

6MAN Working Group
Internet-Draft
Updates: [RFC4291](#), [RFC4007](#) (if approved)
Intended status: Standards Track
Expires: December 8, 2019

A. Petrescu
CEA, LIST
L. Velvindron
Cyberstorm.mu
N. Kottapalli
Benu Networks
G. Mishra
Verizon Communications Inc. (VZ)
D. Mudric
Avaya
June 6, 2019

The length of the prefix of an IPv6 link-local address ranges from 10 to
127
[draft-petrescu-6man-ll-prefix-len-21](#)

Abstract

A rejected Erratum to [RFC4291](#) "IPv6 Addr Archi" on the topic of link-local addresses 'would need' a draft. This draft is an answer to that need.

The length of the prefix of an IPv6 link-local address is variable. The minimal value is 10 decimal. The maximum value is 127 decimal.

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 <https://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 December 8, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

Internet-Draft

IPv6-LL-plen

June 2019

This document is subject to [BCP 78](https://trustee.ietf.org/license-info) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://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.	Definitions and Statements	2
2.	Terminology	4
3.	Justification	4
4.	Problem Statement	4
5.	Kinds of Solutions	5
6.	Context of Documents	6
7.	Context of OS Behaviour	8
8.	Historical Note	9
9.	Example of use of LL Prefix Length 32	9
10.	Use-Cases	10
10.1.	Use-Case Convoy	10
10.2.	Intuitive Next-Hop	12
11.	Security Considerations	12
12.	IANA Considerations	13
13.	Contributors	13
14.	Acknowledgements	13
15.	Normative References	13
Appendix A.	ChangeLog	13
	Authors' Addresses	15

[1.](#) Definitions and Statements

The prefix of an IP address is formed by the n leftmost bits of the address. (in a left-to-right writing system).

The prefix of an IP address is used for goals such as: identify the type of an IPv6 address (link-local, global, others), identify the belonging of an IP address to a particular subnetwork, assist the forwarding (or not forwarding) decisions, and others.

The minimal length of the prefix of an IPv6 link-local address (the

value of n) is equal to 10 decimal. The maximum is 127.

The prefix of an IPv6 link-local address is represented textually as "fe80::/n", where n MAY be any value between 10 and 127.

Regardless of the prefix length, the leftmost 10 bits of an IPv6 link-local address MUST be set to binary 1111111010 (hexadecimal fe80).

The [RFC 4291](#) illustration of an IPv6 link-local address is:

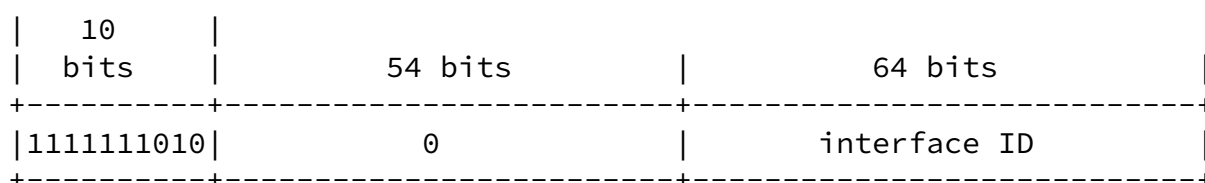


Figure 1: The IPv6 link-local address

There is an error in this [RFC 4291](#) illustration. The error is in requiring the 54 bits to be 0. The bits at position 11 to 16 are not 0 (the first 6 bits of the 54 bits). If they were 0 then 0xFEAF::1/10 were an invalid link-local address, whereas it is.

The better illustration of an IPv6 link-local address is:

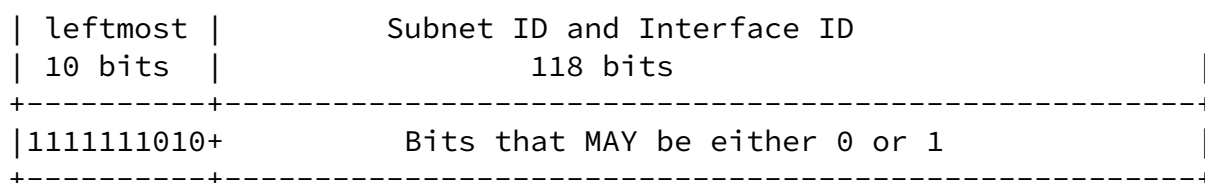


Figure 2: The IPv6 link-local address, better illustration

Examples: fe80::1/10, fe80:1::1/32, fe80::1:1/64 are all IPv6 link-local addresses; their prefix lengths are 10, 32 and 64 respectively. Also, 0xfeaf::1/10, 0xfebf:1::1/82 are also valid IPv6 link-local

addresses (remark no 'fe80'). Each such IPv6 address has the leftmost 10 bits equal to binary 1111111010.

A notation difficulty: the number binary 1111111010 can not be written in hexadecimal without specifying the number of significant bits (fe80::/10); yet that does not make it a 'prefix'. Converting 1111111010 to hexadecimal leads to 3FA (because in a left-to-right writing system the leading 0s before comma are irrelevant); yet '3FA' is not commonly known to be the leading bits of an IPv6 link-local address, fe80::/10 is.

[2.](#) Terminology

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](#)].

prefix: a contiguous string of bits valid for forwarding operations and for subnet formation. A prefix, in general, MUST have an integer length value from 1 to 127 (except when the prefix length is for default route, in which case the value is 0) and a prefix length must be indicated in its textual representation (e.g. 2001:db8::/32 is the prefix and 32 is the prefix length).

textual representation of a prefix: e.g. fe80::/64.

n leading bits: the first n bits in a string of bits read from left to right in a writing system that is read left-to-right. E.g. the 10 leading bits of the fe80::/64 textual representation of the IPv6 link-local prefix are 1111111010.

[3.](#) Justification

One justification is the following: in a managed network the administrator configures link-local addresses with the 54 bits set. The network runs dynamic routing protocols OSPF, ISIS and EIGRP. The adjacency must work in a mixed vendor environment.

[4.](#) Problem Statement

IPv6 link-local addresses are typically self-configured according to 4 RFCs and relying on the fe80::/10 IANA allocation, [RFC4291](#) 54 0 bits, and [RFC2464](#) MAC-based 64bit Interface IDs.

In some cases, it is advantageous to manually configure link-local addresses. Manual configuration is useful for easy remembering by humans, and for parameter resilience during network interface replacement (set addresses in computer startup configuration files). Further, the manual configuration of addresses can be scripted by automated software for rapid prototyping; still, this automated formation of addresses is not the 'self-configuration' described in the 4 RFCs mentioned previously.

A self-configured link-local address according to the 4 RFCs is of the form fe80::64bitIID; an example of such address is fe80::dabc:fe13:5246:7109. This address is difficult to remember for humans because each of the 16 hex characters appears, and the appearance is disordered. Not only it is difficult to remember but it takes long to type. This is a problem on small screens and mouse-

less keyboards. An easy to remember and type link-local address is, for example, fe80:1::1.

Manual configuration of an LL address may use short IID and Subnet ID. The Subnet ID presence in the link-local address is useful in some wireless settings where the subnet structure parameters depend on the link locality. Other settings may also benefit.

When manually setting the link-local address it is necessary to know the length of the prefix of the subnet on which this link-local address is present. This length is necessary for on-link determination.

The problem is: manually setting a prefix length other than 64 to link-local addresses may provoke glitches.

5. Kinds of Solutions

Some solutions to the problem are: use an address of the form fe80:1::2/32, or use an address of the form fe80::1:2/64, where 1 is the Subnet ID and 2 is the Interface ID. Other solutions involve a hidden 'scope_id' and the use of special syntax ('%') to denote an

interface. Each of these solutions have other problems of their own: set some of the 54 mandatorily reset bits of [RFC4291](#), not implementable on some OS; invade the IID with a Subnet ID, and potentially others.

Invading the IID with a Subnet ID happens in the following situation: if fe80::1:2 assumes fe80::/64 as prefix length, then it is impossible for '1' to be a Subnet ID. A Subnet ID must be covered by a prefix length, otherwise routing and on-link determination dont work. One cant have fe80::/64 as prefix, and '1' as prefix, and a 64bit ::1:2 Interface ID.

The 'scope_id' is 'hidden' in some operating systems; this hide is known by noticing that the use of 'scope_id' is not mandatory for LL addresses; instead of using 'scope_id' it is possible to rely on the interface name. Some ifconfig commands on some OSs rely on the interface name and dont require the use of a 'scope_id' (%) parameter. It is the case for linux and Windows.

In practice, the use of fe80::1:2 was tried. It used the 64bit prefix length. But it does not perform on-link determination meaningfully (the '1' is part of IID, not of Subnet ID).

Another solution is: use Unique Local Addresses [RFC 4193](#). For example: Car1-front interface uses fc00:1::1, Car1-rear fc00:1::2, Car2-front fc00:2::1. A pseudo random number would rather be

generated for Global ID, when in production. A kind of solution involving ULAs has interesting properties, yet the ULA-addressed packets may be forwarded across subnets. This forwarding may be an inconvenient in some setting. The use of other than LL addresses, i.e. GUAs or ULAs; this has some advantages and some inconvenients (cant put LL in src of RA).

Other solutions involve the use of an 'fe80' prefix in the RA such that to configure link-local prefixes by a similar means than GUAs/ULAs. This also has advantages and drawbacks.

Another solution is: use DNS to hold long interface IDs and Subnet IDs. Such solutions recommend the use of name-to-address mapping, instead of easy to remember LLs; DNS is such tool; can be used in order to facilitate the remembering by humans. However, this has

some advantages and some inconvenients (e.g. needs DNS-SD, mDNS and IP multicast routing for multi-subnets; chicken-and-egg between formation of LLs needing these DNS tools to work in the first place). A particular inconvenient is the movement of the problem instead of solving it: upon interface change (replace faulty interface with a new one) one has to configure the DNS configuration files with a new pair name-address, instead of needing to configure startup scripts.

6. Context of Documents

[draft-bourbaki-6man-classless-ipv6-05](#) describes the motivation of considering IPv6 to be classless. It gives a little bit of history of why it is how it is. It proposes the rigid 64 IID length to be probably the last remnant of the boundary.

[draft-farmer-6man-routing-64-02](#) describes the relationship between routing and the 64-bit boundary; mainly GUA, no LL; it is ambiguous in its recommendation.

[draft-farmer-6man-exceptions-64-09](#) describes the exceptions to the standard subnet boundary in IPv6 addressing; mainly about GUA, not about LL; the exceptions are: GUA with the first 3 bits 0, manually config'ed addresses, DHCPv6 assigned addresses, ND on-link determination, IPv6-over-foo.

A memo describes the use of IPv6 link-local addresses in applications. The filename of the Internet Draft is [draft-smith-ipv6-link-locals-apps-00](#).

[RFC7404](#) describes "Using Only Link-Local Addressing inside an IPv6 Network".

The RFC "IPv6 Address Archi" illustrates the format of the link-local addresses. From the illustration it MAY be understood that the length of the link-local prefix is 10 bits of value 1111111010 and 54 0 bits.

IANA lists the "IPv6 prefix", and "Address Block", to be "fe80::/10" on its website. It is possible that in the future the IETF could decide to use the bits 11-53.

The [RFC 2464](#) "IPv6-over-Ethernet" states that the prefix for link-local addresses is "fe80::/64".

[RFC 6874](#), "Representing IPv6 Zone Identifiers in Address Literals and Uniform Resource Identifiers" specifies the link-local addresses to be under prefix "fe80::/10".

[RFC 8415](#) "DHCPv6" considers that link-local addresses are designated by the prefix fe80::/10.

[RFC 4007](#) "IPv6 Scoped Address Architecture" discusses Zone ID. A Zone ID may be used - internally - in the 54bits of a link-local address, even though these 54bits appear to be reset. The document mentions at a point that fe80::1 could be used in two separate physical links (not virtual, like the loopback).

[RFC4291](#) requires that an IPv6 link-local address be assigned on each interface. Yet, it does not require the link-local prefix to be associated to an interface.

[RFC4861](#) requires that the link local prefix be present in the Prefix List associated with an interface, although it does not specify the length of the link local prefix.

[RFC4862](#) "SLAAC" defines how GUAs and LLs self-configure. Whereas the GUA gets its prefix length from the RA (not from an RFC), the LL gets it from [RFC4291](#) (not from RA). They are independent choices based on distinct sources.

Several knowledgeable interpretations state that, generally speaking, the prefix length of link-local addresses is 10, but it is 64 in the particular case of Stateless Address-Autoconfiguration (SLAAC). In this latter case, the prefix is named a "subnet prefix", or "prefix on a link", and it is "fe80::/64".

The term "link-local prefix" is sometimes used to mean the prefix for on-link determination, and is sometimes used to mean the reserved address space for link-local addresses (including all current and future use). The latter is fe80::/10. Of which the address

standard format (by specifying intermediate 54 bits are all 0). As a result the former is (currently) only fe80::/64.

For people in the RIR world it's a common thing: you get a prefix from the RIR and then make assignments from it for specific purposes. You can route the aggregate allocation, but you're not allowed to use the unassigned parts (until you make an assignment). In this case fe80::/10 is the allocation and fe80::/64 is the assignment.

7. Context of OS Behaviour

Interpretations of the situation of linux working ok with fe80:1::1 call it a violation of standards.

Independent testing shows that 'ifconfig add fe80:1::1' works on linux but fails on openbsd.

A command to assign fe80:1::1 on a Cisco router works ok.

On Cisco platforms IOS, XE ver 16.x, XR and NXOS it is possible to populate the entire set of 54 bits with one's (instead of the 0s requested by [RFC 4291](#)). A test was performed between such Cisco routers running OSPF. The OSPF neighbors were still coming up. The error that was expected (and that did not happen) was that some glitches may appear due to the lack of textual compression of 54bits into '::'. However, there is a caveat: it is unknown what might happen with OSPF in a mixed environment, where not only Cisco is used.

The address fe80::1/128 is present on the loopback interface of BSD.

Implementations of an IPv6 stack in a particular operating system (linux) allow for the manual configuration of both prefix lengths 64 and 10 for link-local addresses.

In another operating system the prefix length for link-local addresses can not be explicitly specified by the end user, but may be indirectly derived from two distinct textual formats by using an unspecified rule.

In yet another operating system (FreeBSD) an end user can not use a link-local address whose value is fe80:1::1; because in that OS the hosts drop incoming packets whose source or destination address matches fe80::/10 and contains a non-0 value in bits 15-31 (like fe80:1::1 does). The URL of the C code in OpenBSD that leads to make that packet drop is https://github.com/openbsd/src/blob/master/sys/netinet6/ip6_input.c

In a particular operating system (openbsd), it is possible to run SLAAC with Interface IDentifiers of length different than 64, e.g. 100; this implements [RFC7217](#). In that same operating system it is not possible to use an Interface Identifier of length 100. At the same time, in another operating system (linux) it is possible to use Interface IDentifiers of length 100, yet SLAAC does not work with IID that is not 64. In an ideal linux-bsd operating system any length of IID would be possible.

On Windows 10 Operating System it is accepted to set fe80:1::1/32 address on a physical network interface, by using the Graphical User Interface.

On MAC OS Operating System it is not possible to set fe80:1::1/32 in the command line; the 'ifconfig en1 fe80:1::1/32' command reports 'bad value'. It also reports 'bad value' with just 'fe80:1::1' (remark - no prefix length specified; note that on linux OS, when the user does not specify the prefix length to an ifconfig command, the OS will make a prefix length of value 128, and the ifconfig command will succeed.)

The loopback interface is required to have a link-local address too ([RFC4291](#)), although some OSs dont (linux). The [RFC4007](#) clarifies that, somehow.

Misconfigurations and lack of interoperability MAY arise between computers that use mixed prefix lengths for link-local addresses.

[8.](#) Historical Note

Historical note: earlier, the link-local prefix fe80::/10 and site-local prefix fec0::/10 were grouped into a common fe80::/9. If bits 10-64 were 0 then the prefix was a link-local, otherwise a site-local. The site-local addresses were later deprecated by [RFC 3879](#).

[9.](#) Example of use of LL Prefix Length 32

This figure shows two routers each with two interfaces; one such interface is connected to the other router; there are two interfaces that point elsewhere.

Internet-Draft

IPv6-LL-plen

June 2019

```

i1 ----- i2          i3-----i4
--|Router1|-----|Router2|---
-----

```

i2 address is fe80:12::1:1/32 ('12' means subnet between R1 and R2, '1' is R1, 2nd '1' is 'front' interface)
i3 address is fe80:12::2:2/32

Figure 3: Figure

One router's interface (connected to the other router) uses address fe80:12::1:1/32 and the other router's corresponding interface uses address fe80:12::2:2/32.

[10.](#) Use-Cases

[10.1.](#) Use-Case Convoy

The topology in a linear convoy of cars, in V2V manner is like this:

```

car1                                car2                                car3
-----                            -----                            -----
| IP-OBU1 | ---subnet1 ---- | IP-OBU2 | --- subnet2--- | IP-OBU3 |
-----                            -----                            -----
| in-car                            |                               |
| subnets: Ethernet, WiFi, CAN, BT, etc |                               |

```

(subnet1 is on 5880 MHz, subnet2 is on 5890 MHz)

(in the triangular convoy the figure is different)

Figure 4: Figure

Details about the restrictions with the current LL definition in the above topology:

In the above topology the restrictions with [RFC4291](#) definitions, and the FreeBSD implementations are the following:

- [RFC4291](#) needs 64bit MAC-based IIDs on the LLs on subnet1 and subnet2. The inconvenients of these restrictions are the following:
- 64bit IIDs are too long to remember and type; easy to remember addresses are good for network debugging.
- MAC-based IIDs may have some privacy risk; attackers on the road may listen to these IIDs (they are sent outside the car) and make associations that may help tracking users, like web cookies do.
- [RFC 4291](#) 54 0 bits make it impossible to assign subnet-specific link-local addresses to subnet1 and subnet2.

A [RFC4291](#)-compliant link-local address, like fe80::IID/64, assigned to

an interface on subnet2, and replying from a ping from subnet1, does not give ensurance that subnets (on 5880 MHz or on 5890 MHz) are set up wrongly. It may be that the channels are set wrong (subnets are set up wrongly) as much as it may be that that fe80::IID/64 is in the same subnet as the pinger and the channels are right.

On another hand, if the LL addresses were like fe80:1::X on subnet1 and fe80:2::Y on subnet 2, then a ping issued from subnet1 to fe80:2::X and replying OK means clearly that the channels are set wrongly.

[RFC4291](#) 54 0 bits prevent this use of subnet-specific LL addresses.

- FreeBSD OS:
- forbids the manual assignment of LL addresses on interfaces (it is impossible to ifconfig add fe80::2 on an interface).
- FreeBSD OS does not implement OCB mode. OCB mode is an

essential kind of link in vehicular communications.

Figure 5: Restrictions

Expected improvements:

Petrescu, et al. Expires December 8, 2019 [Page 11]

Internet-Draft IPv6-LL-plen June 2019

- human users type short LL addresses, like fe80:1::1 instead of long to type addresses like fe80::IID64bit
- use fe80:1::1 and fe80:2::1 in two distinct subnets; if a ping from fe80:1::1 to fe80:1::2 that does not reply means the channels are wrong; otherwise (with fe80::IID) it is impossible to say whether the channels are wrong or that wrong address was used to ping (all fe80::IID64bit) look the same to a human - they are 'random').
- BSD allowing manual configuration of LL addresses may have other benefits outside the OCB context

Figure 6: Improvements

10.2. Intuitive Next-Hop

In some IP networks only link-local addresses are used as next hops, as described in [RFC 7404](#). The next hop is part of an entry in the routing table. Make the next hop intuitive. The next hop is routinely used by sysadmin to ping and check whether it is reachable. Making the next hop intuitive can be achieved by mapping the global unicast address into both the subnet and interface id fields; in the past, experience shown that substituting just 'fe80' for the first 16

bytes of a GUA (such that to transform a GUA into an LL, to be used as next hop) ended up bleeding into the 54 0 bits required by [RFC 4291](#).

[11.](#) Security Considerations

The clarification of the definition of the prefix length of the IPv6 link-local prefix at IANA is: call it 'leading bits' and not 'prefix', or state that the IPv6 prefix length of link-local addresses is 10 decimal. This clarification has beneficial impact in the algorithm implementation for calculation of the opaque and stable Interface Identifiers for IPv6 link-local addresses. It also positively impacts some implementations of IPv6 forwarding.

A prefix length of value 65 would lead to an Interface ID length of value 63; a 63 bit IID would provide less privacy protection than a 64 bit IID, but more than a 62 bit IID. A prefix length of value 127 would lead to a 1 bit IID which would provide almost no privacy protection.

[12.](#) IANA Considerations

IANA is requested to change the name of the column head in the table that depicts the "Internet Protocol Version 6 Address Space". The name should be "The n leading bits of an address" instead of "IPv6 Prefix".

The desired effect of this change is that the IPv6 link-local prefix be "fe80::/n" and that the 10 leading bits of this prefix be 1111111010. A second effect would be that the textual representation "fe80::/10" as an IPv6 link-local prefix would disappear from that IANA page.

[13.](#) Contributors

Listed from 6man WG discussion.

[14.](#) Acknowledgements

The following persons are acknowledged for the discussion that is reflected in this draft. Not all points are reflected. Some points are copied almost entirely.

Ole Troan, Scott Timothy Morizot, Brian Carpenter, Fred Baker, Mark Smith, Peter Occil, Philip Homburg, Albert Manfredi, _–3/4’BAE (TATUYA Jinmei), Fernando Gont, Christian Huitema, Simon Hobson, Matthew Petach, Yucel Guven, Sander Steffann, Dennis Ferguson, Musa Stephen Honlue, Fred Templin, Gyan Mishra, Yu Tianpeng, Darren Dukes, Dusan Mudric.

Peter Paluch submitted the Erratum suggestion to [RFC 4291](#) about link-local addresses, and Brian Haberman rejected it, by noting 'would need' a draft. Igor Lubashev pointed to that Erratum.

[15.](#) 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, <<https://www.rfc-editor.org/info/rfc2119>>.

[Appendix A.](#) ChangeLog

The changes are listed in reverse chronological order, most recent changes appearing at the top of the list.

-21: clarified that the prefix length ranges from 1 to 127 - in general (in other cases than link local prefix).

Petrescu, et al.	Expires December 8, 2019	[Page 13]
------------------	--------------------------	-----------

Internet-Draft	IPv6-LL-plen	June 2019
----------------	--------------	-----------

-20: added brief explanation of IID len and privacy; added more examples of valid IPv6 link-local addresses; added an explanation of why the [RFC 4291](#) figure of a link local address has errors.

-19: updated authorship.

-18: updated an author address; added a justification about the support of 54 set bits in LL addresses in a mixed vendor environment; added an illustration of the [RFC4291](#) link-local address.

-17: added a new use-case for sysadmins in need of an intuitive LL

address (to check with ping) used for next-hop of routing protocols.

-16: added a description of the behaviour of `ifconfig fe80::1/32` on MAC and Windows 10 Operating Systems; added a suggestion about the use of ULA prefixes instead of LL prefixes; added a reference to an [RFC 7404](#) about the use of only LL addresses in an IPv6 network; explained the result from practice of the use of `'fe80::1:2/64'`; explained why the text says 'hidden' for '%' on some OSs; mentioned the DNS kind of solutions; added explanation of manual configuration and automation; added explanation of an example of complex to remember and type link-local addresses; added explanation of why DNS solution is a problem movement, not problem resolution.

-15: added references to [draft-farmer-6man-exceptions-64-09](#) and [draft-farmer-6man-routing-64-02](#), and interpreted them; added explanations of the solutions mentioned in WG discussion; added a use-case of car convoy with details about current restrictions of LL addressing and how a variable len plen for LL can improve the situation.

-14: updated authorship.

-13: added a Problem Statement section; added the name of the Organisation of one co-author; distinguished between 'need' and 'would need' a draft.

-12: the '64' in GUA vs '64' in LL issued by distinct sources: RA vs [RFC4291](#) respectively; the address `fe80::1/128` is present on the loopback interface of BSD; detailed, again, the distinction for 'on-link' determination; detailed, again, the distinction between 'assignment' and 'allocation'; added the fact that Cisco supports manual assignment of `fe80::1`.

-11: trying the attribute updates=[RFC4291](#), RFC4007 in the rfc tag.

-10: syntax error corrected; more explanation about how FreeBSD C code blocks `fe80::1`; clarification in IANA section, but doubtful.

-09: added a reference to [RFC 4007](#) about Zone ID in LL; added a reference to [draft-bourbaki](#) about IPv6 being classless; added the result of independent testing showing `ifconfig add fe80::1` works on linux but fails on BSD; added URL to C code in BSD flavor that may be

in charge of dropping packets whose src/dst is an LL like fe80:1::1;
added two co-authors.

-08: added explanation of which RFC requires the LL address to be present, and which requires the LL prefix to be present; named the OSs, instead of staying generic; explained that the lack of requirement of ll address on lo in [RFC4291](#) is covered by another [RFC4007](#); explained that openbsd allows variable len IID for GUAs but not for LLs, yet linux allows the reverse, and concluded on an obvious ideal.

-07: added the fact that DHCPv6 spec considers the link-local addresses to be fe80::/10; added a valuable explanation of ll behaviour of a particularly important OS.

-04: added an example advantage of using prefix length 32.

-03:

-02: corrected a typo in "fe80::/1" and added a 7-bit encoding for one persons name (in addition to the japanese-shift-jis encoding which is not understood by xml2rfc.)

Authors' Addresses

Alexandre Petrescu
CEA, LIST
CEA Saclay
Gif-sur-Yvette , Ile-de-France 91190
France

Phone: +33169089223
Email: Alexandre.Petrescu@cea.fr

Loganaden Velvindron
Cyberstorm.mu
street
city , region code
Mauritius

Phone: +phonenummer
Email: loganaden@gmail.com

Naveen Kottapalli
Benu Networks
street
City , Region code
United States

Phone: +phonenumber
Email: naveen.sarma@gmail.com

Gyan S. Mishra
Verizon Communications Inc. (VZ)
13101 Columbia Pike FDC1 Rm 304-D
Silver Spring MD 20904
United States

Phone: 301 502-1347
Email: gyan.s.mishra@verizon.com

Dusan Mudric
Avaya

Email: dmudric@avaya.com

