CHAPTER 57

Wireshark の設定

（注） Wireshark は、Supervisor Engine 7-E、Supervisor Engine 7L-E、Catalyst 4500X-16、および Catalyst 4500X-32 でのみサポートされます。

（注） Wireshark は VSS でサポートされ、機能は、少数の設定変更を除き、スタンドアロン スイッチの機能と一致します。詳細については、「VSS での Wireshark の設定」（P.57-12）を参照してください。

IP Base および Enterprise Services フィーチャ セットでは、Cisco IOS Release XE 3.3.0SG 以降、Catalyst 4500 シリーズ スイッチは、パケット アナライザ プログラム Wireshark（旧名 Ethereal）をサポートします。これは、複数のプロトコルをサポートし、テキスト ベースのユーザ インターフェイスで情報を提供します。

この章で説明する内容は、次のとおりです。

（注） この章で使用するスイッチ コマンドの構文および使用方法の詳細については、次の URL で『*Cisco Catalyst 4500 Series Switch Command Reference*』と関連資料を参照してください。

http://www.cisco.com/en/US/products/hw/switches/ps4324/index.html

『Catalyst 4500 Series Switch Command Reference』に掲載されていないコマンドについては、より詳細な Cisco IOS ライブラリを参照してください。次の URL で『*Cisco IOS Command Reference*』と関連資料を参照してください。

http://www.cisco.com/en/US/products/ps6350/index.html

Wireshark について

ネットワーク内で何が起きているかを理解するには、トラフィックをキャプチャし、分析する機能が必要です。Cisco IOS Release XE 3.3.0SG より前では、Catalyst 4500 シリーズ スイッチは、このニーズに対応するために 2 つの機能、PAN および **debug platform packet** を提供していました。両機能には制限があります。SPAN は、パケットをキャプチャするのに理想的ですが、指定したローカルまたはリモートの宛先にパケットを転送することによりこれを実現しているだけで、ローカル表示や分析をサポートしていません。**debug platform packet** コマンドは、Catalyst 4500 シリーズ スイッチに固有で、ソフトウェア プロセス フォワーディング パスから生成されたパケットだけで動作します。限定的なローカル表示機能がありますが、分析機能のサポートはありません。

したがって、ハードウェアおよびソフトウェアの両方によって転送されるトラフィックに適用可能で、かつ強力なパケット キャプチャ、表示、および分析機能をサポートする（既知のインターフェイスの使用が望ましい）、トラフィックのキャプチャと分析のメカニズムが求められています。

Wireshark は以前 Ethereal と呼ばれていたオープンソース パケット アナライザ プログラムです。そのアナライザは何百ものプロトコルをサポートし、グラフィカルとテキスト ベースの両方のユーザ インターフェイスを備えています。

Wireshark は、.pcap と呼ばれる既知の形式を使用した、ファイルへのパケットのダンプをサポートし、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。この機能は、**start** コマンドを入力した場合にだけ実際に適用され、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。

（注）　Cisco IOS Release XE 3.3.0SG では、Wireshark でのグローバルなパケット キャプチャはサポートされません。

ここでは、Wireshark の一部の重要な概念について説明します。

- 「キャプチャ ポイント」（P.57-2）
- 「接続ポイント：インターフェイスとトラフィックの方向」（P.57-3）
- 「フィルタ」（P.57-3）
- 「処理」（P.57-4）
- 「キャプチャされたパケットのメモリ内バッファへの保存」（P.57-4）

キャプチャ ポイント

キャプチャ ポイントとは、Wireshark 機能の中央ポリシー定義です。これは Wireshark の特定のインスタンスに関連付けられたすべての特性（キャプチャ対象のパケット、パケットのキャプチャ元、キャプチャされたパケットについての処理内容、パケットのキャプチャ停止時点）を定義します。キャプチャ ポイントは作成後に変更する場合があり、**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 ポイントが同時に定義され、アクティブになる可能性があります。

接続ポイント：インターフェイスとトラフィックの方向

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。(キャプチャ ポイントの属性として接続ポイントと考えてください)。接続ポイントに影響するパケットはキャプチャ ポイントのフィルタでテストされます。一致するパケットは、キャプチャ ポイントの関連付けられた 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 ポイントは、以下に示すように、異なるタイプの混合接続ポイントに制限のある複数の接続ポイントに関連付けられている可能性があります。(一部の制限は異なるタイプの接続ポイントを指定すると適用されます)。常に双方向であるレイヤ 2 VLAN の接続ポイントを除いて、接続ポイントは方向性（入力 / 出力 / 両方）があります。

フィルタ

フィルタは、Wireshark にコピーされ、渡されるキャプチャ ポイントの接続ポイントを通過するトラフィックのサブセットを識別し制限するキャプチャ ポイントの属性です。Wireshark で表示されるためには、パケットは接続ポイントと、キャプチャ ポイントに関連付けられたすべてのフィルタも通過する必要があります。

フィルタには次の 3 種類があります。

- コア システム フィルタ：コア システム フィルタはハードウェアによって適用され、一致基準はハードウェアによって制限されます。このフィルタは、ハードウェア転送トラフィックが Wireshark の目的でソフトウェアにコピーするかどうかを決定します。
- キャプチャ フィルタ：キャプチャ フィルタは、Wireshark によって適用されます。その一致基準は、コア フィルタによってサポートされるものよりも詳細に表示されます。コア フィルタを通過するが、キャプチャ フィルタに失敗するパケットは CPU/ ソフトウェアにコピーされ、送信されますが、Wireshark プロセスによって廃棄されます。キャプチャ フィルタの構文は、表示フィルタの構文と同じです。

（注）　Catalyst 4500 シリーズ スイッチの Wireshark はキャプチャ フィルタの構文を使用しません。

- 表示フィルタ：表示フィルタは、Wireshark によって適用され、その一致基準はキャプチャ フィルタと似ています。表示フィルタに失敗したパケットは表示されません。

コア システム フィルタ

クラス マップまたは ACL による、または明示的に CLI によるなど、複数の方法でコア システム フィルタの一致基準を指定できます。

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スイッチ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を、トラフィックの監視および分析に制限します。この問題に対処するため、Wireshark は EXEC モード CLI からコア システム フィルタの一致基準を明示的に指定することをサポートします。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（基本的に、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）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。内部的には、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 マップの作成に使用されます。これらの暗黙的に構築されたクラス マップはスイッチの実行コンフィギュレーションに反映されず、NVGEN の対象ではありません。

（注）　ACL およびクラス マップの設定はシステムの一部であり、Wireshark 機能の側面ではありません。

キャプチャ フィルタ

キャプチャ フィルタはさまざまな条件に基づいて着信パケットをさらにフィルタリングするように Wireshark に指示することができます。Wireshark は、パケットを受信するとただちにキャプチャ フィルタを適用します。キャプチャ フィルタに失敗したパケットは格納も表示もされません。

スイッチはこのパラメータを受信し、Wireshark にそれを変更せずに渡します。Wireshark がアプリケーションのフィルタ定義を解析するため、定義の構文は、Wireshark の表示フィルタによって提供される構文となります。この構文と標準 IOS の構文は異なり、標準構文では表現できない ACL 一致基準を指定することができます。

（注）　キャプチャ フィルタの構文は、Wireshark の表示フィルタのものと同じです。このように、キャプチャの構文と表示フィルタの構文は、Catalyst 4500 シリーズ スイッチの Wireshark の実装と同じです。

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。表示フィルタの構文はキャプチャ フィルタと同じなので、キャプチャ フィルタも定義されている場合は、表示フィルタはなくてもかまいません。

キャプチャフィルタおよび表示フィルタの構文の詳細については、
http://wiki.wireshark.org/DisplayFilters を参照してください。

処理

Wireshark はライブ トラフィックまたは前の既存 .pcap ファイルで呼び出すことができます。ライブ トラフィックに対して起動されたとき、そのキャプチャ フィルタおよび表示フィルタを通過するパケットに対して次の 4 種類の処理を実行できます。

- デコード、分析、保存のためにメモリ内バッファへキャプチャ
- .pcap ファイルへ保存
- デコードおよび表示
- 保存および表示。

.pcap ファイルのみに対して起動された場合は、デコードと表示の処理だけが適用できます。

キャプチャされたパケットのメモリ内バッファへの保存

パケットは、メモリ内のキャプチャ バッファに格納して、後でデコード、分析、または .pcap ファイルへ保存できます。

キャプチャ バッファは線形モードまたは循環モードにすることができます。線形モードでは、バッファがいっぱいになると、新しいパケットが廃棄されます。循環モードでは、バッファがいっぱいになると新しいパケットを格納するために最も古いパケットから廃棄されます。必要に応じてバッファをクリアすることもできますが、このモードは、ネットワーク トラフィックのデバッグに主に使用されます。

.pcap ファイルへのキャプチャされたパケットの保存

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。

- Catalyst 4500 シリーズ スイッチのオンボード フラッシュ ストレージ（bootflash:）
- 外部フラッシュ ディスク（slot:）
- USB ドライブ（usb0:）

（注）　他のデバイスを使用しないでください。

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。ファイルが関連付けられるかキャプチャ ポイントをアクティブにするときにファイルがすでに存在する場合、Wireshark はファイルを上書きできるかどうかについて問い合わせます。これは、1 つのキャプチャ ポイントのみが、特定のファイル名に関連付けられている可能性があることを意味します。

Wireshark の書き込みプロセスの宛先が満杯になると、Wireshark はファイルの一部のデータで失敗します。したがって、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存在することを確認する必要があります。リリース IOS XE 3.3.0SG では、ファイルシステムが満杯であるというステータスは、一部のストレージ デバイスについて検出されません。

パケット全体ではなくセグメントだけを保持して、必要な記憶域を減らすことができます。通常、最初の 64 または 128 バイトを超える詳細は必要ではありません。デフォルトの動作は、パケット全体を保存することです。

ファイルシステムの処理およびファイルシステムへの書き込み時に、パケットのドロップの可能性を避けるために、Wireshark は必要に応じてメモリ バッファを使用して到着時にパケットを一時的に保持することができます。メモリ バッファのサイズは、キャプチャ ポイントが .pcap ファイルに関連付けられるときに指定できます。

パケットのデコードと表示

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 への負荷が高い場合があります。

Wireshark はさまざまなパケット形式のパケットの詳細をデコードして表示できます。**monitor capture name start** コマンドにより、次の方法またはモードで表示します。

- **brief** : パケットごとに 1 行表示します（デフォルト）。
- **detailed** : プロトコルがサポートされているすべてのパケットのすべてのフィールドをデコードして表示します。このモードでは、他の 2 種類のモードよりも多くの CPU が必要です。
- 16 進 **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。

decode および display オプションを付けて capture コマンドを開始すると、Wireshark からの出力は IOS に戻り、コンソールに変更されずに表示されます。

ライブ トラフィックの表示

Wireshark は、Catalyst 4500 シリーズ スイッチのコア システムからパケットのコピーを受信します。Wireshark は、残りのパケットをデコードして表示する不要なパケットを廃棄するため、キャプチャおよび表示フィルタを適用します。

.pcap ファイルからの表示

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。キャプチャ フィルタは、このシナリオでは適用されません。

パケットの保存と表示

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 フィルタおよびキャプチャ フィルタだけが適用できます。

Wireshark のキャプチャ ポイントをアクティブまたは非アクティブにする

Wireshark のキャプチャ ポイントが、接続ポイント、フィルタ、アクション、およびその他のオプションで定義されている場合、Wireshark はアクティブである必要があります。それまでは、キャプチャ ポイントは、実際にパケットをキャプチャしません。

キャプチャ ポイントがアクティブになる前に、一部の健全性チェックが実行されます。キャプチャ ポイントは、コア システム フィルタと接続ポイントのどちらも定義されていない場合はアクティブにできません。このチェックに失敗したキャプチャ ポイントをアクティブにしようとすると、エラーが生成されます。

キャプチャおよび表示フィルタを必要に応じて指定します。

Wireshark のキャプチャ ポイントはアクティブになると、複数の方法で非アクティブにできます。.pcap ファイルにパケットを格納するだけのキャプチャ ポイントは手動で停止することも、また時間制限またはパケット制限付きで設定することもでき、その後でキャプチャ ポイントは自動的に停止します。Wireshark のキャプチャ フィルタを通過するパケットのみが、パケットの制限のしきい値としてカウントされます。

Wireshark のキャプチャ ポイントがアクティブになると、固定レート フィルタがハードウェアに自動的に適用され、CPU が Wireshark によって指示されたパケットでフラッディングしないようになります。レート フィルタの欠点は、より多くのリソースを使用できる場合でも設定レートを超える連続的なパケットをキャプチャすることはできないことです。

機能の相互作用

ここでは、Wireshark と他の機能との相互作用について説明します。

- レイヤ 2 セキュリティ機能：レイヤ 2 セキュリティ機能（ポート セキュリティ、MAC アドレス フィルタリングおよびスパニングツリーなど）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。これは、SPAN の動作とは異なります。
- 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（ACL および IPSG など）によってドロップされたパケットは同じレイヤの接続ポイントに接続されている Wireshark のキャプチャ ポイントによってつかまえられません。これに対し、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レイヤの接続ポイントに接続されている Wireshark のキャ

プチャ ポイントによって捕捉されます。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、およびこれとは対称的に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 ポートおよびレイヤ 3 ポート チャネル：ルーテッド ポートまたはレイヤ 3 ポート チャネルが Wireshark の接続ポイントとして使用されている場合は、パケットをキャプチャするために適用されるポリシーは、レイヤ 3 で接続されるように扱われます。このように、Wireshark はインターフェイスによってルーティングされるパケットだけをキャプチャします。
- VLAN：VLAN が Wireshark の接続ポイントとして使用されている場合、パケットは、入力と出力の両方の方向でキャプチャされます。VLAN でブリッジングされるパケットは、2 つのコピー（1 つは入力用、もう 1 つは出力用）を生成します。
- プライベート VLAN：セカンダリ PVLAN は Wireshark の接続ポイントとして拒否されます。Wireshark の接続ポイントとしてプライマリ PVLAN を使用すると、プライマリおよび関連するすべてのセカンダリでパケットのキャプチャが有効になります。実質的に、PV ドメイン全体が接続ポイントになります。
- 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、最終的に別のレイヤ 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、このようなパケットをキャプチャします。対照的に、レイヤ 3（出力 WCCP など）によりリダイレクトされる出力機能は、レイヤ 3 の Wireshark の接続ポイントよりも論理的に前です。したがって、Wireshark はこれらをキャプチャしません。
- 他の分類コピー機能：ロールベースおよびセキュリティの検索タイプからパケットのコピーを生成する機能は、Wireshark と互換性があります。このようなパケットの複数のコピーが生成されます。
- SPAN：Wireshark と SPAN 送信元には互換性があります。SPAN 送信元および Wireshark 接続ポイントとしてインターフェイスを同時に設定することは、サポートされます。Wireshark の接続ポイントとして SPAN 宛先ポートを設定することはサポートされていません。

入力および出力分類には 4 つの分類の結果があります。入力方向は、ロール ベース、セキュリティ、QoS、転送の上書きの順序になります。出力方向は、転送の上書き、ロール ベース、セキュリティ、QoS の順序になります。

入力側では、Wireshark のキャプチャ機能は、転送の上書きの結果タイプになり、他の FO 機能（マルチキャスト ローカル ソースのキャプチャ、PBR、入力 WCCP など）の上で優先されます。これは、Wireshark でキャプチャされたパケットが、PBR または WCCP によるすべてのリダイレクトの前にあることを意味します。また、セキュリティ ACL が FO 関連機能よりも前に適用されるため、セキュリティ ACL によってドロップされたパケットは、Wireshark がキャプチャされません。

出力側では、Wireshark のキャプチャ機能は、転送上書きの結果タイプになり、他の FO 機能（出力 WCCP など）の下で優先されます。これは、他の出力 FO 機能が適用されない場合にだけ、Wireshark がパケットをキャプチャすることを意味します。

Wireshark の設定

Wireshark 機能を設定するための CLI は、EXEC モードだけから機能を実行する必要があるため非定型です。コンフィギュレーション サブモード（キャプチャ ポイントの定義など）で通常発生する処理は EXEC モードで代わりに扱われます。また、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。

ここでは、Wireshark を設定する方法について説明します。

デフォルトの Wireshark の設定

表 57-1 に、デフォルトの Wireshark の設定を示します。

表 57-1　デフォルトの Wireshark の設定

機能	デフォルト設定
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	No
バッファのストレージ モード	線形

Wireshark の設定時の注意事項

Wireshark を設定する場合は、次の点を確認します。

- トラフィックは、Wireshark ポリシーが適用されているインターフェイスでアクティブです。
- フィルタ規則がトラフィックに一致します。
- 必須パラメータが設定されます。

キャプチャ ポイントの定義、変更、削除

オプションの値を指定する手順は、順番にリストされますが、任意の順序で実行できます。1 行、2 行、または複数行で指定できます。複数となる接続ポイントを除き、次の順序で同じオプションを再指定することにより、値を最新の値で置き換えることができます。

ステップ 1　キャプチャ ポイントを識別する名前を定義します。

ステップ 2　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ます。

複数の接続ポイントが指定されることがあります。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。

ステップ 3　ACL または class-map で明示的に定義されたコア システム フィルタを定義します。

ステップ 4　セッション制限を指定します（キャプチャされた秒単位またはパケット単位で）。

ステップ 5　Wireshark が保持するパケット セグメント長を指定します。

ステップ 6　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。

ステップ 7　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。

キャプチャ ポイントを定義、変更、または削除するには、次のコマンドを入力します。

コマンド	目的
キャプチャ バッファの内容をクリアするか、ファイルにパケットを保存するには、**monitor capture [clear | export** *filename***]** コマンドを使用します。 **monitor capture name** [**clear**] [**export** *filename*]	バッファの内容をエクスポートまたは消去するには。

コマンド	目的
方向を持つ 1 つ以上の接続ポイントを指定するには、**monitor capture** [**interface** | **vlan** | **control-plane**] コマンドを使用します。接続ポイントを削除するには、このコマンドの **no** 形式を使用します。 **monitor capture** *name* [{**interface** *name* | **vlan** *num* | **control-plane**} {**in** | **out** | **both**} キャプチャ宛先を指定するには、**monitor capture** コマンドを使用します。詳細を削除するには、このコマンドの **no** 形式を使用します。 **monitor capture** *name* **[[file location** *filename* [**buffer-size** <*1-100*>] [**ring** <2-10>] [**size** <*1-100*>]] | [**buffer** [**circular**] **size** <*1-100*>]] キャプチャの制限を指定するには、**monitor capture limit** コマンドを使用します。制限を削除するには、このコマンドの **no** 形式を使用します。 **[no] monitor capture** *name* **limit** {**duration** *seconds*] [**packet-length** *size*] [**packets** *num*] 明示的にインライン コア フィルタを定義するには、**monitor capture mycap match** コマンドを使用します。これを削除するには、このコマンドの **no** 形式を使用します。 Switch# [**no**] **monitor capture mycap match** {**any** | **mac** *mac-match-string* | **ipv4** *ipv4-match-string* | **ipv6** *ipv6-match-string*} MAC にフィルタを使用するには、次の形式を使用します。フィルタを削除するには、このコマンドの **no** 形式を使用します。 Switch# [**no**] **monitor capture mycap match** *mac* {*src-mac-addr src-mac-mask* | **any** | **host** *src-mac-addr*} | {*dest-mac-addr dest-mac-mask* | **any** | **host** *dest-mac-addr*} IPv4/IPv6 にフィルタを使用するには、次の形式の 1 つを使用します。フィルタを削除するには、このコマンドの **no** 形式を使用します。 Switch# [**no**] **monitor capture mycap match** {**ipv4** | **ipv6**} [*src-prefix/length* | **any** | **host** *src-ip-addr*] [*dest-prefix/length* | **any** | **host** *dest-ip-addr*] Switch# [**no**] **monitor capture mycap match** {**ipv4** | **ipv6**} **proto** {**tcp** | **udp**} [*src-prefix/length* | **any** | **host** *src-ip-addr*] [**eq** | **gt** | **lt** | **neq** <*0-65535*>] [*dest-prefix/length* | **any** | **host** *dest-ip-addr*] [**eq** | **gt** | **lt** | **neq** <*0-65535*>] キャプチャ ポイントを開始または停止するには、**monitor capture** コマンドを使用します。 **monitor capture** *name* **start** [**capture-filter** *filter-string*] [**display** [**display-filter** *filter-string*]] [**brief** | **detailed** | **dump** | **stop**]	キャプチャ ポイントを指定されたパラメータで定義します。

例

ここでは、次の例を示します。

- 「キャプチャ ファイルの関連付け / 関連付け解除」（P.57-11）
- 「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」（P.57-11）
- 「IPv4 と IPv6 の両方の TCP トラフィックに一致するように、明示的なコア システム フィルタを定義する」（P.57-11）
- 「既存の ACL またはクラス マップを使用してコア システム フィルタを定義する」（P.57-11）

キャプチャ ファイルの関連付け / 関連付け解除

```
Switch# monitor capture point mycap file location bootdisk:mycap.pcap

Switch# 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

```
Switch# monitor capture mycap buffer-size 1000000
```

IPv4 と IPv6 の両方の TCP トラフィックに一致するように、明示的なコア システム フィルタを定義する

```
Switch# monitor capture mycap match any protocol tcp
```

既存の ACL またはクラス マップを使用してコア システム フィルタを定義する

```
Switch# monitor capture mycap match access-list myacl

Switch# monitor capture mycap match class-map mycm
```

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

接続ポイントとコア システム フィルタが定義されておらず、関連するファイル名（存在する場合）がすでにある場合、キャプチャ ポイントをアクティブにできません。関連するファイル名のないキャプチャ ポイントだけが、表示するためにアクティブにできます。キャプチャ フィルタまたは表示フィルタが指定されていない場合、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。

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、次のコマンドを入力します。

コマンド	目的
`Switch# monitor capture` *name* `start [capture-filter` *filter-string*`] [display [display-filter` *filter-string*`]] [brief | detailed | dump]`	キャプチャ ポイントをアクティブにします。
`Switch# monitor capture` *name* `stop`	キャプチャ ポイントを非アクティブにします。

例

キャプチャ フィルタを使用したキャプチャ ポイントのアクティブ化

```
Switch# monitor capture mycap start capture-filter "net 10.1.1.0 0.0.0.255 and port 80"
```

ライブ トラフィック用の表示フィルタを使用したキャプチャ ポイントのアクティブ化

```
Switch# monitor capture mycap start display display-filter "net 10.1.1.0 0.0.0.255 and
port 80"
```

VSS での Wireshark の設定

Wireshark 機能は、VSS でサポートされ、使用方法はスタンドアロン スイッチと若干異なります。接続ポイントが VSS アクティブ スイッチにあるか、VSS スタンバイ スイッチにあるかまたはその両方であるかによって、特定の処理を実行する必要があります。

VSS アクティブ スイッチと VSS スタンバイ スイッチで別々にキャプチャ ポイントを作成し、削除する必要があります。これらのスイッチでキャプチャ動作を別々に開始および停止する必要がありますが、両方のスイッチでキャプチャ セッションに同じ接続ポイントのセットを発行できます。個々のスイッチはそれに対してローカルではない接続ポイントを無視します。

VSS アクティブ スイッチでは、動作はスタンドアロン スイッチの動作と同じです。VSS スタンバイでは、ハードウェアが関連パケットをコピーし CPU に送信し、そこでそのパケットは、VSL リンクによって VSS アクティブ スイッチにソフトウェア トンネリングされます。VSS アクティブ スイッチの場合、これらのパケットはパケットがローカル ハードウェアから送信されたかのようにソフトウェアに引き渡されます。したがって、ローカル スイッチからのパケットとスタンバイ スイッチからのパケットは、VSS アクティブ スイッチの Wireshark セッションによって処理される一方、VSS スタンバイ スイッチは単に関連パケットをコピーし、それらを VSS アクティブ スイッチに渡します。VSS スタンバイの Wireshark セッションはその他の場合は未使用で、パケットは配信されません。

接続ポイントが VSS アクティブ スイッチだけにある場合、動作はスタンドアロン スイッチの動作と似ています。接続ポイントが両方のスイッチにあるか、VSS スタンバイ スイッチにのみある場合は、VSS アクティブ スイッチと VSS スタンバイ スイッチの両方でセッションを開始および停止する必要があります。

VSS スタンバイの Wireshark セッションはパケット キャプチャだけに役立ち、それ以外には関与しないことに注意してください。**display** オプションおよび関連するすべてのパラメータを指定して VSS スタンバイで Wireshark セッションを開始しないでください。

VSS スタンバイ スイッチで設定するには、「リモート ログイン」ファシリティを使用します。次に例を示します。

スタンバイ シャーシの設定

スタンバイ コンソールにはリモート ログインを使用してアクティブ シャーシからアクセスできます。

```
Predator_VSS# remote login module 11
Connecting to standby virtual console
Type "exit" or "quit" to end this session

Predator_VSS-standby-console# monitor capture mycap match any interface gi2/1/1 in file
location bootflash:text.pcap
```

VSS 設定での Wireshark の使用方法を理解するのに役立つ例がこのマニュアルで後述されています。

注意事項および制約事項

次のガイドラインに留意してください。

- パケット キャプチャが入力方向でイネーブルの場合、一致するパケットは、最初の 15 秒間、CPU でのソフトウェア ベースの検索が行われます。この間に、CPU 使用率が高く、キャプチャ レートが低くなります。

- パケット キャプチャが出力方向でイネーブルの場合、パケットは最初の 15 秒でキャプチャされません。
- 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、スイッチ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 アドレスなど）が反映されないこともあります。
- 別の論理ポートに属する物理ポートでのキャプチャはサポートされない場合があります。たとえば、EtherChannel メンバー ポートのキャプチャはサポートされません。
- ファイル サイズによる循環ファイル ストレージの制限はサポートされていません。
- Wireshark は、キャプチャ ポイントの class-map フィルタを次のいずれかに一致させようとすると、IPv6 パケットをキャプチャできません。
  - 拡張ヘッダーの次にホップバイホップ ヘッダーが続く（CSCtt16385 経由）
  - DSCP 値（CSCtx75765 経由）

注意すべき事項およびベスト プラクティス

次の点に注意してください。

- 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。
- 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。
- ストレージ ファイルにパケットを格納する場合は十分なスペースが Wireshark のキャプチャ プロセスを開始する前に使用できることを確認します。
- Wireshark のキャプチャ中の CPU 使用率は、指定された条件と一致するパケットの個数と、一致パケットに意図する処理（保存、デコードして表示、またはその両方）によって異なります。
- 可能な限り、キャプチャは最小限（パケット、期間などによる制限）に保ちます。そうしないと、CPU 使用率が高くなったり、その他の望ましくない状態になったりする可能性があります。
- 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。

  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。
- 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** コマンドの次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
- 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。

- 次のいずれかの場合に高い 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し、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。
- キャプチャ セッション中に、スイッチ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。これらの状況が発生する場合は、Wireshark セッションをすぐに停止します。
- 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。
- CPU またはメモリ リソースの浪費を避けるために、2 以下に Wireshark インスタンスの個数を制限します。

  Wireshark インスタンスは最大 8 個まで使用できます。.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。
- ACL が入力方向でスイッチにインストールされるか変更されるたびに、最初の 15 秒間、ソフトウェアはハードウェアによって送信されるパケット分類の詳細を無視します。代わりに、CPU で受信されるパケットに対してソフトウェア ベースの分類を使用します。したがって、この期間中、システムは（最初の 15 秒より後の時間に比べ）より少数のパケットだけをキャプチャすることができ、CPU 使用率が高くなります。

  （注）　出力方向では、パケットは最初の 15 秒間キャプチャされません。

- パケット損失を防ぐには、次の点を考慮します。
  - ライブ パケットをキャプチャしている間は、CPU に負荷のかかる操作である *デコードと表示* ではなく（特に detailed モードの場合）、*保存のみ* を使用します（**display** オプションを指定しない場合）。
  - デフォルトのバッファ サイズを使用する場合、パケットがドロップされる可能性があります。バッファ サイズを大きくし、パケット損失を避けてください。
  - さらに、フラッシュ ディスクへの書き込みは CPU に負荷のかかる操作であるため、キャプチャ レートは十分でない場合があります。
  - Wireshark キャプチャ セッションは、パケットのキャプチャおよび処理の両方が行われる *ストリーミング モード* で正常に動作しています。ただし、少なくとも 32 MB のバッファ サイズを指定すると、セッションは Wireshark のキャプチャ セッションがキャプチャおよびプロセスの 2 フェーズに分割される *ロックステップ モード* を自動的にオンにします。キャプチャ フェーズでは、パケットは一時バッファに保存されます。ロックステップ モードの duration パラメータは、セッション期間ではなくキャプチャ期間として機能します。バッファがいっぱいになるかまたはキャプチャ期間が終了すると、パケットの受け入れを停止してバッファのパケットの処理を開始するプロセス フェーズへセッションは移行します。2 番目のアプローチ（ロックステップ モード）では、より高いキャプチャ スループットが実現可能です。
  - ストリーミング キャプチャ モードは約 1500 pps をサポートし、ロックステップ モードはほぼ 45 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。

- コンソール ウィンドウのライブ パケットをデコードして表示する場合は、Wireshark セッションが短いキャプチャ期間によって抑制されていることを確認します。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。
- Wireshark を使用して、やむを得ず CPU 使用率が高くなるライブ トラフィックをキャプチャする場合には、キャプチャ処理が完了するまで、実際のトラフィックを制限するために、QoS ポリシーを一時的に適用することを検討してください。

Wireshark CLI に固有の注意事項

次の注意事項は、Wireshark CLI に関連します。

- すべての Wireshark 関連のコマンドは EXEC モードで、コンフィギュレーション コマンドは、Wireshark にありません。

  Wireshark CLI でアクセス リストまたはクラス マップを使用する必要がある場合は、コンフィギュレーション コマンドでアクセス リストおよびクラス マップを定義する必要があります。
- 特定の順序はキャプチャ ポイントを定義する場合には適用されません。CLI で許可されている任意の順序でキャプチャ ポイント パラメータを定義できます。Wireshark CLI は単一回線でできるだけ多くのパラメータを許可します。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、新しい値でコマンドを再入力することにより値を置き換えることができます。ユーザ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、新しい値の入力が不要です。ただし、パラメータの削除が必要です。
- 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、**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 int gi 3/1 in** では、インターフェイス gi 3/1 が接続ポイントです。

  また gi 3/2 に接続する必要がある場合、次のように、別の回線で指定します。

  monitor capture mycap int gi 3/2 in
- セッションがアクティブな間、キャプチャ ポイントのパラメータはいずれも変更できません。いずれかのパラメータを変更するには、セッションを停止し、変更を行い、セッションを再開します。アクセス リストは、スイッチに汎用で Wireshark プロセスに関係しないため、Wireshark セッション中に変更できます。
- 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** コマンドを発行すると、Wireshark はすべての必須パラメータが入力されたと判断した後でのみ開始します。
- キャプチャ ファイルがすでに存在する場合、警告を提供し、続行する前に確認を受け取ります。これは、誤ってファイルを上書きすることを防止します。
- コア フィルタは明示的なフィルタ、アクセス リスト、またはクラス マップにできます。これらのタイプの新しいフィルタを指定すると、既存のものを置き換えます。
- 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。

Wireshark 情報の表示

2 種類の **show** コマンドが Wireshark でサポートされます（表 57-2）。

表 57-2　MLD スヌーピング情報表示用のコマンド

コマンド	目的
show monitor capture point *name*	キャプチャ ポイント ステートが表示され、キャプチャ ポイントの定義状況、属性、アクティブ状況を確認することができます。キャプチャ *ポイントの名前*を指定すると、特定のキャプチャ ポイントの細部が表示されます。
show monitor capture file *name* [**display-filter** *filter-string*] [**brief** | **detailed** | **dump**]	パケットの送信元として既存の .pcap ファイルを使用して Wireshark をアクティブにします。表示フィルタが指定されていない場合は、ファイルの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。

例

ここでは、次の例を示します。

- 「.pcap ファイルからの概要出力の表示」（P.57-16）
- 「.pcap ファイルからの詳細出力の表示」（P.57-17）
- 「.pcap ファイルからの 16 進ダンプ出力の表示」（P.57-18）
- 「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」（P.57-19）

.pcap ファイルからの概要出力の表示

次のように入力して出力を表示できます。

```
Switch# show monitor capture file bootflash:mycap.pcap
  1   0.000000   10.1.1.140 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2   1.000000   10.1.1.141 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3   2.000000   10.1.1.142 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4   3.000000   10.1.1.143 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5   4.000000   10.1.1.144 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6   5.000000   10.1.1.145 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7   6.000000   10.1.1.146 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8   7.000000   10.1.1.147 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9   8.000000   10.1.1.148 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 10   9.000000   10.1.1.149 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 11  10.000000   10.1.1.150 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 12  11.000000   10.1.1.151 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 13  12.000000   10.1.1.152 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 14  13.000000   10.1.1.153 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 15  14.000000   10.1.1.154 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 16  15.000000   10.1.1.155 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 17  16.000000   10.1.1.156 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 18  17.000000   10.1.1.157 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 19  18.000000   10.1.1.158 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 20  19.000000   10.1.1.159 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 21  20.000000   10.1.1.160 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 22  21.000000   10.1.1.161 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 23  22.000000   10.1.1.162 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 24  23.000000   10.1.1.163 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 25  24.000000   10.1.1.164 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 26  25.000000   10.1.1.165 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 27  26.000000   10.1.1.166 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 28  27.000000   10.1.1.167 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 29  28.000000   10.1.1.168 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 30  29.000000   10.1.1.169 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

```
 31  30.000000   10.1.1.170 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 32  31.000000   10.1.1.171 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 33  32.000000   10.1.1.172 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 34  33.000000   10.1.1.173 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 35  34.000000   10.1.1.174 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 36  35.000000   10.1.1.175 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 37  36.000000   10.1.1.176 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 38  37.000000   10.1.1.177 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 39  38.000000   10.1.1.178 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 40  39.000000   10.1.1.179 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 41  40.000000   10.1.1.180 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 42  41.000000   10.1.1.181 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 43  42.000000   10.1.1.182 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 44  43.000000   10.1.1.183 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 45  44.000000   10.1.1.184 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 46  45.000000   10.1.1.185 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 47  46.000000   10.1.1.186 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 48  47.000000   10.1.1.187 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 49  48.000000   10.1.1.188 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 50  49.000000   10.1.1.189 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 51  50.000000   10.1.1.190 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 52  51.000000   10.1.1.191 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 53  52.000000   10.1.1.192 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 54  53.000000   10.1.1.193 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 55  54.000000   10.1.1.194 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 56  55.000000   10.1.1.195 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 57  56.000000   10.1.1.196 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 58  57.000000   10.1.1.197 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 59  58.000000   10.1.1.198 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

.pcap ファイルからの詳細出力の表示

次のように入力して出力を表示できます。

```
Switch# show monitor capture file bootflash:mycap.pcap detailed
Frame 1: 256 bytes on wire (2048 bits), 256 bytes captured (2048 bits)
    Arrival Time: Mar 21, 2012 14:35:09.111993000 PDT
    Epoch Time: 1332365709.111993000 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: 256 bytes (2048 bits)
    Capture Length: 256 bytes (2048 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:udp:data]
Ethernet II, Src: 00:00:00:00:03:01 (00:00:00:00:03:01), Dst: 54:75:d0:3a:85:3f
(54:75:d0:3a:85:3f)
    Destination: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
        Address: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Source: 00:00:00:00:03:01 (00:00:00:00:03:01)
        Address: 00:00:00:00:03:01 (00:00:00:00:03:01)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Type: IP (0x0800)
    Frame check sequence: 0x03b07f42 [incorrect, should be 0x08fcee78]
Internet Protocol, Src: 10.1.1.140 (10.1.1.140), Dst: 20.1.1.2 (20.1.1.2)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 238
    Identification: 0x0000 (0)
```

```
    Flags: 0x00
        0... .... = Reserved bit: Not set
        .0.. .... = Don't fragment: Not set
        ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: UDP (17)
    Header checksum: 0x5970 [correct]
        [Good: True]
        [Bad: False]
    Source: 10.1.1.140 (10.1.1.140)
    Destination: 20.1.1.2 (20.1.1.2)
User Datagram Protocol, Src Port: 20001 (20001), Dst Port: 20002 (20002)
    Source port: 20001 (20001)
    Destination port: 20002 (20002)
    Length: 218
    Checksum: 0x6e2b [validation disabled]
        [Good Checksum: False]
        [Bad Checksum: False]
Data (210 bytes)

0000  00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f   ................
0010  10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f   ................
0020  20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f    !"#$%&'()*+,-./
0030  30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f   0123456789:;<=>?
0040  40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f   @ABCDEFGHIJKLMNO
0050  50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f   PQRSTUVWXYZ[\]^_
0060  60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f   `abcdefghijklmno
0070  70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f   pqrstuvwxyz{|}~.
0080  80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e 8f   ................
0090  90 91 92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e 9f   ................
00a0  a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae af   ................
00b0  b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc bd be bf   ................
00c0  c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf   ................
00d0  d0 d1                                            ..
    Data: 000102030405060708090a0b0c0d0e0f1011121314151617...
    [Length: 210]

Frame 2: 256 bytes on wire (2048 bits), 256 bytes captured (2048 bits)
    Arrival Time: Mar 21, 2012 14:35:10.111993000 PDT
```

.pcap ファイルからの 16 進ダンプ出力の表示

次のように入力して出力を表示できます。

```
Switch# show monitor capture file bootflash:mycap.pcap dump
  1   0.000000   10.1.1.140 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00   Tu.:.?........E.
0010  00 ee 00 00 00 00 40 11 59 70 0a 01 01 8c 14 01   ......@.Yp......
0020  01 02 4e 21 4e 22 00 da 6e 2b 00 01 02 03 04 05   ..N!N"..n+......
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15   ................
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25   .......... !"#$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35   &'()*+,-./012345
0060  36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45   6789:;<=>?@ABCDE
0070  46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55   FGHIJKLMNOPQRSTU
0080  56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65   VWXYZ[\]^_`abcde
0090  66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75   fghijklmnopqrstu
00a0  76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85   vwxyz{|}~.......
00b0  86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95   ................
00c0  96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5   ................
00d0  a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5   ................
```

```
00e0  b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5   ................
00f0  c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 03 b0 7f 42   ...............B

  2   1.000000   10.1.1.141 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00   Tu.:.?........E.
0010  00 ee 00 00 00 00 40 11 59 6f 0a 01 01 8d 14 01   ......@.Yo......
0020  01 02 4e 21 4e 22 00 da 6e 2a 00 01 02 03 04 05   ..N!N"..n*......
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15   ................
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25   .......... !"#$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35   &'()*+,-./012345
0060  36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45   6789:;<=>?@ABCDE
0070  46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55   FGHIJKLMNOPQRSTU
0080  56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65   VWXYZ[\]^_`abcde
0090  66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75   fghijklmnopqrstu
00a0  76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85   vwxyz{|}~.......
00b0  86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95   ................
00c0  96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5   ................
00d0  a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5   ................
00e0  b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5   ................
00f0  c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 95 2c c3 3f   ............,.?

  3   2.000000   10.1.1.142 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00   Tu.:.?........E.
0010  00 ee 00 00 00 00 40 11 59 6e 0a 01 01 8e 14 01   ......@.Yn......
0020  01 02 4e 21 4e 22 00 da 6e 29 00 01 02 03 04 05   ..N!N"..n).....
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15   ................
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25   .......... !"#$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35   &'()*+,-./012345
0060  36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45   6789:;<=>?@ABCDE
0070  46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55   FGHIJKLMNOPQRSTU
0080  56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65   VWXYZ[\]^_`abcde
0090  66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75   fghijklmnopqrstu
00a0  76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85   vwxyz{|}~.......
00b0  86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95   ................
00c0  96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5   ................
00d0  a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5   ................
00e0  b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5   ................
00f0  c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 6c f8 dc 14   ............l...

  4   3.000000   10.1.1.143 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00   Tu.:.?........E.
0010  00 ee 00 00 00 00 40 11 59 6d 0a 01 01 8f 14 01   ......@.Ym......
0020  01 02 4e 21 4e 22 00 da 6e 28 00 01 02 03 04 05   ..N!N"..n(......
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15   ................
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25   .......... !"#$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35   &'()*+,-./012345
```

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を表示できます。

```
Switch# show monitor capture file bootflash:mycap.pcap display-filter "ip.src ==
10.1.1.140" dump
  1   0.000000   10.1.1.140 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00   Tu.:.?........E.
```

```
0010  00 ee 00 00 00 00 40 11 59 70 0a 01 01 8c 14 01   ......@.Yp......
0020  01 02 4e 21 4e 22 00 da 6e 2b 00 01 02 03 04 05   ..N!N"..n+......
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15   ................
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25   .......... !"#$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35   &'()*+,-./012345
0060  36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45   6789:;<=>?@ABCDE
0070  46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55   FGHIJKLMNOPQRSTU
0080  56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65   VWXYZ[\]^_`abcde
0090  66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75   fghijklmnopqrstu
00a0  76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85   vwxyz{|}~.......
00b0  86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95   ................
00c0  96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5   ................
00d0  a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5   ................
00e0  b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5   ................
00f0  c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 03 b0 7f 42   ...............B
```

使用例

ここでは、次の例を示します。

- 「例 1：単純なキャプチャおよび表示」（P.57-20）
- 「例 2：単純なキャプチャおよび保存」（P.57-21）
- 「例 3：バッファのキャプチャの使用」（P.57-22）
- 「例 4：キャプチャ セッション」（P.57-26）

例 1：単純なキャプチャおよび表示

たとえば、レイヤ 3 インターフェイス ギガビット 3/1 のトラフィックを監視するとします。

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。

```
Switch# monitor capture mycap interface gi 3/1 in match ipv4 any any
Switch# monitor capture mycap limit duration 60 packets 100
```

（注）　CPU 使用率の上昇を避けるには、制限として最も低いパケット数および時間を設定します。

ステップ 2　次のように入力することにより、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。

```
Switch# show monitor capture mycap parameter
   monitor capture mycap interface GigabitEthernet3/1 in
   monitor capture mycap match ipv4  any any
   monitor capture mycap limit packets 100 duration 60
Switch# show monitor capture mycap
Status Information for Capture mycap
  Target Type:
   Interface: GigabitEthernet3/1, Direction: in
  Status : Inactive
  Filter Details:
   IPv4
    Source IP:  any
    Destination IP:  any
   Protocol: any
  File Details:
   File not associated
  Buffer Details:
```

```
   Buffer Type: LINEAR (default)
  Limit Details:
   Number of Packets to capture: 100
   Packet Capture duration: 60
```

ステップ 3　次のように入力して、キャプチャ プロセスを開始し、画面に結果を表示します。

```
Switch# monitor capture mycap start display
  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 4　次の入力によってキャプチャ ポイントを削除します。

```
Switch# no monitor capture mycap
```

例 2：単純なキャプチャおよび保存

この例は、前のキャプチャおよび表示のシナリオと同様です。ライブ ストリームからのパケットを直接確認する代わりに、パケットはファイルにキャプチャされます。

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。

```
Switch# monitor capture mycap interface gi 3/1 in match ipv4 any any
Switch# monitor capture mycap limit duration 60 packets 100
Switch# monitor cap mycap file location bootflash:mycap.pcap
```

ステップ 2　次のように入力することにより、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。

```
Switch# show monitor capture mycap parameter
   monitor capture mycap interface GigabitEthernet3/1 in
   monitor capture mycap match ipv4  any any
   monitor capture mycap file location bootflash:mycap.pcap
   monitor capture mycap limit packets 100 duration 60
Switch# show monitor capture mycap
  Target Type:
   Interface: GigabitEthernet3/1, Direction: in
  Status : Inactive
  Filter Details:
   IPv4
    Source IP:  any
    Destination IP:  any
   Protocol: any
  File Details:
   Associated file name: bootflash:mycap.pcap
  Buffer Details:
   Buffer Type: LINEAR (default)
  Limit Details:
   Number of Packets to capture: 100
   Packet Capture duration: 60
```

ステップ 3　次の入力によりパケット キャプチャを起動します。

```
Switch# monitor capture mycap start
```

ステップ 4　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。

```
Switch# monitor capture mycap stop
```

（注）　あるいは、時間の経過またはパケット カウントが一致した後に、キャプチャ操作を自動的に停止させることもできます。

mycap.pcap ファイルには、キャプチャしたパケットが含まれます。

ステップ 5　次のように入力してパケットを表示します。

```
Switch# show monitor capture file bootflash:mycap.pcap
  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 6　次の入力によってキャプチャ ポイントを削除します。

```
Switch# no monitor capture mycap
```

例 3：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。

```
Switch# monitor capture mycap interface gi 3/1 in
Switch# monitor capture mycap match ipv4 any any
Switch# monitor capture mycap buffer circular size 1
Switch# monitor capture mycap start
```

ステップ 2　次のように入力することにより、キャプチャがアクティブであるかどうかを決定します。

```
Switch# show monitor capture mycap

Status Information for Capture mycap
  Target Type:
   Interface: GigabitEthernet3/1, Direction: in
  Status : Active
  Filter Details:
   IPv4
    Source IP:  any
    Destination IP:  any
   Protocol: any
  File Details:
   File not associated
  Buffer Details:
   Buffer Type: CIRCULAR
   Buffer Size (in MB): 1
  Limit Details:
```

```
   limit not set
```

ステップ 3　次のように入力してバッファのパケットを表示します。

```
Switch# show monitor capture mycap buffer brief
  0.000000   10.1.1.215 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  1.000000   10.1.1.216 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000   10.1.1.217 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000   10.1.1.218 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000   10.1.1.219 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000   10.1.1.220 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.000000   10.1.1.221 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.000000   10.1.1.222 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.000000   10.1.1.223 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.000000   10.1.1.224 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 10.000000   10.1.1.225 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 11.000000   10.1.1.226 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 12.000000   10.1.1.227 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 13.000000   10.1.1.228 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 14.000000   10.1.1.229 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 15.000000   10.1.1.230 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 16.000000   10.1.1.231 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 17.000000   10.1.1.232 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 18.000000   10.1.1.233 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 19.000000   10.1.1.234 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 20.000000   10.1.1.235 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 21.000000   10.1.1.236 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

パケットがバッファに入ったことに注意してください。

ステップ 4　次のように入力して他の表示モードでパケットを表示します。

```
Switch# show monitor capture mycap buffer detailed
Frame 1: 256 bytes on wire (2048 bits), 256 bytes captured (2048 bits)
    Arrival Time: Apr 15, 2012 15:50:02.398966000 PDT
    Epoch Time: 1334530202.398966000 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: 256 bytes (2048 bits)
    Capture Length: 256 bytes (2048 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:udp:data]
Ethernet II, Src: 00:00:00:00:03:01 (00:00:00:00:03:01), Dst: 54:75:d0:3a:85:3f
(54:75:d0:3a:85:3f)
    Destination: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
        Address: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Source: 00:00:00:00:03:01 (00:00:00:00:03:01)
        Address: 00:00:00:00:03:01 (00:00:00:00:03:01)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
…
Switch# show monitor capture mycap buffer dump
  0.000000   10.1.1.215 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00   Tu.:.?........E.
0010  00 ee 00 00 00 00 40 11 59 25 0a 01 01 d7 14 01   ......@.Y%......
0020  01 02 4e 21 4e 22 00 da 6d e0 00 01 02 03 04 05   ..N!N"..m.......
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15   ................
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25   .......... !"#$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35   &'()*+,-./012345
```

```
0060  36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45   6789:;<=>?@ABCDE
0070  46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55   FGHIJKLMNOPQRSTU
0080  56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65   VWXYZ[\]^_`abcde
0090  66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75   fghijklmnopqrstu
00a0  76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85   vwxyz{|}~.......
00b0  86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95   ................
00c0  96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5   ................
00d0  a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5   ................
00e0  b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5   ................
00f0  c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 03 3e d0 33   .............>.3
```

ステップ 5　次のように入力して、バッファを一度クリアし、10 秒待機した後でトラフィックを停止します。

```
Switch# monitor capture mycap clear
```

10 秒待機し、トラフィックを停止します。

次のように入力することにより、同じパケット セットがこの時間空白の後に表示されることを確認します。

```
Switch# show monitor capture mycap buffer brief
  0.000000     10.1.1.2 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  1.000000     10.1.1.3 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000     10.1.1.4 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000     10.1.1.5 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000     10.1.1.6 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000     10.1.1.7 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.000000     10.1.1.8 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.000000     10.1.1.9 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.000000    10.1.1.10 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.000000    10.1.1.11 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

[Wait for about 10 secs]

Switch# show monitor capture mycap buffer brief
  0.000000     10.1.1.2 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  1.000000     10.1.1.3 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000     10.1.1.4 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000     10.1.1.5 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000     10.1.1.6 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000     10.1.1.7 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.000000     10.1.1.8 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.000000     10.1.1.9 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.000000    10.1.1.10 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.000000    10.1.1.11 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

[Wait for about 10 secs]

Switch# show monitor capture mycap buffer brief
  0.000000     10.1.1.2 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  1.000000     10.1.1.3 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000     10.1.1.4 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000     10.1.1.5 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000     10.1.1.6 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000     10.1.1.7 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.000000     10.1.1.8 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.000000     10.1.1.9 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.000000    10.1.1.10 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.000000    10.1.1.11 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

ステップ 6　次のように入力してバッファからパケットをクリアします。

```
Switch# monitor capture mycap clear
```

ステップ 7　次のように入力することにより、バッファが現在空であることを確認してください。

```
Switch# show monitor capture mycap buffer brief
```

約 10 秒待機します。

ステップ 8　次のように入力して、バッファの内容を表示します。

```
Switch# show monitor capture mycap buffer brief
```

ステップ 9　トラフィックを再開し、10 秒待機してから、次の入力によってバッファの内容を表示します。

```
Switch# show monitor capture mycap buffer brief
  0.000000     10.1.1.2 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  1.000000     10.1.1.3 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000     10.1.1.4 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000     10.1.1.5 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000     10.1.1.6 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000     10.1.1.7 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.000000     10.1.1.8 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.000000     10.1.1.9 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.000000    10.1.1.10 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.000000    10.1.1.11 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 10.000000    10.1.1.12 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

ステップ 10　次の入力によって、バッファの内容を内部「bootflash:」ストレージ デバイスのファイル mycap1.pcap に保存します。

```
Switch# monitor capture mycap export bootflash:mycap1.pcap
Exported Successfully
```

ステップ 11　次のように入力することにより、ファイルが作成されたこと、また、そこにパケットが含まれていることを確認してください。

```
Switch# dir bootflash:mycap1.pcap
Directory of bootflash:/mycap1.pcap

14758  -rw-       20152  Apr 15 2012 16:00:28 -07:00  mycap1.pcap

831541248 bytes total (831340544 bytes free)
Switch# show monitor capture file bootflash:mycap1.pcap brief
  1   0.000000     10.1.1.2 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002
  2   1.000000     10.1.1.3 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002
  3   2.000000     10.1.1.4 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002
  4   3.000000     10.1.1.5 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002
  5   4.000000     10.1.1.6 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002
  6   5.000000     10.1.1.7 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002
  7   6.000000     10.1.1.8 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002
  8   7.000000     10.1.1.9 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002
  9   8.000000    10.1.1.10 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002
 10   9.000000    10.1.1.11 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002
 11  10.000000    10.1.1.12 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002
 12  11.000000    10.1.1.13 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002
```

```
 13  12.000000    10.1.1.14 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002
 14  13.000000    10.1.1.15 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002
 15  14.000000    10.1.1.16 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002
 16  15.000000    10.1.1.17 -> 20.1.1.2     UDP Source port: 20001  Destination port:
20002
```

ステップ 12　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。

```
Switch# monitor capture mycap stop
Switch# show monitor capture mycap buffer brief
  0.000000     10.1.1.2 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  1.000000     10.1.1.3 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000     10.1.1.4 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000     10.1.1.5 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000     10.1.1.6 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000     10.1.1.7 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.000000     10.1.1.8 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.000000     10.1.1.9 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.000000    10.1.1.10 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.000000    10.1.1.11 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 10.000000    10.1.1.12 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 11.000000    10.1.1.13 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

ステップ 13　次のように入力して、バッファをクリアしてから、バッファからのパケットを表示してみてください。

```
Switch# monitor capture mycap clear
Switch# show monitor capture mycap buffer brief
```

ステップ 14　次の入力によってキャプチャ ポイントを削除します。

```
Switch# no monitor capture mycap
```

例 4：キャプチャ セッション

次の例では、さまざまなモードでキャプチャ セッションを開始および停止する方法について説明します。

```
Switch# monitor capture mycap int gi 3/1 in match ipv4 any any
Switch# monitor capture mycap file location bootflash:mycap.pcap
Switch# monitor capture mycap limit packets 100 duration 60

Switch# monitor capture mycap start
Switch#
Switch# monitor capture mycap stop
Switch# monitor capture mycap start capture-filter "udp.port == 20001"
Switch# monitor capture mycap stop
Switch# monitor capture mycap start capture-filter "udp.port == 20001" display
A file by the same capture file name already exists, overwrite?[confirm]

  0.000000     10.1.1.9 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.10 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.11 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.12 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.13 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.14 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.15 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.16 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.17 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
```

```
  0.000000    10.1.1.18 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.19 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.20 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.21 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.22 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.23 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.24 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.25 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.26 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.27 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.28 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.29 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.30 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

Switch# monitor capture mycap start capture-filter "udp.port == 20001" display
display-filter "udp.port == 20002"
%Display-filter cannot be specified when capture is associated to a file. Ignoring
display  filter%
A file by the same capture file name already exists, overwrite?[confirm]

  0.000000    10.1.1.96 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.97 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.98 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000    10.1.1.99 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000   10.1.1.100 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000   10.1.1.101 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000   10.1.1.102 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000   10.1.1.103 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000   10.1.1.104 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000   10.1.1.105 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000   10.1.1.106 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000   10.1.1.107 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000   10.1.1.108 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  0.000000   10.1.1.109 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

Switch# monitor capture mycap start capture-filter "udp.port == 20001" display
display-filter "udp.port == 20002" detailed
%Display-filter cannot be specified when capture is associated to a file. Ignoring
display  filter%
A file by the same capture file name already exists, overwrite?[confirm]

Frame 1: 256 bytes on wire (2048 bits), 256 bytes captured (2048 bits)
    Arrival Time: Dec 31, 1969 17:00:00.000000000 PDT
    Epoch Time: 0.000000000 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: 256 bytes (2048 bits)
    Capture Length: 256 bytes (2048 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:udp:data]
Ethernet II, Src: 00:00:00:00:03:01 (00:00:00:00:03:01), Dst: 54:75:d0:3a:85:3f
(54:75:d0:3a:85:3f)
    Destination: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
        Address: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Source: 00:00:00:00:03:01 (00:00:00:00:03:01)
        Address: 00:00:00:00:03:01 (00:00:00:00:03:01)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
```

```
Switch# monitor capture mycap start capture-filter "udp.port == 20001" display dump
A file by the same capture file name already exists, overwrite?[confirm]

  0.000000     10.1.1.6 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00   Tu.:.?........E.
0010  00 ee 00 00 00 00 40 11 59 f6 0a 01 01 06 14 01   ......@.Y.......
0020  01 02 4e 21 4e 22 00 da 6e b1 00 01 02 03 04 05   ..N!N"..n.......
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15   ................
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25   .......... !"#$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35   &'()*+,-./012345
0060  36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45   6789:;<=>?@ABCDE
0070  46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55   FGHIJKLMNOPQRSTU
0080  56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65   VWXYZ[\]^_`abcde
0090  66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75   fghijklmnopqrstu
00a0  76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85   vwxyz{|}~.......
00b0  86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95   ................
00c0  96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5   ................
00d0  a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5   ................
00e0  b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5   ................
00f0  c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 ac 69 6e fd   ............in.

  0.000000     10.1.1.7 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

Switch# monitor capture mycap start display display-filter "udp.port == 20002"
%Display-filter cannot be specified when capture is associated to a file. Ignoring
display  filter%
A file by the same capture file name already exists, overwrite?[confirm]

  0.000000    10.1.1.41 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  1.000000    10.1.1.42 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000    10.1.1.43 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000    10.1.1.44 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000    10.1.1.45 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000    10.1.1.46 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.998993    10.1.1.47 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.998993    10.1.1.48 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.998993    10.1.1.49 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.998993    10.1.1.50 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.998993    10.1.1.51 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 10.998993    10.1.1.52 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

Switch# monitor capture mycap start display display-filter "udp.port == 20002" dump
%Display-filter cannot be specified when capture is associated to a file. Ignoring
display  filter%
A file by the same capture file name already exists, overwrite?[confirm]

  0.000000   10.1.1.117 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00   Tu.:.?........E.
0010  00 ee 00 00 00 00 40 11 59 87 0a 01 01 75 14 01   ......@.Y....u..
0020  01 02 4e 21 4e 22 00 da 6e 42 00 01 02 03 04 05   ..N!N"..nB......
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15   ................
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25   .......... !"#$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35   &'()*+,-./012345
0060  36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45   6789:;<=>?@ABCDE
0070  46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55   FGHIJKLMNOPQRSTU
0080  56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65   VWXYZ[\]^_`abcde
0090  66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75   fghijklmnopqrstu
00a0  76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85   vwxyz{|}~.......
00b0  86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95   ................
00c0  96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5   ................
00d0  a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5   ................
00e0  b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5   ................
```

```
00f0  c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 41 0c b4 5d   ............A..]

  1.000000   10.1.1.118 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002


Switch# no monitor capture mycap file

Switch# monitor capture mycap start display display-filter "udp.port == 20002" dump

  0.000000   10.1.1.160 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00   Tu.:.?........E.
0010  00 ee 00 00 00 00 40 11 59 5c 0a 01 01 a0 14 01   ......@.Y\......
0020  01 02 4e 21 4e 22 00 da 6e 17 00 01 02 03 04 05   ..N!N"..n.......
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15   ................
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25   .......... !"#$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35   &'()*+,-./012345
0060  36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45   6789:;<=>?@ABCDE
0070  46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55   FGHIJKLMNOPQRSTU
0080  56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65   VWXYZ[\]^_`abcde
0090  66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75   fghijklmnopqrstu
00a0  76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85   vwxyz{|}~.......
00b0  86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95   ................
00c0  96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5   ................
00d0  a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5   ................
00e0  b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5   ................
00f0  c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 9f 20 8a e5   ............. ..

  1.000000   10.1.1.161 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

Switch# monitor capture mycap start display display-filter "udp.port == 20002"

  0.000000   10.1.1.173 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  1.000000   10.1.1.174 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  2.000000   10.1.1.175 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  3.000000   10.1.1.176 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  4.000000   10.1.1.177 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  5.000000   10.1.1.178 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  6.000000   10.1.1.179 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  7.000000   10.1.1.180 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  8.000000   10.1.1.181 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
  9.000000   10.1.1.182 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 10.000000   10.1.1.183 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 11.000000   10.1.1.184 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002
 12.000000   10.1.1.185 -> 20.1.1.2     UDP Source port: 20001  Destination port: 20002

Switch# monitor capture mycap start display detailed

Frame 1: 256 bytes on wire (2048 bits), 256 bytes captured (2048 bits)
    Arrival Time: Apr 12, 2012 11:46:54.245974000 PDT
    Epoch Time: 1334256414.245974000 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: 256 bytes (2048 bits)
    Capture Length: 256 bytes (2048 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ip:udp:data]
Ethernet II, Src: 00:00:00:00:03:01 (00:00:00:00:03:01), Dst: 54:75:d0:3a:85:3f
(54:75:d0:3a:85:3f)
    Destination: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
        Address: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
```

```
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Source: 00:00:00:00:03:01 (00:00:00:00:03:01)
        Address: 00:00:00:00:03:01 (00:00:00:00:03:01)
        .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
        .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)

Switch#
```

VSS 固有の例

例 1：ファイルのキャプチャと保存（VSS アクティブの接続ポイント）

次に、接続ポイントが VSS アクティブ スイッチ自体にある場合に単純なキャプチャおよび保存操作を行う例を示します。

ステップ 1　VSS アクティブ スイッチで次のコマンドを入力して、キャプチャ セッションを起動します。

```
vss_dut1#monitor capture mycap interface gi 1/1/1 in
vss_dut1#monitor capture mycap match ipv4 any any
vss_dut1#monitor capture mycap file location bootflash:mycap.pcap
vss_dut1#monitor capture mycap limit packets 10 duration 10
vss_dut1#monitor capture mycap start
```

ステップ 2　キャプチャ セッションが終了したら、キャプチャ ファイルがパケットを保存したことを確認してください。

```
*Nov 15 00:04:08.337 PDT: %BUFCAP-6-ENABLE: Capture Point mycap enabled.
*Nov 15 00:04:08.339 PDT: Policy name = mycap, Instance ID = 4
vss_dut1#
*Nov 15 00:04:13.736 PDT: %BUFCAP-6-DISABLE_ASYNC: Capture Point mycap disabled. Reason :
Wireshark Session Ended
vss_dut1#
vss_dut1#dir bootflash:mycap.pcap
Directory of bootflash:/mycap.pcap
72971  -rw-         824  Nov 15 2012 00:04:13 -07:00  mycap.pcap
822910976 bytes total (304648192 bytes free)
```

接続ポイントは VSS アクティブ スイッチに限られているので、VSS スタンバイ スイッチでいずれのコマンドも実行する必要はありません。したがって、手順はスタンドアロン スイッチのシナリオの手順と一致しています。

例 2：表示を伴うファイルのキャプチャと保存（VSS アクティブの接続ポイント）

次に、接続ポイントが VSS アクティブ スイッチだけにある場合に、brief モードにて **display** オプション付きで基本キャプチャおよび保存操作を実行する例を示します。

ステップ 1　VSS アクティブ スイッチで次のコマンドを入力して、キャプチャ セッションを準備します。

```
vss_dut1#monitor capture mycap interface gi 1/1/1 in
vss_dut1#monitor capture mycap match ipv4 any any
```

```
vss_dut1#monitor capture mycap file location bootflash:mycap.pcap
vss_dut1#monitor capture mycap limit packets 60 duration 60
```

ステップ 2　brief モードにて display オプション付きでキャプチャ セッションを開始します。

```
vss_dut1#monitor capture mycap start display

*Nov 14 23:43:20.506 PDT: Policy name = mycap, Instance ID = 3
  0.000000     20.1.1.2 -> 30.1.1.2     TCP [TCP ZeroWindow] 0 > 0 [<None>] Seq=1 Win=0
Len=6
  0.595022     20.1.1.2 -> 30.1.1.2     TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
  1.012008     20.1.1.2 -> 30.1.1.2     TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
  1.500026     20.1.1.2 -> 30.1.1.2     TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
  2.005005     20.1.1.2 -> 30.1.1.2     TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
  2.500026     20.1.1.2 -> 30.1.1.2     TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
  3.000000     20.1.1.2 -> 30.1.1.2     TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
…
```

接続ポイントは VSS アクティブ スイッチに限られているので、VSS スタンバイ スイッチでいずれのコマンドも実行する必要はありません。したがって、手順はスタンドアロン スイッチのシナリオの手順と一致しています。

例 3：ファイルのキャプチャと保存（VSS スタンバイの接続ポイント）

次に、接続ポイントが VSS スタンバイ スイッチにある場合に単純なキャプチャおよび保存操作を行う例を示します。

ステップ 1　VSS アクティブ スイッチで次のコマンドを入力して、キャプチャ セッションを準備します。

```
vss_dut1#monitor capture mycap interface gi 2/1/1 in
vss_dut1#monitor capture mycap match ipv4 any any
vss_dut1#monitor capture mycap file location bootflash:mycap.pcap
vss_dut1#monitor capture mycap limit packets 30 duration 60
```

ステップ 2　スタンバイ スイッチのモジュール番号を確認し、スイッチにログインします（この場合、スタンバイ スイッチに対応するモジュール番号は 14 です）。VSS アクティブ スイッチのように VSS スタンバイ スイッチのキャプチャ セッションを準備し、キャプチャ セッションを開始します。パケットが実際に VSS スタンバイの Wireshark セッションに到達することはないため、これは安全に廃棄できる空の pcap ファイルを VSS スタンバイ ファイル システムに残します。

```
vss_dut1# remote login module 14
Connecting to standby virtual console
Type "exit" or "quit" to end this session
vss_dut1-standby-console#monitor capture mycap interface gi 2/1/1 in
vss_dut1-standby-console#monitor capture mycap match ipv4 any any
vss_dut1-standby-console#monitor capture mycap file location bootflash:mycap.pcap
vss_dut1-standby-console#monitor capture mycap limit packets 30 duration 60
vss_dut1-standby-console#monitor capture mycap start
vss_dut1-standby-console#exit
```

ステップ 3　VSS アクティブ スイッチ上でセッションを開始します。定期的に、パケットがキャプチャ ファイルに保存されていることを確認します。

```
vss_dut1#monitor capture mycap start
…
vss_dut1#dir bootflash:mycap.pcap
Directory of bootflash:/mycap.pcap
72971  -rw-        2424  Nov 14 2012 20:56:34 -07:00  mycap.pcap
822910976 bytes total (304648192 bytes free)
```

ステップ 4　キャプチャ セッションが終了したら、もう不要な場合はキャプチャ ポイントを削除します。

```
Vss_dut1#no monitor capture mycap
```

ステップ 5　再び VSS スタンバイ スイッチにログインします。もう実行されないことを確認するためにキャプチャ セッションを停止し、キャプチャ ポイントを削除して終了します。

```
vss_dut1# remote login mod 14
Connecting to standby virtual console
Type "exit" or "quit" to end this session
vss_dut1-standby-console# monitor capture mycap stop
Capture mycap is not activefl already stopped: so ignore
Unable to deactivate Capture
vss_dut1-standby-console# no monitor capture mycap
vss_dut1-standby-console# exit
```

これらの手順は、接続ポイントが VSS アクティブ スイッチとスタンバイ スイッチの両方にある場合の手順と一致します。

例 4：表示を伴うファイルのキャプチャと保存（VSS スタンバイの接続ポイント）

次に、接続ポイントが VSS スタンバイ スイッチにある場合に display オプション付きで単純なキャプチャおよび保存操作を行う例を示します。

ステップ 1　VSS アクティブ スイッチで次のコマンドを入力して、キャプチャ セッションを準備します。

```
vss_dut1#monitor capture mycap interface gi 2/1/1 in
vss_dut1#monitor capture mycap match ipv4 any any
vss_dut1#monitor capture mycap file location bootflash:mycap.pcap
vss_dut1#monitor capture mycap limit packets 30 duration 60
```

ステップ 2　スタンバイ スイッチのモジュール番号を確認し、スイッチにログインします（この場合、スタンバイ スイッチに対応するモジュール番号は 14 です）。VSS アクティブ スイッチのように VSS スタンバイ スイッチのキャプチャ セッションを準備するが display オプションなしでキャプチャ セッションを開始します。パケットが実際に VSS スタンバイの Wireshark セッションに到達することはないため、これは安全に廃棄できる空の pcap ファイルを VSS スタンバイ ファイル システムに残すことに注意してください。

```
vss_dut1# remote login module 14
Connecting to standby virtual console
Type "exit" or "quit" to end this session
vss_dut1-standby-console# monitor capture mycap interface gi 2/1/1 in
vss_dut1-standby-console# monitor capture mycap match ipv4 any any
vss_dut1-standby-console# monitor capture mycap file location bootflash:mycap.pcap
vss_dut1-standby-console# monitor capture mycap limit packets 30 duration 60
vss_dut1-standby-console# monitor capture mycap start<- Do not use the "display" option
vss_dut1-standby-console# exit
```

ステップ 3　display オプション付きで VSS アクティブ スイッチ上でセッションを開始します。

```
vss_dut1# monitor capture mycap start display
  1   0.000000     30.1.1.2 -> 20.1.1.2     UDP Source port: whois++  Destination port:
whois++
  2   1.000000     30.1.1.2 -> 20.1.1.2     UDP Source port: whois++  Destination port:
whois++
  3   2.000000     30.1.1.2 -> 20.1.1.2     UDP Source port: whois++  Destination port:
whois++
  4   3.000000     30.1.1.2 -> 20.1.1.2     UDP Source port: whois++  Destination port:
whois++
  5   4.000000     30.1.1.2 -> 20.1.1.2     UDP Source port: whois++  Destination port:
whois++
  6   5.000000     30.1.1.2 -> 20.1.1.2     UDP Source port: whois++  Destination port:
whois++
  7   6.000000     30.1.1.2 -> 20.1.1.2     UDP Source port: whois++  Destination port:
whois++
  8   7.000000     30.1.1.2 -> 20.1.1.2     UDP Source port: whois++  Destination port:
whois++
  9   8.000000     30.1.1.2 -> 20.1.1.2     UDP Source port: whois++  Destination port:
whois++
 10   9.000000     30.1.1.2 -> 20.1.1.2     UDP Source port: whois++  Destination port:
whois++
```

ステップ 4　キャプチャ セッションが終了したら、もう不要な場合はキャプチャ ポイントを削除します。

```
Vss_dut1#no monitor capture mycap
```

ステップ 5　再び VSS スタンバイ スイッチにログインします。もう実行されないことを確認するためにキャプチャ セッションを停止し、キャプチャ ポイントを削除して終了します。

```
vss_dut1# remote login mod 14
Connecting to standby virtual console
Type "exit" or "quit" to end this session
vss_dut1-standby-console# monitor capture mycap stop
Capture mycap is not activefl already stopped: so ignore
Unable to deactivate Capture
vss_dut1-standby-console# no monitor capture mycap
vss_dut1-standby-console# exit
```

これらの手順は、接続ポイントが VSS アクティブ スイッチとスタンバイ スイッチの両方にある場合の手順と一致します。

例 5：循環バッファの使用（VSS スタンバイの接続ポイント）

次に、接続ポイントが VSS スタンバイ スイッチにある場合に display オプション付きで単純なキャプチャおよび保存操作を行う例を示します。

ステップ 1　VSS アクティブ スイッチで次のコマンドを入力して、キャプチャ セッションを準備します。

```
vss_dut1#monitor capture mycap interface gi 2/1/1 in
vss_dut1#monitor capture mycap match ipv4 any any
vss_dut1#monitor capture mycap buffer size 1 circular
vss_dut1#monitor capture mycap limit packets 10
```

ステップ 2　スタンバイ スイッチのモジュール番号を確認し、スイッチにログインします（この場合、スタンバイ スイッチに対応するモジュール番号は 14 です）。VSS アクティブ スイッチのように VSS スタンバイ スイッチのキャプチャ セッションを準備し、キャプチャ セッションを開始します。

```
vss_dut1# remote login module 14
Connecting to standby virtual console
```

```
Type "exit" or "quit" to end this session
vss_dut1-standby-console# monitor capture mycap interface GigabitEthernet2/1/1 in
vss_dut1-standby-console# monitor capture mycap match ipv4 any any
vss_dut1-standby-console# monitor capture mycap buffer size 1 circular
vss_dut1-standby-console# monitor capture mycap limit packets 10
vss_dut1-standby-console# monitor capture mycap start
vss_dut1-standby-console# exit
```

ステップ 3　VSS アクティブ スイッチ上でセッションを開始します。定期的に、パケットがキャプチャ ファイルに保存されていることを確認します。

```
vss_dut1# monitor capture mycap start
vss_dut1# show monitor capture mycap buffer
  0.000000     20.1.1.2 -> 30.1.1.2     TCP [TCP ZeroWindow] 0 > 0 [<None>] Seq=1 Win=0
Len=6
  0.497035     20.1.1.2 -> 30.1.1.2     TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
  0.997009     20.1.1.2 -> 30.1.1.2     TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
  1.497035     20.1.1.2 -> 30.1.1.2     TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
  1.997009     20.1.1.2 -> 30.1.1.2     TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
  2.497035     20.1.1.2 -> 30.1.1.2     TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
  2.997009     20.1.1.2 -> 30.1.1.2     TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
  3.507029     20.1.1.2 -> 30.1.1.2     TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
  3.997009     20.1.1.2 -> 30.1.1.2     TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
  4.497035     20.1.1.2 -> 30.1.1.2     TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
vss_dut1# show monitor capture mycap buffer detailed
…
vss_dut1# monitor capture mycap export bootflash:mycap_exp.pcap
…
vss_dut1# monitor capture mycap export bootflash:mycap_exp.pcap
…
vss_dut1# show monitor capture mycap buffer
```

ステップ 4　キャプチャ セッションが終了したら、もう不要な場合はキャプチャ ポイントを削除します。

```
vss_dut1# monitor capture mycap stop
*Nov 15 01:08:58.627 PDT: %BUFCAP-6-DISABLE: Capture Point mycap disabled
vss_dut1# no monitor capture mycap
```

ステップ 5　再び VSS スタンバイ スイッチにログインします。もう実行されないことを確認するためにキャプチャ セッションを停止し、キャプチャ ポイントを削除して終了します。

```
vss_dut1# remote login mod 14
Connecting to standby virtual console
Type "exit" or "quit" to end this session
vss_dut1-standby-console# monitor capture mycap stop
Capture mycap is not activefl already stopped: so ignore
Unable to deactivate Capture
vss_dut1-standby-console# no monitor capture mycap
vss_dut1-standby-console# exit
```

手順は、接続ポイントが VSS アクティブ スイッチとスタンバイ スイッチの両方にある場合の手順と一致します。