StarOS의 L2TP - ASR5k에서 구현 및 L2TP 피어링 문제 해결 - L2TPTunnelDownPeerUnreachable

목차

소개
L2TP란?
모빌리티에서는 어디에서 사용합니까?
이 설치에서 ASR5x00이란 무엇입니까?
L2TP LAC 지원
L2TP LNS 지원
ASR5k의 Cisco 디바이스에서 서비스를 활성화하는 구성
ASR5k의 LAC에 대한 구성 샘플
ASR5k의 LNS에 대한 구성 샘플
Cisco IOS 디바이스의 LNS에 대한 구성 샘플
피어 연결 불가 이벤트 문제 해결
활용 사례:재시도 시간 제한으로 인해 초기 터널 설정 실패
활용 사례:keepalive로 인해 초기 터널 설정 실패
출력 고려 사항 표시

소개

이 문서에서는 StarOS의 L2TP(Layer 2 Tunneling Protocol)가 ASR5k에서 구현되고 L2TP 피어링 - L2TPTunnelDownPeerUnreachable 문제를 해결하는 방법에 대해 설명합니다.

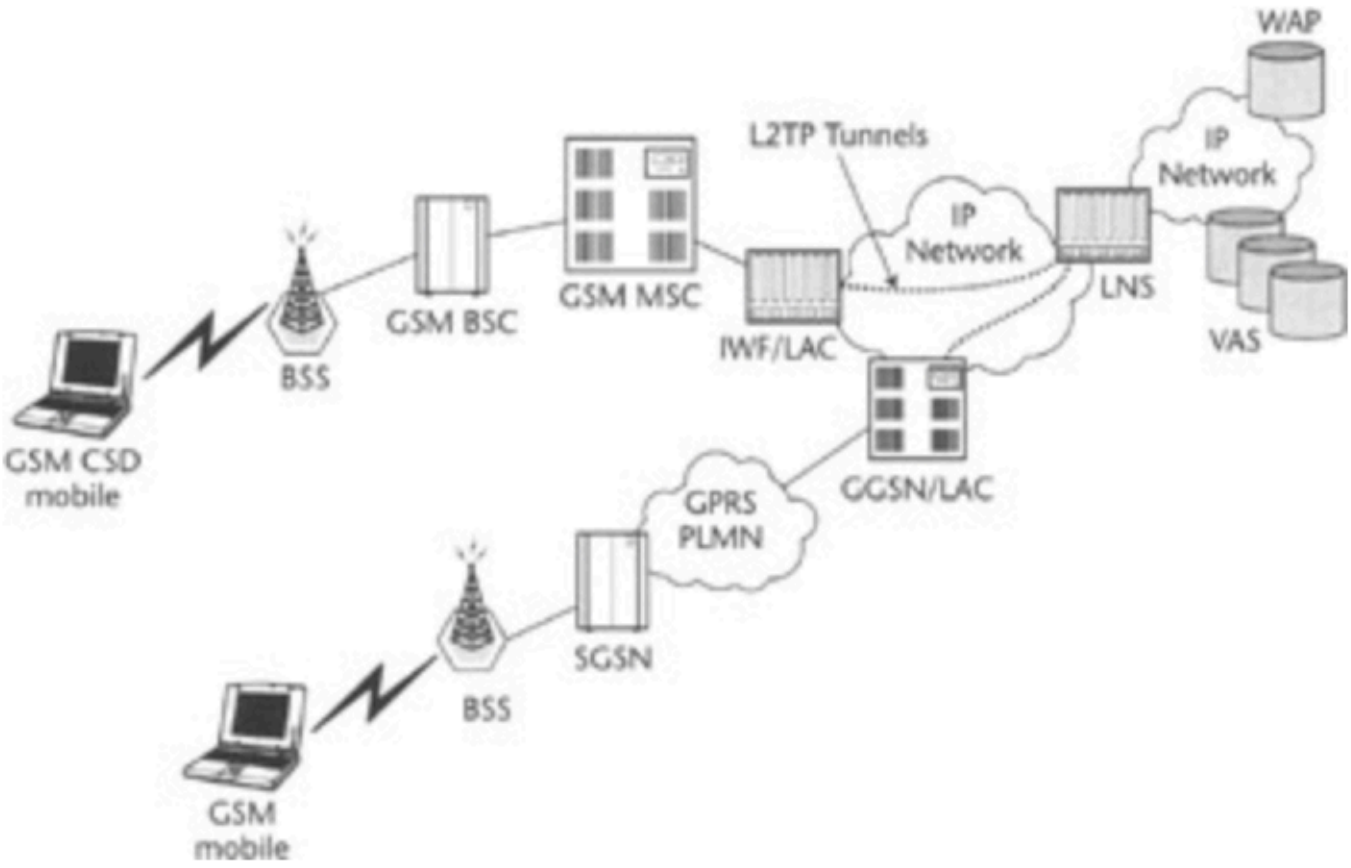
L2TP란?

L2TP는 PPP의 포인트 투 포인트 특성을 확장합니다.L2TP는 터널링된 PPP 프레임 전송을 위한 캡슐화 방법을 제공하며, 이를 통해 PPP 엔드포인트가 패킷 스위치드 네트워크를 통해 터널링될 수 있습니다.L2TP는 인터넷을 사용하여 인트라넷 유형 서비스를 제공하는 원격 액세스 유형 시나리오에서 가장 일반적으로 구축됩니다.개념은 VPN(Virtual Private Network)의 개념입니다.

L2TP의 두 가지 기본 물리적 요소는 L2TP LAC(Access Concentrator)와 L2TP Network Server(LNS)입니다.

- LAC:LAC는 터널 엔드포인트의 한 쪽 역할을 하는 LNS에 대한 피어입니다.LAC는 원격 PPP 연결을 종료하고 원격 및 LNS 사이에 위치합니다.패킷은 PPP 연결을 통해 원격 연결과 주고받습니다.LNS에서 들어오고 나가는 패킷은 L2TP 터널을 통해 전달됩니다.
- LNS:LNS는 터널 엔드포인트의 한 쪽 역할을 하는 LAC에 대한 피어입니다.LNS는 LAC PPP 터널링 세션의 종료 지점입니다.이는 여러 LAC 터널링된 PPP 세션 및 인그레스(ingress)를 프라이빗 네트워크로 통합하는 데 사용됩니다.

이 이미지에 표시된 대로 모바일 네트워크에서 간소화된 L2TP 설정



L2TP에서 사용하는 두 가지 메시지 유형이 있습니다.

- 제어 메시지:L2TP는 제어 및 데이터 메시지를 별도의 제어 및 데이터 채널에 전달합니다.대역 내 제어 채널은 순차 제어 연결 관리, 통화 관리, 오류 보고 및 세션 제어 메시지를 전달합니다 .제어 연결 시작이 LAC 또는 LNS에 한정되지 않고 제어 연결 설정과 관련성이 있는 터널 생성자 및 수신자에 한정되지 않습니다.공유 암호 챌린지 인증 방법은 터널 엔드포인트 간에 사용됩니다.
- 데이터 메시지:데이터 메시지는 L2TP 터널로 전송되는 PPP 프레임을 캡슐화하는 데 사용됩니다.

자세한 통화 흐름 및 터널 설정에 대해서는 여기에서 설명합니다.

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

모빌리티에서는 어디에서 사용합니까?

일반적인 구축은 GGSN이 LAC 역할을 하며 기업 네트워크에서 작동하는 LNS에 대한 보안 터널을 설정하는 기업 사용자를 위한 것입니다.자세한 통화 흐름은 GGSN 컨피그레이션 가이드의 부록에서 확인할 수 있으며, 특정 소프트웨어 버전별로 여기에서 확인할 수 있습니다.

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

이 설치에서 ASR5x00이란 무엇입니까?

ASR5k는 LAC 및 LNS 기능을 지원할 수 있습니다.

L2TP LAC 지원

L2TP는 가입자 PPP 연결을 L2TP 세션으로 터널링하기 전에 LAC와 LNS 간에 L2TP 제어 터널을 설정합니다.LAC 서비스는 GGSN과 동일한 아키텍처를 기반으로 하며, 동적 리소스 할당 및 분산된 메시지 및 데이터 처리 기능의 이점을 제공합니다.이 설계를 통해 LAC 서비스는 초당 4,000개 이상의 설정 또는 최대 3G 이상의 처리량을 지원할 수 있습니다.단일 터널에는 최대 65535개의 세션이 있을 수 있으며 시스템당 32,000개의 터널을 사용하여 최대 500,000개의 L2TP 세션이 있을 수 있습니다.

L2TP LNS 지원

LNS(Layer 2 Tunneling Protocol Network Server)로 구성된 시스템은 L2TP LAC(Access Concentrator) 간 종료 보안 VPN(Virtual Private Network) 터널을 지원합니다.

L2TP는 가입자 PPP 연결을 L2TP 세션으로 터널링하기 전에 LAC와 LNS 간에 L2TP 제어 터널을 설정합니다.단일 터널에서 최대 65535개의 세션과 LNS당 최대 500,000개의 세션이 있을 수 있습니다.

LNS 아키텍처는 GGSN과 유사하며, 디멀티플렉서 개념을 활용하여 운영자의 개입 없이 플랫폼에서 사용 가능한 소프트웨어 및 하드웨어 리소스에 새 L2TP 세션을 지능적으로 할당합니다.

자세한 내용은 PGW/GGSN 컨피그레이션 가이드를 참조하십시오.

ASR5k의 Cisco 디바이스에서 서비스를 활성화하기 위한 구성

ASR5k의 LAC에 대한 구성 샘플

```
 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
```

ASR5k의 LNS에 대한 구성 샘플

```
  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
}
```

참고:동일한 IP 인터페이스의 여러 주소를 서로 다른 LNS 서비스에 바인딩할 수 있습니다.그러나 각 주소는 하나의 LNS 서비스에만 바인딩할 수 있습니다.또한 LNS 서비스는 LAC 서비스와 같은 다른 서비스와 동일한 인터페이스에 바인딩할 수 없습니다.

Cisco IOS 디바이스의 LNS에 대한 구성 샘플

이는 Cisco IOS 컨피그레이션의 지원 컨피그레이션 샘플로 사용할 수 있으며 이 문서의 적용을 받지 않습니다.

LNS 컨피그레이션

```
 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
```

피어 연결 불가 이벤트 문제 해결

이 섹션에서는 네트워크에서 L2TPTunnelDownPeerUnreachable 이벤트를 해결하는 방법에 대한 몇 가지 지침을 제공합니다.이 설명서는 PDSN 닫힌 RP에 대한 참조와 함께 설명되지만 GGSN/PGW를 사용한 트러블슈팅 단계도 동일합니다.

LAC-LNS 터널은 가입자 세션을 포함하는 동안 가입자 연결을 PDSN/HA/GGSN/PGW에서 종료되고 IP 주소가 제공되는 LNS로 확장하기 위해 생성됩니다.StarOS 섀시의 경우 LNS는 구성된 IP 풀에서 IP 주소를 가져옵니다.예를 들어 고객 프레미스에서 다른 LNS에 있는 경우 LNS에서 IP 주소를 제공합니다.후자의 시나리오에서는 사용자가 로밍 파트너에서 실행되는 LAC를 통해 홈 네트워크에 효과적으로 연결할 수 있습니다.

LAC LNS 터널은 첫 번째 가입자 세션을 설정하려고 시도할 때 처음 생성되며 터널에 세션이 있는 한 계속 작동합니다.

지정된 터널에 대한 마지막 세션이 종료되면 해당 터널이 닫히거나 종료됩니다.동일한 LAC-LNS 피어 간에 둘 이상의 터널을 설정할 수 있습니다.

다음은 **show l2tp tunnels all**의 출력 조각으로, 이 경우 섀시가 LAC 및 LNS 서비스(TestLAC 및 TestLNS)를 모두 호스팅합니다. LAC 및 LNS 터널에는 모두 세션이 있지만 일부 닫힌 RP 터널에는 세션이 없습니다.

```
 [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
```

서비스 구성은

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

다음은 LAC 서비스 1.1.1.2 및 LNS 서비스(피어)를 사용하는 L2TPTunnelDownPeerUnreachable 트랩의 예입니다. 1.1.1.1

```
 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 명령을 사용하여 이 트랩이 트리거된 횟수(통계를 다시 로드하거나 마지막으로 재설정한 이후)를 **확인합니다.**

L2TPTunnelDownPeerUnreachable 트랩은 터널 설정 시간 초과가 발생하거나 Keep-alive(Hello) 패킷이 응답하지 않을 때 L2TP에 대해 트리거됩니다.원인은 일반적으로 LNS 피어가 LAC의 요청에 응답하지 않거나 전송 문제가 어느 방향으로든 발생했기 때문입니다.

피어가 도달 가능함을 나타내는 트랩이 없으며, 추가 조사 방법을 모르는 경우 조사 시 여전히 문제가 있는지 여부(기능 요청이 제출됨)에 대한 혼동을 일으킬 수 있습니다.

계속하려면 피어 IP 주소가 가장 필요합니다.첫 번째 단계는 PING을 통해 확인할 수 있는 IP 연결이 있는지 확인하는 것입니다.연결이 있는 경우 디버그를 진행할 수 있습니다

```
****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)
```
참고:

l2tpmgr 특정 가입자 세션 설정 추적

l2tp 제어 트랙 터널 설정:

이 출력의 샘플 디버그

활용 사례:재시도 시간 제한으로 인해 초기 터널 설정 실패

```
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
```

다음은 시스템이 장애를 확인하는 순간까지 위의 로그와 일치하도록 트리거된 결과 SNMP 트랩입니다

```
 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
```

활용 사례:재시도 시간 제한으로 인해 초기 터널 설정 실패 - 분석

우리가 보는 것은 그 터널이 16시 34분에 나타나며 그것은 5번이나 도전장을 보내려고 시도한다는 것입니다.회신이 없어 결국 터널의 연결이 끊어지는 게 분명하다.

컨피그레이션 기본값 또는 구성된 값을 확인하고

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

이 컨피그레이션은 1초 후 첫 번째 재전송으로 상호 연결되고, 그 다음 기하급수적으로 증가하여 매번 두 배로 증가합니다.1, 2, 4, 8, 8.

max-retransmission (5)이라는 용어는 첫 번째 시도/전송을 포함합니다.
retransmission-timeout-max는 이 제한에 도달한 후 전송 사이의 최대 시간입니다.
retransmission-timeout-first는 첫 번째 재전송까지 기다리는 시간의 시작점입니다.

따라서 계산을 수행하면 기본 매개변수의 경우 1+ 2 + 4 + 8 + 8초 = 23초 후에 오류가 발생합니다.이는 아래 출력과 정확히 같습니다.

활용 사례:keepalive로 인해 초기 터널 설정 실패

L2TPTunnelDownPeerUnreachable 트랩의 다른 이유는 keepalive-interval 메시지에 대한 응답이 아닙니다.이러한 항목은 다른 끝이 계속 활성 상태를 유지하도록 터널을 통해 전송되는 제어 메시지나 데이터가 없는 기간 동안 사용됩니다.터널에 세션이 있지만 아무 작업도 수행하지 않는 경우 이 명령을 사용하면 패킷 교환 없이 구성된 기간(예: 60초) 후에 keepalive 메시지가 전송되고 응답이 예상되므로 터널이 제대로 작동하는지 확인합니다.첫 번째 응답을 보내고 응답을 받지 못한 후 keepalive를 전송하는 빈도는 터널 설정에 대해 위에서 설명한 것과 동일합니다.따라서 hello(keepalive) 메시지에 대한 응답을 받지 못한 23초가 지나면 터널이 해제됩니다.구성 가능한 keepalive-interval(기본값 = 60s)을 참조하십시오.

다음은 모니터 가입자 및 로깅 모두에서 성공적인 keep-alive 교환의 예입니다.1분 동안 사용자 데이터가 전송되지 않아 메시지 집합 사이의 1분 간격에 유의하십시오.이 예에서 LAC 및 LNS 서비스는 각각로 명명된 컨텍스트**에서** 동일한 섀시에 **있습니다**.

```
 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)
```

마지막으로, EXISTING 터널의 경우 hello 메시지가 응답하지 않고 통화와 터널이 해제되는 예가 있습니다.가입자 출력 모니터링:

```
 <<<<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)
```

각 로그는 다음과 같습니다.

실패한 시도에 대한 Control 터널 시간 초과 - retry-attempted 5, last-interval 8000ms의 출력을 확인합니다.

```
 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
```

해당 SNMP 트랩

```
 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
```

출력 고려 사항 표시

다음 명령을 실행하면 특정 피어(또는 특정 lac/lns 서비스의 모든 터널)에 대한 피어 연결 문제가 발생했는지 여부를 나타냅니다.

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

Active Connections 카운터는 해당 피어에 대한 기존 터널의 수를 일치시킵니다. 이는 show l2tp tunnels의 출력에 나와 있는 것처럼 둘 이상의 터널이 있을 수 있습니다.

Failed to Connect 카운터는 발생한 터널 설정 실패 횟수를 나타냅니다.
Max Retry Exceeded 카운터는 시간 제한으로 인해 연결하지 못했음을 나타내므로 가장 중요한 카운터일 수 있습니다(각 Retry가 초과되면 L2TPTunnelDownPeerUnreachable 트랩이 발생함). 이 정보는 지정된 피어에 대한 문제의 빈도만 알려주며, 시간 제한이 발생한 이유를 알려주지 않습니다.그러나 빈도를 알면 전반적인 문제 해결 프로세스에 여러 요소를 통합하는 데 도움이 될 수 있습니다.

[세션] 섹션은 가입자 세션 레벨(터널 레벨 대)에서 세부 정보를 제공합니다.
Active Sessions 카운터는 특정 피어에 대한 show l2tp 터널의 Active Sess 열 출력의 합계(피어에 대해 둘 이상의 터널이 있는 경우)와 일치합니다.
Failed to Connect 카운터는 연결하지 못한 세션 수를 나타냅니다.실패한 세션 설정은 L2TPTunnelDownPeerUnreachable 트랩을 트리거하지 않으며 실패한 터널 설정만 트리거합니다.

유용한 show l2tp tunnels 명령의 카운터 버전도 있습니다.

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

마지막으로, 세션 레벨에서 지정된 피어에 대한 모든 가입자를 볼 수 있습니다.

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

찾은 가입자 수는 앞서 설명한 활성 세션 수와 일치해야 합니다.