

IPv6 maintenance Working Group (6man)
Internet-Draft
Updates: [2464](#), [2467](#), [2470](#), 2491, 2492,
[2497](#), [2590](#), [3146](#), [3315](#), [3572](#),
4291, 4338, 4391, 5072, 5121
(if approved)

Intended status: Standards Track

Expires: August 20, 2016

F. Gont
SI6 Networks / UTN-FRH
A. Cooper
Cisco
D. Thaler
Microsoft
W. Liu
Huawei Technologies
February 17, 2016

Recommendation on Stable IPv6 Interface Identifiers
draft-ietf-6man-default-iids-10

Abstract

This document changes the default Interface Identifier generation scheme for SLAAC to that specified in [RFC7217](#), and recommends against embedding link-layer addresses in IPv6 Interface Identifiers. It formally updates [RFC2464](#), [RFC2467](#), [RFC2470](#), [RFC2491](#), [RFC2492](#), [RFC2497](#), [RFC2590](#), [RFC3146](#), [RFC3572](#), [RFC4291](#), [RFC4338](#), [RFC4391](#), [RFC5072](#), and [RFC5121](#), by removing the text in these RFCs that required the IPv6 Interface Identifiers to be derived from the underlying link-layer address, and replacing the aforementioned text with a pointer to this document. Additionally, this document updates [RFC3315](#) by specifying additional requirements on the generation of Interface Identifiers used in Dynamic Host Configuration Protocol version 6 (DHCPv6). It also provides advice to system administrators who employ manual configuration.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on August 20, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Terminology	3
3.	Generation of IPv6 Interface Identifiers with SLAAC	3
4.	Generation of IPv6 Interface Identifiers with DHCPv6	4
5.	Generation of IPv6 Interface Identifiers with Manual Configuration	5
6.	Update to existing RFCs	5
7.	Future Work	17
8.	IANA Considerations	18
9.	Security Considerations	18
10.	Acknowledgements	18
11.	References	18
	Authors' Addresses	21

[1.](#) Introduction

[RFC4862] specifies Stateless Address Autoconfiguration (SLAAC) for IPv6 [[RFC2460](#)], which typically results in hosts configuring one or more "stable" addresses composed of a network prefix advertised by a local router, and an Interface Identifier (IID) [[RFC4291](#)] that typically embeds a link-layer address (e.g., an IEEE LAN MAC address).

In some network technologies and adaptation layers, the use of an IID based on a link-layer address may offer some advantages. For example, the IP-over-IEEE802.15.4 standard in [[RFC6775](#)] allows for compression of IPv6 addresses when the IID is based on the underlying link-layer address.

The security and privacy implications of embedding a link-layer address in an IPv6 IID have been known for some time now, and are

discussed in great detail in [\[I-D.ietf-6man-ipv6-address-generation-privacy\]](#); they include:

- o Network activity correlation
- o Location tracking
- o Address scanning
- o Device-specific vulnerability exploitation

Some popular IPv6 implementations have already deviated from the traditional stable IID generation scheme to mitigate the aforementioned security and privacy implications [\[Microsoft\]](#).

As a result of the aforementioned issues, this document changes the default Interface Identifier generation scheme for SLAAC to that specified in [\[RFC7217\]](#), and recommends against embedding link-layer addresses in IPv6 Interface Identifiers, such that the aforementioned issues are mitigated.

NOTE: [\[RFC4291\]](#) defines the "Modified EUI-64 format" for IIDs. [Appendix A of \[RFC4291\]](#) then describes how to transform an IEEE EUI-64 identifier, or an IEEE 802 48-bit MAC address from which an EUI-64 identifier is derived, into an IID in the Modified EUI-64 format.

Finally this document updates [\[RFC3315\]](#) by specifying additional requirements on the generation of Interface Identifiers used in Dynamic Host Configuration Protocol version 6 (DHCPv6), and also provides advice to system administrators who employ manual configuration.

2. Terminology

Stable address:

An address that does not vary over time within the same network (as defined in [\[I-D.ietf-6man-ipv6-address-generation-privacy\]](#)).

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 [RFC 2119](#) [\[RFC2119\]](#).

3. Generation of IPv6 Interface Identifiers with SLAAC

Link layers MUST define a mechanism that provides mitigation of the security and privacy implications discussed in [Section 1](#). Such mechanism MUST meet the following requirements:

1. The resulting Interface Identifiers remain stable for each prefix used with SLAAC within each subnet for the same network interface. That is, the algorithm generates the same Interface Identifier when configuring an address (for the same interface) belonging to the same prefix within the same subnet
2. The resulting Interface Identifiers must change when addresses are configured for different prefixes. That is, if different autoconfiguration prefixes are used to configure addresses for the same network interface card, the resulting Interface Identifiers must be (statistically) different. This means that, given two addresses, it must be difficult for an outside entity to tell whether the addresses have been generated by the same host.
3. It must be difficult for an outside entity to predict the Interface Identifiers that will be generated by the algorithm, even with knowledge of the Interface Identifiers generated for configuring other addresses.
4. The resulting Interface Identifiers must be semantically opaque.

Nodes SHOULD implement and employ [\[RFC7217\]](#) as the default scheme for generating stable IPv6 addresses with SLAAC. A link layer MAY also define a mechanism that is more efficient and does not comply with the aforementioned requirements. The choice of whether to enable the security- and privacy-preserving mechanism or not SHOULD be configurable in such a case.

By default, nodes SHOULD NOT employ IPv6 address generation schemes that embed the underlying link-layer address in the IID. In particular, this document RECOMMENDS that nodes do not generate IIDs with the schemes specified in [\[RFC2464\]](#), [\[RFC2467\]](#), [\[RFC2470\]](#), [\[RFC2491\]](#), [\[RFC2492\]](#), [\[RFC2497\]](#), [\[RFC2590\]](#), [\[RFC3146\]](#), [\[RFC3572\]](#), [\[RFC4338\]](#), [\[RFC4391\]](#), [\[RFC5121\]](#), and [\[RFC5072\]](#). The recommendations in this document override any other recommendations on the generation of IIDs in the updated RFCs. The specific updates to these documents to effectuate this recommendation are included in [Section 6](#).

4. Generation of IPv6 Interface Identifiers with DHCPv6

By default, DHCPv6 server implementations SHOULD NOT generate predictable IPv6 addresses (such as IPv6 addresses where the IIDs are consecutive small numbers). [\[I-D.ietf-dhc-stable-privacy-addresses\]](#) specifies one possible algorithm that could be employed to comply with this requirement. Another possible algorithm would be to select a pseudo-random value chosen from a discrete uniform distribution,

while avoiding the reserved IPv6 Interface Identifiers [[RFC5453](#)] [[IANA-RESERVED-IID](#)].

5. Generation of IPv6 Interface Identifiers with Manual Configuration

Network administrators should be aware of the security implications of predictable Interface Identifiers [[I-D.ietf-6man-ipv6-address-generation-privacy](#)], and avoid the use of predictable addresses when the aforementioned issues are of concern.

6. Update to existing RFCs

The following subsections clarify how each of the RFCs affected by this document are updated.

Note to the RFC Editor:

In the following subsections, the legend "[RFCXXXX]" should be replaced with the RFC number assigned to this document, and this note to the RFC Editor should be removed before publication of this document as an RFC.

6.1. Update to [RFC2464](#)

The entire text of [Section 4 of \[RFC2464\]](#) is replaced with the following text:

```
----- cut here ----- cut here -----
The Interface Identifier [AARCH] for an Ethernet interface MUST be
generated as specified in [RFCXXXX].
----- cut here ----- cut here -----
```

The following text from [Section 6 of \[RFC2464\]](#):

```
----- cut here ----- cut here -----
Ethernet Address
    The 48 bit Ethernet IEEE 802 address, in canonical bit
    order. This is the address the interface currently
    responds to, and may be different from the built-in
    address used to derive the Interface Identifier.
----- cut here ----- cut here -----
```

is formally replaced with:

----- cut here ----- cut here -----
Ethernet Address
 The 48 bit Ethernet IEEE 802 address, in canonical bit
 order. This is the address the interface currently
 responds to.
----- cut here ----- cut here -----

6.2. Update to [RFC2467](#)

The entire text of [Section 5 of \[RFC2467\]](#) is replaced with the following text:

----- cut here ----- cut here -----
The Interface Identifier [AARCH] for an FDDI interface MUST be
generated as specified in [RFCXXXX].
----- cut here ----- cut here -----

The following text from [Section 7 of \[RFC2467\]](#):

----- cut here ----- cut here -----
FDDI Address
 The 48 bit FDDI IEEE 802 address, in canonical bit order.
 This is the address the interface currently responds to,
 and may be different from the built-in address used to
 derive the Interface Identifier.
----- cut here ----- cut here -----

is formally replaced with:

----- cut here ----- cut here -----
FDDI Address
 The 48 bit FDDI IEEE 802 address, in canonical bit order.
 This is the address the interface currently responds to.
----- cut here ----- cut here -----

6.3. Update to [RFC2470](#)

The entire text of [Section 4 of \[RFC2470\]](#) is replaced with the following text:

----- cut here ----- cut here -----
The Interface Identifier [AARCH] for a Token Ring interface MUST be
generated as specified in [RFCXXXX].
----- cut here ----- cut here -----

The following text from [Section 6 of \[RFC2470\]](#):

----- cut here ----- cut here -----
Token Ring Address: The 48 bit Token Ring IEEE 802
address, in canonical bit order. This is the address the
interface currently responds to, and may be different from
the built-in address used to derive the Interface
Identifier.
----- cut here ----- cut here -----

is formally replaced with:

----- cut here ----- cut here -----
Token Ring Address: The 48 bit Token Ring IEEE 802
address, in canonical bit order. This is the address the
interface currently responds to.
----- cut here ----- cut here -----

6.4. Update to [RFC2491](#)

The entire text of [Section 5.1](#), [Section 5.1.1](#), and [Section 5.1.2 of \[RFC2491\]](#) is replaced with the following text:

----- cut here ----- cut here -----
5.1 Interface Tokens

The Interface Token (or Interface Identifier [AARCH]) for each IPv6 interface MUST be generated as specified in [RFCXXXX].

All implementations MUST support manual configuration of interface tokens to allow operators to manually change a interface token on a per-LL basis. Operators may choose to manually set interface tokens for reasons other than eliminating duplicate addresses.

All interface tokens MUST be 64 bits in length.

----- cut here ----- cut here -----

6.5. Update to [RFC2492](#)

The entire text of [Section 5](#) (and all the corresponding subsections) of of [\[RFC2492\]](#) is replaced with the following text:

----- cut here ----- cut here -----
5.1 Interface Tokens

The Interface Token (or Interface Identifier [AARCH]) for each IPv6 interface MUST be generated as specified in [RFCXXXX].

All implementations MUST support manual configuration of interface tokens to allow operators to manually change a interface token on a per-LL basis. Operators may choose to manually set interface tokens for reasons other than eliminating duplicate addresses.

All interface tokens MUST be 64 bits in length.

----- cut here ----- cut here -----

6.6. Update to [RFC2497](#)

The entire text of [Section 4 of \[RFC2497\]](#) is replaced with the following text:

----- cut here ----- cut here -----
The Interface Identifier [AARCH] for an ARCnet interface MUST be generated as specified in [RFCXXXX].
----- cut here ----- cut here -----

The entire text of [Section 8 of \[RFC2497\]](#) is replaced with the following text:

----- cut here ----- cut here -----
Interface Identifiers generated as specified in [RFCXXXX] mitigate the security and privacy implications discussed in [Section 1](#) of such document.
----- cut here ----- cut here -----

6.7. Update to [RFC2590](#)

The entire [Section 4](#) and [Section 4.1 of \[RFC2590\]](#) is replaced with the following text:

----- cut here ----- cut here -----

4. Stateless Autoconfiguration

An interface identifier [AARCH] for an IPv6 Frame Relay interface MUST be unique on a Frame Relay link [AARCH], and MUST be unique on each of the virtual links represented by the VCs terminated on the interface.

The interface identifier for the Frame Relay interface MUST be generated as specified in [RFCXXXX].

We note that each virtual circuit in a Frame Relay network is uniquely identified on a Frame Relay interface by a DLCI. Furthermore, a DLCI can be seen as an identification of the end point of a virtual circuit on a Frame Relay interface. Since each Frame Relay VC is configured or established separately, and acts like an independent virtual-link from other VCs in the network, or on the interface, link, wire or fiber, it seems beneficial to view each VC's termination point on the Frame Relay interface as a "pseudo-interface" or "logical-interface" overlaid on the Frame Relay interface. Furthermore, it seems beneficial to be able to generate and associate an IPv6 autoconfigured address (including an IPv6 link local address) to each "pseudo-interface", i.e. end-point of a VC, i.e. to each DLCI on a Frame Relay interface.

----- cut here ----- cut here -----

The entire [Section 9 of \[RFC2590\]](#) is replaced as follows:

----- cut here ----- cut here -----

9. Security Considerations

Security protection against forgery or accident at the level of the mechanisms described here is provided by the IPv6 security mechanisms [IPSEC], [IPSEC-Auth], [IPSEC-ESP] applied to Neighbor Discovery [IPv6-ND] or Inverse Neighbor Discovery [IND] messages.

To avoid an IPsec Authentication verification failure, the Frame Relay specific preprocessing of a Neighbor Discovery Solicitation message that contains a DLCI format Source link-layer address option, MUST be done by the receiver node after it completed IP Security processing.

----- cut here ----- cut here -----

6.8. Update to [RFC3146](#)

The entire [Section 6 of \[RFC3146\]](#) is replaced with the following text:

----- cut here ----- cut here -----

6. STATELESS AUTOCONFIGURATION

The Interface Identifier [AARCH] for an IEEE1394 interface MUST be generated as specified in [RFCXXXX].

An IPv6 address prefix used for stateless autoconfiguration [ACONF] of an IEEE1394 interface MUST have a length of 64 bits.

----- cut here ----- cut here -----

6.9. Update to [RFC3315](#)

The following text in [Section 11](#) of [RFC3315](#):

----- cut here ----- cut here -----

Any address assigned by a server that is based on an EUI-64 identifier MUST include an interface identifier with the "u" (universal/local) and "g" (individual/group) bits of the interface identifier set appropriately, as indicated in section 2.5.1 of [RFC 2373](#) [5].

----- cut here ----- cut here -----

is formally replaced with:

----- cut here ----- cut here -----

By default, DHCPv6 server implementations SHOULD NOT generate predictable IPv6 addresses (such as IPv6 addresses where the IIDs are consecutive small numbers). [\[I-D.ietf-dhc-stable-privacy-addresses\]](#) specifies one possible algorithm that could be employed to comply with this requirement. Another possible algorithm would be to select a pseudo-random value chosen from a discrete uniform distribution, while avoiding the reserved IPv6 Interface Identifiers [\[RFC5453\]](#) [\[IANA-RESERVED-IID\]](#).

----- cut here ----- cut here -----

Additionally, the following references should be added to [Section 26](#) of [RFC3315](#):

----- cut here ----- cut here -----

[[IANA-RESERVED-IID](#)]

IANA, "Reserved IPv6 Interface Identifiers",
<<http://www.iana.org/assignments/ipv6-interface-ids>>.

[RFC5453] Krishnan, S., "Reserved IPv6 Interface Identifiers",
[RFC 5453](#), DOI 10.17487/RFC5453, February 2009,
<<http://www.rfc-editor.org/info/rfc5453>>.

[I-D.ietf-dhc-stable-privacy-addresses]

Gont, F. and S. LIU, "A Method for Generating Semantically
Opaque Interface Identifiers with Dynamic Host
Configuration Protocol for IPv6 (DHCPv6)", [draft-ietf-dhc-stable-privacy-addresses-02](#) (work in progress), April
2015.

----- cut here ----- cut here -----

[6.10.](#) Update to [RFC3572](#)

The entire text of [Section 3 of \[RFC3572\]](#) is replaced as follows:

----- cut here ----- cut here -----

3. Interface Identifier

The Interface Identifier [AARCH] for a MAPOS interface MUST be
generated as specified in [RFCXXXX].

----- cut here ----- cut here -----

Additionally, [Section 6.2](#) ("Uniqueness of Interface Identifiers") of
[[RFC3572](#)] is entirely eliminated.

[6.11.](#) Update to [RFC4291](#)

The entire text of [Section 2.5.1 of \[RFC4291\]](#) is replaced with the
following text:

----- cut here ----- cut here -----

2.5.1. Interface Identifiers

Interface identifiers in IPv6 unicast addresses are used to identify interfaces on a link. They are required to be unique within a subnet prefix. It is recommended that the same interface identifier not be assigned to different nodes on a link. They may also be unique over a broader scope. The same interface identifier may be used on multiple interfaces on a single node, as long as they are attached to different subnets.

For all unicast addresses, except those that start with the binary value 000, Interface IDs are required to be 64 bits long, and MUST be generated as specified in [RFCXXXX].

The details of forming interface identifiers are defined in the appropriate "IPv6 over <link>" specification, such as "IPv6 over Ethernet" [ETHER], and "IPv6 over FDDI" [FDDI].

----- cut here ----- cut here -----

6.12. Update to [RFC4338](#)

The entire text of [Section 5](#) (and of all the corresponding subsections) of [[RFC4338](#)] is replaced with the following text:

----- cut here ----- cut here -----

5. IPv6 Stateless Address Autoconfiguration

The IPv6 Interface ID [AARCH] for an Nx_Port MUST be generated as specified in [RFCXXXX].

IPv6 stateless address autoconfiguration MUST be performed as specified in [ACONF]. An IPv6 Address Prefix used for stateless address autoconfiguration of an Nx_Port MUST have a length of 64 bits.

----- cut here ----- cut here -----

6.13. Update to [RFC4391](#)

The entire text of [Section 8 of \[RFC4391\]](#) is replaced with the following text:

----- cut here ----- cut here -----

8. IPv6 Stateless Autoconfiguration

The IPv6 Interface ID [AARCH] MUST be generated as specified in [RFCXXXX].

----- cut here ----- cut here -----

6.14. Update to [RFC5072](#)

The entire text of [Section 4.1 of \[RFC5072\]](#) is replaced with the following text:

----- cut here ----- cut here -----

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.

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. Additionally, the tentative interface identifier SHOULD be generated as specified in [RFCXXXX].

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). Additionally, the value suggested **SHOULD** be generated as specified in [RFCXXXX].

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). Additionally, the value suggested **SHOULD** be generated as specified in [RFCXXXX].

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.

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Type      |      Length      | Interface-Identifier (MS Bytes)
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
                        Interface-Identifier (cont)
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
Interface-Identifier (LS Bytes) |
+---+---+---+---+---+---+---+---+---+---+---+---+

```

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 will depend on the algorithm employed to generate the interface identifier. One approach is to manually configure the interface identifier of the interface.

----- cut here ----- cut here -----

Additionally, the following text of [Section 5 of \[RFC5072\]](#):

----- cut here ----- cut here -----

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.

is formally replaced with the following text:

----- cut here ----- cut here -----

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 will depend on the algorithm employed to generate the interface identifier. 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].

----- cut here ----- cut here -----

6.15. Update to [RFC5121](#)

The entire text of [Section 9.1 of \[RFC5121\]](#) is replaced with the following text:

----- cut here ----- cut here -----
9.1. Interface Identifier

The MS SHOULD generate interface identifiers as specified in [RFCXXXX].

----- cut here ----- cut here -----

Additionally, [Section 9.2](#) is replaced as follows:

----- cut here ----- cut here -----
9.2. Duplicate Address Detection

DAD SHOULD be performed as per "Neighbor Discovery for IP Version 6 (IPv6)", [\[RFC4861\]](#) and "IPv6 Stateless Address Autoconfiguration" [\[RFC4862\]](#). The IPv6 link over 802.16 is specified in this document as a point-to-point link. Based on this criteria, it may be redundant to perform DAD on a global unicast address that is configured as part of the IPv6 Stateless Address Autoconfiguration Protocol [\[RFC4862\]](#) as long as the following two conditions are met:

1. The prefixes advertised through the router advertisement messages by the access router terminating the 802.16 IPv6 link are unique to that link.
2. The access router terminating the 802.16 IPv6 link does not autoconfigure any IPv6 global unicast addresses from the prefix that it advertises.

----- cut here ----- cut here -----

7. Future Work

At the time of this writing, the mechanisms specified in the following documents might require updates to be fully compatible with the recommendations in this document:

- o "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks" [\[RFC6282\]](#)
- o "Transmission of IPv6 Packets over IEEE 802.15.4 Networks" [\[RFC4944\]](#)
- o "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)" [\[RFC6775\]](#)

- o "Transmission of IPv6 Packets over ITU-T G.9959 Networks"[[RFC7428](#)]

Future revisions or updates of these documents should take the issues of privacy and security mentioned in [Section 1](#) and explain any design and engineering considerations that lead to the use of IIDs based on a node's link-layer address.

[8.](#) IANA Considerations

There are no IANA registries within this document. The RFC-Editor can remove this section before publication of this document as an RFC.

[9.](#) Security Considerations

This recommends against the (default) use of predictable Interface Identifiers in IPv6 addresses. It recommends [[RFC7217](#)] as the default scheme for generating IPv6 stable addresses with SLAAC, such that the security and privacy issues of IIDs that embed link-layer addresses are mitigated. Additionally, it recommends against predictable IIDs in DHCPv6 and manual configuration

[10.](#) Acknowledgements

The authors would like to thank (in alphabetical order) Bob Hinden, Ray Hunter and Erik Nordmark, for providing a detailed review of this document.

The authors would like to thank (in alphabetical order) Fred Baker, Carsten Bormann, Scott Brim, Brian Carpenter, Samita Chakrabarti, Tim Chown, Lorenzo Colitti, Jean-Michel Combes, Greg Daley, Esko Dijk, Ralph Droms, David Farmer, Brian Haberman, Ulrich Herberg, Philip Homburg, Jahangir Hossain, Jonathan Hui, Christian Huitema, Ray Hunter, Erik Kline, Sheng Jiang, Roger Jorgensen, Dan Luedtke, Kerry Lynn, George Mitchel, Gabriel Montenegro, Erik Nordmark, Simon Perreault, Tom Petch, Alexandru Petrescu, Michael Richardson, Arturo Servin, Mark Smith, Tom Taylor, Ole Troan, Tina Tsou, Glen Turner, Randy Turner, and James Woodyatt, for providing valuable comments on earlier versions of this document.

[11.](#) References

[11.1.](#) Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 2460](#), DOI 10.17487/RFC2460, December 1998, <<http://www.rfc-editor.org/info/rfc2460>>.
- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", [RFC 2464](#), DOI 10.17487/RFC2464, December 1998, <<http://www.rfc-editor.org/info/rfc2464>>.
- [RFC2467] Crawford, M., "Transmission of IPv6 Packets over FDDI Networks", [RFC 2467](#), DOI 10.17487/RFC2467, December 1998, <<http://www.rfc-editor.org/info/rfc2467>>.
- [RFC2470] Crawford, M., Narten, T., and S. Thomas, "Transmission of IPv6 Packets over Token Ring Networks", [RFC 2470](#), DOI 10.17487/RFC2470, December 1998, <<http://www.rfc-editor.org/info/rfc2470>>.
- [RFC2491] Armitage, G., Schuler, P., Jork, M., and G. Harter, "IPv6 over Non-Broadcast Multiple Access (NBMA) networks", [RFC 2491](#), DOI 10.17487/RFC2491, January 1999, <<http://www.rfc-editor.org/info/rfc2491>>.
- [RFC2492] Armitage, G., Schuler, P., and M. Jork, "IPv6 over ATM Networks", [RFC 2492](#), DOI 10.17487/RFC2492, January 1999, <<http://www.rfc-editor.org/info/rfc2492>>.
- [RFC2497] Souvatzis, I., "Transmission of IPv6 Packets over ARCnet Networks", [RFC 2497](#), DOI 10.17487/RFC2497, January 1999, <<http://www.rfc-editor.org/info/rfc2497>>.
- [RFC2590] Conta, A., Malis, A., and M. Mueller, "Transmission of IPv6 Packets over Frame Relay Networks Specification", [RFC 2590](#), DOI 10.17487/RFC2590, May 1999, <<http://www.rfc-editor.org/info/rfc2590>>.
- [RFC3146] Fujisawa, K. and A. Onoe, "Transmission of IPv6 Packets over IEEE 1394 Networks", [RFC 3146](#), DOI 10.17487/RFC3146, October 2001, <<http://www.rfc-editor.org/info/rfc3146>>.
- [RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 3315](#), DOI 10.17487/RFC3315, July 2003, <<http://www.rfc-editor.org/info/rfc3315>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", [RFC 4291](#), DOI 10.17487/RFC4291, February 2006, <<http://www.rfc-editor.org/info/rfc4291>>.

- [RFC4338] DeSanti, C., Carlson, C., and R. Nixon, "Transmission of IPv6, IPv4, and Address Resolution Protocol (ARP) Packets over Fibre Channel", [RFC 4338](#), DOI 10.17487/RFC4338, January 2006, <<http://www.rfc-editor.org/info/rfc4338>>.
- [RFC4391] Chu, J. and V. Kashyap, "Transmission of IP over InfiniBand (IPoIB)", [RFC 4391](#), DOI 10.17487/RFC4391, April 2006, <<http://www.rfc-editor.org/info/rfc4391>>.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", [RFC 4862](#), DOI 10.17487/RFC4862, September 2007, <<http://www.rfc-editor.org/info/rfc4862>>.
- [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", [RFC 4944](#), DOI 10.17487/RFC4944, September 2007, <<http://www.rfc-editor.org/info/rfc4944>>.
- [RFC5072] Varada, S., Ed., Haskins, D., and E. Allen, "IP Version 6 over PPP", [RFC 5072](#), DOI 10.17487/RFC5072, September 2007, <<http://www.rfc-editor.org/info/rfc5072>>.
- [RFC5121] Patil, B., Xia, F., Sarikaya, B., Choi, JH., and S. Madanapalli, "Transmission of IPv6 via the IPv6 Convergence Sublayer over IEEE 802.16 Networks", [RFC 5121](#), DOI 10.17487/RFC5121, February 2008, <<http://www.rfc-editor.org/info/rfc5121>>.
- [RFC5453] Krishnan, S., "Reserved IPv6 Interface Identifiers", [RFC 5453](#), DOI 10.17487/RFC5453, February 2009, <<http://www.rfc-editor.org/info/rfc5453>>.
- [RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", [RFC 6282](#), DOI 10.17487/RFC6282, September 2011, <<http://www.rfc-editor.org/info/rfc6282>>.
- [RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", [RFC 6775](#), DOI 10.17487/RFC6775, November 2012, <<http://www.rfc-editor.org/info/rfc6775>>.

- [RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", [RFC 7217](#), DOI 10.17487/RFC7217, April 2014, <<http://www.rfc-editor.org/info/rfc7217>>.
- [RFC7428] Brandt, A. and J. Buron, "Transmission of IPv6 Packets over ITU-T G.9959 Networks", [RFC 7428](#), DOI 10.17487/RFC7428, February 2015, <<http://www.rfc-editor.org/info/rfc7428>>.

11.2. Informative References

- [I-D.ietf-6man-ipv6-address-generation-privacy] Cooper, A., Gont, F., and D. Thaler, "Privacy Considerations for IPv6 Address Generation Mechanisms", [draft-ietf-6man-ipv6-address-generation-privacy-08](#) (work in progress), September 2015.
- [I-D.ietf-dhc-stable-privacy-addresses] Gont, F. and S. LIU, "A Method for Generating Semantically Opaque Interface Identifiers with Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [draft-ietf-dhc-stable-privacy-addresses-02](#) (work in progress), April 2015.
- [IANA-RESERVED-IID] IANA, "Reserved IPv6 Interface Identifiers", <<http://www.iana.org/assignments/ipv6-interface-ids>>.
- [Microsoft] Davies, J., "Understanding IPv6, 3rd. ed", page 83, Microsoft Press, 2012, <<http://it-ebooks.info/book/1022/>>.
- [RFC3572] Ogura, T., Maruyama, M., and T. Yoshida, "Internet Protocol Version 6 over MAPOS (Multiple Access Protocol Over SONET/SDH)", [RFC 3572](#), DOI 10.17487/RFC3572, July 2003, <<http://www.rfc-editor.org/info/rfc3572>>.

Authors' Addresses

Fernando Gont
SI6 Networks / UTN-FRH
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: fgont@si6networks.com
URI: <http://www.si6networks.com>

Alissa Cooper
Cisco
707 Tasman Drive
Milpitas, CA 95035
US

Phone: +1-408-902-3950
Email: alcoop@cisco.com
URI: <https://www.cisco.com/>

Dave Thaler
Microsoft
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Phone: +1 425 703 8835
Email: dthaler@microsoft.com

Will Liu
Huawei Technologies
Bantian, Longgang District
Shenzhen 518129
P.R. China

Email: liushucheng@huawei.com

