L2TP in StarOS - Implementazione su ASR5k e risoluzione dei problemi di peer L2TP - L2TPunnelDownPeerUnreachable

Sommario

Introduzione

Questo documento descrive come il Layer 2 Tunneling Protocol (L2TP) in StarOS viene implementato su ASR5k e come risolvere i problemi di peer L2TP - L2TPunnelDownPeerUnreachable.

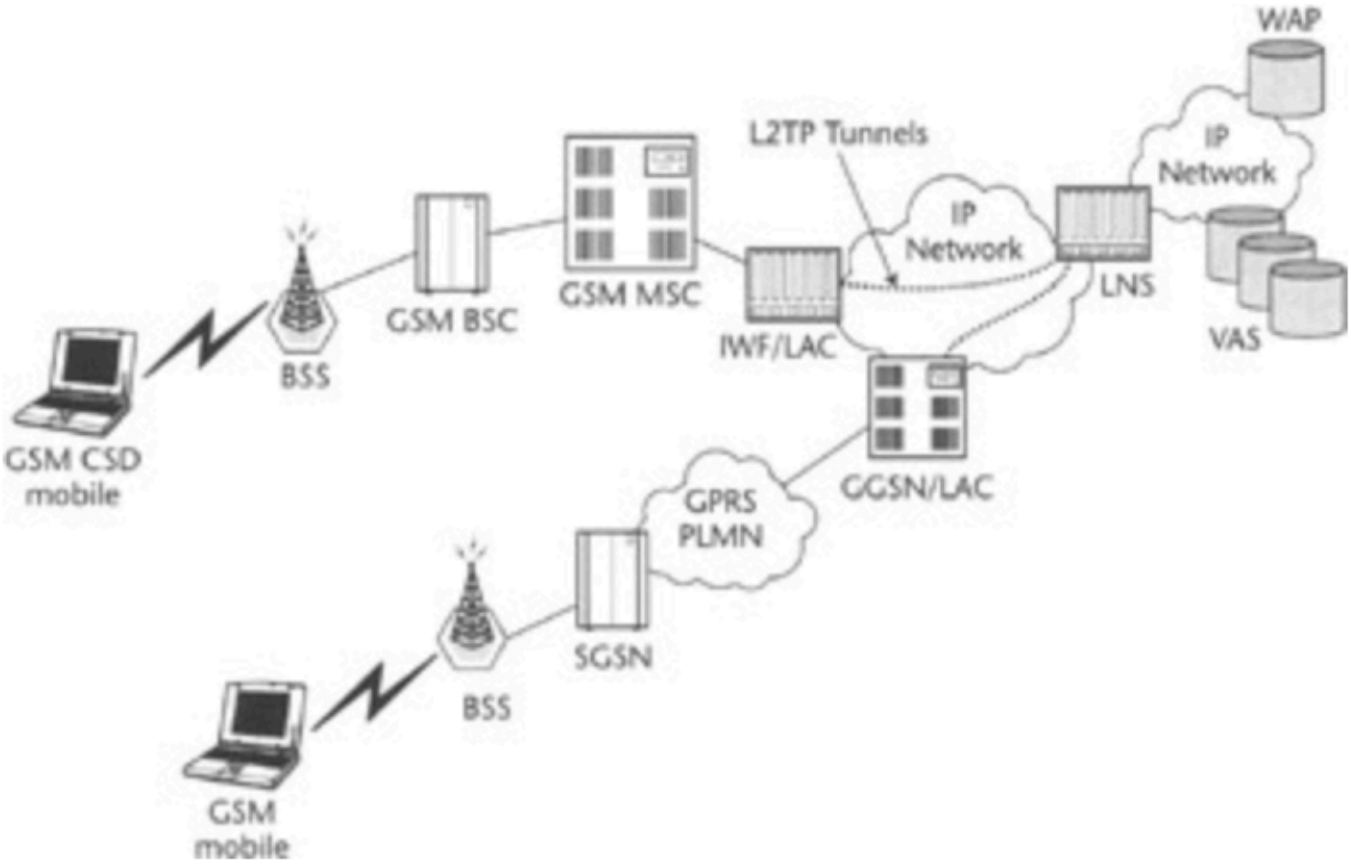
Cos'è L2TP?

L2TP estende la natura point-to-point del PPP. L2TP fornisce un metodo di incapsulamento per la trasmissione di frame PPP tunneling, che consente il tunneling degli endpoint PPP su una rete a commutazione di pacchetto. L2TP viene comunemente implementato in scenari di tipo accesso remoto che utilizzano Internet per offrire servizi di tipo Intranet. Il concetto è quello di rete privata virtuale (VPN).

I due elementi fisici principali di L2TP sono il L2TP Access Concentrator (LAC) e il L2TP Network Server (LNS):

- LAC Il LAC è un peer dell'LNS che agisce come un lato dell'endpoint del tunnel. Il LAC interrompe la connessione PPP remota e si trova tra il telecomando e l'LNS. I pacchetti vengono inoltrati da e verso la connessione remota tramite la connessione PPP. I pacchetti da e verso l'LNS vengono inoltrati attraverso il tunnel L2TP.
- LNS: L'LNS è un peer del LAC che agisce come un lato dell'endpoint del tunnel. L'LNS è il punto di terminazione per le sessioni di tunneling LAC PPP. Questa opzione viene utilizzata

per aggregare le sessioni PPP con tunneling LAC multiple e per accedere alla rete privata. Configurazione L2TP semplificata nella rete mobile, come illustrato in questa immagine.



L2TP utilizza due diversi tipi di messaggi:

- Messaggi di controllo: L2TP trasmette messaggi di dati e di controllo su canali dati e di controllo separati. Il canale di controllo in-band passa messaggi di gestione delle connessioni di controllo in sequenza, gestione delle chiamate, segnalazione degli errori e controllo delle sessioni. L'avvio della connessione di controllo non è specifico del LAC o dell'LNS, ma piuttosto dell'iniziatore e del ricevitore del tunnel che ha rilevanza nella connessione di controllo stabilita. Tra gli endpoint del tunnel viene utilizzato un metodo di autenticazione delle richieste di verifica con segreto condiviso.
- Messaggi dati: I messaggi di dati vengono utilizzati per incapsulare i frame PPP inviati nel tunnel L2TP.

Il flusso di chiamate dettagliato e la creazione del tunnel sono spiegati qui:

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

Dove la usiamo in Mobility?

L'implementazione tipica è per gli utenti aziendali in cui il GSN agisce come LAC e stabilisce tunnel sicuri verso LNS che vengono gestiti nella rete aziendale. I flussi di chiamate dettagliati sono disponibili nell'appendice della guida alla configurazione del GSN, disponibile per versione software specifica, qui:

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

Cos'è ASR5x00 in questa configurazione?

ASR5k può supportare le funzionalità LAC e LNS.

Supporto L2TP LAC

L2TP stabilisce tunnel di controllo L2TP tra LAC e LNS prima di eseguire il tunneling delle connessioni PPP del sottoscrittore come sessioni L2TP. Il servizio LAC si basa sulla stessa architettura del GSN e trae vantaggio dall'allocazione dinamica delle risorse e dall'elaborazione distribuita di messaggi e dati. Questa struttura consente al servizio LAC di supportare oltre 4000 configurazioni al secondo o un massimo di oltre 3G di throughput. In un singolo tunnel possono essere presenti fino a 65535 sessioni e fino a 500.000 sessioni L2TP che utilizzano 32.000 tunnel per sistema.

Supporto L2TP LNS

Il sistema configurato come LNS (Layer 2 Tunneling Protocol Network Server) supporta la terminazione dei tunnel VPN (Virtual Private Network) sicuri tra i controller L2TP Access concentrator (LAC).

L2TP stabilisce tunnel di controllo L2TP tra LAC e LNS prima di eseguire il tunneling delle connessioni PPP del sottoscrittore come sessioni L2TP. In un singolo tunnel possono essere presenti fino a 65535 sessioni e fino a 500.000 sessioni per LAN.

L'architettura LNS è simile al GSN e utilizza il concetto di demultiplexer per assegnare in modo intelligente nuove sessioni L2TP tra le risorse software e hardware disponibili sulla piattaforma senza l'intervento dell'operatore.

Per ulteriori informazioni, consultare le guide alla configurazione di PGW/GSN.

Configurazione per abilitare i servizi sui dispositivi Cisco sull'appliance ASR5k

Esempio di configurazione di LAC su 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
```

Esempio di configurazione di LNS su 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
}
```

> **Nota:** È possibile associare più indirizzi sulla stessa interfaccia IP a servizi LNS diversi. Tuttavia, ogni indirizzo può essere associato a un solo servizio LNS. Inoltre, il servizio LNS non può essere associato alla stessa interfaccia di altri servizi, ad esempio un servizio LAC.

Esempio di configurazione di LNS sul dispositivo Cisco IOS

Può essere utilizzato come esempio di configurazione di supporto per la configurazione di Cisco IOS e non è soggetto a questo articolo.

Configurazione LNS

```
 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
```

Risoluzione dei problemi relativi all'evento Peer Unreachable

In questa sezione vengono fornite alcune linee guida per la risoluzione dei problemi relativi all'evento L2TPTunnelDownPeerUnreachable nella rete. Viene qui spiegato con riferimento al PDN chiuso RP, ma i passaggi per la risoluzione dei problemi sono gli stessi quando si risolvono

problemi con GSN/PGW.

Come promemoria, viene creato un tunnel LAC - LNS in modo da contenere le sessioni del sottoscrittore mentre estende la connessione del sottoscrittore da un PDSN/HA/GSN/PGW all'LNS dove viene terminato e dove viene fornito un indirizzo IP. Se su uno chassis StarOS, l'LNS riceverà un indirizzo IP da un pool IP configurato. Se ci si trova su altri LAN, ad esempio presso la sede del cliente, l'indirizzo IP viene fornito dall'LNS. In quest'ultimo caso, ciò potrebbe effettivamente consentire agli utenti di connettersi alla propria rete domestica tramite un LAC in esecuzione su un partner di roaming.

Viene prima creato un tunnel LAC LNS come prima sessione del destinatario che viene tentata la configurazione e rimarrà attivo finché ci sono sessioni nel tunnel.

Al termine dell'ultima sessione per un determinato tunnel, il tunnel viene chiuso o arrestato. È possibile stabilire più tunnel tra gli stessi peer LAC-LNS.

Di seguito viene riportato un frammento di output del comando **show l2tp tunnel all** che mostra questo caso lo chassis che ospita i servizi LAC e LNS (TestLAC e TestLNS). Notare che i tunnel LAC e LNS TUTTE hanno sessioni, mentre alcuni tunnel RP chiusi non hanno sessioni.

```
 [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
```

La configurazione dei servizi può essere visualizzata con

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

Di seguito è riportato un esempio di trap L2TPTunnelDownPeerUnreachable con servizio LAC 1.1.1.2 e servizio LNS (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
```

Ottenere un conteggio del numero di volte in cui la trap è stata attivata (dal ricaricamento o dall'ultima reimpostazione delle statistiche) utilizzando il comando **show snmp trap statistics**

La trap L2TPTunnelDownPeerUnreachable viene attivata per L2TP quando si verifica un timeout di configurazione del tunnel OPPURE i pacchetti keep-alive (Hello) non ricevono risposta. La causa è in genere dovuta al fatto che il peer LNS non risponde alle richieste del LAC o ai problemi di trasporto in entrambe le direzioni.

Non vi è alcuna trappola per indicare che il peer diventa raggiungibile, il che, se non si capisce come indagare ulteriormente, può portare a confusione se c'è ancora un problema al momento

dell'indagine (richiesta di funzionalità presentata).

Per continuare, la parte più importante di cui abbiamo bisogno è l'indirizzo IP del peer. Il primo passo è verificare la presenza di una connettività IP da controllare con il comando PING. In caso di connettività, procedere con i debug

```
****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)
```

Note:

l2tpmgr tiene traccia delle impostazioni di sessioni del sottoscrittore specifiche

l2tp-control track: creazione tunnel:

Di seguito viene riportato un esempio di debug da questo output

Scenario d'uso: Errore di configurazione iniziale del tunnel a causa di timeout dei tentativi

```
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
```

Di seguito viene riportata la trap SNMP risultante attivata per far corrispondere i log indicati per il momento in cui il sistema ha determinato l'errore

```
 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
```

Scenario d'uso: Errore di configurazione iniziale del tunnel a causa di timeout dei tentativi - Analisi

Vediamo che il tunnel arriva alle 16:34 e cerca di mandare la sfida per cinque volte. Sembra che non ci sia risposta e alla fine il tunnel si disconnette.

Esaminare i valori di configurazione predefiniti o configurati e vedere

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

Questa configurazione deve essere interpretata come prima ritrasmissione dopo 1 secondo, poi come aumento esponenziale - raddoppiando ogni volta: 1, 2, 4, 8, 8.

Si noti che il termine max-retransmissions (cinque) include il primo tentativo/trasmissione.
retransmission-timeout-max è il periodo di tempo massimo tra le trasmissioni una volta raggiunto (se) questo limite
retransmission-timeout-first è il punto di partenza del tempo di attesa prima della prima ritrasmissione.

Quindi, facendo i calcoli, nel caso dei parametri predefiniti, un errore si verificherebbe dopo 1+ 2 + 4 + 8 + 8 secondi = 23 secondi, che è visualizzato esattamente come nell'output sottostante.

Scenario d'uso: Errore di configurazione iniziale del tunnel a causa di keepalive

L'altro motivo per cui l'trap L2TPTunnelDownPeerUnreachable non è una risposta ai messaggi keepalive-interval. Questi vengono usati quando non ci sono messaggi di controllo o dati inviati attraverso il tunnel, per assicurarsi che l'altra estremità sia ancora in vita. Se alcune sessioni sono presenti nel tunnel, ma non stanno eseguendo alcuna operazione, questo comando assicura che il tunnel funzioni ancora correttamente, in quanto, abilitandolo, i messaggi keepalive vengono inviati dopo il periodo configurato di non scambio dei pacchetti (ad esempio, 60 secondi), quindi sono previste risposte. La frequenza di invio di keepalive dopo l'invio del primo messaggio e la mancata risposta è la stessa descritta in precedenza per l'impostazione del tunnel. Così, dopo 23 secondi che non ricevo una risposta ai messaggi di saluto (keepalive), il tunnel verrà demolito. Vedere intervallo keepalive configurabile (predefinito = 60 s).

Di seguito sono riportati alcuni esempi di scambi keep-alive riusciti, sia dal sottoscrittore di monitoraggio che dalla registrazione. Si noti l'intervallo di un minuto tra una serie di messaggi e l'altra in seguito alla mancata trasmissione di dati utente per un minuto. Nell'esempio, i servizi LAC e LNS si trovano nello stesso chassis, in contesti denominati rispettivamente **destination** (**destinazione)** e **lns (linea)**.

```
 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)
```

Infine, ecco un esempio di tunnel ESISTENTE a cui non viene data risposta ai messaggi di benvenuto e in cui la chiamata e il tunnel vengono eliminati. Uscita Monitor Subscriber:

```
 <<<<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)
```

Di seguito sono riportati i rispettivi registri.

Notare il timeout del tunnel di controllo dell'output - tentativi cinque, ultimo intervallo 8000 ms per i tentativi non riusciti.

```
 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
```

E trap SNMP corrispondente

```
 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
```

Mostra considerazioni sull'output

L'esecuzione del comando seguente indica se si sono verificati problemi di raggiungibilità del peer con un peer specifico (o per tutti i tunnel in un particolare servizio lac/lns)

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

Il contatore Connessioni attive corrisponde al numero di tunnel esistenti per il peer. È possibile che ce ne sia più di uno, come mostrato nell'output del comando show l2tp tunnel di versioni precedenti.

Il contatore Failed to Connect (Impossibile connettersi) indica quanti errori di installazione del tunnel si sono verificati.
Il contatore Numero massimo tentativi superato è probabilmente il più importante, in quanto indica che la connessione non è riuscita a causa di un timeout (ogni tentativo superato genera una trap L2TPTunnelDownPeerUnreachable). Queste informazioni indicano solo la frequenza del problema per un dato peer, ma non il motivo per cui si è verificato il timeout. Ma conoscere la frequenza può essere utile per mettere insieme i pezzi nel processo complessivo di risoluzione dei problemi.

La sezione Sessioni fornisce i dettagli a livello di sessione del sottoscrittore (rispetto al livello di tunnel)
Il contatore Sessioni attive corrisponde alla somma dell'output della colonna Sessioni attive restituita dalla visualizzazione dei tunnel l2tp del peer specifico (se sono presenti più tunnel per un peer).
Il contatore Failed to Connect (Impossibile connettersi) indica quante sessioni non sono riuscite a connettersi. Si noti che le impostazioni di sessione non riuscite NON attivano la trap L2TPTunnelDownPeerUnreachable, a differenza delle impostazioni di tunnel non riuscite.

Inoltre, è disponibile una versione con contatori del comando show l2tp tunnel che può essere utile.

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

Infine, a livello di sessione, è possibile visualizzare tutti i sottoscrittori di un determinato peer.

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

Il numero di sottoscrittori trovati deve corrispondere al numero di sessioni attive come descritto.