニタリング

```

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

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

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

Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

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

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

```

例 : IPv4 出カトラフィックのモニタリング

```

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

Device(config)# flow exporter fe-1

```

```
Device(config-flow-exporter)# destination 10.5.120.16
Device(config-flow-exporter)# source Vlan105
Device(config-flow-exporter)# dscp 32
Device(config-flow-exporter)# ttl 200
Device(config-flow-exporter)# transport udp 2055
Device(config-flow-exporter)# template data timeout 240
Device(config-flow-exporter)# exit

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

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

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

Device# show flow monitor fm-1-output cache format table
```

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
```

例：出力 VRF サポート用の Flexible NetFlow の設定

```
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
```

例：出力 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出力フローモニターを適用することで、デバイスの発信パケットからの 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 output
Device(config-if)# end
```

Flexible NetFlow 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 16.5.1a	Flexible NetFlow	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。

リリース	機能	機能情報
Cisco IOS XE Gibraltar 16.12.1	IPv6 トラフィックの Flexible NetFlow の送信元グループタグ (SGT) と宛先グループタグ (DGT)	IPv6 トラフィックの Flexible NetFlow の送信元グループタグ (SGT) と宛先グループタグ (DGT) フィールドのサポートが導入されました。
Cisco IOS XE Amsterdam 17.1.1	MPLS での入力および出力の柔軟な Netflow	マルチプロトコル ラベル スイッチング (MPLS) ネットワークに入るときに、MPLS ラベルインポジションの状態にあるパケットを対象とした IP フロー情報のキャプチャを許可します。これらのパケットは、IP パケットとしてデバイスに到着し、MPLS パケットとして送信されます。
Cisco IOS XE Amsterdam 17.2.1	NetFlow の VPN ID	Flexible NetFlow の仮想プライベートネットワーク識別子 (VPN-ID) の設定をサポートします。VPN-ID はグローバルで一意です。自律システム (AS) 全体で VPN を識別するために使用されます。
Cisco IOS XE Bengaluru 17.5.1	出力 VRF サポート用の Flexible NetFlow	出力 VRF サポート用の Flexible NetFlow を設定するためのサポートが導入されました。

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 にアクセスするには、<https://cfmng.cisco.com/> に進みます。



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 \(251 ページ\)](#)
- [暗号化トラフィック分析について \(251 ページ\)](#)
- [暗号化トラフィック分析の設定方法 \(252 ページ\)](#)
- [暗号化トラフィック分析の設定例 \(255 ページ\)](#)
- [その他の参考資料 \(256 ページ\)](#)
- [暗号化トラフィック分析の機能履歴 \(256 ページ\)](#)

暗号化トラフィック分析の制約事項

- SD-Access 導入環境の場合、ETA は、アクセスポートとワイヤレス VLAN でサポートされます。
- ETA は、管理、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ません。
- 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 データとともに、対応するフローの FNF データもエクスポートされます。

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ケット (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。

暗号化トラフィック分析のエクスポート

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

グローバルコレクタの宛先 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

	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 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。

Active-Timeout 値の設定

次の手順に従って、active-timeout 値を設定します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# 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	active-timeout <i>time in seconds</i> 例： Device(config-et-analytics)# active-timeout 300	active-timeout 値を設定します。範囲は 1 ～ 604800 で、デフォルト値は 30 分です。

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

	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

	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	interface <i>interface-id</i> 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 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：Active-Timeout 値の設定

この例は、300 秒の active-timeout value 値を設定する方法を示しています。

```
Device#config terminal
Device (config)#et-analytics
Device (config-et-analytics)#active-timeout 300
```

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 gil/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

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 Fuji 16.8.1a	暗号化トラフィック分析	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。
Cisco IOS XE Amsterdam 17.3.1	アプリケーションの可視性およびコントロールと暗号化トラフィック分析の相互運用性	同じポートでのアプリケーションの表示およびコントロールと暗号化トラフィック分析の相互運用性のサポートが導入されました。
Cisco IOS XE Bengaluru 17.6.1	active-timeout 値の設定のサポート	active-timeout 値を設定するサポートが導入されました。Cisco IOS XE Bengaluru 17.6.1 以前では、active-timeout 値はデフォルトで1800秒に設定されていました。

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 にアクセスするには、<https://cfmng.cisco.com/> に進みます。



第 11 章

ネットワークの検出と対応の設定

- ネットワークの検出と対応の前提条件 (259 ページ)
- ネットワークの検出と対応の制約事項 (259 ページ)
- ネットワークの検出と対応について (260 ページ)
- ネットワークの検出と対応の設定方法 (260 ページ)
- ネットワークの検出と対応の設定の確認 (269 ページ)
- ネットワークの検出と対応の設定例 (270 ページ)
- ネットワークの検出と対応の機能履歴 (273 ページ)

ネットワークの検出と対応の前提条件

- ネットワーク内のすべてのデバイスは、Stealthwatch Cloud ポータルに到達可能である必要があります。また、すべての暗号化トラフィックは、HTTPS (TCP ポート 443) を使用して Stealthwatch Cloud ポータルに到達する必要があります。
- Stealthwatch Cloud ポータルによって提供される証明書を検証するには、該当するルート認証局 (CA) をデバイスで設定する必要があります。
- データの損失を避けるため、十分な帯域幅があることを確認してください。

ネットワークの検出と対応の制約事項

- Cisco Encrypted Traffic Analytics は、Stealthwatch Cloud ポータルではサポートされていません。
- HTTP プロキシはサポートされていません。
- Stealthwatch Cloud ポータルは、プライマリ DNS サーバーのみを使用します。プライマリ DNS サーバーに障害が発生すると、エラーが表示されます。
- デバイスで設定されている DNS サーバーが Stealthwatch Cloud モニターの URL を解決できない場合、Stealthwatch Cloud センサーが Stealthwatch Cloud ポータルに登録されていても、ファイルのアップロードは失敗します。

ネットワークの検出と対応について

Cisco Secure Network Analytics (Stealthwatch Cloud と呼ばれる) は、エンタープライズテレメトリを使用して脅威を検出し、ネットワークセグメンテーションとともに脅威への迅速な対応を提供するネットワーク検出および対応ソリューションです。Cisco Secure Network Analytics を使用すると、ネットワーク管理者は、ネットワークにログインしているすべてのユーザーを追跡し、そのアクティビティを監視することもできます。

Cisco Catalyst スイッチのネットワーク検出および対応ソリューションの一部として、分析に使用されるエンタープライズテレメトリは、Flexible NetFlow フローです。

デバイスで Stealthwatch Cloud プロパティを設定する必要があります。次に、Stealthwatch Cloud ポータルのフローレコードとフローエクスポートを作成する必要があります。



(注) フローレコードでは、必須の5タプルフィールド（プロトコル、送信元アドレス、送信元ポート、宛先アドレス、および宛先ポート）が、フローの開始、フローの終了、パケット数、Stealthwatch Cloud ポータルにアップロードされるレコードのバイト数とともに設定されていることが必要です。

フローモニターに対してフローレコードおよびフローエクスポートを設定します。設定後にフローモニターから生成されたすべてのフローは、カスタム形式に変換され、Stealthwatch Cloud ポータルにアップロードされます。

ネットワークの検出と対応の設定方法

次のセクションでは、ネットワークの検出と対応の設定に関する情報を示します。

登録に向けた証明書の設定

登録に向けて証明書を設定するには、次の手順を実行します。

手順

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

	コマンドまたはアクション	目的
ステップ 3	crypto pki trustpoint <i>name</i> 例： Device(config)# crypto pki trustpoint stealthwatch1	トラストポイントおよび設定された名前を宣言して、CA トラストポイン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	revocation-check <i>method1</i> [<i>method2</i> <i>method3</i>] 例： Device(ca-trustpoint)# revocation-check none	証明書の失効ステータスをチェックします。 none : 証明書のチェックは無視されません。
ステップ 5	enrollment <i>mode</i> 例： Device(ca-trustpoint)# enrollment terminal	証明書の登録モードとして端末を指定します。
ステップ 6	exit 例： Device(ca-trustpoint)# exit	CA トラストポイン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に戻ります。
ステップ 7	crypto pki authenticate <i>name</i> 例： Device(config)# crypto pki authenticate stealthwatch1 Enter the base 64 encoded CA certificate. End with a blank line or the word "quit" on a line by itself	トラストポイント名を認証して、CA トラストポイントコンフィギュレーションモードを開始します。 プロンプトが表示されたら、 https://www.amazontrust.com/repository/SFC2CA-SFSRootCAG2.pem から Starfield Services ルート証明書をコピーして貼り付けます。 システムは、次のステートメントとともにプロンプトを表示します。 % Do you accept this certificate? [yes/no]: 確認するには、 yes と入力します。
ステップ 8	exit 例： Device(config)# exit	特権 EXEC モードに戻ります。
ステップ 9	show pki trustpoints <i>name</i> 例： Device# show crypto pki trustpoints stealthwatch1	(オプション) 設定したトラストポイントに関する情報を表示します。

ファイルアップロードに向けた証明書の設定

ファイルアップロードに向けて証明書を設定するには、次の手順を実行します。

始める前に

Baltimore CyberTrust ルート証明書をダウンロードします。

1. Web ブラウザで <https://www.digicert.com/kb/digicert-root-certificates.htm> を開きます。
2. [Baltimore CyberTrust Root] で、[Download PEM] をクリックします。
3. 場所を選択して、BaltimoreCyberTrustRoot.crt.pem ファイルを保存します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーションモードを開始します。
ステップ 3	crypto pki trustpoint name 例： Device(config)# crypto pki trustpoint stealthwatch2	トラストポイントおよび設定された名前を宣言して、CA トラストポイン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	revocation-check method1 [method2 method3] 例： Device(ca-trustpoint)# revocation-check none	証明書の失効ステータスをチェックします。 none ：証明書のチェックは無視されます。
ステップ 5	enrollment terminal 例： Device(ca-trustpoint)# enrollment terminal	証明書の登録モードとして terminal を指定します。
ステップ 6	exit 例： Device(ca-trustpoint)# exit	CA トラストポイン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に戻ります。

	コマンドまたはアクション	目的
ステップ 7	crypto pki authenticate name 例 : <pre>Device(config)# crypto pki authenticate stealthwatch2 Enter the base 64 encoded CA certificate. End with a blank line or the word "quit" on a line by itself</pre>	トラストポイント名を認証して、CA トラストポイント コンフィギュレーション モードを開始します。 プロンプトが表示されたら、 BaltimoreCyberTrustRoot.crt.pem ファイルからテキストをコピーして貼り付けます。 システムは、次のステートメントとともにプロンプトを表示します。 <pre>% Do you accept this certificate? [yes/no]:</pre> 確認のために yes を入力します。
ステップ 8	exit 例 : <pre>Device(config)# exit</pre>	特権 EXEC モードに戻ります。
ステップ 9	show pki trustpoints name 例 : <pre>Device# show crypto pki trustpoints stealthwatch2</pre>	(オプション) 設定したトラストポイントに関する情報を表示します。

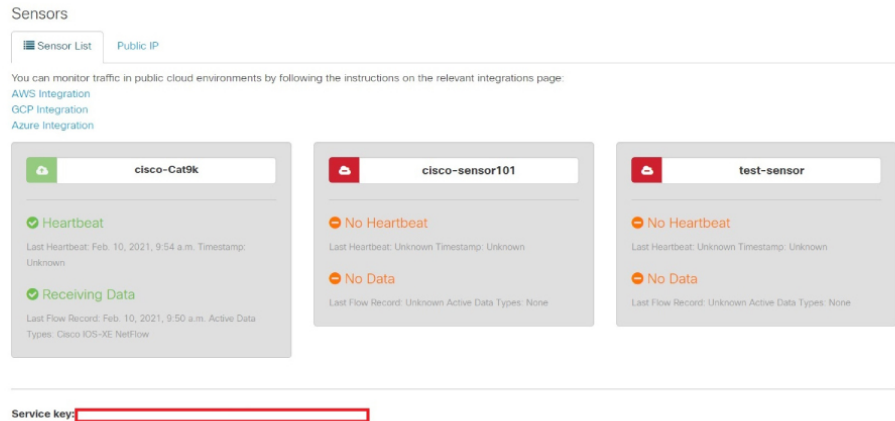
デバイスでの Stealthwatch Cloud の設定

デバイスで Stealthwatch Cloud を設定するには、次の手順を実行します。

始める前に

Stealthwatch Cloud ポータルからサービスキーを表示するには、次の手順を実行します。

1. ブラウザから Stealthwatch Cloud ポータルを開きます。
2. [Dashboard] ビューで、ウィンドウの右隅にあるクラウドアイコンをクリックし、[Sensors] を選択します。
3. ウィンドウの下部に移動して、サービスキーを見つけます。



手順

	コマンドまたはアクション	目的
ステップ 1	enable 例 : Device> enable	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	configure terminal 例 : Device# configure terminal	グローバル コンフィギュレーションモードを開始します。
ステップ 3	stealthwatch-cloud-monitor 例 : Device(config)# stealthwatch-cloud-monitor	Stealthwatch Cloud モニターを設定し、 stealthwatch-cloud-monitor 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	service-key SwC-service-key 例 : Device(config-stealthwatch-cloud-monitor)# service-key xx	Stealthwatch Cloud サービスキーを設定します。
ステップ 5	sensor-name SwC-sensor-name 例 : Device(config-stealthwatch-cloud-monitor)# sensor-name mysensor	(オプション) Stealthwatch Cloud 登録に使用するセンサー名を設定します。デフォルトでは、デバイスのシリアル番号がセンサー名として使用されます。
ステップ 6	url SwC-server-url 例 : Device(config-stealthwatch-cloud-monitor)# url https://sensors.eu-2.obsrvbl.com	(オプション) Stealthwatch Cloud サーバーの URL を設定します。 リダイレクトを回避するため、適切な Stealthwatch Cloud サーバーの URL を設

	コマンドまたはアクション	目的
		定めます。URL が設定されていない場合は、米国内の Stealthwatch Cloud サーバーの URL がデフォルトで使用されます。ロケーションに基づいて、デフォルトの URL は最も近い Stealthwatch Cloud サーバーの URL にリダイレクトされます。
ステップ 7	end 例 : Device(config-stealthwatch-cloud-monitor)# end	特権 EXEC モードに戻ります。

Flexible NetFlow を Stealthwatch Cloud ポータルと統合する方法

次のセクションでは、Flexible Netflow を Stealthwatch Cloud ポータルと統合する方法に関する設定情報を提供します。

フロー レコードの作成

フローレコードを作成するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例 : Device> enable	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	configure terminal 例 : Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	flow record record-name 例 : Device(config)# flow record SWCRec	フローレコードを作成し、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 4	description description 例 : Device(config-flow-record)# description swc flow	(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 ipv4 source address 例： Device(config-flow-record)# match ipv4 source address	レコードのキーフィールドとして IPv4 送信元アドレスを設定します。
ステップ 6	match ipv4 destination address 例： Device(config-flow-record)# match ipv4 destination address	レコードのキーフィールドとして IPv4 宛先アドレスを設定します。
ステップ 7	match transport source-port 例： Device(config-flow-record)# match transport source-port	レコードのキーフィールドとして送信元ポートを設定します。
ステップ 8	match transport destination-port 例： Device(config-flow-record)# match transport destination-port	レコードのキーフィールドとして宛先ポートを設定します。
ステップ 9	match ipv4 protocol 例： Device(config-flow-record)# match ipv4 protocol	レコードのキーフィールドとして IPv4 プロトコルを設定します。
ステップ 10	collect counter bytes long 例： Device(config-flow-record)# collect counter bytes long	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。
ステップ 11	collect counter packets long 例： Device(config-flow-record)# collect counter packets long	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。
ステップ 12	collect timestamp absolute first 例： Device(config-flow-record)# collect timestamp absolute first	フロー内の最初に確認されたタイムスタンプを非キーフィールドとして設定し、フローからの最初のパケットが確認された絶対時間の収集を有効にします。
ステップ 13	collect timestamp absolute last 例： Device(config-flow-record)# collect timestamp absolute last	フロー内の最初に確認されたタイムスタンプを非キーフィールドとして設定し、フローからの最新のパケットが確認された絶対時間の収集を有効にします。

	コマンドまたはアクション	目的
		認された絶対時間の収集を有効にします。
ステップ 14	end 例： Device (config-flow-record) # end	特権 EXEC モードに戻ります。

フロー エクスポートの作成

フローエクスポートを作成するには、次の手順を実行します。



(注) Stealthwatch Cloud に設定できるアクティブなフローエクスポートは 1 つだけです。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーションモードを開始します。
ステップ 3	flow exporter name 例： Device (config) # flow exporter SWCExp	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 4	destination {hostname} 例： Device (config-flow-exporter) # destination stealthwatch-cloud	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	end 例： Device (config-flow-record) # end	特権 EXEC モードに戻ります。

フロー モニタの設定

フローモニターを設定するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	flow monitor flow-monitor-name 例： Device(config)# flow monitor SWCMon	フローモニターを定義します。
ステップ 4	cache timeout active seconds 例： Device(config-flow-monitor)# cache timeout active 60	アクティブ フロー タイムアウトを秒単位で指定します。
ステップ 5	exporter flow-exporter-name 例： Device(config-flow-monitor)# exporter SWCExp	フロー情報をエクスポートにエクスポートします。
ステップ 6	record flow-exporter-name 例： Device(config-flow-monitor)# record SWCRec	基本の IPv4 テンプレートを使用してフローレコードを指定します。
ステップ 7	end 例： Device(config-flow-monitor)# end	特権 EXEC モードに戻ります。

インターフェイスへのフローの適用

インターフェイスにフローを適用するには、次の手順を実行します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。

	コマンドまたはアクション	目的
ステップ 2	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 type number 例： Device(config)# interface gigabitethernet 1/0/1	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	ip flow monitor monitor-name input 例： Device(config-if)# ip flow monitor SWCMon input	入力パケット用のインターフェイスに IPv4 フローモニターを関連付けます。
ステップ 5	ip flow monitor monitor-name output 例： Device(config-if)# ip flow monitor SWCMon output	出力パケット用のインターフェイスに IPv4 フローモニターを関連付けます。
ステップ 6	end 例： Device(config-if)# end	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

ネットワークの検出と対応の設定の確認

ネットワークの検出と対応の設定を確認するには、特権 EXEC モードで次のコマンドを使用します。

表 14: ネットワークの検出と対応の設定を確認するためのコマンド

コマンド	目的
show stealth-watch-cloud detail	Stealthwatch Cloud 登録ステータスとその設定値を表示します。
show platform software fed switch switch-number swc statistics	Stealthwatch Cloud 統合の統計情報を表示します。
clear platform software fed switch switch-number swc statistics	Stealthwatch Cloud 統合の統計情報をクリアします。
show platform software fed switch switch-number swc connection	Stealthwatch Cloud 統合の接続の詳細とイベントを表示します。

コマンド	目的
clear platform software fed switch <i>switch-number</i> swc connection	Stealthwatch Cloud 統合の接続の詳細とイベントをクリアします。

ネットワークの検出と対応の設定例

次のセクションでは、ネットワークの検出と対応の設定例を示します。

例：デバイスでの Stealthwatch Cloud の設定と統合

次に、デバイスで Stealthwatch Cloud を設定し統合する方法の例を示します。

```
Device> enable
Device# configure terminal
Device(config)# stealthwatch-cloud-monitor
Device(stealthwatch-cloud-monitor)# service-key XXXXXXXXXXXXXXXXXXXXXXXX
Device(stealthwatch-cloud-monitor)# sensor-name mysensor
Device(stealthwatch-cloud-monitor)# url https://sensors.eu-2.obsrvbl.com
Device(stealthwatch-cloud-monitor)# exit
Device(config)# flow record SWCRec
Device(config-flow-record)# description for stealthwatch cloud
Device(config-flow-record)# match ipv4 source address
Device(config-flow-record)# match ipv4 destination address
Device(config-flow-record)# match ipv4 protocol
Device(config-flow-record)# match transport source-port
Device(config-flow-record)# match transport destination-port
Device(config-flow-record)# match flow cts source group-tag
Device(config-flow-record)# match flow cts destination group-tag
Device(config-flow-record)# collect counter byte long
Device(config-flow-record)# collect counter packet long
Device(config-flow-record)# collect timestamp absolute first
Device(config-flow-record)# collect timestamp absolute last
Device(config-flow-record)# exit
Device(config)# flow exporter SWCExp
Device(config-flow-exporter)# destination stealthwatch-cloud
Device(config-flow-exporter)# exit
Device(config)# flow monitor SWCMon
Device(config-flow-monitor)# cache timeout active 60
Device(config-flow-monitor)# exporter SWCExp
Device(config-flow-monitor)# record SWCRec
Device(config-flow-monitor)# exit
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# ip flow monitor SWCMon input
Device(config-if)# ip flow monitor SWCMon output
Device(config-if)# end
```

例：Stealthwatch Cloud の設定の確認

次に、**show stealthwatch-cloud detail** コマンドの出力例を示します。

```

Device> enable
Device# show stealthwatch-cloud detail
=====
Stealthwatch Cloud Parameters
=====
Service Key   : XXXXXXXXXXXXXXXXXXXXXXXXXX
Sensor Name   : C9200
URL           : https://sensor.eu-prod.obsrvbl.com
=====
Stealthwatch Cloud Sensor Info
=====
Sensor Status : Registered
Last heartbeat : 2020-08-21T10:35:16

```

次に、**show platform fed switch active swc statistics** コマンドの出力例を示します。

```

Device> enable
Device# show platform software fed switch active swc statistics
=====
SWC Upload Statistics:
=====
1: Last file uploaded   : 202102100928_1
2: Time of upload      : 02/10/21 09:29:41 UTC
3: Current file uploading :
4: Files queued for upload :
5: Number of files queued : 0
6: Last failed upload   :
7: Files failed to upload : 0
8: Files successfully uploaded : 1
=====
SWC File Creation Statistics:
=====
9: Last file created    : 202102100929_1
10: Time of creation   : 02/10/21 09:29:08 UTC
=====
SWC Flow Statistics:
=====
11: Number of flows in prev file: 15
12: Number of flows in curr file: 11
13: Invalid dropped flows : 0
14: Error dropped flows  : 0
=====
SWC Flags:
=====
15: Is Registered   : Registered
16: Delete debug    : Disabled
17: Exporter delete debug : Disabled
18: Certificate Validation : Enabled

```

次に、**show platform software fed switch active swc connection** コマンドの出力例を示します。

```

Device> enable
Device# show platform software fed switch active swc connection
Stealthwatch-Cloud details
Registration
#ID       : 0xc000001
URL       : https://sensor.ext.obsrvbl.com
Service Key : XXXXXXXXXXXXXXXXXXXXXXXXXX
Sensor Name : C9200

```

例 : Stealthwatch Cloud の設定の確認

```

Registered : N/A
Connection
Status      : DOWN
<<- Status will be in UP state only when the flow uploads into the Stealthwatch Cloud.
Last status update : 02/09/2021 10:10:47
# Flaps        : 0
# Heartbeats   : 0
# Lost heartbeats : 0
Total RX bytes : 7360
Total TX bytes : 869
Upload Speed (B/s) : 127
Download Speed (B/s) : 58
# Open sessions : 0
# Redirections  : 0
# Timeouts     : 0

HTTP Events
GET response      : 4
GET request       : 4
GET Status Code 2XX : 4
PUT response      : 12
PUT request       : 12
PUT Status Code 2XX : 2
POST response     : 2
POST request      : 2
POST Status Code 2XX : 2

API Events
TX                : 4
OK                : 2
Error             : 2

Event History
Timestamp          #Times  Event                      RC Context
-----
02/10/2021 09:29:41.126 2      SEND_OK                    0 ID:0003
02/10/2021 09:29:39.795 2      SIGNAL_DATA                 0 ID:0003
02/10/2021 09:29:38.279 12     PUT_DATA                    0 ID:0003
02/10/2021 09:29:37.962 4      GET_URL                     0 ID:0003
02/10/2021 09:29:37.961 4      SEND_START                  0 ID:0003
02/10/2021 09:27:41.484 2      SEND_ERR                    0 ID:0001
02/10/2021 09:27:41.484 2      MAX_ATTEMPTS                0 ID:0001
02/10/2021 09:22:53.670 4      REGISTER_OK                 0 Not applicable
02/10/2021 09:22:53.670 4      SEND_ABORT_ALL              0 config change
02/10/2021 09:22:53.670 1      OPTIONS_CONFIG              0 File Extension: .csv.gz (reset)
02/10/2021 09:22:53.669 1      OPTIONS_CONFIG              0 Data Type: ios-xe-catalyst
02/10/2021 09:22:53.669 1      OPTIONS_CONFIG              0 URL: https://sensor.ext.obsrvbl.com
(res
02/10/2021 09:22:53.668 1      OPTIONS_CONFIG              0 Sensor Name: niinamdaUS (reset)
02/10/2021 09:22:53.553 1      OPTIONS_CONFIG              0 Service Key:
b5tQtXJM8AGZSp6oB8FvK4H0FiW

```

ネットワークの検出と対応の機能履歴

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 Bengaluru 17.5.1	ネットワークの検出と対応	Cisco Secure Network Analytics (Stealthwatch Cloudとも呼ばれる)は、高度な脅威検出、脅威への迅速な対応、簡素化されたネットワークセグメンテーションを提供するネットワーク検出および対応ソリューションです。

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。

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。