

# StarOS中的L2TP — 在ASR5k上实施并排除L2TP对等故障 — L2TPTunnelDownPeerUnreachable

## 目录

### [简介](#)

#### [什么是L2TP?](#)

#### [我们在移动性中如何使用它?](#)

#### [ASR5x00在此设置中是什么?](#)

#### [L2TP LAC支持](#)

#### [L2TP LNS支持](#)

#### [在ASR5k上启用思科设备上服务的配置](#)

#### [ASR5k上LAC的配置示例](#)

#### [ASR5k上LNS的配置示例](#)

#### [Cisco IOS设备上LNS的配置示例](#)

#### [排除对等体不可达事件故障](#)

#### [使用案例:由于重试超时,初始隧道设置失败](#)

#### [使用案例:由于keepalive,初始隧道设置失败](#)

#### [显示输出注意事项](#)

## 简介

本文档介绍如何在ASR5k上实施StarOS中的第2层隧道协议(L2TP),并排除L2TP对等故障 — L2TPTunnelDownPeerUnreachable。

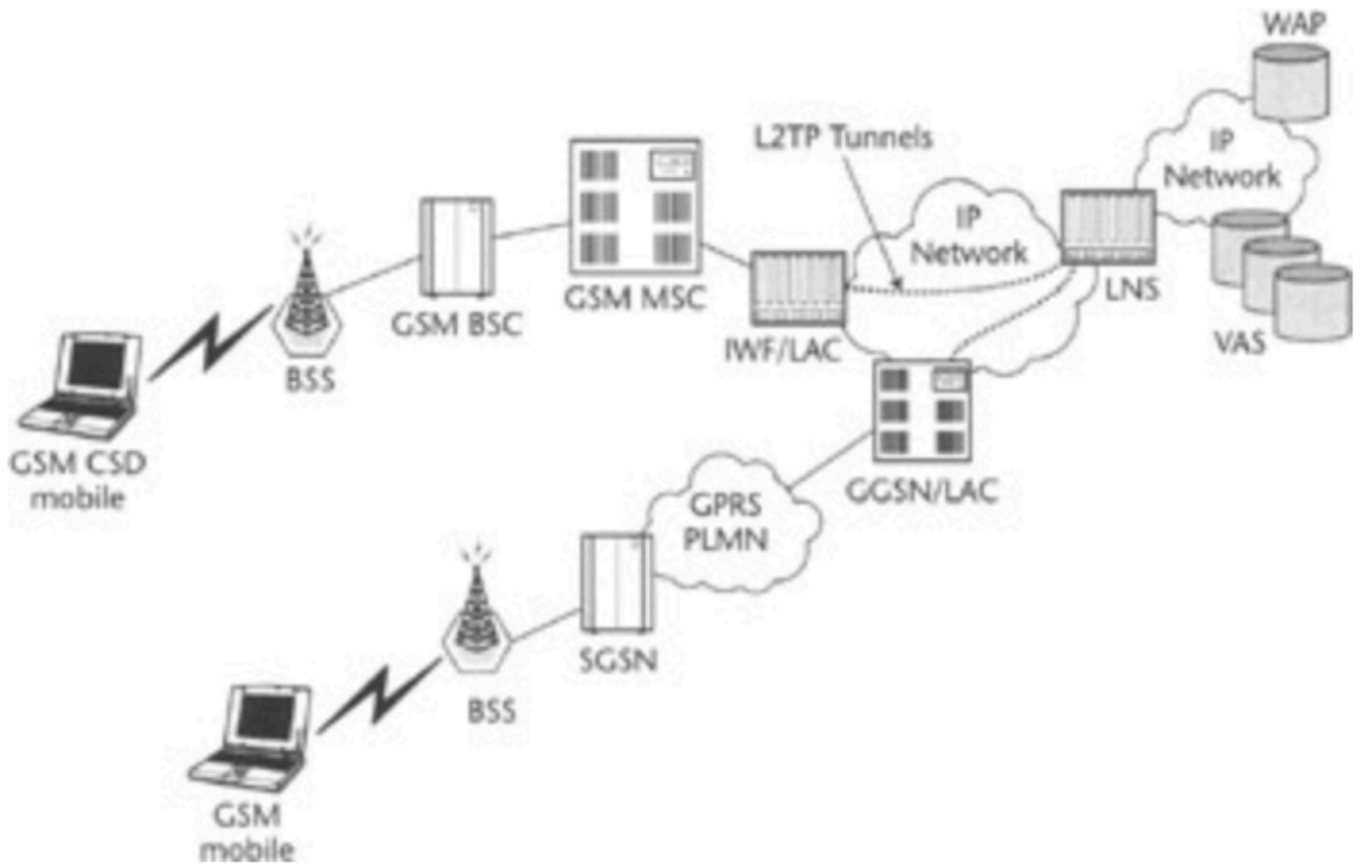
## 什么是L2TP?

L2TP 扩展了 PPP 的点对点性质。L2TP 为采用隧道的 PPP 帧传输提供了一种封装方法,允许 PPP 终点通过分组交换网络建立隧道。在远程访问类型的情况下最常部署 L2TP,这些情况使用 Internet 提供 Intranet 类型的服务。其概念是虚拟专用网络(VPN)。

L2TP的两个主要物理元素是L2TP接入集中器(LAC)和L2TP网络服务器(LNS):

- LAC:LAC是充当隧道终端一端的LNS的对等体。LAC 将终止远程 PPP 连接并处于远程与 LNS 之间。转发的数据包通过 PPP 连接进出远程连接。进出 LNS 的数据包通过 L2TP 隧道进行转发。
- LNS:LNS是充当隧道终端一端的LAC的对等体。LNS 是 LAC PPP 隧道会话的终点。它用于聚集多个采用 LAC 隧道的 PPP 会话并进入专用网络。

如下图所示,简化了移动网络中的L2TP设置。



L2TP 使用以下两种不同的消息类型：

- 控制消息：L2TP通过单独的控制通道和数据通道传递控制和数据消息。带内控制信道用于传递顺序控制连接管理、呼叫管理、错误报告和会话控制消息。控制连接的建立并非特定于 LAC 或 LNS，而是特定于与控制连接建立相关的隧道发送方和接收方。在隧道终点之间采用共享密钥身份验证方法。
- 数据消息：数据消息用于封装发送到L2TP隧道的PPP帧。

详细的呼叫流程和隧道建立说明如下：

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

## 我们在移动性中如何使用它？

典型部署适用于GGSN充当LAC并建立到在公司网络中运行的LNS的安全隧道的企业用户。GGSN配置指南附录中提供了详细的呼叫流程，可根据特定软件版本在以下位置找到：

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

## ASR5x00在此设置中是什么？

ASR5k可支持LAC和LNS功能。

## L2TP LAC支持

L2TP在将用户PPP连接隧道化为L2TP会话之前，在LAC和LNS之间建立L2TP控制隧道。LAC服务基于与GGSN相同的架构，并受益于动态资源分配和分布式消息和数据处理。此设计允许LAC服务支持每秒4000次以上的设置，或最大3G以上的吞吐量。在单个隧道中最多可以有65535个会话，每个系统使用32,000个隧道最多可以有500,000个L2TP会话。

## L2TP LNS支持

配置为第2层隧道协议网络服务器(LNS)的系统支持从L2TP接入集中器(LAC)在终端安全虚拟专用网络(VPN)隧道之间。

L2TP在将用户PPP连接隧道化为L2TP会话之前，在LAC和LNS之间建立L2TP控制隧道。每个LNS最多可以有65535个会话和500,000个会话。

LNS架构类似于GGSN，利用解复用器的概念，在无需操作员干预的情况下，在平台上的可用软件和硬件资源之间智能地分配新的L2TP会话。

有关详细信息，请参阅PGW/GGSN配置指南。

## 在ASR5k上启用思科设备上的服务的配置

### ASR5k上LAC的配置示例

```
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上LNS的配置示例

```
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
}
```

**注意：**同一IP接口上的多个地址可以绑定到不同的LNS服务。但是，每个地址只能绑定到一个LNS服务。此外，LNS服务无法绑定到与其他服务（如LAC服务）相同的接口。

## Cisco IOS设备上LNS的配置示例

这可用作Cisco IOS配置的支持配置示例，不受本文的约束。

## LNS 配置

```
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
```

## 排除对等体不可达事件故障

本节将介绍如何排除网络中L2TPTunnelDownPeerUnreachable事件故障的一些指南。此处参考PDSN关闭RP进行说明，但使用GGSN/PGW进行故障排除时，故障排除步骤相同。

作为提醒，创建LAC到LNS隧道以包含用户会话，同时将用户连接从PDSN/HA/GGSN/PGW扩展到终止LNS并提供IP地址的LNS。如果在StarOS机箱上，LNS将从已配置的IP池获取IP地址。如果在某些其他LNS上，例如在客户驻地，则IP地址由LNS提供。在后一种情况下，这可以有效地允许用户通过在漫游合作伙伴上运行的LAC连接到其家庭网络。

在尝试设置第一个用户会话时，首先创建LAC LNS隧道，只要隧道中有会话，该隧道就会保持运行。

当给定隧道的最后一个会话结束时，该隧道将关闭或关闭。同一LAC-LNS对等体之间可以建立多个隧道。

以下是命令**show l2tp tunnels**的输出片段，在本例中，所有输出都显示了这一点：机箱同时托管LAC和LNS服务（TestLAC和TestLNS）。请注意，LAC和LNS隧道ALL都有会话，而某些关闭的RP隧道没有会话。

```
[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

可以使用

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

以下是使用LAC服务1.1.1.2和LNS服务（对等体）1.1.1.1的L2TPTunnelDownPeerUnreachable陷阱的示例

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

使用命令show snmp trap statistics获取此陷阱触发次数（自重新加载或上次重置统计信息后）的计数

当隧道设置超时发生或保持连接(Hello)数据包未响应时，会为L2TP触发L2TPTunnelDownPeerUnreachable陷阱。原因通常是LNS对等体未响应来自LAC的请求或在任一方向上的传输问题。

没有陷阱表示对等体可访问，如果不了解如何进一步调查，则会导致在调查时是否仍然存在问题（已提交功能请求）的困惑。

要继续，我们最需要的部分是对等IP地址。第一步是确保PING可检查IP连接。如果存在连接，您可以继续调试

```
****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)
```

注意：

**l2tpmgr跟踪特定用户会话设置**

## L2tp-control跟踪隧道建立：

以下是此输出的调试示例

### 使用案例:由于重试超时，初始隧道设置失败

```
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
```

以下是系统确定故障时触发的与上述日志匹配的SNMP陷阱

```
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
```

使用案例:由于重试超时，初始隧道设置失败 — 分析

我们看到隧道在16:34出现，它尝试发送5次挑战。显然，没有回复，隧道最终断开。

查看配置默认值或配置的值，并参阅

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

此配置将分为1秒后首次重传，然后呈指数级增长 — 每次翻倍：1、2、4、8、8。

请注意，术语max-retransmissions(5)包括第一次尝试/传输。  
retransmission-timeout-max是达到此限制后传输的最长时间  
retransmission-timeout-first是第一次重新传输前等待多长时间的起点。

因此，在计算默认参数时，在 $1 + 2 + 4 + 8 + 8$ 秒= 23秒后会发生故障，这与以下输出中的结果完全相同。

## 使用案例:由于keepalive，初始隧道设置失败

L2TPTunnelDownPeerUnreachable陷阱的另一个原因是没有响应keepalive-interval消息。在没有控制消息或数据通过隧道发送的期间使用这些消息，以确保另一端仍处于活动状态。如果隧道中有会话，但它们未执行任何操作，则此命令可确保隧道仍能正常工作，因为通过启用该会话，keepalive消息将在配置的无数据包交换时间段（即60秒）后发送，并且预期会做出响应。发送第一个保持连接后不收到响应的频率与上述隧道设置的相同。因此，在23秒内未收到对hello(keepalive)消息的响应后，隧道将断开。请参阅可配置的keepalive-interval（默认值为60s）。

以下是成功保持连接交换的示例，包括监控用户和日志记录。请注意，由于一分钟内未传输用户数据，消息集之间的间隔为一分钟。在本示例中，LAC和LNS服务位于同一机箱中，分别位于名为destination和Ins的情景中。

```
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)
```

最后，以下是一个示例，对于EXISTING隧道，呼叫消息未响应，呼叫和隧道已断开。监控用户输出：

```
<<<<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)
```

以下是各自的日志。

**请注意输出Control tunnel timeout - retry-attempted 5, last-interval 8000 ms for the failed attempts。**

```
14:06:21.406 [l2tp-control 50001 debug] [7/0/9133 <l2tpmgr:2> l2tpsrx_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> l2tpsrx_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> l2tpsrx_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)
14:06:28.451 [l2tp-control 50001 debug] [7/0/9133 <l2tpmgr:2> l2tpsrx_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> l2tpsrx_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> l2tpsrx_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> l2tpsrx_fsm.c:105] [callid 42c22625] [context: destination, contextID: 3] [software internal user] l2tp fsm: state L2TPSNX_STATE_CONNECTED event L2TPSNX_EVNT_PROTO_SESSION_DISCONNECTED
```



## 和相应的SNMP陷阱

```
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
```

### 显示输出注意事项

运行以下命令将指示特定对等体（或特定lac/lns服务中的所有隧道）是否存在对等体可达性问题

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

Active Connections计数器匹配该对等体中可能存在多个现有隧道的数量，如之前show l2tp tunnels的输出所示。

Failed to Connect计数器将指示发生了多少个隧道设置失败。

Max Retry Exceeded计数器可能是最重要的计数器，因为它表示由于超时而无法连接（每个Retry exceeded都会导致L2TPTunnelDownPeerUnreachable陷阱）。此信息只告诉您给定对等体的问题频率，而不告诉您超时的原因。但了解频率有助于在整个故障排除过程中将各部分组合在一起。

“会话”部分提供用户会话级别（与隧道级别）的详细信息

“活动会话”计数器与特定对等体的show l2tp tunnels的“活动会话”列输出的总和（如果对等体有多个隧道）匹配。

Failed to Connect计数器指示连接失败的会话数。请注意，失败的会话设置不会触发L2TPTunnelDownPeerUnreachable陷阱，只有失败的隧道设置才会触发。

还有show l2tp tunnels命令的计数器版本可能有用。

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

最后，在会话级别，可以查看给定对等体的所有用户。

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

找到的用户数应与讨论的活动会话数相匹配。