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

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 *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。

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 ポイントを複数の接続ポイントに関連付けることができます。異なるタイプ接続ポイントの混合に制限はありません。一部の制限は、異なるタイプの添付ポイントを指定すると適用されます。接続ポイントは、常に双方向であるレイヤ 2 VLAN の接続ポイントを除き、方向性あり（入力/出力/両方）です。

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

フィルタ

フィルタは、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。

（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ファイルを作成し、パケットを書き込みます。キャプチャポイントの作成時にファイルが存在する場合、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と、エラーが生成されます。

表示フィルタを、必要に応じて指定します。

Wireshark 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か通過するデータパケットをキャプチャできます。ネットワーク管理者は、キャプチャバッファサイズとタイプ（循環またはリニア）およびキャプチャする各パケットの最大バイト数を定義

する場合があります。パケットキャプチャレートは、詳細な管理制御を使用してスロットリングできます。たとえば、アクセスコントロールリストを使用して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 トランジット ポイントです。
- 外部ツールを使用した分析に適したパケット キャプチャ ファイル（PCAP）形式でパケット キャプチャをエクスポートする機能。
- さまざまな詳細レベルでキャプチャされたデータ パケットをデコードする方法。

パケット データ キャプチャ

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

こうしたキャプチャでは、次のアクションを実行できます。

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

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

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

パケットキャプチャの設定方法

ここでは、パケットキャプチャの設定について説明します。

Wireshark の設定方法

Wireshark を設定するには、次の基本的な手順を実行します。

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

キャプチャ ポイントの定義

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

（注） 接続ポイント、キャプチャの方向、およびコアフィルタを、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しません。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** {*capture-name*} {**interface** *interface-type interface-id* | **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
	`Device# monitor capture mycap interface GigabitEthernet1/0/1 in`	• *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* には次のいずれかを使用します。 • **GigabitEthernet**：接続ポイントを GigabitEthernet として指定します。 • **vlan**：接続ポイントを VLAN として指定します。 （注） このインターフェイスを接続ポイントとして使用する場合は、入力キャプチャのみが可能です。

	コマンドまたはアクション	目的
		• **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。

	コマンドまたはアクション	目的
		（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。 • **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。

	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 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。複数が可能な接続ポイントを除いて、任意のパラメータを削除できます。

キャプチャポイントのパラメータを削除するには、次の手順に従います。

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`
	`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

削除したものと同じ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。

	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** {*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。

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

キャプチャされたデータのモニタリングとメンテナンス

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

手順

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

```
        .... ..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 00 09 c9   ................
```

```
0030  8f 77 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   ................
0050  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   ................
0070  ab cd                                       ..

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

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

例：表示フィルタを使用した .pcap ファイルからのパケットの表示

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

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

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

例：.pcap ファイルにキャプチャされたパケットの数を表示

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

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

例：.pcap ファイルから単一パケット ダンプの表示

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

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

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

例：.pcap ファイルにキャプチャされたパケットの統計情報を表示

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

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

例：単純なキャプチャおよび表示

次の例は、レイヤ 3 インターフェイス ギガビット イーサネット 1/0/1 でトラフィックをモニタ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ドキャプチャの統計情報を表示します。

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

ステップ 7：次のように入力してパケットを表示します。

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

  1 0.000000000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=0/0, ttl=254
  2 0.002555000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=1/256, ttl=254
  3 0.006199000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=2/512, ttl=254
  4 0.009199000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=3/768, ttl=254
  5 0.011647000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=4/1024, ttl=254
  6 0.014168000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=5/1280, ttl=254
  7 0.016737000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=6/1536, ttl=254
  8 0.019403000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=7/1792, ttl=254
  9 0.022151000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
seq=8/2048, ttl=254
 10 0.024722000   10.10.10.2 -> 10.10.10.1   ICMP 114 Echo (ping) request  id=0x0031,
```

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

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

ステップ 8：次のように入力して、キャプチャ ポイントを削除します。

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

例：バッファのキャプチャの使用

次に、バッファのキャプチャを使用する例を示します。

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

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

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

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

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

```
Device# show monitor capture mycap capture-statistics
Capture statistics collected at software:
        Capture duration - 88 seconds
        Packets received - 1000
        Packets dropped - 0
```

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

ステップ 4：次のように入力してキャプチャを停止します。

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

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

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

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

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

ステップ 7：次のように入力してバッファのパケットを表示します。

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

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

パケットがバッファ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                           ........
        Data: 000000000b153063abcdabcdabcdabcdabcdabcdabcdabcd...
        [Length: 72]

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


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

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

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

```
0070  ab cd
```

ステップ 9：次のように入力してバッファをクリアします。

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

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

（注） バッファ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‹
```

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. Rea
son : Wireshark Session Ended
```

mycap.pcap ファイルには、キャプチャしたパケットが含まれます。

ステップ 4：次のように入力してパケットを表示します。

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

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

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

ステップ 5：次のように入力してキャプチャ ポイントを削除します。

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

組み込みパケット キャプチャの設定例

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

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

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

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

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

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

その他の参考資料

関連資料

関連項目	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://cfnng.cisco.com/ に進みます。

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。