L2TP dans StarOS - Mise en oeuvre sur ASR5k et dépannage de l'appairage L2TP - L2TPTunnelDownPeerUnreachable

Contenu

Introduction
Qu’est-ce que L2TP ?
Où l'utilisons-nous dans la mobilité ?
Qu'est-ce que ASR5x00 dans cette configuration ?
Prise en charge LAC L2TP
Prise en charge LNS L2TP
Configuration pour activer les services sur les périphériques Cisco sur l'ASR5k
Exemple de configuration pour LAC sur ASR5k
Exemple de configuration pour LNS sur ASR5k
Exemple de configuration de LNS sur le périphérique Cisco IOS
Dépanner un événement homologue inaccessible
Cas d’utilisation : Échec de la configuration initiale du tunnel en raison des délais d'attente des nouvelles tentatives
Cas d’utilisation : Échec de la configuration initiale du tunnel en raison de keepalives
Afficher les considérations de sortie

Introduction

Ce document décrit comment le protocole L2TP (Layer 2 Tunneling Protocol) dans StarOS est mis en oeuvre sur ASR5k et dépannage de l'appairage L2TP - L2TPTunnelDownPeerUnreachable.

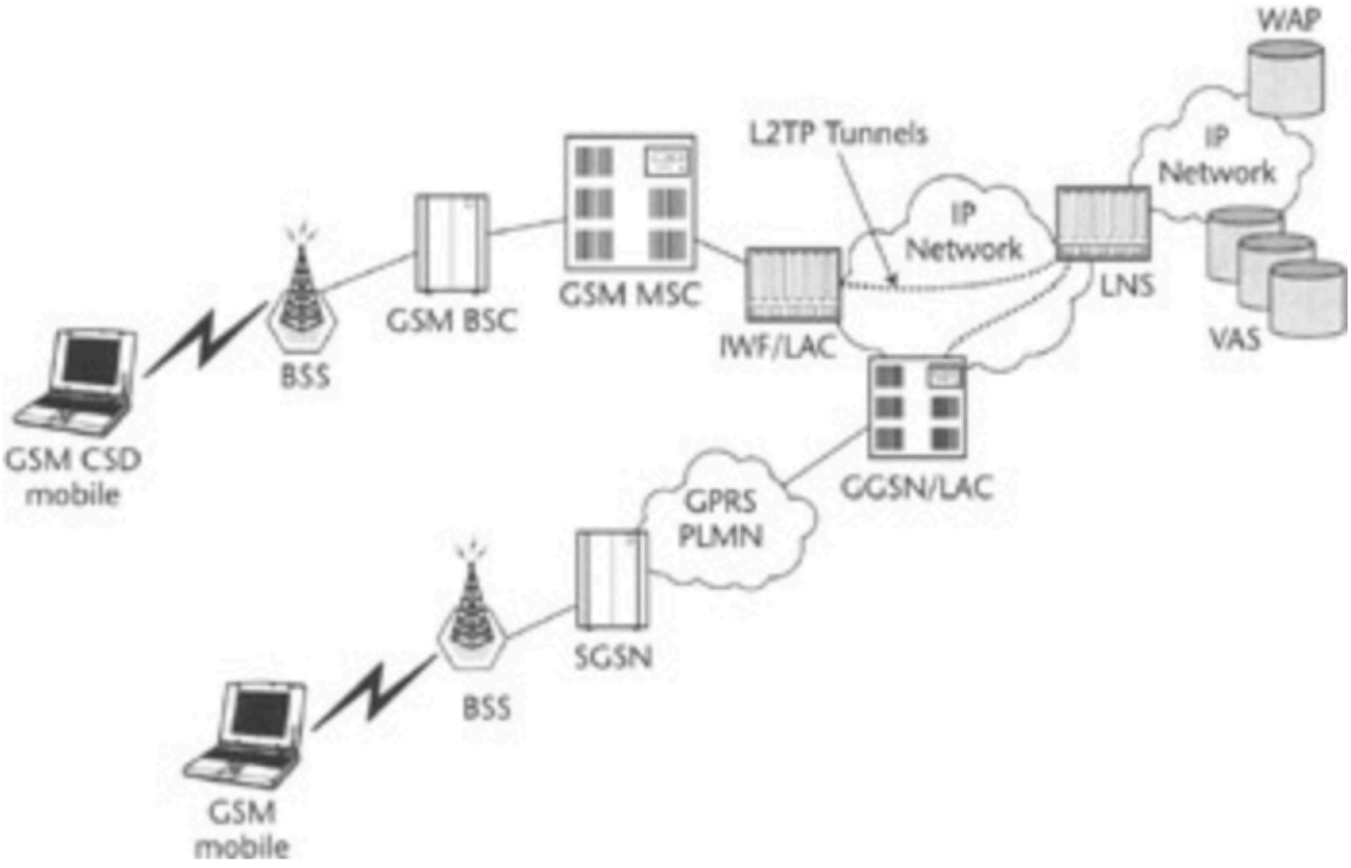
Qu’est-ce que L2TP ?

L2TP étend la nature point à point du protocole PPP. L2TP fournit une méthode d’encapsulation pour la transmission de trames PPP à canaux, qui permet aux points d’extrémité PPP d’être tunnellisés sur un réseau à commutation de paquets. Le protocole L2TP est le plus souvent déployé dans des scénarios de type accès distant qui utilisent Internet pour offrir des services de type intranet. Le concept est celui d'un réseau privé virtuel (VPN).

Les deux principaux éléments physiques de L2TP sont le concentrateur d’accès L2TP (LAC) et le serveur de réseau L2TP (LNS) :

- BAC : Le LAC est un homologue du LNS qui agit comme un côté du point d'extrémité du tunnel. Le LAC met fin à la connexion PPP distante et se trouve entre le distant et le LNS. Les paquets sont transférés vers et depuis la connexion distante via la connexion PPP. Les paquets en provenance et à destination du LNS sont transférés via le tunnel L2TP.
- LNS : Le LNS est un homologue du LAC qui agit comme un côté du point d'extrémité du tunnel. Le LNS est le point de terminaison des sessions en tunnel PPP de LAC. Il est utilisé

pour agréger les sessions PPP multicanaux de LAC et pour entrer dans le réseau privé. Configuration L2TP simplifiée dans le réseau mobile, comme illustré dans cette image.



Il existe deux types de message différents que L2TP utilise :

- Messages de contrôle : L2TP transmet les messages de contrôle et de données sur des canaux de contrôle et de données distincts. Le canal de contrôle intrabande transmet les messages de gestion de connexion de contrôle séquencé, de gestion des appels, de rapport d'erreurs et de contrôle de session. L'initialisation de la connexion de contrôle n'est pas spécifique au LAC ou au LNS, mais plutôt à l'émetteur et au récepteur du tunnel qui a de la pertinence dans l'établissement de la connexion de contrôle. Une méthode d'authentification par défi à secret partagé est utilisée entre les points d'extrémité du tunnel.
- Messages de données : Les messages de données sont utilisés pour encapsuler les trames PPP envoyées dans le tunnel L2TP.

L'établissement détaillé du flux d'appels et du tunnel est expliqué ici :

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

Où l'utilisons-nous dans la mobilité ?

Le déploiement type est destiné aux utilisateurs d'entreprise où le GGSN agit comme LAC et établit des tunnels sécurisés vers LNS qui sont exploités dans le réseau d'entreprise. Des flux d'appels détaillés sont disponibles dans l'annexe du guide de configuration GGSN qui se trouve, par version logicielle spécifique, ici :

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

Qu'est-ce que ASR5x00 dans cette configuration ?

ASR5k peut prendre en charge les fonctionnalités LAC et LNS.

Prise en charge LAC L2TP

L2TP établit des tunnels de contrôle L2TP entre LAC et LNS avant de tunneliser les connexions PPP de l'abonné en tant que sessions L2TP. Le service BAC est basé sur la même architecture que le GGSN et bénéficie d'une allocation dynamique des ressources et d'un traitement distribué des messages et des données. Cette conception permet au service LAC de prendre en charge plus de 4 000 configurations par seconde ou un débit maximum de plus de 3G. Il peut y avoir jusqu'à 65 535 sessions dans un tunnel unique et jusqu'à 500 000 sessions L2TP utilisant 32 000 tunnels par système.

Prise en charge LNS L2TP

Le système configuré en tant que serveur LNS (Layer 2 Tunneling Protocol Network Server) prend en charge les tunnels VPN sécurisés de terminaison entre les concentrateurs d'accès L2TP (LAC).

L2TP établit des tunnels de contrôle L2TP entre LAC et LNS avant de tunneliser les connexions PPP de l'abonné en tant que sessions L2TP. Il peut y avoir jusqu'à 65 535 sessions dans un tunnel unique et jusqu'à 500 000 sessions par LNS.

L'architecture LNS est similaire au GGSN et utilise le concept d'un démultiplexeur pour attribuer intelligemment de nouvelles sessions L2TP sur les ressources logicielles et matérielles disponibles sur la plate-forme sans intervention de l'opérateur.

Pour plus d'informations, reportez-vous aux guides de configuration PGW/GGSN.

Configuration pour activer les services sur les périphériques Cisco sur l'ASR5k

Exemple de configuration pour LAC sur 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
```

Exemple de configuration pour LNS sur 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
}
```

> **Note**: Plusieurs adresses sur la même interface IP peuvent être liées à différents services LNS. Cependant, chaque adresse peut être liée à un seul service LNS. En outre, le service LNS ne peut pas être lié à la même interface que d'autres services tels qu'un service LAC.

Exemple de configuration de LNS sur le périphérique Cisco IOS

Ceci peut être utilisé comme exemple de configuration de prise en charge pour la configuration de Cisco IOS et n'est pas soumis à cet article.

Configuration 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
```

Dépanner un événement homologue inaccessible

Cette section donne quelques directives sur la façon de dépanner l'événement L2TPTunnelDownPeerUnreachable dans le réseau. Il est expliqué ici en référence au RP fermé PDSN, mais les étapes de dépannage sont les mêmes lors du dépannage avec GGSN/PGW.

Àtitre de rappel, un tunnel LAC vers LNS est créé afin de contenir les sessions d'abonnés pendant qu'il étend la connexion d'abonné d'un PDSN/HA/GGSN/PGW au LNS où il est terminé et où une adresse IP est fournie. S'il se trouve sur un châssis StarOS, le LNS obtient une adresse IP à partir d'un pool d'adresses IP configuré. Si sur un autre LNS, par exemple dans les locaux du client, l'adresse IP est fournie par le LNS. Dans ce dernier scénario, cela pourrait permettre aux utilisateurs de se connecter à leur réseau domestique via un LAC exécuté sur un partenaire d'itinérance.

Un tunnel LNS LAC est d'abord créé lorsque la première session d'abonné est tentée d'être configurée et reste active tant qu'il y a des sessions dans le tunnel.

Lorsque la dernière session se termine pour un tunnel donné, ce tunnel est fermé ou arrêté. Plusieurs tunnels peuvent être établis entre les mêmes homologues LAC-LNS.

Voici un extrait du résultat de la commande **show l2tp tunnels all** qui montre ceci dans ce cas le châssis héberge à la fois les services LAC et LNS (TestLAC et TestLNS). Notez que les tunnels LAC et LNS ALL ont des sessions, tandis que certains tunnels RP fermés n'ont aucune session.

```
 [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 configuration des services peut être vue avec

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

Voici un exemple du piège L2TPTunnelDownPeerUnreachable avec le service LAC 1.1.1.2 et le service LNS (homologue) 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
```

Récupère le nombre de fois que ce déroutement a été déclenché (depuis le rechargement ou la dernière réinitialisation des statistiques) à l'aide de la commande **show snmp trap statistics**

Le déroutement L2TPTunnelDownPeerUnreachable est déclenché pour L2TP lorsqu'un délai d'installation du tunnel se produit OU que des paquets de maintien en vie (Hello) ne reçoivent pas de réponse. La cause est généralement due au fait que l'homologue LNS ne répond pas aux demandes de BAC ou aux problèmes de transport dans les deux directions.

Il n'y a pas de piège pour indiquer que l'homologue devient accessible, ce qui, s'il n'est pas compris comment enquêter plus loin, peut conduire à la confusion quant à savoir s'il existe encore un problème ou non au moment de l'enquête (demande de fonction soumise).

Pour continuer, la partie la plus importante dont nous avons besoin est l'adresse IP de l'homologue. La première étape consiste à s'assurer que la connectivité IP peut être vérifiée avec PING. Si la connectivité existe, vous pouvez poursuivre les débogages

```
****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)
```

Remarques :

l2tpmgr suit la configuration de la session d'abonné spécifique

l2tp-control trace l'établissement du tunnel :

Voici un exemple de débogage de cette sortie

Cas d'utilisation : Échec de la configuration initiale du tunnel en raison des délais d'attente des nouvelles tentatives

```
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
```

Voici le déroutement SNMP résultant déclenché pour correspondre aux journaux ci-dessus au moment où le système a déterminé la défaillance

```
 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
```

Cas d'utilisation : Échec de la configuration initiale du tunnel en raison des délais d'attente des nouvelles tentatives - Analyse

Ce que nous voyons, c'est que le tunnel arrive à 16h34 et il tente d'envoyer le défi cinq fois. Apparemment, il n'y a pas de réponse et finalement le tunnel se déconnecte.

Examinez les valeurs de configuration par défaut ou configurées et reportez-vous à la section

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

Cette configuration doit être interprétée comme une première retransmission après 1 seconde, puis une augmentation exponentielle - doublant à chaque fois : 1, 2, 4, 8, 8.

Notez que le terme max-retransmissions (cinq) inclut la première tentative/transmission.
retransmission-timeout-max est la durée maximale entre les transmissions après (si) cette limite est atteinte
retransmission-timeout-first est le point de départ du délai d'attente avant la première retransmission.

Donc, en faisant le calcul, dans le cas des paramètres par défaut, une défaillance se produirait après 1+ 2 + 4 + 8 + 8 secondes = 23 secondes, qui est vu exactement comme dans le résultat ci-dessous.

Cas d’utilisation : Échec de la configuration initiale du tunnel en raison de keepalives

L'autre raison du déroutement L2TPTunnelDownPeerUnreachable n'est pas une réponse aux messages keepalive-interval. Ils sont utilisés pendant les périodes où aucun message de contrôle ou données n'est envoyé par le tunnel, pour s'assurer que l'autre extrémité est toujours en vie. S'il y a des sessions dans le tunnel, mais qu'elles ne font rien, cette commande garantit que le tunnel fonctionne toujours correctement, car en l'activant, les messages de test d'activité sont envoyés après la période configurée d'absence d'échange de paquets (c'est-à-dire 60 secondes), et les réponses sont attendues. La fréquence d'envoi du keepalive après l'envoi du premier et l'absence de réponse est la même que celle décrite ci-dessus pour la configuration du tunnel. Ainsi, après 23 secondes après ne pas recevoir de réponse aux messages Hello (keepalive), le tunnel sera désactivé. Voir intervalle de conservation configurable (par défaut = 60).

Voici des exemples d'échanges de maintien en vie réussis, à partir de l'abonné de surveillance et de la journalisation. Notez l'intervalle d'une minute entre les jeux de messages en raison de l'absence de transmission de données utilisateur pendant une minute. Dans cet exemple, les services LAC et LNS sont situés dans le même châssis, dans des contextes nommés **destination** et **lns** respectivement.

```
 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)
```

Enfin, voici un exemple où, pour un tunnel EXISTANT, les messages Hello ne reçoivent pas de réponse et l'appel et le tunnel sont désactivés. Sortie de l'abonné moniteur :

```
 <<<<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)
```

Voici les journaux respectifs.

Notez le délai d'attente du tunnel de contrôle de sortie - tentative de nouvelle tentative de cinq, dernier intervalle de 8 000 ms pour les tentatives échouées.

```
 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
```

Et déroutement SNMP correspondant

```
 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
```

Afficher les considérations de sortie

L'exécution de la commande suivante indique s'il y a eu des problèmes d'accessibilité des homologues avec un homologue spécifique (ou pour tous les tunnels d'un service lac/ln

particulier).

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

Le compteur Connexions actives correspond au nombre de tunnels existants pour cet homologue, il peut y en avoir plusieurs, comme le montre le résultat de show l2tp tunnels tout à partir de plus tôt.

Le compteur Echec de connexion indique le nombre d'échecs de configuration de tunnel survenus.
Le compteur Max Retry Exceeded est probablement le compteur le plus important, car il indique un échec de connexion en raison d'un délai d'attente (chaque nouvelle tentative a pour résultat un déroutement L2TPTunnelDownPeerUnreachable). Ces informations vous indiquent uniquement la fréquence du problème pour un homologue donné, mais ne vous indiquent pas pourquoi le délai d'attente s'est produit. Mais connaître la fréquence peut être utile pour assembler les éléments dans le processus de dépannage global.

La section Sessions fournit des détails au niveau de la session de l'abonné (par rapport au niveau du tunnel)
Le compteur Sessions actives correspond à la somme (si plusieurs tunnels pour un homologue) de la sortie de colonne Sessions actives de la commande show l2tp tunnels pour l'homologue particulier.
Le compteur Échec de connexion indique le nombre de sessions qui n'ont pas pu se connecter. Notez que les échecs de configuration de session ne déclenchent PAS le piège L2TPTunnelDownPeerUnreachable, seuls les échecs de configuration de tunnel le font.

Il existe également une version counters de la commande show l2tp tunnels qui peut être utile.

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

Enfin, au niveau de la session, tous les abonnés d'un homologue donné peuvent être affichés.

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

Le nombre d'abonnés trouvés doit correspondre au nombre de sessions actives tel que discuté.