Network Working Group
Request for Comments: 5072
Obsoletes: 2472
Category: Standards Track

S. Varada, Ed.
Transwitch
D. Haskins
E. Allen

September 2007

IP Version 6 over PPP

Status of This Memo

Abstract

The Point-to-Point Protocol (PPP) 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 sending IPv6 packets over PPP links, the NCP for establishing and configuring the IPv6 over PPP, and the method for forming IPv6 link-local addresses on PPP links.

It also specifies the conditions for performing Duplicate Address Detection on IPv6 global unicast addresses configured for PPP links either through stateful or stateless address autoconfiguration.

This document obsoletes RFC 2472.

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 to 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.).

This document obsoletes the earlier specification from RFC 2472 [7]. Changes from RFC 2472 are listed in Appendix B.

1.1. Specification of Requirements

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

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [6].

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 the information field to be at least as large as the minimum link MTU size required for IPv6 [2].

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 LCP. IPV6CP packets may not be exchanged until PPP has reached the network-layer protocol phase. IPV6CP packets that are received before this phase is reached should be silently discarded.

The IPv6 Control Protocol is exactly the same as the LCP [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.

4. IPV6CP Configuration Options

IPV6CP Configuration Options allow negotiation of desirable IPv6 parameters. IPV6CP uses the same Configuration Option format defined for LCP [1] but 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 online database of "Assigned Numbers" maintained at IANA [9]. The current value assignment is as follows:

1 Interface-Identifier

The only IPV6CP option defined in this document is the interface identifier. Any other IPV6CP configuration options that can be defined over time are to be defined in separate documents.

4.1. Interface Identifier

Description

This Configuration Option provides a way to negotiate a unique, 64-bit interface identifier to be used for the address autoconfiguration [3] at the local end of the link (see Section 5). A Configure-Request MUST contain exactly one instance of the interface-identifier option [1]. The interface identifier MUST be unique within the PPP

link; i.e., upon completion of the negotiation, different interface-identifier values are to be selected for the ends of the PPP link. The interface identifier may also be unique over a broader scope.

Before this Configuration Option is requested, an implementation chooses its tentative interface identifier. The non-zero value of the tentative interface identifier SHOULD be chosen such that the value is unique to the link and, preferably, consistently reproducible across initializations of the IPV6CP finite state machine (administrative Close and reOpen, reboots, etc.). The rationale for preferring a consistently reproducible unique interface identifier to a completely random interface identifier is to provide stability to global scope addresses (see Appendix A) that can be formed from the interface identifier.

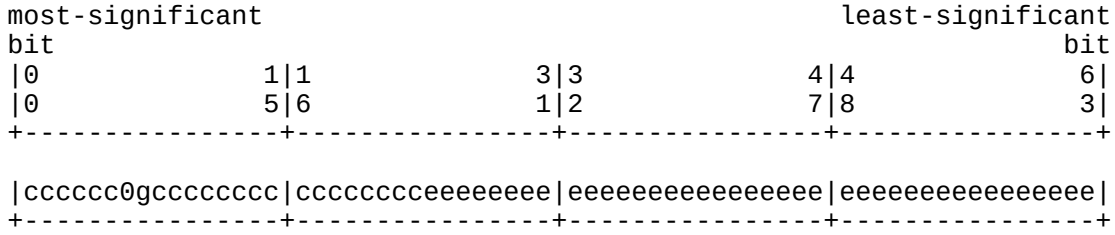
Assuming that interface identifier bits are numbered from 0 to 63 in canonical bit order, where the most significant bit is the bit number 0, the bit number 6 is the "u" bit (universal/local bit in IEEE EUI-64 [4] terminology), which indicates whether or not the interface identifier is based on a globally unique IEEE identifier (EUI-48 or EUI-64 [4])(see case 1 below). It is set to one (1) if a globally unique IEEE identifier is used to derive the interface identifier, and it is set to zero (0) otherwise.

The following are methods for choosing the tentative interface identifier in the preference order:

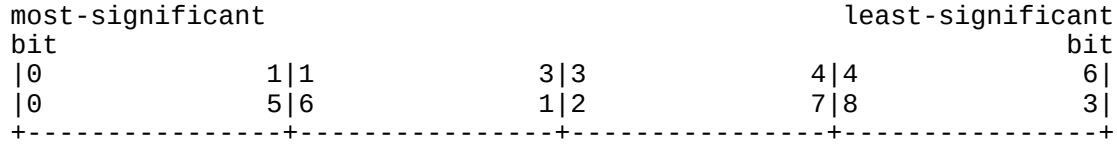
1) If an IEEE global identifier (EUI-48 or EUI-64) is available anywhere on the node, it should be used to construct the tentative interface identifier due to its uniqueness properties. When extracting an IEEE global identifier from another device on the node, care should be taken that the extracted identifier is presented in canonical ordering [14].

   The only transformation from an EUI-64 identifier is to invert the "u" bit (universal/local bit in IEEE EUI-64 terminology).

   For example, for a globally unique EUI-64 identifier of the form:

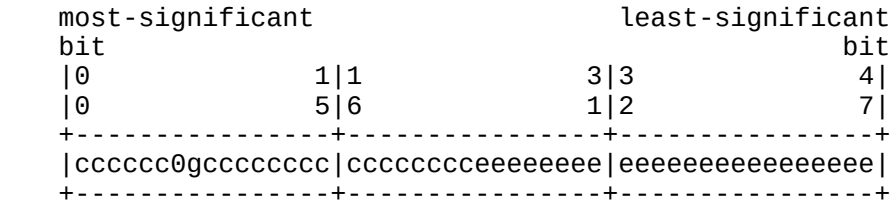


where "c" are the bits of the assigned company_id, "0" is the value of the universal/local bit to indicate global scope, "g" is the group/individual bit, and "e" are the bits of the extension identifier, the IPv6 interface identifier would be of the form:

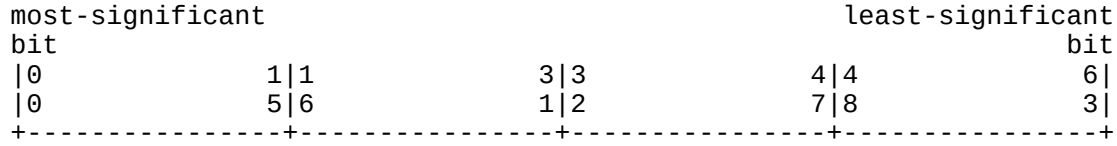


The only change is inverting the value of the universal/local bit.

In the case of a EUI-48 identifier, it is first converted to the EUI-64 format by inserting two bytes, with hexa-decimal values of 0xFF and 0xFE, in the middle of the 48-bit MAC (between the company_id and extension identifier portions of the EUI-48 value). For example, for a globally unique 48-bit EUI-48 identifier of the form:



where "c" are the bits of the assigned company_id, "0" is the value of the universal/local bit to indicate global scope, "g" is the group/individual bit, and "e" are the bits of the extension identifier, the IPv6 interface identifier would be of the form:



2) If an IEEE global identifier is not available, a different source of uniqueness should be used. Suggested sources of uniqueness include link-layer addresses, machine serial numbers, et cetera.

In this case, the "u" bit of the interface identifier MUST be set to zero (0).

3) If a good source of uniqueness cannot be found, it is recommended that a random number be generated. In this case, the "u" bit of the interface identifier MUST be set to zero (0).

Good sources [1] of uniqueness or randomness are required for the interface identifier negotiation to succeed. If neither a unique number nor a random number can be generated, it is recommended that a zero value be used for the interface identifier transmitted in the Configure-Request. In this case, the PPP peer may provide a valid non-zero interface identifier in its response as described below. Note that if at least one of the PPP peers is able to generate separate non-zero numbers for itself and its peer, the identifier negotiation will succeed.

When a Configure-Request is received with the Interface-Identifier Configuration Option and the receiving peer implements this option, the received interface identifier is compared with the interface identifier 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 identifiers are different but the received interface identifier is zero, a Configure-Nak is sent with a non-zero interface-identifier value suggested for use by the remote peer. Such a suggested interface identifier MUST be different from the interface identifier of the last Configure-Request sent to the peer. It is recommended that the value suggested be consistently reproducible across initializations of the IPV6CP finite state machine (administrative Close and reOpen, reboots, etc). The "u" (universal/local) bit of the suggested identifier MUST be set to zero (0) regardless of its source unless the globally unique EUI-48/EUI-64 derived identifier is provided for the exclusive use by the remote peer.

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

If the two interface identifiers are equal and are not zero, Configure-Nak MUST be sent specifying a different non-zero interface-identifier value suggested for use by the remote peer. It is recommended that the value suggested be consistently reproducible across initializations of the IPV6CP finite state machine

(administrative Close and reOpen, reboots, etc). The "u" (universal/local) bit of the suggested identifier MUST be set to zero (0) regardless of its source unless the globally unique EUI-48/EUI-64 derived identifier is provided for the exclusive use by the remote peer.

If the two interface identifiers are equal to zero, the interface identifier's negotiation MUST be terminated by transmitting the Configure-Reject with the interface-identifier value set to zero. In this case, a unique interface identifier cannot be negotiated.

If a Configure-Request is received with the Interface-Identifier Configuration Option and the receiving peer does not implement this option, Configure-Reject 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 [1]).

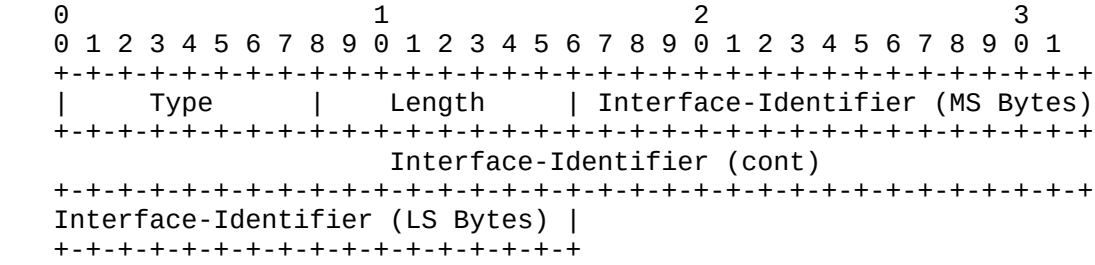
A new Configure-Request MUST NOT contain the interface-identifier option if a valid Interface-Identifier Configure-Reject is received.

Reception of a Configure-Nak with a suggested interface identifier different from that of the last Configure-Nak sent to the peer indicates a unique interface identifier. In this case, a new Configure-Request MUST be sent with the identifier value suggested in the last Configure-Nak from the peer. But if the received interface identifier is equal to the one sent in the last Configure-Nak, a new interface identifier MUST be chosen. In this case, a new Configure-Request SHOULD be sent with the new tentative interface identifier. 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 identifiers chosen at either end will quickly diverge, terminating the sequence.

If negotiation of the interface identifier 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 identifier given must be acceptable as the remote interface identifier; i.e., it should be different from the identifier 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 identifier for its end of the PPP connection.

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



Type

1

Length

10

Interface-Identifier

The 64-bit interface identifier, which is very likely to be unique on the link, or zero if a good source of uniqueness cannot be found.

Default

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

5. Stateless Autoconfiguration and Link-Local Addresses

The interface identifier of IPv6 unicast addresses [5] of a PPP interface SHOULD be negotiated in the IPV6CP phase of the PPP connection setup (see Section 4.1). If no valid interface identifier has been successfully negotiated, procedures for recovering from such a case are unspecified. One approach is to manually configure the interface identifier of the interface.

The negotiated interface identifier is used by the local end of the PPP link to autoconfigure an IPv6 link-local unicast address for the PPP interface. However, it SHOULD NOT be assumed that the same interface identifier is used in configuring global unicast addresses for the PPP interface using IPv6 stateless address autoconfiguration [3]. The PPP peer MAY generate one or more interface identifiers, for instance, using a method described in [8], to autoconfigure one or more global unicast addresses.

As long as the interface identifier is negotiated in the IPV6CP phase of the PPP connection setup, it is redundant to perform duplicate address detection (DAD) as a part of the IPv6 Stateless Address Autoconfiguration protocol [3] on the IPv6 link-local address generated by the PPP peer. It may also be redundant to perform DAD on any global unicast addresses configured (using an interface identifier that is either negotiated during IPV6CP or generated, for instance, as per [8]) for the interface as part of the IPv6 Stateless Address Autoconfiguration protocol [3] provided that the following two conditions are met:

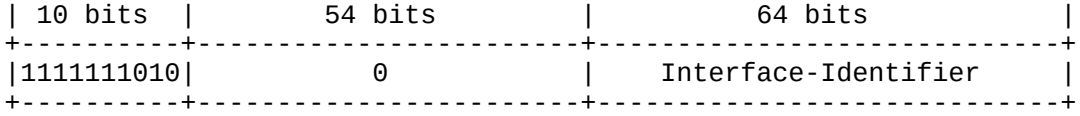
1) The prefixes advertised through the Router Advertisement messages by the access router terminating the PPP link are exclusive to the PPP link.

2) The access router terminating the PPP link does not autoconfigure any IPv6 global unicast addresses from the prefixes that it advertises.

Therefore, it is RECOMMENDED that for PPP links with the IPV6CP interface-identifier option enabled and satisfying the aforementioned two conditions, the default value of the DupAddrDetectTransmits autoconfiguration variable [3] is set to zero by the system management. 3GPP2 networks are an example of a technology that uses PPP to enable a host to obtain an IPv6 global unicast address and satisfies the aforementioned two conditions [10]. 3GPP networks are another example ([11] [13]).

Link-local addresses

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



The most significant 10 bits of the address is the Link-Local prefix FE80::. 54 zero bits pad out the address between the Link-Local prefix and the interface-identifier fields.

6. Security Considerations

Lack of link security, such as authentication, trigger the security concerns raised in [3] when the stateless address autoconfiguration method is employed for the generation of global unicast IPv6 addresses out of interface identifiers that are either negotiated through the IPV6CP or generated, for instance, using a method described in [8]. Thus, the mechanisms that are appropriate for ensuring PPP link security are addressed below, together with the reference to a generic threat model.

The mechanisms that are appropriate for ensuring PPP link Security are: 1) Access Control Lists that apply filters on traffic received over the link for enforcing admission policy, 2) an Authentication protocol that facilitates negotiations between peers [15] to select an authentication method (e.g., MD5 [16]) for validation of the peer, and 3) an Encryption protocol that facilitates negotiations between peers to select encryption algorithms (or crypto-suites) to ensure data confidentiality [17].

There are certain threats associated with peer interactions on a PPP link even with one or more of the above security measures in place. For instance, using the MD5 authentication method [16] exposes one to replay attack, where an attacker could intercept and replay a station’s identity and password hash to get access to a network. The user of this specification is advised to refer to [15], which presents a generic threat model, for an understanding of the threats posed to the security of a link. The reference [15] also gives a framework to specify requirements for the selection of an authentication method for a given application.

7. IANA Considerations

The IANA has assigned value 1 for the Type field of the IPv6 datagram interface-identifier option specified in this document. The current assignment is up-to-date at [9].

8. 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.

The editor is grateful for the input provided by members of the IPv6 community in the spirit of updating RFC 2472. Thanks, in particular,

go to Pete Barany and Karim El Malki for their technical contributions. Also, thanks to Alex Conta for a thorough review, Stephen Kent for helping with security aspects, and Spencer Dawkins and Pekka Savola for the nits. Finally, the author is grateful to Jari Arkko for his initiation to bring closure to this specification.

9. References

9.1. Normative References

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

[2] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[3] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

[4] IEEE, "Guidelines For 64-bit Global Identifier (EUI-64)", http://standards.ieee.org/regauth/oui/tutorials/EUI64.html

[5] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[6] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[7] Haskin, D. and E. Allen, "IP Version 6 over PPP", RFC 2472, December 1998.

[8] Narten T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.

9.2. Informative references

[9] IANA, "Assigned Numbers," http://www.iana.org/numbers.html

[10] 3GPP2 X.S0011-002-C v1.0, "cdma2000 Wireless IP Network Standard: Simple IP and Mobile IP Access Services," September 2003.

[11] 3GPP TS 29.061 V6.4.0, "Interworking between the Public Land Mobile Network (PLMN) Supporting packet based services and Packet Data Networks (PDN) (Release 6)," April 2005.

[12] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[13] 3GPP TS 23.060 v6.8.0, "General Packet Radio Service (GPRS); Service description; Stage 2 (Release 6)," March 2005.

[14] Narten, T. and C. Burton, "A Caution On The Canonical Ordering Of Link-Layer Addresses", RFC 2469, December 1998.

[15] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.

[16] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[17] Meyer, G., "The PPP Encryption Control Protocol (ECP)", RFC 1968, June 1996.

Appendix A: Global Scope Addresses

A node on the PPP link creates global unicast addresses either through stateless or stateful address autoconfiguration mechanisms. In the stateless address autoconfiguration [3], the node relies on sub-net prefixes advertised by the router via the Router Advertisement messages to obtain global unicast addresses from an interface identifier. In the stateful address autoconfiguration, the host relies on a Stateful Server, like DHCPv6 [12], to obtain global unicast addresses.

Appendix B: Changes from RFC 2472

The following changes were made from RFC 2472 "IPv6 over PPP":

- Minor updates to Sections 3 and 4
- Updated the text in Section 4.1 to include the reference to Appendix A and minor text clarifications.
- Removed Section 4.2 on IPv6-Compression-Protocol based on IESG recommendation, and created a new standards-track document to cover negotiation of the IPv6 datagram compression protocol using IPV6CP.
- Updated the text in Section 5 to: (a) allow the use of one or more interface identifiers generated by a peer, in addition to the use of interface identifier negotiated between peers of the link, in the creation of global unicast addresses for the local PPP interface, and (b) identify cases against the DAD of created non-link-local addresses.
- Added new and updated references.
- Added Appendix A

Authors' Addresses

Dimitry Haskin
Ed Allen

Srihari Varada (Editor)
TranSwitch Corporation
3 Enterprise Dr.
Shelton, CT 06484. US.

Phone: +1 203 929 8810
EMail: varada@ieee.org

Full Copyright Statement

Intellectual Property