Network Working Group
Request for Comments: 3153
Category: Standards Track

R. Pazhyannur
I. Ali
Motorola
C. Fox
Cisco Systems
August 2001

PPP Multiplexing

Status of this Memo

Copyright Notice

Abstract

This document describes a method to reduce the PPP (Point-to-Point Protocol) framing overhead used to transport small packets over slow links.

1. Description

The method, PPP Multiplexing, sends multiple PPP encapsulated packets in a single PPP frame. As a result, the PPP overhead per packet is reduced. PPP encapsulation (for example with PPP in HDLC framing) adds several bytes of overhead: a HDLC flag (at least one to separate adjacent packets), the Address (0xFF) and Control (0x03) field bytes, a two byte PPP Protocol ID, and the two byte CRC field. Even with the Address and Control Fields negotiated off and the PPP Protocol ID compressed, each PPP encapsulated frame will include four bytes of overhead. When PPP frames are tunneled, as in L2TP [1], the L2TP overhead per PPP frame is significant.

The key idea is to concatenate multiple PPP encapsulated frames into a single PPP multiplexed frame by inserting a delimiter before the beginning of each frame. The description of the delimiters is provided in Subsection 1.1. The delimiters are used by the demultiplexor to separate the PPP frames within the multiplexed frame. Each PPP encapsulated frame within the multiplexed frame is called a PPP subframe.

During the NCP negotiation phase of PPP, a receiver can offer to receive multiplexed frames using the PPP Mux Control Protocol (PPPMuxCP), as described in Section 2. Once PPPMuxCP has been negotiated, the transmitter may choose which PPP frames to multiplex. Frames should not be re-ordered by either the transmitter or receiver regardless of whether they arrive as part of the PPP multiplexed frame or by themselves.

The scheme proposed is similar to the concatenated framing option [2]. The key differences are that PPP multiplexing is more efficient and that it allows concatenation of variable sized frames. This is unlike concatenated framing which restricts all frames to be of fixed length.

As with any concatenation scheme, the implementer has to consider the tradeoff between increased delay for multiplexing/demultiplexing and reduced packet overhead as the length of the multiplexed frame increases.

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 [7].

1.1. Payload Format

The format of the complete PPP frame along with multiple subframes for PPP in HDLC-like framing [3] is shown in Figure 1. Note that regardless of the order in which individual bits are transmitted, i.e., LSB first or MSB first, the PFF bit will be seen to be the MSB of a byte that contains both the PFF and the subframe length field.

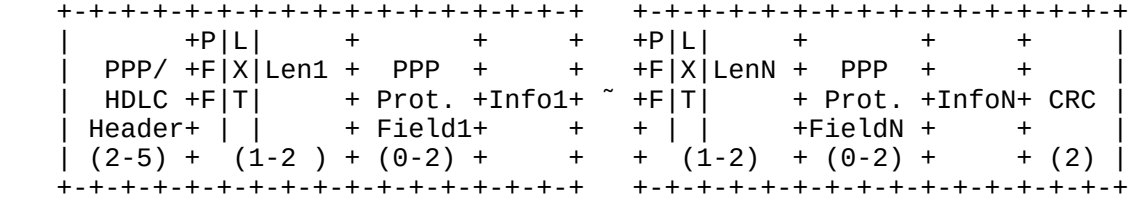

Figure 1. Multiplexing subframes in a PPP frame.

PPP Header:
The PPP header contains the PPP Protocol Field for a PPP Multiplexed Frame (0x0059). The PPP header compression options (ACFC and PFC) may be negotiated during LCP and could thus affect the format of this header.

Length Field:

The length field consists of three subfields:

1. Protocol Field Flag (PFF):

   The PFF refers to the most significant bit of the first byte of each subframe. This one bit field indicates whether the PPP Protocol ID of the subframe follows the subframe length field. For the first subframe, the PFF bit could be set to zero if the PPP protocol ID of the first subframe is equal to the default PID value negotiated in PPPMuxCP. PFF = 1 indicates that the protocol field is present (and follows the length field) for this subframe. PFF = 0 indicates that the protocol field is absent for this subframe. If PFF = 0 then the PPP Protocol ID is the same as that of the preceding subframe with PFF = 1, or it is equal to default PID value of the PPPMuxCP Option for the first subframe. The transmitter is not obligated to remove the PPP Protocol ID for any subframe.

2. Length Extension (LXT)

   This one bit field indicates whether the length field is one byte or two bytes long. If the LXT bit is set, then the length field is two bytes long (a PFF bit, a length extension bit, and 14 bits of sub-frame length). If the LXT bit is cleared, then the length field is one byte long (a PFF bit, a length extension bit, and 6 bits of sub-frame length).

3. Sub-frame Length (LEN):

   This is the length of the subframe in bytes not including the length field. However, it does include the PPP Protocol ID if present (i.e., if PFF = 1). If the length of the subframe is less than 64 bytes (less than or equal to 63 bytes), LXT is set to zero and the last six bits of the length field is the subframe length. If the length of the subframe is greater than 63 bytes, LXT is set to one and the last 14 bits of the length field is the length of the subframe. The maximum length of a subframe is 16,383 bytes. PPP packets larger than 16,383 bytes will need to be sent in their own PPP frame. A transmitter is not required to multiplex all frames smaller than 16,383 bytes. It may chose to only multiplex frames smaller than a configurable size into a PPP multiplexed frame.

Protocol Field:

This field contains the Protocol Field value for the subframe. This field is optional. If PFF = 1 for a subframe, the protocol field is present in the subframe, otherwise it is inferred at the receiver.

The receiver MUST support Protocol-Field-Compression (PFC) one or two bytes long. The transmitter SHOULD compress PPP Protocol IDs in this field that have an upper byte of zero (i.e., Protocol IDs from 0x21 thru 0xFD). This Protocol Field Compression in each PPP subframe is not related to the negotiation of PFC during LCP negotiation which affects the length of PPP Multiplexed Frame Protocol ID.

Information Field:

This field contains the actual packet being encapsulated. Any frame may be included here with the exception of LCP Configure Request, ACK, NAK and Reject frames and PPP Multiplexed frames. If LCP is renegotiated then PPP Multiplexing MUST be disabled until the PPP Mux Control Protocol is negotiated.

1.2 Transmitter procedure

A simple implementation of the transmitter is provided. During the transmission of a multiplexed PPP frame, the transmitter has a state variable, Last_PID, which is used to hold the most recent value of protocol field in a subframe with PFF=1. At the start of the multiplexing process, Last_PID is set equal to the default PID value negotiated in PPPMuxCP. Also, a user configurable parameter, maximum subframe length (MAX_SF_LEN), is used to determine the maximum size of a PPP frame which can be multiplexed. The value of MAX_SF_LEN should be less or equal to the minimum of MRU-2(maximum size of length field) and 16,383 (14 bits).

After transmitting a PPP frame (multiplexed or not) on the channel, the PPP multiplexing logic looks at the buffers that hold the PPP frames to be transmitted. In case there are multiple frames, the PPP multiplexing logic checks if the length of the first frame in the buffer is less than or equal to MAX_SF_LEN bytes. If so, the transmitter starts compiling a multiplexed PPP frame with the protocol field value corresponding to PPP Multiplexed Frame (0x59). For each subframe, the test for deciding to prepend the protocol field to a subframe is to compare the protocol field value of the subframe to Last_PID. If they are equal, PFF is set to 0 and the protocol field is deleted. If not, PFF is set to 1, the protocol field is included, after PFC, in the subframe and Last_PID is set to

the protocol field value of the current subframe. The stopping criteria in the concatenation process are (i) when the length of the next subframe is greater than MAX_SF_LEN bytes or (ii) the length of the entire PPP frame by including the new subframe exceeds the maximum receive unit (MRU) parameter negotiated during LCP [4], or (iii) there are no more subframes to concatenate.

Implementers may choose additionally to implement using timers. In such a case a timeout in addition to the conditions stated above is used as a stopping criteria of the multiplexing process. Moreover, it may be desirable to limit the maximum size of a multiplexed packet to be considerably smaller than MRU for reasons of multiplexing latency and packet error considerations.

1.3 Receiver procedure

If a multiplexed frame, i.e., a frame with Protocol field value equal to PPP Multiplexed Frame (0x0059), is received, the frame is demultiplexed in order using the following input demultiplexing logic. Similar to a transmitter, the receiver has a state variable called Last_rcvd_PID, which is the value of the protocol field in the most recently demultiplexed subframe with PFF=1. Last_rcvd_PID is initialized to default PID value negotiated by PPPMuxCP. If PFF=0 for a subframe, Last_rcvd_PID is appended to the beginning of the subframe before handing the subframe, as determined by the length field, to the PPP logic. If PFF=1 for a subframe, Last_rcvd_PID is set to this value and the subframe, as determined by the length field, is passed to PPP logic. The remainder of the frame is returned to the demultiplexor. Each succeeding subframe is processed similarly. This processing is complete when the remainder of the frame is empty, or when the size field of a subframe exceeds the amount of data remaining in a packet. In the latter case, there is an error either in the length field of the last subframe or in the length field of one of the previous subframes. In either case the last subframe must be dropped by the demultiplexing logic.

It is illegal to put a multiplexed frame within a multiplexed frame.

2. PPP Network Control Protocol for PPP Multiplexing (PPPMuxCP)

A receiver will offer its ability to received multiplexed frames by negotiating NCP for PPP multiplexing, PPPMuxCP. The protocol field value for a PPPMuxCP frames is 0x8059. PPPMuxCP is similar to other NCPs such as IPCP [6]. A transmitter may not send a multiplexed frame unless the peer has offered to receive multiplexed frames. Support of multiplexed frame reception is negotiated in each direction independently. Successful negotiation of PPPMuxCP does not obligate a peer to transmit multiplexed frames.

As part of the PPPMuxCP negotiation, a 'default PID' option is always negotiated. This enables the transmitter to transmit the first subframe of a PPP multiplexed frame without a PID (PFF=0), thus resulting in a saving of one or two bytes. Note that the negotiation of default PID does not require the transmitter to send the first subframe with PFF=0 even if doing so would optimize the transmission. And, as always, the option (and thus the default PID) is negotiated by the receiver, i.e., the receiver will interpret a received PPPmux packet using the default PID it offered.

LCP frames MUST NOT be sent in Multiplexed frames. The only option in PPPMuxCP is the negotiation of Default PID and is shown below

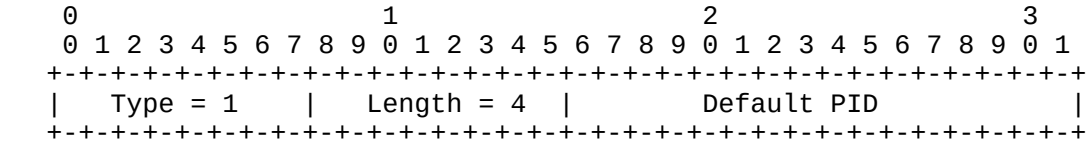

Figure 2. Default PID option for PPPMuxCP

3. Interaction with PPP Multilink (MP) Protocol

PPP multiplexed frame option is negotiated by an NCP. LCP is negotiated over each member link of a multilink bundle and not on the bundle itself [5]. Thus in case of MP, PPPmux cannot be negotiated for individual links, but only for the bundle.

Hence, on the transmitter side PPP multiplexing always occurs before multilink PPP encapsulation. On a link, an MP header (if present) MUST be outside of a PPPmux header (if present). Multilink frames must not be sent in Multiplexed frames.

4. Interaction with CCP and ECP

PPP multiplexing must be performed below (after) any bundle-level CCP and/or ECP, and above (before) MP and any per-link CCP and/or ECP. Thus, to negotiate the hypothetical transmit path sequence CCP -> PPPMux -> ECP, the bundle-level version of CCP (80fd) and the per-link version of ECP (8055) are negotiated along with the PPPMux Option.

An implementation that cannot perform PPPMux above CCP or ECP MUST issue Protocol-Reject for the per-link forms of CCP and ECP if PPPMux has been negotiated.

5. Security Considerations

This document does not impose additional security considerations beyond those that apply to PPP and header-compression schemes over PPP.

6. Acknowledgements

The authors would like to thank contributors on the PPPext mailing list, especially James Carlson, for valuable inputs to this document.

7. References

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

[2] Simpson, W., Ed., "PPP LCP extensions", RFC 1570, January, 1994.

[3] Simpson, W., Ed., "PPP in HDLC-like Framing", STD 51, RFC 1662, July 1994.

[4] Simpson, W., Ed., "The Point-To-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.

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

[6] McGregor, G., "The PPP Internet Protocol Control Protocol (IPCP)", RFC 1332, May 1992.

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

8. Author's Addresses

Rajesh Pazhyannur
Motorola, Network Solutions Sector
1501, W. Shure Drive
Arlington Heights, IL 60004

Phone: (847) 632-4524
EMail: pazhynnr@cig.mot.com

Irfan Ali
Motorola, Network Solutions Sector
1501, W. Shure Drive
Arlington Heights, IL 60004

Phone: (847) 632-3281
EMail: fia225@email.mot.com

Craig Fox
Cisco Systems
170 W. Tasman Street
San Jose, CA 95134

Phone: (408) 526-6296
EMail: fox@cisco.com

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.