Network Working Group
Request for Comments: 2023
Category: Standards Track

D. Haskin
E. Allen
Bay Networks, Inc.
October 1996

IP Version 6 over PPP

Status of this Memo

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 method for transmission of IP Version 6 [2] packets over PPP links as well as the Network Control Protocol (NCP) for establishing and configuring the IPv6 over PPP. It also specifies the method of forming IPv6 link-local addresses on PPP links.

Table of Contents

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.

In this document, the NCP for establishing and configuring the IPv6 over PPP is referred as the IPv6 Control Protocol (IPV6CP).

The link will remain configured for communications until explicit LCP or NCP packets close the link down, or until some external event occurs (power failure at the other end, carrier drop, etc.).

1.1. 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 inter-operate with another implementation which does include the option.

2. Sending IPv6 Datagrams

Before any IPv6 packets may be communicated, PPP must reach the Network-Layer Protocol phase, and the IPv6 Control Protocol must reach the Opened state.

Exactly one IPv6 packet is encapsulated in the Information field of PPP Data Link Layer frames where the Protocol field indicates type hex 0057 (Internet Protocol Version 6).

The maximum length of an IPv6 packet transmitted over a PPP link is the same as the maximum length of the Information field of a PPP data link layer frame. PPP links supporting IPv6 must allow at least 576 octets in the information field of a data link layer frame.

3. A PPP Network Control Protocol for IPv6

The IPv6 Control Protocol (IPV6CP) is responsible for configuring, enabling, and disabling the IPv6 protocol modules on both ends of the point-to-point link. IPV6CP uses the same packet exchange mechanism as the Link Control Protocol (LCP). IPV6CP packets may not be exchanged until PPP has reached the Network-Layer Protocol phase. IPV6CP packets received before this phase is reached should be silently discarded.

The IPv6 Control Protocol is exactly the same as the Link Control Protocol [1] with the following exceptions:

Data Link Layer Protocol Field

Exactly one IPV6CP packet is encapsulated in the Information field of PPP Data Link Layer frames where the Protocol field indicates type hex 8057 (IPv6 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

IPV6CP 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

IPV6CP has a distinct set of Configuration Options, which are defined below.

4. IPV6CP Configuration Options

IPV6CP Configuration Options allow negotiation of desirable IPv6 parameters. IPV6CP uses the same Configuration Option format defined for LCP [1], with a separate set of Options. If a Configuration Option is not included in a Configure-Request packet, the default value for that Configuration Option is assumed.

Up-to-date values of the IPV6CP Option Type field are specified in the most recent "Assigned Numbers" RFC [5]. Current values are assigned as follows:

1 Interface-Token
2 IPv6-Compression-Protocol

4.1. Interface-Token

Description

This Configuration Option provides a way to negotiate a unique 32-bit interface token to be used for the address autoconfiguration [3] at the local end of the link (see section 5). The interface token MUST be unique within the PPP link; i.e. upon completion of the negotiation different Interface-Token values are to be selected for the ends of the PPP link.

Before this Configuration Option is requested, an implementation must choose its tentative Interface-Token. It is recommended that a non-zero value be chosen in the most random manner possible in order to guarantee with very high probability that an implementation will arrive at a unique token value. A good way to choose a unique random number is to start with a unique seed. Suggested sources of uniqueness include machine serial numbers,

other network hardware addresses, system clocks, etc. Note that it may not be sufficient to use a link-layer address alone as the seed, since it will not always be unique. Thus it is suggested that the seed should be calculated from a variety of sources that are likely to be different even on identical systems and as many sources as possible be used simultaneously. Good sources of uniqueness or randomness are required for the Interface-Token negotiation to succeed. If a good source of randomness cannot be found, it is recommended that a zero value be used for the Interface-Token transmitted in the Configure-Request. In this case the PPP peer may provide a valid non-zero Interface-Token in its response as described below. Note that if at least one of the PPP peers is able to generate a unique random number, the token negotiation will succeed.

When a Configure-Request is received with the Interface-Token Configuration Option and the receiving peer implements this option, the received Interface-Token is compared with the Interface-Token of the last Configure-Request sent to the peer. Depending on the result of the comparison an implementation MUST respond in one of the following ways:

If the two Interface-Tokens are different but the received Interface-Token is zero, a Configure-Ack is sent with a non-zero Interface-Token value suggested for use by the remote peer. Such a suggested Interface-Token MUST be different from the Interface-Token of the last Configure-Request sent to the peer.

If the two Interface-Tokens are different and the received Interface-Token is not zero, the Interface-Token MUST be acknowledged, i.e. a Configure-Ack is sent with the requested Interface-Token, meaning that the responding peer agrees with the Interface-Token requested.

If the two Interface-Tokens are equal and are not zero, a Configure-Nak MUST be sent specifying a different non-zero Interface-Token value suggested for use by the remote peer.

If the two Interface-Tokens are equal to zero, the Interface-Tokens negotiation MUST be terminated by transmitting the Configure-Reject with the Interface-Token value set to zero. In this case a unique Interface-Token can not be negotiated.

If a Configure-Request is received with the Interface-Token Configuration Option and the receiving peer does not implement this option, Configure-Rej is sent.

A new Configure-Request SHOULD NOT be sent to the peer until normal processing would cause it to be sent (that is, until a Configure-Nak is received or the Restart timer runs out).

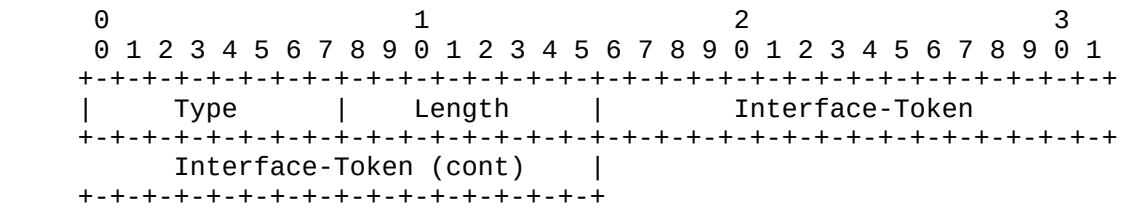
A new Configure-Request MUST NOT contain the Interface-Token option if a valid Interface-Token Configure-Reject is received.

Reception of a Configure-Nak with a suggested Interface-Token different from that of the last Configure-Nak sent to the peer indicates a unique Interface-Token. In this case a new Configure-Request MUST be sent with the token value suggested in the last Configure-Nak from the peer. But if the received Interface-Token is equal to the one sent in the last Configure-Nak, a new Interface-Token MUST be chosen. In this case, a new Configure-Request SHOULD be sent with the new tentative Interface-Token. This sequence (transmit Configure-Request, receive Configure-Request, transmit Configure-Nak, receive Configure-Nak) might occur a few times, but it is extremely unlikely to occur repeatedly. More likely, the Interface-Tokens chosen at either end will quickly diverge, terminating the sequence.

If negotiation about the Interface-Token is required, and the peer did not provide the option in its Configure-Request, the option SHOULD be appended to a Configure-Nak. The tentative value of the Interface-Token given must be acceptable as the remote Interface-Token; i.e. should be different from the token value selected for the local end of the PPP link. The next Configure-Request from the peer may include this option. If the next Configure-Request does not include this option the peer MUST NOT send another Configure-Nak with this option included. It should assume that the peer’s implementation does not support this option.

By default, an implementation SHOULD attempt to negotiate the Interface-Token for its end of the PPP connection.

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



Type

1

Length

6

Interface-Token

The 32-bit Interface-Token which is very likely to be unique on the link or zero if a good source of uniqueness can not be found.

Default Token Value

If no valid interface token can be successfully negotiated, no default Interface-Token value should be assumed. The procedures for recovering from such a case are unspecified. One approach is to manually configure the interface token of the interface.

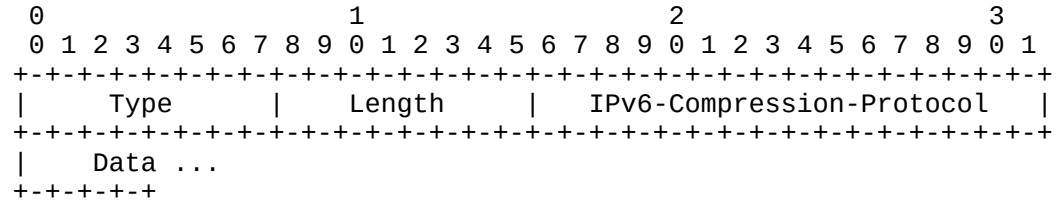
4.2. IPv6-Compression-Protocol

Description

This Configuration Option provides a way to negotiate the use of a specific IPv6 packet compression protocol. The IPv6-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. By default, compression is not enabled.

IPv6 compression negotiated with this option is specific to IPv6 datagrams and is not to be confused with compression resulting from negotiations via Compression Control Protocol (CCP), which potentially effect all datagrams.

A summary of the IPv6-Compression-Protocol Configuration Option format is shown below. The fields are transmitted from left to right.



Type

2

Length

>= 4

IPv6-Compression-Protocol

The IPv6-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.

Up-to-date values of the IPv6-Compression-Protocol field are specified in the most recent "Assigned Numbers" RFC [5].

Current values are assigned as follows:

Value (in hex)	Protocol
004f	IPv6 Header Compression

Data

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

Default

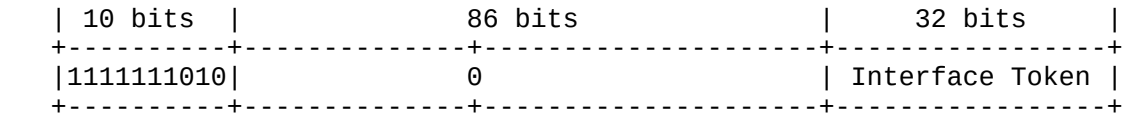
No IPv6 compression protocol enabled.

5. Stateless Autoconfiguration and Link-Local Addresses

The interface token, which is used for forming IPv6 addresses of a PPP interface, SHOULD be negotiated in the IPV6CP phase of the PPP connection setup (see section 4.1). If no valid interface token has been successfully negotiated, procedures for recovering from such a case are unspecified. One approach is to manually configure the interface token of the interface.

As long as the interface token is negotiated in the IPV6CP phase of the PPP connection setup, it is redundant to perform duplicate address detection as a part of the IPv6 Stateless Autoconfiguration protocol [3]. Therefore it is recommended that for PPP links with the IPV6CP Interface-Token option enabled the default value of the DupAddrDetectTransmits autoconfiguration variable [3] be zero.

Link-local addresses of PPP interfaces have the following format:



The most significant 10 bits of the address is the Link-Local prefix FE80::. 86 zero bits pad out the address between the Link-Local prefix and the Interface Token fields.

A. IPV6CP Recommended Options

The following Configurations Options are recommended:

Interface-Token

IPv6-Compression-Protocol

Security Considerations

Security issues are not discussed in this memo.

References

[1] Simpson, W., "The Point-to-Point Protocol", STD 51, RFC 1661, July 1994.

[2] Deering, S., and R. Hinden, Editors, "Internet Protocol, Version 6 (IPv6) Specification", RFC 1883, December 1995.

[2] Hinden, R., and S. Deering, "IP Version 6 Addressing Architecture", RFC 1884, December 1995.

[3] Thomson, S., and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 1971, August 1996.

[4] Narten, T., Nordmark, E., and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 1970, August 1996.

[5] Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994.

Acknowledgments

This document borrows from the Magic-Number LCP option and as such is partially based on previous work done by the PPP working group.

Authors' Addresses

Dimitry Haskin
Bay Networks, Inc.
2 Federal Street
Billerica, MA 01821
email: dhaskin@baynetworks.com

Ed Allen
Bay Networks, Inc.
2 Federal Street
Billerica, MA 01821
email: eallen@baynetworks.com