Network Working Group
Request for Comments: 4623
Category: Standards Track

A. Malis
Tellabs
M. Townsley
Cisco Systems

August 2006

Pseudowire Emulation Edge-to-Edge (PWE3) Fragmentation and Reassembly

Status of This Memo

Copyright Notice

Abstract

This document defines a generalized method of performing fragmentation for use by Pseudowire Emulation Edge-to-Edge (PWE3) protocols and services.

Table of Contents

1. Introduction

The Pseudowire Emulation Edge-to-Edge Architecture Document [Architecture] defines a network reference model for PWE3:

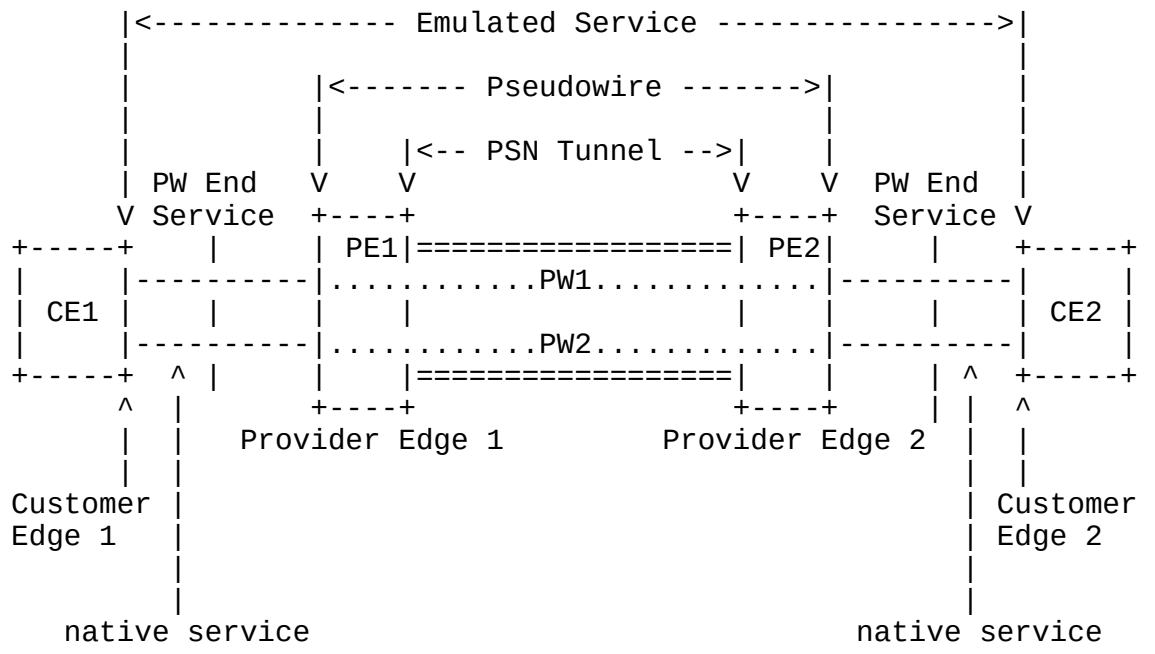

Figure 1: PWE3 Network Reference Model

A Pseudowire (PW) payload is normally relayed across the PW as a single IP or MPLS Packet Switched Network (PSN) Protocol Data Unit (PDU). However, there are cases where the combined size of the payload and its associated PWE3 and PSN headers may exceed the PSN path Maximum Transmission Unit (MTU). When a packet exceeds the MTU of a given network, fragmentation and reassembly will allow the packet to traverse the network and reach its intended destination.

The purpose of this document is to define a generalized method of performing fragmentation for use with all PWE3 protocols and services. This method should be utilized only in cases where MTU-management methods fail. Due to the increased processing overhead, fragmentation and reassembly in core network devices should always be considered something to avoid whenever possible.

The PWE3 fragmentation and reassembly domain is shown in Figure 2:

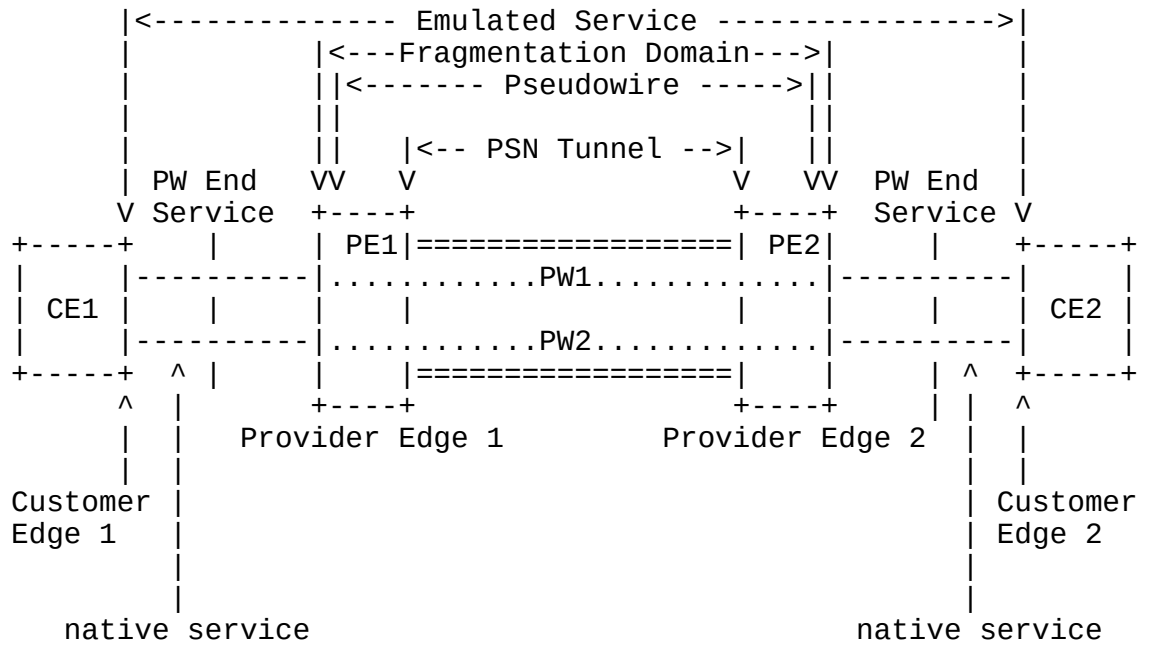

Figure 2: PWE3 Fragmentation/Reassembly Domain

Fragmentation takes place in the transmitting PE immediately prior to PW encapsulation, and reassembly takes place in the receiving PE immediately after PW decapsulation.

Since a sequence number is necessary for the fragmentation and reassembly procedures, using the Sequence Number field on fragmented packets is REQUIRED (see Sections 4.1 and 5.5 for the location of the Sequence Number fields for MPLS and L2TPv3 encapsulations, respectively). The order of operation is that first fragmentation is performed, and then the resulting fragments are assigned sequential sequence numbers.

Depending on the specific PWE3 encapsulation in use, the value 0 may not be a part of the sequence number space, in which case its use for fragmentation must follow this same rule: as the sequence number is incremented, it skips zero and wraps from 65535 to 1. Conversely, if the value 0 is part of the sequence space, then the same sequence space is also used for fragmentation and reassembly.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [KEYWORDS].

3. Alternatives to PWE3 Fragmentation/Reassembly

Fragmentation and reassembly in network equipment generally requires significantly greater resources than sending a packet as a single unit. As such, fragmentation and reassembly should be avoided whenever possible. Ideal solutions for avoiding fragmentation include proper configuration and management of MTU sizes between the Customer Edge (CE) router and Provider Edge (PE) router and across the PSN, as well as adaptive measures that operate with the originating host (e.g., [PATHMTU], [PATHMTUv6]) to reduce the packet sizes at the source.

In some cases, a PE may be able to fragment an IP version 4 (IPv4) [RFC791] packet before it enters a PW. For example, if the PE can fragment and forward IPv4 packets with the DF bit clear in a manner that is identical to an IPv4 router, it may fragment packets arriving from a CE, forwarding the IPv4 fragments with associated framing for that attachment circuit (AC) over the PW. Architecturally, the IPv4 fragmentation happens before reaching the PW, presenting multiple frames to the PW to forward in the normal manner for that PWType. Thus, this method is entirely transparent to the PW encapsulation and to the remote end of the PW itself. Packet fragments are ultimately reassembled on the destination IPv4 host in the normal way. IPv6 packets are not to be fragmented in this manner.

4. PWE3 Fragmentation with MPLS

When using the signaling procedures in [MPLS-Control], there is a Pseudowire Interface Parameter Sub-TLV type used to signal the use of fragmentation when advertising a VC label [IANA]:

Parameter	Length	Description
0x09	4	Fragmentation indicator

The presence of this parameter in the VC FEC element indicates that the receiver is able to reassemble fragments when the control word is in use for the VC label being advertised. It does not obligate the sender to use fragmentation; it is simply an indication that the sender MAY use fragmentation. The sender MUST NOT use fragmentation if this parameter is not present in the VC FEC element.

If [MPLS-Control] signaling is not in use, then whether or not to use fragmentation MUST be configured in the sender.

4.1. Fragment Bit Locations for MPLS

MPLS-based PWE3 uses the following control word format [Control-Word], with the B and E fragmentation bits identified in position 8 and 9:

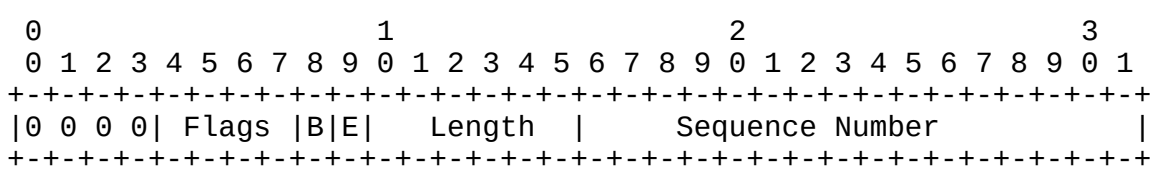

Figure 3: Preferred PW MPLS Control Word

The B and E bits are defined as follows:

BE
--
00 indicates that the entire (un-fragmented) payload is carried in a single packet
01 indicates the packet carrying the first fragment
10 indicates the packet carrying the last fragment
11 indicates a packet carrying an intermediate fragment

See Appendix A for a discussion of the derivation of these values for the B and E bits.

See Section 1 for the description of the use of the Sequence Number field.

4.2. Other Considerations

Path MTU [PATHMTU] [PATHMTUv6] may be used to dynamically determine the maximum size for fragments. The application of path MTU to MPLS is discussed in [LABELSTACK]. The maximum size of the fragments may also be configured. The signaled Interface MTU parameter in [MPLS-Control] SHOULD be used to set the maximum size of the reassembly buffer for received packets to make optimal use of reassembly buffer resources.

5. PWE3 Fragmentation with L2TP

This section defines the location of the B and E bits for L2TPv3 [L2TPv3] and L2TPv2 [L2TPv2] headers, as well as the signaling mechanism for advertising MRU (Maximum Receive Unit) values and support for fragmentation on a given PW. As IP is the most common PSN used with L2TP, IP PSN fragmentation and reassembly is discussed as well.

5.1. PW-Specific Fragmentation vs. IP fragmentation

When proper MTU management across a network fails, IP PSN fragmentation and reassembly may be used to accommodate MTU mismatches between tunnel endpoints. If the overall traffic requiring fragmentation and reassembly is very light, or there are sufficient optimized mechanisms for IP PSN fragmentation and reassembly available, IP PSN fragmentation and reassembly may be sufficient.

When facing a large number of PW packets requiring fragmentation and reassembly, a PW-specific method has properties that potentially allow for more resource-friendly implementations. Specifically, the ability to assign buffer usage on a per-PW basis and PW sequencing may be utilized to gain advantage over a general mechanism applying to all IP packets across all PWs. Further, PW fragmentation may be more easily enabled in a selective manner for some or all PWs, rather than enabling reassembly for all IP traffic arriving at a given node.

Deployments SHOULD avoid a situation that uses a combination of IP PSN and PW fragmentation and reassembly on the same node. Such operation clearly defeats the purpose behind the mechanism defined in this document. This is especially important for L2TPv3 pseudowires, since potentially fragmentation can take place in three different places (the IP PSN, the PW, and the encapsulated payload). Care must be taken to ensure that the MTU/MRU values are set and advertised properly at each tunnel endpoint to avoid this. When fragmentation is enabled within a given PW, the DF bit MUST be set on all L2TP over IP packets for that PW.

L2TPv3 nodes SHOULD participate in Path MTU ([PATHMTU], [PATHMTUv6]) for automatic adjustment of the PSN MTU. When the payload is IP, Path MTU should be used at they payload level as well.

5.2. Advertising Reassembly Support in L2TP

The constructs defined in this section for advertising fragmentation support in L2TP are applicable to [L2TPv3] and [L2TPv2].

This document defines two new AVPs to advertise maximum receive unit values and reassembly support. These AVPs MAY be present in the Incoming-Call-Request (ICRQ), Incoming-Call-Reply (ICRP), Incoming-Call-Connected (ICCN), Outgoing-Call-Request (OCRQ), Outgoing-Call-Reply (OCRP), Outgoing-Call-Connected (OCCN), or Set-Link-Info (SLI) messages. The most recent value received always takes precedence over a previous value and MUST be dynamic over the life of the

session if received via the SLI message. One of the two new AVPs (MRRU) is used to advertise that PWE3 reassembly is supported by the sender of the AVP. Reassembly support MAY be unidirectional.

5.3. L2TP Maximum Receive Unit (MRU) AVP

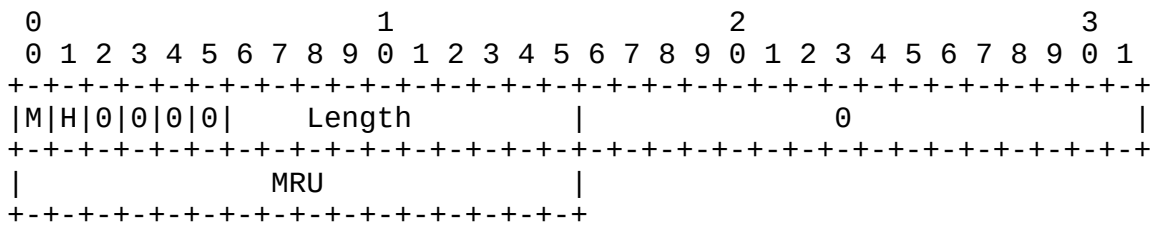

Figure 4: L2TP Maximum Receive Unit (MRU) AVP

MRU (Maximum Receive Unit), attribute number 94, is the maximum size, in octets, of a fragmented or complete PW frame, including L2TP encapsulation, receivable by the side of the PW advertising this value. The advertised MRU does NOT include the PSN header (i.e., the IP and/or UDP header). This AVP does not imply that PWE3 fragmentation or reassembly is supported. If reassembly is not enabled or unavailable, this AVP may be used alone to advertise the MRU for a complete frame.

This AVP MAY be hidden (the H bit MAY be 0 or 1). The mandatory (M) bit for this AVP SHOULD be set to 0. The Length (before hiding) is 8. The Vendor ID is the IETF Vendor ID of 0.

5.4. L2TP Maximum Reassembled Receive Unit (MRRU) AVP

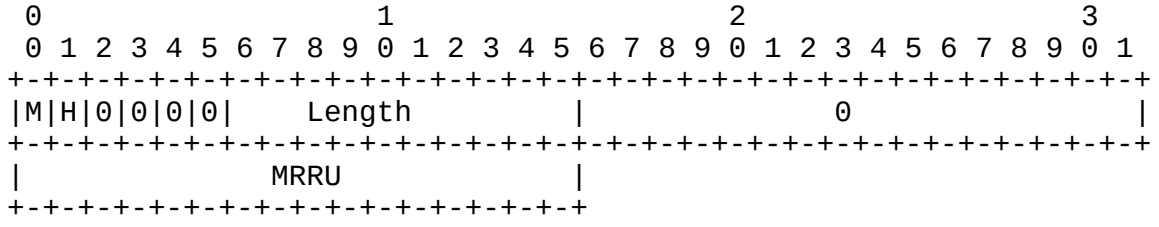

Figure 5: L2TP Maximum Reassembled Receive Unit (MRRU) AVP

MRRU (Maximum Reassembled Receive Unit AVP), attribute number 95, is the maximum size, in octets, of a reassembled frame, including any PW framing, but not including the L2TP encapsulation or L2-specific sublayer. Presence of this AVP signifies the ability to receive PW fragments and reassemble them. Packet fragments MUST NOT be sent by a peer that has not received this AVP in a control message. If the MRRU is present in a message, the MRU AVP MUST be present as well.

The MRRU SHOULD be used to set the maximum size of the reassembly buffer for received packets to make optimal use of reassembly buffer resources.

This AVP MAY be hidden (the H bit MAY be 0 or 1). The mandatory (M) bit for this AVP SHOULD be set to 0. The Length (before hiding) is 8. The Vendor ID is the IETF Vendor ID of 0.

5.5. Fragment Bit Locations for L2TPv3 Encapsulation

The usage of the B and E bits is described in Section 4.1. For L2TPv3 encapsulation, the B and E bits are defined as bits 2 and 3 in the leading bits of the Default L2-Specific Sublayer (see Section 7).

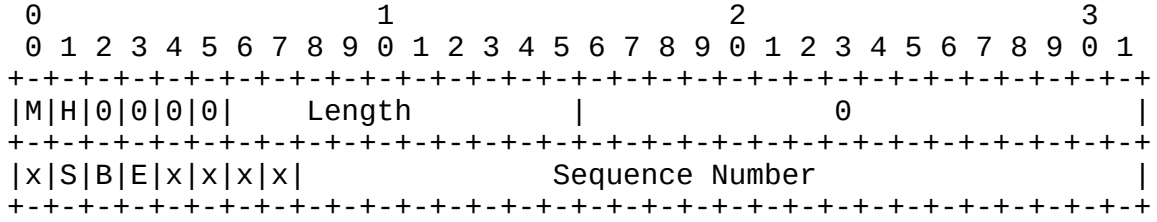

Figure 6: B and E Bits Location in the Default L2-Specific Sublayer

The S (Sequence) bit is as defined in [L2TPv3]. Location of the B and E bits for PW-Types that use a variant L2 specific sublayer are outside the scope of this document.

When fragmentation is used, an L2-Specific Sublayer with B and E bits defined MUST be present in all data packets for a given session. The presence and format of the L2-Specific Sublayer is advertised via the L2-Specific Sublayer AVP, Attribute Type 69, defined in Section 5.4.4 of [L2TPv3].

See Section 1 for the description of the use of the Sequence Number field.

5.6. Fragment Bit Locations for L2TPv2 Encapsulation

The usage of the B and E bits is described in Section 4.1. For L2TPv2 encapsulation, the B and E bits are defined as bits 8 and 9 in the leading bits of the L2TPv2 header as depicted below (see Section 7).

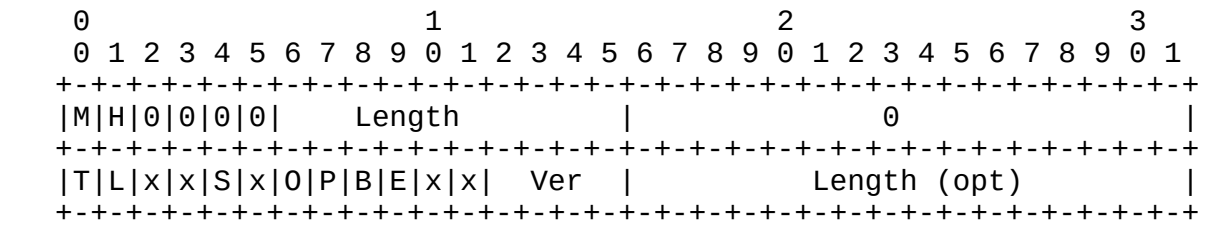

Figure 7: B and E bits location in the L2TPv2 Message Header

6. Security Considerations

As with any additional protocol construct, each level of complexity adds the potential to exploit protocol and implementation errors. Implementers should be especially careful of not tying up an abundance of resources, even for the most pathological combination of packet fragments that could be received. Beyond these issues of general implementation quality, there are no known notable security issues with using the mechanism defined in this document. It should be pointed out that RFC 1990, on which this document is based, and its derivatives have been widely implemented and extensively used in the Internet and elsewhere.

[IPFRAG-SEC] and [TINYFRAG] describe potential network attacks associated with IP fragmentation and reassembly. The issues described in these documents attempt to bypass IP access controls by sending various carefully formed "tiny fragments", or by exploiting the IP offset field to cause fragments to overlap and rewrite interesting portions of an IP packet after access checks have been performed. The latter is not an issue with the PW-specific fragmentation method described in this document, as there is no offset field. However, implementations MUST be sure not to allow more than one whole fragment to overwrite another in a reconstructed frame. The former may be a concern if packet filtering and access controls are being placed on tunneled frames within the PW encapsulation. To circumvent any possible attacks in either case, all filtering and access controls should be applied to the resulting reconstructed frame rather than any PW fragments.

7. IANA Considerations

This document does not define any new registries for IANA to maintain.

Note that [IANA] has already allocated the Fragmentation Indicator interface parameter, so no further IANA action is required.

This document requires IANA to assign new values for registries already managed by IANA (see Sections 7.1 and 7.2) and two reserved bits in an existing header (see Section 7.3).

7.1. Control Message Attribute Value Pairs (AVPs)

Two additional AVP Attributes are specified in Sections 5.3 and 5.4. They are required to be defined by IANA as described in Section 2.2 of [BCP0068].

Control Message Attribute Value Pairs
---

94 - Maximum Receive Unit (MRU) AVP
95 - Maximum Reassembled Receive Unit (MRRU) AVP

7.2. Default L2-Specific Sublayer Bits

This registry was created as part of the publication of [L2TPv3]. This document defines two reserved bits in the Default L2-Specific Sublayer in Section 5.5, which may be assigned by IETF Consensus [RFC2434]. They are required to be assigned by IANA.

Default L2-Specific Sublayer bits - per [L2TPv3]

Bit 2 - B (Fragmentation) bit
Bit 3 - E (Fragmentation) bit

7.3. Leading Bits of the L2TPv2 Message Header

This document requires definition of two reserved bits in the L2TPv2 [L2TPv2] header. Locations are noted by the "B" and "E" bits in Section 5.6.

Leading Bits of the L2TPv2 Message Header - per [L2TPv2, L2TPv3]

Bit 8 - B (Fragmentation) bit
Bit 9 - E (Fragmentation) bit

8. Acknowledgements

The authors wish to thank Eric Rosen and Carlos Pignataro, both of Cisco Systems, for their review of this document.

9. Normative References

[Control-Word] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, February 2006.

[IANA] Martini, L., "IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)", BCP 116, RFC 4446, April 2006.

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[LABELSTACK] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, January 2001.

[L2TPv2] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, August 1999.

[L2TPv3] Lau, J., Townsley, M., and I. Goyret, "Layer Two Tunneling Protocol - Version 3 (L2TPv3)", RFC 3931, March 2005.

[MLPPP] Sklower, K., Lloyd, B., McGregor, G., Carr, D., and T. Coradetti, "The PPP Multilink Protocol (MP)", RFC 1990, August 1996.

[MPLS-Control] Martini, L., Rosen, E., El-Aawar, N., Smith, T., and G. Heron, "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", RFC 4447, April 2006.

[PATHMTU] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, November 1990.

[PATHMTUv6] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.

10. Informative References

[Architecture] Bryant, S. and P. Pate, "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, March 2005.

[BCP0068] Townsley, W., "Layer Two Tunneling Protocol (L2TP) Internet Assigned Numbers Authority (IANA) Considerations Update", BCP 68, RFC 3438, December 2002.

[FAST] ATM Forum, "Frame Based ATM over SONET/SDH Transport (FAST)", af-fbatm-0151.000, July 2000.

[FRF.12] Frame Relay Forum, "Frame Relay Fragmentation Implementation Agreement", FRF.12, December 1997.

[IPFRAG-SEC] Ziemba, G., Reed, D., and P. Traina, "Security Considerations for IP Fragment Filtering", RFC 1858, October 1995.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[TINYFRAG] Miller, I., "Protection Against a Variant of the Tiny Fragment Attack (RFC 1858)", RFC 3128, June 2001.

Appendix A. Relationship between This Document and RFC 1990

The fragmentation of large packets into smaller units for transmission is not new. One fragmentation and reassembly method was defined in RFC 1990, Multi-Link PPP [MLPPP]. This method was also adopted for both Frame Relay [FRF.12] and ATM [FAST] network technology. This document adopts the RFC 1990 fragmentation and reassembly procedures as well, with some distinct modifications described in this appendix. Familiarity with RFC 1990 is assumed.

RFC 1990 was designed for use in environments where packet fragments may arrive out of order due to their transmission on multiple parallel links, specifying that buffering be used to place the fragments in correct order. For PWE3, the ability to reorder fragments prior to reassembly is OPTIONAL; receivers MAY choose to drop frames when a lost fragment is detected. Thus, when the sequence number on received fragments shows that a fragment has been skipped, the partially reassembled packet MAY be dropped, or the receiver MAY wish to wait for the fragment to arrive out of order. In the latter case, a reassembly timer MUST be used to avoid locking up buffer resources for too long a period.

Dropping out-of-order fragments on a given PW can provide a considerable scalability advantage for network equipment performing reassembly. If out-of-order fragments are a relatively rare event on a given PW, throughput should not be adversely affected by this. Note, however, if there are cases where fragments of a given frame are received out-or-order in a consistent manner (e.g., a short fragment is always switched ahead of a larger fragment), then dropping out-of-order fragments will cause the fragmented frame never to be received. This condition may result in an effective denial of service to a higher-lever application. As such, implementations fragmenting a PW frame MUST at the very least ensure that all fragments are sent in order from their own egress point.

An implementation may also choose to allow reassembly of a limited number of fragmented frames on a given PW, or across a set of PWs with reassembly enabled. This allows for a more even distribution of reassembly resources, reducing the chance that a single or small set of PWs will exhaust all reassembly resources for a node. As with dropping out-of-order fragments, there are perceivable cases where this may also provide an effective denial of service. For example, if fragments of multiple frames are consistently received before each frame can be reconstructed in a set of limited PW reassembly buffers, then a set of these fragmented frames will never be delivered.

RFC 1990 headers use two bits that indicate the first and last fragments in a frame, and a sequence number. The sequence number may be either 12 or 24 bits in length (from [MLPPP]):

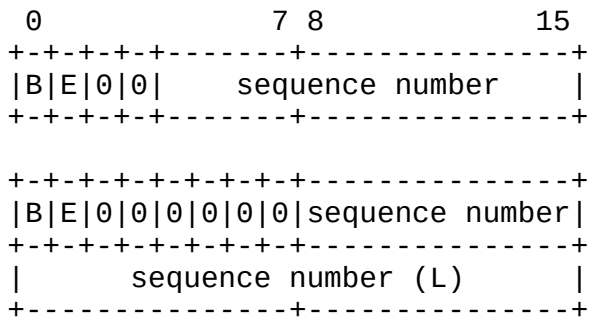

Figure 6: RFC 1990 Header Formats

PWE3 fragmentation takes advantage of existing PW sequence numbers and control bit fields wherever possible, rather than defining a separate header exclusively for the use of fragmentation. Thus, it uses neither of the RFC 1990 sequence number formats described above, relying instead on the sequence number that already exists in the PWE3 header.

RFC 1990 defines two one-bit fields: a (B)eginning fragment bit and an (E)nding fragment bit. The B bit is set to 1 on the first fragment derived from a PPP packet and set to 0 for all other fragments from the same PPP packet. The E bit is set to 1 on the last fragment and set to 0 for all other fragments. A complete unfragmented frame has both the B and E bits set to 1.

PWE3 fragmentation inverts the value of the B and E bits, while retaining the operational concept of marking the beginning and ending of a fragmented frame. Thus, for PW the B bit is set to 0 on the first fragment derived from a PW frame and set to 1 for all other fragments derived from the same frame. The E bit is set to 0 on the last fragment and set to 1 for all other fragments. A complete unfragmented frame has both the B and E bits set to 0. The motivation behind this value inversion for the B and E bits is to allow complete frames (and particularly, implementations that only support complete frames) simply to leave the B and E bits in the header set to 0.

In order to support fragmentation, the B and E bits MUST be defined or identified for all PWE3 tunneling protocols. Sections 4 and 5 define these locations for PWE3 MPLS [Control-Word], L2TPv2 [L2TPv2], and L2TPv3 [L2TPv3] tunneling protocols.

Authors' Addresses

Andrew G. Malis
Tellabs
1415 West Diehl Road
Naperville, IL 60563

EMail: Andy.Malis@tellabs.com

W. Mark Townsley
Cisco Systems
7025 Kit Creek Road
PO Box 14987
Research Triangle Park, NC 27709

EMail: mark@townsley.net

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).