StarOS の L2TP - ASR5k への実装と L2TP のピアリングのトラブルシューティング - L2TPTunnelDownPeerUnreachable

目次

概要

このドキュメントでは、ASR5k で StarOS の Layer 2 Tunneling Protocol（L2TP）を実装する方法、および L2TP ピアリングの L2TPTunnelDownPeerUnreachable をトラブルシュートする方法について説明します。

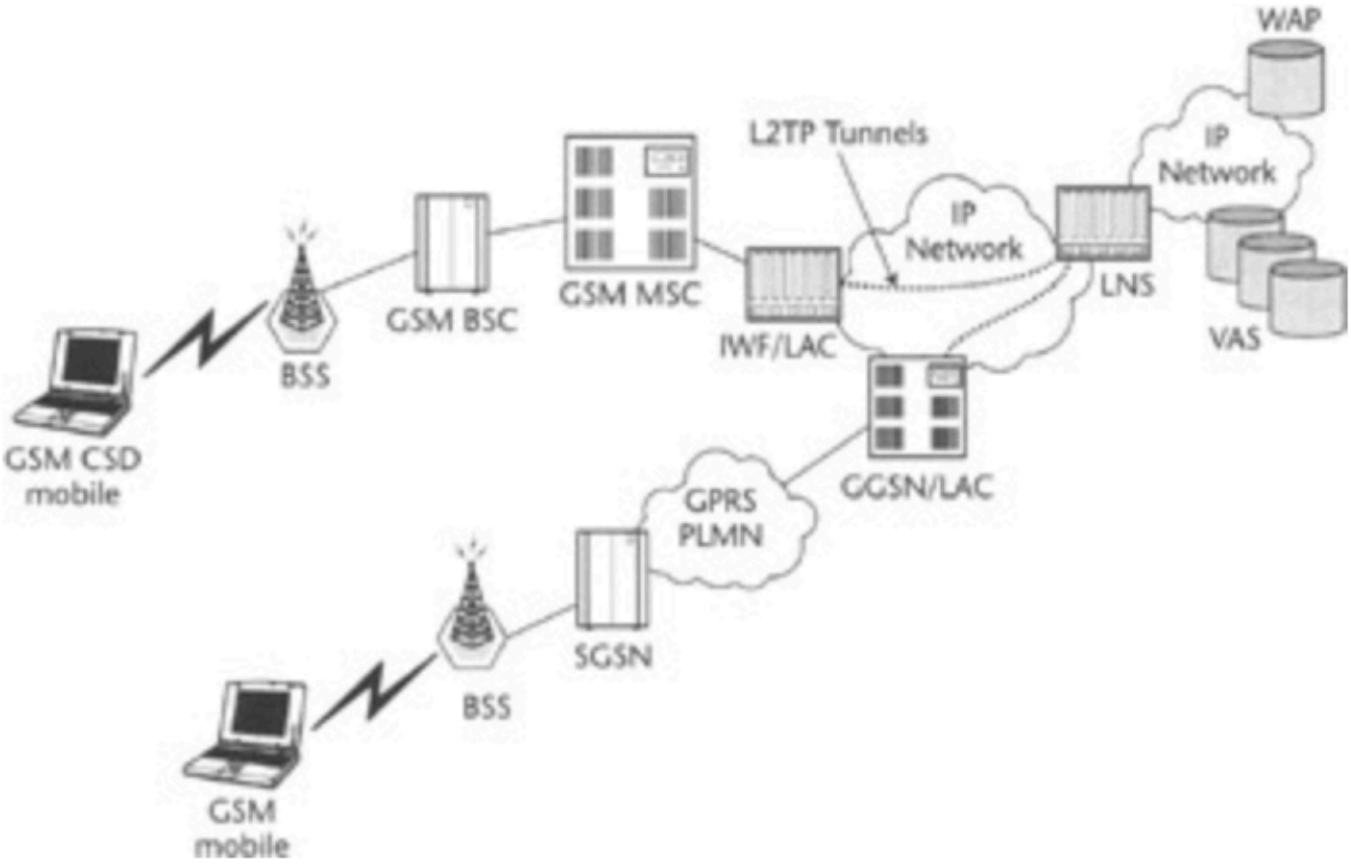
L2TP の概要

L2TP は、PPP のポイントツーポイントの特性を拡張するものです。 L2TP ではトンネル化 PPP フレームを送信するためのカプセル化方式が提供され、これにより PPP エンドポイント間をパケット スイッチド ネットワーク上でトンネル化できます。 L2TP は、インターネットを使用してイントラネット型のサービスを提供するリモート アクセス型のシナリオでは最も一般的に使用されます。 この概念は Virtual Private Network（VPN; バーチャル プライベート ネットワーク）のものです。

L2TP の 2 つの主な物理的要素として、L2TP Access Concentrator（LAC; L2TP アクセス コンセントレータ）と L2TP Network Server（LNS; L2TP ネットワーク サーバ）があります。

- LAC： LAC は LNS に対するピアであり、トンネルのエンドポイントの片側として動作します。 LAC はリモート PPP 接続を終端し、リモートと LNS の中間に位置します。 パケットは PPP 接続を経由してリモート接続との間で転送されます。 LNS との間でやりとりされるパケットは、L2TP トンネルを経由して転送されます。
- LNS： LNS は LAC に対するピアであり、トンネルのエンドポイントの片側として動作しま

す。 LNS は LAC PPP トンネル化セッションの終端ポイントです。 これは複数の LAC トンネル化 PPP セッションを集約し、プライベート ネットワークに入るために使用されます。 モバイル ネットワークでの L2TP のセットアップが単純化されました（次の図を参照）。



L2TP では、次の 2 種類のメッセージ タイプが使用されます。

- コントロール メッセージ： L2TP では、コントロール メッセージとデータ メッセージを別々のコントロール チャネルおよびデータ チャネルを使用して受け渡しします。 インバンド コントロール チャネルでは、順序に則したコントロール接続管理メッセージ、コール管理メッセージ、エラー レポート メッセージ、セッション コントロール メッセージが渡されます。 コントロール接続の開始は LAC や LNS に特有のものではなく、コントロール接続の確立に関連するトンネルの発信元または受信側に特有のものです。 トンネルのエンドポイント間では、共有秘密鍵のチャレンジ認証方式が使用されます。
- データ メッセージ： データ メッセージは、L2TP トンネルに送出される PPP フレームのカプセル化に使用されます。

詳細なコール フローとトンネル確立については、以下を参照してください。

http://www.cisco.com/c/en/us/support/docs/dial-access/virtual-private-dialup-network-vpdn/23980-l2tp-23980.html

モビリティでの活用場所

一般的には社内ユーザ向けに導入され、GGSN が LAC として機能し、社内ネットワークで稼働する LNS への安全なトンネルを確立します。 ソフトウェア バージョン別の GGSN 構成ガイドの付録で詳細なコール フローを確認できます。ガイドは次の場所に用意されています。

http://www.cisco.com/c/en/us/support/wireless/asr-5000-series/products-installation-and-configuration-guides-list.html

このセットアップにおける ASR5x00

ASR5k は LAC と LNS の機能に対応できます。

L2TP LAC のサポート

L2TP は、L2TP セッションとしてサブスクライバ PPP 接続をトンネリングする前に、LAC と LNS 間に L2TP コントロール トンネルを確立します。 LAC サービスは、GGSN と同じアーキテクチャに基づいており、動的なリソース割り当て、分散メッセージ、データ処理などを活用します。 この設計により LAC サービスは、1 秒当たり 4000 超のセットアップや最大 3G を超えるスループットに対応できます。 1 つのトンネルで最大 65535 セッションに対応可能で、システム単位では 32,000 個のトンネルを使用して 500,000 もの L2TP セッションを扱うことができます。

L2TP LNS のサポート

Layer 2 Tunneling Protocol ネットワーク サーバ（LNS）として設定されているシステムは、L2TP アクセス コンセントレータ（LAC）からのセキュアなバーチャル プライベート ネットワーク（VPN）トンネルを終端させる機能を持ちます。

L2TP は、L2TP セッションとしてサブスクライバ PPP 接続をトンネリングする前に、LAC と LNS 間に L2TP コントロール トンネルを確立します。 1 つのトンネルで最大 65535 セッションに対応可能で、LNS 単位では最大 500,000 セッションを扱うことができます。

LNS のアーキテクチャは GGSN に類似していて、デマルチプレクサの概念を活用して、新しい L2TP セッションを、プラットフォーム上の使用可能なソフトウェア リソースおよびハードウェア リソースにオペレータの介入なしでインテリジェントに割り当てます。

詳細については、PGW/GGSN 構成ガイドを参照してください。

ASR5k 上のシスコ デバイスでサービスを有効にするための設定

ASR5k での LAC の設定例

```
 apn test-apn
accounting-mode none
     aaa group AAA
     authentication msisdn-auth
     ip context-name destination
     tunnel l2tp peer-address 1.1.1.1 local-hostname lac_l2tp     configure
context destination-gi
lac-service l2tp_service
     allow called-number value apn
     peer-lns 1.1.1.1 encrypted secret pass
     bind address 1.1.1.2
```

ASR5k での LNS の設定例

```
  configure
context destination-gi
lns-service lns-svc
bind address 1.1.1.1
authentication { { [ allow-noauth | chap < pref > | mschap < pref > | | pap < pref > | msid-auth
}
```

> **注:** 同じ IP インターフェイス上の複数のアドレスを、別々の LNS サービスにバインドできます。 ただし、各アドレスは 1 つの LNS サービスにしかバインドできません。 さらに、LNS サービスは、LAC サービスなどの他のサービスと同じインターフェイスにはバインドできません。

Cisco IOS デバイスでの LNS の設定例

これは Cisco IOS の設定例として使用できますが、この記事の説明範囲ではありません。

LNS の設定

```
 aaa group server radius AAA
server 2.2.2.2 auth-port 1812 acct-port 1813
ip radius source-interface GigabitEthernet0/1
!  aaa authentication login default local
aaa authentication ppp AAA group AAA
aaa authorization network AAA group AAA
aaa accounting network default
action-type start-stop
group radius   vpdn-group vpdn
accept-dialin
protocol l2tp
virtual-template 10
l2tp tunnel password pass   interface Virtual-Template10
ip unnumbered GigabitEthernet0/1
peer default ip address pool AAA
ppp authentication pap chap AAA
ppp authorization AAA
```

ピア到達不能イベントのトラブルシューティング

このセクションでは、ネットワーク内の L2TPTunnelDownPeerUnreachable イベントをトラブルシュートする方法のガイドラインを示します。 ここでは PDSN のクローズした RP に関して説明しますが、トラブルシューティング手順は、GGSN/PGW でのトラブルシューティングと同じです。

注意点として、LAC から LNS までのトンネルは、サブスクライバ セッションを含める目的で作成される一方、サブスクライバの接続を PDSN/HA/GGSN/PGW から LNS にまで拡張します。 LNS が終端で、IP アドレスが提供されます。 StarOS シャーシ上の LNS は、設定されている IP プールから IP アドレスを取得します。 その他の一部の LNS 上（顧客宅内など）では、IP アドレスはそこにある LNS によって提供されます。 後者のシナリオでは、ユーザは口ーミング パートナー上で動作する LAC 経由でホーム ネットワークに接続できるため、効率的です。

最初のサブスクライバ セッションのセットアップが試行されるときに、LAC LNS トンネルは初めて作成され、トンネル内にセッションが存在する限り機能し続けます。

所定のトンネルで最後のセッションが終了すると、そのトンネルはクローズまたはシャットダウンします。 同じ LAC-LNS ピア間に複数のトンネルを確立できます。

以下は **show l2tp tunnels all** コマンドの出力の一部です。この例では、シャーシが LAC と LNS の両方のサービス (TestLAC と TestLNS) をホストしています。 LAC と LNS のすべてのトンネルにセッションがあるのに対し、クローズした RP トンネルの一部にはセッションがないことに注目してください。

```
 [local]1X-PDSN# show l2tp tunnels all | more
|+----State: (C) - Connected       (c) - Connecting
|            (d) - Disconnecting   (u) - Unknown
|
|
v  LocTun ID  PeerTun ID Active Sess Peer IPAddress  Service Name   Uptime
--- ---------- ---------- ----------- --------------- ------------   ---------
.............
C  30         1          511         214.97.107.28   TestLNS         00603h50m
C  31         56         468         214.97.107.28   TestLNS         00589h31m
C  10         105        81          79.116.237.27   TestLAC         00283h53m
C  29         16         453         79.116.231.27   TestLAC         00521h32m
C  106        218        63          79.116.231.27   TestLAC         00330h10m
C  107        6          464         79.116.237.27   TestLAC         00329h47m
C  30         35         194         214.97.107.28   TestLNS         00596h06m
```

サービス設定を確認するには、次のコマンドを発行します。

```
  show (lac-service | lns-service) name <lac or lns service name>
```

以下は、LAC サービス 1.1.1.2 および LNS サービス (ピア) 1.1.1.1 での L2TPTunnelDownPeerUnreachable トラップの例です。

```
 Internal trap notification 92 (L2TPTunnelDownPeerUnreachable)  context destination service lac
peer address 1.1.1.1 local address 1.1.1.2
```

show snmp trap statistics コマンドを使用して、このトラップが (統計のリロードまたは最終リセット以降に) トリガーされた回数を取得します。

トンネル セットアップがタイムアウトになるか、またはキープアライブ (Hello) パケットが応答しないときに、L2TP に対して L2TPTunnelDownPeerUnreachable トラップがトリガーされます。 これは通常、LNS ピアが LAC からの要求に応答していないこと、またはいずれかの方向で転送の問題が発生していることに起因します。

ピアが到達可能になることを示すトラップは存在しません。そのため、それ以上の調査方法が不明な場合は、調査時に問題がまだ解消されていないかどうかについて明らかにならないことがあります。

続行するために最も重要な要素は、ピアの IP アドレスです。 最初の手順として、PING でチェックできる IP 接続があることを確認します。 接続がある場合は、デバッグを進めることができます。

```
****THIS IS TO BE RUN CAREFULLY and UPON verification of TAC/BU****


Active logging (exec mode) – logs written to terminal window

logging filter active facility l2tpmgr level debug
logging filter active facility l2tp-control level debug
logging active

To stop logging:
```

```
no logging active

Runtime logging (global config mode) – logs saved internally

logging filter runtime facility l2tpmgr level debug
logging filter runtime facility l2tp-control level debug

To view logs:

show logs (and/or check the syslog server if configured)
```

注：

l2tpmgr は、特定のサブスクライバ セッションのセットアップをトラックします。

l2tp-control は、トンネルの確立をトラックします。

以下はこの出力からのデバッグの例です。

使用例： 再試行タイムアウトが原因のトンネルの初期セットアップ失敗

```
16:34:00.017 [l2tpmgr 48140 debug] [7/0/555 <l2tpmgr:1> l2tpmgr_call.c:591] [callid 4144ade2]
[context: destination, contextID: 3]  [software internal system] L2TPMgr-1  msid 0000012345
username laclnsuser  service <lac> - IPSEC tunnel does not exist
16:34:00.018 [l2tp-control 50069 debug] [7/0/555 <l2tpmgr:1> l2tpsnx_fsm.c:105] [callid
4144ade2] [context: destination, contextID: 3]  [software internal user] l2tp fsm: state
L2TPSNX_STATE_OPEN event L2TPSNX_EVNT_APP_NEW_SESSION ---------------------
16:34:00.018 [l2tp-control 50001 debug] [7/0/555 <l2tpmgr:1> l2tpsnx_proto.c:1474] [callid
4144ade2] [context: destination, contextID: 3]  [software internal user outbound protocol-log]
L2TP Tx PDU, from 1.1.1.2:13660 to 1.1.1.1:1701 (138)
l2tp:[TLS](0/0)Ns=0,Nr=0 *MSGTYPE(SCCRQ) *PROTO_VER(1.0) *FRAMING_CAP(AS) *BEARER_CAP(AD)
TIE_BREAKER(0706050403020100) FIRM_VER(256) *HOST_NAME(lac) VENDOR_NAME(StarentNetworks)
*ASSND_TUN_ID(10) *RECV_WIN_SIZE(16) *CHALLENGE(dbed79cdc497f266bd374d427607cd52)
16:34:00.928 [l2tp-control 50001 debug] [7/0/555 <l2tpmgr:1> l2tpsnx_proto.c:1474] [callid
4144ade2] [context: destination, contextID: 3]  [software internal user outbound protocol-log]
L2TP Tx PDU, from 1.1.1.2:13660 to 1.1.1.1:1701 (138)
l2tp:[TLS](0/0)Ns=0,Nr=0 *MSGTYPE(SCCRQ) *PROTO_VER(1.0) *FRAMING_CAP(AS) *BEARER_CAP(AD)
TIE_BREAKER(0706050403020100) FIRM_VER(256) *HOST_NAME(lac) VENDOR_NAME(StarentNetworks)
*ASSND_TUN_ID(10) *RECV_WIN_SIZE(16) *CHALLENGE(dbed79cdc497f266bd374d427607cd52)
16:34:02.943 [l2tp-control 50001 debug] [7/0/555 <l2tpmgr:1> l2tpsnx_proto.c:1474] [callid
4144ade2] [context: destination, contextID: 3]  [software internal user outbound protocol-log]
L2TP Tx PDU, from 1.1.1.2:13660 to 1.1.1.1:1701 (138)
l2tp:[TLS](0/0)Ns=0,Nr=0 *MSGTYPE(SCCRQ) *PROTO_VER(1.0) *FRAMING_CAP(AS) *BEARER_CAP(AD)
TIE_BREAKER(0706050403020100) FIRM_VER(256) *HOST_NAME(lac) VENDOR_NAME(StarentNetworks)
*ASSND_TUN_ID(10) *RECV_WIN_SIZE(16) *CHALLENGE(dbed79cdc497f266bd374d427607cd52)
16:34:06.870 [l2tp-control 50001 debug] [7/0/555 <l2tpmgr:1> l2tpsnx_proto.c:1474] [callid
4144ade2] [context: destination, contextID: 3]  [software internal user outbound protocol-log]
L2TP Tx PDU, from 1.1.1.2:13660 to 1.1.1.1:1701 (138)
l2tp:[TLS](0/0)Ns=0,Nr=0 *MSGTYPE(SCCRQ) *PROTO_VER(1.0) *FRAMING_CAP(AS) *BEARER_CAP(AD)
TIE_BREAKER(0706050403020100) FIRM_VER(256) *HOST_NAME(lac) VENDOR_NAME(StarentNetworks)
*ASSND_TUN_ID(10) *RECV_WIN_SIZE(16) *CHALLENGE(dbed79cdc497f266bd374d427607cd52)
16:34:14.922 [l2tp-control 50001 debug] [7/0/555 <l2tpmgr:1> l2tpsnx_proto.c:1474] [callid
4144ade2] [context: destination, contextID: 3]  [software internal user outbound protocol-log]
L2TP Tx PDU, from 1.1.1.2:13660 to 1.1.1.1:1701 (138)
l2tp:[TLS](0/0)Ns=0,Nr=0 *MSGTYPE(SCCRQ) *PROTO_VER(1.0) *FRAMING_CAP(AS) *BEARER_CAP(AD)
TIE_BREAKER(0706050403020100) FIRM_VER(256) *HOST_NAME(lac) VENDOR_NAME(StarentNetworks)
*ASSND_TUN_ID(10) *RECV_WIN_SIZE(16) *CHALLENGE(dbed79cdc497f266bd374d427607cd52)
------------------  16:34:22.879 [l2tp-control 50001 debug] [7/0/555 <l2tpmgr:1>
l2tpsnx_proto.c:1474] [callid 4144ade2] [context: destination, contextID: 3]  [software internal
user outbound protocol-log] L2TP Tx PDU, from 1.1.1.2:13660 to 1.1.1.1:1701 (38)
l2tp:[TLS](0/0)Ns=1,Nr=0 *MSGTYPE(StopCCN) *RESULT_CODE(2/0) *ASSND_TUN_ID(10)
```

```
16:34:22.879 [l2tp-control 50069 debug] [7/0/555 <l2tpmgr:1> l2tpsnx_fsm.c:105] [callid
4144ade2] [context: destination, contextID: 3]  [software internal user] l2tp fsm: state
L2TPSNX_STATE_WAIT_TUNNEL_ESTB event L2TPSNX_EVNT_PROTO_TUNNEL_DISCONNECTED
```

以下は、システムが失敗と判別したときに上記のログと照合するためにトリガーされた結果の SNMP トラップです。

```
 16:34:22 2009 Internal trap notification 92 (L2TPTunnelDownPeerUnreachable)  context
destination service lac peer address 1.1.1.1 local address 1.1.1.2
```

使用例： 再試行タイムアウトが原因のトンネルの初期セットアップ失敗：分析

トンネルが 16:34 に稼働状態になり、チャレンジの送信を 5 回試行していることを確認できます。 応答がなく、最終的にトンネルが切断されているようです。

デフォルト設定または設定値を確認すると、次のようになっています。

```
       max-retransmission 5
      retransmission-timeout-first 1
      retransmission-timeout-max 8
```

この設定は、最初の再送信を 1 秒後に行い、その後の間隔は指数関数的に毎回 2 倍に延ばした秒数、つまり 1、2、4、8、8 になると解釈されます。

max-retransmissions 5 には、最初の試行または伝送が含まれます。
retransmission-timeout-max は、この上限に達した後の伝送間隔の最大時間です。
retransmission-timeout-first は、最初の再送信が実行されるまでに待機する時間の開始点です。

したがってこれらを計算すると、デフォルト パラメータの場合は、1 + 2 + 4 + 8 + 8 秒 = 23 秒後に失敗となっており、下記の出力と正確に一致します。

使用例： キープアライブが原因のトンネルの初期セットアップ失敗

L2TPTunnelDownPeerUnreachable トラップが発生するもう 1 つの理由は、keepalive-interval メッセージに対する応答がないことです。 これらのメッセージは、トンネルを経由して送信されるコントロール メッセージやコントロール データが存在しない期間中に、トンネルのもう一端が引き続き有効であることを確認するために使用されます。 トンネルにセッションが存在するが、何も実行されていない場合は、このコマンドによって、トンネルが引き続き正常に機能しているかどうかが確認されます。トンネルを有効にすると、パケット交換がない状態が設定期間（つまり、60 秒間）を超えたらキープアライブ メッセージが送信され、応答が返ってくるはずです。 最初のキープアライブを送信して応答がなかった場合、次のキープアライブを送信するまでの間隔は、上記で説明したトンネル セットアップの場合と同じです。 そのため、hello（キープアライブ）メッセージへの応答を受信しない状態が 23 秒間続くと、トンネルが切断されます。 設定可能なキープアライブ間隔（デフォルト = 60 秒）を確認してください。

以下は、monitor Subscriber と logging の両方からキープアライブが正常に交換されている例です。 ユーザ データが 1 分間送信されていないので、メッセージ セットの送信間隔が 1 分になっていることに注目してください。 この例では、LAC サービスと LNS サービスが同じシャーシ内にあり、それぞれ **destination** および **lns** という名前のコンテキストに含まれています。

```
 INBOUND>>>>>  12:54:35:660 Eventid:50000(3)
L2TP Rx PDU, from 1.1.1.1:13660 to 1.1.1.2:13661 (20)
l2tp:[TLS](5/0)Ns=19,Nr=23 *MSGTYPE(HELLO)

<<<<OUTBOUND  12:54:35:661 Eventid:50001(3)
L2TP Tx PDU, from 1.1.1.2:13661 to 1.1.1.1:13660 (12)
l2tp:[TLS](1/0)Ns=23,Nr=20 ZLB
```

```
<<<<OUTBOUND  12:55:35:617 Eventid:50001(3)
L2TP Tx PDU, from 1.1.1.2:13661 to 1.1.1.1:13660 (20)
l2tp:[TLS](1/0)Ns=23,Nr=20 *MSGTYPE(HELLO)

INBOUND>>>>>  12:55:35:618 Eventid:50000(3)
L2TP Rx PDU, from 1.1.1.1:13660 to 1.1.1.2:13661 (12)
l2tp:[TLS](5/0)Ns=20,Nr=24 ZLB  12:54:35.660 [l2tp-control 50001 debug] [7/0/555 <l2tpmgr:1>
l2tpsnx_proto.c:1474] [callid 106478e8] [context: lns, contextID: 11]  [software internal user
outbound protocol-log] L2TP Tx PDU, from 1.1.1.1:13660 to 1.1.1.2:13661 (20)
l2tp:[TLS](5/0)Ns=19,Nr=23 *MSGTYPE(HELLO)

12:55:35.618 [l2tp-control 50000 debug] [7/0/555 <l2tpmgr:1> l2tp.c:13050] [callid 106478e8]
[context: lns, contextID: 11]  [software internal user inbound protocol-log] L2TP Rx PDU, from
1.1.1.2:13661 to 1.1.1.1:13660 (20) l2tp:[TLS](1/0)Ns=23,Nr=20 *MSGTYPE(HELLO)
```

最後に、既存のトンネルで hello メッセージへの応答がなく、コールおよびトンネルが切断される例を示します。 Monitor Subscriber の出力：

```
 <<<<OUTBOUND  14:06:21:406 Eventid:50001(3)
L2TP Tx PDU, from 1.1.1.2:13661 to 1.1.1.1:13661 (20)
l2tp:[TLS](2/0)Ns=4,Nr=2 *MSGTYPE(HELLO)

<<<<OUTBOUND  14:06:22:413 Eventid:50001(3)
L2TP Tx PDU, from 1.1.1.2:13661 to 1.1.1.1:13661 (20)
l2tp:[TLS](2/0)Ns=4,Nr=2 *MSGTYPE(HELLO)

<<<<OUTBOUND  14:06:24:427 Eventid:50001(3)
L2TP Tx PDU, from 1.1.1.2:13661 to 1.1.1.1:13661 (20)
l2tp:[TLS](2/0)Ns=4,Nr=2 *MSGTYPE(HELLO)

<<<<OUTBOUND  14:06:28:451 Eventid:50001(3)
L2TP Tx PDU, from 1.1.1.2:13661 to 1.1.1.1:13661 (20)
l2tp:[TLS](2/0)Ns=4,Nr=2 *MSGTYPE(HELLO)

<<<<OUTBOUND  14:06:36:498 Eventid:50001(3)
L2TP Tx PDU, from 1.1.1.2:13661 to 1.1.1.1:13661 (20)
l2tp:[TLS](2/0)Ns=4,Nr=2 *MSGTYPE(HELLO)

<<<<OUTBOUND  14:06:44:446 Eventid:50001(3)
L2TP Tx PDU, from 1.1.1.2:13661 to 1.1.1.1:13661 (38)
l2tp:[TLS](2/0)Ns=5,Nr=2 *MSGTYPE(StopCCN) *RESULT_CODE(2/0) *ASSND_TUN_ID(6)
```

以下はそれぞれのログです。

試行失敗の出力「Control tunnel timeout - retry-attempted 5, last-interval 8000 ms」に注目してください。

```
 14:06:21.406 [l2tp-control 50001 debug] [7/0/9133 <l2tpmgr:2> l2tpsnx_proto.c:1474] [callid
42c22625] [context: destination, contextID: 3]  [software internal user outbound protocol-log]
L2TP Tx PDU, from 1.1.1.2:13661 to 1.1.1.1:13661 (20)
l2tp:[TLS](2/0)Ns=4,Nr=2 *MSGTYPE(HELLO)
14:06:22.413 [l2tp-control 50001 debug] [7/0/9133 <l2tpmgr:2> l2tpsnx_proto.c:1474] [callid
42c22625] [context: destination, contextID: 3]  [software internal user outbound protocol-log]
L2TP Tx PDU, from 1.1.1.2:13661 to 1.1.1.1:13661 (20)
l2tp:[TLS](2/0)Ns=4,Nr=2 *MSGTYPE(HELLO)
14:06:24.427 [l2tp-control 50001 debug] [7/0/9133 <l2tpmgr:2> l2tpsnx_proto.c:1474] [callid
42c22625] [context: destination, contextID: 3]  [software internal user outbound protocol-log]
L2TP Tx PDU, from 1.1.1.2:13661 to 1.1.1.1:13661 (20)
l2tp:[TLS](2/0)Ns=4,Nr=2 *MSGTYPE(HE LLO)
14:06:28.451 [l2tp-control 50001 debug] [7/0/9133 <l2tpmgr:2> l2tpsnx_proto.c:1474] [callid
42c22625] [context: destination, contextID: 3]  [software internal user outbound protocol-log]
L2TP Tx PDU, from 1.1.1.2:13661 to 1.1.1.1:13661 (20)
l2tp:[TLS](2/0)Ns=4,Nr=2 *MSGTYPE(HELLO)
14:06:36.498 [l2tp-control 50001 debug] [7/0/9133 <l2tpmgr:2> l2tpsnx_proto.c:1474] [callid
```

```
42c22625] [context: destination, contextID: 3]  [software internal user outbound protocol-log]
L2TP Tx PDU, from 1.1.1.2:13661 to 1.1.1.1:13661 (20)
l2tp:[TLS](2/0)Ns=4,Nr=2 *MSGTYPE(HELLO)
14:06:44.446 [l2tp-control 50068 warning] [7/0/9133 <l2tpmgr:2> l2tp.c:14841] [callid 42c22625]
[context: destination, contextID: 3]  [software internal user] L2TP (Local[svc: lac]: 6
Remote[1.1.1.1]: 2): Control tunnel timeout -  retry-attempted 5 , last-interval 8000 ms, Sr 2,
Ss 5, num-pkt-not-acked 1, Sent-Q-len 1, tun-recovery-flag 0, instance-recovery-flag 0, msg-type
Hello
14:06:44.446 [l2tp-control 50001 debug] [7/0/9133 <l2tpmgr:2> l2tpsnx_proto.c:1474] [callid
42c22625] [context: destination, contextID: 3]  [software internal user outbound protocol-log]
L2TP Tx PDU, from 1.1.1.2:13661 to 1.1.1.1:13661 (38)
l2tp:[TLS](2/0)Ns=5,Nr=2 *MSGTYPE(StopCCN) *RESULT_CODE(2/0) *ASSND_TUN_ID(6)
14:06:44.447 [l2tp-control 50069 debug] [7/0/9133 <l2tpmgr:2> l2tpsnx_fsm.c:105] [callid
42c22625] [context: destination, contextID: 3]  [software internal user] l2tp fsm: state
L2TPSNX_STATE_CONNECTED event L2TPSNX_EVNT_PROTO_SESSION_DISCONNECTED
```

以下は対応する SNMP トラップです。

```
 14:06:44 2009 Internal trap notification 92 (L2TPTunnelDownPeerUnreachable)  context
destination service lac peer address 1.1.1.1 local address 1.1.1.2
```

Show output の考慮事項

次のコマンドを実行すると、特定のピア (または特定の LAC/LNS サービスのすべてのトンネル) にピアの到達可能性の問題があったかどうかが示されます。

```
 show l2tp statistics (peer-address <peer ip address> | ((lac-service | lns-service) <lac or lns
service name>))
```

[Active Connections] カウンタは、そのピアの既存のトンネルの数と一致します。上記の show l2tp tunnels all の出力で確認したように、この数は 1 より大きいことがあります。

[Failed to Connect] カウンタは、トンネル セットアップが失敗した回数を示します。
[Max Retry Exceeded] カウンタは最も重要なカウンタである可能性があります。このカウンタは、タイムアウトが原因の接続失敗を示します (再試行期間が経過するたびに、L2TPTunnelDownPeerUnreachable トラップが発生します)。 この情報からは、特定のピアでの問題発生の頻度のみがわかり、タイムアウトになった理由を特定することはできません。 ただし、あらゆる情報を組み合わせて解決するトラブルシューティング プロセス全体においては、頻度も重要な解決要素となる可能性があります。

[Sessions] セクションには、サブスクライバ セッション レベルの詳細情報が (トンネル レベルと比較する形で) 提供されます。
[Active Sessions] カウンタは、特定のピアに対して show l2tp tunnels を実行した場合の [Active Sess] 列の出力の合計 (1 つのピアにトンネルが複数ある場合) と一致します。
[Failed to Connect] カウンタは、接続に失敗したセッション数を示します。 セッション セットアップが失敗しても L2TPTunnelDownPeerUnreachable トラップはトリガーされません。トリガーされるのは、トンネル セットアップが失敗した場合のみです。

show l2tp tunnels コマンドのカウンタ版も役立つ場合があります。

```
 show l2tp tunnels counters peer-address <peer address>
```

さらに、セッション レベルでは、特定のピアのすべてのサブスクライバを確認できます。

```
 show l2tp sessions peer-address <peer ip address>
```

検出されるサブスクライバの数は、前述したとおり、アクティブ セッションの数と一致します。