

39756

Network Working Group
Request for Comments: 1990
Obsoletes: 1717
Category: Standards Track

K. Sklower
University of California, Berkeley
B. Lloyd
G. McGregor
Lloyd Internetworking
D. Carr
Newbridge Networks Corporation
T. Coradetti
Sidewalk Software
August 1996

The PPP Multilink Protocol (MP)

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

This document proposes a method for splitting, recombining and sequencing datagrams across multiple logical data links. This work was originally motivated by the desire to exploit multiple bearer channels in ISDN, but is equally applicable to any situation in which multiple PPP links connect two systems, including async links. This is accomplished by means of new PPP [2] options and protocols.

The differences between the current PPP Multilink specification (RFC 1717) and this memo are explained in Section 11. Any system implementing the additional restrictions required by this memo will be backwards compatible with conforming RFC 1717 implementations.

Acknowledgements

The authors specifically wish to thank Fred Baker of ACC, Craig Fox of Network Systems, Gerry Meyer of Spider Systems, Dan Brennan of Penril Datability Networks, Vernon Schryver of SGI (for the comprehensive discussion of padding), and the members of the IP over Large Public Data Networks and PPP Extensions working groups, for much useful discussion on the subject.

Table of Contents

1. Introduction	2
1.1. Motivation	2
1.2. Functional Description	3
1.3. Conventions	4
2. General Overview	4
3. Packet Formats	7
3.1. Padding Considerations	10
4. Trading Buffer Space Against Fragment Loss	10
4.1. Detecting Fragment Loss	11
4.2. Buffer Space Requirements	12
5. PPP Link Control Protocol Extensions	13
5.1. Configuration Option Types	13
5.1.1. Multilink MRRU LCP option	14
5.1.2. Short Sequence Number Header Format Option	15
5.1.3. Endpoint Discriminator Option	15
6. Initiating use of Multilink Headers	19
7. Closing Member links	20
8. Interaction with Other Protocols	20
9. Security Considerations	21
10. References	21
11. Differences from RFC 1717	22
11.1. Negotiating Multilink, per se	22
11.2. Initial Sequence Number defined	22
11.3. Default Value of the MRRU	22
11.4. Config-Nak of EID prohibited	22
11.5. Uniformity of Sequence Space	22
11.6. Commencing and Abating use of Multilink Headers	23
11.7. Manual Configuration and Bundle Assignment	23
12. Authors' Addresses	24

1. Introduction

1.1. Motivation

Basic Rate and Primary Rate ISDN both offer the possibility of opening multiple simultaneous channels between systems, giving users additional bandwidth on demand (for additional cost). Previous proposals for the transmission of internet protocols over ISDN have stated as a goal the ability to make use of this capability, (e.g., Leifer et al., [1]).

There are proposals being advanced for providing synchronization between multiple streams at the bit level (the BONDING proposals); such features are not as yet widely deployed, and may require additional hardware for end system. Thus, it may be useful to have a purely software solution, or at least an interim measure.

There are other instances where bandwidth on demand can be exploited, such as using a dialup async line at 28,800 baud to back up a leased synchronous line, or opening additional X.25 SVCs where the window size is limited to two by international agreement.

The simplest possible algorithms of alternating packets between channels on a space available basis (which might be called the Bank Teller's algorithm) may have undesirable side effects due to reordering of packets.

By means of a four-byte sequencing header, and simple synchronization rules, one can split packets among parallel virtual circuits between systems in such a way that packets do not become reordered, or at least the likelihood of this is greatly reduced.

1.2. Functional Description

The method discussed here is similar to the multilink protocol described in ISO 7776 [4], but offers the additional ability to split and recombine packets, thereby reducing latency, and potentially increase the effective maximum receive unit (MRU). Furthermore, there is no requirement here for acknowledged-mode operation on the link layer, although that is optionally permitted.

Multilink is based on an LCP option negotiation that permits a system to indicate to its peer that it is capable of combining multiple physical links into a "bundle". Only under exceptional conditions would a given pair of systems require the operation of more than one bundle connecting them.

Multilink is negotiated during the initial LCP option negotiation. A system indicates to its peer that it is willing to do multilink by sending the multilink option as part of the initial LCP option negotiation. This negotiation indicates three things:

1. The system offering the option is capable of combining multiple physical links into one logical link;
2. The system is capable of receiving upper layer protocol data units (PDU) fragmented using the multilink header (described later) and reassembling the fragments back into the original PDU for processing;
3. The system is capable of receiving PDUs of size N octets where N is specified as part of the option even if N is larger than the maximum receive unit (MRU) for a single physical link.

Once multilink has been successfully negotiated, the sending system is free to send PDUs encapsulated and/or fragmented with the multilink header.

1.3. Conventions

The following language conventions are used in the items of specification in this document:

- o MUST, SHALL or MANDATORY -- the item is an absolute requirement of the specification.
- o SHOULD or RECOMMENDED -- the item should generally be followed for all but exceptional circumstances.
- o MAY or OPTIONAL -- the item is truly optional and may be followed or ignored according to the needs of the implementor.

2. General Overview

In order to establish communications over a point-to-point link, each end of the PPP link must first send LCP packets to configure the data link during Link Establishment phase. After the link has been established, PPP provides for an Authentication phase in which the authentication protocols can be used to determine identifiers associated with each system connected by the link.

The goal of multilink operation is to coordinate multiple independent links between a fixed pair of systems, providing a virtual link with greater bandwidth than any of the constituent members. The aggregate link, or bundle, is named by the pair of identifiers for two systems connected by the multiple links. A system identifier may include information provided by PPP Authentication [3] and information provided by LCP negotiation. The bundled links can be different physical links, as in multiple async lines, but may also be instances of multiplexed links, such as ISDN, X.25 or Frame Relay. The links may also be of different kinds, such as pairing dialup async links with leased synchronous links.

We suggest that multilink operation can be modeled as a virtual PPP link-layer entity wherein packets received over different physical link-layer entities are identified as belonging to a separate PPP network protocol (the Multilink Protocol, or MP) and recombined and sequenced according to information present in a multilink fragmentation header. All packets received over links identified as belonging to the multilink arrangement are presented to the same

network-layer protocol processing machine, whether they have multilink headers or not.

Sklower, et. al.

Standards Track

[Page 4]

T

RFC 1990

PPP Multilink

August 1996

The packets to be transmitted using the multilink procedure are encapsulated according to the rules for PPP where the following options would have been manually configured:

- o No async control character Map
- o No Magic Number
- o No Link Quality Monitoring
- o Address and Control Field Compression
- o Protocol Field Compression
- o No Compound Frames
- o No Self-Describing-Padding

According to the rules specified in RFC1661, this means that an implementation MUST accept reassembled packets with and without leading zeroes present in the Protocol Field of the reassembled packet. Although it is explicitly forbidden below to include the Address and Control fields (usually, the two bytes FF 03) in the material to be fragmented, it is a good defensive programming practice to accept the packet anyway, ignoring the two bytes if present, as that is what RFC1661 specifies.

As a courtesy to implementations that perform better when certain alignment obtains, it is suggested that a determination be made when a bundle is created on whether to transmit leading zeroes by examining whether PFC has been negotiated on the first link admitted into a bundle. This determination should be kept in force so long as a bundle persists.

Of course, individual links are permitted to have different settings for these options. As described below, member links SHOULD negotiate Self-Describing-Padding, even though pre-fragmented packets MUST NOT be padded. Since the Protocol Field Compression mode on the member link allows a sending system to include a leading byte of zero or not at its discretion, this is an alternative mechanism for generating even-length packets.

LCP negotiations are not permitted on the bundle itself. An implementation MUST NOT transmit LCP Configure-Request, -Reject, -Ack, -Nak, Terminate-Request or -Ack packets via the multilink procedure, and an implementation receiving them MUST silently discard them. (By "silently discard" we mean to not generate any PPP packets in response; an implementation is free to generate a log entry registering the reception of the unexpected packet). By contrast, other LCP packets having control functions not associated with changing the defaults for the bundle itself are permitted. An

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.