組み込みパケット キャプチャ

組み込みパケットキャプチャの概要

パケットキャプチャは、ネットワークを通過するデータパケットを傍受して記録するために使用される、基本的なネットワーク診断技術です。これらのパケットには、プロトコルヘッダーやペイロードデータなど、デバイス間で交換される未加工の情報が含まれています。ネットワークエンジニアは、パケットをキャプチャして分析することにより、ネットワークの動作を可視化し、接続の問題、パフォーマンスのボトルネック、セキュリティインシデントのトラブルシュートを可能にします。

パケット データ キャプチャ

これは、分析のためにバッファに保存されるデータパケットをキャプチャするプロセスです。一意の名前を割り当て、特定のパラメータを定義することによって、パケットキャプチャを作成できます。

次の操作を実行できます。

- 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。
- 不要になった場合のキャプチャの破棄。
- 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。バッファサイズは 1MB ～ 100MB の範囲です。デフォルトのバッファタイプは線形であり、代替として循環バッファリングが利用可能です。
- キャプチャされたトラフィックをフィルタ処理するために、プロトコル、IP アドレス、またはポート番号に基づいて一致基準を定義します。

パケットキャプチャは、複雑なネットワークのやり取りを理解し、プロトコル運用を検証するのに必須です。従来、パケットキャプチャは、ネットワークタップやスパン/ミラーポートなどの外部デバイスを使用して実行され、分析ツールへのトラフィックを複製していました。

パケットキャプチャは、ローカルで使用することも、組み込みパケットキャプチャ（EPC）やWiresharkなどのツールを使用してオフライン分析用にエクスポートすることもできます。EPCとは、外部ハードウェアを必要とせずにデバイスでパケットを直接キャプチャおよびフィルタリングできるシスコのオンデバイスパケットキャプチャ機能のことを指します。一方、Wiresharkは、ネットワークトラフィックの詳細な検査と障害対応に広範に使用されている強力なオープンソース パケット アナライザです。

組み込みパケットキャプチャ

組み込みパケットキャプチャ（EPC）は、外部ハードウェアまたはポートミラーリングを必要とせずに、ネットワークデバイス上で直接パケットキャプチャを有効にするシスコの機能です。EPCは、デバイスの内部リソースを利用して、指定されたインターフェイスでトラフィックをキャプチャし、オンボードバッファまたはファイルに一時的にデータを保存します。その後、キャプチャしたパケットを分析用にエクスポートします。

EPC には以下の利点があります。

オンデバイスキャプチャ：追加のハードウェアは必要ありません。

詳細なフィルタリング：特定のトラフィックタイプまたはフローをキャプチャします。

低影響：デバイスのパフォーマンス低下を最小限に抑えるための効率的なリソースを使用します。

柔軟性：物理インターフェイスまたは他の論理インターフェイスでキャプチャします。

組み込みパケット キャプチャの利点

- このデバイスは、MACフィルタを使用して、または任意のMACアドレスに一致させて、IPv4 および IPv6 パケットだけでなく、非 IP パケットもキャプチャできます。
- 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、パケットをキャプチャしてバッファに関連付けるために使用されるトラフィック中継の場所です。
- パケットキャプチャは、パケットキャプチャファイル（PCAP）形式でエクスポートできます。この形式は、外部ツールを使用した分析に適しています。
- 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。

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 *1 : 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。

組み込みパケットキャプチャの設定の制約事項

次の制約事項が組み込みパケットキャプチャ（EPC）に適用されます。

- VRF、管理ポート、またはプライベート VLAN を接続ポイントとして使用することはできません。
- 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。
- ユーザーがスイッチポートからルーテッドポート（レイヤ 2 からレイヤ 3）へ、またはその逆へインターフェイスを変更した場合、インターフェイスが再び起動したときに、そのキャプチャポイントを削除し、新しいファイルを作成する必要があります。キャプチャ ポイントの停止/開始が機能しません。
- インターフェイスの出力方向にキャプチャされたパケットは、デバイスの書き換えによって行われた変更（TTL、VLAN タグ CoS、チェックサム、および MAC アドレス、DSCP、プレシデント、UP など）が反映されないこともあります。
- パケットキャプチャの最小設定可能期間は 1 秒ですが、パケットキャプチャは少なくとも 2 秒間機能します。
- キャプチャがすでにアクティブである、または開始されている場合、キャプチャ ポイント パラメータを変更することはできません。
- EPC は、入力のマルチキャストパケットのみをキャプチャし、出力の複製パケットはキャプチャしません。
- 入力および出力の両方のパケットの書き換え情報はキャプチャされません。
- CPU が挿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、これらのタイプのパケットはインターフェイス出力キャプチャで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 インターフェイスとレイヤ 2 パケット（ARP）をキャプチャすることはできません。
- 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。
- EPC は、Locator/ID Separation Protocol（LISP）インターフェイスおよびトンネルインターフェイスではサポートされていません。
- EPC は、Ethernet-over-MPLS（EoMPLS）ではサポートされていません。
- 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。

Wireshark とは何ですか？

Wireshark は、ネットワークキャプチャの詳細な調査に使用されるオープンソースのパケット分析ツールです。EPC はスイッチ自体でトラフィックをキャプチャしますが、Wireshark は PC またはワークステーションでそのデータを分析する環境を提供します。

一般的なワークフローは次のとおりです。

1. **キャプチャ**：EPC は、設定されたフィルタとインターフェイスに従ってデバイスでパケットを収集します。
2. **エクスポート**：キャプチャされたパケットは、プロトコルを使用して `.pcap` ファイルとしてスイッチからエクスポートされます。
3. **分析**：これらのキャプチャファイルは Wireshark で開き、プロトコルの復号、フィルタ、および可視化を適用して、ネットワークの問題を特定したり、動作を検証したりできます。

この統合プロセスにより、シスコの組み込みキャプチャ機能と Wireshark の分析ツールが統合され、専用のキャプチャアプライアンスを必要とせずに包括的なネットワークの障害対応が可能になります。

Wireshark の仕組み

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。

次のセクションでは、Wireshark に関連するコンポーネントについて説明します。

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。特定の Wireshark インスタンスの特性、例えばキャプチャするパケット、その送信元、キャプチャされたパケットに対して実行するアクション、停止条件などを概説します。キャプチャポイントは作成後に変更可能ですが、**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、アクティブにはなりません。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開始と呼ばれます。キャプチャポイントは名前で識別され、手動または自動で非アクティブ化または停止することができます。

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あり、キャプチャポイントフィルタに対してテストされます。

フィルタに一致するパケットがコピーされ、関連する Wireshark インスタンスに送信されます。特定のキャプチャポイントは複数の接続ポイントに関連付けることができますが、異なるタイプの接続ポイントの混在に制限があります。一部の制限は、異なるタイプの添付ポイントを指定すると適用されます。接続ポイントは、常に双方向であるレイヤ 2 VLAN の接続ポイントを除き、方向性あり（入力/出力/両方）です。

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

フィルタ

フィルタは、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す。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。
- 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。

（注） Wireshark はキャプチャフィルタの構文を使用しません。

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では、デバイス設定を変更する権限を取得する際、承認プロセスが長くなることで大幅な遅延が発生する可能性があります。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。

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へキャプチャ
- .pcap ファイルへ保存
- デコードおよび表示
- 保存および表示

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。

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

キャプチャされたパケットのストレージ

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

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

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

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

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

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

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

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 の機能

- 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、パケットがコピーされてCPUに送信されるため、CPU使用率が高くなります。.
- 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。

- 高い CPU 使用率を最小限に抑えるには、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。
  - フィルタルールに厳密に従ってください。不要なトラフィックを呼び込む可能性がある緩やかな ACL ではなく、厳密な ACL を使用してトラフィックタイプ（IPv4のみなど）を制限します。
  - 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ているアクセス制御リスト（ACL）を変更する場合は、キャプチャを再起動して変更を適用します。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。

Wireshark 設定の前提条件

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

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、 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）のみが許可されます。
- 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化は使用できません。
- 出力形式は、以前のリリースとは異なります。
- 期間制限がより長いまたはキャプチャ期間がない（**term len 0** コマンドを使用して auto-more サポートのない端末を使用した）Wireshark セッションでは、コンソールまたは端末が使用できなくなる場合があります。
- 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。

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。

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

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。

ステップ 6 **monitor capture** *capture-name* **start**

例：

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

トラフィック トレース ポイントでパケット データのバッファへのキャプチャを開始します。

ステップ 7 **monitor capture** *capture-name* **stop**

例：

```
Device# monitor capture mycap stop
```

トラフィック トレース ポイントでパケット データのキャプチャを停止します。

ステップ 8 **monitor capture** *capture-name* **export** *file-location/file-name*

例：

```
Device# monitor capture mycap export tftp://10.1.88.9/mycap.pcap
```

分析のためにキャプチャされたデータをエクスポートします。

ステップ 9 **end**

例：

```
Device# end
```

特権 EXEC モードに戻ります。

キャプチャされたデータのモニタリングとメンテナンス

キャプチャされたパケットデータのモニタリングとメンテナンスを行うには、次の作業を実行します。キャプチャバッファとキャプチャポイントの詳細が、手順中に表示されます。

手順

ステップ 1 **enable**

例：

```
Device> enable
```

特権 EXEC モードを有効にします。

パスワードを入力します（要求された場合）。

ステップ 2 **show monitor capture** *capture-buffer-name* **buffer dump**

例：

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

（任意）キャプチャ パケットの 16 進数ダンプおよびそのメタデータを表示します。

ステップ 3 **show monitor capture** *capture-buffer-name* **parameter**

例：

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

（任意）キャプチャを指定するために使用されたコマンドのリストを表示します。

ステップ 4 **debug epc capture-point**

例：

```
Device# debug epc capture-point
```

（任意）パケット キャプチャ ポイントのデバッグを有効にします。

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。

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. 不要になったらキャプチャポイントを削除します。

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、キャプチャポイントのすべてのパラメータを 1 つの **monitor capture** コマンドで定義できます。

（注） 機能するキャプチャポイントを設定するには、接続ポイントを定義し、キャプチャの方向を指定して、コアフィルタを設定します。

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 ポイントを作成する場合、コアフィルタを定義する必要はありません。

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

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*}{**interface** *interface-type interface-id* | **control-plane**}{**in** | **out** | **both**}

例：

```
Device# monitor capture mycap interface GigabitEthernet1/0/1 in
```

キャプチャ ポイントを定義し、キャプチャ ポイントが関連付けられている接続ポイントを指定し、キャプチャの方向を指定します。

キーワードの意味は次のとおりです。

- *capture-name*：定義するキャプチャ ポイントの名前を指定します（例では mycap が使用されています）。キャプチャ名の長さは 8 文字以下にしてください。英数字、アンダースコア（_）のみが許可されます
- （任意）**interface***interface-type interface-id*：キャプチャポイントが関連付けられる接続ポイントを指定します（例では GigabitEthernet1/0/1 が使用されています）。

（注）

オプションで、このコマンドインスタンス 1 つでこのキャプチャポイントの複数の接続ポイントおよびパラメータすべてを定義できます。これらのパラメータについては、キャプチャ ポイント パラメータの変更に関する手順で説明されています。範囲のサポートは、接続ポイントを追加および削除するためにも使用できます。

interface-type には次のいずれかのオプションを使用します。

- **AppGigabitEthernet**：接続ポイントを AppGigabitEthernet として指定します。
- **GigabitEthernet**：接続ポイントを GigabitEthernet として指定します。
- **vlan**：接続ポイントを VLAN として指定します。

  （注）
  このインターフェイスを接続ポイントとして使用する場合は、入力キャプチャのみが可能です。

- **capwap**：接続ポイントを CAPWAP トンネルとして指定します。

  （注）
  このインターフェイスを接続ポイントとして使用する場合、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。

ステップ 3 **monitor capture** {*capture-name*}[**match** {**any** | **ipv4 any any** | **ipv6**} **any any**}]

例：

```
Device# monitor capture mycap interface GigabitEthernet1/0/1 in match any
```

コア システムのフィルタを定義します。

（注）
コアフィルタが使用できなくなるため、CAPWAP のトンネリング インターフェイスを接続ポイントとして使用する場合はこの手順を実行しないでください。

キーワードの意味は次のとおりです。

- *capture-name*：定義するキャプチャ ポイントの名前を指定します（例では mycap が使用されています）。
- **match**：フィルタを指定します。定義されている最初のフィルタはコア フィルタです。

  （注）
  キャプチャポイント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。キャプチャポイントがすべての要件を満たしており、アクティブ化前にエラーを回避していることを確認します。

- **ipv4**：IP バージョン 4 のフィルタを指定します。

- **ipv6**：IP バージョン 6 のフィルタを指定します。

ステップ 4 **show monitor capture** {*capture-name*}[**parameter**]

例：

```
Device# show monitor capture mycap parameter
   monitor capture mycap interface GigabitEthernet1/0/1 in
   monitor capture mycap match any
```

ステップ 2 で定義したキャプチャ ポイント パラメータを表示し、キャプチャポイントを定義したことを確認します。

ステップ 5 **show capwap summary**

例：

```
Device# show capwap summary
```

ワイヤレス キャプチャの接続ポイントとして使用できる CAPWAP トンネルを表示します。

（注）
このコマンドは、ワイヤレス キャプチャを実行するために CAPWAP トンネルを接続ポイントとして使用している場合にのみ使用します。例の項の CAPWAP の例を参照してください。

ステップ 6 **show running-config**

例：

```
Device# show running-config
```

入力を確認します。

ステップ 7 **copy running-config startup-config**

例：

```
Device# copy running-config startup-config
```

（任意）コンフィギュレーション ファイルに設定を保存します。

例

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

```
Device# show capwap summary

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


Name   APName                           Type PhyPortIf Mode      McastIf
------ -------------------------------- ---- --------- --------- -------
```

```
Ca0    AP442b.03a9.6715                data Gi3/0/6   unicast   -


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

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

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

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

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

```
 11   8.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
 12   9.225986   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 13   9.225986   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 14   9.225986   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 15   9.231998   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 16   9.231998   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 17   9.231998   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 18   9.236987   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 19  10.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
 20  10.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
 21  12.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
 22  12.239993   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 23  12.244997   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 24  12.244997   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 25  12.250994   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 26  12.256990   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 27  12.262987   10.10.14.2 -> 10.10.14.32  DTLSv1.0 Application Data
 28  12.499974 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
 29  12.802012   10.10.14.3 -> 10.10.14.255 NBNS Name query NB WPAD.<00>
 30  13.000000 00:00:00:00:00:00 -> 3c:ce:73:39:c6:60 IEEE 802.11 Probe Request, SN=0,
FN=0, Flags=........
```

次のタスク

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モニタリング

パラメータは順番にリストされていますが、任意の順番で手順を実行して値を指定できます。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を変更する場合は、インタラクティブに確認する必要があります。

キャプチャポイントのパラメータを変更するには、次の手順を使用します。

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

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。

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
```

すでに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。

ステップ 7 **end**

例：

```
Device(config)# end
```

特権 EXEC モードに戻ります。

パラメータの変更

キャプチャ ファイルの関連付けまたは関連付け解除

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

パケット バーストの処理にメモリ バッファ サイズを指定する

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

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

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

パケットのイーサタイプの指定

```
MAC ACL:
Device(config)#mac access-list extended mac1
Device(config-ext-macl)#permit any any 0x806 0x0
Device(config-ext-macl)exit
Device(config)#monitor capture mycap access-list mac1

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

順番に表示されていますが、パラメータを削除する手順は任意の順序で実行できます。1 行、2 行、または複数行で削除できます。複数のエントリを含む可能性のある接続ポイントを除くすべてのパラメータを削除できます。

キャプチャポイントパラメータを削除するには、次の手順を実行します。

始める前に

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

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
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

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。

ステップ 6 **copy running-config startup-config**

例：

```
Device# copy running-config startup-config
```

（任意）コンフィギュレーション ファイルに設定を保存します。

次のタスク

削除したものと同じ名前の新規キャプチャポイントを定義します。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、コアシステムフィルタ、および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と、パケットはバッファに保管されます。ライブ表示（キャプチャ時の表示）は、ファイルおよびバッファモードの両方で使用できます。

表示フィルタ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。ただし、コアシステムフィルタでキャプチャされたパケットは表示され、デフォルトの表示モードは短時間です。

（注） CAPWAP トンネリングインターフェイスが接続ポイントとして使用される場合、コアフィルタは使用されません。したがって、それらを定義する必要はありません。

手順

ステップ 1 **enable**

例：

```
Device> enable
```

特権 EXEC モードを有効にします。

パスワードを入力します（要求された場合）。

ステップ 2 **monitor capture** {*capture-name*} **start** [**display** [**display-filter** *filter-string*]] [**brief** | **detailed** | **dump**]

例：

```
Device# monitor capture mycap start display display-filter "stp"
```

キャプチャポイントをアクティブ化し、「stp」を含むパケットだけが表示されるように表示をフィルタ処理します。

ステップ 3 **monitor capture** {*capture-name*} **stop**

例：

```
Device# monitor capture name stop
```

キャプチャ ポイントを非アクティブにします。

ステップ 4 end

例：

```
Device(config)# end
```

特権 EXEC モードに戻ります。

ステップ 5 show running-config

例：

```
Device# show running-config
```

入力を確認します。

ステップ 6 copy running-config startup-config

例：

```
Device# copy running-config startup-config
```

（任意）コンフィギュレーション ファイルに設定を保存します。

次のタスク

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

接続ポイントが定義されていない場合に、「アクティベーション時に接続ポイントがありません」というエラーが発生します。

```
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でください。

手順

ステップ 1 **enable**

例：

```
Device> enable
```

特権 EXEC モードを有効にします。

パスワードを入力します（要求された場合）。

ステップ 2 **monitor capture** {*capture-name*} [**clear** | **export** *filename*]

例：

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

clear：完全にバッファを削除します。

（注）
clear コマンドを実行すると、

- DNA Advantage ライセンスでは、このコマンドはバッファを削除せずにバッファの内容をクリアします
- 他のすべてのライセンスでは、このコマンドはバッファ自体を削除します。

Export：バッファでキャプチャされたパケットを保存し、バッファを削除します。

ステップ 3 **end**

例：

```
Device(config)# end
```

特権 EXEC モードに戻ります。

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。

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。