

OTV ソリューションを解決するのに Wireshark を使用して下さい

目次

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[問題の説明](#)

[OTV パケットフォーマット](#)

[トポロジ](#)

[パケット キャプチャ](#)

[解決策](#)

[VLAN 100 のデコード パケット](#)

[VLAN 200 のデコード パケット](#)

[OTV ヘッダを削除するのに Editcap を使用して下さい](#)

[Windows プラットフォームの Editcap を実行して下さい](#)

[Mac OS プラットフォームの Editcap を実行して下さい](#)

[結論](#)

概要

この資料は Wireshark の使用、Cisco OTV ソリューションの解決のよく知られている な フリーウェア パケットキャプチャおよび分析ツールを、示したものです。

前提条件

要件

次の項目に関する知識が推奨されます。

- オーバーレイ トランスポート 仮想化 (OTV) (OTV) Nexus シリーズ スイッチで
- マルチプロトコル ラベル スイッチング (MPLS) レイヤ2 仮想 な 私用 Networks (VPN) の 基本
- Wireshark、自由なおよびオープン ソース パケットアナライザ (<https://www.wireshark.org>)

使用するコンポーネント

この 文書に記載されている 情報は Nexus 7000 シリーズ スイッチ プラットフォームに基づいています。

本書の情報は、特定のラボ環境にあるデバイスに基づいて作成されたものです。このドキュメントで使用するすべてのデバイスは、初期 (デフォルト) 設定の状態から起動しています。稼働中

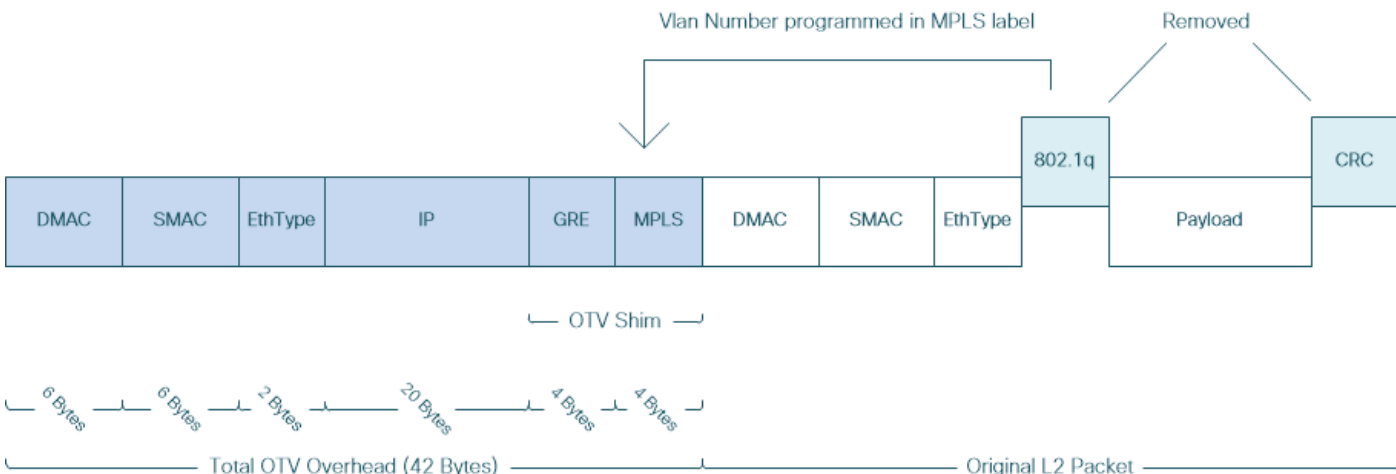
のネットワークで作業を行う場合、コマンドの影響について十分に理解したうえで作業してください。

問題の説明

VPN環境のネットワーク上の問題をトラブルシューティングするとき、手法の1つはカプセル化されたパケットのキャプチャおよび分析を伴います。ただし、Cisco OTV ネットワーク環境でこのアプローチはある特定のチャレンジに会います。広く使われたパケット分析ツールは、Wiresharkのような、自由なおよびオープンソースパケットアナライザ、正しくOTVカプセル化されたトラフィックのコンテンツを解読しないかもしれません。正常にデータ解析を行うために通常OTVパケットからのカプセル化されたデータの抽出のようなそれ故に困難な回避策が、必要となります。

OTVパケットフォーマット

OTVカプセル化は42バイトにつきパケットの全面的なMTUサイズを増加します。これはオリジナルレイヤ2フレームからCRCおよび802.1Qフィールドを削除し、OTVシム(VLANおよびオーバーレイID情報がまた含まれています)および外部IPヘッダを付加するOTVエッジデバイスのオペレーションの結果で。



MPLS L2VPN ソリューションでは、下敷きネットワークのデバイスに十分な情報が正しく MPLS パケット ペイロードをデコードするありません。通常、これは従って下敷きネットワークの MPLS パケットのコンテンツの詳細な分析が必要とならない MPLS コアネットワークのパケット転送がラベルに基づいていた実行されたので、問題ではないです。

ただしこれは OTV パケットのデータ解析が目的を解決することおよび/または監視するために必要となる場合チャレンジを示します。

パケット分析ツールは、Wiresharkのような、ルールを解析する規則的な MPLS パケットの適用によって MPLS ヘッダに続くパケットデータをデコードするように試みます。ただし、それは MPLS L2VPN ヘッドエンドおよびテールエンド ルータの間で普通実行された制御ワードネゴシエーションの結果についての情報がないかもしれないので、パケット分析ツールはデフォルト解析動作に戻って落ち、MPLS ヘッダに続くパケットデータにそれを適用します。

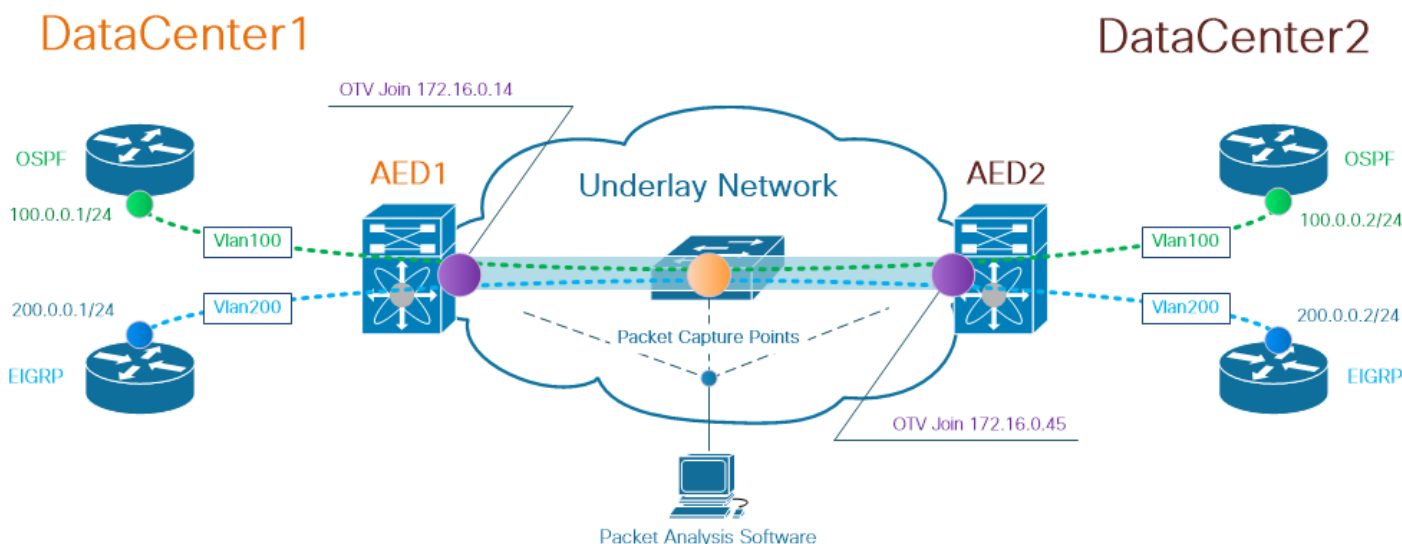
注: MPLS (原子) 上のあらゆる転送するような MPLS L2VPN ソリューションでは、pseudowire エンドポイントは制御ワードパラメータの使用をネゴシエートします。制御

ワードは pseudowire パケットに MPLS ラベルスタックとレイヤ2 ペイロードの間にあるオプションの 4 バイト フィールドです。制御ワードはジェネリックおよびレイヤ2 ペイロード仕様情報を伝えます。C ビットが 1 に設定される場合、アドバタイジング Provider Edge (PE) は信号を送られている pseudowire の各 pseudowire パケットにあると制御ワードが期待します。C ビットが 0 に設定される場合、制御ワードはあると期待されません。

従ってその結果、動作を解析するデフォルト Wireshark は OTV パケットのコンテンツを正しく解読しないかもしれ OTV ネットワークのトラブルシューティング プロセスを複雑にします。

トポロジ

以下は簡単な OTV ネットワークのネットワークダイアグラムです。VLAN 100 および VLAN 200 確立する OSPF および 2 データセンター間の EIGRP 隣接関係のルータ、DataCenter1 および DataCenter2、それぞれ。データセンター相互接続 (DCI) は AED1 としてダイアグラムで表示される N7k スイッチと AED2 間の OTV トンネルと設定されます。



注: Cisco OTV ソリューションは特定のサイトで OTV トラフィックをカプセル化し、カプセル化を解除するネットワークデバイスに割り当てられる保証された エッジ デバイス (AED) ロールの概念を使用します。

頻繁にトンネリング ソリューションで見られるチャレンジはオーバーレイ パケット (IGP、FHRP、等) の特定の種類が下敷きネットワークのある特定のポイントにそれを作るかどうか確かめることです。OSPF および EIGRP オーバーレイトラフィックは一例として使用されます。

パケット キャプチャ

ネットワークでパケットキャプチャを行う複数の方法があります。1つのオプションは Cisco スイッチド ポート アナライザ (SPAN) 機能、利用可能な Catalyst および Cisco Nexus スイッチングプラットフォームを on Cisco 使用することです。

トラブルシューティング プロセスの一部として、複数のポイントのパケットキャプチャは実行された必要がある場合もあります。下敷きネットワークの OTV 加入 インターフェイスおよびインターフェイスは SPAN パケットキャプチャ ポイントとして使用することができます。

解決策

Wireshark デフォルト解析エンジンは MPLS パケット交換網上の MPLS L2VPN で一般的に使用される Pseudowire エミュレーション エッジ ツー エッジ (PWE3) 制御 ワードの一部であるように OTV カプセル化されたオーバーレイ パケットのはじめの幾つかのバイトを誤解するかもしれません。

注: MPLS Pseudowire エミュレーション エッジ ツー エッジ (PWE3) 制御 ワードはようにこの資料の他の制御 ワード参照されます。

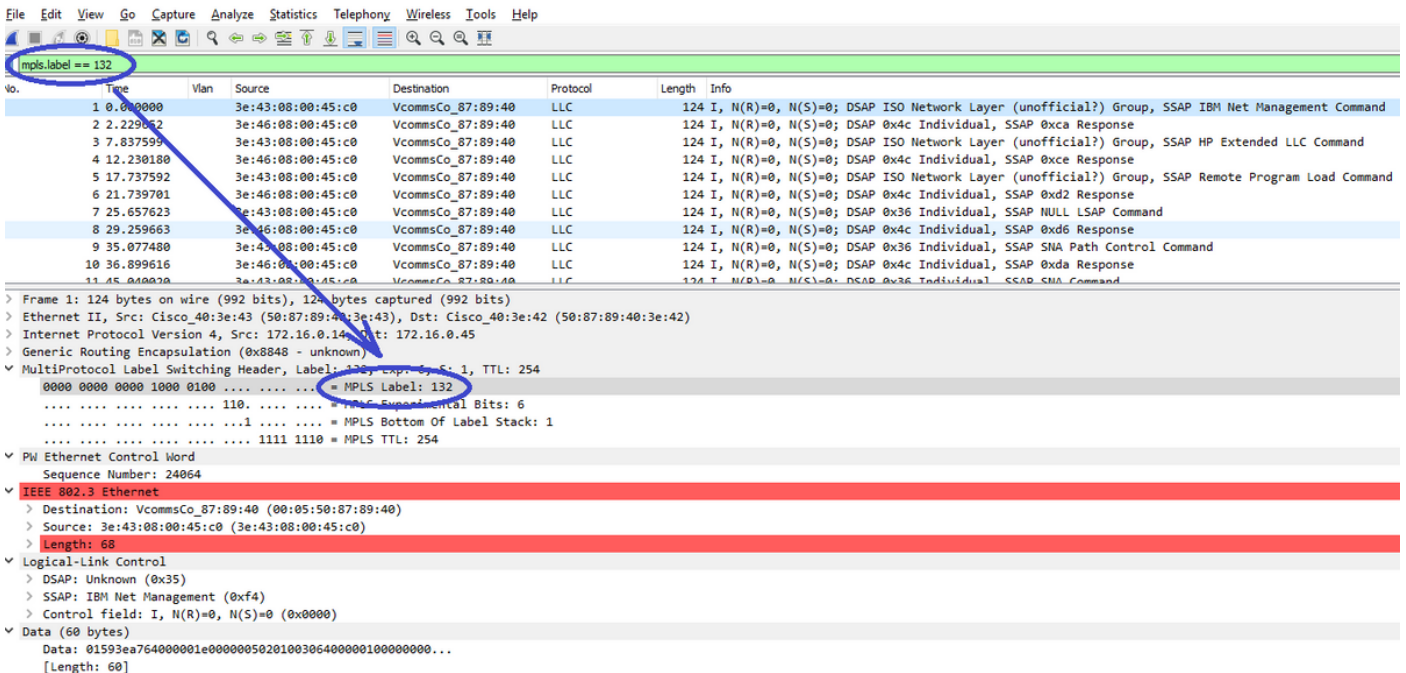
Wireshark パケット 分析ツールを確認するためにパケット デコード プロセスに OTV カプセル化されたパケットのコンテンツを、手動調整です必要正しく解釈します。

注: OTV ヘッダで使用される MPLS ラベルはオーバーレイ VLAN ナンバーに + 32 匹敵します。

VLAN 100 のデコード パケット

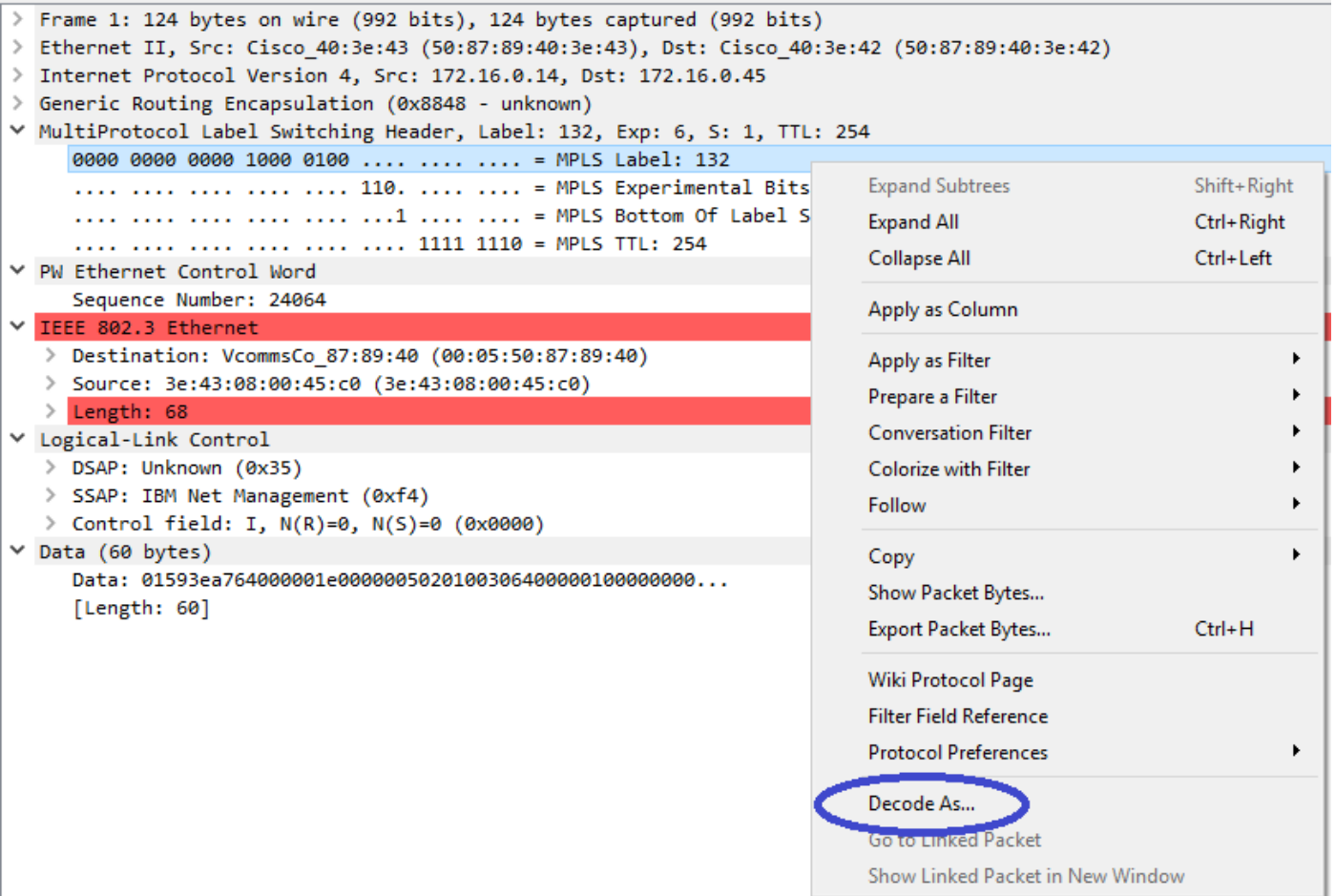
手始めとしてデコード プロセスの、OTV 拡張 VLAN 100 のコンテンツを運ぶ OTV カプセル化されたパケットだけ表示する。使用されるフィルタは VLAN 100 を表す `mpls.label == 132` です。

注: 表示する OTV に伸びた特定の VLAN のためのパケットを使用します次の Wireshark デisplay フィルタを OTV カプセル化しました: `mpls.label == <<vlan 数> + 32` 伸びました



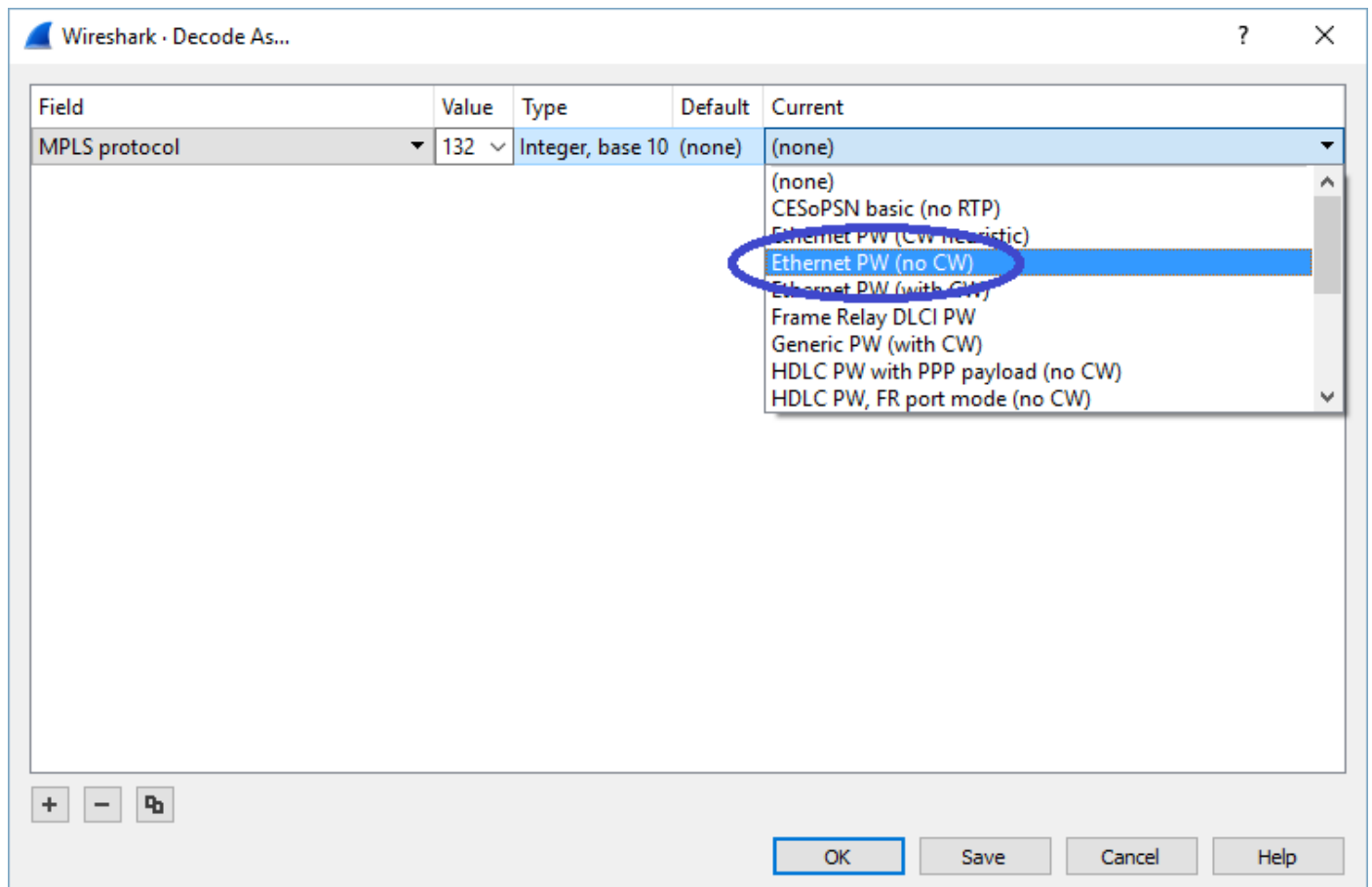
OTV に伸びる VLAN 100 のための OTV カプセル化されたパケットを表示する

デフォルトで Wireshark は制御 ワードとして MPLS L2VPN パケットのコンテンツの最初の 4 バイトを解釈します。これは OTV カプセル化されたパケットを修正される必要があります。これを、パケットの何れかの MPLS ラベル フィールドの右クリックは...オプションとしてするために、デコードを選択し。



MPLS ラベル フィールドを右クリックし、...オプションとしてデコードを選択して下さい

次のステップはカプセル化されたコンテンツに制御ワードがないように Wireshark に言うことです。



オプションを"no CW"を選択する

この変更がボタンを『OK』をクリックすることによって入ったら、Wireshark 分析ツールは OTV カプセル化されたパケットのコンテンツを正しく表示する。

The screenshot shows the Wireshark interface with a filter 'mpls.label == 132'. The packet list table is as follows:

No.	Time	Vlan	Source	Destination	Protocol	Length	Info
1	0.000000		100.0.0.1	224.0.0.5	OSPF	124	Hello Packet
2	2.229652		100.0.0.2	224.0.0.5	OSPF	124	Hello Packet
3	7.837599		100.0.0.1	224.0.0.5	OSPF	124	Hello Packet
4	12.230180		100.0.0.2	224.0.0.5	OSPF	124	Hello Packet
5	17.737592		100.0.0.1	224.0.0.5	OSPF	124	Hello Packet
6	21.739701		100.0.0.2	224.0.0.5	OSPF	124	Hello Packet
7	25.657623		100.0.0.1	224.0.0.5	OSPF	124	Hello Packet
8	29.259663		100.0.0.2	224.0.0.5	OSPF	124	Hello Packet
9	35.077480		100.0.0.1	224.0.0.5	OSPF	124	Hello Packet
10	36.899616		100.0.0.2	224.0.0.5	OSPF	124	Hello Packet
11	45.040020		100.0.0.1	224.0.0.5	OSPF	124	Hello Packet

The packet details pane shows the following structure for the selected packet:

- > Frame 1: 124 bytes on wire (992 bits), 124 bytes captured (992 bits)
- > Ethernet II, Src: Cisco_40:3e:43 (50:87:89:40:3e:43), Dst: Cisco_40:3e:42 (50:87:89:40:3e:42)
- > Internet Protocol Version 4, Src: 172.16.0.14, Dst: 172.16.0.45
- > Generic Routing Encapsulation (0x8848 - unknown)
- ✓ MultiProtocol Label Switching Header, Label: 132, Exp: 6, S: 1, TTL: 254
 - 0000 0000 0000 1000 0100 = MPLS Label: 132
 - 110. = MPLS Experimental Bits: 6
 - 1 = MPLS Bottom Of Label Stack: 1
 - 1111 1110 = MPLS TTL: 254
- > Ethernet II, Src: Cisco_40:3e:43 (50:87:89:40:3e:43), Dst: IPv4mcast_05 (01:00:5e:00:00:05)
- > Internet Protocol Version 4, Src: 100.0.0.1, Dst: 224.0.0.5
- ✓ Open Shortest Path First
 - > OSPF Header
 - > OSPF Hello Packet

OTV カプセル化されたパケットの正しい Wireshark ディスプレイ コンテンツ

VLAN 200 のデコード パケット

ステップの上で OTV に伸びるあらゆる VLAN に適当であって下さい。たとえば、VLAN 200 のパケットだけ表示する Wireshark フィルタを使用して分析ツールで出力するために次を得ます。

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

mpls.label == 232

No.	Time	Vlan	Source	Destination	Protocol	Length	Info
1	0.000000		3e:46:08:00:45:c0	Remotek_87:89:40	LLC	116	I, N(R)=0, N(S)=0; DSAP 0x3e Group, SSAP 0xae Command
2	2.346992		3e:43:08:00:45:c0	Remotek_87:89:40	LLC	116	I, N(R)=0, N(S)=0; DSAP 0x3c Group, SSAP 0x70 Command
3	4.603176		3e:46:08:00:45:c0	Remotek_87:89:40	LLC	116	I, N(R)=0, N(S)=0; DSAP 0x3e Group, SSAP 0xae Response
4	6.981213		3e:43:08:00:45:c0	Remotek_87:89:40	LLC	116	I, N(R)=0, N(S)=0; DSAP 0x3c Group, SSAP 0x70 Response
5	9.373389		3e:46:08:00:45:c0	Remotek_87:89:40	LLC	116	I, N(R)=0, N(S)=0; DSAP 0x3e Group, SSAP 0xb0 Command
6	11.330387		3e:43:08:00:45:c0	Remotek_87:89:40	LLC	116	I, N(R)=0, N(S)=0; DSAP 0x3c Group, SSAP 0x72 Command
7	13.715773		3e:46:08:00:45:c0	Remotek_87:89:40	LLC	116	I, N(R)=0, N(S)=0; DSAP 0x3e Group, SSAP 0xb0 Response
8	16.102792		3e:43:08:00:45:c0	Remotek_87:89:40	LLC	116	I, N(R)=0, N(S)=0; DSAP 0x3c Group, SSAP 0x72 Response
9	18.185963		3e:46:08:00:45:c0	Remotek_87:89:40	LLC	116	I, N(R)=0, N(S)=0; DSAP 0x3e Group, SSAP 0xb2 Command
10	20.554788		3e:43:08:00:45:c0	Remotek_87:89:40	LLC	116	I, N(R)=0, N(S)=0; DSAP 0x3c Group, SSAP 0x74 Command
11	23.051203		3e:46:08:00:45:c0	Remotek_87:89:40	LLC	116	I, N(R)=0, N(S)=0; DSAP 0x3e Group, SSAP 0xb2 Response

> Frame 1: 116 bytes on wire (928 bits), 116 bytes captured (928 bits)

> Ethernet II, Src: Cisco_40:3e:46 (50:87:89:40:3e:46), Dst: Cisco_40:3e:42 (50:87:89:40:3e:42)

> Internet Protocol Version 4, Src: 172.16.0.45, Dst: 172.16.0.14

> Generic Routing Encapsulation (0x8848 - unknown)

MultiProtocol Label Switching Header, Label: 232, Exp: 6, S: 1, TTL: 254

0000 0000 0000 1110 1000 = MPLS Label: 232

.... 110. = MPLS Experimental Bits: 6

.... 1 = MPLS Bottom Of Label Stack: 1

.... 1111 1110 = MPLS TTL: 254

PW Ethernet Control Word

Sequence Number: 24064

IEEE 802.3 Ethernet

> Destination: Remotek_87:89:40 (00:0a:50:87:89:40)

> Source: 3e:46:08:00:45:c0 (3e:46:08:00:45:c0)

> Length: 60

Logical-Link Control

> DSAP: Unknown (0x3f)

> SSAP: Unknown (0xae)

> Control field: I, N(R)=0, N(S)=0 (0x0000)

Data (52 bytes)

Data: 0158d0efc8000002e000000a0205f208000000000000000...

[Length: 52]

OTV に伸びる VLAN 200 のためのパケットを表示する

Wireshark が PW 制御ワードとして MPLS パケットのはじめの幾つかのバイトを解読しないように指示されたらプロセスを正常に完了できませんデコードして下さい。

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

mpls.label == 232

No.	Time	Vlan	Source	Destination	Protocol	Length	Info
1	0.000000		200.0.0.2	224.0.0.10	EIGRP	116	Hello
2	2.346992		200.0.0.1	224.0.0.10	EIGRP	116	Hello
3	4.603176		200.0.0.2	224.0.0.10	EIGRP	116	Hello
4	6.981213		200.0.0.1	224.0.0.10	EIGRP	116	Hello
5	9.373389		200.0.0.2	224.0.0.10	EIGRP	116	Hello
6	11.330387		200.0.0.1	224.0.0.10	EIGRP	116	Hello
7	13.715773		200.0.0.2	224.0.0.10	EIGRP	116	Hello
8	16.102792		200.0.0.1	224.0.0.10	EIGRP	116	Hello
9	18.185963		200.0.0.2	224.0.0.10	EIGRP	116	Hello
10	20.554788		200.0.0.1	224.0.0.10	EIGRP	116	Hello
11	23.051203		200.0.0.2	224.0.0.10	EIGRP	116	Hello

> Frame 1: 116 bytes on wire (928 bits), 116 bytes captured (928 bits)

> Ethernet II, Src: Cisco_40:3e:46 (50:87:89:40:3e:46), Dst: Cisco_40:3e:42 (50:87:89:40:3e:42)

> Internet Protocol Version 4, Src: 172.16.0.45, Dst: 172.16.0.14

> Generic Routing Encapsulation (0x8848 - unknown)

MultiProtocol Label Switching Header, Label: 232, Exp: 6, S: 1, TTL: 254

0000 0000 0000 1110 1000 = MPLS Label: 232

.... 110. = MPLS Experimental Bits: 6

.... 1 = MPLS Bottom Of Label Stack: 1

.... 1111 1110 = MPLS TTL: 254

Ethernet II, Src: Cisco_40:3e:46 (50:87:89:40:3e:46), Dst: IPv4mcast_0a (01:00:5e:00:00:0a)

Internet Protocol Version 4, Src: 200.0.0.2, Dst: 224.0.0.10

Cisco EIGRP

Wireshark は EIGRP パケットとして正しく VLAN 200 トラフィックを表示する

OTV ヘッダを削除するのに Editcap を使用して下さい

通常、Wireshark インストールはコマンド・ライン パケット編集ツールによって呼出される *Editcap* が付いています。このツールはキャプチャされるパケットから OTV オーバーヘッドを永久削除にすることができます。これは手動で Wireshark の解析動作を調節する必要なしで Wireshark グラフィカル ユーザ インターフェイス (GUI) のキャプチャされるパケットの容易なディスプレイおよび分析を、可能にします。

Windows プラットフォームの Editcap を実行して下さい

ON ウィンドウ オペレーティング システムは c:\Program Files\Wireshark > ディレクトリに、*editcap.exe* デフォルトでインストールされています。

実行して下さいこのツールをと - OTV オーバーヘッドを取除き、.pcap ファイルの結果を保存する C フラグ。

```
c:\Users\cisco\Desktop> "c:\Program Files\Wireshark\editcap.exe" -C 42 otv-underlay-capture.pcap  
otv-underlay-capture-no-header.pcap  
c:\Users\cisco\Desktop>
```

Mac OS プラットフォームの Editcap を実行して下さい

Mac OS オペレーティング システムで、editcap は /usr/local/bin フォルダで利用できます。

```
CISCO:cisco$ /usr/local/bin/editcap -C 42 otv-underlay-capture.pcap otv-underlay-capture-no-  
header.pcap  
CISCO:cisco$
```

キャプチャされるパケット with Editcap tool から OTV ヘッダを削除することによって、1 つは次々と OTV シムの部品である、MPLS ヘッダの一部として符号化される VLAN 情報を失います。「Editcap tool と OTV ヘッダを削除する前に OTV> + 32> Wireshark GUI フィルタ拡張される mpls.label == <<vlan 数を使用するために忘れないようにして下さい特定の VLAN のだけトラフィックの分析が必要となる場合。

結論

Cisco OTV ソリューションを解決することはコントロールプレーン オペレーションおよびデータ平らなカプセル化観点からの両方テクノロジーのよい理解を、必要とします。実際にはナレッジを適用して、Wireshark のようなフリーウェア パケット 分析ツールは OTV パケット 分析の非常に強力証明できます。さまざまなパケット デ스플레이 オプションに加えて、Wireshark 典型的なインストールはパケット 分析を簡素化できるパケット編集ツールを提供します。これは特定のトラブルシューティング セッションに最も関連しているパケットコンテンツの方に焦点を合わせるために解決することを割り当てます。