

IPv6 Operations Working Group (v6ops)  
Internet-Draft  
Intended status: Informational  
Expires: May 4, 2020

F. Gont  
SI6 Networks  
J. Zorz

R. Patterson  
Sky UK  
November 1, 2019

Improving the Reaction of Customer Edge Routers to Renumbering Events  
draft-gont-v6ops-cpe-slaac-renum-00

Abstract

In scenarios where network configuration information related to IPv6 prefixes becomes invalid without any explicit signaling of that condition (such as when a CPE crashes and reboots without knowledge of the previously-employed prefixes), hosts on the local network will continue using stale prefixes for an unacceptably long period of time, thus resulting in connectivity problems. This document specifies improvements to Customer Edge Routers that help mitigate the aforementioned problem for typical residential and small office scenarios.

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 May 4, 2020.

Copyright Notice

Copyright (c) 2019 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

(<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. Introduction . . . . .	2
2. Improved CPE behavior . . . . .	2
2.1. Interaction Between DHCPv6-PD and SLAAC . . . . .	3
2.2. Signaling Stale Configuration Information . . . . .	3
3. IANA Considerations . . . . .	4
4. Security Considerations . . . . .	4
5. Acknowledgments . . . . .	5
6. References . . . . .	5
6.1. Normative References . . . . .	5
6.2. Informative References . . . . .	5
Authors' Addresses . . . . .	6

## 1. Introduction

In scenarios where network configuration information related to IPv6 prefixes becomes invalid without any explicit signaling of that condition, nodes on the local network will continue using stale prefixes for an unacceptably long period of time, thus resulting in connectivity problems. This problem is documented in detail in [I-D.gont-v6ops-slaac-renum].

This document specifies improvements to Customer Edge Routers that help mitigate the aforementioned problem for residential or small office scenarios.

## 2. Improved CPE behavior

This section specifies and clarifies requirements for CPE routers (particularly when they advertise with SLAAC [RFC4862] prefixes learned via DHCPv6-PD [RFC8415]) that can help mitigate the problem discussed in Section 1. This would obviously make robustness dependent on the CPE (on which the user or ISP may have no control), as opposed to the host itself.

The updated behaviour is as follows:

- o CPE routers MUST signal stale configuration information as specified in Section 2.2

- o CPE routers MUST implement the DHCPv6-PD/SLAAC interface specified in Section 2.1.
- o CPE routers SHOULD NOT automatically send DHCPv6-PD RELEASE messages upon reboot events.

## 2.1. Interaction Between DHCPv6-PD and SLAAC

The "Preferred Lifetime" and "Valid Lifetime" of PIOs [RFC4861] corresponding to prefixes learned via DHCPv6-PD MUST NOT span past the lease time of the DHCPv6-PD prefixes. This means that the advertised "Preferred Lifetime" and "Valid Lifetime" MUST be dynamically adjusted such that the advertised lifetimes never span past the lease time of the prefixes delegated via DHCPv6-PD.

This is in line with these existing requirements from other specifications, which we reference here for clarity:

- o [RFC8415] specifies, in Section 6.3, that "if the delegated prefix or a prefix derived from it is advertised for stateless address autoconfiguration [RFC4862], the advertised preferred and valid lifetimes MUST NOT exceed the corresponding remaining lifetimes of the delegated prefix."

### RATIONALE:

- \* The lifetime values employed for the "Preferred Lifetime" (AdvPreferredLifetime) and "Valid Lifetime" (AdvValidLifetime) should never be larger than the remaining lease time for the corresponding prefix (as learned via DHCPv6-PD).
- \* The lifetime values advertised for prefixes corresponding to a prefix leased via DHCPv6-PD should be dynamically updated (rather than static values), since otherwise the advertised lifetimes would eventually span past the DHCPv6-PD lease time.

## 2.2. Signaling Stale Configuration Information

In order to phase-out stale configuration information:

- o A CPE router sending RAs that advertise dynamically-learned prefixes (e.g. via DHCPv6-PD) on an interface MUST record, on stable storage, the list of prefixes being advertised on each network segment.
- o Upon changes to the advertised prefixes, and after bootstrapping, the CPE router advertising prefix information via SLAAC should proceed as follows:

- \* Any prefixes that were previously advertised via SLAAC, but that are not currently intended for address configuration, MUST be advertised with a PIO option with the "A" bit set to 1 and the "Valid Lifetime" and a "Preferred Lifetime" set to 0.
- \* Any prefixes that were previously advertised via SLAAC as "on-link", but that are not currently not considered "on-link", MUST be advertised with a PIO option with the "L" bit set to 1 and the "Valid Lifetime" and a "Preferred Lifetime" set to 0.
- \* If both of the previous conditions are met (a prefix was previously advertised with both the "A" and "L" bits set, but is currently *\*not\** intended for address configuration and is *\*not\** considered on-link), the prefix MUST be advertised with a PIO option with both the "A" and "L" bits set to 1 and the "Valid Lifetime" and a "Preferred Lifetime" set to 0. That is, the advertisements of the previous two steps can be coalesced into a single one with both the "A" and "L" bits set.
- \* The aforementioned advertisement SHOULD be performed for at least the "Valid Lifetime" previously employed for such prefix.

The aforementioned improved behaviour assumes compliance with the following existing requirements from other specifications, which we reference here for clarity:

- o [RFC7084] specifies (requirement LE-13, in Section 4.3) that when the delegated prefix changes (i.e., the current prefix is replaced with a new prefix without any overlapping time period), "the IPv6 CE router MUST immediately advertise the old prefix with a Preferred Lifetime of zero and a Valid Lifetime of either a) zero or b) the lower of the current Valid Lifetime and two hours (which must be decremented in real time) in a Router Advertisement message as described in Section 5.5.3, (e) of [RFC4862]"

### 3. IANA Considerations

This document has no actions for IANA.

### 4. Security Considerations

This document discusses a problem that may arise in scenarios where dynamic IPv6 prefixes are employed, and proposes improvements to Customer Edge Routers [RFC7084] to mitigate the problem for residential or small office scenarios. It does not introduce new security issues.

## 5. Acknowledgments

The authors would like to thank (in alphabetical order) Mikael Abrahamsson, Luis Balbinot, Brian Carpenter, Owen DeLong, Gert Doering, Steinar Haug, Nick Hilliard, Philip Homburg, Lee Howard, Christian Huitema, Ted Lemon, Albert Manfredi, Jordi Palet Martinez, Richard Patterson, Michael Richardson, Mark Smith, Tarko Tikan, and Ole Troan, for providing valuable comments on [I-D.gont-6man-slaac-renum], on which this document is based.earlier versions of this document.

Fernando would like to thank Alejandro D'Egidio and Sander Steffann for a discussion of these issues. Fernando would also like to thank Brian Carpenter who, over the years, has answered many questions and provided valuable comments that has benefited his protocol-related work.

## 6. References

### 6.1. Normative References

- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <<https://www.rfc-editor.org/info/rfc4862>>.
- [RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <<https://www.rfc-editor.org/info/rfc8415>>.

### 6.2. Informative References

- [I-D.gont-6man-slaac-renum] Gont, F. and J. Zorz, "Reaction of Stateless Address Autoconfiguration (SLAAC) to Renumbering Events", draft-gont-6man-slaac-renum-01 (work in progress), February 2019.

[I-D.gont-v6ops-slaac-renum]

Gont, F., Zorz, J., and R. Patterson, "Reaction of Stateless Address Autoconfiguration (SLAAC) to Renumbering Events", draft-gont-v6ops-slaac-renum-00 (work in progress), July 2019.

[RFC7084] Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, DOI 10.17487/RFC7084, November 2013, <<https://www.rfc-editor.org/info/rfc7084>>.

#### Authors' Addresses

Fernando Gont  
SI6 Networks  
Segurola y Habana 4310, 7mo Piso  
Villa Devoto, Ciudad Autonoma de Buenos Aires  
Argentina

Phone: +54 11 4650 8472  
Email: [fgont@si6networks.com](mailto:fgont@si6networks.com)  
URI: <https://www.si6networks.com>

Jan Zorz  
Frankovo n. 165  
Skofja Loka  
Slovenia

Email: [jan@go6.si](mailto:jan@go6.si)

Richard Patterson  
Sky UK

Email: [richard.patterson@sky.uk](mailto:richard.patterson@sky.uk)

v6ops  
Internet-Draft  
Intended status: Informational  
Expires: March 10, 2021

J. Linkova  
Google  
September 6, 2020

Neighbor Cache Entries on First-Hop Routers: Operational Considerations  
draft-ietf-v6ops-nd-cache-init-05

Abstract

Neighbor Discovery (RFC4861) is used by IPv6 nodes to determine the link-layer addresses of neighboring nodes as well as to discover and maintain reachability information. This document discusses how the neighbor discovery state machine on a first-hop router is causing user-visible connectivity issues when a new (not being seen on the network before) IPv6 address is being used. The various approaches to mitigate the problem are described, with the proposed solution fully documented in I-D.ietf-6man-grand.

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 March 10, 2021.

Copyright Notice

Copyright (c) 2020 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 (<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. Introduction . . . . .	2
1.1. Requirements Language . . . . .	4
1.2. Terminology . . . . .	4
2. Proposed Solution . . . . .	5
2.1. Solution Requirements . . . . .	5
2.2. Solution Overview . . . . .	5
3. Solutions Considered but Discarded . . . . .	6
3.1. Do Nothing . . . . .	7
3.2. Change to the Registration-Based Neighbor Discovery . . . . .	7
3.3. Host Sending NS to the Router Address from Its GUA . . . . .	7
3.4. Host Sending Router Solicitation from its GUA . . . . .	8
3.5. Routers Populating Their Caches by Gleaning From Neighbor Discovery Packets . . . . .	9
3.6. Initiating Hosts-to-Routers Communication . . . . .	9
3.7. Transit Dataplane Traffic From a New Address Triggering Address Resolution . . . . .	10
4. IANA Considerations . . . . .	10
5. Security Considerations . . . . .	10
6. Acknowledgements . . . . .	10
7. References . . . . .	11
7.1. Normative References . . . . .	11
7.2. Informative References . . . . .	12
Author's Address . . . . .	12

## 1. Introduction

The section 7.2.5 of [RFC4861] states: "When a valid Neighbor Advertisement is received (either solicited or unsolicited), the Neighbor Cache is searched for the target's entry. If no entry exists, the advertisement SHOULD be silently discarded. There is no need to create an entry if none exists, since the recipient has apparently not initiated any communication with the target."

This approach is perfectly suitable for host-to-host communications, which are in most cases bi-directional, and it could be expected that if a host A has an neighbor cache entry for the host B IPv6 address, host B also has the corresponding entry for the host A address in its cache. However when a host communicates to off-link destinations via its first-hop router, that logic does not apply. The most typical scenario when the problem may arise is a host joining the network, forming a new address and using that address for accessing the Internet:



1. A host joins the network and receives a Router Advertisement (RA) packet from the first-hop router (either a periodic unsolicited RA or a response to a Router Solicitation sent by the host). The RA contains information the host needs to perform Stateless Address Autoconfiguration ([RFC4862]) and to configure its network stack. As in most cases the RA also contains the link-layer address of the router, the host can populate its Neighbor Cache with the router's link-local and link-layer addresses.
2. The host starts opening connections to off-link destinations. A very common use case is a mobile device sending probes to detect the Internet connectivity and/or the presence of a captive portal on the network. To speed up that process many implementations use Optimistic Duplicate Address Detection [RFC4429] which allows them to send probes before the Duplicate Address Detection (DAD) process is completed. At that moment the device neighbor cache contains all information required to send those probes (such as the default router link-local the link-layer addresses). The router neighbor cache, however, might contain an entry for the device link-local address (if the device has been performing the address resolution for the router link-local address), but there are no entries for the device global addresses.
3. Return traffic is received by the first-hop router. As the router does not have any cache entry for the host global address yet, the router starts the neighbor discovery process by creating an INCOMPLETE cache entry and then sending a Neighbor Solicitation to the Solicited Node Multicast Address. Most router implementations buffer only one data packet while resolving the packet destination address, so it would drop all subsequent packets for the host global address, until the address resolution process is completed.
4. If the host sends multiple probes in parallel, it would consider all but one of them failed. That leads to user-visible delay in connecting to the network, especially if the host implements some form of backoff mechanism and does not retransmit the probes as soon as possible.

This scenario illustrates the problem occurring when the device connects to the network for the first time or after a timeout long enough for the device address to be removed from the router's neighbor cache. However, the same sequence of events happen when the host starts using a new global address previously unseen by the router, such as a new privacy address [RFC4941] or if the router's Neighbor Cache has been flushed.

While in dual-stack networks this problem might be hidden by Happy Eyeballs [RFC8305] it manifests quite clearly in IPv6-only environments, especially wireless ones, leading to poor user experience and contributing to a negative perception of IPv6-only solutions as unstable and non-deployable.

This document discusses the operational implications of not proactively creating Neighbor Cache entries on first-hop routers and summarizes various approaches to mitigate the problem. The document provides an overview of the proposed solution which is fully described in [I-D.ietf-6man-grand].

### 1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

### 1.2. Terminology

ND: Neighbor Discovery, [RFC4861].

SLAAC: IPv6 Stateless Address Autoconfiguration, [RFC4862].

NS: Neighbor Solicitation, [RFC4861].

NA: Neighbor Advertisement, [RFC4861].

RS: Router Solicitation, [RFC4861].

RA: Router Advertisement, [RFC4861].

SLLAO: Source link-layer Address Option, an option in the ND packets containing the link-layer address of the sender of the packet, [RFC4861].

GUA: Global Unicast Address, [RFC4291].

DAD: Duplicate Address Detection, [RFC4862].

Optimistic DAD: a modification of DAD, [RFC4429].

FCFS SAVI: First-Come, First-Served Source Address Validation, [RFC6620].

## 2. Proposed Solution

### 2.1. Solution Requirements

It would be highly desirable to improve the Neighbor Discovery mechanics so routers have a usable cache entry for a host address by the time the router receives the first packet for that address. In particular:

- o If the router does not have a Neighbor Cache entry for the address, a STALE entry needs to be created.
- o The solution needs to work for Optimistic addresses as well. Devices implementing the Optimistic DAD usually attempt to minimize the delay in connecting to the network and therefore are more likely to be affected by the problem described in this document.
- o In case of duplicate addresses present in the network, the proposed solution MUST NOT override the existing entry.
- o In topologies with multiple first-hop routers the cache needs to be updated on all of them, as traffic might be asymmetric: outgoing flows leaving the network via one router while the return traffic enters the segment via another one.

In addition the solution MUST NOT exacerbate issues described in [RFC6583] and MUST be compatible with the recommendations provided in [RFC6583].

### 2.2. Solution Overview

The Neighbor Discovery is designed to allow IPv6 nodes to discover neighboring nodes' reachability and learn IPv6 to link-layer addresses mapping. Therefore ND seems to be the most appropriate tool to inform the first-hop routers about addresses the host is going to use.

Section 4.4 of [RFC4861] says:

"A node sends Neighbor Advertisements in response to Neighbor Solicitations and sends unsolicited Neighbor Advertisements in order to (unreliably) propagate new information quickly."

Propagating information about new GUA as quickly as possible is exactly what is required to solve the problem outlined in this document. Therefore the host might send an unsolicited NA with the

target link-layer address option to advertise its GUA as soon as the said address enters Optimistic or Preferred state.

The proposed solution is discussed in [I-D.ietf-6man-grand]. In summary, the following changes to [RFC4861] are suggested:

- o A node SHOULD send up to MAX\_NEIGHBOR\_ADVERTISEMENT unsolicited NA packets with the Override flag cleared to all-routers multicast address (ff02::2) as soon as one of the following events happens:
  - \* (if Optimistic DAD is used): a new Optimistic address is assigned to the node interface.
  - \* (if Optimistic DAD is not used): an address changes the state from tentative to preferred.
- o Routers SHOULD create a new STALE ND cache entry upon receiving unsolicited NAs.

It should be noted that some routing and switching platforms have implemented such behaviour already. Administrators could enable the creation of neighbor discovery cache entries based on unsolicited NA packets sent from the previously unknown neighbors on that interface.

Network devices implementing FCFS SAVI might drop Neighbor Advertisements received through a Validating Port which is in the TENTATIVE state (see Section 2.3.2 of [RFC6620]). Therefore hosts using Optimistic DAD might not benefit from the proposed solution if FCFS SAVI is implemented on the network infrastructure. [I-D.ietf-6man-grand] discusses in more details how the proposed solution interacts with SAVI.

### 3. Solutions Considered but Discarded

The problem could be addressed from different angles. Possible approaches are:

- o Just do nothing.
- o Migrate from the "reactive" Neighbor Discovery ([RFC4861]) to the registration-based mechanisms ([RFC8505]).
- o The router creates new entries in its Neighbor Cache by gleaning from Neighbor Discovery DAD messages.
- o The host initiates bidirectional communication to the router using the host GUA.

- o Making the probing logic on hosts more robust.
- o Increasing the buffer size on routers.
- o Transit dataplane traffic from an unknown address (an address w/o the corresponding neighbor cache entry) triggers an address resolution process on the router.

It should be noted that some of those options are already implemented by some vendors. The following sections discuss those approaches and the reasons they were discarded.

### 3.1. Do Nothing

One of the possible approaches might be to declare that everything is working as intended and let the upper-layer protocols to deal with packet loss. The obvious drawbacks include:

- o Unhappy users.
- o Many support tickets.
- o More resistance to deploy IPv6 and IPv6-Only networks.

### 3.2. Change to the Registration-Based Neighbor Discovery

The most radical approach would be to move away from the reactive ND as defined in [RFC4861] and expand the registration-based ND ([RFC6775], [RFC8505]) used in Low-Power Wireless Personal Area Networks (6LoWPANs) to the rest of IPv6 deployments. This option requires some investigation and discussions and seems to be excessive for the problem described in this document.

### 3.3. Host Sending NS to the Router Address from Its GUA

The host could force creating a STALE entry for its GUA in the router ND cache by sending the following Neighbor Solicitation message:

- o The NS source address is the host GUA.
- o The destination address is the default router IPv6 address.
- o The Source Link-Layer Address option contains the host link-layer address.
- o The target address is the host default router address (the default router address the host received in the RA).

The main disadvantages of this approach are:

- o Would not work for Optimistic addresses as section 2.2 of [RFC4429] explicitly prohibits sending Neighbor Solicitations from an Optimistic Address.
- o If first-hop redundancy is deployed in the network, the NS would reach the active router only, so all backup routers (or all active routers except one) would not get their neighbor cache updated.
- o Some wireless devices are known to alter ND packets and perform various non-obvious forms of ND proxy actions. In some cases, unsolicited NAs might not even reach the routers.

### 3.4. Host Sending Router Solicitation from its GUA

The host could send a router solicitation message to 'all routers' multicast address, using its GUA as a source. If the host link-layer address is included in the Source Link-Layer Address option, the router would create a STALE entry for the host GUA as per the section 6.2.6 of [RFC4861]. However, this approach can not be used if the GUA is in optimistic state: section 2.2 of [RFC4429] explicitly prohibits using an Optimistic Address as the source address of a Router Solicitation with a SLLAO as it might disrupt the rightful owner of the address in the case of a collision. So for the optimistic addresses the host can send an RS without SLLAO included. In that case the router may respond with either a multicast or a unicast RA (only the latter would create a cache entry).

This approach has the following drawbacks:

- o If the address is in the Optimistic state the RS can not contain SLLAO. As a result the router would only create a cache entry if the solicited RAs is sent as a unicast. Routers sending solicited RAs as multicast would not create a new cache entry as they do not need to send a unicast packet back to the host.
- o There might be a random delay between receiving an RS and sending a unicast RA back (and creating a cache entry) which might undermine the idea of creating the cache entry proactively.
- o Some wireless devices are known to intercept ND packets and perform various non-obvious forms of ND proxy actions. In some cases the RS might not even reach the routers.

### 3.5. Routers Populating Their Caches by Gleaning From Neighbor Discovery Packets

Routers may be able to learn about new addresses by gleaning from the DAD Neighbor Solicitation messages. The router could listen to all solicited node multicast address groups and upon receiving a Neighbor Solicitation from the unspecified address search its Neighbor Cache for the solicitation's Target Address. If no entry exists, the router may create an entry, set its reachability state to 'INCOMPLETE' and start the address resolution for that entry.

The same solution was proposed in [I-D.halpern-6man-nd-pre-resolve-addr]. Some routing vendors support such optimization already. However, this approach has a number of drawbacks and therefore should not be used as the only solution:

- o Routers need to receive all multicast Neighbor Discovery packets which might negatively impact the routers CPU.
- o If the router starts the address resolution as soon as it receives the DAD Neighbor Solicitation the host might be still performing DAD and the target address might be tentative. In that case, the host SHOULD silently ignore the received Neighbor Solicitation from the router as per the Section 5.4.3 of [RFC4862]. As a result the router might not be able to complete the address resolution before the return traffic arrives.

### 3.6. Initiating Hosts-to-Routers Communication

The host may force the router to start address resolution by sending a data packet such as ping or traceroute to its default router link-local address, using the GUA as a source address. As the RTT to the default router is lower than RTT to any off-link destinations it's quite likely that the router would start the neighbor discovery process for the host GUA before the first packet of the returning traffic arrives.

This approach has the following drawbacks:

- o Data packets to the router link-local address could be blocked by security policy or control plane protection mechanism.
- o It introduces an additional overhead for routers control plane (in addition to processing ND packets, the data packet needs to be processed as well).

- o Unless the data packet is sent to 'all routers' ff02::2 multicast address, if the network provides a first-hop redundancy then only the active router would create a new cache entry.

### 3.7. Transit Dataplane Traffic From a New Address Triggering Address Resolution

When a router receives a transit packet, it might check the presence of the neighbor cache entry for the packet source address and if the entry does not exist start address resolution process. This approach does ensure that a Neighbor Cache entry is proactively created every time a new, previously unseen GUA is used for sending offlink traffic. However this approach has a number of limitations, in particular:

- o If traffic flows are asymmetrical the return traffic might not transit the same router as the original traffic which triggered the address resolution. So the neighbor cache entry is created on the "wrong" router, not the one which actually needs the neighbor cache entry for the host address.
- o The functionality needs to be limited to explicitly configured networks/interfaces, as the router needs to distinguish between onlink addresses (ones the router needs to have Neighbor Cache entries for) and the rest of the address space.
- o Implementing such functionality is much more complicated than all other solutions as it would involve complex data-control planes interaction.

## 4. IANA Considerations

This memo asks the IANA for no new parameters.

## 5. Security Considerations

This memo documents the operational issue and does not introduce any new security considerations. Security considerations of the proposed solution are discussed in the corresponding section of [I-D.ietf-6man-grand].

## 6. Acknowledgements

Thanks to the following people (in alphabetical order) for their review and feedback: Mikael Abrahamsson, Stewart Bryant, Lorenzo Colitti, Owen DeLong, Igor Gashinsky, Fernando Gont, Tatuya Jinmei, Erik Kline, Warren Kumari, Barry Leiba, Jordi Palet Martinez, Michael



Richardson, Dave Thaler, Pascal Thubert, Loganaden Velvindron, Eric Vyncke.

## 7. References

### 7.1. Normative References

- [I-D.ietf-6man-grand]  
Linkova, J., "Gratuitous Neighbor Discovery: Creating Neighbor Cache Entries on First-Hop Routers", draft-ietf-6man-grand-01 (work in progress), July 2020.
- [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>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC4429] Moore, N., "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, DOI 10.17487/RFC4429, April 2006, <<https://www.rfc-editor.org/info/rfc4429>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <<https://www.rfc-editor.org/info/rfc4862>>.
- [RFC6583] Gashinsky, I., Jaeggli, J., and W. Kumari, "Operational Neighbor Discovery Problems", RFC 6583, DOI 10.17487/RFC6583, March 2012, <<https://www.rfc-editor.org/info/rfc6583>>.
- [RFC6620] Nordmark, E., Bagnulo, M., and E. Levy-Abegnoli, "FCFS SAVI: First-Come, First-Served Source Address Validation Improvement for Locally Assigned IPv6 Addresses", RFC 6620, DOI 10.17487/RFC6620, May 2012, <<https://www.rfc-editor.org/info/rfc6620>>.

- [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, <<https://www.rfc-editor.org/info/rfc6775>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8305] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <<https://www.rfc-editor.org/info/rfc8305>>.
- [RFC8505] Thubert, P., Ed., Nordmark, E., Chakrabarti, S., and C. Perkins, "Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery", RFC 8505, DOI 10.17487/RFC8505, November 2018, <<https://www.rfc-editor.org/info/rfc8505>>.

## 7.2. Informative References

- [I-D.halpern-6man-nd-pre-resolve-addr]  
Chen, I. and J. Halpern, "Triggering ND Address Resolution on Receiving DAD-NS", draft-halpern-6man-nd-pre-resolve-addr-00 (work in progress), January 2014.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <<https://www.rfc-editor.org/info/rfc4941>>.

## Author's Address

Jen Linkova  
Google  
1 Darling Island Rd  
Pyrmont, NSW 2009  
AU  
  
Email: [furry@google.com](mailto:furry@google.com)

v6ops  
Internet-Draft  
Intended status: Informational  
Expires: May 7, 2020

J. Palet Martinez  
The IPv6 Company  
A. D'Egidio  
Telecentro  
November 4, 2019

464XLAT Optimization  
draft-palet-v6ops-464xlat-opt-cdn-caches-04

Abstract

This document proposes possible solutions to avoid certain drawbacks of IP/ICMP Translation Algorithm (SIIT) when the destinations are available with IPv6. When SIIT is used as a NAT46 and IPv4-only devices or applications initiate traffic flows to dual-stack CDNs (Content Delivery Networks), Caches or other network resources (in the operator network or Internet), those flows will be translated back to IPv4 by a NAT64. This is the case for 464XLAT and MAP-T.

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 May 7, 2020.

Copyright Notice

Copyright (c) 2019 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 (<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. Introduction . . . . .	2
2. Requirements Language . . . . .	4
3. Problem Statement . . . . .	4
4. Solution Approaches . . . . .	6
4.1. Approach 1: DNS/Routing-based Solution . . . . .	6
4.2. Approach 2: NAT46/CLAT/DNS-proxy-EAM-based Solution . . . . .	7
4.2.1. Detection of IPv4-only devices or applications . . . . .	7
4.2.2. Detection of IPv6-enabled service . . . . .	8
4.2.3. Creation of EAMT entries . . . . .	8
4.2.4. Forwarding path via stateful NAT for existing EAMT entries . . . . .	10
4.2.5. Maintenance of the EAMT entries . . . . .	10
4.2.6. Usage example . . . . .	10
4.2.7. Behavior in case of multiple A/AAAA RRs . . . . .	11
4.2.8. Behavior in presence/absence of DNS64 . . . . .	11
4.2.9. Behavior when using literal addresses or non IPv6-compliant APIs . . . . .	11
4.2.10. False detection of a dual-stack host as IPv4-only . . . . .	11
4.2.11. Behaviour in presence of Happy Eyeballs . . . . .	12
4.2.12. Behavior in case of Foreign DNS . . . . .	12
4.3. Approach 3: NAT46/CLAT-provider-EAM-based Solution . . . . .	13
5. IPv6-only Services become accessible to IPv4-only devices/apps . . . . .	14
6. Conclusions . . . . .	14
7. Security Considerations . . . . .	15
8. IANA Considerations . . . . .	15
9. Acknowledgements . . . . .	15
10. References . . . . .	15
10.1. Normative References . . . . .	15
10.2. Informative References . . . . .	16
Authors' Addresses . . . . .	17

## 1. Introduction

Different transition mechanisms, typically in the group of the so-called IPv6-only with IPv4aaS (IPv4-as-a-Service), such as 464XLAT ([RFC6877]) or MAP-T ([RFC7599]), allow IPv4-only devices or applications to connect with IPv4 services in Internet, by means of a NAT46 SIIT (IP/ICMP Translation Algorithm) as described by [RFC7915].

This is done by the implementation of SIIT at the CE (Customer Edge) Router or sometimes at the end-device, for example, the UE (User

Equipment) in cellular networks. This functionality is the CLAT (Customer Translator) in the case of 464XLAT.

The NAT46/CLAT (WAN side) is connected by IPv6-only to the operator network, which in turn, will have a reverse function, the NAT64 ([RFC6146]), known as PLAT (Provider Translator) in the case of 464XLAT. This allows to translate the IPv6-only flow back to IPv4, in order to forward it to Internet.

The translation of the packet headers is done using the IP/ICMP translation algorithm defined in [RFC7915] and algorithmically translating the IPv4 addresses to IPv6 addresses following [RFC6052].

In the case of 464XLAT, a DNS64 ([RFC6147]) optionally is in charge of the synthesis of AAAA records from the A records, so they can use a NAT64, without the need of doing a double-translation by means of the CLAT. However, the DNS64 is not useful for the IPv4-only devices or applications in the LANs, as they will not be able to use the AAAA records.

A typical 464XLAT deployment is depicted in Figure 1.

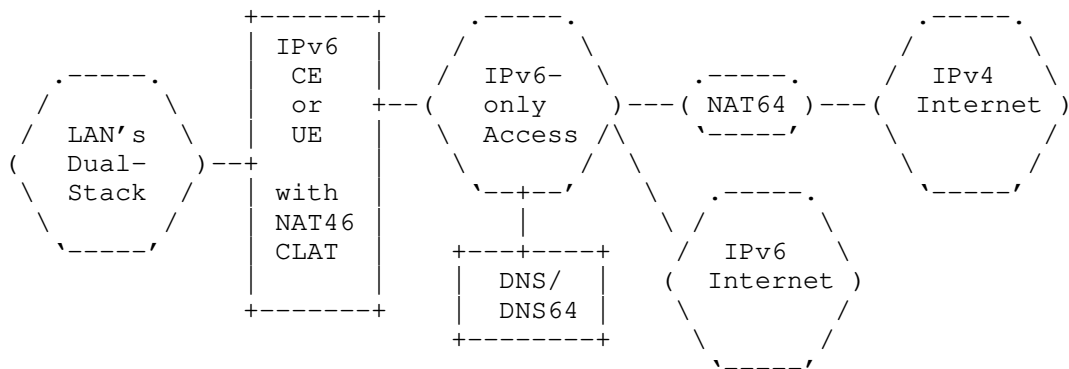


Figure 1: Typical 464XLAT Deployment

As it can be observed in the preceding picture, the situation is the same, regardless of in case of a wired network with a CE Router or a cellular network where a UE is connecting other devices (which may be IPv4-only or have IPv4-only apps), by means of a tethering functionality.

If the operator is providing direct access to Content Delivery Networks (CDNs), caches, or other resources, and they are dual-stacked, the situation can be described as shown in Figure 2.

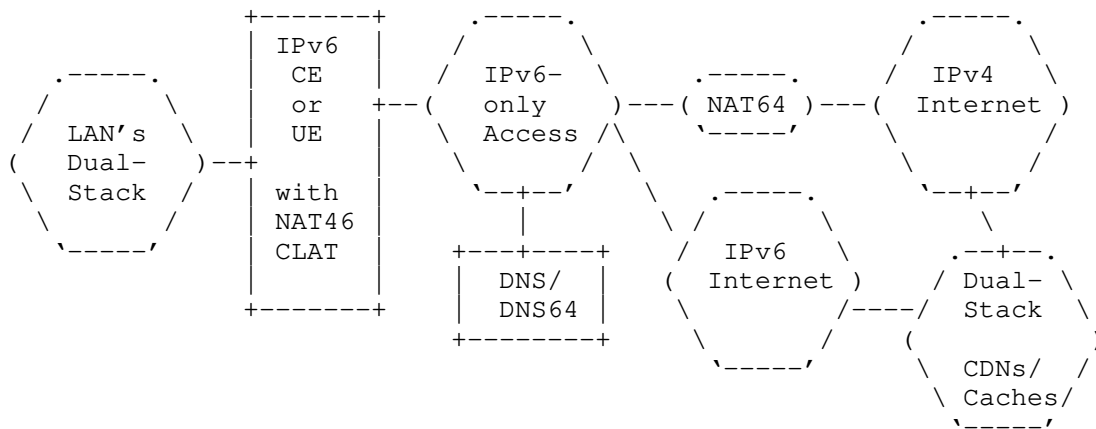


Figure 2: Typical 464XLAT Deployment with CDNs/Caches

## 2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 3. Problem Statement

If the devices or applications in the customer LAN are IPv6-capable, then the access to the CDNs, caches or other resources, will be made in an optimized way, by means of IPv6-only, not using the NAT64, as depicted in Figure 3.

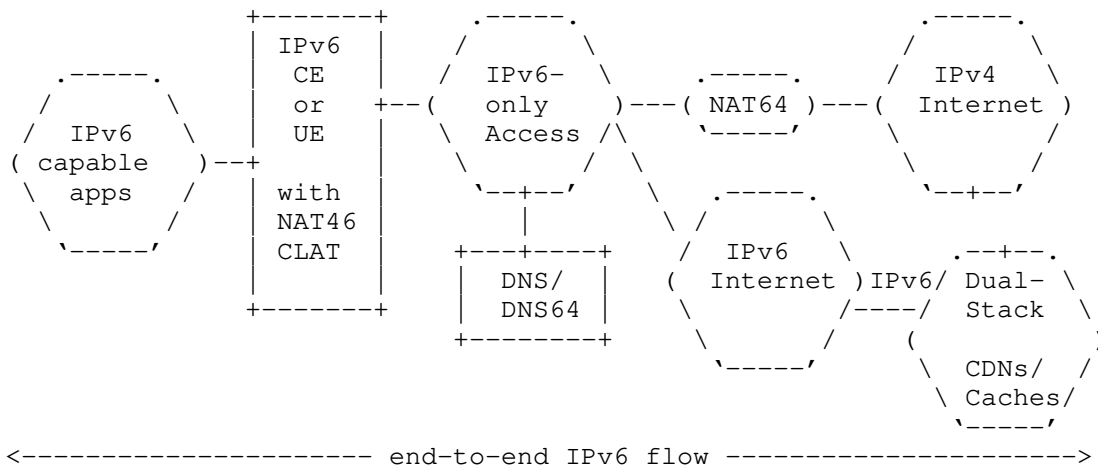


Figure 3: 464XLAT access to CDNs/Caches by IPv6-capable apps

However, if the devices or applications are IPv4-only, for example, most of the SmartTVs and Set-Top-Boxes available today, a non-optimal double translation will occur (NAT46 at the CLAT and NAT64 at the PLAT), as illustrated in Figure 4.

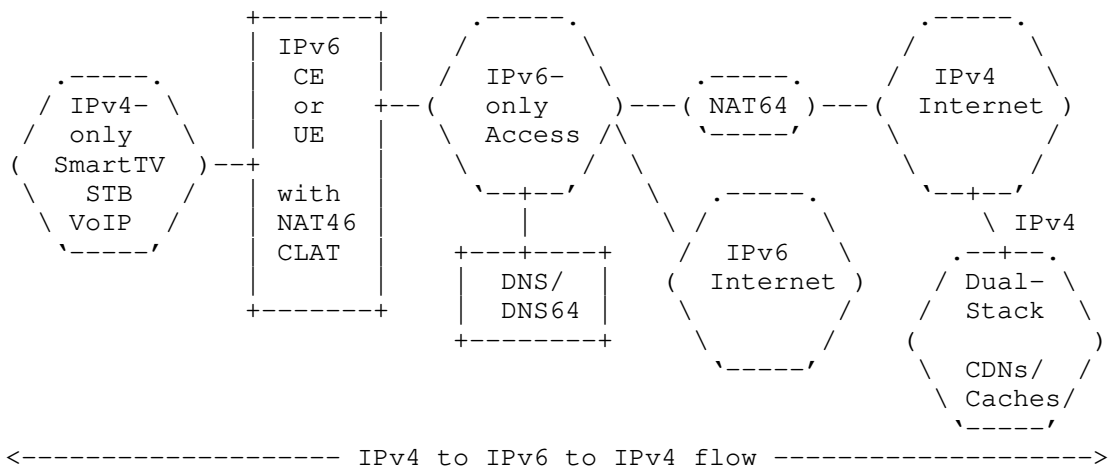


Figure 4: 464XLAT access to CDNs/Caches by IPv4-only apps

Clearly, this is a non-optimal situation, as it means that even if there is a dual-stack service, the NAT46/CLAT translated IPv4 to IPv6 traffic flow, is unnecessarily translated back to IPv4, traversing the stateful NAT64. This has a direct impact in the need to scale the NAT64 beyond what will be actually needed if possible solutions,

in order to keep using the IPv6 path towards those services, are considered.

As shown in the Figure 4, this is also the case for many other services, not just CDNs or caches, such as VoIP access to the relevant operator infrastructure, which may be also dual-stack. This is true as well for many other dual-stack or IPv6-enabled services, which may be directly reachable from the operator infrastructure, even if they are not part of it, for example peering agreements, services in IXs, etc. In general, this will become a more frequent situation for many other services, which are not yet dual-stack.

For simplicity, across the rest of this document, references to CDNs/caches, should be understood, unless otherwise stated, as any dual-stacked resources.

This document looks into different possible solution approaches in order to optimize the IPv4-only SIIT translation providing a direct path to IPv6-capable services, as depicted in Figure 5.

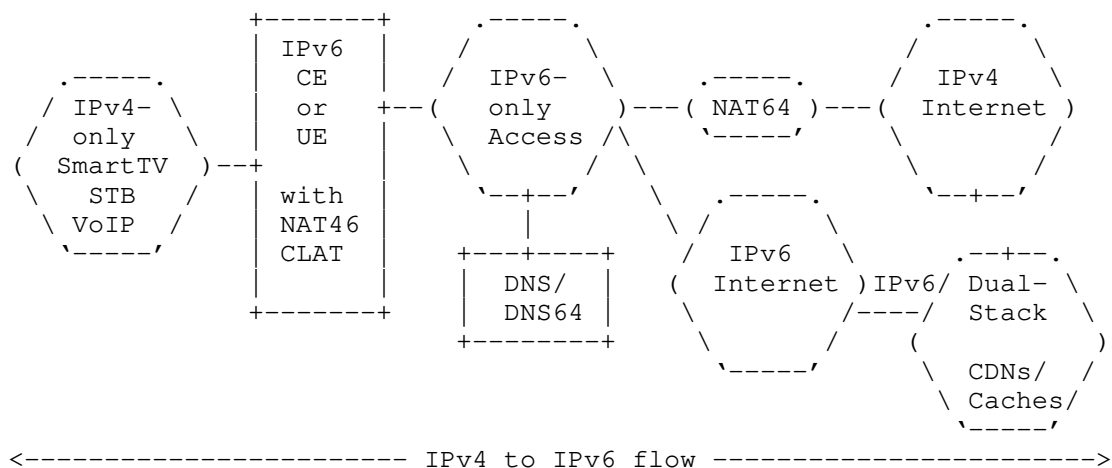


Figure 5: Optimized 464XLAT access to CDNs/Caches by IPv4-only apps

## 4. Solution Approaches

### 4.1. Approach 1: DNS/Routing-based Solution

Because the IPv4-only devices will not be able to query for AAAA records, the NAT46/CLAT/CE will translate the IPv4 addresses from the A record for the CDN/cache destination, using the WKP or NSP, as configured by the operator.



If the CDN/cache provider is able to configure, in the relevant interfaces of the CDN/caches, the same IPv6 addresses that will naturally result as the translated destination addresses for the queried A records, preceded by the WKP or NSP, then having more specific routing prefixes, will result in traffic to those destinations being directly forwarded towards those interfaces, instead of needing to traverse the NAT64.

For example, let's suppose a provider using the WKP (64:ff9b::/96) and a SmartTV querying for www.example.com:

www.example.com	A	192.0.2.1
NAT46/CLAT translated to		64:ff9b::192.0.2.1
CDN IPv6 interface must be		64:ff9b::192.0.2.1
Operator must have a specific route to		64:ff9b::192.0.2.1

Note: Examples using text representation as per Section 2.3 of [RFC6052].

Because the WKP is non-routable, this solution will only be possible if the CDN/cache is in the same ASN as the provider network, or somehow interconnected without routing thru Internet.

This solution has the additional drawback of the operational complexity/issues added to the operation of the CDN/cache, and the need to synchronize any IPv4 interface address changes with the relevant IPv6 ones, and possibly with routing.

#### 4.2. Approach 2: NAT46/CLAT/DNS-proxy-EAM-based Solution

If the NAT46/CLAT/CE, as commonly is the case, is also a DNS proxy/stub resolver, it is possible to modify the behavior and create an "internal" interaction among both of them.

This approach uses the existing IPv4 and IPv6 addresses in the A and AAAA records, respectively, so no additional complexity/issues added to the CDN/caches operations.

The following sub-sections detail this approach and provide a step-by-step example case.

##### 4.2.1. Detection of IPv4-only devices or applications

The assumption is that, typically a dual-stack device will prefer using IPv6 as the DNS transport. So, when there is a DNS query, transported with IPv4, for an A record, and there is not a query for the AAAA record from the same IPv4 source (to the same destination), the DNS proxy/stub resolver can infer that, most probably, it is an

IPv4-only device or application.

It needs to be remarked that, if the detection of the IPv4-only device or application is done incorrectly (either not detecting it or by a false detection), no harm is caused. In the worst case, optimization will not be performed, at least, at the time being. However, optimization maybe performed later on, if a new detection succeeds (for example, another device using the same A record).

#### 4.2.2. Detection of IPv6-enabled service

In the case of an IPv4-only detected device or application, the DNS proxy/stub resolver MUST actually perform an additional AAAA query, unless the information is already present in the Additional Section, as per Section 3 of [RFC3596]. Note that the NAT46/CLAT MUST already know the WKP or NSP being used in that network. If the response contains at least one IPv6 address not using the WKP/NSP, it means that the destination is IPv6-enabled (because at least one of the IPv6 addresses is not synthesized). This means that it is possible for the NAT46/CLAT, to create an Explicit Address Mapping ([RFC7757]).

#### 4.2.3. Creation of EAMT entries

This way, an EAM Table (EAMT used for short, across the rest of this document) is created/maintained automatically by the DNS proxy/stub resolver in the NAT46/CLAT, and the NAT46/CLAT is responsible to prioritize any available entries in the EAMT, versus the use of any synthetic AAAA.

In order to create the EAMT entry, to determine if there is an AAAA record after an A record query, it is suggested to use the same delay value (50 milliseconds) as the "Resolution Delay" indicated by Happy Eyeballs [RFC8305]. This avoids a slight NAT64 overload and flapping between destination addresses (IPv4/IPv6), which may impact some applications, at the cost of a small extra delay for the initial communication setup, when the EAMT entry doesn't yet exist.

Each EAMT entry will contain, the fields already described in [RFC7757] and a few new ones:

1. ID: EAMT Entry Index (optional).
2. IPv4 address/prefix: By default, the prefix length is 32 bits.
3. IPv6 address/prefix: By default, the prefix length is 128 bits.
4. TTL: Because the optimization will make use of the AAAA (IPv6

address), the TTL for the EAMT entry must be the one of the AAAA RR. In normal conditions the TTL for both A and AAAA records, of a given FQDN, should be the same, so this ensures a proper behavior if there is any DNS mismatch.

5. FQDN: The one that originated the A query for this EAMT entry. Required in order to ensure a correct detection of cases such as the use of reverse-proxy with a single IPv4 address to multiple IPv6 addresses.
6. Valid/Invalid: When set to 1, means that this EAMT entry MUST NOT be used and consequently no optimization performed. It may be used also for an explicit configuration (GUI, CLI, provisioning system, etc.) to disallow optimization for any IPv4 addresses.
7. Auto/Static: When set to 1, means that this EAMT entry has been manually/statically configured, for example by means of an explicit configuration (GUI, CLI, provisioning system, etc.), so it doesn't expire with TTL.

When a new EAMT entry is first automatically created, it is marked as "Valid" and "Auto" (both bits cleared). If a subsequent A query, with a different FQDN, results in an IPv4 address that has already an EAMT entry and a different IPv6 address, it means that some reverse-proxy or similar functionality is being used by the IPv6-enabled service. In this case, the existing EAMT entry will be marked as "Invalid" (bit set). No new EAMT entry is created for that IPv4 address. Otherwise, the optimization will only allow to access the first set of IPv4/IPv6/FQDN, which may break the access to other FQDN that share the same IPv4 address and different IPv6 addresses.

In this case the EAMT entry will still expire according the TTL, which allows to re-enable optimization if a new query for the A record has changed the situation. For example, maybe the reverse-proxy has been removed, or there is now only a single device using it, so at the time being, the optimization is again possible without creating troubles to other hosts.

Note that when an EAMT entry is marked as "invalid", it will not affect the devices or applications, as they will still be able to use the regular CLAT+NAT64 flow, of course, without the optimization.

\*\*\*\*\* Open question regarding TTL and maybe FQDN and valid/auto bits. Is this always a good thing to do for EAM? Should this document update [RFC7757] to support this by default? Or it is just an "extension" as per section 3.1 of [RFC7757].

#### 4.2.4. Forwarding path via stateful NAT for existing EAMT entries

Following this approach, if there is a valid EAMT entry, for a given IPv4-destination, the IPv6-native path pointed by the IPv6 address of that EAMT entry, will take precedence versus the NAT64 path, so the traffic will not be forwarded to the NAT64.

However, this is not sufficient to ensure that individual applications are able to keep existing connections. In many cases, audio and video streaming may use a single TCP connection lasting from minutes to hours. Instead, the CDN TTLs may be configured in the range from 10 to 300 seconds in order to allow new resolutions to switch quickly and to handle large recursive resolvers (with hundreds of thousands of clients behind them).

Consequently, the EAMT entries should not be used directly to establish a forwarding path, but instead, to create a stateful NAT entry for the 4-tuple for the duration of the session/connection.

#### 4.2.5. Maintenance of the EAMT entries

The information in the EAMT MUST be kept timely-synchronized with the AAAA records TTL's, so the EAMT entries MUST expire on the AAAA TTL expiry and consequently be deleted.

However, EAMT entries with the Auto/Static bit set, will not be deleted.

#### 4.2.6. Usage example

Using the same example as in the previous approach:

www.example.com	A	192.0.2.1
	AAAA	2001:db8::a:b:c:d
EAMT entry	192.0.2.1	2001:db8::a:b:c:d
NAT46/CLAT translated to		2001:db8::a:b:c:d
CDN IPv6 interface already is		2001:db8::a:b:c:d
Operator already has a specific route to		2001:db8::a:b:c:d

The following is an example of the CE behavior after the previous case has already created an EAMT entry and a reverse-proxy is detected:

1. A query for www.another-example.com A RR is received
2. www.another-example.com A 192.0.2.1
3. www.another-example.com AAAA 2001:db8::e:e:f:f

4. A conflict has been detected

5. The existing EAMT entry for 192.0.2.1 is set as invalid

#### 4.2.7. Behavior in case of multiple A/AAAA RRs

If multiple A and/or AAAA records are available, the DNS proxy/stub resolver MUST follow existing procedures to choose each one. In other words, the chosen pair of A/AAAA records doesn't present any different result compared with a situation when this mechanism is not used.

#### 4.2.8. Behavior in presence/absence of DNS64

This mechanism performs the same in both cases, if a DNS64 is present/used and if it is not present/used. This is explained because the mechanism is only relevant for destinations which don't have AAAA records, and in those cases DNS64 is not relevant. Furthermore, because as indicated in Section 4.2.2, the EAMT entry is not created when the service is IPv6-enabled. This is relevant because 464XLAT can be deployed/used with and without a DNS64.

#### 4.2.9. Behavior when using literal addreses or non IPv6-compliant APIs

Because the EAMT entries are only created when the NAT46/CLAT/CE proxy/stub DNS is being used, any devices or applications that don't use DNS, will not create the relevant entries.

They will be however optimized if devices or applications using DNS, at some point, query for the same A RRs, or if EAMT entries are statically configured.

#### 4.2.10. False detection of a dual-stack host as IPv4-only

If a dual-stack host is issuing the A query using IPv4 transport, and the AAAA query using IPv6 transport, or using different IPv4 addresses for the A and AAAA queries, the EAMT entry will be created. However, this EAMT entry may not be used by dual-stack devices or applications, because those devices or applications should prefer IPv6. If the host is preferring IPv4 for connecting to the CDN/cache or IPv6-enabled service, it will be actually using the NAT46/CLAT, including the EAMT entry and consequently IPv6, so this mechanism will be correcting an undesirable behavior. This is a special case, which actually seems to be an incoherent host or application implementation.

However, if other IPv4-only devices or applications subsequently need to connect to the same IPv6-enabled service, they will take advantage

of the already existing EAMT entry, and consequently use the IPv6-optimised path.

#### 4.2.11. Behaviour in presence of Happy Eyeballs

Happy Eyeballs [RFC8305] is only available in dual-stack hosts. Consequently, is not affected by this mechanism because both, the A and the AAAA queries should be issued by the host as soon after one another as possible. However, if the same NAT46/CLAT/CE is serving IPv4-only hosts and dual-stack hosts and both of them are using the same destinations, an EAMT entry will be created for that destination. Consequently, a Happy Eyeballs fallback to IPv4 will actually be using the relevant EAMT entry IPv6 destination. This has the disadvantage that the IPv4-IPv6-IPv4 translation path can't be used by Happy Eyeballs-enabled applications. However, this may be actually considered as a good thing, in the sense that an operator is interested in knowing as soon as possible, if the IPv6-only network is not performing correctly, because that means also IPv4 will not be working. If the issue is related to extra IPv6 delay versus the IPv4 delay, Happy Eyeballs will not be able to offer a significative advantage here, but it looks like an acceptable trade-off.

Note that when using 464XLAT, the WAN link of the NAT46/CLAT/CE is IPv6-only. So even if Happy Eyeballs is present, the fallback to IPv4-only typically, will be slower than native IPv6 itself, because the added detail in the NAT46+NAT64 translations, when not using this optimization.

#### 4.2.12. Behavior in case of Foreign DNS

Devices or applications may use DNS servers from other networks. For a complete description of reasons for that, refer to Section 4.4 of [I-D.ietf-v6ops-nat64-deployment]. In the case the DNS is modified, or some devices or applications use other DNS servers, the possible scenarios and the implications are:

- a. Devices configured to use a DNS proxy/resolver which is not the CE/NAT46/CLAT. In this case this optimization will not work, because the EAMT entry will not be created based on their own flows. Nevertheless, the EAMT entry may be created by other devices using the same destinations. However, the lack of EAMT entry, will not impact negatively in the user's devices/applications (the optimization is not performed). It should be noticed that users commonly, don't change the configuration of devices such as SmartTVs or STBs (if they do, some other functionalities, such as CDN/caches optimizations may not work as well), so this only happens typically if the vendor is doing it on-purpose and for good well-known reasons.

- b. DNS privacy/encryption. Hosts or applications that use mechanisms for DNS privacy/encryption, such as DoT ([RFC7858], [RFC8094]), DoH ([RFC8484]) or DoQ ([I-D.huitema-quic-dnsquic]), will not make use of the stub/proxy resolver, so the same considerations as for the previous case apply.
- c. Users that modify the DNS in their Operating Systems. This is quite frequent, however commonly Operating Systems are dual-stack, so aren't part of the problem statement described by this document and will not be adversely affected.
- d. Users that modify the DNS in the CE. This is less common. In this case, this optimization is not adversely affected, because it doesn't depend on the operator DNS, it works only based on the internal CE interaction between the NAT46/CLAT and the stub/proxy resolver. Note that it may be affected if the operator offers different "DNS views" or "split DNS", however this is not related to this optimization and will anyway impact in the other possible operator optimizations (i.e. CDN/cache features).
- e. Combinations of the above ones. No further impact, than the one already described, is observed.

#### 4.3. Approach 3: NAT46/CLAT-provider-EAM-based Solution

Instead of using the DNS proxy/stub resolver to create the EAMT entries, the operator may push this table (or parts of it) into the CE/NAT46/CLAT, by using configuration/management mechanisms.

This solution has the advantage of not being affected by any DNS changes from the user (the EAMT is created by the operator) and ensures a complete control from the operator. However, it may impact the cases of devices with a DNS configured by the vendor.

In general, most of the considerations from the previous approach will apply.

One more advantage of this solution is that the EAMT pairs doesn't need to match the "real" IPv4/IPv6 addresses available in the A/AAAA records, as shown in the next example.

www.example.com	A	192.0.2.1
	AAAA	2001:db8::a:b:c:d
EAMT pulled/pushed entry	192.0.2.1	2001:db8::f:e:d:c
NAT46/CLAT translated to		2001:db8::f:e:d:c
CDN IPv6 interface already is		2001:db8::f:e:d:c
Operator already has a specific route to		2001:db8::f:e:d:c

EAMT may contain TTLs which probably are derived from DNS ones, or alternatively, a global TTL for the full table.

An alternative way to configure the table, is that the CE is actually pulling the table (or parts of it) from the operator infrastructure. In this case it will be mandatory that the entries have individual TTLs, again probably derived from the DNS ones.

The major drawback of this approach is that it requires a new protocol, or an extension to existing ones, in order to push or pull the EAMT, in addition to the possible impact in terms of bandwidth each time the CEs reboot, or an EAMT must be pushed to all the CEs, etc.

## 5. IPv6-only Services become accessible to IPv4-only devices/apps

One of the issues with the IPv6 deployment, is that those services which become IPv6-only in Internet, aren't reachable by IPv4-only devices and applications. This means that new content providers must support dual-stack even for new services, even while IPv4 public addresses aren't available.

If NAT46/CLAT/DNS-proxy-EAM approach (Section 4.2) is chosen, it can be complemented to resolve this issue, by means of making sure that IPv6-only destinations have one A resource record (even an invalid one), despite they aren't actually connected to IPv4. This will mean that those services will work fine if there is a NAT46/CLAT, and will have no impact if that one doesn't exist, not a different situation than not having an A resource record.

In fact, it may become an incentive for the IPv6 deployment in Internet services and provides the option to use an IPv4 address (maybe anycast) for the "non-valid" A resource record, that points to a "universal" web page (maybe hosted by IETF?) that displays a warning such as "Sorry, you don't IPv6 support in your operator, so this service is not available for you".

## 6. Conclusions

NAT46/CLAT/DNS-proxy-EAM approach (Section 4.2) seems the right solution for optimizing the access to dual-stack services, whether they are located inside or outside the ISP.

Having this type of optimization facilitates and increases the usage of IPv6, even for IPv4-only devices and applications, at the same time that decreases the use of the NAT64.

SIIT already has a SHOULD for EAM support. Should 464XLAT be updated



by this document so the CLAT has a MUST for EAM support?.

Should we recommend having A records for IPv6-only services in Internet? The A record may point to a "reserved" or "special" IPv4 address. A web page IPv4-only hosted by IETF(?) showing "sorry this web page/service is only available from IPv6 enabled operators"?.

Open question: Should we consider any other risks? If CE's implementing this optimization create troubles, it may bring the content providers to switch back to IPv4-only. So possible failure cases need to be carefully considered for every possible solution approach.

## 7. Security Considerations

This document does not have any new specific security considerations.

## 8. IANA Considerations

This document does not have any new specific IANA considerations, unless we decide to define a "special reserved IPv4 address".

## 9. Acknowledgements

The authors would like to acknowledge the inputs of Erik Nygren, Fred Baker, Martin Hunek and TBD ...

## 10. References

### 10.1. 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, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", STD 88, RFC 3596, DOI 10.17487/RFC3596, October 2003, <<https://www.rfc-editor.org/info/rfc3596>>.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, DOI 10.17487/RFC6052, October 2010, <<https://www.rfc-editor.org/info/rfc6052>>.

- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <<https://www.rfc-editor.org/info/rfc6146>>.
- [RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, DOI 10.17487/RFC6147, April 2011, <<https://www.rfc-editor.org/info/rfc6147>>.
- [RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT: Combination of Stateful and Stateless Translation", RFC 6877, DOI 10.17487/RFC6877, April 2013, <<https://www.rfc-editor.org/info/rfc6877>>.
- [RFC7599] Li, X., Bao, C., Dec, W., Ed., Troan, O., Matsushima, S., and T. Murakami, "Mapping of Address and Port using Translation (MAP-T)", RFC 7599, DOI 10.17487/RFC7599, July 2015, <<https://www.rfc-editor.org/info/rfc7599>>.
- [RFC7757] Anderson, T. and A. Leiva Popper, "Explicit Address Mappings for Stateless IP/ICMP Translation", RFC 7757, DOI 10.17487/RFC7757, February 2016, <<https://www.rfc-editor.org/info/rfc7757>>.
- [RFC7915] Bao, C., Li, X., Baker, F., Anderson, T., and F. Gont, "IP/ICMP Translation Algorithm", RFC 7915, DOI 10.17487/RFC7915, June 2016, <<https://www.rfc-editor.org/info/rfc7915>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8305] Schinazi, D. and T. Pauly, "Happy Eyeballs Version 2: Better Connectivity Using Concurrency", RFC 8305, DOI 10.17487/RFC8305, December 2017, <<https://www.rfc-editor.org/info/rfc8305>>.

## 10.2. Informative References

- [I-D.huitema-quic-dnsquic]  
Huitema, C., Shore, M., Mankin, A., Dickinson, S., and J. Iyengar, "Specification of DNS over Dedicated QUIC Connections", draft-huitema-quic-dnsquic-07 (work in progress), September 2019.

- [I-D.ietf-v6ops-nat64-deployment]  
Palet, J., "Additional NAT64/464XLAT Deployment Guidelines in Operator and Enterprise Networks", draft-ietf-v6ops-nat64-deployment-08 (work in progress), July 2019.
- [RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <<https://www.rfc-editor.org/info/rfc7858>>.
- [RFC8094] Reddy, T., Wing, D., and P. Patil, "DNS over Datagram Transport Layer Security (DTLS)", RFC 8094, DOI 10.17487/RFC8094, February 2017, <<https://www.rfc-editor.org/info/rfc8094>>.
- [RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <<https://www.rfc-editor.org/info/rfc8484>>.

#### Authors' Addresses

Jordi Palet Martinez  
The IPv6 Company  
Molino de la Navata, 75  
La Navata - Galapagar, Madrid 28420  
Spain

Email: [jordi.palet@theipv6company.com](mailto:jordi.palet@theipv6company.com)  
URI: <http://www.theipv6company.com/>

Alejandro D'Egidio  
Telecentro  
Argentina

Email: [adegidio@telecentro.net.ar](mailto:adegidio@telecentro.net.ar)

v6ops  
Internet-Draft  
Intended status: Informational  
Expires: May 7, 2020

J. Palet Martinez  
The IPv6 Company  
November 4, 2019

IPv6 Point-to-Point Links  
draft-palet-v6ops-p2p-links-04

Abstract

This document describes different alternatives for configuring IPv6 point-to-point links, considering the prefix size, numbering choices and prefix pool to be used.

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 May 7, 2020.

Copyright Notice

Copyright (c) 2019 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 (<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. Introduction . . . . .	2
2. Requirements Language . . . . .	3
3. The Ping-Pong Problem in Point-to-Point Links . . . . .	3
4. Prefix Size Choices . . . . .	3
4.1. Rationale for using /64 . . . . .	3
4.2. Rationale for using /127 . . . . .	4
4.3. Rationale for using /126 and Other Options . . . . .	5
4.4. A Possible Middle-Term Choice . . . . .	5
5. Numbering Choices . . . . .	5
5.1. GUA (Global Unicast Addresses) . . . . .	5
5.2. ULA (Unique Local Addresses) . . . . .	5
5.3. Link-Local Addresses Only . . . . .	6
6. Prefix Pool Choices . . . . .	7
7. /64 from Customer Prefix for point-to-point links . . . . .	7
7.1. Numbering Interfaces . . . . .	7
7.2. Routing Aggregation of the Point-to-Point Links . . . . .	8
7.3. DHCPv6 Considerations . . . . .	9
7.4. Router Considerations . . . . .	9
8. Security Considerations . . . . .	10
9. IANA Considerations . . . . .	10
10. Acknowledgements . . . . .	10
11. References . . . . .	10
11.1. Normative References . . . . .	10
11.2. Informative References . . . . .	11
Author's Address . . . . .	12

## 1. Introduction

There are different alternatives for numbering IPv6 point-to-point links, and from an operational perspective, there may have different advantages or disadvantages that need to be taken in consideration under the scope of each specific network architecture design.

[RFC6164] describes using /127 prefixes for inter-router point-to-point links, using two different address pools, one for numbering the point-to-point links and another one for delegating the prefixes at the end of the point-to-point link. However, this doesn't exclude other choices.

This document describes alternative approaches, for the prefix size, the numbering of the link and the prefix pool.

The proposed approaches are suitable for those point-to-point links connecting ISP to customers, but not limited to those cases, and in fact, all them are being used by a relevant number of networks worldwide, in several different scenarios (service providers,

enterprise networks, etc.).

## 2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 3. The Ping-Pong Problem in Point-to-Point Links

Some point-to-point links may present the ping-pong problem, (a forwarding loop). The fundamental root cause of this problem is an IPv6 implementations not performing full Neighbor Discovery (NS/NA) on addresses that the prefix says could exist on the link.

IPv6 implementations are assuming that all addresses within the prefix must exist at the other end of the point-to-point link, and send the traffic straight onto the link. If the address doesn't exist, and there is a covering route back in the other direction, the ping-pong problem occurs.

Full Neighbor Discovery is doing more than just resolving the link-layer address of an IPv6 address. Neighbor Discovery is also determining if the address exists. Even if a point-to-point link doesn't have link-layer addresses to resolve, ND determining if an address exists on the link is very beneficial because it will prevent the ping-pong problem occurring entirely regardless of the IPv6 prefix length being used on the link.

## 4. Prefix Size Choices

[RFC7608] already discusses about the IPv6 prefix length recommendations for forwarding, and the need for routing and forwarding implementations to ensure that longest-prefix-match works on any prefix length. So, in this document, we concentrate in the most commonly used choices, not excluding other options.

### 4.1. Rationale for using /64

The IPv6 Addressing Architecture ([RFC4291]) specifies that all the Interface Identifiers for all the unicast addresses (except for 000/3) are required to be 64 bits long and to be constructed in Modified EUI-64 format.

The same document also mandates the usage of the predefined subnet-router anycast address, which has cleared to zero all the bits that

do not form the subnet prefix.

Using /64 is the most common scenario and currently the best practice by the number of service providers using this approach compared to others.

Using a /64 has the advantage of being future proof and avoids renumbering, in the event that new standards take advantage of the 64 bits for other purposes, or the link becomes a point-to-multipoint, or there is a need to use more addresses in the link (e.g., monitoring equipment, managed bridges).

It has been raised also the issue of some hardware having limitations in using prefixes longer than /64, for example using extra hardware resources.

Section 5. of [RFC6164] describes possible issues when using /64 for the point-to-point links, such as the ping-pong and the neighbor cache exhaustion. However, it also states that they can be mitigated by other means, including the latest ICMPv6 [RFC4443] ND [RFC4861]. Indeed, considering the publication date of that document, those issues should not be any longer a concern. The fact is that many operators worldwide, today use /64 without any concerns, as vendors have taken the necessary code updates.

Consequently, we shall conclude that it is a valid approach to use /64 prefixes for the point-to-point links.

#### 4.2. Rationale for using /127

[RFC6164] already do a complete review of reasons why /127 is a good approach vs other options. However, it needs to be considered that it was published a number of years ago, and most of the hardware today already incorporate mitigations.

It should be noted that, when using a /127 prefix, configuration of each of the addresses within the /127 prefix, at each respective end of the link, must be actively validated by the network operator. A missing /127 address from one end of the link, with a local route pointing out that end of the link that covers the missing /127 address, such as a default route, causes a "ping-pong" scenario to exist for the missing /127 address. The link could still be successfully carrying transit traffic, and IPv6 will not report any errors, because IPv6 doesn't require or nor check to ensure all interfaces attached to a link has addresses from all prefixes assigned to the link, excepting the Link-Local prefix per [RFC4291].

It is a valid approach to use /127 for the point-to-point links,

however is not future proof considering the comments from the previous section, and older equipment may not support it.

#### 4.3. Rationale for using /126 and Other Options

/126 was considered by [RFC3627], and despite this document has been obsoleted, because was considering /127 as harmful, the considerations in Section 4.3 are still valid.

The same document describes options such as /112 and /120, and all those are commonly used in worldwide IPv6 deployments [IPv6-Survey], though in a lesser degree than /64 or /127.

Consequently, we shall conclude that /126, /120 and /112 are valid approaches for the point-to-point links.

#### 4.4. A Possible Middle-Term Choice

A possible "middle-term" approach, will be to allocate a /64 for each point-to-point link, but use just one /127 out of it, making it future proof and at the same time avoiding possible issues indicated in the previous sections.

### 5. Numbering Choices

IPv6 provides different unicast addressing scopes which can be considered when numbering a point-to-point link.

It has been reported that certain hardware may consume resources when using numbered links. This is a very specific situation that may need to be consider on a case by case basis.

#### 5.1. GUA (Global Unicast Addresses)

Using GUA is the most common approach. It provides full functionality for both end-points of the point-to-point link and consequently, facilitates troubleshooting.

#### 5.2. ULA (Unique Local Addresses)

Some networks use ULAs for numbering the point-to-point links. This approach may cause numerous problems when carrying Internet traffic and therefore, is strongly discouraged. For example, if the CE needs to send an ICMPv6 message to a host outside that network (to the Internet), the packet with ULA source address will not get thru and PMTUD will break, which in turn will completely break that IPv6 connection when the MTU is not the same for all the path.



ULAs are IPv6 private addresses, not intended to be used as source or destination addresses across the Internet. This issue also exists in IPv4 when using [RFC1918] addresses on links carrying IPv4 Internet traffic. [RFC6752] discusses this issue for IPv4, with much of the discussion applying similarly to IPv6 and ULAs.

However, this approach is valid if, following Section 2.2 of [RFC4443], and despite using ULA for the point-to-point link, the router is configured with at least one GUA and the source of the ICMPv6 messages are always a GUA, per the IPv6 Default Address Selection algorithm [RFC6724].

### 5.3. Link-Local Addresses Only

Some networks leave the point-to-point links with only Link-Local addresses used at both ends of the link. This is sometimes improperly referred as "unnumbered", because the Link-Local addresses are also "numbers". Furthermore, [RFC4291] requires that all interfaces attached to a link have at least a Link-Local "number" or address from the Link-Local prefix.

[RFC7404] (Using Only Link-Local Addressing inside an IPv6 Network) discusses pros and cons of this alternative, which in general apply for the point-to-point links.

While this choice might work if the point-to-point link is terminated in a router, which typically will get configured with a suitable routable GUA or ULA, it will not work for devices that can't be further configured, for example if they do not support DHCPv6-PD. This is the case for hosts, when the Operating System is not expected to be a DHCPv6-PD client and are therefore left without any usable GUA to allow traffic forwarding.

In the case of a router, the route for the assigned prefix is pointed towards the link-local address on the router WAN port and the default route on the router is pointed towards the link-local address on the upstream network equipment port.

This choice seems easier to implement, compared the previous ones, but it also brings some drawbacks, such as difficulties with troubleshooting and monitoring. For example, link local addresses do not appear in traceroute, so it makes more difficult to locate the exact point of failure.

It is more useful in scenarios where it is known that only a router will be attached to the point-to-point link, and where the configured address of the router is known. Non-routers connecting to a network, which can't initiate DHCPv6-PD might experience problems and will

stay unnumbered upon connection, if a /64 prefix is not used to number the link. This may be also the case for routers, which will not be able to complete the DHCPv6-PD in unnumbered links.

The considerations indicated in the previous section, regarding not using ULA as source address of ICMPv6 messages, and instead ensuring there is at least one GUA configured for that, also apply if link-locals are used for the point-to-point link.

## 6. Prefix Pool Choices

The logic choice seems to use a dedicated pool of IPv6 addresses, as this is the way we are "used to" with IPv4. Actually, this is done often by means of different IPv6 pools at every PoP in a service provider network.

A possible benefit of using a dedicated IPv6 pool, is that allows applying security policies without harming the customers. This is only true if customers always have a CE at their end of the WAN link.

However, the fact that the default IPv6 link size is /64 and commonly multiple /64's are assigned to a single customer, provides an interesting alternative approach for combining "best practices" described in the precedent sections.

The following section depicts this alternative.

## 7. /64 from Customer Prefix for point-to-point links

Using a /64 from the customer prefix, in addition to the advantages already indicated when using /64, simplifies the addressing plan.

The use of /64 also facilitates an easier way for routing the shorter aggregated prefix into the point-to-point link. Consequently it simplifies the "view" of a more unified addressing plan, providing an easier path for following up any issue when operating IPv6 networks and typically, will have a great impact in saving expensive hardware resources (lower usage of TCAM, typically by half).

This mechanism would not work in broadcast layer two media that rely on ND, because it will try ND for all the addresses within the shorter prefix that is being routed thru the point-to-point link.

### 7.1. Numbering Interfaces

Often, in point-to-point links, hardware tokens are not available, or there is the need to keep certain bits (u, g) cleared, so the links can be manually numbered sequentially with most of the bits cleared

to zero. This numbering makes as well easier to remember the interfaces, which typically will become numbered as 0 (with 63 leading zero bits) for the provider side and 1 (with 63 leading zero bits) for the customer side.

Using interface identifiers as 0 and 1 is not only a very simple approach, but also a very common practice. Other different choices can as well be used as required in each case.

On the other hand, using the EUI-64, makes it more difficult to remember and handle the interfaces, but provides an additional degree of protection against port (actually address) scanning as described at [RFC7707].

## 7.2. Routing Aggregation of the Point-to-Point Links

Following this approach and assuming that a shorter prefix is typically delegated to a customer, for example a /48, it is possible to simplify the routing aggregation of the point-to-point links. Towards this, the point-to-point link may be numbered using the first /64 of the /48 delegated to the customer.

Let's see a practical example:

- o A service provider uses the prefix 2001:db8::/32 and is using 2001:db8:aaaa::/48 for a given customer.
- o Instead of allocating the point-to-point link from a different addressing pool, it may use 2001:db8:aaaa::/64 (which is the first /64 subnet from the 2001:db8:aaaa::/48) to number the link.
- o This means that, in the case the non-EUI-64 approach is used, the point-to-point link may be numbered as 2001:db8:aaaa::1/64 for the provider side and 2001:db8:aaaa::2/64 for the customer side.
- o Note that using the first /64 and interface identifiers 1 and 2 is a very common practice. However other values may be chosen according to each case specific needs.

In this way, as the same address pool is being used for both, the prefix and the point-to-point link, one of the advantages of this approach is to make very easy the recognition of the point-to-point link that belongs to a given customer prefix, or in the other way around, the recognition of the prefix that is linked by a given point-to-point link.

For example, making a trace-route to debug any issue to a given address in the provider network, will show a straight view, and it

becomes unnecessary one extra step to check a database that correlate an address pool for the point-to-point links and the customer prefixes, as all they are the same.

Moreover, it is possible to use the shorter prefix as the provider side numbering for the point-to-point link and keep the /64 for the customer side. In our example, it will become:

- o Point-to-point link at provider side: 2001:db8:aaaa::1/48
- o Point-to-point link at customer side: 2001:db8:aaaa::2/64

This provides one additional advantage as in some platforms the configuration may be easier saving one step for the route of the delegated prefix (no need for two routes to be configured, one for the delegated prefix, one for the point-to-point link). It is possible because the longest-prefix-match rule.

The behavior of this type of configuration has been successfully deployed in different operator and enterprise networks, using commonly available implementations with different routing protocols, including RIP, BGP, IS-IS, OSPF, along static routing, and no failures or interoperability issues have been reported.

### 7.3. DHCPv6 Considerations

As stated in [RFC3633], "the requesting router MUST NOT assign any delegated prefixes or subnets from the delegated prefix(es) to the link through which is received the DHCP message from the delegating router", however the approach described in this document is still useful in other DHCPv6 scenarios or non-DHCPv6 scenarios.

Furthermore, [RFC3633] was updated by Prefix Exclude Option for DHCPv6-based Prefix Delegation ([RFC6603]), precisely to define a new DHCPv6 option, which covers the case described by this document.

Moreover, [RFC3769] has no explicit requirement that avoids the approach described in this document.

### 7.4. Router Considerations

This approach is being used by operators in both, residential/SOHO and enterprise networks, so the routers at the customer end for those networks MUST support [RFC6603] if DHCPv6-PD is used.

In the case of Customer Edge Routers there is a specific requirement ([RFC7084]) WPD-8 (Prefix delegation Requirements), marked as SHOULD for [RFC6603]. However, in a scenario where the approach described

in this document is followed, together with DHCPv6-PD, the CE Router MUST support [RFC6603].

## 8. Security Considerations

This document does not have any new specific security considerations.

## 9. IANA Considerations

This document does not have any new specific IANA considerations.

## 10. Acknowledgements

The author would like to acknowledge the inputs of Mikael Abrahamsson, Brian Carpenter, Eric Vyncke, Mark Smith and TBD.

Acknowledge is also due to my co-authors of RIPE-690 (Best Current Operational Practice for Operators: IPv6 prefix assignment for end-users - persistent vs non-persistent, and what size to choose, <https://www.ripe.net/publications/docs/ripe-690>) and global community, which provided valuable inputs which have been key for this document.

Acknowledgement to co-authors, Cesar Olvera and Miguel Angel Diaz, of a previous related document (draft-palet-v6ops-point2point, 2006), as well as inputs for that version from Alain Durand, Chip Popoviciu, Daniel Roesen, Fred Baker, Gert Doering, Olaf Bonness, Ole Troan, Pekka Savola and Vincent Jardin, are also granted.

## 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, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, DOI 10.17487/RFC3633, December 2003, <<https://www.rfc-editor.org/info/rfc3633>>.
- [RFC3769] Miyakawa, S. and R. Droms, "Requirements for IPv6 Prefix Delegation", RFC 3769, DOI 10.17487/RFC3769, June 2004, <<https://www.rfc-editor.org/info/rfc3769>>.

- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC6603] Korhonen, J., Ed., Savolainen, T., Krishnan, S., and O. Troan, "Prefix Exclude Option for DHCPv6-based Prefix Delegation", RFC 6603, DOI 10.17487/RFC6603, May 2012, <<https://www.rfc-editor.org/info/rfc6603>>.
- [RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <<https://www.rfc-editor.org/info/rfc6724>>.
- [RFC7084] Singh, H., Beebe, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, DOI 10.17487/RFC7084, November 2013, <<https://www.rfc-editor.org/info/rfc7084>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

## 11.2. Informative References

- [IPv6-Survey] Palet Martinez, J., "IPv6 Deployment Survey (Residential/Household Services)", January 2018, <<https://indico.uknwf.org.uk/event/41/contribution/5/material/slides/0.pdf>>.
- [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <<https://www.rfc-editor.org/info/rfc1918>>.

- [RFC3627] Savola, P., "Use of /127 Prefix Length Between Routers Considered Harmful", RFC 3627, DOI 10.17487/RFC3627, September 2003, <<https://www.rfc-editor.org/info/rfc3627>>.
- [RFC6164] Kohno, M., Nitzan, B., Bush, R., Matsuzaki, Y., Colitti, L., and T. Narten, "Using 127-Bit IPv6 Prefixes on Inter-Router Links", RFC 6164, DOI 10.17487/RFC6164, April 2011, <<https://www.rfc-editor.org/info/rfc6164>>.
- [RFC6752] Kirkham, A., "Issues with Private IP Addressing in the Internet", RFC 6752, DOI 10.17487/RFC6752, September 2012, <<https://www.rfc-editor.org/info/rfc6752>>.
- [RFC7404] Behringer, M. and E. Vyncke, "Using Only Link-Local Addressing inside an IPv6 Network", RFC 7404, DOI 10.17487/RFC7404, November 2014, <<https://www.rfc-editor.org/info/rfc7404>>.
- [RFC7608] Boucadair, M., Petrescu, A., and F. Baker, "IPv6 Prefix Length Recommendation for Forwarding", BCP 198, RFC 7608, DOI 10.17487/RFC7608, July 2015, <<https://www.rfc-editor.org/info/rfc7608>>.
- [RFC7707] Gont, F. and T. Chown, "Network Reconnaissance in IPv6 Networks", RFC 7707, DOI 10.17487/RFC7707, March 2016, <<https://www.rfc-editor.org/info/rfc7707>>.

## Author's Address

Jordi Palet Martinez  
The IPv6 Company  
Molino de la Navata, 75  
La Navata - Galapagar, Madrid 28420  
Spain

EMail: [jordi.palet@theipv6company.com](mailto:jordi.palet@theipv6company.com)  
URI: <http://www.theipv6company.com/>

Internet Engineering Task Force  
Internet-Draft  
Intended status: Informational  
Expires: April 16, 2020

M. Smith  
October 14, 2019

Default IPv6 Local Only Addressing for Non-Internet Devices  
draft-smith-v6ops-local-only-addressing-00

Abstract

For certain types or models of devices it should be clear and obvious that, by default, they should not be reachable from the global IPv6 Internet, or able to reach the global IPv6 Internet, even though the network they are attached to provides global IPv6 Internet connectivity. This memo proposes that these types of devices refuse to configure and use global IPv6 Internet addresses by default.

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 April 16, 2020.

Copyright Notice

Copyright (c) 2019 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 (<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. Introduction . . . . .	2
1.1. Requirements Language . . . . .	3
2. Default Local Only Addresses . . . . .	3
3. SLAAC Address Configuration . . . . .	3
4. DHCPv6 Address Configuration . . . . .	4
5. Permitted Incoming and Outgoing Connections . . . . .	5
6. Example Device Types . . . . .	5
7. Security Considerations . . . . .	6
8. Acknowledgements . . . . .	6
9. Change Log [RFC Editor please remove] . . . . .	6
10. References . . . . .	6
10.1. Normative References . . . . .	6
10.2. Informative References . . . . .	7
Author's Address . . . . .	7

## 1. Introduction

For some types of IPv6 devices, their access to the Internet, and access from the Internet, should be prevented under normal circumstances. Examples of these types of devices are network attached paper printers, local network file and print servers, and various types of "Internet of Things" devices.

As a basic and fundamental prevention measure, these types of devices can have their ability to reach the Internet, or to be reachable from the Internet, prevented by only attaching them to local network links and routers that only support and provide Unique Local Unicast Addresses (ULA) [RFC4193]. These nodes and devices would then only have addresses from within the Link-Local [RFC4291] prefix and ULA prefix(es) available on the link.

In some networks, it may not be possible or easy to use "ULA Only" links to isolate these devices. For example, these devices may need to be attached to the same link as other devices that do have global IPv6 addresses and can reach the Internet. This may be because these local network only devices may need to be discoverable by devices with global Internet addresses via link-only discovery protocols such as multicast DNS (mDNS) [RFC6762].

This memo proposes that when it is clear to a device manufacturer that a device should be isolated from the Internet by default, due its functions and role, the device only configures Link-Local Addresses and non-Internet usable addresses such as ULAs on its

interfaces, even though the link may support and provide global IPv6 Internet addresses. This memo also proposes that these devices should have available an override configuration switch that causes these devices to configure addresses from all prefixes available on the link, including global IPv6 Internet address prefixes.

These types of devices are known as Local Only Address devices in this memo.

### 1.1. Requirements Language

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].

## 2. Default Local Only Addresses

By default, a Local Only Address device MUST only configure Link-Local and non-global IPv6 addresses, currently Unique Local Addresses [RFC4193], on its network interfaces.

The device SHOULD provide a default override configuration option, known as Configure All IPv6 Addresses, allowing the device to configure addresses from all available IPv6 address prefixes on the link, including global IPv6 addresses.

This Configure All IPv6 Addresses configuration switch SHOULD be available via a device's administrative interface. There may be some devices where it is clear that attachment to the public IPv6 Internet should never occur; for these devices, this configuration switch SHOULD be omitted. An example would be IoT devices such as Smart Grid Advanced Metering Infrastructure (AMI) devices [RFC6272].

(Further thought, there could probably be an RA PIO flag or similar to override this default for all devices on a link, and a similar DHCPv6 flag/option. Would mean this ID would be in 6man WG scope rather than v6ops.)

## 3. SLAAC Address Configuration

By default, when the Local Only Addresses device is processing IPv6 Router Advertisement Prefix Information Options (PIOs) [RFC4861], to configure IPv6 interface addresses via SLAAC [RFC4862], the device MUST only configure addresses using PIOs that provide a prefix that falls within the Unique Local Unicast Address [RFC4193] address range of fc::/7, should the A or autonomous address-configuration flag be set for the PIO.

By default, if there are no ULA prefix PIOs in the received RAs, or no ULA prefix PIOs with the A flag set, the Local Only Addresses device MUST only configure IPv6 Link-Local addresses on its network interface.

By default, if there are ULA prefix PIOs that do not have the A flag set, they MUST be processed per standard RA PIO processing for other flags. For example, a PIO for a ULA prefix, with the A flag unset, and the L or on-link flag set, is still processed, and is asserting that the specified ULA prefix is on-link.

If the Configure All IPv6 Addresses configuration switch is enabled, then the Local Only Addresses device MUST process all IPv6 RA PIOs received for SLAAC address configuration, per [RFC4862], from that point in time onwards.

If the Configure All IPv6 Addresses configuration switch is changed from enabled to disabled, then the Local Only Addresses device MUST immediately remove all global IPv6 addresses from the interface, immediately terminating all upper layer application connections that are using these global IPv6 addresses. This is regardless of any remaining preferred and valid lifetimes for the addresses [RFC4862]. This is immediately enforcing the intention that this Local Address Only device should now be isolated from the global IPv6 Internet.

#### 4. DHCPv6 Address Configuration

By default, if the Local Only Addresses device is using DHCPv6 [RFC8415] for address acquisition and configuration, the device MUST ignore any received IPv6 addresses in either IA\_TA or IA\_NA options, that not with the ULA prefix of fd00::/7.

Be default, if the Local Only Addresses device does not receive any IA\_TA or IA\_NA options containing addresses from within the ULA prefix of fd00::/7, then the device MUST only configure Link-Local addresses on its interface.

Note that a device using DHCPv6 for address acquisition and configuration could also be using SLAAC for address configuration in parallel. All of the SLAAC Address Configuration procedures described previously will also apply.

If the Configure All IPv6 Addresses configuration switch is enabled, then the Local Only Addresses device MUST then acquire and accept all IPv6 addresses provided by the DHCPv6 server in either IA\_NA or IA\_TA options.

If the Configure All IPv6 Addresses configuration switch is changed from enabled to disabled, then the Local Only Addresses device MUST immediately remove all global IPv6 addresses from the interface, immediately terminating all upper layer application connections that are using these global IPv6 addresses. This is regardless of any remaining preferred and valid lifetimes for the addresses [RFC4862]. This is immediately enforcing the intention that this Local Address Only device should now be isolated from the global IPv6 Internet. The Local Address Only device should gracefully close its DHCPv6 leases for these global IPv6 addresses, returning them to the DHCPv6 server's address pool.

## 5. Permitted Incoming and Outgoing Connections

By default, a Local Address Only device MUST NOT accept any upper layer connections from any global IPv6 addresses. Any connection attempts from global IPv6 addresses MUST be silently ignored, meaning that no connection failure ICMPv6 or transport layer protocol error messages are sent. Connection attempts from other address types, such as Link-Local or ULA addresses are accepted, should other Local Address Only device security policies permit them.

As a Local Address Only device, by default, MUST NOT have any valid global IPv6 addresses, outgoing connections using global IPv6 addresses should not occur.

An application may attempt to overcome this global IPv6 address constraint by constructing packets itself that contain a global IPv6 address source address. These types of packets MUST be dropped by the Local Address Only device, and a system message alerting the Local Only Address device operator to this possible security violation SHOULD be logged with appropriate severity.

If the Configure All IPv6 Addresses configuration switch is changed from disabled to enabled, all incoming and outgoing connections from any type of IPv6 address are permitted, assuming any other Local Address Only device security policies permit them.

## 6. Example Device Types

The following are some example types of devices for which this default Local Only Address behaviour should be implemented. This is not exhaustive, and should be judged by a vendor on a device by device type basis, by considering the device's purpose, and most typical and common deployment scenarios.

- o Network attached paper printers

- o File Server and Network Attached Storage
- o IoT devices such as Advanced Metering Infrastructure "smart" electricity meters [RFC6272].
- o Networking device Operations, Administration and Maintenance (OAM) and Out-of-Band (OOB) management interfaces, used for and by device monitoring and management protocols such as SNMP [RFC1157].

## 7. Security Considerations

This memo is specifically about increasing device security by limiting their network accessibility and reachability by default, when it suits the intended use of the device. It is imposing a fundamental truth and constraint that if a device cannot be reached by a packet, the device cannot be attacked by the contents of that packet. By default, suitable devices are not reachable from the Internet, and therefore cannot be attacked from devices on the Internet.

However, this security mechanism is both baseline and coarse. It does not protect against attacks from other devices that can reach the Local Only Address device via ULA or Link-Local addresses.

This mechanism should be considered a minimum measure for suitable devices to implement. It should be combined with other security mechanisms, such as IPsec [RFC4301] for IPv6 layer authentication and application layer authentication.

## 8. Acknowledgements

Review and comments were provided by YOUR NAME HERE!

This memo was prepared using the xml2rfc tool.

## 9. Change Log [RFC Editor please remove]

draft-smith-v6ops-local-only-addressing-00, initial version,  
2019-09-15

## 10. References

### 10.1. Normative References

- [RFC1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin,  
"Simple Network Management Protocol (SNMP)", RFC 1157,  
DOI 10.17487/RFC1157, May 1990,  
<<https://www.rfc-editor.org/info/rfc1157>>.

- [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>>.

## 10.2. Informative References

- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<https://www.rfc-editor.org/info/rfc4193>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <<https://www.rfc-editor.org/info/rfc4862>>.
- [RFC6272] Baker, F. and D. Meyer, "Internet Protocols for the Smart Grid", RFC 6272, DOI 10.17487/RFC6272, June 2011, <<https://www.rfc-editor.org/info/rfc6272>>.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <<https://www.rfc-editor.org/info/rfc6762>>.
- [RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <<https://www.rfc-editor.org/info/rfc8415>>.

Author's Address

Mark Smith  
PO BOX 521  
HEIDELBERG, VIC 3084  
AU

Email: markzzzsmith@gmail.com