PPP/IPCP Extension for Word Alignment of IP Datagrams

Status of this Memo

This document is an Internet-Draft. 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."

To learn the current status of any Internet-Draft, please check the "id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Abstract

The Point-to-Point Protocol (PPP) [1] provides a standard method for transporting multi-protocol datagrams over point-to-point links. The PPP/IPCP extension for 32 bit Alignment of IP datagrams provides a method to negotiate and align IP datagrams on a 32 bit boundary. This document describes the use of the Internet Protocol Control Protocol (IPCP) [2] option and the PPP framing that is required for alignment of all IP datagrams.

Table of Contents

1. Introduction ............................................. 2
1.1. Specification of Requirements ....................... 2
2.1 Configuration Option Format ........................... 3
3.1 Frame Format .......................................... 4
SECURITY CONSIDERATIONS ..................................... 5
REFERENCES .................................................. 5
CHAIR’S ADDRESS ............................................ 5
AUTHORS’ ADDRESS ........................................... 5
1. Introduction

Many processors today are 32 bit word bound. This is especially true for processors like the TI TMS320C3X. Other processors are 16 bit word bound like the Motorola 68000. Today with the ever increasing use of microcontrollers that are used to convey IP datagrams to the home in consumer electronic devices, it is becoming more and more important to leave an IP datagram aligned on a 32 bit boundary for performance. The cost of building these consumer electronic devices is so sensitive that the minimum amount of RAM, ROM and processing power are used. For these devices an addition to the IPCP protocol needed to be defined that will add a simple padding character to the data stream to allow the IP datagram to remain on a word bit boundary. The proposed solution is designed to require the minimum overhead in code, ram, and processing requirements to achieve the goal of leaving the IP datagram in it’s original word boundary state. The effects of compression on the packet (like Protocol Field, Address/Control field and Van-Jacobson compression), cause the microcontroller to either spend a lot of time processing the packet in order to get it aligned or force the microcontroller not to gain the benefits of these compression techniques.

1.1. Specification of Requirements

A new IPCP option SHALL be added to negotiate the desire to have IP datagrams aligned on a negotiated octet boundary. Upon successful negotiation of the IP Padding Option, the Acknowledger of the request SHALL pad prior to the IP datagram with zero to the negotiated alignment value - 1 octets with the negotiated pad character for all PPP frames containing an IP datagram. These MUST be identified by the protocol field containing the following values:

0021 IP Datagram
002D VJ Compressed TCP/IP Datagram
002E Uncompressed TCP/IP Datagram

The implementation MAY be extended to provide for other compressed IP Datagrams not yet defined.

The requesting system SHOULD discard the pad character(s) when passing the datagram to the upper layers. The requesting system MAY also check the IP Datagram to verify that it is on the word boundary that was requested.
2.1. IPCP Option Negotiation

A new negotiation option is added for 32 bit padding. If the receiver of such an option acknowledges this option then the receiver SHALL send all IP datagrams with the padding character sufficient to force the IP datagram (after unstuffing and decompression) to a 32 bit aligned value within the frame.

### IP Padding Option

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Alignment</th>
<th>Pad Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>64 (0x40)</td>
<td>0x04</td>
<td>0x04</td>
<td>0xFE</td>
</tr>
</tbody>
</table>

**Type**

64 (0x40)

**Length**

4

**Alignment**

Number of octets to align to:
1. No Alignment
2. 16 bit Alignment
4. 32 Bit Alignment
8. 64 Bit Alignment
16. 128 bit Alignment

**Pad Character**

The character to be used for padding

An example for 32 bit alignment using 0xFE for the PAD Character

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0x40</th>
<th>0x04</th>
<th>0x04</th>
<th>0xFE</th>
</tr>
</thead>
</table>
3.1. Frame Format

If the IP Padding Option is successfully negotiated, then the acknowledger of that option SHALL send all IP datagrams with the following format. This will allow the system requesting the IP Padding Option to assume that all IP datagrams before passing to the network layer will be aligned to a negotiated boundary.

The PPP frame is padded following the Protocol Field so that the beginning of the IP datagram is placed on the requested word boundary after all unstuffing and decompression take place. The pad character to be used SHALL be the one negotiated in the IPCP option 64.

Examples of Padding taking place with 32 bit alignment and the pad character being 0xFE.

1. Protocol Field Compression

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      FF       |       03      |       21      |    FE (PAD)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      IP Datagram                                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

2. Protocol & Address/Control Field Compression

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      21       |    FE (PAD)   |    FE (PAD)   |    FE (PAD)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      IP Datagram                                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

3. Van-Jacobson Compression

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      FF       |      03       |       00      |       2D      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      FE (PAD) |      VJ Compressed TCP/IP Header              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| TCP Payload...                                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
Security Considerations

Security issues are not discussed in this memo.

References


Chair’s Address

The working group can be contacted via the current chair:

Karl F. Fox
Ascend Communications
3518 Riverside Dr., Suite 101
Columbus, Ohio 43221
(614) 451-1883
EMail: karl@ascend.com

Author’s Address

Questions about this memo can also be directed to:

Paul S. Martin
Pacific Softworks Inc.
4000 Via Pescador
Camarillo, CA 93012
(805) 484-2128
Email: paulmart@microsoft.com