L2TPEXT Working Group
Internet Draft
November 2, 2002
Expires June 2003
draft-ietf-l2tpext-l2tphc-06.txt

Andrew J. Valencia
Tmima Koren
Cisco Systems

# L2TP Header Compression ("L2TPHC")

## Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

## Abstract

The Layer 2 Tunneling Protocol ("L2TP") defined in RFC 2661 defines a mechanism for tunneling PPP sessions over IP. There exists a class of specific media applications for which protocol overhead may be optimized, and where such reduction results in improved operation. This document describes the solution space addressed, its underlying motivations, and the protocol modifications required. The enhancement to the L2TP protocol is called L2TP Header Compression, or "L2TPHC".

## 1. Introduction

L2TP [RFC2661] defines a general purpose mechanism for tunneling PPP over various media. In most cases, the header overhead of the L2TP tunnel is negligible. However, when L2TP operates over bandwidth constrained networks such as dialup links or some classes of WAN backhauls, any savings of bytes transmitted results in a substantial efficiency gain. This effect is further amplified when streams of small IP packets dominate the traffic (thus increasing the header-to-payload ratio), as is common with multimedia and other types of real-time data traffic.

## 2. Simplifying Assumptions

If several simplifying assumptions are met, it is possible to reduce the size of the L2TP encapsulation over IP:

- The tunnel will not operate through a NAT interface
- The tunnel uses a single IP address for the life of the tunnel
- The tunnel's host uses only one public IP network interface
- There will be only one tunnel between the LAC and the LNS
- There might be only one session within a tunnel
- There might be only one protocol active on that session
- Alignment is not required
- Packet length is preserved by the IP header

Each of these simplifying assumptions directly relates to an L2TP protocol header field's function. Because NAT functionality is not needed, the UDP header is not required. Because the endpoints will not change their source IP addresses (due to either changing IP addresses, moving among IP egress points, or switching to a distinct backup IP interface), the identity of the peer may be determined by its source IP address, rather than the Tunnel ID. If there is only one tunnel, it is trivial to determine the Tunnel ID. Because each byte is a measurable component of overhead, it is better to send fields on unaligned boundaries rather than ever pad. Because IP will preserve the packet length end-to-end, there is no need to communicate this in the header itself.

In addition, several operational considerations permit further simplification:

- There is no need to optimize control packet overhead
- Version compatibility may be determined by control packets

The first two bytes of an L2TP payload header determine the presence of further, optional, fields. It also contains a Version field, used to detect compatible version operation.

In the presence of the simplying assumptions listed above, it is possible to systematically minimize or eliminate the L2TP fields in the header of an L2TP data message. For example, if one assumes that there is no more than a single session between two L2TP peers, then the session ID in the L2TP header becomes irrelevant and may be eliminated. Further, if there is only one version of L2TP running on a pair of L2TP nodes (or, specifically, IP addresses on two L2TP nodes), then there is no need for a version field in each data packet.

Each assumption translates to a piece of information that may be left out of the header. This document describes the most extreme case where the entire L2TP header and/or the entire PPP header is eliminated, resulting in a zero-byte "header" for each of these. Data packets which meet the simplifying assumptions are then sent and received over what is effectively a parallel data channel for packets with the ultra-compressed L2TP and PPP header. The uncompressed data channel still exists over UDP/IP as defined in RFC 2661, and any packets not meeting the simplifying assumptions may still be sent over this channel.

## 3. Tunnel Establishment

### 3.1 Negotiation

In order for two L2TP peers to send and receive data packets with the compressed L2TP header of zero octets, the "L2TPHC-No-Header" AVP MUST be sent and received in the ICRQ/ICRP or OCRQ/OCRP exchange during session setup. If either side did not send, or did not receive, this AVP during session establishment, L2TP MUST fall back to utilization of an uncompressed header format for its data.

A second AVP, "L2TPHC-PPP-Protocol," may also be included in the ICRQ/ICRP and OCRQ/OCRP message exchange to allow compression of the PPP framing fields. As with the L2TPHC-No-Header AVP, this AVP MUST be sent and received by both L2TP endpoints in order to enable PPP framing compression.

The Value of the L2TPHC-PPP-Protocol AVP contains a two octet PPP protocol number which will be assumed to be the single protocol type carried in the payload of all PPP packets carried by L2TPHC. This AVP indicates that the payload transmitted through L2TPHC will also omit PPP HDLC flags and control fields, in addition to the one or two byte protocol field indicated by the value in the L2TPHC-PPP-Protocol AVP. Any PPP packets with a protocol ID other than that indicated in this AVP, including any LCP or NCP control packets, MUST be sent over the uncompressed data channel with the entire L2TP over UDP/IP header intact.

### 3.2 AVP Formats

All AVP's MUST always be sent with the M, H, and "rsvd" bits all set to 0. All Attribute fields are 16-bit quantities in network byte order.

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |M|H| rsvd  |          6        |               0               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |             TBD1              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

L2TPHC-No-Header's Attribute is value TBD1.
There is no Value field. When L2TPHC-No-Header

is both sent and received, L2TPHC will directly encapsulate the PPP payload without any L2TPHC header byte.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|M|H| rsvd  |          8          |               0             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              TBD2             |          PPP Protocol         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

L2TPHC-PPP-Protocol's Attribute is value TBD2. The Value field is any legal PPP value for an NCP protocol. PPP allows some protocol types to be expressed in a compressed, 8 bit, form. The value included in this AVP is always the 16-bit form. This AVP indicates that PPP traffic carried over L2TPHC will not only have no L2TPHC header, but will also have no PPP address, control, or protocol fields. If necessary, these fields will be reconstructed on the receiving L2TPHC peer side, with the protocol value being always set to the Value indicated by this AVP.

## 4. Payload Exchange

After the L2TPHC-No-Header AVP is sent to and received from the peer, two data channels exist between the peers, one for compressed packets, the other for non-compressed packets. PPP payload packets may be sent to the peer's IP address over each of these two data channels. The compressed packets are sent as raw IP packets, with the IP protocol number set to 115 (the IANA-assigned value for L2TP). At the same time non-compressed packets may be sent over the non-compressed data channel as UDP-based L2TP packets. The payload so exchanged is always associated with the tunnel on which the AVP was received, and with the single session within that tunnel.

Note that the active L2TP control channel and associated Hello messages are sent as non-compressed packets and hence indicate tunnel endpoint reachability only for the non-compressed channel.

An L2TPHC packet is encoded as:

```
 0                   1
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- ...+
|        PPP packet...           ...|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+- ...+
```

The PPP frame will consist of the usual PPP-over-HDLC address, control, and protocol fields. However, if the L2TPHC-PPP-Protocol AVP has been sent and received, these fields are not present in the PPP payload, and must be re-inserted by the receiving side, using the protocol value indicated in the Value field of the L2TPHC-PPP-Protocol AVP.

## 5. Efficiency Considerations

Some rough calculations will illustrate the environments in which L2TPHC may be beneficial. Overhead as a percentage of the carried traffic will be calculated for a typical packet size involved in bulk data transfer (700 bytes), and the canonical 64-byte "small IP packet". Percentages will be rounded to the nearest whole number. Overhead is tallied for an IP header of 20 bytes, a UDP header of 8 bytes, an L2TP header of 8 bytes, and a PPP encapsulation of 4 bytes.

The worst case is a 64-byte packet carried within a UDP L2TP header. The 64 bytes of payload is carried by an overall header of 40 bytes, resulting in an overhead of 63%. With the larger payload size of 700 bytes, the header is amortized over many more bytes, reducing the overhead to 6%.

With L2TPHC, the UDP and L2TP headers are absent, and the 4 bytes of PPP encapsulation have been deleted. Overall size is thus 20 bytes of IP header. The small packet now suffers an overhead of only 31%, and the larger packet 3%.

Percentage overhead does not represent all the considerations involved in reducing overhead. Consider a modem connection operating at 14,400 bits per second, which translates to a per-byte real-time cost of 0.6 milliseconds (14400 divided by 8 bits, as async framing characters are not included in the modem-to-modem data transfer). A savings of 16 bytes per packet can also be viewed as a reduction of almost 10 milliseconds of latency per packet. While this latency is short enough to be unnoticeable by a human, it may impact real-time protocols such as streaming audio or video.

Thus, L2TP Header Compression provides most of its benefits when carrying streams of small packets. In environments such as downloading of graphic files, or where human interaction is intermingled with the short packets, the benefits of L2TP Header Compression will probably be undetectable.

## 6. Security Considerations

Because L2TPHC has no security facilities, it is critical that its operation be reconciled with the security policy of its environment. Since L2TPHC may have no protocol header at all, it is trivial to spoof a source IP address and inject malicious packets into an ongoing session. There are several suitable techniques for controlling this exposure.

In the simplest case, L2TPHC operates across a private network. For instance, a remote user may dial into a private NAS located on this network, and use L2TP (with or without L2TPHC) to cross an IP-only portion of this network to establish a multi-protocol session connected at a convenient point in the network. In this environment, no additional security may be required, and L2TPHC would operate trusting to the integrity of this private network.

If the weak protection of a difficult-to-guess protocol header is deemed sufficient, expanded protocol overhead has clearly been determined to be acceptable, and L2TP over UDP can be used without L2TPHC.

If PPP encryption under ECP [RFC1968] is active, malicious PPP packets are trivially detected and discarded as they are received on the raw IP port number. Similarly, if an IPsec session is protecting the IP packets themselves, malicious packets will also be discarded. Note that in both cases, an expanded header is implicit in these security facilities, which will greatly reduce the overhead efficiencies gained by L2TPHC.

## 7. IANA Considerations

This protocol defines two new Control Message Attribute Value Pairs (AVP's) in the IANA Layer Two Tunneling Protocol registry. As defined in [RFC2661] section 10.1, assignment of new AVP's is through IETF consensus. This document is intended to satisfy that requirement.

The two new AVP's are:

```
TBD1    L2TPHC-No-Header
TBD2    L2TPHC-PPP-Protocol
```

No registry of values is required for either AVP. Since these are IETF-adopted (not private) AVP's, the vendor ID field of the AVP should be set to zero.

[Note to RFC Editor: Please replace all instances of TBD1 and TBD2 here and in Section 3.2 with the IANA-assigned values.]

## 8. References

Normative References

[RFC2661] M. Townsley, "Layer 2 Tunnel Protocol (L2TP)", RFC 2661, August 1999

Informative References

[RFC1968] G. Meyer, "PPP Encryption Control Protocol (ECP)", RFC 1968, June 1996

## 9. Acknowledgments

Many thanks to W. Mark Townsley of Cisco Systems for his contributions in both the review and content of this document. Thanks to Gurdeep Singh Pall of Microsoft for identifying and describing scenarios in which L2TP header size become a concern. Thanks to Stephen Casner of Packet Design and to Bill Palter of Redback Networks for help in reviewing this document.

## 10. Authors' Addresses

Andrew J. Valencia
P.O. Box 2928
Vashon, WA  98070

Email: vandys@zendo.com

Tmima Koren
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
United States

EMail: tmima@cisco.com