Network Working Group
Request for Comments: 2118
Category: Informational

G. Pall
Microsoft Corporation

March 1997

Microsoft Point-To-Point Compression (MPPC) Protocol

Status of this Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Abstract

The Point-to-Point Protocol (PPP) [1] provides a standard method for transporting multi-protocol datagrams over point-to-point links.

The PPP Compression Control Protocol [2] provides a method to negotiate and utilize compression protocols over PPP encapsulated links.

This document describes the use of the Microsoft Point to Point Compression protocol (also referred to as MPPC in this document) for compressing PPP encapsulated packets.

Table of Contents

1. Introduction

The Microsoft Point to Point Compression scheme is a means of representing arbitrary Point to Point Protocol (PPP) packets in a compressed form. The MPPC algorithm is designed to optimize processor utilization and bandwidth utilization in order to support large number of simultaneous connections. The MPPC algorithm is also optimized to work efficiently in typical PPP scenarios (1500 byte MTU, etc.).

The MPPC algorithm uses an LZ [3] based algorithm with a sliding window history buffer.

The MPPC algorithm keeps a continous history so that after 8192 bytes of data has been transmitted compressed there is always 8192 bytes of history to use for compressing, except when the history is flushed.

1.1. Licensing

MPPC can only be used in products that implement the Point to Point Protocol AND for the sole purpose of interoperating with other MPPC and Point to Point Protocol implementations.

Source and object licenses are available on a non-discriminatory basis from Stac Electronics. Please contact:

Cheryl Poland
Stac Electronics
12636 High Bluff Drive,
San Deigo, CA 92130
Phone: (619)794-4534
Email: cherylp@stac.com

1.2. Specification of Requirements

In this document, several words are used to signify the requirements of the specification. These words are often capitalized.

MUST This word, or the adjective "required", means that the definition is an absolute requirement of the specification.

MUST NOT This phrase means that the definition is an absolute prohibition of the specification.

SHOULD This word, or the adjective "recommended", means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications MUST be understood and carefully weighed before choosing a different course.

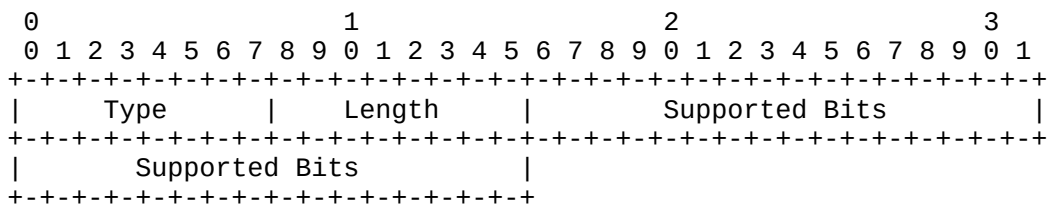
MAY This word, or the adjective "optional", means that this item is one of an allowed set of alternatives. An implementation which does not include this option MUST be prepared to interoperate with another implementation which does include the option.

2. Configuration Option Format

Description

The CCP Configuration Option negotiates the use of MPPC on the link. By default or ultimate disagreement, no compression is used.

A summary of the CCP Configuration Option format is shown below. The fields are transmitted from left to right.



Type

18

Length

6

Supported Bits

This field is 4 octets, most significant octet first. The least significant bit in the least significant octet set to 1 indicates desire to negotiate MPPC.

All other bits MUST be set to 0.

3. MPPC Packets

Before any MPPC packets may be communicated, PPP must reach the Network-Layer Protocol phase, and the CCP Control Protocol must reach the Opened state.

Exactly one MPPC datagram is encapsulated in the PPP Information field. The PPP Protocol field indicates type hex 00FD for all compressed datagrams.

The maximum length of the MPPC datagram transmitted over a PPP link is the same as the maximum length of the Information field of a PPP encapsulated packet. Since the history buffer is limited to 8192 bytes, this length cannot be greater than 8192 bytes.

Only packets with PPP Protocol numbers in the range hex 0021 to hex 00FA are compressed. Other packets are not passed thru the MPPC processor and are sent with their original PPP Protocol numbers.

Padding

It is recommended that padding not be used with MPPC since it defeats the purpose of compression. If the sender must use padding it MUST negotiate the Self-Describing-Padding Configuration option during LCP phase and use self-describing pads.

Reliability and Sequencing

The MPPC scheme does not require a reliable link. Instead, it relies on a 12 bit coherency count in each packet to keep the history buffers synchronized. If the receiver recognizes that the coherency count received in the packet does not match the count it is expecting, it sends a CCP Reset-Request packet to resynchronize its history buffer with the sender’s history buffer.

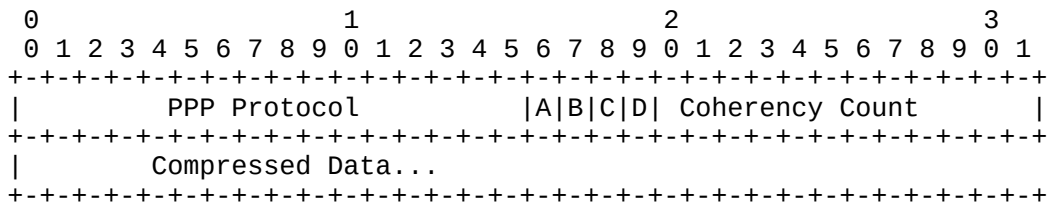
MPPC expects the packets to be delivered in sequence, otherwise history buffer re-synchronization will not occur.

MPPC MAY be used over a reliable link, as described in "PPP Reliable Transmision" [5], but this typically just adds unnecessary overhead since only the coherency count is required.

Data Expansion

If compressing the data results in data expansion, the original data is sent as an uncompressed MPPC packet. The sender must flush the history before compressing any more data and set the FLUSHED bit on the next outgoing packet.

3.1. Packet Format



PPP Protocol

The PPP Protocol field is described in the Point-to-Point Protocol Encapsulation [1].

When the MPPC compression protocol is successfully negotiated by the PPP Compression Control Protocol, the value is hex 00FD. This value MAY be compressed when Protocol-Field-Compression is negotiated.

Bit A

This bit indicates that the history buffer has just been initialized before this packet was generated. This packet can ALWAYS be decompressed because it is not based on any previous history. This bit is typically sent to inform the peer that the sender has initialized its history buffer before compressing the packet and that the receiving peer must initialize its history buffer before decompressing the packet. This bit is referred to as FLUSHED bit in this document.

Implementation Note: Compression and decompression histories are always initialized with all zeroes.

Bit B

This bit indicates that the packet was moved to the front of the history buffer typically because there was no room at the end of the history buffer. This bit is used to tell the decompressor to set its history pointer to the beginning of the history buffer.

Implementation Notes:

1. It is implied that this bit must be set at least once for every 8192 bytes of data that is sent compressed.
2. It is also implied that this bit can be set even if the sender's history buffer is not full. Initialized history that has not been used for compressing data must not be referred to in the compressed packets.

DOCKET ALARM

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.