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") [RFC2661] defines a mechanism for tunneling PPP sessions over arbitrary media. 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:

- 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 determined the presence of further, optional, fields. It also contains a Version field, used to detect compatible version operation. Realistically, these may all be determined in advance of payload exchange. Thus, by choosing very reasonable simplifying assumptions, it is possible to minimize the L2TP fields in the header of a payload packet. The resulting protocol is a 0-byte "header" (i.e., the header is absent). This would then be followed by a PPP frame (whose PPP encapsulation is also made optional), all encapsulated within a raw IP protocol header. These packets are exchanged in parallel with the regular UDP-based L2TP tunnel which provides all management and related functions.

3. Tunnel Establishment

3.1 Negotiation

L2TPHC is negotiated by an optional AVP "L2TPHC-No-Header" which is placed in the ICRQ/ICRP and OCRQ/OCR Pit session establishment messages. The effect of this AVP will never occur until L2TP reaches a state where the session within the tunnel is fully established (i.e., success indicated in the ICCN/OCCN). Additionally, each side intending to use L2TPHC MUST NOT do so unless it both sends and receives this AVP. Thus, unless both sides support L2TPHC, the optional AVP (in the ICRQ or OCRQ message) will be ignored by one side, and not sent (in the corresponding ICRP or OCRP) to the other side, and L2TP will operate in its regular mode. If this AVP is both sent and received, then payload is sent with no L2TPHC header at all--the PPP payload is carried directly within the IP packet. There can be only a single tunnel and a single session using L2TPHC between any two IP peers. Each individual PPP session MUST be identified by its IP source/destination pair.

Another AVP, "L2TPHC-PPP-Protocol", may also be included in the control message. If both sent and received, its Value field indicates the PPP protocol number which will be the single value carried in the payload of all PPP packets, and the payload transmitted through L2TPHC will omit PPP HDLC flags and control, as well as the one or two byte protocol field. The receiving side would then have to recreate a suitable PPP header and insert it onto received payload.

When L2TPHC is negotiated, all control messages are sent over the L2TP tunnel with an L2TP header. If a default PPP protocol number was also negotiated, all PPP packets with protocols different than the default must also be sent with an L2TP header.
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. The Vendor ID of each is 9, reflecting Cisco Systems, the initial developer of this extension. New Attribute values under Vendor ID 0 MUST be assigned when this document advances on the standards track.

L2TPHC-No-Header’s Attribute is value 1. 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.

L2TPHC-PPP-Protocol’s Attribute is value 2. The Value field is any legal PPP value for an NCP protocol. Note that all PPP protocol values which can be sent in 8-bit format always have a corresponding 16-bit format, and this 16-bit format is always the one used in this AVP. This AVP can only follow an L2TPHC-No-Header AVP, and 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

If the L2TPHC-No-Header AVP is sent to and received from the peer, PPP payload packets may be sent to the peer’s IP address as raw IP packets, with the IP protocol number set to 115 (the IANA-assigned value for L2TP). Such payload may be sent any time it would have been legal to send such payload over the regular UDP-based L2TP tunnel. Similarly, payload over the UDP tunnel MUST always be accepted, even after payload has flowed using the header compressed raw IP packet format. The payload so exchanged is always associated with the tunnel on which the AVP was received, and with the single session within that tunnel.

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 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

Two new AVP’s are defined in this paper: L2TPHC-No-Header and L2TPHC-PPP-Protocol. The vendor ID of these AVP’s should be 0. An Attribute needs to be assigned to each of the two new AVP’s through IETF Consensus, as defined in [RFC2661] section 10.1.

8. References

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


9. Acknowledgments

Thanks to Gurdeep Singh Pall of Microsoft for identifying and describing scenarios in which L2TP header size become a concern.

Thanks to Bill Palter of Redback Networks and W. Mark Townsley of Cisco Systems for help in reviewing this draft.

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

Valencia expires in 6 months