

Network Working Group  
Request for Comments: 1332  
Obsoletes: RFC 1172

G. McGregor  
Merit  
May 1992

## The PPP Internet Protocol Control Protocol (IPCP)

### Status of this Memo

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

### Abstract

The Point-to-Point Protocol (PPP) [1] provides a standard method of encapsulating Network Layer protocol information over point-to-point links. PPP also defines an extensible Link Control Protocol, and proposes a family of Network Control Protocols (NCPs) for establishing and configuring different network-layer protocols.

This document defines the NCP for establishing and configuring the Internet Protocol [2] over PPP, and a method to negotiate and use Van Jacobson TCP/IP header compression [3] with PPP.

This RFC is a product of the Point-to-Point Protocol Working Group of the Internet Engineering Task Force (IETF).

## Table of Contents

1.	Introduction .....	1
2.	A PPP Network Control Protocol (NCP) for IP .....	2
2.1	Sending IP Datagrams .....	2
3.	IPCP Configuration Options .....	4
3.1	IP-Addresses .....	5
3.2	IP-Compression-Protocol .....	6
3.3	IP-Address .....	8
4.	Van Jacobson TCP/IP header compression .....	9
4.1	Configuration Option Format .....	9
	APPENDICES .....	11
A.	IPCP Recommended Options .....	11
	SECURITY CONSIDERATIONS .....	11
	REFERENCES .....	11
	ACKNOWLEDGEMENTS .....	11
	CHAIR'S ADDRESS .....	12
	AUTHOR'S ADDRESS .....	12

## 1. Introduction

PPP has three main components:

1. A method for encapsulating datagrams over serial links.
2. A Link Control Protocol (LCP) for establishing, configuring, and testing the data-link connection.
3. A family of Network Control Protocols (NCPs) for establishing and configuring different network-layer protocols.

In order to establish communications over a point-to-point link, each end of the PPP link must first send LCP packets to configure and test the data link. After the link has been established and optional facilities have been negotiated as needed by the LCP, PPP must send NCP packets to choose and configure one or more network-layer protocols. Once each of the chosen network-layer protocols has been configured, datagrams from each network-layer protocol can be sent over the link.

The link will remain configured for communications until explicit LCP or NCP packets close the link down, or until some external event occurs (an inactivity timer expires or network administrator intervention).

## 2. A PPP Network Control Protocol (NCP) for IP

The IP Control Protocol (IPCP) is responsible for configuring, enabling, and disabling the IP protocol modules on both ends of the point-to-point link. IPCP uses the same packet exchange mechanism as the Link Control Protocol (LCP). IPCP packets may not be exchanged until PPP has reached the Network-Layer Protocol phase. IPCP packets received before this phase is reached should be silently discarded.

The IP Control Protocol is exactly the same as the Link Control Protocol [1] with the following exceptions:

### Data Link Layer Protocol Field

Exactly one IPCP packet is encapsulated in the Information field of PPP Data Link Layer frames where the Protocol field indicates type hex 8021 (IP Control Protocol).

### Code field

Only Codes 1 through 7 (Configure-Request, Configure-Ack, Configure-Nak, Configure-Reject, Terminate-Request, Terminate-Ack and Code-Reject) are used. Other Codes should be treated as unrecognized and should result in Code-Rejects.

### Timeouts

IPCP packets may not be exchanged until PPP has reached the Network-Layer Protocol phase. An implementation should be prepared to wait for Authentication and Link Quality Determination to finish before timing out waiting for a Configure-Ack or other response. It is suggested that an implementation give up only after user intervention or a configurable amount of time.

### Configuration Option Types

IPCP has a distinct set of Configuration Options, which are defined below.

#### 2.1. Sending IP Datagrams

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

Exactly one IP packet is encapsulated in the Information field of PPP Data Link Layer frames where the Protocol field indicates type hex 0021 (Internet Protocol).

The maximum length of an IP packet transmitted over a PPP link is the same as the maximum length of the Information field of a PPP data link layer frame. Larger IP datagrams must be fragmented as necessary. If a system wishes to avoid fragmentation and reassembly, it should use the TCP Maximum Segment Size option [4], and MTU discovery [5].

### 3. IPCP Configuration Options

IPCP Configuration Options allow negotiatiation of desirable Internet Protocol parameters. IPCP uses the same Configuration Option format defined for LCP [1], with a separate set of Options.

The most up-to-date values of the IPCP Option Type field are specified in the most recent "Assigned Numbers" RFC [6]. Current values are assigned as follows:

- 1 IP-Addresses
- 2 IP-Compression-Protocol
- 3 IP-Address

### 3.1. IP-Addresses

#### Description

The use of the Configuration Option IP-Addresses has been deprecated. It has been determined through implementation experience that it is difficult to ensure negotiation convergence in all cases using this option. RFC 1172 [7] provides information for implementations requiring backwards compatibility. The IP-Address Configuration Option replaces this option, and its use is preferred.

This option SHOULD NOT be sent in a Configure-Request if a Configure-Request has been received which includes either an IP-Addresses or IP-Address option. This option MAY be sent if a Configure-Reject is received for the IP-Address option, or a Configure-Nak is received with an IP-Addresses option as an appended option.

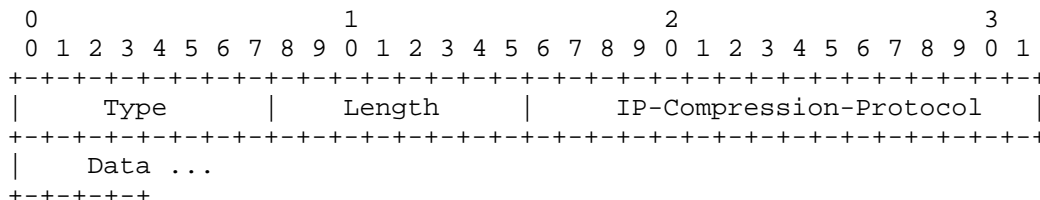
Support for this option MAY be removed after the IPCP protocol status advances to Internet Draft Standard.

3.2. IP-Compression-Protocol

Description

This Configuration Option provides a way to negotiate the use of a specific compression protocol. By default, compression is not enabled.

A summary of the IP-Compression-Protocol Configuration Option format is shown below. The fields are transmitted from left to right.



Type

2

Length

>= 4

IP-Compression-Protocol

The IP-Compression-Protocol field is two octets and indicates the compression protocol desired. Values for this field are always the same as the PPP Data Link Layer Protocol field values for that same compression protocol.

The most up-to-date values of the IP-Compression-Protocol field are specified in the most recent "Assigned Numbers" RFC [6]. Current values are assigned as follows:

Value (in hex)	Protocol
002d	Van Jacobson Compressed TCP/IP

Data

The Data field is zero or more octets and contains additional data as determined by the particular compression protocol.



Default

No compression protocol enabled.

### 3.3. IP-Address

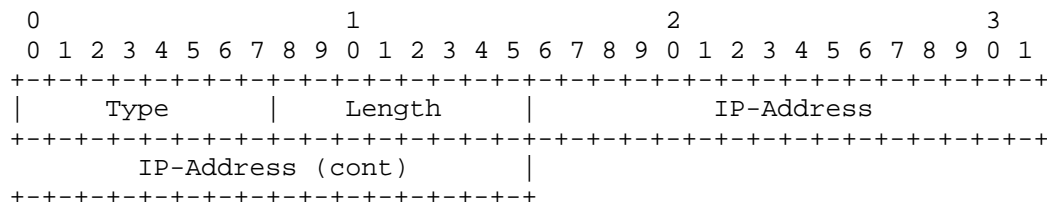
#### Description

This Configuration Option provides a way to negotiate the IP address to be used on the local end of the link. It allows the sender of the Configure-Request to state which IP-address is desired, or to request that the peer provide the information. The peer can provide this information by NAKing the option, and returning a valid IP-address.

If negotiation about the remote IP-address is required, and the peer did not provide the option in its Configure-Request, the option SHOULD be appended to a Configure-Nak. The value of the IP-address given must be acceptable as the remote IP-address, or indicate a request that the peer provide the information.

By default, no IP address is assigned.

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



Type

3

Length

6

IP-Address

The four octet IP-Address is the desired local address of the sender of a Configure-Request. If all four octets are set to zero, it indicates a request that the peer provide the IP-Address information.

Default

No IP address is assigned.

4. Van Jacobson TCP/IP header compression

Van Jacobson TCP/IP header compression reduces the size of the TCP/IP headers to as few as three bytes. This can be a significant improvement on slow serial lines, particularly for interactive traffic.

The IP-Compression-Protocol Configuration Option is used to indicate the ability to receive compressed packets. Each end of the link must separately request this option if bi-directional compression is desired.

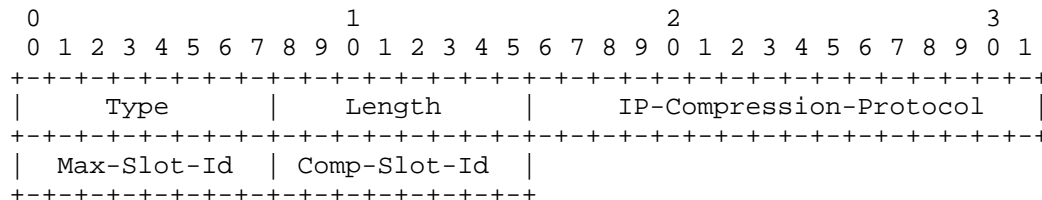
The PPP Protocol field is set to the following values when transmitting IP packets:

Value (in hex)

- 0021 Type IP. The IP protocol is not TCP, or the packet is a fragment, or cannot be compressed.
- 002d Compressed TCP. The TCP/IP headers are replaced by the compressed header.
- 002f Uncompressed TCP. The IP protocol field is replaced by the slot identifier.

4.1. Configuration Option Format

A summary of the IP-Compression-Protocol Configuration Option format to negotiate Van Jacobson TCP/IP header compression is shown below. The fields are transmitted from left to right.



Type

2

Length

6

### IP-Compression-Protocol

002d (hex) for Van Jacobson Compressed TCP/IP headers.

### Max-Slot-Id

The Max-Slot-Id field is one octet and indicates the maximum slot identifier. This is one less than the actual number of slots; the slot identifier has values from zero to Max-Slot-Id.

Note: There may be implementations that have problems with only one slot (Max-Slot-Id = 0). See the discussion in reference [3]. The example implementation in [3] will only work with 3 through 254 slots.

### Comp-Slot-Id

The Comp-Slot-Id field is one octet and indicates whether the slot identifier field may be compressed.

- 0 The slot identifier must not be compressed. All compressed TCP packets must set the C bit in every change mask, and must include the slot identifier.
- 1 The slot identifier may be compressed.

The slot identifier must not be compressed if there is no ability for the PPP link level to indicate an error in reception to the decompression module. Synchronization after errors depends on receiving a packet with the slot identifier. See the discussion in reference [3].

#### A. IPCP Recommended Options

The following Configurations Options are recommended:

IP-Compression-Protocol -- with at least 4 slots, usually 16 slots.

IP-Address -- only on dial-up lines.

#### Security Considerations

Security issues are not discussed in this memo.

#### References

- [1] Simpson, W., "The Point-to-Point Protocol", RFC 1331, May 1992.
- [2] Postel, J., "Internet Protocol", RFC 791, USC/Information Sciences Institute, September 1981.
- [3] Jacobson, V., "Compressing TCP/IP Headers", RFC 1144, January 1990.
- [4] Postel, J., "The TCP Maximum Segment Size Option and Related Topics", RFC 879, USC/Information Sciences Institute, November 1983.
- [5] Mogul, J., and S. Deering, "Path MTU Discovery", RFC 1191, November 1990.
- [6] Reynolds, J., and J. Postel, "Assigned Numbers", RFC 1060, USC/Information Sciences Institute, March 1990.
- [7] Perkins, D., and R. Hobby, "Point-to-Point Protocol (PPP) initial configuration options", RFC 1172, August 1990.

#### Acknowledgments

Some of the text in this document is taken from RFCs 1171 & 1172, by Drew Perkins of Carnegie Mellon University, and by Russ Hobby of the University of California at Davis.

Information leading to the expanded IP-Compression option provided by Van Jacobson at SIGCOMM '90.

Bill Simpson helped with the document formatting.

#### Chair's Address

The working group can be contacted via the current chair:

Brian Lloyd  
Lloyd & Associates  
3420 Sudbury Road  
Cameron Park, California 95682

Phone: (916) 676-1147

EMail: brian@ray.lloyd.com

#### Author's Address

Questions about this memo can also be directed to:

Glenn McGregor  
Merit Network, Inc.  
1071 Beal Avenue  
Ann Arbor, MI 48109-2103

Phone: (313) 763-1203

EMail: Glenn.McGregor@Merit.edu