



Flexible NetFlow の設定

- [Flexible NetFlow の前提条件](#) (1 ページ)
- [Flexible Netflow に関する制約事項](#) (2 ページ)
- [Flexible NetFlow に関する情報](#) (4 ページ)
- [Flexible NetFlow の設定方法](#) (23 ページ)
- [Flexible NetFlow の監視](#) (37 ページ)
- [Flexible NetFlow の設定例](#) (37 ページ)
- [Flexible NetFlow の機能情報](#) (40 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。
- レイヤ 3 ポートチャネル メンバポートの 1 つに適用された Flexible NetFlow 設定は、同じデバイス内の他のすべてのメンバポートにも適用されます。スタックの一部として、または StackWise Virtual を使用して設定されたデバイスの場合、ポートチャネルインターフェイスで Flexible NetFlow を設定することを推奨します。

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 ポートチャネル メンバポートではサポートされます。
- Flexible NetFlow は、レイヤ 3 ポートチャネル インターフェイスとメンバポートでサポートされますが、同じトラフィックタイプと方向の両方に対して同時にサポートされることはありません。
- Traditional NetFlow のアカウンティングはサポートされていません。
- Flexible NetFlow バージョン 9 およびバージョン 10 のエクスポートフォーマットがサポートされています。ただし、エクスポートプロトコルが設定されていない場合は、バージョン 9 のエクスポートフォーマットがデフォルトで適用されます。
- 有線 Application Visibility and Control (AVC) トラフィックの場合、システム上の 1 つ以上のレイヤ 2 またはレイヤ 3 の物理インターフェイスに設定できるフローモニターは 1 つのみです。
- Flexible NetFlow および NBAR は同じインターフェイスで同時に設定できません。

- レイヤ 2、IPv4、および IPv6 のトラフィック タイプがサポートされています。異なるトラフィック タイプの複数のフロー モニターを、指定したインターフェイスと方向に適用できます。同じトラフィック タイプの複数のフロー モニターを指定したインターフェイスと方向には適用できません。
- 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。
- IPv6 フローモニターの場合、送信元グループタグ（SGT）フィールドと宛先グループタグ（DGT）フィールドは、MAC アドレスフィールドと共存できません。
- Quality of Service（QoS）のマークが付けられたパケットが入力方向に NetFlow が設定されているインターフェイスで受信されると、パケットの QoS 値が NetFlow コレクタによってキャプチャされます。ただし、パケットが出力方向に設定された NetFlow を備えたインターフェイスで受信され、スイッチによって入力時に QoS 値が書き換えられた場合、パケットの新しい QoS 値はコレクタによってキャプチャされません。
- NetFlow レコードは、マルチプロトコル ラベル スイッチング 対応（MPLS 対応）インターフェイスをサポートしません。
- MPLS ネットワーク内の MPLS ラベルに基づくデータキャプチャはサポートされていません。MPLS タグ付きパケットの IP ヘッダーフィールドのキャプチャはサポートされていません。
- 出力フローモニターは、EoMPLS モードまたは L3VPN Per-Prefix モードで出力されるフローをキャプチャしません。
- フローモニターは、レイヤ 3 物理インターフェイスと論理インターフェイス（レイヤ 3 ポートチャンネル インターフェイス、レイヤ 3 ポートチャンネル メンバ、スイッチ 仮想 インターフェイス（SVI）など）間で共有することはできませんが、論理インターフェイス間またはレイヤ 3 物理インターフェイス間で共有できます。

Flexible NetFlow に関する情報

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パケットストリームで、キーの値は同じです。キーは、パケット内のフィールドを識別する値です。フローを作成するには、フローレコードを使用して、フロー固有のキーを定義します。

デバイスは、ネットワーク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。

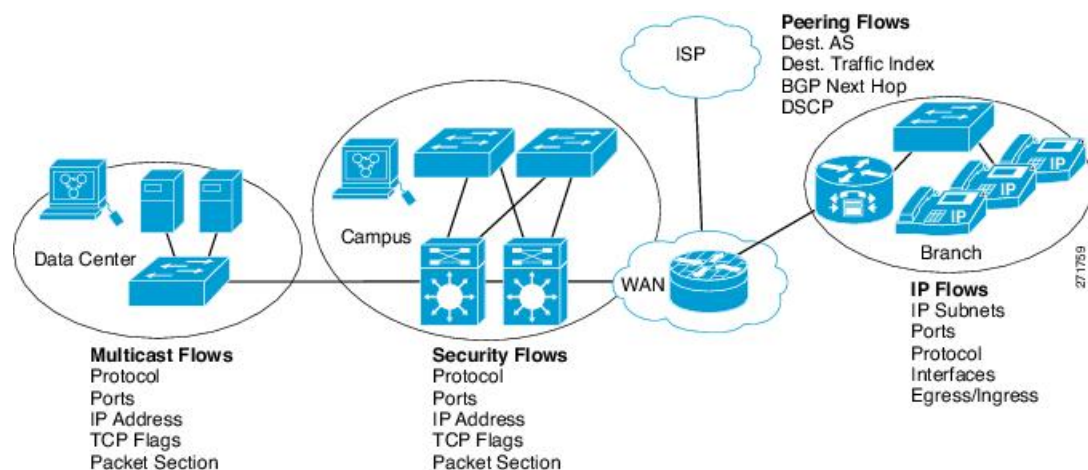
- スケーラビリティ、フロー情報の集約などの、大容量フロー認識。
- セキュリティの監視と dDoS の検出および識別のための拡張されたフローインフラストラクチャ。
- フロー情報をネットワーク内の特定のサービスまたはオペレーションに適応させるパケットからの新しい情報。利用できるフロー情報は、Flexible NetFlow ユーザーがカスタマイズ可能。
- Cisco の柔軟で拡張可能な NetFlow Version 9 の活用。
- IP アカウンティング、ボーダーゲートウェイプロトコル (BGP) ポリシーアカウンティング、永続的キャッシュなどの多数のアカウントリング機能を置換するため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

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

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

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

図 1: 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、およびキャッシュタイプの固有の組み合わせを設定できます。フローエクスポータの宛先 IP アドレスなどのパラメータを変更する場合、フローエクスポータを使用するすべてのフローモニタ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 1: 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 2: 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。

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

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。

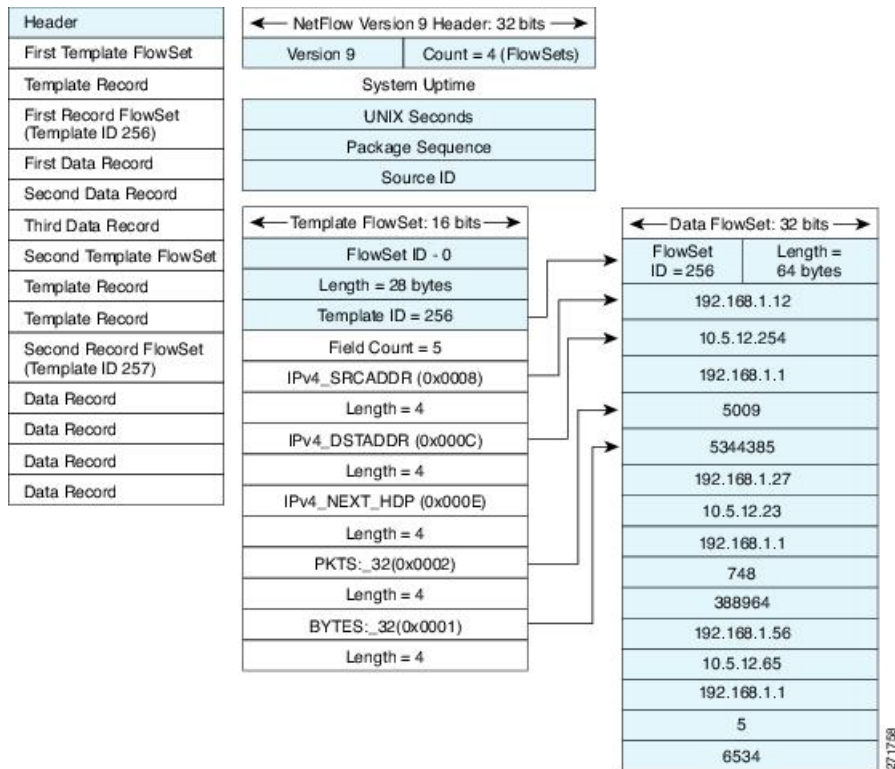
バージョン 9 のエクスポート フォーマットは、パケット ヘッダーとそれに続く 1 つ以上のテンプレート フローセットまたはデータ フローセットで構成されています。テンプレート フローセットでは、将来のデータ フローセットに表示されるフィールドの説明が提供されます。このようなデータ フローセットは、後で同じエクスポート パケットまたは後続のエクスポート パケットで発生する可能性があります。テンプレート フローセットおよびデータ フローセットは、次の図に示すように、単一のエクスポート パケットに混在させることができます。

図 2: 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 to 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 3: 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

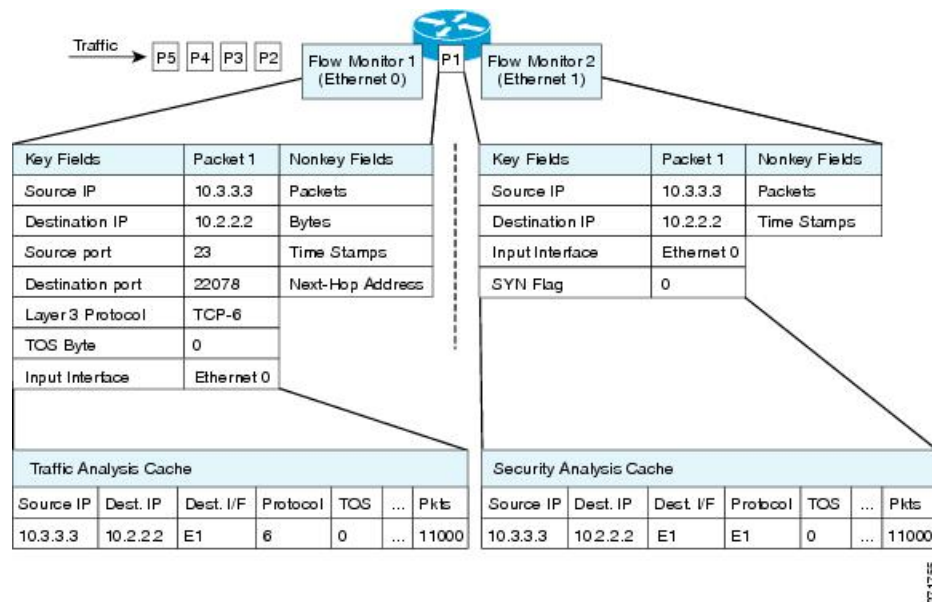
フロー モニター

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。

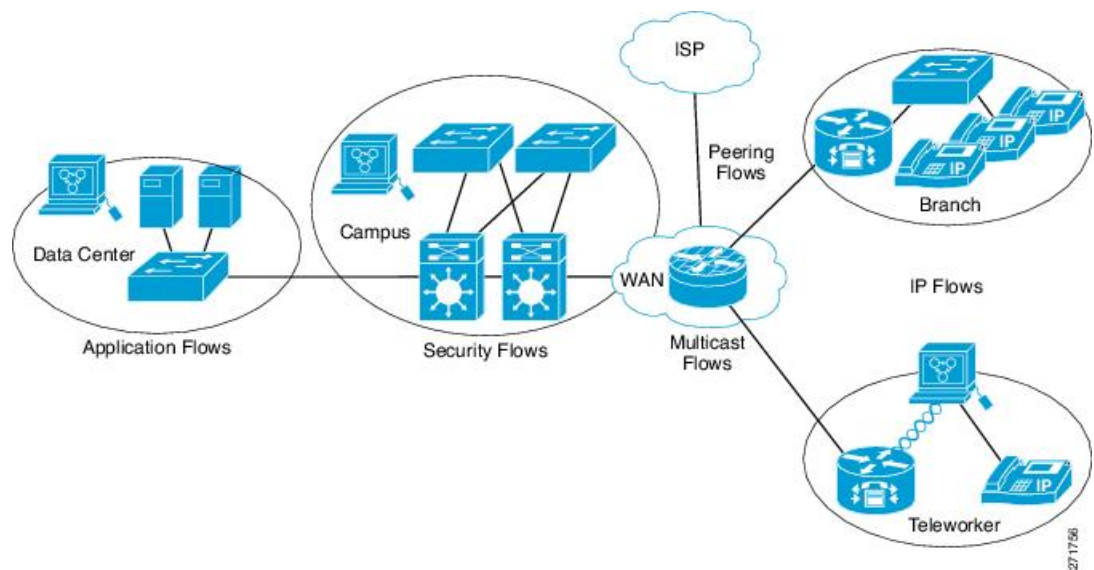
Flexible NetFlow は、同じトラフィックのさまざまなタイプの分析を実行するために使用できます。下の図では、入力インターフェイス上の標準トラフィック分析のために設計されたレコードと、出力インターフェイス上のセキュリティ分析のために設計されたレコードを使用してパケット 1 が分析されます。

図 4: 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 5: 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							

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPv4 Out	IPv6 In	IPv6 Out	注意
インターフェイス入力	あり	—	あり	—	あり	—	<p>フロー モニターを入力方向に適用する場合：</p> <ul style="list-style-type: none"> • match キーワードを使用し、入力インターフェイスを key フィールドとして使用します。 • collect キーワードを使用し、出力インターフェイスを collect フィールドとして使用します。このフィールドはエクスポートされるレコードに含まれますが、値は0になります。
インターフェイス出力	—	あり	—	あり	—	あり	<p>フロー モニターを出力方向に適用する場合：</p> <ul style="list-style-type: none"> • match キーワードを使用し、出力インターフェイスを key フィールドとして使用します。 • collect キーワードを使用し、入力インターフェイスを collect フィールドとして使用します。このフィールドはエクスポートされるレコードに含まれますが、値は0になります。

フィールド	レイヤ 2 In	レイヤ 2 Out	IPv4 In	IPv4 Out	IPv6 In	IPv6 Out	注意
Key フィールド							

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

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

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 バージョン	—	—	対応	対応	対応	対応	IP バージョンと同じです。
IPv6 プロトコル	—	—	対応	対応	対応	対応	IP プロトコルと同じです。送信元/宛先ポート、ICMP コード/タイプ、IGMP タイプ、TCP フラグのいずれかが使用されている場合に使用する必要があります。
IPv6 送信元アドレス	—	—	—	—	対応	対応	
IPv6 宛先アドレス	—	—	—	—	対応	対応	
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	

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
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。

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
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 3: デフォルトの Flexible NetFlow 設定

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

Flexible NetFlow : 入力 VRF サポートの概要

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

自律システム番号

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

自律システム番号 (AS 番号) は、主にボーダー ゲートウェイ プロトコルで使用される IANA によって割り当てられる特別な番号です。一意のルーティングポリシーを持つ単一の技術管理下にあるネットワーク、またはパブリックインターネットにマルチホーム接続されているネットワークを一意に識別します。この自律システム番号は、ピアリングポイントのインターネット サービスプロバイダとインターネット エクスチェンジ (IX) の間で、BGP およびピアをインターネット サービスプロバイダと実行するために必要です。AS 番号はグローバルに一意である必要があります。これにより、BGP が検出してルーティングできる一意の場所から IP アドレスブロックが送信されるようになります。BGP は、プレフィックスと自律システムパス (AS パス) を使用して、プレフィックスが存在する宛先への最短パスを決定します。

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

NetFlow では、次の BGP パラメータをエクスポートできます。

- BGP 送信元起源またはピア AS 番号

- BGP 宛先起源またはピア AS 番号

設定

AS 番号システムを設定するには、次のコマンドを使用します。

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

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。 <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 description 例： <pre>Device(config-flow-record)# description Used for basic traffic analysis</pre>	（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： <pre>Device(config-flow-record)# match ipv4 destination address</pre>	（注） この例では、IPv4宛先アドレスをレコードの key フィールドとして設定します。
ステップ 6	必要に応じてステップ 5 を繰り返し、レコードの追加 key フィールドを設定します。	—

	コマンドまたはアクション	目的
ステップ 7	<p>match flow cts {source destination} group-tag</p> <p>例 :</p> <pre>Device(config-flow-record)# match flow cts source group-tag</pre> <pre>Device(config-flow-record)# match flow cts destination group-tag</pre>	<p>(注)</p> <p>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。</p>

	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 の強制が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 : <pre>Device# show flow record FLOW_RECORD-1</pre>	(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 : <pre>Device# show running-config flow record FLOW_RECORD-1</pre>	(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 : <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(config)# configure terminal</pre>	グローバル コンフィギュレーション モードを開始します。
ステップ 3	flow exporter <i>name</i> 例 : <pre>Device(config)# flow exporter ExportTest</pre>	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 {ipv4-address} 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 value 例 : Device(config-flow-exporter) # dscp 0	(任意) DSCP (DiffServ コードポイン ト) 値を指定します。範囲は 0 ~ 63 で す。デフォルトは 0 です。
ステップ 7	source { } 例 : Device(config-flow-exporter) # source gigabitEthernet1/0/1	(任意) 設定された宛先で NetFlow コ ネクタに到達するために使用するイン ターフェイスを指定します。送信元と して次のインターフェイスを設定でき ます。
ステップ 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。

	コマンドまたはアクション	目的
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 <i>monitor-name</i> 例： Device(config)# flow monitor FLOW-MONITOR-1	フロー モニターを作成し、Flexible NetFlow フロー モニター コンフィギュレーション モードを開始します。 • このコマンドでは、既存のフロー モニターを変更することもできます。
ステップ 4	description <i>description</i> 例： Device(config-flow-monitor)# description Used for basic ipv4 traffic analysis	(任意) フローモニターの説明を作成します。
ステップ 5	record {<i>record-name</i> netflow-original netflow {ipv4 ipv6} record [peer]} 例： Device(config-flow-monitor)# record FLOW-RECORD-1	フローモニターのレコードを指定します。
ステップ 6	cache {timeout {active inactive update rate-limit} <i>seconds</i> type normal } 例： Device(config-flow-monitor)# cache type normal Device(config-flow-monitor)# cache timeout active	(任意) フロー モニター キャッシュ パラメータ (タイムアウト値、キャッシュタイプなど) を変更します。指定したフロー モニターとフロー キャッシュを関連付けます。
ステップ 7	必要に応じてステップ 6 を繰り返して、このフローモニターのキャッシュパラメータの変更を完了します。	—

	コマンドまたはアクション	目的
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 exporter-name 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。
ステップ 11	end 例 : <pre>Device(config-flow-monitor)# end</pre>	Flexible NetFlow フロー モニター コンフィギュレーション モードを終了して、特権 EXEC モードに戻ります。
ステップ 12	show flow monitor [[name] monitor-name [cache [format {csv record table}]] [statistics]] 例 : <pre>Device# show flow monitor FLOW-MONITOR-2 cache</pre>	(任意) Flexible NetFlow フロー モニターのステータスおよび統計情報が表示されます。
ステップ 13	show running-config flow monitor monitor-name 例 : <pre>Device# show running-config flow monitor FLOW_MONITOR-1</pre>	(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 : <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。

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。 <ul style="list-style-type: none">パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	sampler <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。 <ul style="list-style-type: none">このコマンドでは、既存のサンプラーを変更することもできます。
ステップ 4	description <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 <i>window-size</i> 例： Device(config-sampler)# mode random 1 out-of 2	サンプラーモードおよびフローサンプラーのウィンドウ サイズを指定します。 <ul style="list-style-type: none"><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。

	コマンドまたはアクション	目的
ステップ 8	{ip ipv6} flow monitor monitor-name [[sampler] sampler-name] {input output} 例 : <pre>Device(config-if)# ip flow monitor FLOW-MONITOR-1 sampler SAMPLER-1 input</pre>	作成したフローモニターおよびフローサンプラーをインターフェイスに割り当てて、サンプリングをイネーブルにします。
ステップ 9	end 例 : <pre>Device(config-if)# end</pre>	インターフェイス コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 10	show sampler sampler-name 例 : <pre>Device# show sampler SAMPLER-1</pre>	設定し有効化したフローサンプラーのステータスおよび統計情報を表示します。

インターフェイスへのフローの適用

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

手順

	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(config)# configure terminal</pre>	グローバル コンフィギュレーション モードを開始します。
ステップ 3	interface type 例 : <pre>Device(config)# interface GigabitEthernet1/0/1</pre>	<p>インターフェイス コンフィギュレーション モードを開始し、インターフェイスを設定します。</p> <p>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。</p> <p>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</p>

	コマンドまたはアクション	目的
		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	<p>{ip flow monitor ipv6 flow monitor datalink flow monitor} name [sampler name] {input output}</p> <p>例 :</p> <pre>Device(config-if)# ip flow monitor MonitorTest input</pre>	<p>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。</p> <p>ip flow monitor – Flexible NetFlow で IPv4 トラフィックを監視できます。</p> <p>ipv6 flow monitor – Flexible NetFlow で IPv6 トラフィックを監視できます。</p> <p>datalink flow monitor – Flexible NetFlow で非IPのトラフィックを監視できます。</p> <p>(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。</p>
ステップ 5	<p>end</p> <p>例 :</p> <pre>Device(config-flow-monitor)# end</pre>	特権 EXEC モードに戻ります。
ステップ 6	<p>show flow interface [interface-type number]</p> <p>例 :</p> <pre>Device# show flow interface</pre>	(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	<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。

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] vlan-id 例： Device(config)# vlan configuration 30 Device(config-vlan-config)#	VLAN または VLAN コンフィギュレーション モードを開始します。
ステップ 4	ip flow monitor monitor name [sampler sampler name] { input } 例： Device(config-vlan-config)# ip flow monitor MonitorTest input	入力パケットに対応する VLAN に、フロー モニターおよびオプションのサンプラーを関連付けます。
ステップ 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。 • 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)# <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	flow record name 例： Device(config)# <code>flow record L2_record</code> 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)# <code>match datalink ethertype</code>	レイヤ 2 属性をキーとして指定します。
ステップ 5	end 例： Device(config-flow-record)# <code>end</code>	特権 EXEC モードに戻ります。
ステップ 6	show flow record [name] 例： Device# <code>show flow record</code>	(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# <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。

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 4: Flexible NetFlow のモニタリングコマンド

コマンド	目的
<code>show flow exporter [broker export-ids name name statistics templates]</code>	NetFlow のフロー エクスポート情報と統計情報を表示します。
<code>show flow exporter [name exporter-name]</code>	NetFlow のフロー エクスポート情報と統計情報を表示します。
<code>show flow interface</code>	NetFlow インターフェイスに関する情報を表示します。
<code>show flow monitor [name exporter-name]</code>	NetFlow のフロー モニター情報と統計情報を表示します。
<code>show flow monitor statistics</code>	フロー モニターの統計情報を表示します。
<code>show flow monitor cache format {table record csv}</code>	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。
<code>show flow record [name record-name]</code>	NetFlow のフローレコード情報を表示します。
<code>show sampler [broker name name]</code>	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
```

例 : IPv4 入カトラフィックのモニタリング

```

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 入カトラフィックのモニタリング

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

```

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

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

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

Device(config) # flow exporter fe-1
Device(config-flow-exporter) # destination 10.5.120.16
Device(config-flow-exporter) # source Vlan105

```

```

Device(config-flow-exporter)# dscp 32
Device(config-flow-exporter)# ttl 200
Device(config-flow-exporter)# transport udp 2055

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

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

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

```

例：IPv4 出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

```

例：入力 VRF サポート用の Flexible NetFlow の設定

```

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
Device(config-if)# ip vrf forwarding green
Device(config-if)# ip address 172.16.2.2 255.255.255.252
Device(config-if)# ip flow monitor mm_1 input
Device(config-if)# end

```

Flexible NetFlow の機能情報

リリース	変更内容
Cisco IOS XE Everest 16.5.1a	この機能が導入されました。
Cisco IOS XE Gibraltar 16.12.1	IPv6 トラフィックについて、FNF の SGT フィールドと DGT フィールドのサポートが導入されました。

リリース	変更内容
Cisco IOS XE Amsterdam 17.1.1	MPLS（IP レベル）での入力および出力 Flexible Netflow のサポートが導入されました。

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。