INTERNET DRAFT
draft-townsley-pwe3-l2tpv3-01.txt
Category: Informational
Expires: December 2003

W. M. Townsley
cisco Systems

June 2003

# Pseudowires and L2TPv3

## Status of this Memo



## Copyright Notice



## Abstract

This document provides an overview of the Layer 2 Tunneling Protocol (L2TPv3), presented in a manner which highlights its applicability for Pseudo Wire Emulation Edge to Edge (PWE3). It is intended as a guide for architects, new implementers, and those adding functionality to L2TPv3 in support of new pseudowire types.

Contents



## 1.0 Introduction

L2TPv3 [L2TPv3] defines a protocol for setup, maintenance and transport of individual pseudowires (PWs) across an IP network. A collection of PWs may be employed to provide a full Layer 2 VPN (L2VPN) service for a variety of circuit types types, including Ethernet, Frame Relay, HDLC, PPP, ATM, AAL5, and others.

L2TPv3 consists of a control protocol and associated state machines for dynamic establishment, maintenance and release of PWs, together with the necessary encapsulation to carry multiple PWs between two tunnel endpoints. In most cases, a given PW-type will operate over L2TPv3 with very few additional protocol constructs beyond that defined in [L2TPv3]. When significant additional PW-type-specific protocol constructs are necessary, they should be defined in brief companion documents.

This document focuses solely on how L2TPv3 is used to support individual Pseudowires. Full L2VPN services including various provisioning methods, auto-discovery, etc. is outside the scope of this specific document.

### 1.2 L2TPv2 and L2TPv3

L2TP (Layer 2 Tunneling Protocol) [RFC2661] defines a method for offloading layer 2 PPP frames from a circuit-switched network to a packet-switched network (e.g. IP), while providing emulation of as many of the characteristics of the native point-to-point link as possible.

L2TP as defined in [RFC2661] is sometimes referred to now as "L2TPv2," while the extended version supporting mutliple PW types defined in [L2TPv3] is referred to as "L2TPv3" (for more information on the similarities and differences between L2TPv2 and L2TPv3, see Appendix A). The remainder of this document will refer simply to L2TP in general unless contrasting specific features of L2TPv3 or L2TPv2 which may differ in function.

New L2TPv3 implementations with no intention of supporting PPP over L2TPv2 may ignore RFC 2661 as the L2TPv3 specification is complete on its own. However, there may be advantages to starting with an available L2TPv2 implementation, and specific advice is given for migration and coexistence with L2TPv2 in [L2TPv3].

## 2. L2TP Tunneling Reference Models and Terminology

Section 2 of [L2TPv3] identifies several reference models for Layer 2 tunneling, and refers to L2TP-specific terminology for these. Two important pieces of L2TP terminology to be aware of are the L2TP Access Concentrator (LAC) and L2TP Network Server (LNS). This section provides an overview of two of these reference models and terminology in specific relation to support of PWs for PWE3.

## 2.1 LAC to LAC Reference Model

An L2TP LAC looks like a PWE3 PE. That is, there are "native" circuits physically connected to the equipment, and pseudowires extending from the LAC over a Packet Switched Network (PSN) for these circuits. For L2TPv3, the default PSN is an IP network.

Following is a diagram of the "LAC to LAC" service model defined in L2TPv3. This is very similar to the PWE3 Network Reference Model [PWE3-REQ], though utilizing L2TP-specific nomenclature which abstracts the protocol terminology from the deployment location.

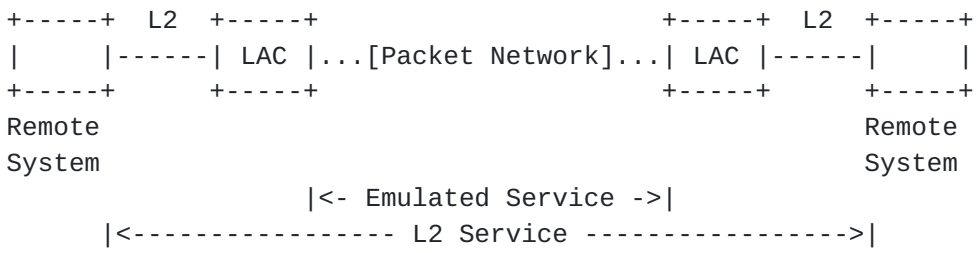


Figure 1: L2TP LAC to LAC Reference Model

LAC - L2TP Access Concentrator, analogous to the PWE3 PE.

Remote System - Analogous to the PWE3 CE.

Emulated Service - Analogous to the PWE3 pseudowire.

## 2.2 LAC to LNS Reference Model

An LNS is different from an LAC in that it does not have native Attachment Circuits on which to directly forward pseudowire frames to. Instead, the pseudowire frames are received and processed on a virtual interface as if received on a local AC as a CE. A convenient way to think of this is to imagine a PWE3 CE and PE collapsed into a single piece of equipment. The LNS MAY support virtual routing or bridging instances for groups of sessions.

An LNS may be used with an LAC to provide access to a network via emulated L2 circuits. This is the most commonly used model in L2TPv2 deployments for PPP tunneling at the time of this writing.

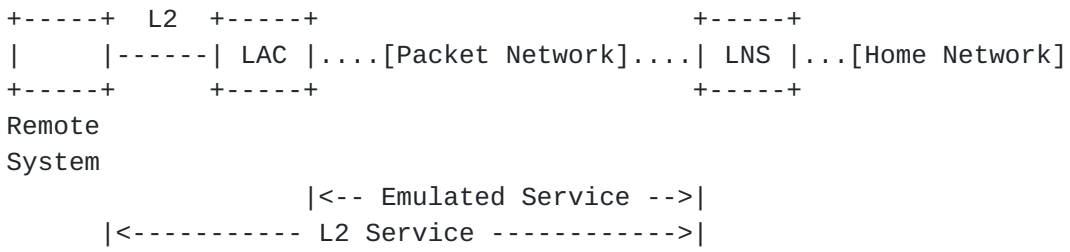


Figure 2: L2TP LAC to LNS Reference Model

LAC - L2TP Access Concentrator, analogous to the PWE3 PE.

Remote System - Analogous to the PWE3 CE.

LNS - L2TP Network Server, a PWE3 PE and CE collapsed into the same device.

Emulated Service - Analogous to the PWE3 pseudowire.

### 2.3 The LAC and LNS in L2TP

From an encapsulation perspective, and during Control Connection setup (described in section 4), L2TPv3 does not distinguish between whether an LAC or LNS is on either side of the PW. During session setup, there are some differences with regard to identification of attachment circuits and the like as an LNS does not contain physical circuits to attach to. For session setup, L2TPv2 makes more reference to the LAC and LNS as very separate different devices, while these differences were minimized in L2TPv3.

The remainder of this document will refer to the PWE3 terms "PE" and "CE" when discussing PWE3 applicability, though it is important to remember what an LAC, LNS, and Remote System are when reading the L2TP Protocol Documents.

## 3. L2TP Control Plane Overview and General Applicability

The L2TP Control Plane is divided into two parts, (1) a reliable Control Connection for sending control messages between PEs, and (2) state machines for establishing, maintaining, and tearing down the Control Connections and L2TP Sessions between each PE.

One L2TP Session is established for each pseudowire, and each Control Connection governs a group of Sessions as well as the common association between two PEs. There may be one or more Control Connections between two PEs, and typically are many Sessions within each Control Connection. It is important to note that L2TPv2 parlance

commonly referred to each established Control Connection as a "Tunnel" which is somewhat different than the definition in PWE3. L2TPv3 does away with "Tunnel" in its terminology to avoid obvious confusion, referring only to "Sessions" and "Control Connections."

## 3.1 L2TP Control Connection Establishment

The L2TP Control Connection is the primary association between two L2TP tunnel endpoints (PEs). A Control Connection must be established before any L2TP Sessions (and hence pseudowires) may be established.

A successful Control Connection establishment looks like this:

```
 PE A        PE B
------      ------
SCCRQ ->
            <- SCCRP
SCCCN ->
```

SCCRQ: Start Control Connection Request
SCCRP: Start Control Connection Reply
SCCCN: Start Control Connection Connected

Note that each of these messages is transported by the reliable control channel (i.e., there is an additional transport-level ack to the SCCCN that will occur, but this is a layer below the L2TP state machine). Either side of a connection may initiate the Control Connection by sending an SCCRQ, and a tie-breaker facility exists in the event that both sides attempt connection establishment at the same time.

During Control Connection establishment, PEs establish their identity and exchange capability information with one another. Identity may be confirmed via an optional challenge-handshake authentication exchange built into the control connection setup. Capability and relevant configuration information is advertised via AVPs in the SCCRQ or SCCRP messages.

The protocol constructs described in this section provide the basis for the following PWE3 requirements as identified in [PWE3-REQ]:

Misconnection and Payload Mistype (Section 3.3.2.)
Collective Status Notification (3.3.5.)
Tunnel Hierarchy (Scalability, Section 6.)

## 3.2 L2TP Session Establishment

Once the Control Connection is established, L2TP sessions (pseudowires) may be established. L2TP has two state machines, and two different sets of messages depending on the type of session to be established. We will limit the discussion in this document to the "Incoming Call" state machine and message exchange, as this is the commonly used case in PWE3 (see Appendix B for a discussion of the term "Call" and the "Outgoing Call" state machine and message exchange).

A successful L2TP session establishment looks like this:

```
 PE A        PE B
------      ------
ICRQ ->
            <- ICRP
ICCN ->
```

ICRQ: Incoming Call Request
ICRP: Incoming Call Reply
ICCN: Incoming Call Connected

During session establishment, characteristics of the individual pseudowire, interfaces, etc. may be exchanged, as well as ephemeral session attributes such as the Session ID used for data packet switching. In the event that both sides of the connection attempt to send an ICRQ at the same time, a tie-breaker AVP is used to determine which side "wins."

The protocol constructs described in this section provide the basis for the following PWE3 requirements as identified in [PWE3-REQ]

Setup and Teardown of Pseudo-Wires (Section 3.1.)
Misconnection and Payload Mistype (Section 3.3.2.)

## 3.3 Summary of Status and Maintenance Messages

All messages referred to here are sent as single independent messages, requiring no explicit state machine to operate above the reliable Control Connection. This section is intended to provide a summary of available messages in the L2TPv3 specification. For more details of each message, including AVPs used in the construction of each, etc. please refer to [L2TPv3].

Hello Message

The Hello message is used as a simple "keepalive" to ensure that the peer is still active. If control and data operate over the same path (as is the default for L2TPv3 over IP), then the delivery of this message may be used as a general indicator that a valid data path exists as well. This message is sent periodically by either peer, based on a configured time period, and may back off during periods of congestion or when other information (e.g. receipt of another control message) is available to indicate that a peer is still active. Thus, the Hello message is never "expected" by either side of the link. Instead, the Hello mechanism relies only on the generic Control Connection transport for delivery. For added scalability, the Hello message acts as a collective keepalive for all sessions associated with a given Control Connection.

The Hello message MUST NOT be used to carry status notification events, or other purposes beyond the scope of a simple peer and path detection facility.

Set Link Info (SLI) Message

The Set Link Info message is used to identify link status changes at a PE interface associated with a pseudowire. The most basic link status change is simply a circuit going up or down, which is advertised via the base Circuit Status AVP. Additional AVPs for each service type may be defined if additional information associated with an interface needs to be updated over the life of a connection. These AVPs MUST be defined in separate service-specific documents.

WAN Error Notify (WEN) Message

This message is used to identify any link interface or pseudowire errors that may occur and be of interest to the operator on the other side of the L2TP connection. This includes statistics on packet loss, out-of-order delivery (if sequencing is enabled) and packet corruption (if some form of CRC or Checksum is available). These values MAY be reported on a periodic interval, and are for logging and troubleshooting purposes. Any necessary values for a given service beyond those defined in [L2TPv3] MUST be defined in separate service-specific documents.

The protocol constructs described in this section provide the basis for the following PWE3 requirements as identified in [PWE3-REQ]:

Status Monitoring (Section 3.2.)

Up/Down Notification (Section 3.3.1.)
Keep-alive (Section 3.4.)
Packet Loss, Corruption, and Out-of-order Delivery (Section 3.3.3.)

## 3.4 Control Message Composition

Control Messages are sent with the L2TP Control Connection header to ensure proper delivery between PEs, followed by a list of Attribute Value Pairs (AVPs). AVPs in L2TP are of a similar form to the "Type Length Value" (TLV) constructs in other control protocols. The Message Type itself is an AVP, as is the specific Session ID or Control Connection ID for which the message applicable for. Please refer to [L2TPv3] for the specific format of these AVPs and the Control Message header.

## 3.5 Creating New Control Messages

Adding additional control messages to L2TP is a natural extension that may be utilized to signal events and information between pseudowire endpoints. Most line events and status notifications may be sent with a single, independent control message. Additional complexity via req/ack messages and state machines should be avoiding when a single reliably delivered message will suffice.

A new message may be created by simply assigning a new Message Type AVP value. The Message Type AVP is present at the beginning of the body of all L2TP Control Messages. This AVP may be vendor-specific, rendering the control message itself as vendor-specific.

The Vendor ID is zero for IETF defined messages, and set to the IANA assigned "SMI Network Management Private Enterprise Codes" [RFC1700] value for vendor-specific messages. IANA also assigns the Message Type value for IETF Control Messages, but vendor-specific messages must maintain their own number space.

L2TP Message Type AVP:

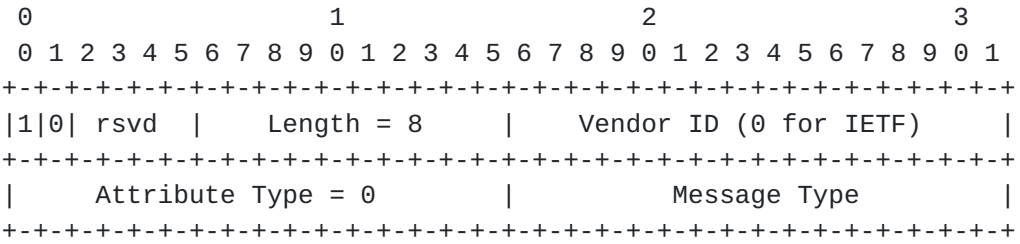


Figure 3: L2TP Message Type AVP

## 3.6 Advertising Support for Optional Features

The SCCRQ, SCCRP, and SCCCN messages should be utilized to advertise availability of a specific optional feature available for the Control Connection or the group of sessions within a Control Connection. This may be as simple as the presence of a single AVP, or a set of AVPs listing multiple parameters.

Support for a given pseudowire type MUST be advertised via the Pseudowire Capabilities List AVP.

## 3.7 Summary of Selected PWE3-Related Control Message AVPs

This section will highlight some more pertinent AVPs used in the setup and maintenance of pseudowires with L2TP.

### 3.7.1 Session Establishment AVPs

The following AVPs are among those exchanged during L2TP Session establishment. For a complete list and full details, refer to section 5.4.4 of [L2TPv3].

End Identifier AVP

The End Identifier AVP is of arbitrary length, and is used to identify the interface, circuit, or other entity to tie the L2TP session to. The field may be a simple 4-octet binary value, an ASCII string, or any other agreed-upon format for the given pseudowire type or application.

Pseudowire Type

The Pseudowire Type indicates the type of pseudowire for this L2TP session.

Assigned Cookie

The 0, 32, or 64-bit random value to be sent in each data packet for sanity checking the session context lookup. A 0-bit value is used to indicate that the Cookie field is not present.

Local Session ID

Used to exchange the pair of 32-bit Session IDs that are to be used in all data packet headers to identify an L2TP Session (and hence a pseudowire).

### 3.7.2 Control Connection Establishment AVPs

The following AVPs are exchanged during Control Connection establishment. For a complete list and full details, refer to section 5.4.3 of [L2TPv3].

Pseudowire Capabilities List (SCCRP, SCCRQ)

List of pseudowire types that this node is capable of receiving packets for. Used to advertise peer PE capabilities before initiating a session.

Host Name (SCCRP, SCCRQ)

The Host Name AVP carries a unique identifier, possible a fully qualified domain name, for the originator of the control connection.

Challenge (SCCRP, SCCRQ)

Random data hashed with a shared password to perform a simple Control Connection (PE to PE) authentication.

Challenge Response (SCCCN, SCCRP)

Result of the hash used for Control Connection (PE to PE) authentication.

## 4. L2TP Data Packet Encapsulation

### 4.1 L2TP over various PSNs

L2TP was designed from its inception to be able to operate over any packet-switched network (PSN). While UDP/IP is by far the most popular PSN for L2TPv2 tunneling PPP, specifications for other PSNs have been deployed [RFC3070], [L2TPAAL5] and is explicitly allowed in [RFC2661].

The L2TP control plane may also be used to setup alternate data encapsulations. For example, [L2TPHC] defines a header format for a specific application where the size of the L2TP and PPP header is of primary importance. L2TPv3 takes advantage of this as well, signaling its extended header format by the presence of the 32-bit Session ID as opposed to the 16-bit Session ID for L2TPv2. Other PW Multiplexing formats could be exchanged here as well, including an MPLS label, or GRE Key.

### 4.2 L2TPv3 over IP

Following is the format of the L2TPv3 data header defined for operation over IP Protocol ID 115.

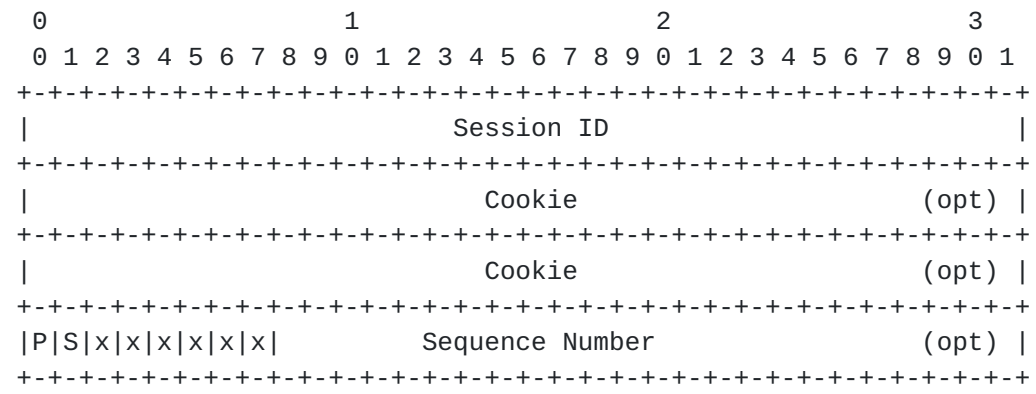


Figure 4: An L2TPv3 over IP Data Packet

The first four octets contain the L2TPv3 Session ID. The Session ID is used as a pseudowire identifier/multiplexing field. It contains 32-bits assigned by the receiver during session establishment. The expectation is that the receiver will optimize the bits in this Session ID for whatever context lookup mechanism may be available on the specific platform. For example, the context lookup mechanism for the pseudowire could be a simple index with the low-order bits used as a direct lookup, or specific bits could be used to distribute a packet to parallel processing paths in a distributed engine, platform, or cluster. If the following optional Cookie is utilized, the Session ID may be selected without concern for choosing a value which has been recently expired. The Session ID of 0 is reserved for L2TPv3 control traffic.

The remainder of L2TPv3 header beyond the Session ID is optional, and may even vary from what is shown above based upon the PW-type in operation. The L2TP control channel signals the presence of (and in many cases the values contained within) each field during session establishment. In order to ensure interoperability while still allowing some flexibility over what fields an implementation is required to know how to process, L2TP uses a simple governing principle for its variant header. This principle states that the receiver of a data packet always identifies the format it expects, and the sender MUST strictly comply. This is the case not only for presence of optional fields such as the Cookie or L2-specific sublayer, but for the contents of the Session ID and Cookie as well, i.e., one side always informs the other exactly what it expects to be present in a header.

The optional Cookie (0, 4 or 8 octets) is designed to be an efficient, lightweight, packet validation mechanism. Like the Session ID, it contains bits assigned by the receiver during session establishment for insertion in each packet. Unlike the Session ID, these bits do not serve as a context lookup mechanism, but rather as a simple check to see if the value in the data packets matches the value advertised for that Session. Also, unlike the Session ID, the value selected for a Cookie MUST be unpredictable and unique across current and recently-expired sessions.

Thus, while the form of the Session ID is free to be optimized for a local lookup mechanism, the Cookie provides an added degree of certainty that the arriving packet does in fact match the context obtained by the Session ID resolution. For example, if the Session ID context lookup for an arriving data packet was implemented as a simple index, a single bit error in the range of bits being used for the index could rather easily cause a packet to be sent to the wrong PW or interface. Since there is no network-level checksum available for the L2TP Session ID running directly over IP, this may be of concern depending on the potential for unchecked bit errors of the underlying data link. The Cookie provides up to a full 64 bits of additional guarantee that a packet is definitively intended for a given destination in face of such bit errors, as well as a way of detecting the arrival of any stale data packets after a Session ID has been released and reallocated. The Cookie also provides protection against simple brute force or blind data insertion attacks by a rogue entity.

The next four bytes in Figure 4 depict of the default L2-specific sublayer for L2TPv3. These fields contain pseudowire-specific values, and as such may vary among pseudowire types (e.g. RTP has its own sequencing, thus a given timing-dependent pseudowire type may choose to not use the default header defined here).

The P bit is a "priority" bit which SHOULD be set for end-to-end signaling packets traversing the session data path. This allows the PE to provide a higher priority to these packets when processing as they may be important for the health of the overall end-to-end link. This may be of particular concern during periods of congestion at the PE if the native service being emulated has an end-to-end keepalive method enabled.

The S bit indicates whether the following sequence number is valid or not.

The Sequence number is a 24-bit free running counter (0 is a valid sequence number).

The x bits are reserved for future use.

For further details of how these fields are used, please consult [L2TPv3].

The protocol constructs described in this section provide the basis for the following PWE3 requirements as identified in [PWE3-REQ]:

Handling Control Messages of the Native Services (Section 2.5.)
Frame Duplication (Section 2.3.)
Frame Ordering (Section 2.2.)
Support of Multiplexing and Demultiplexing (Section 2.1.3)
Support of Variable Length PDUs (Section 2.1.2.)

## 5. Security Considerations

L2TP provides a number of security features which may be used depending on the needs of a given deployment. Details of these are provided in [L2TPv3].

L2TP acts much like a client/server application operating between two nodes on a network. Thus, the pseudowire which L2TP operates with inherits many of the security properties of the underlying PSN (for good or for ill). In cases where the PSN is protected, physically, cryptographically, or otherwise, the pseudowire itself is also protected. This is discussed further in [PWE3-PLD].

[RFC3193] provides details on how to protect L2TPv2 with IPsec transport mode. This is applicable to L2TPv3 over IPsec as well, though the sections on UDP port "floating" may be ignored for the IP-only encapsulation. When running over IP, L2TP with IPsec provides cryptographic security for all data traffic within and control traffic in support of the L2TP pseudowires. Running IPsec beneath L2TP in this manner does not require any extensions to IPsec, and does not rely on IPsec for Tunnel Mode encapsulation or any extensions to the IPsec suite of protocols.

In the event that IPsec is not available, L2TP also provides a simple shared-secret Control Connection Authentication method to ensure that only authorized PE are permitted to establish a Control Connection (and hence any pseudowire connections within).

L2TPv3 includes an optional 64-bit Cookie in the header of each L2TP data packet which contains a random value identified at session start. While not its only functional purpose, the cookie field may be used to protect against blind insertion attacks for a given tunnel. This is not intended as a replacement for proper traffic filtering, nor for cryptographic security with IPsec when warranted.

However, in the absence of advanced traffic snooping, this field does provide a reasonable defense against brute force packet insertion attacks, and certainly provides for protection against misconfiguration or other potential means of inadverdent packet misdirection.

## 6. IANA Considerations

There are no new numbers defined in this document for IANA to maintain.

## 7. Acknowledgements

I would like to acknowledge the author of the first Ethernet over L2TP internet draft, Suhail Nanji, the first Frame Relay over L2TP authors, Nishit Vasavada, Jim Boyle, Chris Garner, Serge Maskalik, Vijay Gill, and the original "Encapsulation Services" authors for L2TP, Nishit Vasavada, Danny McPherson, Ravi Bail Bhat, Andy Koscinski, Chi Fai Ho. These individuals were among the original purveyors of non-PPP encapsulations over L2TP before L2TPv3 and the PWE3 Working Group was created.

Stewart Bryant, Eric Rosen, Lloyd Wood, and Wei Luo provided helpful review of this document.

## 8. Author's Address

W. Mark Townsley
cisco Systems
7025 Kit Creek Road
PO Box 14987
Research Triangle Park, NC 27709
mark@townsley.net

## 9. References

[RFC1700] Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994. See also: http://www.iana.org/numbers.html

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[RFC2661] Townsley, Valencia, Rubens, Pall, Zorn, Palter, "Layer Two Tunneling Protocol 'L2TP'", RFC 2661, June 1999.

[RFC2434] Alvestrand, H. and Narten, T., "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434,

October 1998.

[RFC3070] V. Rawat, R. Tio, S. Nanji, R. Verma, "Layer Two Tunneling Protocol (L2TP) over Frame Relay," RFC 3070, February 2001.

[PWE3-REQ] XiPeng Xiao, Danny McPherson, Prayson Pate, Craig White, Kireeti Kompella, Vijay Gill, Thomas D. Nadeau, "Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)," work in progress, draft-ietf-pwe3-requirements-02.txt, May 2002.

[PWE3-PLD] Stewart Bryant, Lloyd Wood, Mark Townsley, Danny McPherson, "Protocol Layering in PWE3," work in progress, draft-bryant-pwe3-protocol-layer-01.txt, February 2002.

[L2TPv3] J. Lau, M. Townsley, A. Valencia, G. Zorn, I. Goyret, G. Pall, A. Rubens, B. Palter, "Layer Two Tunneling Protocol (Version 3) 'L2TPv3'" work in progress, draft-ietf-l2tpext-l2tp-base-02.txt, February 2002.

[L2TP-PPP] J. Lau, M. Townsley, A. Valencia, G. Zorn, M. Verma, I. Goyret, G. Pall, A. Rubens, B. Palter, "PPP Tunneling Using Layer Two Tunneling Protocol" work in progress, draft-ietf-l2tpext-l2tp-ppp-01.txt, November 2001.

[L2TPAAL5] Mike Davison, Arthur Lin, Ajoy Singh, John Stephens, Rollins Turner, Rene Tio, Suhail Nanji, "L2TP over AAL5," work in progress, draft-ietf-l2tpext-l2tp-atm-02.txt, August 2001.

[L2TPHC] A. Valencia, "L2TP Header Compression ('L2TPHC')," work in progress, draft-ietf-l2tpext-l2tphc-04.txt, October 2001.

## Appendix A: L2TPv2 and L2TPv3

L2TPv2 [RFC2661] was first deployed as a method for bypassing expensive PSTN (Public Switching Telephone Network) networks with faster, cheaper, IP networks. L2VPNs were created by tunneling point to point PPP connections over IP (and to a lesser extent, Frame Relay and ATM) networks to multiple service providers. As broadband DSL was deployed, L2TP was used to offload PPP from ATM PVC networks, again utilizing the ability to dynamically create emulated point-to-point connections between any L2TP-enabled IP connected endpoint. More recently, L2TP has been used to tunnel Ethernet and Frame Relay, utilizing the familiar L2TP constructs for tunneling of PPP.

To provide specification modularity for tunneling pseudowire types other than PPP, RFC 2661 was split into a base specification [L2TPv3] and an accompanying PPP over L2TP specification [L2TP-PPP]. This base specification also introduced a new encapsulation format for transport directly over IP protocol ID 115 (as opposed to only UDP), an expanded Session ID space from 16 to 32 bits, an extended sequence number space from 16 to 32 bits, and other changes based on past implementation experience. These changes required an L2TP version field increment to 3, thus the name "L2TPv3" that has been adopted. The L2TPv2 header over UDP port 1701 may still be used for tunneling PPP as defined in [RFC2661], and likely will be for some time. Layer 2 tunneling other than PPP, MUST use the improved encapsulation method defined in the L2TPv3 specification.

The L2TPv2 state machines, reliable transport for message delivery, maintenance and status messages, etc. remain largely unchanged in L2TPv3 beyond the separation of PPP-specific AVPs (Attribute Value Pairs) from the base specification, and the addition of new AVPs to support the negotiation of pseudowire types and their associated connection to attached circuits.

For more information on the differences between L2TPv2 and L2TPv3, please see Section 1.1, "Changes from RFC 2661" and Section 4.7, "L2TPv2/v3 Interoperability and Migration" in [L2TPv3].

## Appendix B: "Incoming Calls" and "Outgoing Calls"

The term "Call" used in L2TP is a holdover from the action of receiving a call on a dialup line. PWE3 was not the first to make this original choice of wording seem a bit out of place; the very common use of L2TP in broadband applies this same "call" action to an ATM PVC. In this light, an L2TP "Incoming Call" becomes the action of a circuit being provisioned and transitioning to the "Up" or "Active" state. When looked at this way, it becomes a bit less awkward. The choice to keep the "Call" name was made to be consistent with the large installed base of existing implementations using this terminology.

An "Outgoing Call" is very different from an "Incoming Call" in L2TP. An "Outgoing Call" is SVC-type connection attempted in response to an L2TP Outgoing Call Request (OCRQ), directed to an arbitrary location identified by information contained in the L2TP message, and may take some time to be completed. For example, the classic L2TPv2 Outgoing Call application is dialing a modem on a PSTN based on a telephone number. The node establishing the SVC connection acts as a slave to the other side of the L2TP connection, establishing the SVC wherever indicated (within policy bounds which may be applied if necessary).

The Outgoing Call message exchange has a very clear initiator and responder, incapable of a "tie" as is the case in an Incoming Call exchange. Also, instead of three messages sent in response to one another in a handshake fashion as with an Incoming Call, an Outgoing Call is established by sending a single message from the initiator, and two messages sent from the responder in return. The first Outgoing Call message, the OCRQ, is sent to request the Outgoing Call to be made with the identified SVC method. The subsequent two messages, the Outgoing Call Reply (OCRQ) and Outgoing Call Connected (OCRP) messages signal that the SVC connection is being attempted, and that the SVC connection was completed, respectivly.

Thus, an Outgoing Call is essentially a method of controlling the originating point of an SVC, allowing it to be established from any reachable L2TP-enabled device able to perform outgoing calls. PWE3 has largely identfied incorporation of various SVC methods as "left for future study." Until said future study is completed, the Outgoing Call model may not find use outside of the dial arena where it is deployed with L2TPv2 today.

Full Copyright Statement