L2TP em StarOS - A aplicação no ASR5k e pesquisa defeitos o L2TP que espreita - L2TPTunnelDownPeerUnreachable

Índice

Introdução

Este documento descreve como o protocolo Layer 2 Tunneling Protocol (L2TP) em StarOS é executado no ASR5k e pesquisa defeitos o L2TP que espreita - L2TPTunnelDownPeerUnreachable.

Que é L2TP?

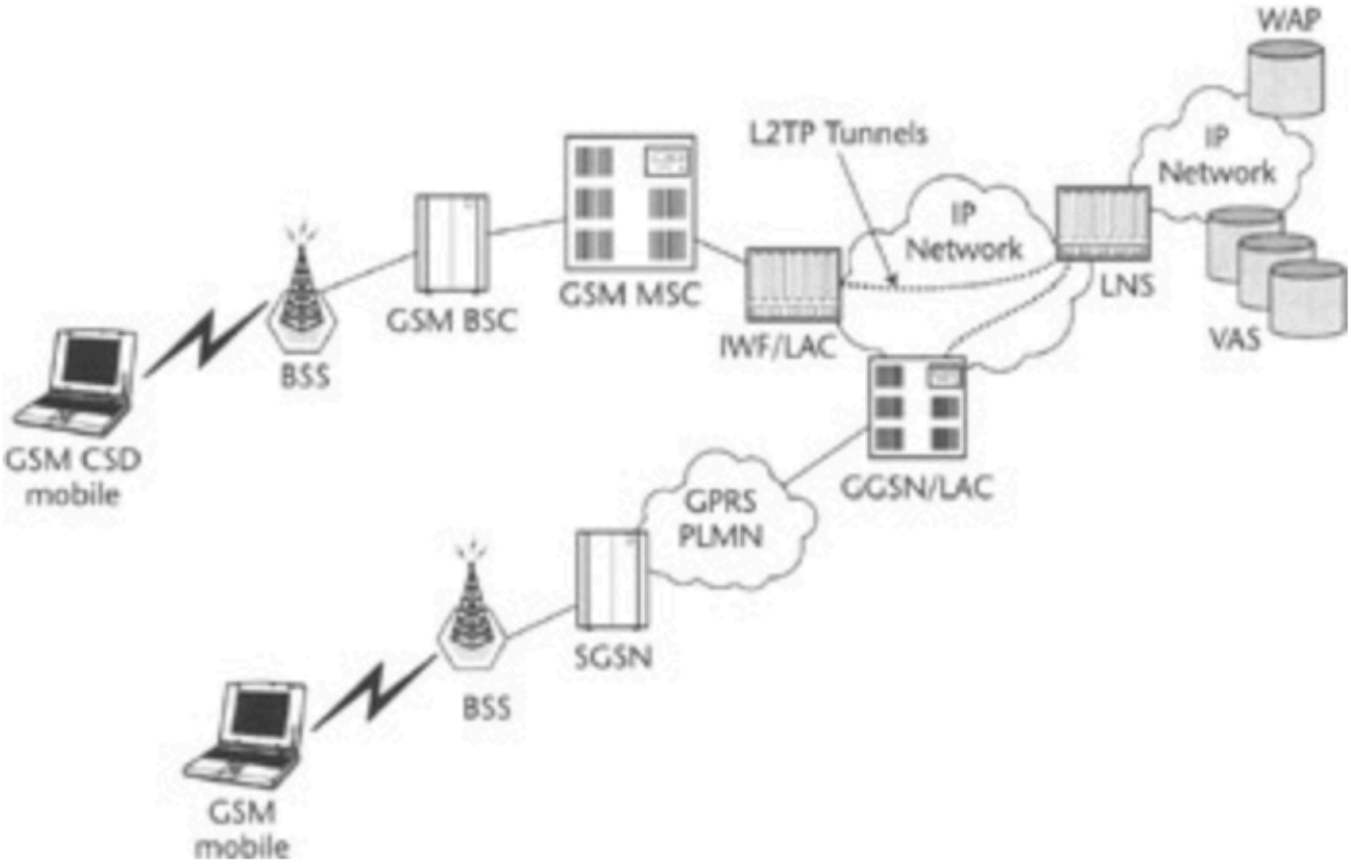
L2TP estende a natureza ponto-a-ponto de PPP. L2TP fornece um método de encapsulamento para a transmissão de frames PPP encapsulados, o que permite aos pontos de extremidade PPP serem encapsulados em uma rede comutada por pacote. L2TP é o mais implantado em cenários do tipo de acesso remoto que utilizam a Internet para oferecer serviços de tipo intranet. O conceito é de uma Rede Virtual Privada (VPN, Virtual Private Network).

Os dois elementos físicos primários de L2TP são o Concentrador de Acesso L2TP (LAC, L2TP Access Concentrator) e o Servidor de Rede L2TP (LNS, L2TP Network Server):

- LAC: O LAC é um par ao LNS que atua como um lado do ponto final de túnel. O LAC termina a conexão PPP remota e passa entre o remoto e o LNS. Os pacotes são encaminhados da e para a conexão remota pela conexão PPP. Os pacotes para e de LNS são encaminhados pelo túnel L2TP.
- LNS: O LNS é um par ao LAC que atua como um lado do ponto final de túnel. O LNS é o ponto de terminação para as sessões encapsuladas LAC PPP. Ele é utilizado para agregar

as várias sessões PPP encapsuladas por LAC e entrar na rede privada.

O L2TP simplificado setup na rede móvel, segundo as indicações desta imagem.



Há dois tipos de mensagem diferentes utilizados por L2TP:

- Mensagens do controle: O L2TP passa o controle e os mensagens de dados sobre o controle e os canais de dados separados. O canal de controle dentro da banda passa pelo gerenciamento de conexão de controle em sequência, pelo gerenciamento de chamadas, pelos relatórios de erro e pelas mensagens de controle de sessão. O início da conexão de controle não é específico do LAC ou do LNS, mas o originador de túnel e o receptor com relevância no estabelecimento da conexão de controle. Um método de autenticação de desafio com segredo compartilhado é utilizado entre os pontos de extremidade do túnel.
- Mensagens de dados: Os mensagens de dados são usados para encapsular os quadros PPP que são enviados no túnel L2TP.

O fluxo de chamadas e o estabelecimento de túnel detalhados são explicados aqui:

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

Onde nós o usamos na mobilidade?

A implementação típica é para os usuários corporativos onde o GGSN atua como o LAC e estabelece túneis seguros para o LNS que é operado na rede corporativa. Os fluxos de chamadas detalhados estão disponíveis no apêndice do manual de configuração GGSN que pode ser encontrado, pela versão de software específica, aqui:

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

Que é ASR5x00 nesta instalação?

ASR5k pode apoiar a funcionalidade LAC e LNS.

Apoio L2TP LAC

O L2TP estabelece túneis do controle L2TP entre o LAC e o LNS antes de escavar um túnel as conexões PPP do subscritor como sessões de L2TP. O serviço LAC é baseado na mesma arquitetura que o GGSN e os benefícios da atribuição dos recursos dinâmicos e mensagem e processo de dados distribuídos. Este projeto permite que o serviço LAC apoie sobre 4000 instalações por segundo ou um máximo de 3G excedente da taxa de transferência. Pode haver um de elevação máxima a 65535 sessões em um único túnel e tanto como como 500,000 sessões de L2TP que usam 32,000 túneis pelo sistema.

Apoio L2TP LNS

O sistema configurado como um servidor de rede do protocolo Layer 2 Tunneling Protocol (LNS) apoia os túneis seguros do Virtual Private Network (VPN) da terminação no meio dos concentradores de acesso L2TP (LAC).

O L2TP estabelece túneis do controle L2TP entre o LAC e o LNS antes de escavar um túnel as conexões PPP do subscritor como sessões de L2TP. Pode haver um máximo de até 65535 sessões em um único túnel e de até 500,000 sessões pelo LNS.

A arquitetura LNS é similar ao GGSN e utiliza o conceito de um desmultiplexador para atribuir inteligentemente sessões de L2TP novas através do software disponível e recursos do hardware na plataforma sem intervenção do operador.

Para mais informação consulte manuais de configuração PGW/GGSN.

Configuração para permitir serviços nos dispositivos Cisco no ASR5k

Exemplo de configuração para o LAC em 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
```

Exemplo de configuração para o LNS em 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: Os endereços múltiplos na mesma interface IP podem ser limitados aos serviços diferentes LNS. Contudo, cada endereço pode ser limitado a somente um serviço LNS. Além, o serviço LNS não pode ser limitado à mesma relação que outros serviços tais como um serviço LAC.

Exemplo de configuração para o LNS no dispositivo IOS Cisco

Isto pode ser usado como um exemplo de configuração de apoio para a configuração do IOS da Cisco e não é sujeito a este artigo.

Configuração de 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
```

Pesquise defeitos o evento inacessível do par

Esta seção dará algumas diretrizes em como pesquisar defeitos o evento L2TPTunnelDownPeerUnreachable na rede. É explicada aqui com referência ao RP fechado PDSN mas as etapas da pesquisa de defeitos são as mesmas ao pesquisar defeitos com GGSN/PGW.

Como um lembrete, um LAC ao túnel LNS está criado a fim conter sessões do subscritor quando estender a conexão do subscritor de um PDSN/HA/GGSN/PGW ao LNS onde está terminado e onde um endereço IP de Um ou Mais Servidores Cisco ICM NT é fornecido. Se em um chassi de StarOS, o LNS obterá um endereço IP de Um ou Mais Servidores Cisco ICM NT de um IP pool configurado. Se em algum outro LNS, por exemplo nas premissas do cliente, o endereço IP de Um ou Mais Servidores Cisco ICM NT é fornecido pelo LNS lá. Na última encenação, isto podia

eficazmente permitir usuários conectar a sua rede home com um LAC que é executado em um sócio vagueando.

Um túnel LAC LNS é criado primeiramente como a primeira sessão do subscritor é tentada ser setup, e ficará acima enquanto há umas sessões no túnel.

Quando a última sessão termina para um túnel dado, esse túnel está fechado ou fechado. Mais de um túnel pode ser estabelecido entre os mesmos pares LAC-LNS.

Está aqui um snippet da saída do comando show que **l2tp escava um túnel tudo** que mostra que isto neste caso o chassi hospeda serviços LAC e LNS (TestLAC e TestLNS). Note que TODOS o LAC e o LNS escavam um túnel têm sessões, quando alguns túneis fechados RP não tiverem nenhuma sessão.

```
 [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
```

A configuração dos serviços pode ser vista com

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

Está aqui um exemplo da armadilha L2TPTunnelDownPeerUnreachable com serviço 1.1.1.2 LAC e serviço LNS (par) 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
```

Obtenha uma contagem de quantas vezes esta armadilha foi provocada (desde que reload ou última restauração das estatísticas) usando as **estatísticas da armadilha do** comando show snmp

A armadilha L2TPTunnelDownPeerUnreachable está provocada para o L2TP quando um intervalo da configuração do túnel ocorre OU pacotes da manutenção de atividade (olá!) não está respondida a. A causa é geralmente devido ao par LNS que não responde aos pedidos do LAC ou às edições do transporte em um ou outro sentido.

Não há nenhuma armadilha para indicar que o par se torna alcançável, que, se não se compreende como investigar mais, pode conduzir à confusão se há ainda uma edição ou não na altura da investigação (pedido da característica submetido).

Para continuar, a maioria de parte importante que nós precisamos é o endereço IP do peer. A primeira etapa é assegurar-se de que haja a conectividade IP que pode ser verificada com o PING. Se há uma Conectividade você pode continuar com debuga

```
****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)
```

Notas:

l2tpmgr segue a instalação específica da sessão do subscritor

l2tp-control segue o estabelecimento de túnel:

Está aqui a amostra debuga desta saída

Caso do uso: Configuração do túnel inicial falha devido para experimentar de novo intervalos

```
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
```

Está aqui a armadilha de SNMP resultante provocada para combinar os logs acima no momento que o sistema determinou a falha

```
 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
```

Caso do uso: Configuração do túnel inicial falha devido para experimentar de novo intervalos - Análise

O que nós vemos é que o túnel vem acima em 16:34 e tenta enviar o desafio por cinco vezes. Aparentemente, não há nenhuma resposta e eventualmente as disconexões do túnel.

Olhe nos padrões ou nos valores configurados da configuração e veja

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

Esta configuração deve ser interpeted como retransmite primeiramente após 1 segundo, então aumento exponencial - dobrando cada vez: 1, 2, 4, 8, 8.

Note as MAX-retransmissões do termo (cinco) inclui a primeiras tentativa/transmissão.
o retransmissão-intervalo-MAX é quantidade máxima de tempo entre transmissões após (se) este limite é alcançado
o retransmissão-intervalo-primeiro é o ponto de início de quanto tempo esperar antes da primeira retransmissão.

Assim, fazendo a matemática, no caso dos parâmetros padrão, uma falha ocorreria após 1+ 2 + 4 + 8 + 8 segundos = 23 segundos, que é visto exatamente como na saída abaixo.

Caso do uso: Configuração do túnel inicial falha devido ao Keepalives

A outra razão para a armadilha L2TPTunnelDownPeerUnreachable não é nenhuma resposta às mensagens do intervalo keepalive. Estes são usados durante os períodos onde não há nenhum mensagem ou dados do controle que estão sendo enviados sobre o túnel, para assegurar-se de que a outra extremidade esteja ainda viva. Se há umas sessões no túnel, mas não estão fazendo qualquer coisa, este comando assegura-se de que o túnel ainda esteja funcionando corretamente, porque permitindo o, os mensagens de keepalive são enviados após o período configurado de nenhum intercâmbio de pacotes (isto é 60 segundos), e as respostas são esperadas. A frequência de enviar o keepalive após ter enviado primeiro e não ter obtido uma resposta é a mesma como descrito acima para a configuração do túnel. Assim, após 23 segundos de não receber uma resposta olá! às mensagens (do keepalive), o túnel será rasgado para baixo. Veja o intervalo keepalive configurável (padrão = 60s).

Estão aqui os exemplos da troca bem sucedida da manutenção de atividade, ambos do subscritor do monitor e do registro. Note o intervalo de um minuto entre grupos de mensagens em

consequência de nenhuns dados do usuário que estão sendo transmitidos para um minuto. Neste exemplo, os serviços LAC e LNS são situados no mesmo chassi, nos contextos nomeados **destino** e **lns** respectivamente.

```
 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)
```

Finalmente, está aqui um exemplo a onde, para um túnel existente, os mensagens Hello Messages não sejam respondidos, e o atendimento e o túnel sejam rasgados para baixo. Monitore o subscritor output:

```
 <<<<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)
```

Estão aqui os logs respectivos.

Note o intervalo do túnel do controle emissor - cinco novo-tentados, Senhora do último-intervalo 8000 para as falhas de tentativa.

```
 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 armadilha de SNMP correspondente

```
 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
```

Mostre considerações da saída

Executar o comando seguinte indicará se houve as edições da alcançabilidade de peer com um par específico (ou para todos os túneis em um serviço particular da laca/lns)

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

As conexões ativa opõem fósforos que o número de túneis existentes para esse par lá pode ser mais de se, como visto na saída da mostra l2tp escava um túnel tudo de mais cedo.

Não é conectado contra indicará quantas falhas da configuração do túnel ocorreram.
A nova tentativa máxima excedida contra é provavelmente o contador o mais importante, porque indica a falha conectar devido a um intervalo (cada nova tentativa excedida conduz a uma armadilha L2TPTunnelDownPeerUnreachable). Esta informação di-lo somente que a frequência do problema para um par dado, ele não lhe diz porque o intervalo ocorreu. Mas conhecer a frequência pode ser útil em unir as partes no processo de Troubleshooting total.

A seção das sessões dá o detalhe a nível da sessão do subscritor (contra o nível do túnel)
As sessões ativa opõem fósforos que a soma (se mais de um túnel para um par) da saída ativa da coluna de Sess da mostra l2tp escava um túnel para o peer particular.
Não é conectado contra indica quantas sessões não conectaram. Note que as instalações falhadas da sessão não provocam a armadilha L2TPTunnelDownPeerUnreachable, simplesmente as configurações do túnel falhadas fazem.

Há igualmente contadores que a versão dos túneis da mostra l2tp comanda que podem ser úteis.

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

Finalmente, a nível da sessão, todos os assinantes para um par dado podem ser vistos.

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

O número de assinantes encontrados deve combinar o número de sessões ativa como discutido.