# L2TP в StarOS - реализации на ASR5k и L2TP устранения неполадок, взаимодействующем - L2TPTunnelDownPeerUnreachable

## Содержание



## Введение

Этот документ описывает, как протокол туннелирования на уровне 2 (L2TP) в StarOS внедрен на ASR5k и L2TP Устранения неполадок, Взаимодействующем - L2TPTunnelDownPeerUnreachable.

## Что такое L2TP?

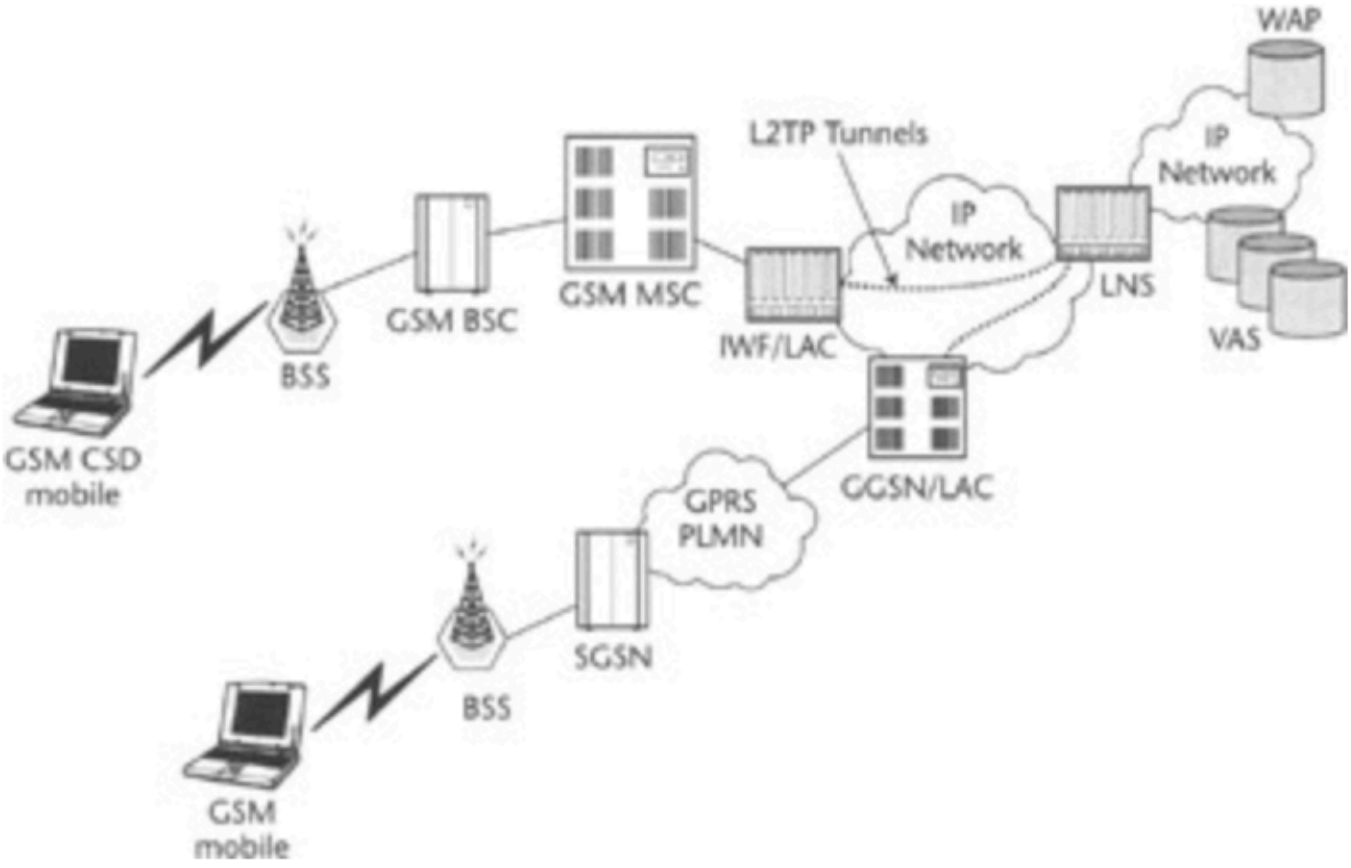
L2TP расширяет природу "точка-точка" PPP. L2TP предоставляет метод инкапсуляции для передачи туннелируемых кадров PPP, которая позволяет оконечным точкам PPP быть туннелированными по сети с коммутацией пакетов. L2TP обычно развернут в сценариях удаленного типа доступа, которые используют Интернет для предложения услуг типа интранет. Понятие является понятием Виртуальной частной сети (VPN).

Двумя основными физическими элементами L2TP является Концентратор доступа L2TP (LAC) и L2TP Network Server (LNS):

- LAC: LAC является узлом к LNS, который действует как одна сторона оконечной точки туннеля. LAC завершает удаленное PPP-соединение и сидит между удаленным узлом и LNS. Пакеты переданы к и от удаленного соединения по PPP - подключению. Пакеты к и от LNS переданы по туннелю L2TP.
- LNS: LNS является узлом к LAC, который действует как одна сторона оконечной точки

туннеля. LNS является оконечной точкой соединения для Туннелируемых сеансов LAC PPP. Это необходимо для объединения множества LAC-тунеллированных PPP-сеансов и входа в частную сеть.

Упрощенная настройка L2TP в Сети мобильной связи, как показано в этом образе.




Существует два других типа сообщения, которые использует L2TP:

- Управляющие сообщения – L2TP передает управляющие сообщения и сообщения о данных через отдельные каналы управления и данных. Внутриполосный управляющий канал передает упорядоченное управление контрольного соединения, управление вызовом, предоставление отчетов об ошибках и управляющие сообщения сеанса. Инициирование контрольного соединения не является определенным или для LAC или для LNS, но, скорее туннельный инициатор и получатель, который имеет уместность в установлении контрольного соединения. Способ аутентификации запроса общего секрета используется между конечными точками туннеля.
- Data messages – используются для инкапсуляции кадров PPP, посланных в туннель L2TP.

Подробный поток вызовов и установка туннеля объяснены здесь:

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

## Где мы используем его в Мобильности?

Типичное развертывание для корпоративных пользователей, где GGSN действует как LAC и устанавливает безопасные туннели к LNS, которым управляют в корпоративной сети.

Подробные диаграммы вызовов доступны в приложении руководства по конфигурации GGSN, которое может быть найдено, на определенную версию программного обеспечения, здесь:

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

# Что такое ASR5x00 в этой настройке?

ASR5k может поддержать функциональность LNS и LAC.

## Поддержка LAC L2TP

L2TP устанавливает туннели контроля за L2TP между LAC и LNS прежде, чем туннелировать PPP - подключения абонента как Сеансы l2tp. Сервис LAC основывается на той же архитектуре как GGSN и извлекает выгоду из выделения динамического ресурса и распределенного сообщения и обработки данных. Этот дизайн позволяет сервису LAC поддерживать более чем 4000 настроек в секунду или максимум по 3G пропускной способности. Может быть максимум до 65535 сеансов в одиночном туннеле и целых 500,000 Сеансов l2tp с помощью 32,000 туннелей на систему.

## Поддержка LNS L2TP

Система, настроенная как Сервер сети Протокола туннелирования Уровня 2 (LNS), поддерживает безопасные туннели Виртуальной частной сети (VPN) завершения между от Концентраторов доступа L2TP (LAC).

L2TP устанавливает туннели контроля за L2TP между LAC и LNS прежде, чем туннелировать PPP - подключения абонента как Сеансы l2tp. Может быть максимум до 65535 сеансов в одиночном туннеле и до 500,000 сеансов на LNS.

Архитектура LNS подобна GGSN и использует понятие демультиплексора для интеллектуального присвоения новых Сеансов l2tp через доступные ресурсы программного и аппаратного обеспечения на платформе без вмешательства оператора.

Для получения дополнительной информации обратитесь руководства по конфигурации PGW/GGSN.

# Конфигурация для включения сервисов на устройствах Cisco на ASR5k

## Пример конфигурации для LAC на 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
```

## Пример конфигурации для LNS на 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
}
```

> **Примечание:** Множественные адреса на том же IP - интерфейсе могут быть связаны с другими сервисами LNS. Однако каждый адрес может быть связан только с одним сервисом LNS. Кроме того, сервис LNS не может быть связан с тем же интерфейсом как другие сервисы, такие как сервис LAC.

## Пример конфигурации для LNS на устройстве Cisco IOS

Это может использоваться в качестве примера конфигурации поддержки для Конфигурации Cisco IOS и не подвергается этой статье.

Конфигурация 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
```

# Одноранговое событие Unreachable устранения неполадок

Этот раздел даст некоторые рекомендации по тому, как устранить неполадки события L2TPTunnelDownPeerUnreachable в сети. Это объяснено здесь в отношении закрытого RP PDSN, но шаги устранения неполадок являются тем же при устранении проблем с GGSN/PGW.

Как напоминание, LAC в туннель LNS создан для содержания сеансов абонента, в то время

как это расширяет абонентское соединение от PDSN/HA/GGSN/PGW до LNS, где это завершено и где предоставлен IP-адрес. Если на шасси StarOS, LNS получит IP-адрес от настроенного пула IP. Если на некотором другом LNS, например в абонентском оборудовании, IP-адрес предоставлен LNS там. В последнем сценарии это могло эффективно обеспечить пользователей для соединения с их домашней сетью через LAC, работающий на бродящем партнере.

Туннель LNS LAC сначала создан, поскольку первый сеанс абонента предпринят, чтобы быть настройкой и не ляжет спать, пока существуют сеансы в туннеле.

Когда последние концы сеанса для данного туннеля, тот туннель закрыт или закрыт. Несколько туннелей могут быть установлены между теми же узлами LAC-LNS.

Вот является фрагмент выходных данных от **show l2tp tunnel** команды **всем**, что показывает это в этом случае, шасси размещает и LAC и сервисы LNS (TestLAC и TestLNS). Обратите внимание на то, что LAC и туннели LNS, ALL имеет сеансы, в то время как некоторые Закрытые туннели RP не имеют никаких сеансов.

```
 [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
```

Конфигурация сервисов может быть просмотрена с

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

Вот пример trap-сообщения L2TPTunnelDownPeerUnreachable с сервисом LAC 1.1.1.2 и сервисом LNS (узел) 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
```

Получите количество того, сколько раз это trap-сообщение было инициировано (так как перезагружаются или последний сброс статистики), использование **статистики команды show snmp trap**

Trap-сообщение L2TPTunnelDownPeerUnreachable инициировано для L2TP, когда таймаут настройки туннеля происходит поддержка активности OR (Hello), на пакеты не отвечают. Причина обычно происходит из-за узла LNS не отвечающие запросы от LAC или транспортных проблем в любом направлении.

Нет никакого trap-сообщения, чтобы указать, что узел становится достижимым, который, если не подразумевается, как заняться расследованиями далее, может привести к беспорядку относительно того, существует ли все еще проблема или не во время расследования (отправленный запрос новых функций).

Для перехода большая часть важной части, в которой мы нуждаемся, является IP - адресом

адресуемым точки. Первый шаг должен гарантировать, что существует возможность подключения с помощью IP-адреса, которая может быть проверена с PING. Если существует подключение , можно продолжить отладки

```
****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)
```

Примечания:

l2tpmgr отслеживает определенную настройку сеанса абонента

l2tp-контроль отслеживает установку туннеля:

Вот пример отладки от этих выходных данных

## Вариант использования: Начальный сбой из-за настройки туннеля ко временам ожидания перед повтором

```
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
```

Вот результирующее trap-сообщение SNMP, инициированное для соответствия с вышеупомянутыми журналами в настоящий момент, система определила сбой

```
 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
```

### Вариант использования: Начальный сбой из-за настройки туннеля ко временам ожидания перед повтором - Анализ

То, что мы видим, - то, что туннель подходит в 16:34, и он пытается передать проблему в течение пяти раз. Очевидно, нет никакого ответа и в конечном счете туннельных разъединений.

Изучите настройки по умолчанию конфигурации или установленные значения и посмотрите

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

Эта конфигурация должна быть interpeted, поскольку сначала повторно передают после 1 секунды, затем экспоненциальное увеличение - удваивающий каждый раз: 1, 2, 4, 8, 8.

Обратите внимание, что термин Max. повторные передачи (пять) включает первую попытку/передачу.
(если) этот предел достигнут, Max. тайм-аутом повторной передачи максимальное количество времени между передачами после
таймаут повторной передачи сначала является отправной точкой того, сколько времени ждать перед первой повторной передачей.

Так, делая математику, в случае параметров по умолчанию, сбой произошел бы после 1 + 2 + 4 + 8 + 8 секунд = 23 секунды, который замечен точно как в выходных данных ниже.

## Вариант использования: Начальный сбой из-за настройки туннеля к пакетам Keepalive

Другая причина для trap-сообщения L2TPTunnelDownPeerUnreachable не является никаким ответом на сообщения интервала проверки активности. Они используются в течение периодов, где нет никаких управляющих сообщений или данных, передаваемых по туннелю, чтобы гарантировать, что другой конец все еще жив. Если существуют сеансы в туннеле, но

они ничего не делают, эта команда гарантирует, что туннель все еще работает должным образом, потому что путем включения его, сообщения поддержки активности передаются после того, как настроенный период никакого обмена пакетами (т.е. 60 секунд), и ответы ожидается. Частота передачи поддержки активности после передачи первой и не получения ответа совпадает с описанный выше для настройки туннеля. Так, после 23 секунд не получения ответа на привет (поддержку активности) сообщения будет разъединен туннель. Посмотрите конфигурируемый интервал проверки активности (по умолчанию = 60-е).

Вот примеры успешного обмена поддержки активности, и от абонента монитора и от регистрации. Обратите внимание на интервал одной минуты между наборами сообщений в результате никаких пользовательских данных, передаваемых в течение одной минуты. В данном примере LAC и сервисы LNS расположены в том же шасси в контекстах, названных **назначением** и **lns** соответственно.

```
 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)
```

Наконец, вот пример, где для СУЩЕСТВУЮЩЕГО ТУННЕЛЯ на приветственные сообщения не отвечают, и вызов и туннель разъединены. Выходные данные 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)
```

Вот соответствующие журналы.

Обратите внимание на туннельный таймаут выходного управляющего - предпринятый повторной попыткой пять, последний интервал 8000 мс для неудачных попыток.

```
 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
```

И соответствующее trap-сообщение SNMP

```
 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
```

## Выходные Show факторы

Выполнение следующей команды укажет, были ли проблемы доступности однорангового узла с определенным узлом (или для всех туннелей в определенном lac/lns сервисе)

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

Счетчик Активных соединений совпадает с количеством существующих туннелей для того узла могут быть несколько, как замечено в выходных данных от show l2tp tunnel все от ранее.

Отказавшее для Соединения счетчика укажет, сколько сбоев настройки туннеля произошло.
Превышенный счетчик Максимального числа попыток является, вероятно, самым важным счетчиком, поскольку это указывает на сбой для соединения из-за таймаута (каждая

Повторная попытка превысила результаты в trap-сообщении L2TPTunnelDownPeerUnreachable). Эта информация только говорит вам частоту проблемы для данного узла, это не говорит вам, почему произошел таймаут. Но знание частоты может быть полезным в соединении частей в полном процессе устранения проблем.

Раздел Сеансов дает подробность на сеансовом уровне абонента (по сравнению с туннельным уровнем)
Счетчик Активных сеансов совпадает с суммой (если несколько туннелей для узла) выходные данные столбца Active Sess от show l2tp tunnel для конкретного однорангового узла.
Отказавшее для Соединения счетчика указывает, сколько сеансов было не в состоянии соединяться. Обратите внимание на то, что отказавшие настройки сеанса НЕ инициируют trap-сообщение L2TPTunnelDownPeerUnreachable, только подведенные настройки туннеля делают.

Существует также версия счетчиков команды show l2tp tunnel, которая может быть полезной.

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

Наконец, на сеансовом уровне, все абоненты для данного узла могут быть просмотрены.

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

Количество найденных абонентов должно совпасть с количеством активных сеансов, как обсуждено.