L2TP in StarOS - Implementierung auf dem ASR5k und Fehlerbehebung beim L2TP-Peering - L2TPTunnelDownPeerUnreachable

Inhalt

Einführung

In diesem Dokument wird beschrieben, wie das Layer 2 Tunneling Protocol (L2TP) in StarOS auf dem ASR5k implementiert wird und L2TP Peering - L2TPTunnelDownPeerUnreachable (L2TP-Peering-Problembehebung) behoben wird.

Was ist L2TP?

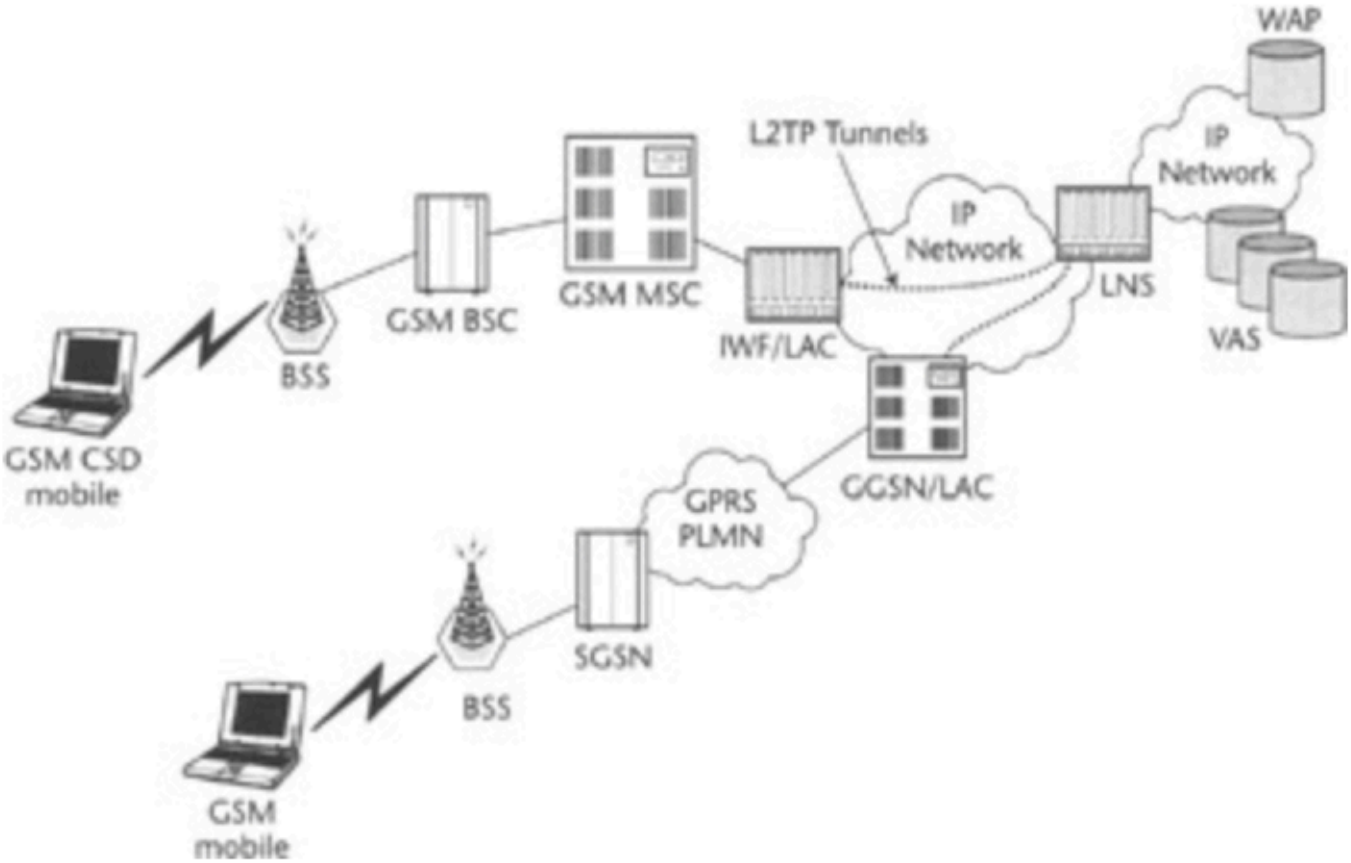
L2TP erweitert den Point-to-Point-Charakter von PPP. L2TP bietet eine Kapselungsmethode für die Übertragung von getunnelten PPP-Frames, mit der die PPP-Endpunkte über ein paketvermitteltes Netzwerk getunnelt werden können. L2TP wird in der Regel in Szenarien mit Remote-Zugriff bereitgestellt, die das Internet verwenden, um Intranet-Dienste anzubieten. Das Konzept ist ein Virtual Private Network (VPN).

Die beiden primären physischen Elemente von L2TP sind der L2TP Access Concentrator (LAC) und der L2TP Network Server (LNS):

- LAC: Die LAC ist ein Peer zum LNS, der als eine Seite des Tunnelendpunkts fungiert. Die LAC terminiert die Remote-PPP-Verbindung und befindet sich zwischen der Remote-Verbindung und dem LNS. Pakete werden über die PPP-Verbindung an die Remote-Verbindung und von dieser an die PPP-Verbindung weitergeleitet. Pakete an das und vom LNS werden über den L2TP-Tunnel weitergeleitet.

- LNS: Das LNS ist ein Peer zur LAC, der als eine Seite des Tunnelendpunkts fungiert. Das LNS ist der Terminationspunkt für die PPP-Sitzungen des LAC. Diese Funktion dient zum Aggregieren mehrerer PPP-Sitzungen mit LAC-Tunneling und zum Eindringen in das private Netzwerk.

Vereinfachte L2TP-Einrichtung im mobilen Netzwerk, wie in diesem Bild gezeigt.



L2TP verwendet zwei verschiedene Meldungstypen:

- Steuerungsmeldungen: L2TP übergibt Kontroll- und Datennachrichten über separate Steuerungs- und Datenkanäle. Der In-Band-Steuerungskanal leitet sequenzielle Nachrichten für das Management von Steuerungsverbindungen, Anrufverwaltung, Fehlerberichte und Sitzungssteuerung weiter. Die Initiierung der Steuerungsverbindung ist nicht spezifisch für die LAC oder das LNS, sondern vielmehr für den Tunneloriginator und -empfänger, der für die Einrichtung der Steuerungsverbindung relevant ist. Zwischen den Tunnelendpunkten wird eine Authentifizierungsmethode für die gemeinsam genutzte geheime Herausforderung verwendet.
- Datenmeldungen: Datennachrichten werden verwendet, um die PPP-Frames zu kapseln, die in den L2TP-Tunnel gesendet werden.

Detaillierte Anrufflüsse und Tunnelaufbau werden hier erläutert:

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

Wo nutzen wir es in Mobility?

Die typische Bereitstellung ist für Unternehmensbenutzer vorgesehen, bei denen das GGSN als

LAC fungiert und sichere Tunnel zu LNS herstellt, die im Unternehmensnetzwerk betrieben werden. Detaillierte Anrufabläufe finden Sie im Anhang des GGSN-Konfigurationsleitfadens, der für die jeweilige Softwareversion hier verfügbar ist:

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

Was ist ASR5x00 in dieser Konfiguration?

ASR5k kann LAC- und LNS-Funktionen unterstützen.

L2TP LAC-Unterstützung

L2TP stellt L2TP-Steuerungstunnel zwischen LAC und LNS her, bevor die PPP-Verbindungen des Teilnehmers als L2TP-Sitzungen getunnelt werden. Der LAC-Service basiert auf derselben Architektur wie der GGSN und profitiert von dynamischer Ressourcenzuweisung und verteilter Nachrichten- und Datenverarbeitung. Dieses Design ermöglicht dem LAC-Service die Unterstützung von mehr als 4.000 Setups pro Sekunde bzw. eines Durchsatzes von mehr als 3 G. Es können maximal 65.535 Sitzungen in einem Tunnel und bis zu 500.000 L2TP-Sitzungen mit 32.000 Tunneln pro System stattfinden.

L2TP-LNS-Unterstützung

Das als Layer 2 Tunneling Protocol Network Server (LNS) konfigurierte System unterstützt die Terminierung sicherer VPN-Tunnel zwischen L2TP Access Concentrators (LACs).

L2TP stellt L2TP-Steuerungstunnel zwischen LAC und LNS her, bevor die PPP-Verbindungen des Teilnehmers als L2TP-Sitzungen getunnelt werden. Es können maximal 65.535 Sitzungen in einem Tunnel und bis zu 500.000 Sitzungen pro LNS durchgeführt werden.

Die LNS-Architektur ähnelt dem GGSN und nutzt das Konzept eines De-Multiplexers, um ohne Benutzereingriff neue L2TP-Sitzungen intelligent über die verfügbaren Software- und Hardware-Ressourcen der Plattform hinweg zuzuweisen.

Weitere Informationen finden Sie in den PGW/GGSN-Konfigurationsleitfäden.

Konfiguration zur Aktivierung von Services auf den Cisco Geräten des ASR5k

Konfigurationsbeispiel für LAC auf ASR5k

```
 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
```

Konfigurationsbeispiel für LNS auf ASR5k

```
  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
}
```

> **Hinweis**: Mehrere Adressen auf derselben IP-Schnittstelle können an unterschiedliche LNS-Dienste gebunden werden. Jede Adresse kann jedoch an nur einen LNS-Dienst gebunden werden. Darüber hinaus kann der LNS-Dienst nicht an dieselbe Schnittstelle gebunden werden wie andere Dienste, z. B. ein LAC-Dienst.

Konfigurationsbeispiel für LNS auf Cisco IOS-Gerät

Dies kann als unterstützendes Konfigurationsbeispiel für die Cisco IOS-Konfiguration verwendet werden und unterliegt nicht diesem Artikel.

LNS-Konfiguration

```
 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
```

Fehlerbehebung bei nicht erreichbaren Peer-Ereignissen

Dieser Abschnitt enthält einige Richtlinien zur Fehlerbehebung für das L2TPTunnelDownPeerUnreachable-Ereignis im Netzwerk. Sie wird hier mit Bezug auf PDSN Closed RP erklärt, die Schritte zur Fehlerbehebung sind jedoch bei der Fehlerbehebung mit GGSN/PGW identisch.

Zur Erinnerung: Es wird ein LAC-zu-LNS-Tunnel erstellt, um Teilnehmersitzungen einzuschließen, während die Teilnehmerverbindung von einem PDSN/HA/GGSN/PGW auf das LNS erweitert wird, an dem er terminiert ist und wo eine IP-Adresse angegeben wird. Wenn sich das LNS in einem StarOS-Chassis befindet, erhält es eine IP-Adresse aus einem konfigurierten IP-Pool. Bei anderen LNS, z. B. am Kundenstandort, wird die IP-Adresse dort vom LNS bereitgestellt. Im zweiten Szenario könnte dies Benutzern effektiv ermöglichen, über eine LAC, die auf einem Roaming-Partner ausgeführt wird, eine Verbindung zu ihrem Heimnetzwerk herzustellen.

Ein LAC-LNS-Tunnel wird zuerst erstellt, wenn versucht wird, die erste Teilnehmersitzung einzurichten, und bleibt verfügbar, solange der Tunnel Sitzungen enthält.

Wenn die letzte Sitzung für einen bestimmten Tunnel endet, wird dieser Tunnel geschlossen oder geschlossen. Es können mehrere Tunnel zwischen denselben LAC-LNS-Peers eingerichtet werden.

Hier ein Ausschnitt der Ausgabe des Befehls **show l2tp tunnels all**, der dies in diesem Fall zeigt, hostet das Chassis sowohl LAC- als auch LNS-Dienste (TestLAC und TestLNS). Beachten Sie, dass alle LAC- und LNS-Tunnel Sitzungen haben, während einige geschlossene RP-Tunnel keine Sitzungen haben.

```
 [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
```

Die Servicekonfiguration kann angezeigt werden mit

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

Hier sehen Sie ein Beispiel für das L2TPTunnelDownPeerUnreachable-Trap mit dem LAC-Dienst 1.1.1.2 und dem LNS-Dienst (Peer) 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
```

Mit dem Befehl **show snmp trap statistics** wird ermittelt, wie oft dieses Trap ausgelöst wurde (seit dem erneuten Laden oder letzten Zurücksetzen der Statistiken).

Das L2TPTunnelDownPeerUnreachable-Trap wird für L2TP ausgelöst, wenn ein Tunnel-Setup-Timeout auftritt ODER Keep-Alive-Pakete (Hello) nicht beantwortet werden. Die Ursache liegt in der Regel darin, dass der LNS-Peer nicht auf Anfragen der LAC antwortet oder Transportprobleme in beide Richtungen auslöst.

Es gibt keine Falle, die darauf hinweist, dass der Peer erreichbar ist, was, wenn nicht verstanden wird, wie weiter untersucht werden soll, zu Verwirrung darüber führen kann, ob zum Zeitpunkt der Untersuchung noch ein Problem besteht (eingereichte Funktionsanfrage).

Um fortzufahren, benötigen wir vor allem die Peer-IP-Adresse. Der erste Schritt besteht darin, sicherzustellen, dass eine IP-Verbindung vorhanden ist, die mit PING überprüft werden kann. Wenn eine Verbindung besteht, können Sie mit dem Debuggen fortfahren

```
****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)
```

Hinweise:

l2tpmgr verfolgt spezielle Einrichtung von Teilnehmersitzungen

l2tp control verfolgt Tunnelaufbau:

Hier sehen Sie ein Beispiel für das Debuggen aus dieser Ausgabe.

Anwendungsfall: Fehler bei der Ersteinrichtung des Tunnels aufgrund von Timeouts für Wiederholungsversuche

```
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
```

Das resultierende SNMP-Trap wird ausgelöst, um die oben genannten Protokolle für den Zeitpunkt zu übernehmen, an dem das System den Ausfall festgestellt hat.

```
 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
```

Anwendungsfall: Fehler beim erstmaligen Tunnel-Setup aufgrund von Timeouts für erneuten Versuch - Analyse

Wir sehen, dass Tunnel um 16:34 Uhr auftaucht und fünf Mal versucht, die Herausforderung zu senden. Offenbar gibt es keine Antwort und schließlich wird der Tunnel getrennt.

Prüfen Sie die Konfigurationsstandardwerte oder konfigurierten Werte, und sehen Sie

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

Diese Konfiguration soll als erste erneute Übertragung nach 1 Sekunde interpettiert werden, dann exponentielle Zunahme - Verdoppelung jedes Mal: 1, 2, 4, 8, 8.

Beachten Sie, dass der Begriff "maximale Neuübertragungen (fünf)" den ersten Versuch/die erste

Übertragung beinhaltet.
reTransmission-Timeout-max ist die maximale Zeit zwischen Übertragungen nach (wenn) Erreichen dieser Grenze.
reübertragung-timeout-first ist der Ausgangspunkt, wie lange vor der ersten erneuten Übertragung gewartet wird.

Also, indem Sie die Mathematik, im Fall der Standard-Parameter, würde ein Fehler nach 1 + 2 + 4 + 8 + 8 Sekunden = 23 Sekunden, die genau wie in der Ausgabe unten gesehen wird.

Anwendungsfall: Fehler bei der Ersteinrichtung des Tunnels aufgrund von Keepalives

Der andere Grund für das L2TPTunnelDownPeerUnreachable-Trap ist keine Antwort auf Keepalive-Intervallmeldungen. Diese werden in Zeiträumen verwendet, in denen keine Kontrollnachrichten oder Daten über den Tunnel gesendet werden, um sicherzustellen, dass das andere Ende noch am Leben ist. Wenn im Tunnel Sitzungen stattfinden, aber nichts geschieht, stellt dieser Befehl sicher, dass der Tunnel immer noch ordnungsgemäß funktioniert, da durch die Aktivierung Keepalive-Nachrichten nach der konfigurierten Zeit ohne Paketaustausch (d. h. 60 Sekunden) gesendet werden und Antworten erwartet werden. Die Häufigkeit, mit der der Keepalive nach dem Senden des ersten Keepalive und dem Nichterhalten einer Antwort gesendet wird, entspricht der oben für die Tunneleinrichtung beschriebenen. Wenn Sie also 23 Sekunden lang keine Antwort auf Hello-Nachrichten (Keepalive) erhalten haben, wird der Tunnel beendet. Siehe konfigurierbares Keepalive-Intervall (Standardwert = 60 s).

Im Folgenden finden Sie Beispiele für einen erfolgreichen Keep-Alive-Austausch, sowohl vom Monitorteilnehmer als auch von der Protokollierung. Beachten Sie das Intervall von einer Minute zwischen Nachrichtensätzen, da keine Benutzerdaten für eine Minute übertragen werden. In diesem Beispiel befinden sich die LAC- und LNS-Services im selben Chassis, in Kontexten mit dem Namen **destination** und **lns**.

```
 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)
```

Hier ein Beispiel, in dem bei einem vorhandenen Tunnel keine Hello-Nachrichten beantwortet werden und der Anruf und der Tunnel abgebrochen werden. Überwachung der

Teilnehmerausgabe:

```
 <<<<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)
```

Hier sind die jeweiligen Protokolle.

Beachten Sie das Output Control Tunnel Timeout (Ausgabe-Kontrolltunnel-Zeitüberschreitung): Wiederholungsversuch von fünf im letzten Intervall liegenden 8000 ms für fehlgeschlagene Versuche.

```
 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
```

und entsprechende SNMP-Traps

```
 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
```

Zeigen Sie Ausgabeüberlegungen.

Der folgende Befehl gibt an, ob Probleme mit der Erreichbarkeit von Peers mit einem bestimmten Peer (oder für alle Tunnel in einem bestimmten Lab/Lns-Dienst) aufgetreten sind.

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

Der Zähler für aktive Verbindungen vergleicht die Anzahl der vorhandenen Tunnel für diesen Peer. Es kann mehr als einen vorhanden sein, wie in der Ausgabe von show l2tp-Tunneln aus früheren Quellen zu sehen ist.

Der Zähler Verbindung fehlgeschlagen zeigt an, wie viele Tunnel-Setup-Fehler aufgetreten sind. Der Zähler Max Retry Exceeded ist wahrscheinlich der wichtigste Zähler, da er auf einen Verbindungsausfall aufgrund eines Timeouts hinweist (jeder Wiederholungswert überschreitet die Ergebnisse in einem L2TPTunnelDownPeerUnreachable-Trap). Diese Informationen geben Ihnen nur die Häufigkeit des Problems für einen bestimmten Peer an. Sie sagen nicht, warum das Timeout aufgetreten ist. Aber die Häufigkeit der Fehlerbehebung zu kennen, kann hilfreich sein, um die einzelnen Elemente im gesamten Fehlerbehebungsprozess zusammenzufassen.

Der Abschnitt Sitzungen enthält Details auf Teilnehmersitzungsebene (im Vergleich zu Tunnelebene).
Der Zähler für aktive Sitzungen entspricht der Summe (wenn mehr als ein Tunnel für einen Peer vorhanden ist) der Ausgabe der Spalte "Aktive Sitzung" aus den show l2tp-Tunneln für den jeweiligen Peer.
Der Zähler Verbindung fehlgeschlagen zeigt an, wie viele Sitzungen keine Verbindung hergestellt haben. Beachten Sie, dass fehlgeschlagene Sitzungseinstellungen NICHT das L2TPTunnelDownPeerUnreachable-Trap auslösen, sondern nur fehlgeschlagene Tunnelkonfigurationen.

Es gibt auch eine Zählerversion des Befehls show l2tp tunnels, die nützlich sein kann.

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

Schließlich können auf Sitzungsebene alle Teilnehmer eines bestimmten Peers angezeigt werden.

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

Die Anzahl der gefundenen Teilnehmer sollte mit der Anzahl der aktiven Sitzungen übereinstimmen, wie bereits besprochen.