

6MAN
Internet-Draft
Intended status: Informational
Expires: August 10, 2014

B. Carpenter, Ed.
Univ. of Auckland
T. Chown
Univ. of Southampton
F. Gont
SI6 Networks / UTN-FRH
S. Jiang
Huawei Technologies Co., Ltd
A. Petrescu
CEA, LIST
A. Yourtchenko
cisco
February 6, 2014

Analysis of the 64-bit Boundary in IPv6 Addressing
draft-carpenter-6man-why64-01

Abstract

The IPv6 unicast addressing format includes a separation between the prefix used to route packets to a subnet and the interface identifier used to specify a given interface connected to that subnet. Historically the interface identifier has been defined as 64 bits long, leaving 64 bits for the prefix. This document discusses the reasons for this fixed boundary and the issues involved in treating it as a variable boundary.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 10, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Scenarios for prefixes longer than /64	3
2.1. Insufficient address space delegated	4
2.2. Concerns over ND cache exhaustion	4
3. Interaction with IPv6 specifications	5
4. Possible areas of breakage	7
5. Experimental observations	8
5.1. Survey of the processing of Neighbor Discovery options with prefixes other than /64	8
5.2. Other Observations	10
6. Privacy issues	11
7. Implementation and deployment issues	11
8. Conclusion	13
9. Security Considerations	13
10. IANA Considerations	14
11. Acknowledgements	14
12. Change log [RFC Editor: Please remove]	14
13. References	14
13.1. Normative References	14
13.2. Informative References	18
Authors' Addresses	19

1. Introduction

IPv6 addresses were originally chosen to be 128 bits long to provide flexibility and new possibilities, rather than simply relieving the IPv4 address shortage by doubling the address size to 64 bits. The notion of a 64-bit boundary in the address was introduced after the initial design was done. There were two motivations for introducing it. One was the original "8+8" proposal [DRAFT-odell] that eventually led to ILNP [RFC6741], which required a fixed point for

the split between local and wide-area parts of the address. The other was the expectation that EUI-64 MAC addresses would become widespread in place of 48-bit addresses, coupled with the plan at that time that auto-configured addresses would normally be based on interface identifiers derived from MAC addresses.

The IPv6 addressing architecture [RFC4291] specifies that a unicast address is divided into n bits of subnet prefix followed by $(128-n)$ bits of interface identifier (IID). Since IPv6 routing is entirely based on variable length subnet masks, there is no architectural assumption that n has any particular fixed value. However, RFC 4291 also describes a method of forming interface identifiers from IEEE EUI-64 hardware addresses [IEEE802] and this does specify that such interface identifiers are 64 bits long. Various other methods of forming interface identifiers also specify a length of 64 bits. This has therefore become the de facto length of almost all IPv6 interface identifiers. One exception is documented in [RFC6164], which standardises 127-bit prefixes for inter-router links.

Recently it has been clarified that the bits in an IPv6 interface identifier have no particular meaning and should be treated as opaque values [I-D.ietf-6man-ug]. Therefore, there are no bit positions in the currently used 64 bits that need to be preserved. The addressing architecture as modified by [I-D.ietf-6man-ug] now states that "For all unicast addresses, except those that start with the binary value 000, Interface IDs are required to be 64 bits long. If derived from an IEEE MAC-layer address, they must be constructed in Modified EUI-64 format."

The question is often asked why the boundary is set rigidly at /64. This limits the length of a routing prefix to 64 bits, whereas architecturally, and from the point of view of routing protocols, it could be anything (in theory) between /1 and /128 inclusive. Here, we only discuss the question of a shorter IID, allowing a longer routing prefix.

The purpose of this document is to analyse the issues around this question. We make no proposal for change, but we do analyse the possible effects of a change.

2. Scenarios for prefixes longer than /64

In this section we describe existing scenarios where prefixes longer than /64 have been used or proposed.

2.1. Insufficient address space delegated

A site may not be delegated a sufficiently large prefix from which to allocate a /64 prefix to all of its internal subnets. In this case the site may either determine that it does not have enough address space to number all its network elements and thus, at the very best, be only partially operational, or it may choose to use internal prefixes longer than /64 to allow multiple subnets and the hosts within them to be configured with addresses.

In this case, the site might choose, for example, to use a /80 per subnet, in combination with hosts using either manually configured addressing or DHCPv6.

Scenarios that have been suggested where an insufficient prefix might be delegated include home or small office networks, vehicles, building services and transportation services (road signs, etc.). It should be noted that the homenet architecture text [I-D.ietf-homenet-arch] states that a CPE should consider the lack of sufficient address space to be an error condition, rather than using prefixes longer than /64 internally.

Another scenario occasionally suggested is one where the Internet address registries actually begin to run out of IPv6 prefix space, such that operators can no longer assign reasonable prefixes to users in accordance with [RFC6177]. We mention this scenario here for completeness, and we briefly analyze it in Section 7.

2.2. Concerns over ND cache exhaustion

A site may be concerned that it is open to neighbour discovery (ND) cache exhaustion attacks, whereby an attacker sends a large number of messages in rapid succession to a series of (most likely inactive) host addresses within a specific subnet, in an attempt to fill a router's ND cache with ND requests pending completion, in so doing denying correct operation to active devices on the network.

An example would be to use a /120 prefix, limiting the number of addresses in the subnet to be similar to an IPv4 /24 prefix, which should not cause any concerns for ND cache exhaustion. Note that the prefix does need to be quite long for this scenario to be valid. The number of theoretically possible ND cache slots on the segment needs to be of the same order of magnitude as the actual number of hosts. Thus small increases from the /64 prefix length do not have a noticeable impact: even 2^{32} potential entries, a factor of two billion decrease compared to 2^{64} , is still more than enough to exhaust the memory on current routers.

As in the previous scenario, hosts would likely be manually configured with addresses, or use DHCPv6.

It should be noted that several other mitigations of the ND cache attack are described in [RFC6583], and that limiting the size of the cache and the number of incomplete entries allowed would also defeat the attack.

3. Interaction with IPv6 specifications

The precise 64-bit length of the Interface ID is widely mentioned in numerous RFCs describing various aspects of IPv6. It is not straightforward to distinguish cases where this has normative impact or affects interoperability. This section aims to identify specifications that contain an explicit reference to the 64-bit size. Regardless of implementation issues, the RFCs themselves would all need to be updated if the 64-bit rule was changed, even if the updates were small.

First and foremost, the RFCs describing the architectural aspects of IPv6 addressing explicitly state, refer and repeat this apparently immutable value: Addressing Architecture [RFC4291], Reserved Interface Identifiers [RFC5453], ILNP [RFC6741]. Customer Edge routers impose /64 for their interfaces [RFC7084]. Only the IPv6 Subnet Model [RFC5942] refers to the assumption of /64 prefix length as a potential implementation error.

Numerous IPv6-over-foo documents make mandatory statements with respect to the 64-bit length of the Interface ID to be used during the Stateless Autoconfiguration. These documents include [RFC2464] (Ethernet), [RFC2467] (FDDI), [RFC2470] (Token Ring), [RFC2492] (ATM), [RFC2497] (ARCnet), [RFC2590] (Frame Relay), [RFC3146] (IEEE 1394), [RFC4338] (Fibre Channel), [RFC4944] (IEEE 802.15.4), [RFC5072] (PPP), [RFC5121] [RFC5692] (IEEE 802.16), [RFC2529] (6over4), [RFC5214] (ISATAP), [I-D.templin-aerolink] (AERO), [I-D.ietf-6lowpan-btle], [I-D.ietf-6man-6lobac], [I-D.brandt-6man-lowpanz].

To a lesser extent, the address configuration RFCs themselves may in some way assume the 64-bit length of an Interface ID (SLAAC for the link-local addresses, DHCPv6 for the potentially assigned EUI-64-based IP addresses, Default Router Preferences [RFC4191] for its impossibility of Prefix Length 4, Optimistic Duplicate Address Detection [RFC4429] which computes 64-bit-based collision probabilities).

The MLDv2 protocol [RFC3810] mandates all queries be sent with the fe80::/64 link-local source address prefix and subsequently bases the

querier election algorithm on the link-local subnet prefix length of length /64.

The IPv6 Flow Label Specification [RFC6437] gives an example of a 20-bit hash function generation which relies on splitting an IPv6 address in two equally-sized 64bit-length parts.

The basic transition mechanisms [RFC4213] refer to IIDs of length 64 for link-local addresses, and other transition mechanisms such as Teredo [RFC4380] assume the use of IIDs of length 64. Similar assumptions are found in 6to4 [RFC3056] and 6rd [RFC5969]. Translation-based transition mechanisms such as NAT64 and NPTv6 have some dependency on prefix length, discussed below.

The proposed method [I-D.ietf-v6ops-64share] of extending an assigned /64 prefix from a smartphone's cellular interface to its WiFi link relies on prefix length, and implicitly on the length of the Interface ID, to be valued at 64.

The CGA and HBA specifications rely on the 64-bit identifier length (see below), as do the Privacy extensions [RFC4941] and some examples in IKEv2bis [RFC5996].

464XLAT [RFC6877] explicitly mentions acquiring /64 prefixes. However, it also discusses the possibility of using the interface address on the device as the endpoint for the traffic, thus potentially removing this dependency.

[RFC2526] reserves a number of subnet anycast addresses by reserving some anycast IIDs. An anycast IID so reserved cannot be less than 7 bits long. This means that a subnet prefix length longer than /121 is not possible, and a subnet of exactly /121 would be useless since all its identifiers are reserved. It also means that half of a /120 is reserved for anycast. This could of course be fixed in the way described for /127 in [RFC6164], i.e., avoiding the use of anycast within a /120 subnet.

While preparing this document, it was noted that many other IPv6 specifications refer to mandatory alignment on 64-bit boundaries, 64-bit data structures, 64-bit counters in MIBs, 64-bit sequence numbers and cookies in security, etc. Finally, the number "64" may be considered "magic" in some RFCs, e.g., 64k limits in DNS and Base64 encodings in MIME. None of this has any influence on the length of the IID, but might confuse a careless reader.

4. Possible areas of breakage

This section discusses several specific aspects of IPv6 where we can expect operational breakage with subnet prefixes other than /64.

- o Multicast: [RFC3306] defines a method for generating IPv6 multicast group addresses based on unicast prefixes. This method assumes a longest network prefix of 64 bits. If a longer prefix is used, there is no way to generate a specific multicast group address using this method. In such cases the administrator would need to use an "artificial" prefix from within their allocation (a /64 or shorter) from which to generate the group address. This prefix would not correspond to a real subnet.

Similarly [RFC3956], which specifies Embedded-RP, allowing IPv6 multicast rendezvous point addresses to be embedded in the multicast group address, would also fail, as the scheme assumes a maximum prefix length of 64 bits.

- o CGA: The Cryptographically Generated Address format (CGA, [RFC3972]) is heavily based on a /64 interface identifier. [RFC3972] has defined a detailed algorithm how to generate 64-bit interface identifier from a public key and a 64-bit subnet prefix. Breaking the /64 boundary would certainly break the current CGA definition. However, CGA might benefit in a redefined version if more bits are used for interface identifier (which means shorter prefix length). For now, 59 bits are used for cryptographic purposes. The more bits are available, the stronger CGA could be. Conversely, longer prefixes would weaken CGA.
- o NAT64: Both stateless [RFC6052] NAT64 and stateful NAT64 [RFC6146] are flexible for the prefix length. [RFC6052] has defined multiple address formats for NAT64. In Section 2 "IPv4-Embedded IPv6 Prefix and Format" of [RFC6052], the network-specific prefix could be one of /32, /40, /48, /56, /64 and /96. The remaining part of the IPv6 address is constructed by a 32-bit IPv4 address, a 8-bit u byte and a variable length suffix (there is no u byte and suffix in the case of 96-bit Well-Known Prefix). NAT64 is therefore OK with a boundary out to /96, but not longer.
- o NPTv6: IPv6-to-IPv6 Network Prefix Translation [RFC6296] is also bound to /64 boundary. NPTv6 maps a /64 prefix with other /64 prefix. When the NPTv6 Translator is configured with a /48 or shorter prefix, the 64-bit interface identifier is kept unmodified during translation. However, the /64 boundary might be broken as long as the "inside" and "outside" prefix has the same length.

- o ILNP: Identifier-Locator Network Protocol (ILNP) [RFC6741] is designed around the /64 boundary, since it relies on locally unique 64-bit interface identifiers. While a re-design to use longer prefixes is not inconceivable, this would need major changes to the existing specification for the IPv6 version of ILNP.
- o shim6: The Multihoming Shim Protocol for IPv6 (shim6) [RFC5533] in its insecure form treats IPv6 address as opaque 128-bit objects. However, to secure the protocol against spoofing, it is essential to either use CGAs (see above) or Hash-Based Addresses (HBA) [RFC5535]. Like CGAs, HBAs are generated using a procedure that assumes a 64-bit identifier. Therefore, in effect, secure shim6 is affected by the /64 boundary exactly like CGAs.
- o others?

It goes without saying that if prefixes longer than /64 are to be used, all hosts must be capable of generating IIDs shorter than 64 bits, in order to follow the auto-configuration procedure correctly [RFC4862]. There is however the rather special case of the link-local prefix. While RFC 4862 is careful not to define any specific length of link-local prefix within fe80::/10, operationally there would be a problem unless all hosts on a link use IIDs of the same length to configure a link-local address during reboot. Typically today the choice of 64 bits for the link-local IID length is hard-coded per interface. There might be no way to change this except conceivably by manual configuration, which will be impossible if the host concerned has no local user interface.

5. Experimental observations

5.1. Survey of the processing of Neighbor Discovery options with prefixes other than /64

This section provides a survey of the processing of Neighbor Discovery options which include prefixes that are different than /64.

The behavior of nodes was assessed with respect to the following options:

- o PIO-A: Prefix Information Option (PIO) [RFC4861] with the A bit set.
- o PIO-L: Prefix Information Option (PIO) [RFC4861] with the L bit set.

- o PIO-AL: Prefix Information Option (PIO) [RFC4861] with both the A and L bits set.
- o RIO: Route Information Option (RIO) [RFC4191].

In the tables below, the following notation is used:

NOT-SUP:

This option is not supported (i.e., it is ignored no matter the prefix length used).

LOCAL:

The corresponding prefix is considered "on-link".

ROUTE

The corresponding route is added to the IPv6 routing table.

IGNORE:

The Option is ignored as an error.

Operating System	PIO-A	PIO-L	PIO-AL	RIO
FreeBSD 9.0	IGNORE	LOCAL	LOCAL	NOT-SUP
Linux 3.0.0-15	IGNORE	LOCAL	LOCAL	NOT-SUP
Linux-current	IGNORE	LOCAL	LOCAL	NOT-SUP
NetBSD 5.1	IGNORE	LOCAL	LOCAL	NOT-SUP
OpenBSD-current	IGNORE	LOCAL	LOCAL	NOT-SUP
Win XP SP2	IGNORE	LOCAL	LOCAL	ROUTE
Win 7 Home Premium	IGNORE	LOCAL	LOCAL	ROUTE

Table 1: Processing of ND options with prefixes longer than /64

Operating System	PIO-A	PIO-L	PIO-AL	RIO
FreeBSD 9.0	IGNORE	LOCAL	LOCAL	NOT-SUP
Linux 3.0.0-15	IGNORE	LOCAL	LOCAL	NOT-SUP
Linux-current	IGNORE	LOCAL	LOCAL	NOT-SUP
NetBSD 5.1	IGNORE	LOCAL	LOCAL	NOT-SUP
OpenBSD-current	IGNORE	LOCAL	LOCAL	NOT-SUP
Win XP SP2	IGNORE	LOCAL	LOCAL	ROUTE
Win 7 Home Premium	IGNORE	LOCAL	LOCAL	ROUTE

Table 2: Processing of ND options with prefixes shorter than /64

The results obtained can be summarized as follows:

- o the "A" bit in the Prefix Information Options is honored only if the prefix length is 64.
- o the "L" bit in the Prefix Information Options is honored for any arbitrary prefix length (whether shorter or longer than /64).
- o nodes that support the Route Information Option, allow such routes to be specified with prefixes of any arbitrary length (whether shorter or longer than /64)

5.2. Other Observations

Participants in the V6OPS working group have indicated that some forwarding devices have been shown to work correctly with long prefix masks such as /80 or /96. Indeed, it is to be expected that longest prefix match based forwarding will work for any prefix length, and no reports of this failing have been noted. Also, DHCPv6 is in widespread use without any dependency on the /64 boundary. Reportedly, there are deployments of /120 subnets configured using DHCPv6.

It has been reported that at least one type of switch has a content-addressable memory limited to 144 bits. This means that filters cannot be defined based on 128-bit addresses and two 16-bit port numbers; the longest prefix that could be used in such a filter is /112.

There have been unconfirmed assertions that some routers have a performance drop-off for prefixes longer than /64, due to design issues.

More input with practical observations is welcomed.

6. Privacy issues

The length of the interface identifier has implications for privacy [I-D.ietf-6man-ipv6-address-generation-privacy]. In any case in which the value of the identifier is intended to be hard to guess, whether or not it is cryptographically generated, it is apparent that more bits are better. For example, if there are only 20 bits to be guessed, at most just over a million guesses are needed, today well within the capacity of a low cost attack mechanism. It is hard to state in general how many bits are enough to protect privacy, since this depends on the resources available to the attacker, but it seems clear that a privacy solution needs to resist an attack requiring billions rather than millions of guesses. Trillions would be better, suggesting that at least 40 bits should be available. Thus we can argue that subnet prefixes longer than say /80 might raise privacy concerns by making the IID guessable.

A prefix long enough to limit the number of addresses comparably to an IPv4 subnet, such as /120, would create exactly the same situation for privacy as IPv4. In particular, a host would be forced to pick a new IID when roaming to a new network, to avoid collisions. An argument could be made that since this reduces traceability, it is a good thing from a privacy point of view.

7. Implementation and deployment issues

From an early stage, implementations and deployments of IPv6 assumed the /64 subnet size, even though routing was based on variable-length subnet masks of any length. As shown above, this became anchored in many specifications (Section 3) and in important aspects of implementations commonly used in local area networks (Section 5). In fact, a programmer might be lulled into assuming a comfortable rule of thumb that subnet prefixes are always /64 and an IID is always of length 64. Apart from the limited evidence in Section 5.1, we cannot tell without code inspections or tests whether existing stacks are able to handle a flexible IID length, or whether they would require modification to do so.

The main practical consequence of the existing specifications is that deployments in which longer subnet prefixes are used cannot make use of SLAAC-configured addresses, and require either statically configured addresses or DHCPv6. To reverse this argument, if it was

considered desirable to allow auto-configured addresses with subnet prefixes longer than /64, all of the specifications identified above as depending on /64 would have to be modified, with due regard to interoperability with unmodified stacks. In fact [I-D.ietf-6man-stable-privacy-addresses] allows for this possibility. Then modified stacks would have to be developed and deployed. It might be the case that some stacks contain dependencies on the /64 boundary which are not directly implied by the specifications, and any such hidden dependencies would also need to be found and removed.

Typical IP Address Management (IPAM) tools treat /64 as the default subnet size, but allow users to specify longer subnet prefixes if desired. Clearly, all IPAM tools and network management systems would need to be checked in detail.

Some implementation issues concerning prefix assignment are worth mentioning.

1. It is sometimes suggested that assigning a prefix such as /48 or /56 to every user site (including the smallest) as recommended by [RFC6177] is wasteful. In fact, the currently released unicast address space, 2000::/3, contains 35 trillion /48 prefixes ($(2^{45} = 35,184,372,088,832)$). With 2000::/3 and 0::/3 currently committed, we still have 75% of the address space in reserve. Thus there is no objective risk of prefix depletion by assigning /48 or /56 prefixes. This should be considered when evaluating the scenario of Section 2.1.
2. Some have argued that more prefix bits are needed to allow a hierarchical addressing scheme within a campus or corporate network. However, flat routing is widely and successfully used within rather large networks, with hundreds of routers and thousands of end systems. Therefore there is no objective need for additional prefix bits to support hierarchy and aggregation.
3. Some network operators wish to know and audit which nodes are active on a network, especially those that are allowed to communicate off link or off site. They may also wish to limit the total number of active addresses and sessions that can be sourced from a particular host, LAN or site, in order to prevent potential resource depletion attacks or other problems spreading beyond a certain scope of control. It has been argued that this type of control would be easier if only long network prefixes with relatively small numbers of possible hosts per network were used, reducing the discovery problem.

We now list some practical effects of the fixed /64 boundary.

- o Everything is the same. Compared to IPv4, there is no more calculating (leaf) subnet sizes, no more juggling between subnets, fewer errors.
- o There are always enough addresses in any subnet to add one or more devices. There might be other limits, but addressing will never get in the way.
- o Adding a subnet is easy - just take the next /64. No estimates, calculations, consideration or judgment is needed.
- o Router configurations are easier to understand.
- o Documentation is easier to write and easier to read; training is easier.

8. Conclusion

Summary of pros and cons; risks (write this bit last!)

9. Security Considerations

In addition to the privacy issues mentioned in Section 6, and the issues mentioned with CGAs and HBAs in Section 4, the length of the subnet prefix affects the matter of defence against scanning attacks [I-D.ietf-opsec-ipv6-host-scanning]. Assuming the attacker has discovered or guessed the prefix length, a longer prefix reduces the space that the attacker needs to scan, e.g., to only 256 addresses if the prefix is /120. On the other hand, if the attacker has not discovered the prefix length and assumes it to be /64, routers can trivially discard attack packets that do not fall within an actual subnet.

However, assume that an attacker finds one valid address A and then starts a scanning attack by scanning "outwards" from A, by trying A+1, A-1, A+2, A-2, etc. This attacker will easily find all hosts in any subnet with a long prefix, because they will have addresses close to A. We therefore conclude that any prefix containing densely packed valid addresses is vulnerable to a scanning attack, without the attacker needing to guess the prefix length. Therefore, to preserve IPv6's advantage over IPv4 in resisting scanning attacks, it is important that subnet prefixes are short enough to allow sparse allocation of identifiers within each subnet. The considerations are similar to those for privacy, and we can again argue that prefixes longer than say /80 might significantly increase vulnerability. Ironically, this argument is exactly converse to the argument for longer prefixes to resist an ND cache attack, as described in Section 2.2.

Denial of service attacks related to Neighbor Discovery are discussed in [RFC6583]. One of the mitigations suggested by that document is "sizing subnets to reflect the number of addresses actually in use", but the fact that this greatly simplifies scanning attacks is not noted. For further discussion of scanning attacks, see [I-D.ietf-opsec-ipv6-host-scanning].

Note that, although not known at the time of writing, there might be other resource exhaustion attacks available, similar in nature to the ND cache attack. We cannot exclude that such attacks might be exacerbated by sparsely populated subnets such as a /64. It should also be noted that this analysis assumes a conventional deployment model with a significant number of end-systems located in a single LAN broadcast domain. Other deployment models might lead to different conclusions.

10. IANA Considerations

This document requests no action by IANA.

11. Acknowledgements

This document was inspired by a vigorous discussion on the V6OPS working group mailing list with at least 20 participants. Later, valuable comments were received from Lorenzo Colitti, David Farmer, Ray Hunter, Mark Smith, Fred Templin, Stig Venaas, and other participants in the IETF.

This document was produced using the xml2rfc tool [RFC2629].

12. Change log [RFC Editor: Please remove]

draft-carpenter-6man-why64-01: WG comments, added experimental results, implementation/deployment text, 2014-02-06.

draft-carpenter-6man-why64-00: original version, 2014-01-06.

13. References

13.1. Normative References

[I-D.ietf-6man-ug]

Carpenter, B. and S. Jiang, "Significance of IPv6 Interface Identifiers", draft-ietf-6man-ug-06 (work in progress), December 2013.

- [I-D.ietf-opsec-ipv6-host-scanning]
Gont, F. and T. Chown, "Network Reconnaissance in IPv6 Networks", draft-ietf-opsec-ipv6-host-scanning-03 (work in progress), January 2014.
- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.
- [RFC2467] Crawford, M., "Transmission of IPv6 Packets over FDDI Networks", RFC 2467, December 1998.
- [RFC2470] Crawford, M., Narten, T., and S. Thomas, "Transmission of IPv6 Packets over Token Ring Networks", RFC 2470, December 1998.
- [RFC2492] Armitage, G., Schulter, P., and M. Jork, "IPv6 over ATM Networks", RFC 2492, January 1999.
- [RFC2497] Souvatzis, I., "Transmission of IPv6 Packets over ARCnet Networks", RFC 2497, January 1999.
- [RFC2526] Johnson, D. and S. Deering, "Reserved IPv6 Subnet Anycast Addresses", RFC 2526, March 1999.
- [RFC2529] Carpenter, B. and C. Jung, "Transmission of IPv6 over IPv4 Domains without Explicit Tunnels", RFC 2529, March 1999.
- [RFC2590] Conta, A., Malis, A., and M. Mueller, "Transmission of IPv6 Packets over Frame Relay Networks Specification", RFC 2590, May 1999.
- [RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.
- [RFC3146] Fujisawa, K. and A. Onoe, "Transmission of IPv6 Packets over IEEE 1394 Networks", RFC 3146, October 2001.
- [RFC3306] Haberman, B. and D. Thaler, "Unicast-Prefix-based IPv6 Multicast Addresses", RFC 3306, August 2002.
- [RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.
- [RFC3956] Savola, P. and B. Haberman, "Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address", RFC 3956, November 2004.

- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.
- [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.
- [RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, October 2005.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4338] DeSanti, C., Carlson, C., and R. Nixon, "Transmission of IPv6, IPv4, and Address Resolution Protocol (ARP) Packets over Fibre Channel", RFC 4338, January 2006.
- [RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.
- [RFC4429] Moore, N., "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, April 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.
- [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.
- [RFC5072] Varada, S., Haskins, D., and E. Allen, "IP Version 6 over PPP", RFC 5072, September 2007.
- [RFC5121] Patil, B., Xia, F., Sarikaya, B., Choi, JH., and S. Madanapalli, "Transmission of IPv6 via the IPv6 Convergence Sublayer over IEEE 802.16 Networks", RFC 5121, February 2008.

- [RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, March 2008.
- [RFC5453] Krishnan, S., "Reserved IPv6 Interface Identifiers", RFC 5453, February 2009.
- [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, June 2009.
- [RFC5535] Bagnulo, M., "Hash-Based Addresses (HBA)", RFC 5535, June 2009.
- [RFC5692] Jeon, H., Jeong, S., and M. Riegel, "Transmission of IP over Ethernet over IEEE 802.16 Networks", RFC 5692, October 2009.
- [RFC5942] Singh, H., Beebe, W., and E. Nordmark, "IPv6 Subnet Model: The Relationship between Links and Subnet Prefixes", RFC 5942, July 2010.
- [RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.
- [RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.
- [RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.
- [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, April 2011.
- [RFC6177] Narten, T., Huston, G., and L. Roberts, "IPv6 Address Assignment to End Sites", BCP 157, RFC 6177, March 2011.
- [RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.

- [RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, November 2011.
- [RFC6741] Atkinson,, RJ., "Identifier-Locator Network Protocol (ILNP) Engineering Considerations", RFC 6741, November 2012.
- [RFC7084] Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, November 2013.

13.2. Informative References

- [DRAFT-odell]
O'Dell, M., "8+8 - An Alternate Addressing Architecture for IPv6", draft-odell-8+8.00 (work in progress), October 1996.
- [I-D.brandt-6man-lowpanz]
Brandt, A. and J. Buron, "Transmission of IPv6 packets over ITU-T G.9959 Networks", draft-brandt-6man-lowpanz-02 (work in progress), June 2013.
- [I-D.ietf-6lowpan-btle]
Nieminen, J., Savolainen, T., Isomaki, M., Patil, B., Shelby, Z., and C. Gomez, "Transmission of IPv6 Packets over BLUETOOTH Low Energy", draft-ietf-6lowpan-btle-12 (work in progress), February 2013.
- [I-D.ietf-6man-6lobac]
Lynn, K., Martocci, J., Neilson, C., and S. Donaldson, "Transmission of IPv6 over MS/TP Networks", draft-ietf-6man-6lobac-01 (work in progress), March 2012.
- [I-D.ietf-6man-ipv6-address-generation-privacy]
Cooper, A., Gont, F., and D. Thaler, "Privacy Considerations for IPv6 Address Generation Mechanisms", draft-ietf-6man-ipv6-address-generation-privacy-00 (work in progress), October 2013.
- [I-D.ietf-6man-stable-privacy-addresses]
Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", draft-ietf-6man-stable-privacy-addresses-16 (work in progress), December 2013.

- [I-D.ietf-homenet-arch]
Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil,
"IPv6 Home Networking Architecture Principles", draft-
ietf-homenet-arch-11 (work in progress), October 2013.
- [I-D.ietf-v6ops-64share]
Byrne, C., Drown, D., and V. Ales, "Extending an IPv6 /64
Prefix from a 3GPP Mobile Interface to a LAN link", draft-
ietf-v6ops-64share-09 (work in progress), October 2013.
- [I-D.templin-aerolink]
Templin, F., "Transmission of IPv6 Packets over AERO
Links", draft-templin-aerolink-01 (work in progress),
January 2014.
- [IEEE802] IEEE, "IEEE Standard for Local and Metropolitan Area
Networks: Overview and Architecture", IEEE Std 802-2001
(R2007), 2007.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629,
June 1999.
- [RFC6583] Gashinsky, I., Jaeggli, J., and W. Kumari, "Operational
Neighbor Discovery Problems", RFC 6583, March 2012.
- [RFC6741] Atkinson, R.J., "Identifier-Locator Network Protocol
(ILNP) Engineering Considerations", RFC 6741, November
2012.
- [RFC6877] Mawatari, M., Kawashima, M., and C. Byrne, "464XLAT:
Combination of Stateful and Stateless Translation", RFC
6877, April 2013.

Authors' Addresses

Brian Carpenter (editor)
Department of Computer Science
University of Auckland
PB 92019
Auckland 1142
New Zealand

Email: brian.e.carpenter@gmail.com

Tim Chown
University of Southampton
Southampton, Hampshire SO17 1BJ
United Kingdom

Email: tjc@ecs.soton.ac.uk

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

Email: fgont@si6networks.com

Sheng Jiang
Huawei Technologies Co., Ltd
Q14, Huawei Campus
No.156 Beijing Road
Hai-Dian District, Beijing 100095
P.R. China

Email: jiangsheng@huawei.com

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

Email: Alexandru.Petrescu@cea.fr

Andrew Yourtchenko
cisco
7a de Kleetlaan
Diegem 1830
Belgium

Email: ayourtch@cisco.com

6man WG
Internet-Draft
Updates: 4861 (if approved)
Intended status: Standards Track
Expires: August 31, 2015

S. Chakrabarti
Ericsson
E. Nordmark
Arista Networks
P. Thubert
Cisco Systems
M. Wasserman
Painless Security
February 27, 2015

IPv6 Neighbor Discovery Optimizations for Wired and Wireless Networks
draft-chakrabarti-nordmark-6man-efficient-nd-07

Abstract

IPv6 Neighbor Discovery (RFC 4861 going back to RFC 1970) was defined at a time when link-local multicast was as reliable and with the same network cost (send a packet on a yellow-coax Ethernet) as unicast and where the hosts were assumed to be always on and connected.

Since 1996 we've seen a significant change with both an introduction of wireless networks and battery operated devices, which poses significant challenges for the old assumptions. We are also seeing datacenter networks where virtual machines are not always on and connected, and scaling of multicast can be challenging.

This specification contains extensions to IPv6 Neighbor Discovery which remove most use of multicast and make sleeping hosts more efficient. The specification includes a default mixed mode where a link can have an arbitrary mix of hosts and/or routers - some implementing legacy Neighbor Discovery and some implementing the optimizations in this specification. The optimizations provide incremental benefits to hosts as soon as the first updated routers are deployed on a link.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 31, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	5
2. Goals and Requirements	6
2.1. Mixed-Mode Operations	7
3. Changes to ND state management	7
4. Definition Of Terms	8
5. Protocol Overview	9
5.1. Proxying to handle Mixed mode	11
6. New Neighbor Discovery Options and Messages	11
6.1. Router Advertisement flag for NEARs	11
6.2. Address Registration Option (ARO)	12
6.3. Registrar Address Option (RAO)	14
7. Conceptual Data Structures	15
8. Host Behavior	16
8.1. Host and/or Interface Initialization	16
8.2. Host Receiving Router Advertisements	16
8.3. Timing out Registrar List entries	17
8.4. Sending AROs	17
8.5. Receiving Neighbor Advertisements	18
8.6. Host Management of the TID	18
8.7. Refreshing a Registration	18
8.8. De-registering	19
8.9. Refreshing RA information	19
8.10. Sleep and Wakeup	21
8.11. Receiving Redirects	21
8.12. Movement Detection	21
9. Router Behavior	21
9.1. Router and/or Interface Initialization	22
9.2. Receiving Router Solicitations	22
9.3. Periodic Multicast RA for legacy hosts	23
9.4. Multicast RA when new information	23
9.5. Receiving ARO	23
9.6. NCE Management in NEARs	23
9.7. Sending non-zero status in ARO	24
9.8. Timing out Registered NCEs	24
9.9. Router Advertisement Consistency	25
9.10. Sending Redirects	25
9.11. Providing Information to Routing Protocols	25
9.12. Creating Legacy NCEs	25
9.13. Proxy Address Resolution and DAD for Legacy Hosts	25
10. Handling ND DoS Attack	26
11. Bootstrapping	27
12. Interaction with other protocols	28
12.1. Detecting Network Attachment (DNA)	28
12.2. DHCPv6 Interaction	28
12.3. Other use of Multicast	29
12.4. VRRP Interaction	29

13. Updated Neighbor Discovery Constants	29
14. Security Considerations	30
15. IANA Considerations	30
16. Changelog	30
17. Acknowledgements	31
18. Open Issues	32
19. References	33
19.1. Normative References	33
19.2. Informative References	33
Authors' Addresses	35

1. Introduction

IPv6 Neighbor Discovery [RFC4861] was defined at a time when local area networks had different properties than today. A common link was the yellow-coax shared wire Ethernet, where a link-layer multicast and unicast worked the same - send the packet on the wire and the interested receivers will pick it up. Thus the network cost (ignoring any processing cost on the receivers that might not completely filter out Ethernet multicast addresses that they did not want) and the reliability of sending a link-layer unicast and multicast was the same. Furthermore, the hosts at the time was always on and connected. Powering on and off the workstation/PC hosts at the time was slow and disruptive process.

Under the above assumptions it was quite efficient to maintain the shared state of the link such as the prefixes and their lifetimes using periodic multicast Router Advertisement messages. It was also efficient to use multicast Neighbor Solicitations for address resolution as a slight improvement over the broadcast use in ARP. And finally, checking for a potential duplicate IPv6 address using broadcast was efficient and believed to be likely to be robust.

Since then we've seen a tremendous change with the wide-spread deployment of WiFi and other wireless network technologies. WiFi is a case in point in that it provides the same network service abstraction as Ethernet and is often bridged with Ethernets, meaning that the Neighbor Discovery software on hosts and routers might be unaware that WiFi is being used. Yet the performance and reliability of multicast is quite different than for unicast on WiFi (see for instance [I-D.vyncke-6man-mcast-not-efficient]). Similarly 3GPP and M2M networks and devices will benefit from a standard specification for optimized Neighbor discovery. Even wired networks have evolved substantially with modern switch fabrics using explicit packet replication logic to handle multicast packets.

The assumptions about the reliability of a single multicast message for duplicate address detection has also shown to be not correct, due to a set of issues listed in [I-D.yourtchenko-6man-dad-issues].

The hosts and usage patterns has undergone radical changes as well. Hosts go to sleep when not in use to save on battery power [RFC6574] or to be more energy efficient even with mains power. The expectation is that waking up doesn't take much time and power otherwise the benefits of sleeping are greatly reduced. Initially sleeping hosts were esoteric sensor nodes, but this sleeping hosts have become mainstream in smartphones.

Some of the above trends were observed early (e.g., Ohta-san

commented on Neighbor Discovery being inefficient on WiFi a long time back) but the issues were not broadly understood. The issues were raised in the 6LowPAN context where the desire was to run IPv6 over low-power radio networks and battery operated devices. That lead to defining a set of optimizations [RFC6775] for that specific category of links. However, the trends are not limited to such specific link types.

This document applies what we have learned from 6LowPAN to all link types. That includes reusing existing support from base Neighbor Discovery (such as Redirect messages) and reusing from 6LowPAN-ND (Address Registration Option). There are additions above and beyond that to improve the robustness with redundant routers and to support mixed mode.

The optimizations are done in a way to provide incremental benefits. As soon as there is at least one router on a link which supports these optimizations, then the updated hosts on the link can sleep better, while co-existing on the same link with unmodified hosts.

2. Goals and Requirements

The goal is to remove as much Neighbor Discovery multicast traffic on the link as possible, and handle Duplicate Address Detection (DAD) without requiring the hosts to always be awake. While not an explicit goal, it turned out that the issues in [I-D.yourtchenko-6man-dad-issues] that are about robustness/correctness are also addressed as a side effect of supporting sleepy hosts.

The optimization will be highly effective for links and nodes which do not support multicast and for multicast networks without MLD snooping switches. Moreover, in the MLD-snooping networks the MLD switches will deal with less number of multicasts.

The requirements handle are:

Remove the use of multicast for DAD and Address Resolution (no multicast NS messages), and remove periodic multicast RAs. Some multicast RS and RA are needed to handle the arrival of new hosts and routers on the link since they need to bootstrap to find each other.

Remove the need for hosts to always be awake to defend their addresses by responding to any DAD probes.

Ensure that the protocol is robust against single points of failure and uses soft state which is automatically rebuilt after a state loss.

A router which does not support legacy hosts will always maintain a complete set of Neighbor Cache Entries (NCEs) for all hosts on the link. Hence there is no need for it to send Neighbor Solicitations. Thus it can avoid the problem specified in [RFC6583].

The optimized solution SHOULD be independent of the addresses allocation mechanism. In addition to supporting SLAAC [RFC4862] and DHCPv6 [RFC3315] it SHOULD also work with hosts with 'Privacy Extensions for Stateless Address Autoconfiguration in IPv6' [RFC4941] and with stable IPv6 private addresses [I-D.ietf-6man-stable-privacy-addresses] thus it handles the recommendations in [I-D.ietf-6man-default-iids].

2.1. Mixed-Mode Operations

Mixed-Mode operation is the protocol behavior when the IPv6 subnet is composed of legacy IPv6 Neighbor Discovery compliant nodes and efficiency-aware IPv6 nodes implementing this specification.

The mixed-mode model SHOULD support arbitrary combinations of legacy [RFC4861] hosts and/or routers with new hosts and/or routers on a link. The introduction of one new router SHOULD provide improved services to a new host, allowing the new host to avoid multicast and not requiring the host to be awake to defend its IPv6 addresses using DAD.

This document assumes that an implementation will have configuration knobs to determine whether it is running in legacy IPv6 ND [RFC4861] or Efficiency Aware only mode (no-legacy mode) or both (Mixed mode).

3. Changes to ND state management

These optimizations change some fundamental aspects of Neighbor Discovery. Firstly, it moves the distributed address resolution state (each node responding to a multicast NS for its own addresses) to a set of routers which maintain a list of Address Registrations for the hosts. Secondly, the information distributed in Router Advertisements changes from being periodically flooded by the routers to explicit requests from the hosts for refreshed information using unicast Router Solicitations.

4. Definition Of Terms

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

IPv6 ND-efficiency-aware Router (NEAR):

A router that implements the optimizations specified in this document. This router should be able to handle both legacy IPv6 nodes and nodes that sends registration request.

Efficiency-Aware Host (EAH):

The EAH is the host which implements the host functionality for optimized Neighbor Discovery mentioned in this document. At least initially implementations are likely to have a fallback to legacy Neighbor Discovery when no NEAR is on the link.

Legacy IPv6 Host:

A IPv6 host that implements [RFC4861] without the extensions in this document.

Legacy IPv6 Router:

A IPv6 router that implements [RFC4861] without the extensions in this document.

Mixed mode

A NEAR supports both legacy hosts and EAH, with a configuration knob to disable the support for legacy hosts. Some routers on the link can be legacy and some can be NEAR.

No-legacy mode

A mode configured on a NEAR to not support any legacy [RFC4861] hosts or routers. Opposite of mixed mode.

IPv6 Address Registrar

Normally the default router(s) on the link will act as IPv6 Address Registrars tracking all the EAHs. But in some cases it is more efficient to use a different set of routers as Address Registrars. The hosts are informed of the address registrars using router advertisement messages, and register with the available registrars.

EUI-64:

It is the IEEE defined 64-bit extended unique identifier formed by concatenation of 24-bit or 36-bit company id value by IEEE Registration Authority and the extension identifier within that company-id assignment. The extension identifiers are 40-bit (for 24-bit company-id) or 28-bit (for the 36-bit company-id)

respectively. The protocol supports EUI-64 for compatibility with [RFC6775].

DUID

It is a DHCP Unique ID of a device [RFC3315]. The DUID is assumed to be stable in a given IPv6 subnet. A device which does not have an EUI-64 can form and use a DUID in its address registrations.

NCE

Neighbor Cache Entry. It is a conceptual data structure introduced in section 5.1 of [RFC4861] and further elaborated in [RFC6775].

TID

The Transaction ID is a device generated sequence number used for registration. This number is used to allow a host to have concurrent registrations with different routers, while also being able to robustly replace a registration with a new one. It facilitates interoperability with protocols like RPL [RFC6550] which use a TID internally to handle host movement.

5. Protocol Overview

In a nutshell, the following basic optimizations are made from the original IPv6 Neighbor Discovery protocol [RFC4861]:

- o Adds Node Registration with the default router(s).
- o Address Resolution and DAD uses the registered addresses instead of multicast Neighbor Solicitation messages for non-link-local IPv6 addresses.
- o Does not require unsolicited periodic Router Advertisements.
- o Supports mixed-mode operation where legacy IPv6 hosts and routers and NEARs and EAHs can co-exist on the same link. This support can be configured off.

When a host powers on it behaves conforms to legacy ND [RFC4861] by multicasting up to MAX_RTR_SOLICITATIONS Router Solicitations and receives Router Advertisements. The additional information in the Router Advertisements by the NEARs is used by the EAH to build a list of IPv6 Address Registrars. As the host is forming its IPv6 addresses (using any of the many stateless and stateful IPv6 address allocation mechanism) then, instead of using a multicast DAD message, it unicasts an Neighbor Solicitation with the new Address Registration Option (ARO) to the Registrars. Assuming an IPv6

addresses are not duplicate the EAH receives a Neighbor Advertisement with the ARO option from the NEARs. The EAH refreshes the registered addresses before they expire, thereby removing the need for the EAH to be awake to defend its addresses using DAD as specified in [RFC4862] as the NEARs will handle DAD.

The routers on the link advertise the prefixes without setting the L flag. Thus only the IPv6 link-local addresses are considered on-link. Thus the hosts will send packets to a default router, and the default routers maintain all the registrations. Hence a router will know the link-layer addresses of all the registered hosts. This enables the router to forward the packet (without needing any Address Resolution with the multicast Neighbor Solicitation), and also to send a Redirect to the originating host informing the host of the link-layer address.

Instead of relying on periodic multicast RAs to refresh the lifetimes of prefixes etc., the hosts ask for refreshed information by unicasting a Router Solicitation before the information expires. Note that [I-D.nordmark-6man-rs-refresh] make that behavior more explicit by having the routers advertise a timeout.

The periodic multicast RAs may be used to provide new information such as additional prefixes, radical reduction in the preferred and/or valid lifetime for a prefix. A host does not know to ask for such information. Thus a router that wishes to quickly disseminate such change can resort to a few multicast RAs, or wait for the hosts to request a refresh using a Router Solicitation.

The routers can crash and loose all their state, including the Address Registrations. On router initialization the router will multicast a few initial RAs. The protocol has a Router Epoch mechanism which is used by the hosts to detect that the router has lost state. In that case the hosts will immediately re-register allowing the router to quickly rebuild its Address Registration state.

Normally it is sufficient for the hosts to register with all the default routers on the link. However, in some cases such as simplistic VRRP deployment the hosts should register with all the VRRP routers even though they only know of one virtual router IPv6 address. And in other cases it would be more efficient to only register with one router even though there are multiple default routers. The RAs can contain a Registrar Address Option to explicitly tell the hosts where to register.

Sleepy hosts are supported by this Neighbor Discovery procedures as they are not woken up periodically by Router Advertisement multicast

messages or Neighbor Solicitation multicast messages. Sleepy nodes may wake up in its own schedule and send unicast registration refresh messages before the registration lifetime expiration. The recommended procedure on wakeup is to unicast a Neighbor Solicitation to the default router(s), which serves as DNA check [RFC6059] that the host is on the same link, performs NUD against the router, and includes the Address Registration Option to refresh the registration.

5.1. Proxying to handle Mixed mode

When there are one or more legacy routers on the link then the recommendation is to configure those to advertise the prefixes with L=0 just as the NEARs. That results in the hosts sending all packets to a default router unless/until they receive a Redirect. However, the legacy routers do not maintain the address registrations. Thus even though all the hosts send the packets to the routers, the legacy routers might in turn need to perform Address Resolution by multicasting a Neighbor Solicitation per [RFC4861]. In addition, legacy hosts and legacy routers will perform DAD as specified in [RFC4862] that is, by sending a multicast NS and waiting for a NS or NA which indicates a conflict. A EAH will not receive and respond to such messages.

If the NEARs have been configured to operate in mixed-mode, then they will respond to multicast NS messages from legacy nodes for both DAD and Address Resolution. They will respond with the Target Link Layer Address being that of the registered host, so that subsequent communication will not go via the routers. (Implementations of "Neighbor Discovery Proxies (ND Proxy)" [RFC4389] might proxy using their own MAC address as TLLA, but that is outside of the scope of this document.)

6. New Neighbor Discovery Options and Messages

This specification introduces a new flag in the RAs, reuses and extends the ARO option from [RFC6775] and introduces a new Registrar Address option.

6.1. Router Advertisement flag for NEARs

A new Router Advertisement flag is needed in order to distinguish a router advertisement sent by a NEAR, which will trigger an EAH to register with the NEAR. This flag is ignored by the legacy IPv6 hosts.

The current flags field in RA is reproduced here with the added 'E' bit.

```

0 1 2 3 4 5 6 7
+---+---+---+---+
|M|O|H|Prf|P|E|R|
+---+---+---+---+

```

The 'E' bit is set to 1 in a RA sent by a NEAR. In all other cases the E bit MUST be 0.

6.2. Address Registration Option (ARO)

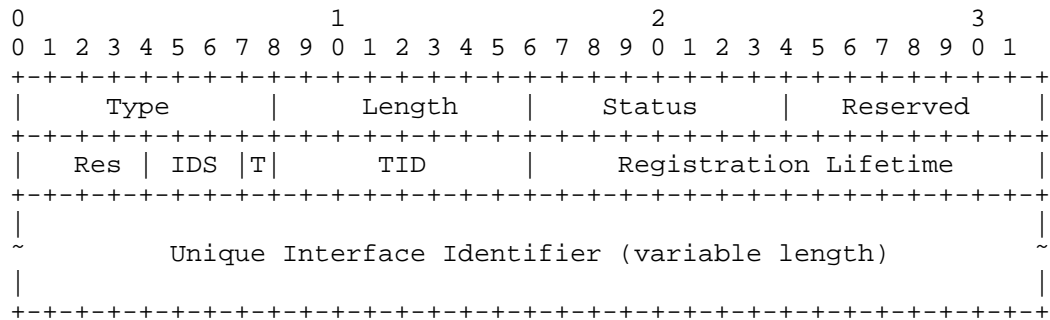
The Address Registration Option was defined in [RFC6775] for the purposes of 6LoWPAN and this document extends it in a backwards compatible way by using some of the reserved fields. The extensions are to handle different unique identifiers than EUI-64 (this document specifies how to use DHCP Unique Identifiers with the ability to use other identifier namespaces in the future) and a Transaction Id.

The Unique Interface Identifier is used by the NEARs to distinguish between a refresh of an existing registration and a different host trying to register an IPv6 address which is already registered by some other host. Thus the requirement is that the unique id is unique per link, but due to the potential for host mobility across links and subnets it should be globally unique. Both an EUI-64 and a DUID satisfies that requirement.

The TID is used by the NEARs to handle the case when due to packet loss one NEAR might have a old registration and another NEAR has a newer registration. The TID allows them to determine which is more recent. The TID also facilitates the interaction with RPL [RFC6550].

An Address Registration Option can be included in unicast Neighbor Solicitation (NS) messages sent by hosts. Thus it can be included in the unicast NS messages that a host sends as part of Neighbor Unreachability Detection to determine that it can still reach the default router(s). The ARO is used by the receiving router to reliably maintain its Neighbor Cache. The same option is included in corresponding Neighbor Advertisement (NA) messages with a Status field indicating the success or failure of the registration.

When the ARO is sent by a host then, for links which have link-layer addresses, a SLLA option MUST be included. The address that is registered is the IPv6 source address of the Neighbor Solicitation message. Typically a host would have several addresses to register, with each one being registered using a separate NS containing an ARO. (This approach facilitates applying SeND [RFC3971].)



Fields:

- Type: 33 [RFC6775]
- Length: 8-bit unsigned integer. The length of the option (including the type and length fields) in units of 8 bytes. The value 0 is invalid.
- Status: 8-bit unsigned integer. Indicates the status of a registration in the NA response. MUST be set to 0 in NS messages. See [RFC6775].
- Reserved: 8 bits. This field is unused. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.
- Res: 4 bits. This field is unused. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.
- IDS: 3 bits. Identifier name Space. Indicates whether the Unique Interface Identifier is a DUID or or a IEEE assigned EUI-64 with room to define additional name spaces.
- T bit: One bit flag. Set if the TID octet is valid.
- TID: 8-bit integer. It is a transaction id maintained by the host and used by the NEARs to determine the most recent registration.
- Registration Lifetime: 16-bit unsigned integer. The amount of time in a unit of 60 seconds that the router should retain the Neighbor Cache entry for the sender of the NS that includes this option. A value of zero means to remove

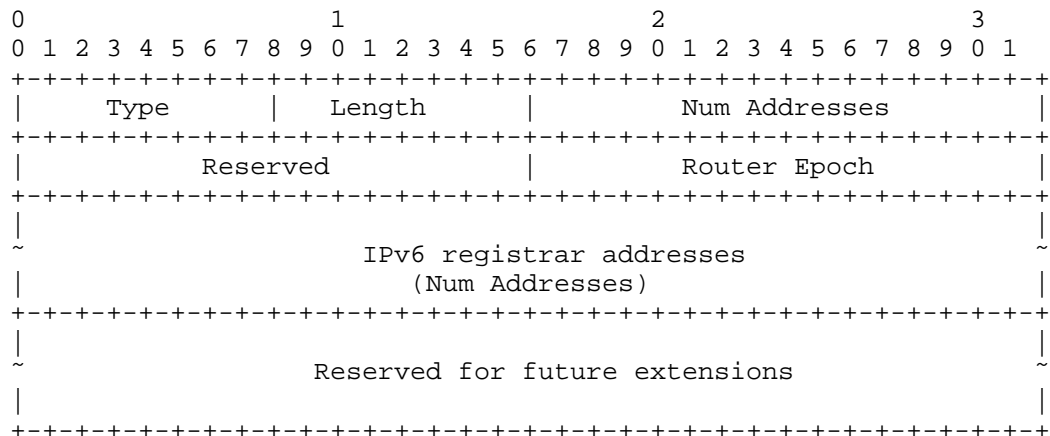
the registration.

Unique Interface Identifier: Variable length in multiples of 8 bytes. If the IDS=000, then it is an 8 byte (64 bit) unmodified EUI-64. If IDS=0011 then it is a variable length DUID. A DUID MUST be zero padded to a multiple of 8 bytes.

When a node includes ARO option in a Neighbor Solicitation it MUST, on links that have link-layer addresses, also include a SLLA option. That option is needed so that the registrar can record the link-layer address on success and send back an error if the address is a duplicate.

6.3. Registrar Address Option (RAO)

Normally the Registrars are the Default Routers. However, there are cases (like some approaches to handle VRRP, or coordinated separate routers) where there is a need to have different (and either more or less) Registrars than Default Routers. Furthermore, to robustly handle NEAR state state loss this option carries a Router Epoch which triggers the EAHs to re-register on a router epoch change. The RAO contains the information for both of those.



Fields:

Type: TBD (IANA)

Length: 8-bit unsigned integer. The length of the option (including the type and length fields) in units of 8 bytes. The value 0 is invalid.

Num Addresses	16-bit unsigned integer. Set to zero if there are no addresses i.e., when the option is used to only carry the router epoch. NumAdressses*2 + 1 must not exceed the Length.
Reserved	16-bit unused field. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.
Router epoch	16-bit integer. A router MUST pick a new epoch after a state loss, either by keeping the epoch in stable storage and incrementing it, or picking a good random number.
IPv6 registrar addresses	Zero or more IPv6 addresses, typically of link-local scope.

The receiver MUST silently ignore any data after the IPv6 registrar addresses field (such data is present when the Length is greater than NumAdressses*2 + 1).

The Registrar Addresses are subject to the same lifetime as the Default Router Lifetime (thus there is no explicit lifetime field in the RAO).

7. Conceptual Data Structures

In addition to the Conceptual Data structures in [RFC4861] a EAH needs to maintain the new Registrar List for each interface. The Registrar List contains the set of IP addresses to which the host needs to send Address Registrations. Each IP address has a Router Epoch (used to determine when a router might have lost state). Also, the host MAY use this data structure to track when it needs to refresh its registrations with the registrar.

The use of explicit registrations with lifetimes plus the desire to not multicast Neighbor Solicitation messages for hosts imply that we manage the Neighbor Cache entries slightly differently than in [RFC4861]. This results in two different types of NCEs and the types specify how those entries can be removed:

Legacy:	Entries that are subject to the normal rules in [RFC4861] that allow for garbage collection when low on memory. Legacy entries are created only when there is no duplicate NCE. The legacy entries are converted to the registered entries upon successful processing of ARO. Legacy type can be considered as union of
---------	---

garbage-collectible and Tentative Type NCEs described in [RFC6775].

Registered: Entries that have an explicit registered lifetime and are kept until this lifetime expires or they are explicitly unregistered.

Note that the type of the NCE is orthogonal to the states specified in [RFC4861]. There can only be one type of NCE for an IP address at a time.

8. Host Behavior

A EAH follows [RFC4861] and applicable parts of [RFC6775] as specified in this section./

A EAH implementation MAY be able to fall back to [RFC4861] host behavior if there is no NEAR on the link.

8.1. Host and/or Interface Initialization

A host multicasts Router Solicitation at system startup or interface initialization as specified in [RFC4861] and its updates such as [I-D.ietf-6man-resilient-rs]. If the interface initialization is due to potential host movement or a wakeup from sleep then the host initially sends a unicast Neighbor Solicitation to the default router(s).

Unlike RFC4861 the RS MUST (on link layers which have addresses) include a SLLA option, which is used by the router to unicast the RA.

The host is not required to join the solicited-node multicast address(es) but it MUST join the all-nodes multicast address.

8.2. Host Receiving Router Advertisements

The RA is validated and processed as specified in [RFC4861] with additional behavior for RAO and the Registrar List as follows.

When a RA is received without a RAO (but with the E flag set), or the RAO contains no Registrar Addresses, then the IPv6 source address is added/updated in the Registrar List. When a RA is received with a RAO then the IPv6 Registrar Addresses in that option are added/updated in the data structure.

If those Registrar List (or entries) already exist and the Router Epoch in the RAO differs from the Router Epoch in the Registrar List

entry, or if the entry does not exist, then the host will initiate sending NS messages with ARO options to the new/updated Registration List entries. Note that if the RA contains no RAO (but the E flag is set) then for the purposes of the epoch comparison one should use a zero Router Epoch.

However, if the Default Router Lifetime in the RA is zero, then any matching Registration List entry (or entries) are instead deleted from the Registration List. It is OPTIONAL for the host to de-register when an entry is deleted from the Registration List.

The host can form its IPv6 address using any available mechanism - SLAAC, DHCPv6, temporary addresses, etc - as the registration mechanism is orthogonal and independent of the address allocation. The Address Registration procedure replaces the DAD procedure in [RFC4862].

8.3. Timing out Registrar List entries

The lifetime for the Registrar List entries are taken from the Default Router Lifetime in the RA. When an entry is removed the host MAY send AROs with a zero Registration Lifetime to the removed Registrar Addresses.

8.4. Sending AROs

When a host has formed a new IPv6 address, or when the host learns of a new NEAR and has existing IPv6 addresses, then it would register the new things (could be new addresses to all the existing Registrars, or the all the IPv6 addresses with the new Registrar. IPv6 link-local addresses are registered as well as the global addresses and ULAs.

If the EAH has a TID then it sets the T-bit and includes the TID in the ARO. When the host registers its addresses with multiple Registrars it uses the same TID. However, if the host has moved (lost its network attachment and determines it is attached to a different link using e.g., DNA [RFC6059]), then it will increment the TID value and use the new value for subsequent registrations.

The host places its Unique Interface Identifier (whether it is a DUID or EUI-64) in the ARO. This identifier is typically kept in stable storage so that the host can use the same identifier over time. It MUST use the same identifier when it re-registers its address, since otherwise all those will be returned as duplicates.

The NS which includes the ARO option MUST include a SLLA option on link layers that have link-layer addresses.

The EAH retransmits NS messages with ARO as specified in [RFC6775] until it receives a NA message from the Registrar containing an ARO. The number of such retransmissions SHOULD be configurable.

8.5. Receiving Neighbor Advertisements

The Neighbor Advertisement are validated and processed as specified in [RFC4861] for example to handle Neighbor Unreachability Detection (NUD). In addition, the host processes any received ARO as follows.

If the ARO has status code 0 (Success), then the host updates the information in the Registrar List to know when it next needs to refresh the registered address with this Registrar. No further processing is needed of the ARO.

If the ARO has status code 1 (Duplicate), then the host can not use the IPv6 address. The host follows the address allocation protocol it used to pick a new address - be that DHCPv6, temporary addresses, etc.

If the ARO has a status code 2 (Neighbor Cache Full) in response to its registration request from a Registrar, then the node SHOULD continue to register the address with other Registrars (when available).

TBD: What about other not yet defined status code values?

8.6. Host Management of the TID

It is RECOMMENDED that the EAH MAY maintain a sequence counter (TID) in stable storage according to section 7 of [RFC6550]. The TID is used to resolve conflicts between different registrations issues by the same host for the same IPv6 address. Conflicts can be due to different link-layer addresses, but it can also be due to registering with different NEARs/Registrars and those routers connect use protocols like RPL for routing, and RPL uses a TID to handle movement vs. multi-homing.

Thus the same TID should be used if the host is registering its addresses with multiple Registrars at the same time. But if the host might have moved to a different attachment point, then it should increment its TID for subsequent registrations.

8.7. Refreshing a Registration

A host SHOULD send a Registration message in order to renew its registration before its registration lifetime expires in order to continue its connectivity with the network.

This specification does not constrain the implementation. One possible implementation strategy is to attempt re-register at 1/3rd of the registration lifetime, and if no response try again at 2/3rd of the lifetime, etc. Another possible strategy is to wait until the end of the registration lifetime and then do the same relatively rapid retransmissions as for the initial registration [RFC6775]. In all cases the host SHOULD apply a random factor to its re-registration timeout to avoid self-synchronizing behavior across lots of hosts. Sleeping hosts would re-register when they are waking up to do other work.

8.8. De-registering

If anytime, the node decides that it does not need a particular default router's service anymore, then it SHOULD send a de-registration message to that NEAR/Registrar. Similarly if the host stops using a particular IPv6 address, then it SHOULD de-register that address with all the Registrars it had registered with. This applies even if the host was using the IPv6 address, then went to sleep, and then picked a different set of IPv6 addresses. In such a case it is preferred if the host de-registers before going to sleep. A mobile host SHOULD first de-register its addresses before it moves away from the subnet (if the mobile host can know that in advance of moving.)

De-registration is performed by unicasting a Neighbor Solicitation with an ARO where the Registration Lifetime is set to zero. Such an ARO should have an incremented TID. De-registration AROs are retransmitted just like other AROs as specified above.

8.9. Refreshing RA information

The EAH is responsible for asking the routers for updates to the information contained in the Router Advertisements, since the NEARs will not multicast such updates. That is done by sending unicast RSs to the router(s) which will result in unicast RAs. However, significant care is required in determining when the RSs should be transmitted.

As part of normal operation the Default Routers, Prefixes, and other RA information have lifetimes, and there are a few common cases:

1. The advertised lifetimes are constant i.e., the routers keep on advancing the time when the information will expire.
2. The routers decrement the advertised lifetimes in real time i.e., the information is set to expire at a determined time and subsequent RAs have lower and lower lifetimes.

3. The routers forcibly expire some information by advertising it with a zero lifetime for a while, and then stop advertising it.
4. A router crashes or is silently removed from the network and as a result there are no more updates. For example, that default router will expire and there is little benefit in trying to refresh it by sending lots of RSs.

The host's logic for refreshing the information needs to be careful to not send a large number of RSs, in particular if there is information that is supposed to expire at a fixed time i.e., the lifetime decrements in real time.

A host MUST NOT try to refresh information because its lifetime is near zero, since that would cause unnecessary RSs. Instead the refresh needs to be based on when the information was most recently received from the router. A lifetime of 10 minutes that was heard from the router 10 minutes ago might be normal as part of expiring some information. But a remaining lifetime of 10 minutes for a prefix that was last heard 24 hours ago with a lifetime of 24 hours means that a refresh is in order.

It is RECOMMENDED that the host track the expiry time (the wall clock time when some information will expire) and when it receives an RA with that information it SHOULD check whether the expiry time is moving forward, or appears to be frozen in time. That can tell the difference between the first two cases above, and avoid unnecessary RSs as some information naturally expires. Furthermore it is RECOMMENDED that the host track which information was received from which router, so that it can see when a router used to provide the information no longer provides it. That helps to see if the third case above might be in play. Finally, if a router has not responded to a few (e.g., MAX_RTR_SOLICITATIONS) attempts to get a refresh, then the router might be unreachable or dead, and there is little benefit in sending further RSs to that router. When the router comes back it will multicast a few RAs.

When the hosts determines that case 1 above is likely, then it should pick a reasonable time to ask for refreshes. The exact refresh behavior is not mandated by this specification, but the same implementation strategies as for refreshing address registrations in Section 8.7 can be considered.

A example simple implementation approach is to only base the refreshing on the default router lifetime (thus ignore prefix and other lifetime), and pick a refresh time which is 1/3 of the default router lifetime. If no RA is received, a subsequent refresh can be done at 2/3 of the default router lifetime. If that does not result

in a RA, then MAX_INITIAL_RTR_ADVERTISEMENTS can be sent as the router lifetime is about to expire. Note that a default router lifetime of zero MUST NOT result in sending a RS refresh based on a timeout of zero.

If the host is unable to refresh the information and as a result ends up with an empty default router list, or all the default routers are marked as UNREACHABLE by NUD, then the host MAY switch to sending initial multicast Router Solicitations as in Section 8.1.

Note that [I-D.nordmark-6man-rs-refresh] make that behavior more explicit by having the routers advertise a timeout.

8.10. Sleep and Wakeup

The protocol allows the sleepy nodes to complete its sleep schedule without waking up due to periodic Router Advertisement messages or due to Multicast Neighbor Solicitation for address resolution. The node registration lifetime SHOULD be related with its sleep interval period in order to avoid waking up in the middle of sleep for registration refresh. Depending on the application, the registration lifetime SHOULD be equal to or integral multiple of a node's sleep interval period.

When a host wakes up it can combine movement detecting (DNA), NUD, and refreshing its Address Registration by sending a unicast NS with an ARO to its existing default router(s).

8.11. Receiving Redirects

An EAH handles Redirect messages as specified in [RFC4861].

8.12. Movement Detection

When a host moves from one subnet to another its IPv6 prefix changes and the movement detection is determined according to the existing IPv6 movement detection described in [RFC6059].

9. Router Behavior

A NEAR follows [RFC4861] and applicable parts of [RFC6775] as follows in this section.

A NEAR SHOULD support legacy hosts and mixed mode as specified in this section by being able to proxy Address Resolution and DAD. The NEAR SHOULD implement a knob to be able to disable this behavior. That knob can either be set to "mixed-mode" or to "no-legacy-mode".

The RECOMMENDED default mode of operation for the NEAR is Mixed-mode.

A NEAR should be configured to advertise prefixes without the on-link (L-bit) unset. Furthermore, any legacy routers attached to the same link as a NEAR should be configured the same way. That reduces the cases in mixed mode when multicast NS messages are needed between legacy hosts and routers.

9.1. Router and/or Interface Initialization

A NEAR multicasts some initial Router Advertisements (MAX_INITIAL_RTR_ADVERTISEMENTS) at system startup or interface initialization as specified in [RFC4861] and its updates.

The NEAR MUST join the all-nodes and all-routers multicast addresses. In mixed mode it MUST also join the solicited-node multicast address(es) for its addresses and also for all the Registered NCEs.

A NEAR picks a new Router Epoch if it has lost the Registered NCEs, which is typically the case for router initialization. Either the Router Epoch can be stored in stable storage and incremented on each router initialization, or the NEAR can pick a good random number on router initialization. The effect of occasionally picking the same Router Epoch as before the state loss is that it will take longer for the router to build up its state of Registered NCEs.

9.2. Receiving Router Solicitations

Periodic RAs SHOULD be avoided. Only solicited RAs are RECOMMENDED. An RA MUST contain the Source Link-layer Address option containing Router's link-layer address (this is optional in [RFC4861]). An RA MUST carry any Prefix information option with L bit being unset, so that hosts do not multicast any NS messages as part of address resolution. A new flag (E-flag) is introduced in the RA which the hosts use to distinguish a NEAR from a legacy router.

Unlike [RFC4861] which suggests multicast Router Advertisements, this specification optimizes the exchange by always unicasting RAs in response to RSs. This is possible since the RS always includes a SLLA option, which is used by the router to unicast the RA.

If the NEAR has been configured to send an explicit set of IPv6 Registrar Addresses, or implements a Router Epoch, then the NEAR includes a RAO in all its RAs.

9.3. Periodic Multicast RA for legacy hosts

The NEAR MUST NOT send periodic RA in no-legacy mode. In mixed mode the NEAR needs to send periodic multicast RAs as specified in [RFC4861] to support legacy hosts.

9.4. Multicast RA when new information

When a router has new information to share (new prefixes, prefixes that should be immediately deprecated, etc) it MAY multicast up to MAX_INITIAL_RTR_ADVERTISEMENTS number of Router Advertisements. Note that such new information is not likely to reach sleeping hosts until those hosts refresh by sending a RS.

9.5. Receiving ARO

The NEAR follows the logic in [RFC6775] for managing the NCEs and responding to NS messages with the ARO option. The slight modification is that instead of using an EUI-64 as the key to check for duplicates, the NEAR uses the IDS value plus the variable length Unique Interface Identifier value as the key. In addition the NEAR checks the new TID field as follows.

The TID field is used together with age of a registration for arbitration between two routers to ensure freshness of the registration of a given target address. Same value of TID indicates that both states of registration are valid. In case of a mismatch between comparable TIDs, the most recent TID wins. The TIDs are compared as specified in section 7 of [RFC6550].

9.6. NCE Management in NEARs

When a host interacts with a router by sending Router Solicitations that does not match with the existing NCE entry of any type, a Legacy NCE is first created. Once a node successfully registers with a Router the result is a Registered NCE. As Routers send RAs to legacy hosts, or receive multicast NS messages from other Routers the result is Legacy NCEs.

A Router Solicitation might be received from a host that has not yet registered its address with the router or from a legacy [RFC4861] host in the Mixed-mode operation.

The router MUST NOT modify an existing Registered Neighbor Cache entry based on the SLLA option from the Router Solicitation. Thus, a router SHOULD create a tentative Legacy Neighbor Cache entry based on SLLA option when there is no match with the existing NCE. Such a legacy Neighbor Cache entry SHOULD be timed out in

TENTATIVE_LEGACY_NCE_LIFETIME seconds unless a registration converts it into a Registered NCE.

However, in 'Mixed-mode' operation, the router does not require to keep track of TENTATIVE_LEGACY_NCE_LIFETIME as it does not know if the RS request is from a legacy host or from a EAH. However, it creates the legacy type of NCE and updates it to a registered NCE if the ARO NS request arrives corresponding to the Legacy NCE. Successful processing of ARO will complete the NCE creation phase.

If ARO did not result in a duplicate address being detected, and the registration life-time is non-zero, the router creates or updates the Registered NCE for the IPv6 address. If the Neighbor Cache is full and new entries need to be created, then the router SHOULD respond with a NA with status field set to 2. For successful creation of NCE, the router SHOULD include a copy of ARO and send NA to the requester with the status field 0. A TLLA (Target Link Layer) Option is not required with this NA.

Typically for efficiency-aware routers (NEAR), the Registration Lifetime and IDS plus Unique Interface Identifier are recorded in the Neighbor Cache Entry along with the existing information described in [RFC4861]. The registered NCE are meant to be ready and reachable for communication and no address resolution is required in the link. An EAH will renew its registration to Registered NCE at the router. However the router may perform NUD towards the EAH hosts as per [RFC4861]. Should NUD fail the NEAR MUST NOT remove the Registered NCE. Instead it marks it as UNREACHABLE.

9.7. Sending non-zero status in ARO

If the Registration fails (due to it being a duplicate or the Neighbor Cache being full), then the NEAR will send an NA with ARO having a non-zero status. However, it needs to send that back to the originator of the failing ARO, and that host and link-layer address will not be present in the Neighbor Cache.

The NEAR forms a NA with ARO, and then forms the link-layer address by using the content of the SLLA option in the NS, bypassing the Neighbor Cache to send this error.

9.8. Timing out Registered NCEs

The router SHOULD NOT garbage collect Registered Neighbor Cache entries since they need to retain them until the Registration Lifetime expires. If a NEAR receives a NS message from the same host one with ARO and another without ARO then the NS message with ARO gets the precedence and the NS without ARO is ignored.

Similarly, if Neighbor Unreachability Detection on the router determines that the host is UNREACHABLE (based on the logic in [RFC4861]), the Neighbor Cache entry SHOULD NOT be deleted but be retained until the Registration Lifetime expires. If an ARO arrives for an NCE that is in UNREACHABLE state, that NCE should be marked as STALE.

The NEAR router SHOULD deny registration to a new registration request with the status code 2 when it reaches the maximum capacity for handling the neighbor cache.

9.9. Router Advertisement Consistency

The NEAR follows section 6.2.7 in [RFC4861] by receiving RAs from other routers (NEAR and legacy) on the link. In addition to the checks in that section it verifies that the prefixes do not have the L flag set, and that the Registrar Address options are consistent. Two RAOs are inconsistent if they contain the have a different Router Epoch and have some IPv6 Registration Addresses in common.

9.10. Sending Redirects

A NEAR sends redirects (with target=destination) to inform the hosts of the link-layer address of the nodes on the link.

This can be disabled on specific link types for instance, radio link technologies with hidden terminal issues.

9.11. Providing Information to Routing Protocols

If there is a routing protocols like RPL which wants visibility into the location of each IPv6 address, then this can be retrieved from the Registered NCEs on the NEAR.

9.12. Creating Legacy NCEs

In mixed-mode a NEAR will create Legacy NCEs when receiving RA, RS, and NS messages based on the source of those packets. When not in mixed-mode it needs to create Legacy NCEs for other routers by looking at those packets.

9.13. Proxy Address Resolution and DAD for Legacy Hosts

This section applies in mixed mode. It does not apply in no-legacy mode.

A NEAR in mixed mode MUST join all solicited-node for all Registered NCEs.

The NEAR SHOULD continue to support the legacy IPv6 Neighbor Solicitation requests in the mixed mode. The NEAR router SHOULD act as the ND proxy on behalf of EAH hosts for the legacy nodes' NS requests for EAH. This form of proxying is to respond with a NA that has a TLLAO taken from the Registered NCE for the target. Thus it is unlike ND Proxy as specified in [RFC4389]. (Implementations of "Neighbor Discovery Proxies (ND Proxy)" [RFC4389] might proxy using their own MAC address as TLLA, but that is outside of the scope of this document.)

In the context of this specification, proxy means:

- o Responding to DAD probes for a registered NCE. A DAD probe from a legacy host would not contain any ARO, hence the NEAR will assume it is always a duplicate if the IPv6 target address has a Registered NCE.
- o Defending a registered address using NA messages with and ARO option and the Override bit set if the ARO option in the NS indicates either a different node (different IDS+Unique Interface Id) or a older registration (when comparing the TID).
- o Advertising a registered address using NA messages, asynchronously or as a response to a Neighbor Solicitation messages.
- o Looking up a destination on the link using Neighbor Solicitation messages in order to deliver packets arriving for the EAH.

The NEAR SHOULD also support DAD from a EAH for IPv6 address that might be in use by a legacy node. Thus when a NEAR in mixed-mode received an ARO for a new address it SHOULD perform DAD as specified in [RFC4862] by sending a multicast DAD message. Once that times out the NEAR can respond to the ARO. If a legacy host responds to the DAD probe, then the NEAR will respond to the ARO with Status=1 (Duplicate Address).

10. Handling ND DoS Attack

IETF community has discussed possible issues with /64 DoS attacks on the ND networks when an attacker host can send thousands of packets to the router with an on-link destination address or sending RS messages to initiate a Neighbor Solicitation from the neighboring router which will create a number of INCOMPLETE NCE entries for non-existent nodes in the network resulting in table overflow and denial of service of the existing communications.

The efficiency-aware behavior documented in this specification avoids

the ND DoS attacks by:

- o Having the hosts register with the default router(s).
- o Having the hosts send their packets via the default router(s).
- o Not resolving addresses for the routing solicitor by mandating SLLA option along with RS
- o Checking for duplicates in NCE before the registration
- o On-link IPv6-destinations on a particular link must be registered else these packets are not resolved and extra NCEs are not created

In order to get maximal benefits from the ND-DoS protection from Address Registrations, the hosts and routers on the link need to be upgraded to NEARs and EAHs, respectively. With some legacy hosts the routers will still need to create INCOMPLETE NCEs and send NSs, which keeps the DoS opportunity open. However, with fewer legacy hosts the lower rate limits can be applied on creation of INCOMPLETE NCEs.

11. Bootstrapping

The bootstrapping mechanism described here is applicable for the efficiency-aware hosts and routers. At the start, the host uses its link-local address to send Router Solicitation and then it sends the Address Registration Option as described in this document in order to verify the IPv6 address. Note that on wakeup from sleep and after potential movement to a different link the host initially sends a unicast Neighbor Solicitation to the default router(s).

The following step 3 and 4 SHOULD be repeated for all the IPv6 addresses that are used for communications.

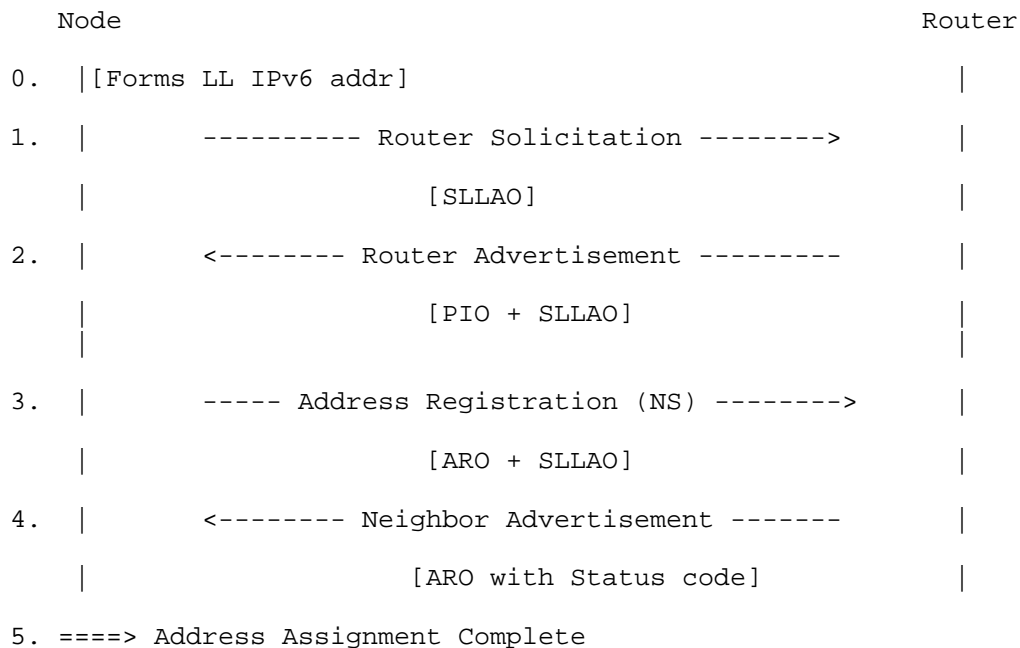


Figure 1: Neighbor Discovery Address Registration and bootstrapping

12. Interaction with other protocols

12.1. Detecting Network Attachment (DNA)

IPv6 DNA [RFC6059] uses unicast NS probes and link-layer indications to detect movement of its network attachments. That is consistent with the mechanism in this specification to unicast a NS when a host wakes up - this document recommends adding the ARO to that NS message.

Thus the ND optimization solution will work seamlessly with DNA implementations and no change is required in DNA solution because of Neighbor Discovery updates. It is a deployment and configuration consideration as to run the network in mixed mode or efficient-mode.

12.2. DHCPv6 Interaction

The protocol mechanisms in this document are orthogonal to the address assignment mechanism in use. If DHCPv6 is used for address assignment by an EAH then, if there are one or more NEARs on the subnet, the EAH will replace the DAD check specified in [RFC3315]

with Address Registration as specified in this document.

12.3. Other use of Multicast

Although the solution described in this document prevents unnecessary multicast messages in the IPv6 ND procedure, it does not affect normal IPv6 multicast packets nor the ability of nodes to join and leave the multicast group or forwarding multicast traffic or responding to multicast queries.

12.4. VRRP Interaction

A VRRP set of routers can operate with efficient-nd in two different ways:

- o Provide the illusion to hosts that they are a single router for the purposes of registrations. No RAO is needed in that case. But the pair needs some mechanism to synchronize their neighbor caches. Such a mechanism is out of scope of this document.
- o Have the hosts register with each router independently. In that case the VRRP router includes the RAO with the individual IP addresses of the routers in the pair. No synchronization of the neighbor caches are needed in that case.

13. Updated Neighbor Discovery Constants

This section discusses the updated default values of ND constants based on [RFC4861] section 10. New and changed constants are listed only for efficiency-aware-nd implementation. These values SHOULD be configurable and tunable to fit implementations and deployment.

Router Constants:

MAX_RTR_ADVERTISEMENTS(NEW)	3 transmissions
MIN_DELAY_BETWEEN_RAS(CHANGED)	1 second
TENTATIVE_LEGACY_NCE_LIFETIME(NEW)	30 seconds

Host Constants:

MAX_RTR_SOLICITATION_INTERVAL(NEW)	60 seconds
------------------------------------	------------

Also refer to [RFC6583] , [RFC7048] and [RFC6775] for further tuning of ND constants.

14. Security Considerations

These optimizations are not known to introduce any new threats against Neighbor Discovery beyond what is already documented for IPv6 [RFC3756].

Section 11.2 of [RFC4861] applies to this document as well.

This mechanism minimizes the possibility of ND /64 DoS attacks in efficiency-aware mode. See Section 10.

The mechanisms in this document work with SeND [RFC3971] in the no-legacy mode. In the mixed mode operation when a NEAR needs to respond to a legacy host sending a NS for a EAH, then SeND would not automatically fit. Potentially proxy SeND [RFC6496] could be used, but that would require explicit awareness and setup between the proxy and the proxied EAHs which seems impractical.

The mechanisms in this specification are orthogonal to the address allocation thus works as well with SLAAC and DHCPv6 as the various privacy-enhanced address allocation specifications. In particular, using an EUI-64 for the Unique Interface Identifier in this protocol does not require or assume that the IPv6 addresses will be formed using the modified EUI-64 format.

The mechanism uses a Unique Interface Identifier for the purposes of telling apart a re-registration from the same host and a duplicate/conflicting registration from a different host. That unique ID is not globally visible. Currently the protocol supports DHCPv6 DUID and EUI-64 format for this I-D, but other formats which facilitate non-linkability (such as strong random numbers large enough to be unlikely to cause collisions) can be defined.

15. IANA Considerations

A new flag (E-bit) in RA has been introduced. IANA assignment of the E-bit flag is required upon approval of this document.

This document needs a new Neighbor Discovery option type for the RAO.

16. Changelog

Changes from draft-chakrabarti-nordmark-energy-aware-nd-06:

- o Added references to dad-issues and rs-refresh.

Changes from draft-chakrabarti-nordmark-energy-aware-nd-05:

- o Fixed typos.
- o Clarified that on interface initialization after sleep or potential movement the host unicasts a NS to the default router(s).
- o Simplified the example timer handling for refreshing RA information.
- o Added handling of DAD from EAH to legacy node that was included in -04 and lost in the -05 edits.

Changes from draft-chakrabarti-nordmark-energy-aware-nd-04:

- o Significantly simplified the description of the protocol.
- o Added clarification on problem statement
- o Clarified that privacy and temporary addresses will be supported
- o Added an IDS field in the ARO to allow a DHCP Unique ID (DUID) as an alternative to EUI-64, with room to define other (pseudo) unique identifiers.
- o Allowed router redirects for NEAR.
- o Addressed some of comments made in the 6man list.
- o Added RAO to handle VRRP and similar cases when the default router list and registrar list needs to be different.
- o Added Router Epoch to cause re-registration on NEAR state loss.
- o Specified considerations for when to refresh address registrations.
- o Specified considerations for when to refresh RA information.

17. Acknowledgements

The primary idea of this document are from 6LoWPAN Neighbor Discovery document [RFC6775] and the discussions from the 6lowpan working group members, chairs Carsten Bormann and Geoff Mulligan and through our discussions with Zach Shelby, editor of the [RFC6775].

The inspiration of such a IPv6 generic document came from Margaret Wasserman who saw a need for such a document at the IOT workshop at Prague IETF.

The authors acknowledge the ND denial of service issues and key causes mentioned in the draft-halpern-6man-nddos-mitigation document by Joel Halpern. Thanks to Joel Halpern for pinpointing the problems that are now addressed in the NCE management discussion in this document.

The authors like to thank Dave Thaler, Stuart Cheshire, Jari Arkko, Ylva Jading, Niklas J. Johnsson, Reda Nedjar, Purvi Shah, Jaume Rius Riu, Fredrik Garneij, Andrew Yourtchenko, Jouni Korhonen, Suresh Krishnan, Brian Haberman, Anders Brandt, Mark Smith, Lorenzo Colitti, David Miles, Eric Vyncke, Mark ZZZ Smith, Mikael Abrahamsson, Eric Levy-Abignoli, and Carsten Bormann for their useful comments and suggestions on this work.

18. Open Issues

The known open issues are:

- o IPv6 link-local addresses are always on-link and in this version of the document that results in multicast NS messages. The technique used in 6LowPAN-ND is too restrictive (extract the link-layer address from the IID). Should we send link-locals to routers and depend on Redirect?
- o If the Router Epoch is critical then we will see a RAO in all the RAs sent by NEARs. In such a case we don't need the E-bit in the RA.
- o Editorial: Add Comparison with 6lowpan-nd and 4861?
- o Editorial: Verify and update the description in this document to make it complete removing the need to read 6LowPAN-ND.
- o When a router has new information for the RA, currently it takes a while to disseminate that to sleeping nodes as this depends on when the hosts send a RS. We could potentially improve this is we could have an "information epoch number" in the ARO sent in the NA. But that only helps if the registrations are refreshed more frequently than the RA information.
- o Future? Currently if a router changes its information, a sleeping host would not find out when it wakes up and sends the NS with ARO. That could be improved if we fit the Router Epoch in NA/ARO.

But there is no room for 16 bits.

- o A separate but related problem is with unused NCEs due to frequent IPv6 address change e.g., hosts which pick a different set of addresses each time they wake up. This document recommends that they be de-registered by the host. That could be made simpler by introducing some Host Epoch counter in the NS/ARO.

19. References

19.1. Normative References

[I-D.ietf-6man-resilient-rs]

Krishnan, S., Anipko, D., and D. Thaler, "Packet loss resiliency for Router Solicitations", draft-ietf-6man-resilient-rs-04 (work in progress), October 2014.

[I-D.nordmark-6man-rs-refresh]

Nordmark, E., Yourtchenko, A., and S. Krishnan, "IPv6 Neighbor Discovery Optional Unicast RS/RA Refresh", draft-nordmark-6man-rs-refresh-01 (work in progress), October 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

[RFC6775] Shelby, Z., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, November 2012.

19.2. Informative References

[I-D.ietf-6man-default-iids]

Gont, F., Cooper, A., Thaler, D., and W. Will, "Recommendation on Stable IPv6 Interface Identifiers", draft-ietf-6man-default-iids-02 (work in progress), January 2015.

[I-D.ietf-6man-stable-privacy-addresses]

Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address

Autoconfiguration (SLAAC)",
draft-ietf-6man-stable-privacy-addresses-17 (work in
progress), January 2014.

[I-D.vyncke-6man-mcast-not-efficient]

Vyncke, E., Thubert, P., Levy-Abegnoli, E., and A.
Yourtchenko, "Why Network-Layer Multicast is Not Always
Efficient At Datalink Layer",
draft-vyncke-6man-mcast-not-efficient-01 (work in
progress), February 2014.

[I-D.yourtchenko-6man-dad-issues]

Yourtchenko, A., "A survey of issues related to IPv6
Duplicate Address Detection",
draft-yourtchenko-6man-dad-issues-00 (work in progress),
October 2014.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C.,
and M. Carney, "Dynamic Host Configuration Protocol for
IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3756] Nikander, P., Kempf, J., and E. Nordmark, "IPv6 Neighbor
Discovery (ND) Trust Models and Threats", RFC 3756,
May 2004.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure
Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC4389] Thaler, D., Talwar, M., and C. Patel, "Neighbor Discovery
Proxies (ND Proxy)", RFC 4389, April 2006.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless
Address Autoconfiguration", RFC 4862, September 2007.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy
Extensions for Stateless Address Autoconfiguration in
IPv6", RFC 4941, September 2007.

[RFC6059] Krishnan, S. and G. Daley, "Simple Procedures for
Detecting Network Attachment in IPv6", RFC 6059,
November 2010.

[RFC6496] Krishnan, S., Laganier, J., Bonola, M., and A. Garcia-
Martinez, "Secure Proxy ND Support for SEcure Neighbor
Discovery (SEND)", RFC 6496, February 2012.

[RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R.,
Levis, P., Pister, K., Struik, R., Vasseur, JP., and R.

Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.

[RFC6574] Tschofenig, H. and J. Arkko, "Report from the Smart Object Workshop", RFC 6574, April 2012.

[RFC6583] Gashinsky, I., Jaeggli, J., and W. Kumari, "Operational Neighbor Discovery Problems", RFC 6583, March 2012.

[RFC7048] Nordmark, E. and I. Gashinsky, "Neighbor Unreachability Detection Is Too Impatient", RFC 7048, January 2014.

Authors' Addresses

Samita Chakrabarti
Ericsson
San Jose, CA
USA

Email: samita.chakrabarti@ericsson.com

Erik Nordmark
Arista Networks
Santa Clara, CA
USA

Email: nordmark@arista.com

Pascal Thubert
Cisco Systems

Email: pthubert@cisco.com

Margaret Wasserman
Painless Security

Email: mrw@painless-security.com

IPv6 maintenance Working Group (6man)
Internet-Draft
Obsoletes: 6564 (if approved)
Intended status: Standards Track
Expires: March 19, 2016

F. Gont
SI6 Networks / UTN-FRH
W. Liu
Huawei Technologies
S. Krishnan
Ericsson
H. Pfeifer
Rohde & Schwarz
September 16, 2015

IPv6 Universal Extension Header
draft-gont-6man-ipv6-universal-extension-header-02

Abstract

In IPv6, optional internet-layer information is encoded in separate headers that may be placed between the IPv6 header and the transport-layer header. There are a small number of such extension headers currently defined. This document describes the issues that can arise when defining new extension headers and specifies a new IPv6 Extension Header - the Universal Extension Header - that overcomes the aforementioned problem, while enabling the extensibility of IPv6. Finally, this document formally obsoletes RFC 6564.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 19, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Terminology	3
3. A Problem with RFC 6564	3
4. Implications	3
5. UEH Specification	4
6. Forbidding New IPv6 Extension Headers	5
7. Operation of the UEH	5
8. IANA Considerations	5
9. Security Considerations	6
10. Acknowledgements	6
11. Contributors	6
12. References	6
12.1. Normative References	6
12.2. Informative References	6
Authors' Addresses	7

1. Introduction

There has recently been a lot of work in the area of IPv6 Extension Headers. Firstly, there has been research about the extent to which IPv6 packets employing Extension Headers are dropped in the public Internet [GONT-IEPG-Nov13] [GONT-IEPG-Mar14], and debate about the motivation behind such policy [I-D.gont-v6ops-ipv6-ehs-packet-drops]. Secondly, there has been a fair share of work to improve some technicalities of IPv6 Extension Headers (see e.g. [RFC7112] [RFC7045]) in the hopes that they can be reliably used in the public Internet.

A key challenge for IPv6 Extension Headers to be "deployable" in the public Internet is that they should not impair any nodes's ability to process the entire IPv6 header chain. One of the steps meant in that direction has been the specification of a Uniform Format for IPv6 Extension Headers [RFC6564], which was meant to be employed by any IPv6 Extension Headers that might be defined in the future, such that middle-boxes can still process the entire IPv6 header chain if new extension headers were specified. However, a problem in the

aforementioned specification prevents such uniform format from being of use.

Section 3 discusses the aforementioned flaw in the Uniform Format for Extension Headers specified in [RFC6564]. Section 4 explicitly describes the implications of the aforementioned flaw. Section 5 specifies the new Universal Extension Header (UEH). Section 7 explains how new IPv6 extensions would be specified with the UEH. Section 6 formally forbids the specification of new IPv6 Extension Headers (with new Next Header values), and mandates that any new IPv6 extensions be conveyed/encoded in the UEH specified in this document.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. A Problem with RFC 6564

A key problem with the Uniform Format for IPv6 Extension Headers [RFC6564] lies in that both IPv6 Extension Headers and Transport Protocols share the same "Next Header" registry/namespace. Thus, given an "unknown Next Header value", it is impossible to tell whether the aforementioned value refers to an IPv6 Extension Header that employs the aforementioned uniform format, or an "unknown" upper-layer protocol (e.g. an "unknown" transport protocol). That is, while [RFC6564] specifies the syntax for a Uniform Format for IPv6 Extension Headers, it does not provide a mechanism for a node to identify whether the aforementioned format is being employed in the first place.

4. Implications

The current impossibility to parse an IPv6 header chain that includes unknown Next Header values results in concrete implications for the extensibility of the IPv6 protocol, and the deployability of new transport protocols. Namely,

- o New IPv6 extension headers cannot be incrementally deployed.
- o New transport protocols cannot be incrementally deployed.

Since there is no way for a node to process IPv6 extension headers that employ unknown next header values, an IPv6 host that receives a packet that employs a new IPv6 extension header will not be able to parse the IPv6 header chain past that unknown extension header, and hence it will drop the aforementioned packet

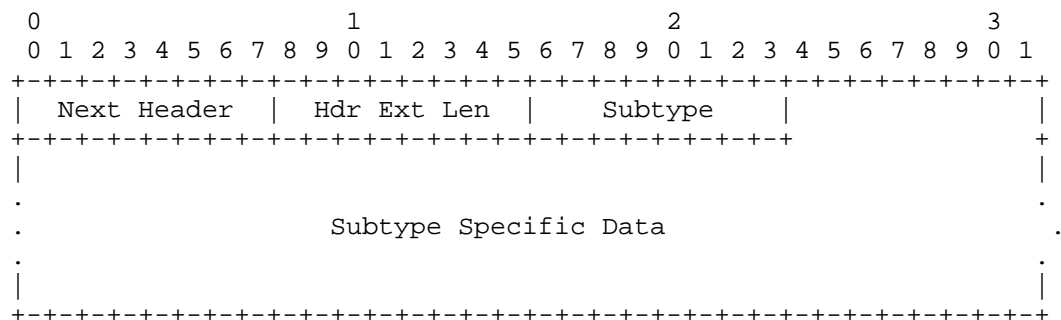
[I-D.gont-v6ops-ipv6-ehs-packet-drops]. In a similar way, a middlebox that needs to process the transport-protocol header will be faced with the dilemma of what to do with packets that employ unknown Next Header values. Since they will not be able to parse the IPv6 header chain past the unknown Next Header, it is very likely that they will drop such packets.

Unfortunately, since transport protocols share the same namespace as IPv6 Extension Headers, new transport protocols will pose the same challenge to middle-boxes, and hence they will be likely dropped in the network.

We believe that the current situation has implications that are generally overlooked, and that, whatever the outcome, it should be the result of an explicit decision by our community, rather than simply "omission".

5. UEH Specification

This document specifies a new IPv6 Extension Header: Universal Extension Header. This Extension Header is identified by the value [TBD] of [IANA-IP-PROTO]. The syntax of the Universal Extension Header is:



where:

Next Header

8-bit selector. Identifies the type of header immediately following the extension header. Uses the same values as the IPv4 Protocol field [IANA-IP-PROTO].

Hdr Ext Len

8-bit unsigned integer. Length of the extension header in 8-octet units, not including the first 8 octets.

Subtype

8-bit unsigned integer. Specifies the subtype for this extension header. It uses a new namespace managed by IANA [IANA-UEH].

Subtype Specific Data

Variable length. Fields specific to this extension header/Subtype.

The Universal Extension Header specified in this document MAY appear multiple times in the same IPv6 packet.

6. Forbidding New IPv6 Extension Headers

Since the specification of any new IPv6 Extension Headers (i.e., with new Next Header values) would hamper (among other things) the incremental deployment of extensions and new transport protocols, and basic operational practices such as the enforcement of simple ACLs, new IPv6 Extension Headers MUST NOT be specified in any future specifications. Any IPv6 extensions that would require a new IPv6 Extension Header MUST be implemented with the Universal Extension Header specified in this document. This minimizes breakage in intermediate nodes that need to parse the entire IPv6 header chain.

7. Operation of the UEH

This section describes the operation of the Universal Extension Header.

The goal of the UEH is to provide a common syntax for all future IPv6 extensions. Any future extension headers will be encoded in a UEH, and will be identified by a specific UEH Subtype assigned by IANA at the time the corresponding specification is published. The UEH thus provides the "common syntax" required to process "unrecognized extensions", and the Subtype field identifies the specific extension being encoded in the UEH. Any "future extension headers" would actually be new Subtypes (assigned by IANA) of the UEH.

As a result, unrecognized Next Header values should be interpreted to identify an upper-layer protocol, rather than an IPv6 extension header.

8. IANA Considerations

IANA is requested to create a new registry to maintain the Universal Extension Header Subtypes [IANA-UEH].

9. Security Considerations

Enabling nodes to parse an entire IPv6 header chain even in the presence of unrecognized extensions allows for security mechanisms to be implemented and deployed.

10. Acknowledgements

The authors would like to thank [TBD] for providing valuable input on earlier versions of this document.

11. Contributors

C.M. Heard identified the problems related with the Uniform Format for IPv6 Extension Headers specified in [RFC6564], and participated in the brainstorming that led to this document.

12. References

12.1. Normative References

- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<http://www.rfc-editor.org/info/rfc2460>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, DOI 10.17487/RFC6564, April 2012, <<http://www.rfc-editor.org/info/rfc6564>>.
- [RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <<http://www.rfc-editor.org/info/rfc7045>>.

12.2. Informative References

- [RFC7112] Gont, F., Manral, V., and R. Bonica, "Implications of Oversized IPv6 Header Chains", RFC 7112, DOI 10.17487/RFC7112, January 2014, <<http://www.rfc-editor.org/info/rfc7112>>.

[I-D.gont-v6ops-ipv6-ehs-packet-drops]

Gont, F., Hilliard, N., Doering, G., LIU, S., and W. Kumari, "Operational Implications of IPv6 Packets with Extension Headers", draft-gont-v6ops-ipv6-ehs-packet-drops-00 (work in progress), July 2015.

[GONT-IEPG-Nov13]

Gont, F., "Fragmentation and Extension Header Support in the IPv6 Internet", IEPG 88, November 3, 2013. Vancouver, BC, Canada, 2013, <<http://www.iepg.org/2013-11-ietf88/fgont-iepg-ietf88-ipv6-frag-and-eh.pdf>>.

[GONT-IEPG-Mar14]

Gont, F. and T. Chown, "More results from measurements of IPv6 Extension Header probing", IEPG 89, March 2, 2014. London, U.K., 2014, <<http://www.iepg.org/2014-03-02-ietf89/fgont-iepg-ietf89-eh-update.pdf>>.

[IANA-IP-PROTO]

Internet Assigned Numbers Authority, "Assigned Internet Protocol Numbers", April 2011, <<http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>>.

[IANA-UEH]

Internet Assigned Numbers Authority, "Universal Extension Header Subtypes", 2014.

Authors' Addresses

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

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

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

Email: liushucheng@huawei.com

Suresh Krishnan
Ericsson
8400 Decarie Blvd.
Town of Mount Royal, QC
Canada

Phone: +1 514 345 7900 x42871
Email: suresh.krishnan@ericsson.com

Hagen Paul Pfeifer
Rohde & Schwarz
Muehldorfstrasse 15
Munich 81671
Germany

Phone: +49 89 4129 15515
Email: hagen.pfeifer@rohde-schwarz.com
URI: <http://www.rohde-schwarz.com/>

IPv6 maintenance Working Group (6man)
Internet-Draft
Updates: 4861 (if approved)
Intended status: Standards Track
Expires: August 18, 2014

F. Gont
SI6 Networks / UTN-FRH
R. Bonica
Juniper Networks
W. Liu
Huawei Technologies
February 14, 2014

Validation of Neighbor Discovery Source Link-Layer Address (SLLA) and
Target Link-layer Address (TLLA) options
draft-gont-6man-lla-opt-validation-00

Abstract

This memo documents two scenarios in which an on-link attacker emits a crafted IPv6 Neighbor Discovery (ND) packet that poisons its victim's neighbor cache. In the first scenario, the attacker causes a victim to map a local IPv6 address to a local router's own link-layer address. In the second scenario, the attacker causes the victim to map a unicast IP address to a link layer broadcast address. In both scenarios, the attacker can exploit the poisoned neighbor cache to perform a subsequent forwarding-loop attack, thus potentially causing a Denial of Service.

Finally, this memo specifies simple validations that the recipient of an ND message can execute in order to protect itself against the above-mentioned threats.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 18, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Terminology	3
3. ND-based Forwarding-Loop Attacks	3
3.1. Mapping an IPv6 Address to a Local Router's Own Link-layer Address	3
3.2. Mapping a Unicast IPv6 Address to A Broadcast Link-Layer Address	4
4. Implications of Malicious Link-layer Address Options	6
5. Validation Checks for the Source Link-Layer Address Option	7
6. Validation Checks for the Target Link-Layer Address Option	8
7. IANA Considerations	8
8. Security Considerations	9
9. Acknowledgements	9
10. References	9
10.1. Normative References	9
10.2. Informative References	9
Authors' Addresses	9

1. Introduction

IPv6 [RFC2460] nodes use a Neighbor Discovery (ND) [RFC4861] mechanism to discover on-link neighbors and learn their link layer addresses. Having discovered an on-link neighbor and learned its link layer address, an IPv6 node stores that information in a local data structure, called the "neighbor cache".

ND defines the following ICMPv6 [RFC4443] messages:

- o Router Solicitation (RS)
- o Router Advertisement (RA)

- o Neighbor Solicitation (NS)
- o Neighbor Advertisement (NA)
- o Redirect

ND also defines a Source Link-Layer Address (SLLA) option and a Target Link-Layer Address (TLLA) option. The RS, RA, and NS messages all typically contain the SLLA option, that contains the link layer address of the node sending the message. The NA and Redirect messages contain the TLLA option, that maps a target IPv6 address that is contained by the NA or Redirect message to a link layer address.

This memo documents two scenarios in which an on-link attacker emits a crafted ND packet that poisons its victim's neighbor cache. In the first scenario, the attacker causes a victim to map an IPv6 address to a the victim router's own link-layer address. In the second scenario, the attacker causes the victim to map a unicast IP address to the link layer broadcast or multicast address. In both scenarios, the attacker can subsequently exploit the poisoned neighbor cache to perform a forwarding-loop attack, thus potentially causing a Denial of Service (DoS).

Finally, this memo specifies simple validations that the recipient of an ND message can execute in order to protect itself against the above-mentioned threats.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. ND-based Forwarding-Loop Attacks

3.1. Mapping an IPv6 Address to a Local Router's Own Link-layer Address

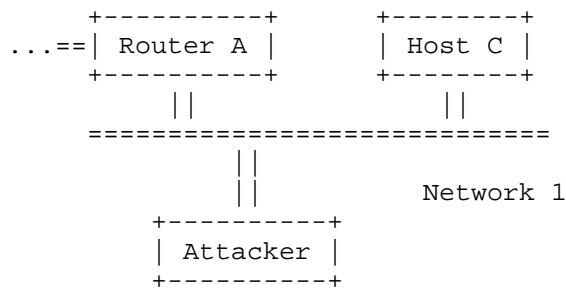


Figure 1: Unicast Forwarding Loop

In Figure 1, the Attacker sends Router A a crafted ND message. The aforementioned ND message contains the Target Address set to Host C's IPv6 address, and a TLLA option set to Router A's link-layer address. The ND message causes Router A to map Host C's IPv6 address to the link layer address of Router A's interface to Network 1. This sets up the scenario for a subsequent attack.

A packet is sent to Router A with the IPv6 Destination Address set to that of Host C. Router A forwards the packet on Network 1, specifying its own Network 1 interface as the link layer destination. Because Router A specified itself as the link layer destination, Router A receives the packet and forwards it again. This process repeats until the IPv6 Hop Limit is decremented to 0 (and hence the packet is discarded). In this scenario, the amplification factor is equal to the Hop Limit minus one.

An attacker can realize this attack by sending either of the following:

- o An ND message whose SLLA maps an IPv6 address to the link layer address of the victim router's (Router A's in our case) interface to the local network (Network 1 in our case)
- o An ND message whose TLLA maps an IPv6 address to the link layer address of the victim router's (Router A's in our case) interface to the local network (Network 1 in our case)

3.2. Mapping a Unicast IPv6 Address to A Broadcast Link-Layer Address

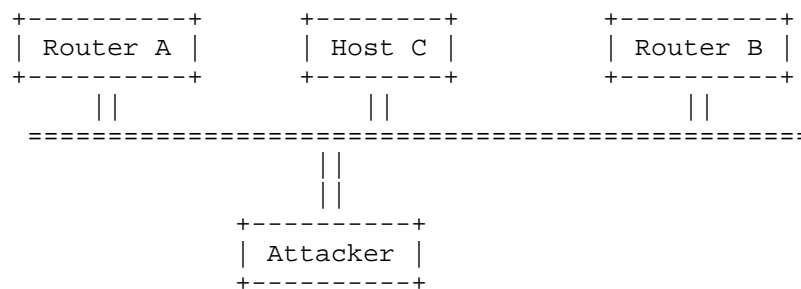


Figure 2: Broadcast Forwarding Loop

In Figure 2, the Attacker sends one crafted ND message to Router A, and one crafted ND message to Router B. Each crafted ND message contains the Target Address set to Host C's IPv6 address, and a TLLA option set to the Ethernet broadcast address (ff:ff:ff:ff:ff:ff). These ND messages causes each router to map Host C's IPv6 address to the Ethernet broadcast address. This sets up the scenario for a subsequent attack.

Subsequently, the Attacker sends a packet to the Ethernet broadcast address (ff:ff:ff:ff:ff:ff), with an IPv6 Destination Address equal to the IPv6 address of Host C. Upon receipt, both Router A and Router C decrement the Hop Limit of the packet, and resend it to the Ethernet broadcast address. As a result, both Router A and Router B receive two copies of the same packet (one sent by Router A, and another sent by Router B). This would result in a "chain reaction" that would only disappear once the Hop Limit of each of the packets is decremented to 0. The equation in Figure 3 describes the amplification factor for this scenario :

$$\text{Packets} = \frac{\text{HopLimit}-1}{x=0} \times \text{Routers}$$

Figure 3: Maximum amplification factor

This equation does not take into account ICMPv6 Redirect messages that each of the Routers could send, nor the possible ICMPv6 "time exceeded in transit" error messages that each of the routers could send to the Source Address of the packet when each of the "copies" of the original packet is discarded as a result of their Hop Limit being decremented to 0.

An attacker can realize this attack by sending either of the following:

- o An ND message whose SLLA maps an IPv6 address not belonging to the victim routers to the broadcast link-layer address
- o An ND message whose TLLA maps an IPv6 address not belonging to the victim routers to the broadcast link-layer address

NOTE: the IPv6 Destination Address of the attack packet should not belong to any of the victim routers, such that they forward the packet rather than "consume" it.

An additional mitigation would be for routers to not forward IPv6 packets on the same interface if the link-layer destination address of the received packet was a broadcast or multicast address.

4. Implications of Malicious Link-layer Address Options

If SLLA or TLLA options are allowed to contain broadcast (e.g., the IEEE 802 "ff:ff:ff:ff:ff:ff") or multicast (e.g., the IEEE 802 "33:33:00:00:00:01") addresses, traffic directed to the corresponding IPv6 address would be sent to the broadcast or multicast address specified in the SLLA or TLLA option. This could have multiple implications:

- o It would have a negative impact on the performance of the nodes attached to the network and on the network itself, as packets sent to these addresses would need to be delivered to multiple nodes (and processed by them) unnecessarily.
- o An attacker could easily capture traffic on a switched network, without the need to forward packets to their intended destinations, as the corresponding packets would be delivered to all (in the case of broadcast) or multiple (in the case of multicast) nodes.
- o Packets could result in forwarding loops at routers, as a router forwarding a packet to the corresponding address would receive itself a copy of the forwarded packet. The loop would end only when the Hop Limit is eventually decremented to 0. The problem would be exacerbated if multiple routers are present on the same link. Section 3 of this document contains further analysis of this vulnerability.

Additionally, if SLLA or TLLA options are allowed to contain the receiving router's own link-layer address, the victim router would

receive a copy of the very same packets it means to forward to other destinations. This could have the following implications:

- o It would have a negative impact on the performance of the victim router and of the network itself, as a single packet would be sent multiple times (up to 255) on the local network, thus serving as an amplification vector.

5. Validation Checks for the Source Link-Layer Address Option

The Source link-layer address option contains the link-layer address of the sender of the packet. It is used by Neighbor Solicitation, Router Solicitation, and Router Advertisement messages.

The following figure illustrates the syntax of the source link-layer address:

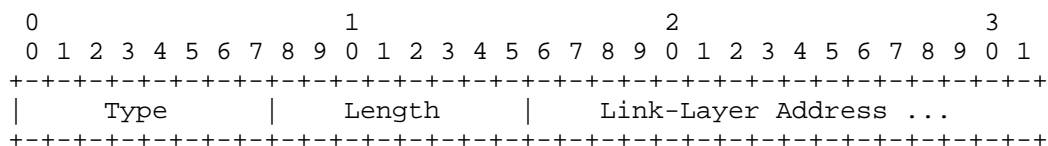


Figure 4: ND Source link-layer address option

The Type field is set to 1. The Length field specifies the length of the option (including the Type and Length octets) in units of 8 octets. A node that receives an ICMPv6 message with this option MUST verify that the Length field is valid for the underlying link layer. For example, for IEEE 802 addresses the Length field MUST be 1 [RFC2464]. If the packet does not pass this check, it MUST be silently dropped.

NOTE: The Link-Layer Address field contains the link-layer address. The length, contents, and format of this field varies from one link layer to another, and is specified in specific documents that describes how IPv6 operates over different link layers.

Additionally, the SLLA option MUST NOT contain a broadcast or multicast address. If the option does not pass this check, the Neighbor Discovery message carrying the option MUST be discarded. Finally, nodes MUST NOT allow the SLLA option to contain one of the receiving node's link-layer addresses. If the option does not pass this check, the Neighbor Discovery message carrying the option MUST be discarded.

6. Validation Checks for the Target Link-Layer Address Option

The Target link-layer address option contains the link-layer address of the Target of the packet. It is used by Neighbor Advertisement and Redirect messages.

The following figure illustrates the syntax of the Target link-layer address:

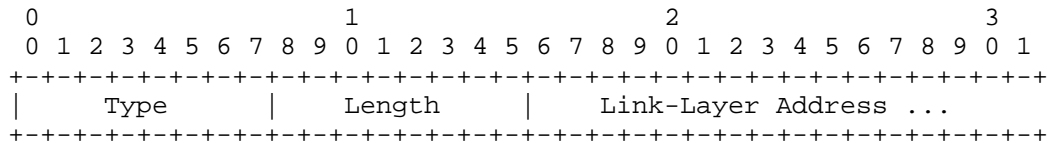


Figure 5: ND Target link-layer address option format

The Type field is set to 2. The Length field specifies the length of the option (including the Type and Length octets) in units of 8 octets. A node that receives a ND message with this option **MUST** verify that the Length field is valid for the underlying link-layer. For example, for IEEE 802 addresses the Length field **MUST** be 1 [RFC2464]. If the packet does not pass this check, it **MUST** be silently dropped.

A node that receives a ND message with this option **MUST** verify that the Length field is valid for the underlying link layer. For example, for IEEE 802 addresses the Length field **MUST** be 1 [RFC2464]. If the packet does not pass this check, it **MUST** be silently dropped.

The TLLA option **MUST NOT** contain a broadcast or multicast address. If the option does not pass this check, the Neighbor Discovery message carrying the option **MUST** be discarded. Finally, nodes **MUST NOT** allow the source link-layer address to contain one of the receiving node's link-layer addresses. If the option does not pass this check, the Neighbor Discovery message carrying the option **MUST** be discarded.

7. IANA Considerations

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

8. Security Considerations

This document discusses how the Neighbor Discovery SLLA and TLLA options can be leveraged to perform a number of attacks, and specifies sanity checks to be enforced by Neighbor Discovery implementations, such that these vulnerabilities are eliminated.

9. Acknowledgements

This document is based on the technical report "Security Assessment of the Internet Protocol version 6 (IPv6)" [CPNI-IPv6] authored by Fernando Gont on behalf of the UK Centre for the Protection of National Infrastructure (CPNI).

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 4443, March 2006.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

10.2. Informative References

- [CPNI-IPv6]
Gont, F., "Security Assessment of the Internet Protocol version 6 (IPv6)", UK Centre for the Protection of National Infrastructure, (available on request).

Authors' Addresses

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

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

Ronald P. Bonica
Juniper Networks
2251 Corporate Park Drive
Herndon, VA 20171
US

Phone: 571 250 5819
Email: rbonica@juniper.net

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

Email: liushucheng@huawei.com

Internet Draft
<draft-halpern-6man-nd-pre-resolve-addr-00.txt>
Category: Informational
Expires in 6 months

I. Chen
J. Halpern
Ericsson
January 10, 2014

Triggering ND Address Resolution on Receiving DAD-NS
<draft-halpern-6man-nd-pre-resolve-addr-00.txt>

Status of this Memo

Distribution of this memo is unlimited.

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on date.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This draft proposes a new optional event to trigger address

resolution using IPv6 Neighbor Discovery. This helps optimize router performance, and can help mitigate certain potential ND-related denial-of-service attacks. Upon receiving a DAD-NS message, the neighbor solicitation message used to detect duplicate addresses, if the target address encoded in the DAD-NS is not a duplicate address, the receiving device responds by triggering address resolution for the target address in the DAD-NS, in preparation for expectant future communication with the sending device.

Table of Contents

1. Introduction	3
2. Proposed Trigger for Address Resolution	4
3. Which Devices to Upgrade and the Consequences	6
4. Security Considerations	6
5. IANA Considerations	6
6. References	6

1. Introduction

Due to the large address space for IPv6 [RFC2460] and a large /64 default subnet size, Neighbor Discovery (ND) for IPv6 [RFC4861] could suffer from off-link flooding Denial-of-Service (DoS) attacks [RFC6583]. In such an attack, a remote malicious device could flood a router with packets destined to billions of unassigned IPv6 addresses. Although these packets are destined to unused IPv6 addresses, cache misses could occur nonetheless. Without special handling of cache misses, the router would trigger address resolution for billions of unused IPv6 addresses. The sheer volume of IPv6 addresses could overwhelm the router's normal ND protocol processing and ultimately prevent the router from forwarding packets destined to legitimate IPv6 addresses.

[RFC6583] proposes implementation and operational practices to reduce the impact of an off-link flooding DoS attack without modifying the ND protocol. The Internet Draft [ndmit] goes further and poses the question whether cache misses, an important trigger for address resolution in the ND protocol, are necessary. If cache misses can be ignored, then an off-link flooding DoS attack that uses cache misses to compromise a router can be neutralized. To eliminate the need for cache misses, a router should retain the neighbor cache entries of all legitimate neighbors on the physical link.

This draft proposes that a router further triggers address resolution based on an event other than a cache miss. In addition to waiting for a cache miss to trigger address resolution, a router should initiate address resolution for the target address in a DAD-NS, provided that the target address is not a duplicate address of the receiving device or a resolved neighbor.

Consequently, to optimize IPv6 router performance and to avoid neighbor cache overrun by remote exploration, an IPv6 device:

- 1) SHOULD NOT remove a populated cache entry to make room for a pending entry based on a received packet trigger.
- 2) SHOULD NOT remove a DAD triggered pending entry to make room

for a remote received packet triggered entry.

- 3) SHOULD remove remote trigger pending entries if needed to make room for DAD triggered pending entries.

For an even stronger solution to prevent neighbor cache overrun by remote exploration, a router can implement [ndmit] in conjunction with the mechanism in this draft.

2. Proposed Trigger for Address Resolution

In IPv6, when a device initializes an interface, a special Neighbor Solicitation (NS) message is sent to perform Duplicate Address Detection (DAD) [RFC4862] to determine whether a particular address is already assigned to a different interface on the same multi-access link. This special message, referred to as DAD-NS in the rest of this draft, is an NS message with an unspecified source address. The target address of this DAD-NS is the IPv6 unicast address that is intended for new interface.

In addition to the detection of duplicate addresses, a DAD-NS can also be treated as an announcement for a new address, the target address in the DAD-NS, which will be used in the near future, after the DAD algorithm has been completed. Consequently, after allowing time for the DAD algorithm to be completed, rather than waiting for a cache miss, the router that received the DAD-NS can perform address resolution for the target address in the DAD-NS.

The proposed steps are similar to how address resolution is initiated when a device receives a regular NS message, one that has a specified source address. The difference in the mechanism proposed by this draft is that the address resolution is not triggered immediately after receiving the DAD-NS. Instead, address resolution is triggered with a time delay to accommodate the DAD algorithm.

For example in Figure 1, when DEV2 initializes an interface that is expected to use the IP address 2001::15, a DAD-NS with an unspecified source address and a target address of 2001::15 is multicast on the physical link. Following the DAD algorithm in [RFC4862], when DEV1 on the physical link receives such a DAD-NS, the DEV1 device does not respond to the DAD-NS if the target address 2001::15 is not used by one of its interfaces. Assuming that DEV1 implements the proposed DAD-NS response in this draft, then after allowing for the DAD algorithm to be completed, DEV1 can trigger address resolution for 2001::15, the target address announced in the previous DAD-NS, without waiting for a cache miss to occur.

Furthermore, when DEV2 receives the NS to query for target address

2001::15, [RFC4861] Section 7.2.3 specifies that DEV2 respond with an NS query of its own for the source address of the NS that DEV2 just received. Thus, at the end of the DAD-NS, NS, and Neighbor Advertisement (NA) message exchanges that are triggered by the initialization of DEV2's interface, DEV1 and DEV2 have each others' neighbor entries. The two devices can immediately begin communication shortly after DEV2 sends the DAD-NS and very likely before any cache miss occurs.

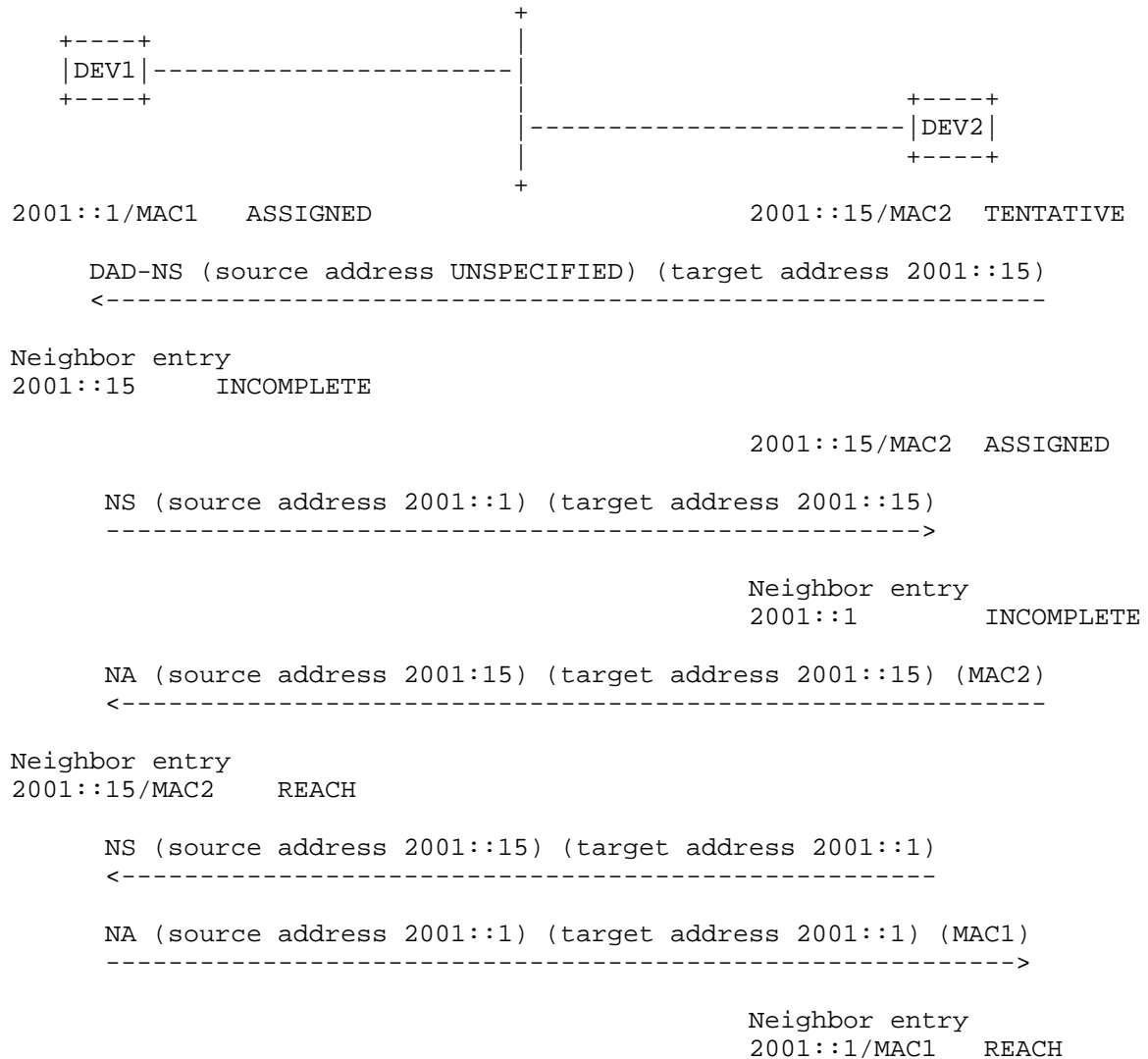


Figure 1. An example of DAD-NS triggering address resolution.

3. Which Devices to Upgrade and the Consequences

This proposed mechanism does not require changes to [RFC4861]. Further, devices that implement this proposal can interoperate with devices that do not implement this proposal. Ericsson's Smart Services Router implemented this change in early 2013, is deployed in operational IPv6 networks, and has not encountered any problems.

The proposed mechanism probably is more useful for routers than for hosts, although nothing prevents a host from implementing this proposal and hosts might benefit from implementing this proposal.

If all devices, both routers and hosts, on a physical link implement the proposed change, then when a device restarts, the restarting device can easily recover all the pre-restart neighbor cache entries. Using Figure 1 as an example, assume that DEV2 restarts and re-initializes its interface, and once again wishes to assign its interface the address 2001::15. Because DEV1 implements this draft and responds to the DAD-NS by querying for 2001::15, both DEV1 and DEV2 end up with the same neighbor cache entries from before DEV2 restarted.

4. Security Considerations

The proposed trigger for address resolution might suffer from certain attacks if the attacker is on the same physical link as the new IPv6 device and sends bogus DAD-NS messages. However, no mechanism can protect a device when the attacker is on the same physical link as the device, other than ensuring that only authorized devices have access to a physical link (e.g., by using link-layer security mechanisms, such as IEEE 802.1AE link encryption [802.1AE]).

5. IANA Considerations

No actions are required from IANA as result of the publication of this document.

6. References

6.1. Normative References

- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

6.2. Informative References

- [RFC6583] Gashinsky, I., Jaeggli, J., and W. Kumari, "Operational Neighbor Discovery Problems", RFC 6583, March 2012.
- [ndmit] Halpern, J, Work in progress, "draft-halpern-6man-nddos-mitigation-00", October 2011.
- [802.1AE] IEEE Standards Association, "IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Security", IEEE Standard 802.1AE, IEEE, Piscataway, NJ, USA, August 18, 2006.

Authors' Addresses

I. Chen
Ericsson
Email: ing-wher.chen@ericsson.com

J. Halpern
Ericsson
EMail: joel.halpern@ericsson.com

IPv6 maintenance Working Group (6man)
Internet-Draft
Updates: 2464, 2467, 2470, 2491, 2492,
2497, 2590, 3146, 3572, 4291,
4338, 4391, 5072, 5121 (if
approved)
Intended status: Standards Track
Expires: March 29, 2017

F. Gont
SI6 Networks / UTN-FRH
A. Cooper
Cisco
D. Thaler
Microsoft
W. Liu
Huawei Technologies
September 28, 2016

Recommendation on Stable IPv6 Interface Identifiers
draft-ietf-6man-default-iids-16

Abstract

This document changes the recommended default IID generation scheme for cases where SLAAC is used to generate a stable IPv6 address. It recommends using the mechanism specified in RFC7217 in such cases, and recommends against embedding stable link-layer addresses in IPv6 Interface Identifiers. It formally updates RFC2464, RFC2467, RFC2470, RFC2491, RFC2492, RFC2497, RFC2590, RFC3146, RFC3572, RFC4291, RFC4338, RFC4391, RFC5072, and RFC5121. This document does not change any existing recommendations concerning the use of temporary addresses as specified in RFC 4941.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 21, 2017.

Copyright Notice

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

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

Table of Contents

1. Introduction	2
2. Terminology	4
3. Generation of IPv6 Interface Identifiers with SLAAC	4
4. Future Work	4
5. IANA Considerations	5
6. Security Considerations	5
7. Acknowledgements	5
8. References	5
Authors' Addresses	8

1. Introduction

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

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

The security and privacy implications of embedding a stable link-layer address in an IPv6 IID have been known for some time now, and are discussed in great detail in [RFC7721]. They include:

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

More generally, the reuse of identifiers that have their own semantics or properties across different contexts or scopes can be detrimental for security and privacy [I-D.gont-predictable-numeric-ids]. In the case of traditional stable IPv6 IIDs, some of the security and privacy implications are dependent on the properties of the underlying link-layer addresses (e.g., whether the link-layer address is ephemeral or randomly generated), while other implications (e.g., reduction of the entropy of the IID) depend on the algorithm for generating the IID itself. In standardized recommendations for stable IPv6 IID generation meant to achieve particular security and privacy properties, it is therefore necessary to recommend against embedding stable link-layer addresses in IPv6 IIDs.

Furthermore, some popular IPv6 implementations have already deviated from the traditional stable IID generation scheme to mitigate the aforementioned security and privacy implications [Microsoft].

As a result of the aforementioned issues, this document changes the recommended default IID generation scheme for generating stable IPv6 addresses with SLAAC to that specified in [RFC7217], and recommends against embedding stable link-layer addresses in IPv6 Interface Identifiers, such that the aforementioned issues are mitigated. That is, this document simply replaces the default algorithm that is recommended to be employed when generating stable IPv6 IIDs.

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

In a variety of scenarios, addresses that remain stable for the lifetime of a host's connection to a single subnet, are viewed as desirable. For example, stable addresses may be viewed as beneficial for network management, event logging, enforcement of access control, provision of quality of service, or for server or routing interfaces. Similarly, stable addresses (as opposed to temporary addresses [RFC4941]) allow for long-lived TCP connections, and are also usually desirable when performing server-like functions (i.e., receiving incoming connections).

The recommendations in this document apply only in cases where implementations otherwise would have configured a stable IPv6 IID containing a link layer address. For example, this document does not change any existing recommendations concerning the use of temporary addresses as specified in [RFC4941], nor do the recommendations apply to cases where SLAAC is employed to generate non-stable IPv6

addresses (e.g. by embedding a link-layer address that is periodically randomized), nor does it introduce any new requirements regarding when stable addresses are to be configured. Thus, the recommendations in this document simply improve the security and privacy properties of stable addresses.

2. Terminology

Stable address:

An address that does not vary over time within the same network (as defined in [RFC7721]).

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Generation of IPv6 Interface Identifiers with SLAAC

Nodes SHOULD implement and employ [RFC7217] as the default scheme for generating stable IPv6 addresses with SLAAC. A link layer MAY also define a mechanism for stable IPv6 address generation that is more efficient and does not address the security and privacy considerations discussed in Section 1. The choice of whether to enable the security- and privacy-preserving mechanism or not SHOULD be configurable in such a case.

By default, nodes SHOULD NOT employ IPv6 address generation schemes that embed a stable link-layer address in the IID. In particular, this document RECOMMENDS that nodes do not generate stable IIDs with the schemes specified in [RFC2464], [RFC2467], [RFC2470], [RFC2491], [RFC2492], [RFC2497], [RFC2590], [RFC3146], [RFC3572], [RFC4338], [RFC4391], [RFC5121], and [RFC5072].

4. Future Work

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

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

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

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

5. IANA Considerations

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

6. Security Considerations

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

7. Acknowledgements

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

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

8. References

8.1. Normative References

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

- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<http://www.rfc-editor.org/info/rfc2460>>.
- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, DOI 10.17487/RFC2464, December 1998, <<http://www.rfc-editor.org/info/rfc2464>>.
- [RFC2467] Crawford, M., "Transmission of IPv6 Packets over FDDI Networks", RFC 2467, DOI 10.17487/RFC2467, December 1998, <<http://www.rfc-editor.org/info/rfc2467>>.
- [RFC2470] Crawford, M., Narten, T., and S. Thomas, "Transmission of IPv6 Packets over Token Ring Networks", RFC 2470, DOI 10.17487/RFC2470, December 1998, <<http://www.rfc-editor.org/info/rfc2470>>.
- [RFC2491] Armitage, G., Schuler, P., Jork, M., and G. Harter, "IPv6 over Non-Broadcast Multiple Access (NBMA) networks", RFC 2491, DOI 10.17487/RFC2491, January 1999, <<http://www.rfc-editor.org/info/rfc2491>>.
- [RFC2492] Armitage, G., Schuler, P., and M. Jork, "IPv6 over ATM Networks", RFC 2492, DOI 10.17487/RFC2492, January 1999, <<http://www.rfc-editor.org/info/rfc2492>>.
- [RFC2497] Souvatzis, I., "Transmission of IPv6 Packets over ARCnet Networks", RFC 2497, DOI 10.17487/RFC2497, January 1999, <<http://www.rfc-editor.org/info/rfc2497>>.
- [RFC2590] Conta, A., Malis, A., and M. Mueller, "Transmission of IPv6 Packets over Frame Relay Networks Specification", RFC 2590, DOI 10.17487/RFC2590, May 1999, <<http://www.rfc-editor.org/info/rfc2590>>.
- [RFC3146] Fujisawa, K. and A. Onoe, "Transmission of IPv6 Packets over IEEE 1394 Networks", RFC 3146, DOI 10.17487/RFC3146, October 2001, <<http://www.rfc-editor.org/info/rfc3146>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<http://www.rfc-editor.org/info/rfc4291>>.
- [RFC4338] DeSanti, C., Carlson, C., and R. Nixon, "Transmission of IPv6, IPv4, and Address Resolution Protocol (ARP) Packets over Fibre Channel", RFC 4338, DOI 10.17487/RFC4338, January 2006, <<http://www.rfc-editor.org/info/rfc4338>>.

- [RFC4391] Chu, J. and V. Kashyap, "Transmission of IP over InfiniBand (IPoIB)", RFC 4391, DOI 10.17487/RFC4391, April 2006, <<http://www.rfc-editor.org/info/rfc4391>>.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <<http://www.rfc-editor.org/info/rfc4862>>.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <<http://www.rfc-editor.org/info/rfc4941>>.
- [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <<http://www.rfc-editor.org/info/rfc4944>>.
- [RFC5072] Varada, S., Ed., Haskins, D., and E. Allen, "IP Version 6 over PPP", RFC 5072, DOI 10.17487/RFC5072, September 2007, <<http://www.rfc-editor.org/info/rfc5072>>.
- [RFC5121] Patil, B., Xia, F., Sarikaya, B., Choi, JH., and S. Madanapalli, "Transmission of IPv6 via the IPv6 Convergence Sublayer over IEEE 802.16 Networks", RFC 5121, DOI 10.17487/RFC5121, February 2008, <<http://www.rfc-editor.org/info/rfc5121>>.
- [RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <<http://www.rfc-editor.org/info/rfc6282>>.
- [RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <<http://www.rfc-editor.org/info/rfc6775>>.

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

8.2. Informative References

- [I-D.gont-predictable-numeric-ids]
Gont, F. and I. Arce, "Security and Privacy Implications of Numeric Identifiers Employed in Network Protocols", draft-gont-predictable-numeric-ids-00 (work in progress), February 2016.
- [Microsoft]
Davies, J., "Understanding IPv6, 3rd. ed", page 83, Microsoft Press, 2012, <<http://it-ebooks.info/book/1022/>>.
- [RFC3572] Ogura, T., Maruyama, M., and T. Yoshida, "Internet Protocol Version 6 over MAPOS (Multiple Access Protocol Over SONET/SDH)", RFC 3572, DOI 10.17487/RFC3572, July 2003, <<http://www.rfc-editor.org/info/rfc3572>>.
- [RFC7721] Cooper, A., Gont, F., and D. Thaler, "Security and Privacy Considerations for IPv6 Address Generation Mechanisms", RFC 7721, DOI 10.17487/RFC7721, March 2016, <<http://www.rfc-editor.org/info/rfc7721>>.

Authors' Addresses

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

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

Alissa Cooper
Cisco
707 Tasman Drive
Milpitas, CA 95035
US

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

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

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

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

Email: liushucheng@huawei.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 26, 2016

A. Cooper
Cisco
F. Gont
Huawei Technologies
D. Thaler
Microsoft
September 23, 2015

Privacy Considerations for IPv6 Address Generation Mechanisms
draft-ietf-6man-ipv6-address-generation-privacy-08.txt

Abstract

This document discusses privacy and security considerations for several IPv6 address generation mechanisms, both standardized and non-standardized. It evaluates how different mechanisms mitigate different threats and the trade-offs that implementors, developers, and users face in choosing different addresses or address generation mechanisms.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 26, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Terminology	4
3. Weaknesses in IEEE-identifier-based IIDs	4
3.1. Correlation of activities over time	5
3.2. Location tracking	6
3.3. Address scanning	6
3.4. Device-specific vulnerability exploitation	7
4. Privacy and security properties of address generation mechanisms	7
4.1. IEEE-identifier-based IIDs	9
4.2. Static, manually configured IIDs	10
4.3. Constant, semantically opaque IIDs	10
4.4. Cryptographically generated IIDs	10
4.5. Stable, semantically opaque IIDs	10
4.6. Temporary IIDs	11
4.7. DHCPv6 generation of IIDs	12
4.8. Transition/co-existence technologies	12
5. Miscellaneous Issues with IPv6 addressing	13
5.1. Network Operation	13
5.2. Compliance	13
5.3. Intellectual Property Rights (IPRs)	13
6. Security Considerations	13
7. IANA Considerations	13
8. Acknowledgements	14
9. References	14
9.1. Normative References	14
9.2. Informative References	15
Authors' Addresses	17

1. Introduction

IPv6 was designed to improve upon IPv4 in many respects, and mechanisms for address assignment were one such area for improvement. In addition to static address assignment and DHCP, stateless autoconfiguration was developed as a less intensive, fate-shared means of performing address assignment. With stateless autoconfiguration, routers advertise on-link prefixes and hosts generate their own interface identifiers (IIDs) to complete their addresses. [RFC7136] clarifies that the IID should be treated as an opaque value, while [RFC7421] provides an analysis of the 64-bit boundary in IPv6 addressing (e.g. the implications of the IID length

on security and privacy). Over the years, many interface identifier generation techniques have been defined, both standardized and non-standardized:

- o Manual configuration
 - * IPv4 address
 - * Service port
 - * Wordy
 - * Low-byte
- o Stateless Address Auto-Configuration (SLAAC)
 - * IEEE 802 48-bit MAC or IEEE EUI-64 identifier [RFC2464]
 - * Cryptographically generated [RFC3972]
 - * Temporary (also known as "privacy addresses") [RFC4941]
 - * Constant, semantically opaque (also known as random) [Microsoft]
 - * Stable, semantically opaque [RFC7217]
- o DHCPv6-based [RFC3315]
- o Specified by transition/co-existence technologies
 - * Derived from an IPv4 address (e.g., [RFC5214], [RFC6052])
 - * Derived from an IPv4 address and port set ID (e.g., [RFC7596], [RFC7597], [RFC7599])
 - * Derived from an IPv4 address and port (e.g., [RFC4380])

Deriving the IID from a globally unique IEEE identifier [RFC2464] [RFC4862] was one of the earliest mechanisms developed (and originally specified in [RFC1971] and [RFC1972]). A number of privacy and security issues related to the IIDs derived from IEEE identifiers were discovered after their standardization, and many of the mechanisms developed later aimed to mitigate some or all of these weaknesses. This document identifies four types of threats against IEEE-identifier-based IIDs, and discusses how other existing techniques for generating IIDs do or do not mitigate those threats.

2. Terminology

This section clarifies the terminology used throughout this document.

Public address:

An address that has been published in a directory or other public location, such as the DNS, a SIP proxy [RFC3261], an application-specific DHT, or a publicly available URI. A host's public addresses are intended to be discoverable by third parties.

Stable address:

An address that does not vary over time within the same IPv6 link. Note that [RFC4941] refers to these as "public" addresses, but "stable" is used here for reasons explained in Section 4.

Temporary address:

An address that varies over time within the same IPv6 link.

Constant IID:

An IPv6 interface identifier that is globally stable. That is, the Interface ID will remain constant even if the node moves from one IPv6 link to another.

Stable IID:

An IPv6 interface identifier that is stable within some specified context. For example, an Interface ID can be globally stable (constant), or could be stable per IPv6 link (meaning that the Interface ID will remain unchanged as long as the node stays on the same IPv6 link, but may change when the node moves from one IPv6 link to another).

Temporary IID:

An IPv6 interface identifier that varies over time.

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 [RFC2119]. These words take their normative meanings only when they are presented in ALL UPPERCASE.

3. Weaknesses in IEEE-identifier-based IIDs

There are a number of privacy and security implications that exist for hosts that use IEEE-identifier-based IIDs. This section discusses four generic attack types: correlation of activities over time, location tracking, address scanning, and device-specific vulnerability exploitation. The first three of these rely on the attacker first gaining knowledge of the IID of the target host. This

could be achieved by a number of different entities: the operator of a server to which the host connects, such as a web server or a peer-to-peer server; an entity that connects to the same IPv6 link as the target (such as a conference network or any public network); a passive observer of traffic that the host broadcasts; or an entity that is on-path to the destinations with which the host communicates, such as a network operator.

3.1. Correlation of activities over time

As with other identifiers, an IPv6 address can be used to correlate the activities of a host for at least as long as the lifetime of the address. The correlation made possible by IEEE-identifier-based IIDs is of particular concern since they last roughly for the lifetime of a device's network interface, allowing correlation on the order of years.

As [RFC4941] explains,

"[t]he use of a non-changing interface identifier to form addresses is a specific instance of the more general case where a constant identifier is reused over an extended period of time and in multiple independent activities. Anytime the same identifier is used in multiple contexts, it becomes possible for that identifier to be used to correlate seemingly unrelated activity. ... The use of a constant identifier within an address is of special concern because addresses are a fundamental requirement of communication and cannot easily be hidden from eavesdroppers and other parties. Even when higher layers encrypt their payloads, addresses in packet headers appear in the clear."

IP addresses are just one example of information that can be used to correlate activities over time. DNS names, cookies [RFC6265], browser fingerprints [Panopticlick], and application-layer usernames can all be used to link a host's activities together. Although IEEE-identifier-based IIDs are likely to last at least as long or longer than these other identifiers, IIDs generated in other ways may have shorter or longer lifetimes than these identifiers depending on how they are generated. Therefore, the extent to which a host's activities can be correlated depends on whether the host uses multiple identifiers together and the lifetimes of all of those identifiers. Frequently refreshing an IPv6 address may not mitigate correlation if an attacker has access to other longer lived identifiers for a particular host. This is an important caveat to keep in mind throughout the discussion of correlation in this document. For further discussion of correlation, see Section 5.2.1 of [RFC6973].

As noted in [RFC4941], in some cases correlation is just as feasible for a host using an IPv4 address as for a host using an IEEE identifier to generate its IID in its IPv6 address. Hosts that use static IPv4 addressing or who are consistently allocated the same address via DHCPv4 can be tracked as described above. However, the widespread use of both NAT and DHCPv4 implementations that assign the same host a different address upon lease expiration mitigates this threat in the IPv4 case as compared to the IEEE identifier case in IPv6.

3.2. Location tracking

Because the IPv6 address structure is divided between a topological portion and an interface identifier portion, an interface identifier that remains constant when a host connects to different IPv6 links (as an IEEE-identifier-based IID does) provides a way for observers to track the movements of that host. In a passive attack on a mobile host, a server that receives connections from the same host over time would be able to determine the host's movements as its prefix changes.

Active attacks are also possible. An attacker that first learns the host's interface identifier by being connected to the same IPv6 link, running a server that the host connects to, or being on-path to the host's communications could subsequently probe other networks for the presence of the same interface identifier by sending a probe packet (ICMPv6 Echo Request, or any other probe packet). Even if the host does not respond, the first hop router will usually respond with an ICMP Destination Unreachable/Address Unreachable (type 1, code 3) when the host is not present, and be silent when the host is present.

Location tracking based on IP address is generally not possible in IPv4 since hosts get assigned wholly new addresses when they change networks.

3.3. Address scanning

The structure of IEEE-based identifiers used for address generation can be leveraged by an attacker to reduce the target search space [I-D.ietf-opsec-ipv6-host-scanning]. The 24-bit Organizationally Unique Identifier (OUI) of MAC addresses, together with the fixed value (0xff, 0xfe) used to form a Modified EUI-64 interface identifier, greatly help to reduce the search space, making it easier for an attacker to scan for individual addresses using widely-known popular OUIs. This erases much of the protection against address scanning that the larger IPv6 address space could provide as compared to IPv4.

3.4. Device-specific vulnerability exploitation

IPv6 addresses that embed IEEE identifiers leak information about the device (Network Interface Card vendor, or even Operating System and/or software type), which could be leveraged by an attacker with knowledge of device/software-specific vulnerabilities to quickly find possible targets. Attackers can exploit vulnerabilities in hosts whose IIDs they have previously obtained, or scan an address space to find potential targets.

4. Privacy and security properties of address generation mechanisms

Analysis of the extent to which a particular host is protected against the threats described in Section 3 depends on how each of a host's addresses is generated and used. In some scenarios, a host configures a single global address and uses it for all communications. In other scenarios, a host configures multiple addresses using different mechanisms and may use any or all of them.

[RFC3041] (later obsoleted by [RFC4941]) sought to address some of the problems described in Section 3 by defining "temporary addresses" for outbound connections. Temporary addresses are meant to supplement the other addresses that a device might use, not to replace them. They use IIDs that are randomly generated and change daily by default. The idea was for temporary addresses to be used for outgoing connections (e.g., web browsing) while maintaining the ability to use a stable address when more address stability is desired (e.g., for IPv6 addresses published in the DNS).

[RFC3484] originally specified that stable addresses be used for outbound connections unless an application explicitly prefers temporary addresses. The default preference for stable addresses was established to avoid applications potentially failing due to the short lifetime of temporary addresses or the possibility of a reverse look-up failure or error. However, [RFC3484] allowed that "implementations for which privacy considerations outweigh these application compatibility concerns MAY reverse the sense of this rule" and instead prefer by default temporary addresses rather than stable addresses. Indeed most implementations (notably including Windows) chose to default to temporary addresses for outbound connections since privacy was considered more important (and few applications supported IPv6 at the time, so application compatibility concerns were minimal). [RFC6724] then obsoleted [RFC3484] and changed the default to match what implementations actually did.

The envisioned relationship in [RFC3484] between stability of an address and its use in "public" can be misleading when conducting privacy analysis. The stability of an address and the extent to

which it is linkable to some other public identifier are independent of one another. For example, there is nothing that prevents a host from publishing a temporary address in a public place, such as the DNS. Publishing both a stable address and a temporary address in the DNS or elsewhere where they can be linked together by a public identifier allows the host's activities when using either address to be correlated together.

Moreover, because temporary addresses were designed to supplement other addresses generated by a host, the host may still configure a more stable address even if it only ever intentionally uses temporary addresses (as source addresses) for communication to off-link destinations. An attacker can probe for the stable address even if it is never used as such a source address or advertised (e.g., in DNS or SIP) outside the link.

This section compares the privacy and security properties of a variety of IID generation mechanisms and their possible usage scenarios, including scenarios in which a single mechanism is used to generate all of a host's IIDs and those in which temporary addresses are used together with addresses generated using a different IID generation mechanism. The analysis of the exposure of each IID type to correlation assumes that IPv6 prefixes are shared by a reasonably large number of nodes. As [RFC4941] notes, if a very small number of nodes (say, only one) use a particular prefix for an extended period of time, the prefix itself can be used to correlate the host's activities regardless of how the IID is generated. For example, [RFC3314] recommends that prefixes be uniquely assigned to mobile handsets where IPv6 is used within GPRS. In cases where this advice is followed and prefixes persist for extended periods of time (or get reassigned to the same handsets whenever those hand sets reconnect to the same network router), hosts' activities could be correlatable for longer periods than the analysis below would suggest.

The table below provides a summary of the whole analysis. A "No" entry indicates that the attack is prevented from being carried out on the basis of the IID, but the host may still be vulnerable depending on how it employs other protocols.

Mechanism(s)	Correlation	Location tracking	Address scanning	Device exploits
IEEE identifier	For device lifetime	For device lifetime	Possible	Possible
Static manual	For address lifetime	For address lifetime	Depends on generation mechanism	Depends on generation mechanism
Constant, semantically opaque	For address lifetime	For address lifetime	No	No
CGA	For lifetime of (modifier block + public key)	No	No	No
Stable, semantically opaque	Within single IPv6 link	No	No	No
Temporary	For temp address lifetime	No	No	No
DHCPv6	For lease lifetime	No	Depends on generation mechanism	No

Table 1: Privacy and security properties of IID generation mechanisms

4.1. IEEE-identifier-based IIDs

As discussed in Section 3, addresses that use IIDs based on IEEE identifiers are vulnerable to all four threats. They allow correlation and location tracking for the lifetime of the device since IEEE identifiers last that long and their structure makes address scanning and device exploits possible.

4.2. Static, manually configured IIDs

Because static, manually configured IIDs are stable, both correlation and location tracking are possible for the life of the address.

The extent to which location tracking can be successfully performed depends, to a some extent, on the uniqueness of the employed IID. For example, one would expect "low byte" IIDs to be more widely reused than, for example, IIDs where the whole 64-bits follow some pattern that is unique to a specific organization. Widely reused IIDs will typically lead to false positives when performing location tracking.

Whether manually configured addresses are vulnerable to address scanning and device exploits depends on the specifics of how the IIDs are generated.

4.3. Constant, semantically opaque IIDs

Although a mechanism to generate a constant, semantically opaque IID has not been standardized, it has been in wide use for many years on at least one platform (Windows). Windows uses the [RFC4941] random generation mechanism in lieu of generating an IEEE-identifier-based IID. This mitigates the device-specific exploitation and address scanning attacks, but still allows correlation and location tracking because the IID is constant across IPv6 links and time.

4.4. Cryptographically generated IIDs

Cryptographically generated addresses (CGAs) [RFC3972] bind a hash of the host's public key to an IPv6 address in the SEcure Neighbor Discovery (SEND) [RFC3971] protocol. CGAs may be regenerated for each subnet prefix, but this is not required given that they are computationally expensive to generate. A host using a CGA can be correlated for as long as the lifetime of the combination of the public key and the chosen modifier block, since it is possible to rotate modifier blocks without generating new public keys. Because the cryptographic hash of the host's public key uses the subnet prefix as an input, even if the host does not generate a new public key or modifier block when it moves to a different IPv6 link, its location cannot be tracked via the IID. CGAs do not allow device-specific exploitation or address scanning attacks.

4.5. Stable, semantically opaque IIDs

[RFC7217] specifies an algorithm that generates, for each network interface, a unique random IID per IPv6 link. The aforementioned algorithm is employed not only for global unicast addresses, but also

for unique local unicast addresses and link-local unicast addresses, since these addresses may leak out via application protocols (e.g., IPv6 addresses embedded in email headers).

A host that stays connected to the same IPv6 link could therefore be tracked at length, whereas a mobile host's activities could only be correlated for the duration of each network connection. Location tracking is not possible with these addresses. They also do not allow device-specific exploitation or address scanning attacks.

4.6. Temporary IIDs

A host that uses only a temporary address mitigates all four threats. Its activities may only be correlated for the lifetime a single temporary address.

A host that configures both an IEEE-identifier-based IID and temporary addresses makes the host vulnerable to the same attacks as if temporary addresses were not in use, although the viability of some of them depends on how the host uses each address. An attacker can correlate all of the host's activities for which it uses its IEEE-identifier-based IID. Once an attacker has obtained the IEEE-identifier-based IID, location tracking becomes possible on other IPv6 links even if the host only makes use of temporary addresses on those other IPv6 links; the attacker can actively probe the other IPv6 links for the presence of the IEEE-identifier-based IID. Device-specific vulnerabilities can still be exploited. Address scanning is also still possible because the IEEE-identifier-based address can be probed.

If the host instead generates a constant, semantically opaque IID to use in a stable address for server-like connections together with temporary addresses for outbound connections (as is the default in Windows), it sees some improvements over the previous scenario. The address scanning and device-specific exploitation attacks are no longer possible because the OUI is no longer embedded in any of the host's addresses. However, correlation of some activities across time and location tracking are both still possible because the semantically opaque IID is constant. And once an attacker has obtained the host's semantically opaque IID, location tracking is possible on any network by probing for that IID, even if the host only uses temporary addresses on those networks. However, if the host generates but never uses a constant, semantically opaque IID, it mitigates all four threats.

When used together with temporary addresses, the stable, semantically opaque IID generation mechanism [RFC7217] improves upon the previous scenario by limiting the potential for correlation to the lifetime of

the stable address (which may still be lengthy for hosts that are not mobile) and by eliminating the possibility for location tracking (since a different IID is generated for each subnet prefix). As in the previous scenario, a host that configures but does not use a stable, semantically opaque address mitigates all four threats.

4.7. DHCPv6 generation of IIDs

The security/privacy implications of DHCPv6-based addresses will typically depend on whether the client requests an IA_NA (Identity Association for Non-temporary Addresses) or an IA_TA (Identity Association for Temporary Addresses) [RFC3315] and the specific DHCPv6 server software being employed.

DHCPv6 temporary addresses have the same properties as SLAAC temporary addresses Section 4.6 [RFC4941]. On the other hand, the properties of DHCPv6 non-temporary addresses typically depend on the specific DHCPv6 server software being employed. Recent releases of most popular DHCPv6 server software typically lease random addresses with a similar lease time as that of IPv4. Thus, these addresses can be considered to be "stable, semantically opaque". [I-D.ietf-dhc-stable-privacy-addresses] specifies an algorithm that can be employed by DHCPv6 servers to generate "stable, semantically opaque" addresses.

On the other hand, some DHCPv6 software leases sequential addresses (typically low-byte addresses). These addresses can be considered to be stable addresses. The drawback of this address generation scheme compared to "stable, semantically opaque" addresses is that, since they follow specific patterns, they enable IPv6 address scans.

4.8. Transition/co-existence technologies

Addresses specified based on transition/co-existence technologies that embed an IPv4 address within an IPv6 address are not included in Table 1 because their privacy and security properties are inherited from the embedded address. For example, Teredo [RFC4380] specifies a means to generate an IPv6 address from the underlying IPv4 address and port, leaving many other bits set to zero. This makes it relatively easy for an attacker to scan for IPv6 addresses by guessing the Teredo client's IPv4 address and port (which for many NATs is not randomized). For this reason, popular implementations (e.g., Windows), began deviating from the standard by including 12 random bits in place of zero bits. This modification was later standardized in [RFC5991].

Some other transition technologies (e.g., [RFC5214], [RFC6052]) specify means to generate an IPv6 address from an underlying IPv4

address without a port. Such mechanisms thus make it much easier for an attacker to conduct an address scan than for mechanisms that require finding a port number as well.

Finally, still other mechanisms (e.g., [RFC7596], [RFC7597], [RFC7599]) are somewhere in between, using an IPv4 address and a port set ID (which for many NATs is not randomized). In general, such mechanisms are thus typically as easy to scan as in the Teredo example above without the 12-bit mitigation.

5. Miscellaneous Issues with IPv6 addressing

5.1. Network Operation

It is generally agreed that IPv6 addresses that vary over time in a specific IPv6 link tend to increase the complexity of event logging, trouble-shooting, enforcement of access controls and quality of service, etc. As a result, some organizations disable the use of temporary addresses [RFC4941] even at the expense of reduced privacy [Broersma].

5.2. Compliance

Some IPv6 compliance testing suites required (and might still require) implementations to support IEEE-identifier-based IIDS in order to be approved as compliant. This document recommends that compliance testing suites be relaxed to allow other forms of address generation that are more amenable to privacy.

5.3. Intellectual Property Rights (IPRs)

Some IPv6 addressing techniques might be covered by Intellectual Property rights, which might limit their implementation in different Operating Systems. [CGA-IPR] and [KAME-CGA] discuss the IPRs on CGAs.

6. Security Considerations

This whole document concerns the privacy and security properties of different IPv6 address generation mechanisms.

7. IANA Considerations

This document does not require actions by IANA.

8. Acknowledgements

The authors would like to thank Bernard Aboba, Brian Carpenter, Tim Chown, Lorenzo Colitti, Rich Draves, Robert Hinden, Robert Moskowitz, Erik Nordmark, Mark Smith, Ole Troan, and James Woodyatt for providing valuable comments on earlier versions of this document.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, DOI 10.17487/RFC2464, December 1998, <<http://www.rfc-editor.org/info/rfc2464>>.
- [RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <<http://www.rfc-editor.org/info/rfc3315>>.
- [RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <<http://www.rfc-editor.org/info/rfc3971>>.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <<http://www.rfc-editor.org/info/rfc3972>>.
- [RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, DOI 10.17487/RFC4380, February 2006, <<http://www.rfc-editor.org/info/rfc4380>>.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <<http://www.rfc-editor.org/info/rfc4862>>.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <<http://www.rfc-editor.org/info/rfc4941>>.

- [RFC5991] Thaler, D., Krishnan, S., and J. Hoagland, "Teredo Security Updates", RFC 5991, DOI 10.17487/RFC5991, September 2010, <<http://www.rfc-editor.org/info/rfc5991>>.
- [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, <<http://www.rfc-editor.org/info/rfc6724>>.
- [RFC7136] Carpenter, B. and S. Jiang, "Significance of IPv6 Interface Identifiers", RFC 7136, DOI 10.17487/RFC7136, February 2014, <<http://www.rfc-editor.org/info/rfc7136>>.
- [RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <<http://www.rfc-editor.org/info/rfc7217>>.

9.2. Informative References

- [Broersma] Broersma, R., "IPv6 Everywhere: Living with a Fully IPv6-enabled environment", Australian IPv6 Summit 2010, Melbourne, VIC Australia, October 2010, October 2010, <http://www.ipv6.org.au/10ipv6summit/talks/Ron_Broersma.pdf>.
- [CGA-IPR] IETF, "Intellectual Property Rights on RFC 3972", 2005, <<https://datatracker.ietf.org/ipr/676/>>.
- [I-D.ietf-dhc-stable-privacy-addresses] Gont, F. and S. LIU, "A Method for Generating Semantically Opaque Interface Identifiers with Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", draft-ietf-dhc-stable-privacy-addresses-02 (work in progress), April 2015.
- [I-D.ietf-opsec-ipv6-host-scanning] Gont, F. and T. Chown, "Network Reconnaissance in IPv6 Networks", draft-ietf-opsec-ipv6-host-scanning-08 (work in progress), August 2015.
- [KAME-CGA] KAME, "The KAME IPR policy and concerns of some technologies which have IPR claims", 2005, <<http://www.kame.net/newsletter/20040525/>>.

- [Microsoft] Microsoft, "IPv6 interface identifiers", 2013, <target='http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/sag_ip_v6_imp_addr7.mspx?mfr=true>.
- [Panopticlick] Electronic Frontier Foundation, "Panopticlick", 2011, <http://panopticlick.eff.org>.
- [RFC1971] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 1971, DOI 10.17487/RFC1971, August 1996, <http://www.rfc-editor.org/info/rfc1971>.
- [RFC1972] Crawford, M., "A Method for the Transmission of IPv6 Packets over Ethernet Networks", RFC 1972, DOI 10.17487/RFC1972, August 1996, <http://www.rfc-editor.org/info/rfc1972>.
- [RFC3041] Narten, T. and R. Draves, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 3041, DOI 10.17487/RFC3041, January 2001, <http://www.rfc-editor.org/info/rfc3041>.
- [RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <http://www.rfc-editor.org/info/rfc3261>.
- [RFC3314] Wasserman, M., Ed., "Recommendations for IPv6 in Third Generation Partnership Project (3GPP) Standards", RFC 3314, DOI 10.17487/RFC3314, September 2002, <http://www.rfc-editor.org/info/rfc3314>.
- [RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, DOI 10.17487/RFC3484, February 2003, <http://www.rfc-editor.org/info/rfc3484>.
- [RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, DOI 10.17487/RFC5214, March 2008, <http://www.rfc-editor.org/info/rfc5214>.
- [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, <http://www.rfc-editor.org/info/rfc6052>.

- [RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, DOI 10.17487/RFC6265, April 2011, <<http://www.rfc-editor.org/info/rfc6265>>.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <<http://www.rfc-editor.org/info/rfc6973>>.
- [RFC7421] Carpenter, B., Ed., Chown, T., Gont, F., Jiang, S., Petrescu, A., and A. Yourtchenko, "Analysis of the 64-bit Boundary in IPv6 Addressing", RFC 7421, DOI 10.17487/RFC7421, January 2015, <<http://www.rfc-editor.org/info/rfc7421>>.
- [RFC7596] Cui, Y., Sun, Q., Boucadair, M., Tsou, T., Lee, Y., and I. Farrer, "Lightweight 4over6: An Extension to the Dual-Stack Lite Architecture", RFC 7596, DOI 10.17487/RFC7596, July 2015, <<http://www.rfc-editor.org/info/rfc7596>>.
- [RFC7597] Troan, O., Ed., Dec, W., Li, X., Bao, C., Matsushima, S., Murakami, T., and T. Taylor, Ed., "Mapping of Address and Port with Encapsulation (MAP-E)", RFC 7597, DOI 10.17487/RFC7597, July 2015, <<http://www.rfc-editor.org/info/rfc7597>>.
- [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, <<http://www.rfc-editor.org/info/rfc7599>>.

Authors' Addresses

Alissa Cooper
Cisco
707 Tasman Drive
Milpitas, CA 95035
US

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

Fernando Gont
Huawei Technologies
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

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

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

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

6man Working Group
Internet-Draft
Updates: 4861 (if approved)
Intended status: Standards Track
Expires: October 11, 2015

S. Krishnan
Ericsson
D. Anipko
Unaffiliated
D. Thaler
Microsoft
April 9, 2015

Packet loss resiliency for Router Solicitations
draft-ietf-6man-resilient-rs-06

Abstract

When an interface on a host is initialized, the host transmits Router Solicitations in order to minimize the amount of time it needs to wait until the next unsolicited multicast Router Advertisement is received. In certain scenarios, these router solicitations transmitted by the host might be lost. This document specifies a mechanism for hosts to cope with the loss of the initial Router Solicitations.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 11, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Conventions used in this document	2
2. Proposed algorithm	4
2.1. Stopping the retransmissions	4
3. Configuring the use of retransmissions	5
4. Known Limitations	5
5. IANA Considerations	5
6. Security Considerations	5
7. Acknowledgements	5
8. References	5
8.1. Normative References	6
8.2. Informative References	6
Authors' Addresses	6

1. Introduction

As specified in [RFC4861], when an interface on a host is initialized, in order to obtain Router Advertisements quickly, a host transmits up to MAX_RTR_SOLICITATIONS (3) Router Solicitation messages, each separated by at least RTR_SOLICITATION_INTERVAL (4) seconds. In certain scenarios, these router solicitations transmitted by the host might be lost. e.g. The host is connected to a bridged residential gateway over Ethernet or WiFi. LAN connectivity is achieved at interface initialization, but the upstream WAN connectivity is not active yet. In this case, the host just gives up after the initial RS retransmits.

Once the initial RSs are lost, the host gives up and assumes that there are no routers on the link as specified in Section 6.3.7 of [RFC4861]. The host will not have any form of Internet connectivity until the next unsolicited multicast Router Advertisement is received. These Router Advertisements are transmitted at most MaxRtrAdvInterval seconds apart (maximum value 1800 seconds). Thus in the worst case scenario a host would be without any connectivity for 30 minutes. This delay may be unacceptable in some scenarios.

1.1. Conventions used in this document

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 [RFC2119].

2. Proposed algorithm

To achieve resiliency to packet loss, the host needs to continue retransmitting the Router Solicitations until it receives a Router Advertisement, or until it is willing to accept that no router exists. If the host continues retransmitting the RSs at RTR_SOLICITATION_INTERVAL second intervals, it may cause excessive network traffic if a large number of such hosts exists. To achieve resiliency while keeping the aggregate network traffic low, the host can use some form of exponential backoff algorithm to retransmit the RSs.

Hosts complying to this specification MUST use the exponential backoff algorithm for retransmits that is described in Section 14 of [RFC3315] in order to continuously retransmit the Router Solicitations until a Router Advertisement is received. The hosts SHOULD use the following variables as input to the retransmission algorithm:

IRT 4 seconds

MRT 3600 seconds

MRC 0

MRD 0

The initial value IRT was chosen to be in line with the current retransmission interval (RTR_SOLICITATION_INTERVAL) that is specified by [RFC4861] and the maximum retransmission time MRT was chosen to be in line with the new value of SOL_MAX_RT as specified by [RFC7083]. This is to ensure that the short term behavior of the RSs is similar to what is experienced in current networks, and longer term persistent retransmission behavior trends towards being similar to that of DHCPv6 [RFC3315] [RFC7083].

2.1. Stopping the retransmissions

On multicast-capable links, the hosts following this specification SHOULD stop retransmitting the RSs when Router Discovery is successful (i.e. an RA with a non-zero Router Lifetime that results in a default route is received). If an RA is received from a router and it does not result in a default route (i.e. Router Lifetime is zero) the host MUST continue retransmitting the RSs.

On non-multicast links, the hosts following this specification MUST continue retransmitting the RSs even after an RA that results in a default route is received. This is required because, in such links,

sending an RA can only be triggered by an RS. Please note that such links have special mechanisms for sending RSes as well. e.g. The mechanism specified in Section 8.3.4. of ISATAP [RFC5214] unicasts the RSes to specific routers.

3. Configuring the use of retransmissions

Implementations of this specification are encouraged to provide a configuration option to enable or disable potentially infinite RS retransmissions. If a configuration option is provided, it **MUST** enable RS retransmissions by default. Providing an option to enable/disable retransmissions on a per-interface basis allows network operators to configure RS behavior most applicable to each connected link.

4. Known Limitations

When an IPv6-capable host attaches to a network that does not have IPv6 enabled, it transmits 3 (MAX_RTR_SOLICITATIONS) Router Solicitations as specified in [RFC4861]. If it receives no Router Advertisements, it assumes that there are no routers present on the link and it ceases to send further RSs. With the mechanism specified in this document, the host will continue to retransmit RSs indefinitely at the rate of approximately 1 RS per hour. It is unclear how to differentiate between such a network with no IPv6 routers and a link where an IPv6 router is temporarily unreachable but could become reachable in the future.

5. IANA Considerations

This document does not require any IANA actions.

6. Security Considerations

This document does not present any additional security issues beyond those discussed in [RFC4861] and those RFCs that update [RFC4861].

7. Acknowledgements

The authors would like to thank Steve Baillargeon, Erik Kline, Andrew Yourtchenko, Ole Troan, Erik Nordmark, Lorenzo Colitti, Thomas Narten, Ran Atkinson, Allison Mankin, Les Ginsberg, Brian Carpenter, Barry Leiba, Brian Haberman, Spencer Dawkins, Alia Atlas, Stephen Farrell and Mehmet Ersue for their reviews and suggestions that made this document better.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC7083] Droms, R., "Modification to Default Values of SOL_MAX_RT and INF_MAX_RT", RFC 7083, November 2013.

8.2. Informative References

- [RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, March 2008.

Authors' Addresses

Suresh Krishnan
Ericsson
8400 Decarie Blvd.
Town of Mount Royal, QC
Canada

Phone: +1 514 345 7900 x42871
Email: suresh.krishnan@ericsson.com

Dmitry Anipko
Unaffiliated

Phone: +1 425 442 6356
Email: dmitry.anipko@gmail.com

Dave Thaler
Microsoft
One Microsoft Way
Redmond, WA
USA

Email: dthaler@microsoft.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: August 18, 2014

E. Vyncke, Ed.
P. Thubert
E. Levy-Abegnoli
A. Yourtchenko
Cisco
February 14, 2014

Why Network-Layer Multicast is Not Always Efficient At Datalink Layer
draft-vyncke-6man-mcast-not-efficient-01

Abstract

Several IETF protocols (IPv6 Neighbor Discovery for example) rely on IP multicast in the hope to be efficient with respect to available bandwidth and to avoid generating interrupts in the network nodes. On some datalink-layer network, for example IEEE 802.11 WiFi, this is not the case because of some limitations in the services offered by the datalink-layer network. This document lists and explains all the potential issues when using network-layer multicast over some datalink-layer networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 18, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Issue on Wired Ethernet Network	3
3. Issues on IEEE 802.11 Wireless Network	4
3.1. Multicast over Wireless	4
3.2. Host Sleep Mode	6
3.3. Low Power WiFi Clients	7
3.4. Vendor and Configuration Optimizations	8
3.5. Even Unicast NDP is not Optimum	8
4. Measuring the Amount of IPv6 Multicast	9
5. Acknowledgements	9
6. IANA Considerations	9
7. Security Considerations	9
8. Informative References	9
Authors' Addresses	10

1. Introduction

Several IETF protocols rely on the use of link-local scoped IP multicast in the hope of reducing traffic over the underlying datalink network and generating less operating systems interrupts for the receiving nodes. For example, IPv6 Neighbor Discovery [RFC4861] uses link-local multicast to:

- o advertise the presence of a router by sending router advertisement to IPv6 address link-local multicast address (LLMA), ff02::1, whose members are only the IPv6 nodes but per [RFC4291] section 3 those messages must be forwarded on all ports. This IPv6 LLMA is mapped to the Ethernet Multicast Address (EMA) 33:33:00:00:00:01;
- o solicit the data-link layer address of an adjacent on-link node by sending a neighbor solicitation to the solicited-node multicast address corresponding to the target address such as ff02:0:0:0:0:1:ffXX:XXXX (where the last 24 bits are the last 24 bits of the target address) as described in [RFC4291]. This IPv6 LLMA is mapped to the EMA 33:33:ff:XX:XX:XX.

2. Issue on Wired Ethernet Network

Most switch vendors implement MLD snooping [RFC4541] in order to forward multicast frames only to switch ports where there is a member of the IPv6 multicast group. This optimization works by installing hardware forwarding states in the switch. As there is a finite amount of memory in the switches, especially when the memory is used by the data plane forwarding, there is also a limit to the number of MLD optimization states i.e. a limit to the number of IPv6 multicast groups that can be optimized by the switch; frames destined to groups without such a state are flooded on all ports in the same datalink domain, and generally the use of MLD snooping is reserved to groups with a scope wider than link local.

With IPv6, all nodes have usually at least two IPv6 addresses: a link-local and a global address. If both addresses are based on EUI-64, then they share the same 24 least-significant bits, hence there is only one solicited-node multicast address per node. Else, there is a high probability that the 24 least-significant bits are different, hence requiring the membership to two solicited-node multicast addresses. If a switch uses MLD snooping to install hardware-optimized multicast forwarding states for LLMA, then the switch installs two hardware-optimized states per node as EUI-64 addresses are no more commonly used. If privacy extension addresses [RFC4941] are used, then every node can have multiple IPv6 global addresses, most of which are not based on EUI-64, a large switch fabric will have to support multiple times more states for multicast EMA than it does for unicast addresses, resulting in an excessive amount of resources in each individual switch to be built at an affordable price.

Therefore, due to cost reason, the multicast optimization by MLD snooping of solicited-node LLMA is disabled on most Ethernet switches. This means wasting:

- o the switch bandwidth as it works as a full-duplex hub;
- o the nodes CPU as all nodes will have to receive the multicast frame (if their network adapter is not optimized to support MAC multicast) and quickly drop it.

A special mention must be paid when a layer-2 domain includes legacy devices working on at 10 Mbps half-duplex; for example, in hospitals having old equipments dated back of 1990. For this case, it takes only 100 300-byte frames per second to already utilize the media to 2.4 % not to mention that the NIC and the processor have to process those frames and that the processor is probably also dated from 1990...

It is unclear what the impact is on virtual machines with different MAC addresses and different IPv6 address connected with a virtual layer-2 switch hosted on a single physical server... The MLD snooping done by the virtual switch will consume CPU by the hypervisor, hence, also reducing the amount of CPU available for the virtual machines.

Leveraging MLD snooping to save layer-2 switches from flooding link-local multicast messages carries additional challenges. Unsolicited MLD reports are usually sent once (when link comes up) and not acknowledged. There exist a retransmission mechanism, but it is not generally deployed, and it does not guarantee that subsequent retransmission won't also get lost. The switch could easily end up with incomplete forwarding states for a given group, with some of the listeners ports, but not all (much worse than no state at all). As the switch does not know one of its forwarding entry is incomplete, it can't fall back to broadcasting. As ordinary MLD routers, the switch could query reports on a periodic basis. However, it is not practical for layer-2 access switches to send periodic general MLD queries to maintain forwarding states accuracy for at least 2 reasons:

- o The queries must be sourced with a link-local IPv6 address, one per link, and, for many practical reasons, layer-2 switches don't have such address on each link (vlan) they operate on.
- o Since address resolution uses a multicast group, and may happen quite frequently on the link, in order to avoid black holing resolution, the interval for a switch to issue MLD general query would have to be very small (a few seconds). These MLD queries are themselves sent to a multicast group that all nodes would need to get. That would completely defeat the purpose of reducing multicast traffic towards end nodes.

3. Issues on IEEE 802.11 Wireless Network

3.1. Multicast over Wireless

Wireless networks are a shared half-duplex media: when one station transmits, then all others must be silent. A multicast or broadcast transmission from an AP is physically transmitted to all WiFi clients (STAs) and no other node can use the wireless medium at that time. This is the first issue with the use of wireless for multicast: the medium access behaves as a Ethernet hub.

Depending on distance and radio propagation, different wireless clients may use different transmission encodings and data rates. A lower data rate effectively locks the medium for a longer time per bit. In order to reach all nodes, and considering that multicast and

broadcast frames are not protected by ARQ (retries), the AP is constrained to transmit all multicast or broadcast frames at the lowest rate possible, which in practice is often translated to rates as low as 1 Mbps or 6 Mbps, even when the unicast rate can reach a hundred of Mbps and above. It results that sending a single multicast frame can consume as much bandwidth as dozens of unicast frames. Table 1 provides some example values of the bandwidth used by multicast frames transmitted from the AP (i.e. not counting the original multicast frame transmitted by the WiFi client to the AP when the source is effectively wireless).

Lowest WiFi rate	Highest WiFi rate	Mcast frame %age	WiFi Utilization by Mcast
1 Mbps	11 Mbps	1 %	9 %
6 Mbps	54 Mbps	1 %	9 %
6 Mbps	54 Mbps	5 %	45 %
6 Mbps	54 Mbps	10 %	90 %

Table 1: Multicast WiFi Usage

If multiple APs cover the same wireless LAN, then the multicast frames must be transmitted by all APs to all their WiFi clients.

Communication of a multicast frame by a WiFi client requires three steps:

1. The WiFi client sends a datalink unicast frame to the AP at its maximum possible rate.
2. The WiFi AP forwards this frame on its wired interface and broadcasts it (as explained above) to all its WiFi clients. If there are multiple APs on the same datalink domain, then, all APs also broadcast this multicast frame to their WiFi clients.
3. A WiFi NIC that implements the STA in the client filters the frames that are effectively expected by this device based on destination address.

Another side effect of multicast frames is that there cannot be an acknowledgement mechanism (ARQ) similar to that used for unicast frame, therefore frames can be missed and NDP does not take this non negligible packet loss into account. This could have a negative impact for Duplicate Address Detection (DAD) if the multicast NS or the multicast NA with override are lost. Assuming a error rate of 8%

of corrupted frame, this means a 8% chance of losing a complete frame, this means a 16% chance of not detecting a duplicate address.

For a well-distributed multicast group where relatively few devices actually participate to any given group, there should be no transmission at all if none of the clients expects the multicast destination address, and there should be very few unicast but fast transmissions to the limited set of interest STAs when there is effectively a match in the set of associated devices. But there is no mechanism in place to ensure that functionality.

3.2. Host Sleep Mode

When a sleeping host wakes up by a user interaction, it cannot determine whether it has moved to another network (SSID are not unique), hence, it has to send a multicast Router Solicitation (which triggers a Router Advertisement message from all adjacent routers) and the mobile host has to do Duplicate Address Detection for its link-local and global addresses, thus means transmitting at least two multicast Neighbour Solicitation messages which will be repeated by the AP to all other WiFi clients.

This process creates a lot of multicast packets:

- o one multicast Router Solicitation from the WiFi client, which is received by the AP and if the AP is not optimized, then the Router Solicitation is broadcasted again over the wireless link;
- o one multicast Neighbor Solicitation for the host LLA from the WiFi client, which is received by the AP and if the AP is not optimized, the message is transmitted back over the wireless link;
- o per global address (usually 1 or 2 depending on whether privacy extension is active), same behavior as above.

In conclusion and in the good case of not having privacy extension, this means 6 WiFi broadcast packets plus the unicast replies on each wake-up of the device. Assuming a packet size of 80 bytes, this translates into about 120 bytes to take into account the WiFi frame format which is larger than the usual Ethernet frame, the table Table 2 gives some result of the WiFi utilization just for the multicast part of the wake-up of sleeping devices... This does not take into account the rest of the multicast utilization used by RS, RA, NS, NA, MLD, ... and the associated unicast traffic.

WiFi Clients	Wake-up Cycle	Mcast packet/sec	Mcast bit/sec	Lowest WiFi Rate	Mcast Utilization
100	600 sec	1	960 bps	1 Mbps	0.1 %
1 000	600 sec	1	9600 bps	1 Mbps	1.0 %
5 000	600 sec	50	48 kbps	1 Mbps	4.8 %
5 000	300 sec	100	96 kbps	1 Mbps	9.6 %

Table 2: Multicast WiFi Usage by Sleeping Devices

3.3. Low Power WiFi Clients

In order to save their batteries, Low Power (LP) hosts go into radio sleep mode until there is a local need to send a wireless frame. Before going into radio sleep mode, the LP hosts signal to the AP that they are going into sleep; this allows the AP to store unicast and multicast frames destined for those sleeping LP clients. LP clients wake up periodically to listen to the WiFi beacon frames transmitted periodically (default every 100 ms) because this beacon frame contains a bit mask (Traffic Indication Map - TIM) indicating for which STA there is waiting unicast traffic and whether there is multicast traffic waiting. If there is multicast traffic waiting, that ALL LP hosts must stay awake to receive all multicast frames sent immediately after by the AP and process them. If there is a bit indicating that unicast traffic is waiting for a specific LP host, then only this LP host will stay awake to poll the AP later to collect its traffic. The TIM maximum length is 2008 bits and the complete beacon frame is less than 300 bytes long.

The table Table 2 indicates the ration of active/sleeping time for LP hosts when multicast is present. In the absence of multicast traffic, the radio is active only 2.4 % of the time while if there are 50 multicast frames of 300 bytes per second, the radio is active 14.4 % of the time, nearly 6 times more often... with a battery life probably reduced by 6...

Beacon frames/sec	Mcast frames/sec	Mcast frame size (bytes)	Lowest WiFi Rate	Awake time/sec
10	0	300 bytes	1 Mbps	2.4 %
10	5	300 bytes	1 Mbps	3.6 %
10	10	300 bytes	1 Mbps	4.8 %
10	50	300 bytes	1 Mbps	14.4 %

Table 3: Multicast WiFi Impact on Low Power Hosts

3.4. Vendor and Configuration Optimizations

Vendors have noticed the problem and have come with several optimizations such as

- o LP hosts not waking up the main processor when they are not member of the multicast group;
- o APs no transmitting back over radio received Router Solicitation multicast messages;
- o ...

AP can also work in 'AP isolation mode' where there is no direct traffic between WiFi clients, this mode has a positive side-effect when a WiFi client transmits a multicast frame as this frame is transmitted at the highest possible rate over the WiFi medium and the AP will not re-transmit it back to all other WiFi clients at the lowest rate.

3.5. Even Unicast NDP is not Optimum

While this is not directly related to the subject of this document, it is worth mentioning anyway as this is important for devices running on battery.

NDP cache needs to be maintained by refreshing the neighbor cache for entries which are in the STALE state. This requires yet another Neighbor Solicitation / Neighbor Advertisement round. Even if the destination IP and MAC addresses are unicast, this traffic is generated and again wakes up mobile devices.

4. Measuring the Amount of IPv6 Multicast

There are basically three ways to measure the amount of IPv6 multicast traffic:

- o sniffing the traffic and generating statistics, somehow an overkill:
- o exporting IPfix data and doing aggregation on the ff02::/16 link-local multicast prefix
- o using SNMP to query on the AP the IP-MIB [RFC4293] with commands such as:
 - * `snmpwalk -c private -v 1 udp6:[2001:db8::1] -Ci -m IP-MIB ifDesc:` to get the interface names and index;
 - * `snmpwalk -c private -v 1 udp6:[2001:db8::1] -Ci -m IP-MIB ipIfStatsOutTransmits.ipv6:` to get the global transmit counters (i.e. unicast and multicast as there is no broadcast in IPv6);
 - * `snmpwalk -c private -v 1 udp6:[2001:db8::1] -Ci -m IP-MIB ipIfStatsOutMcastPkts.ipv6:` to get the multicast packet counter.

5. Acknowledgements

The authors would like to thank Norman Finn, Michel Fontaine, Steve Simlo, Ole Troan, and Stig Venaas for their suggestions and comments.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

The only security considerations about this document is that by forcing a lot of traffic to be multicast, then, a denial of service (DoS) attack could be mounted on available bandwidth and battery of some network nodes.

8. Informative References

- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4293] Routhier, S., "Management Information Base for the Internet Protocol (IP)", RFC 4293, April 2006.

- [RFC4541] Christensen, M., Kimball, K., and F. Solensky,
"Considerations for Internet Group Management Protocol
(IGMP) and Multicast Listener Discovery (MLD) Snooping
Switches", RFC 4541, May 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman,
"Neighbor Discovery for IP version 6 (IPv6)", RFC 4861,
September 2007.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy
Extensions for Stateless Address Autoconfiguration in
IPv6", RFC 4941, September 2007.
- [packet_loss]
Department of Computer Sciences, University of Wisconsin
Madison, USA, "Diagnosing Wireless Packet Losses in
802.11: Separating Collision from Weak Signal",
<<http://pages.cs.wisc.edu/~suman/pubs/diagnose.pdf>>.

Authors' Addresses

Eric Vyncke (editor)
Cisco
De Kleetlaan, 6A
Diegem 1831
BE

Phone: +32 2 778 4677
Email: evyncke@cisco.com

Pascal Thubert
Cisco
Batiment D, 45 Allee des Ormes
MOUGINS, PROVENCE-ALPES-COTE D'AZUR 06250
France

Email: pthubert@cisco.com

Eric Levy-Abegnoli
Cisco
Batiment D, 45 Allee des Ormes
MOUGINS, PROVENCE-ALPES-COTE D'AZUR 06250
France

Email: elevyabe@cisco.com

Andrew Yourtchenko
Cisco
De Kleetlaan, 6A
Diegem 1831
BE

Phone: +32 2 704 5494
Email: ayourtch@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 18, 2014

A. Yourtchenko
cisco
L. Colitti
Google
February 14, 2014

Reducing Multicast in IPv6 Neighbor Discovery
draft-yourtchenko-colitti-nd-reduce-multicast-00

Abstract

IPv6 Neighbor Discovery protocol makes wide use of multicast traffic, which makes it not energy efficient for the mobile WiFi hosts. This document describes two classes of possible ways to reduce the multicast traffic within IPv6 ND. First, within the boundaries of existing protocols. Second - with what the authors deem to be "minor changes" to the existing protocols.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 18, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Impact of Multicast Packets in 802.11 Networks	3
3. Quantifying the use of Multicast in Neighbor Discovery	4
4. Multicast-limiting measures with no changes in specifications	4
4.1. On-device robust multicast filtering	5
4.2. Unicast Solicited Router Advertisements	5
4.3. Infrastructure-based multicast filtering	5
4.4. Proxy the Neighbor Discovery protocol on the access point	6
4.5. Maximized Interval for Periodic RAs	6
4.6. Increasing the advertised Reachable value	7
4.7. Clearing the on-link bit in the advertized prefixes	7
4.8. Explicit creation of state with DHCPv6 address assignment	8
4.9. Client link shutdown within the router lifetime expiry	8
5. Multicast-limiting measures with small changes in specifications	8
5.1. Remove the send-side limit on AdvDefaultLifetime of 9000 Seconds	8
5.2. Explicitly Client-Driven Router Advertisements	9
6. Acknowledgements	9
7. IANA Considerations	10
8. Security Considerations	10
9. Normative References	10
Authors' Addresses	10

1. Introduction

Wireless networks based on the IEEE 802.11 standard (WiFi) are ubiquitous in today's life. The multicast/broadcast behavior in these networks has significantly lower performance than unicast in the majority of the cases.

Also, in the current standard and implementations of the 802.11 protocols from the link-layer media standpoint the multicast is the same as broadcast.

The Neighbor Discovery protocol makes substantial use of multicast packets on the assumption that they provide the same or better efficiency compared to unicast packets.

This misalignment results that the nodes on IPv6 networks with the default configuration perform significantly poorer both from the battery life standpoint and the bandwidth efficiency standpoint.

This document presents two groups of measures which reduce the shortcoming:

- o The measures which are possible without any changes to the existing standards.
- o The measures which require minimal changes to the standards.

Add some text here. You will need to use these references somewhere within the text: [RFC4862] [RFC4861] [RFC6620] [RFC3315]

2. Impact of Multicast Packets in 802.11 Networks

NOTE: much if not all of the subsequent text in this section might need to be transferred to vyncke-6man-mcast-not-efficient-01, which discusses why multicast is not an efficient media in the WiFi environments.

1. Multicast can impact power consumption on hosts if hosts receive multicast packets that are not addressed to them.
2. Excessive use of multicast can reduce the performance of wireless networks.
3. The extra packets are more expensive when they occur with the host not otherwise engaged in using the network.
4. Mobile nodes often have more than one processor and multiple power management states both for the central processing unit and for the WiFi portion (e.g. using only one antenna out of multiple). Often, the battery impact of rejecting a packet in the radio firmware is substantially lower than the impact of passing the packet to the main processor and rejecting it there.

In 802.11 networks, multicast frames towards clients have a greater battery impact than the unicast frames because they are transmitted to all hosts at once, with the AP setting the DTIM bit on the beacon packet to signal to the dozing hosts that the transmission is about to begin.

Thus, if the host were not to wake up right there and then, it would miss the multicast frame. Unicast packets are buffered on the AP and may have a more lenient delivery schedule, which would allow the devices to not have to wake up at every beacon interval (100ms).

The tradeoff between the energy savings and the latency of the multicast delivery may be manipulated by changing the parameter called DTIM interval, which determines how often (every Nth beacon)

the AP can send the indication about the multicast traffic to the clients - with the default values being fairly low, usually in the range of one to three.

Increasing these values increases the latency for the multicast packets, therefore changing the DTIM interval beyond the defaults is usually not recommended.

3. Quantifying the use of Multicast in Neighbor Discovery

Normal operation of Neighbor Discovery uses the following multicast packets.

1. Duplicate Address Detection.
Expected impact: One packet per IPv6 address (a host may be configured to do 2 or more) every time a host joins the network
2. Router Solicitations.
Expected impact: One packet every time a host joins the network.
3. Router Advertisements.
Expected impact:
 - * One multicast RAs every [RA interval] seconds
 - * One solicited RA per host joining the network (if solicited RAs are sent using multicast)
4. Neighbor solicitations. Expected impact: One every time a host talks to a new on-link destination talked to. The response is cached and typically does not expire unless the ND cache is under pressure and subject to garbage collection. Cache entries are refreshed (and possibly deleted) using unicast NUD packets, so cache refreshes do not cause multicast packets to be sent..

With the exception of periodic RAs (and possibly solicited RAs), none of these packets are addressed to all nodes. RS packets are addressed to all routers, and NS packets are addressed to solicited-node multicast groups. Because solicited-node multicast groups contain the last 24 bits of the IPv6 address, in most networks, each solicited-node group will have at most one member.

4. Multicast-limiting measures with no changes in specifications

4.1. On-device robust multicast filtering

The hosts may implement on-device multicast filtering, such that if devices receive multicast packets that are not addressed to them, they will not send the packets to the main CPU but instead remain in a lower sleep state.

It is worth noting that this may require a less deep sleep state than the one required to monitor the TIM in the beacon frames. Also, filtering the packets on the device does not address the inefficiency in spectrum utilisation caused by excessive multicast frames.

4.2. Unicast Solicited Router Advertisements

[RFC4861] in section 6.2.6 already allows to do so via a MAY verb (if the solicitation's source address is not the unspecified address). This is further weakened by the subsequent qualifier being "but the usual case is to multicast the response to the all-nodes group." As a result of this, a lot of implementations do multicast the solicited RAs, significantly impacting the devices.

To help address this, all router implementations SHOULD have a way to send solicited RAs unicast in the environments which wish to do so.

4.3. Infrastructure-based multicast filtering

Ensure that solicited-node multicasts only go to the specific nodes. This can be implemented either using multicast snooping or by converting multicast packets to unicast packets that are addressed to a subset of the hosts..

The latter can be done in two ways:

- o on the 802.11 level alone, preserving the destination within the inner Ethernet frame as multicast
- o on the 802.11 and 802.3 levels, as clarified by the [RFC6085]

Some networks track individual device IP addresses for security and tracking reasons, typically by snooping DAD packets or device traffic as described in [RFC6620]

In these networks, the infrastructure is already aware of which IP addresses are mapped to which MAC addresses, and can use this information to selectively unicast neighbor solicitations to the nodes that will be interested in them.

Most wireless networks are infrastructure-based. The 802.11 standard defines that all communications in such networks will happen via the access points. Therefore, the infrastructure has a chance to intelligently filter any multicast packets that are coming from both local (served by the same access point) and remote (located behind the wired infrastructure) hosts or routers, before forwarding them onto the air to their ultimate destination.

4.4. Proxy the Neighbor Discovery protocol on the access point

802.11 standard defines also that all of packets sent from the client to the Access Point (either for the local over-the-air delivery or for forwarding on to the wired side) are acknowledged (even the multicast ones).

With this in mind, in the scenarios like DAD, a proxy ND implementation has inherently a much better chance of working than the "regular" forwarding of the multicast DAD NS (and the return forwarding of the multicast DAD NA in case of DAD collision that was detected).

Therefore, the environments which want to increase the robustness of the DAD, may wish to proxy the ND on behalf of the clients, therefore reducing the overall client-directed multicast traffic (which is unacknowledged) and increasing the robustness against the poor radio conditions.

4.5. Maximized Interval for Periodic RAs

Assuming the solicited RAs are sent unicast, increasing the interval of the periodic RAs is a natural way of further reducing the amount of multicast packets in the air.

The bounding factor is AdvDefaultLifetime, which is limited by the [RFC4861], section 6.1 on the sending side to 9000 seconds.

Thus, to find the "right" value one will have to balance the robustness in the face of higher packet loss on the segment with the energy consumption by the endpoints. Some real-world mid-scale networks (on the order of 10000 hosts within a single /64) successfully used a value of one RA in 1800 seconds.

However, it is impossible to specify the "best" value - everything will depend on the quality of the local WiFi installation and the radio conditions, with the constraint of 9000 seconds currently specified by the standard.

4.6. Increasing the advertised Reachable value

The NUD with the default settings and active traffic will enter the PROBE state as frequently as every ~30 seconds. [RFC4861] section 7.3.3 defines: "If no response is received after waiting RetransTimer milliseconds after sending the MAX_UNICAST_SOLICIT solicitations, retransmissions cease and the entry SHOULD be deleted. Subsequent traffic to that neighbor will recreate the entry and perform address resolution again."

Short-term connectivity issues at link layer may cause a trigger for the symptoms described in the [RFC7048], therefore triggering the nodes to send multicast neighbor solicitations. However, most of the hosts do not implement at this time the changes suggested there. With the default short timeouts and a wireless environment which forwards multicasts without the filtering, these retransmissions may contribute to further possible failures of NUD in other hosts. In the extreme high density and mobility environments (conferences, stadiums) this may result in avalanche effect and significantly increase the portion of multicast traffic.

Furthermore, an 802.11 segment usually has a single gateway (possibly in a FHRP redundant configuration), therefore making NUD not very useful at all: if that gateway does not function, there is no alternative.

For these kinds of environments it may be useful to significantly increase the REACHABLE_TIME from 30000 milliseconds to 600000 seconds and higher. One possible concern here, however, may be the overflow of the ND table on the gateway, so, again, there is no "best" value suitable for all the networks.

4.7. Clearing the on-link bit in the advertized prefixes

The mobile nodes have generally fairly limited memory, so in the environments where there are thousands of nodes on a single /64, it might be burdensome for them to manage a large neighbor table. Having a lot of hosts with large neighbor tables may mean also a lot of NUD maintenance activity, with the potential for the catastrophic failure of the NUD therefore increasing in the high-density environments.

Clearing the on-link bit in the advertised prefixes causes the hosts to send all the traffic to each other via the default gateway - thus dramatically reducing the size of the neighbor table and the burden of its maintenance on the hosts.

The remaining impact of the link-local addresses still present in the cache can then be mitigated by blocking the direct communications

between the hosts at L2, which is a standard feature in the wireless LAN equipment. This operation effectively turns a wireless LAN segment into a collection of point-to-point links between the hosts and the access point, not dissimilar to the operation of private VLANs in the wired LAN case - making the subnet effectively NBMA.

4.8. Explicit creation of state with DHCPv6 address assignment

Turning the WLAN subnet into an NBMA has a consequence that the DAD may no longer work - which may create a problem with the global addresses. Therefore, it may be necessary to transfer the control over the address assignment to a centralized entity.

Also, the 802.11 protocols operate in the unlicensed bands, which means that the radio conditions may vary greatly. The 802.11 LLC protocol itself does have a fairly robust L2 retransmission mechanism for the acknowledged packets (up to 64 retransmissions). However, there still may be times when the radio conditions are so poor that this robustness is not enough. If the network were to use the snooping to maintain the strict policies (e.g. restrict the source addresses of the traffic), merely snooping the ND may not work, and the data-driven recovery mechanisms might be unacceptable.

In these cases one may consider using DHCPv6 as an address assignment mechanism, which would provide the explicit management of state by the client, and the retransmissions required to create the necessary state on the network side without requiring the node to send the data.

4.9. Client link shutdown within the router lifetime expiry

Some nodes after a longer period of time may decide to completely shut down the radio. This will of course result in the best battery usage, but will incur a tradeoff that waking up the client from the network side will be impossible. However, this mode of operation is the only one not using DHCPv6 which may allow complete avoidance of multicast RA packets: if the client never stays awake for longer than the router lifetime, it will not require the multicast RA processing. This optimization is here for completeness of the discussion - since it changes the connectivity of the client.

5. Multicast-limiting measures with small changes in specifications

5.1. Remove the send-side limit on AdvDefaultLifetime of 9000 Seconds

[RFC4861], section 6.1 limits the AdvDefaultLifetime on the sending side to 9000 seconds, while explicitly requiring the receiving side

to process all the values up to 65535 (maximum allowed by 16-bit unsigned integer that the AdvDefaultLifetime is).

This artificial limit means a hard limit on the maximum router lifetime that can be specified in the configuration. (The authors tried two router implementations: Cisco IOS and radvd. More information welcome).

This artificial restriction prevents from using very long router advertisement intervals that would otherwise be possible - with the difference being more than 7x!

Additionally, allowing the router lifetime of 65535 seconds, coupled with sufficiently long lifetimes for the prefix, would cover the vast majority of the lifetimes of the devices on the WiFi networks. 65535 seconds is 18.2 hours, and the typical mobile devices might not even stay on the same network for such a long period of time. This would allow to increase the robustness of the network in the face of bad radio conditions causing the high loss of the multicast RAs.

5.2. Explicitly Client-Driven Router Advertisements

We can logically extend the "client link shutdown" in the direction of smaller connectivity loss, and imagine that the client, instead of completely shutting the radio down, would flap its radio link somewhere close to router lifetime expiry, therefore, while acting fully within the standards it will be able to maintain the connectivity during all but very short period of time, without any use of periodic RAs.

It may be interesting to explore a modification of the client behavior such that the "flap time" converges to zero, and eventually allowing the client to initiate a unicast Router Solicitation some time shortly before the router lifetime expires. This will have the result of the client being able to maintain the connectivity without the need of processing any periodic RAs. The advantage of doing so is that the RS-RA exchange will happen at the time convenient for the client sleep schedule - thus allowing to maximize the battery life.

6. Acknowledgements

Thanks to the following people for the very useful discussions. In no particular order: Erik Nordmark, Pascal Thubert, Eric Levy-Abegnoli, Ole Troan, Eric Vyncke, Federico Lovison, Jerome Henry.

7. IANA Considerations

None.

8. Security Considerations

Not discussed in -00.

9. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [RFC6085] Gundavelli, S., Townsley, M., Troan, O., and W. Dec, "Address Mapping of IPv6 Multicast Packets on Ethernet", RFC 6085, January 2011.
- [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, May 2012.
- [RFC7048] Nordmark, E. and I. Gashinsky, "Neighbor Unreachability Detection Is Too Impatient", RFC 7048, January 2014.

Authors' Addresses

Andrew Yourtchenko
cisco
7a de Kleetlaan
Diegem, 1831
Belgium

Phone: +32 2 704 5494
Email: ayourtch@cisco.com

Lorenzo Colitti
Google

Email: lorenzo@google.com