

6man Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 28, 2015

M. Boucadair
France Telecom
A. Petrescu
CEA, LIST
September 24, 2014

IPv6 Prefix Length Recommendation for Forwarding
draft-boucadair-6man-prefix-routing-reco-03

Abstract

The length of IP prefixes is an information used by forwarding and routing processes is policy-based. As such, no maximum length must be assumed by design.

Discussions on the 64-bit boundary in IPv6 addressing revealed a need for a clear recommendation on which bits must be used by forwarding decision-making processes. This document sketches a recommendation to be followed by forwarding and routing designs with regards to the prefix length. The aim is to avoid hard-coded routing and forwarding designs that exclude some IP prefix lengths.

Requirements Language

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

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 28, 2015.

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. Recommendation	3
3. IANA Considerations	3
4. Security Considerations	3
5. Acknowledgements	3
6. References	3
6.1. Normative References	4
6.2. Informative References	4
Authors' Addresses	4

1. Introduction

Recent discussions on the 64-bit boundary in IPv6 addressing ([I-D.ietf-6man-why64]) revealed a need for a clear recommendation on which bits must be used by forwarding decision-making processes.

A detailed analysis of the 64-bit boundary in IPv6 addressing, and the implication for end-site prefix assignment, is documented in [I-D.ietf-6man-why64]. No recommendation is included in [I-D.ietf-6man-why64].

It is fundamental to not link routing and forwarding to the IPv6 prefix/address semantics [RFC4291]. This document includes a recommendation for that aim.

Forwarding decisions made by routers primarily rely upon a longest prefix-match algorithm. Like in IPv4, the IPv6 prefix-match algorithms involve one critical operation which is the comparison of a destination address with a prefix present in a routing table (e.g., compare the 2001:db8::1 address with the 2001:db8::/64 prefix). The

recommendation of this document is to be followed by that critical operation.

It is important that the compare operation be a bit-wise comparison, and not a byte-wise comparison.

2. Recommendation

Forwarding decision-making processes MUST NOT restrict by design the length of IPv6 prefixes. In particular, forwarding processes MUST be designed to process prefixes of any length up to /128, by increments of 1.

Obviously, policies can be enforced to restrict the length of IP prefixes advertised within a given domain or in a given interconnection link. These policies are deployment-specific and/or driven by administrative (interconnection) considerations.

This recommendation does not conflict with the 64-bit boundary involved when IPv6 stateless address autoconfiguration (SLAAC, [RFC4862]) is used on links such as Ethernet [RFC2464].

Some lookup algorithm implementations (find the prefix matching a given destination address) may be affected by this recommendation, even more so for IPv6 than IPv4. The performance of some implementations may be degraded when prefix lengths are longer than /64.

3. IANA Considerations

This document does not require any action from IANA.

4. Security Considerations

This document does not introduce security issues in addition to what is discussed in [RFC4291].

5. Acknowledgements

Thanks to Eric Vyncke and Christian Jacquenet for their comments.

Special thanks to Randy Bush and Brian Carpenter for their support.

6. References

6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

6.2. Informative References

- [I-D.ietf-6man-why64] Carpenter, B., Chown, T., Gont, F., Jiang, S., Petrescu, A., and A. Yourtchenko, "Analysis of the 64-bit Boundary in IPv6 Addressing", draft-ietf-6man-why64-05 (work in progress), September 2014.
- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

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

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

IPv6 maintenance Working Group (6man)
Internet-Draft
Updates: 2460, 6145 (if approved)
Intended status: Standards Track
Expires: February 28, 2015

F. Gont
SI6 Networks / UTN-FRH
W. Liu
Huawei Technologies
T. Anderson
Redpill Linpro
August 27, 2014

Deprecating the Generation of IPv6 Atomic Fragments
draft-gont-6man-deprecate-atomfrag-generation-01

Abstract

The core IPv6 specification requires that when a host receives an ICMPv6 "Packet Too Big" message reporting a "Next-Hop MTU" smaller than 1280, the host includes a Fragment Header in all subsequent packets sent to that destination, without reducing the assumed Path-MTU. The simplicity with which ICMPv6 "Packet Too Big" messages can be forged, coupled with the widespread filtering of IPv6 fragments, results in an attack vector that can be leveraged for Denial of Service purposes. This document briefly discusses the aforementioned attack vector, and formally updates RFC2460 such that generation of IPv6 atomic fragments is deprecated, thus eliminating the aforementioned attack vector. Additionally, it formally updates RFC6145 such that the Stateless IP/ICMP Translation Algorithm (SIIT) does not rely on the generation of IPv6 atomic fragments, thus improving the robustness of the protocol.

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 28, 2015.

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. Denial of Service (DoS) attack vector	3
4. Additional Considerations	5
5. Updating RFC2460	7
6. Updating RFC6145	7
7. IANA Considerations	14
8. Security Considerations	14
9. Acknowledgements	15
10. References	15
10.1. Normative References	15
10.2. Informative References	15
Appendix A. Small Survey of OSes that Fail to Produce IPv6 Atomic Fragments	16
Authors' Addresses	17

1. Introduction

[RFC2460] specifies the IPv6 fragmentation mechanism, which allows IPv6 packets to be fragmented into smaller pieces such that they fit in the Path-MTU to the intended destination(s).

Section 5 of [RFC2460] states that, when a host receives an ICMPv6 "Packet Too Big" message [RFC4443] advertising a "Next-Hop MTU" smaller than 1280 (the minimum IPv6 MTU), the host is not required to reduce the assumed Path-MTU, but must simply include a Fragment Header in all subsequent packets sent to that destination. The resulting packets will thus **not** be actually fragmented into several pieces, but rather just include a Fragment Header with both the "Fragment Offset" and the "M" flag set to 0 (we refer to these packets as "atomic fragments"). As required by [RFC6946], these

atomic fragments are essentially processed by the destination host as non-fragment traffic (since there are not really any fragments to be reassembled). IPv6/IPv4 translators will typically employ the Fragment Identification information found in the Fragment Header to select an appropriate Fragment Identification value for the resulting IPv4 fragments.

While atomic fragments might seem rather benign, there are scenarios in which the generation of IPv6 atomic fragments can introduce an attack vector that can be exploited for denial of service purposes. Since there are concrete security implications arising from the generation of IPv6 atomic fragments, and there is no real gain in generating IPv6 atomic fragments (as opposed to e.g. having IPv6/IPv4 translators generate a Fragment Identification value themselves), this document formally updates [RFC2460], forbidding the generation of IPv6 atomic fragments, such that the aforementioned attack vector is eliminated. Additionally, it formally updates [RFC6145] such that the Stateless IP/ICMP Translation Algorithm (SIIT) does not rely on the generation of IPv6 atomic fragments.

Section 3 describes some possible attack scenarios. Section 4 provides additional considerations regarding the usefulness of generating IPv6 atomic fragments. Section 5 formally updates RFC2460 such that this attack vector is eliminated. Section 6 formally updates RFC6145 such that it does not relies on the generation of IPv6 atomic fragments.

2. Terminology

IPv6 atomic fragments

IPv6 packets that contain a Fragment Header with the Fragment Offset set to 0 and the M flag set to 0 (as defined by [RFC6946]).

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. Denial of Service (DoS) attack vector

Let us assume that Host A is communicating with Server B, and that, as a result of the widespread filtering of IPv6 packets with extension headers (including fragmentation) [I-D.gont-v6ops-ipv6-ehs-in-real-world], some intermediate node filters fragments between Host A and Server B. If an attacker sends a forged ICMPv6 "Packet Too Big" (PTB) error message to server B, reporting a Next-Hop MTU smaller than 1280, this will trigger the generation of IPv6 atomic fragments from that moment on (as required by [RFC2460]). When server B starts sending IPv6 atomic fragments

(in response to the received ICMPv6 PTB), these packets will be dropped, since we previously noted that packets with IPv6 EHS were being dropped between Host A and Server B. Thus, this situation will result in a Denial of Service (DoS) scenario.

Another possible scenario is that in which two BGP peers are employing IPv6 transport, and they implement ACLs to drop IPv6 fragments (to avoid control-plane attacks). If the aforementioned BGP peers drop IPv6 fragments but still honor received ICMPv6 Packet Too Big error messages, an attacker could easily attack the peering session by simply sending an ICMPv6 PTB message with a reported MTU smaller than 1280 bytes. Once the attack packet has been fired, it will be the aforementioned routers themselves the ones dropping their own traffic.

The aforementioned attack vector is exacerbated by the following factors:

- o The attacker does not need to forge the IPv6 Source Address of his attack packets. Hence, deployment of simple BCP38 filters will not help as a counter-measure.
- o Only the IPv6 addresses of the IPv6 packet embedded in the ICMPv6 payload need to be forged. While one could envision filtering devices enforcing BCP38-style filters on the ICMPv6 payload, the use of extension (by the attacker) could make this difficult, if at all possible.
- o Many implementations fail to perform validation checks on the received ICMPv6 error messages, as recommended in Section 5.2 of [RFC4443] and documented in [RFC5927]. It should be noted that in some cases, such as when an ICMPv6 error message has (supposedly) been elicited by a connection-less transport protocol (or some other connection-less protocol being encapsulated in IPv6), it may be virtually impossible to perform validation checks on the received ICMPv6 error messages. And, because of IPv6 extension headers, the ICMPv6 payload might not even contain any useful information on which to perform validation checks.
- o Upon receipt of one of the aforementioned ICMPv6 "Packet Too Big" error messages, the Destination Cache [RFC4861] is usually updated to reflect that any subsequent packets to such destination should include a Fragment Header. This means that a single ICMPv6 "Packet Too Big" error message might affect multiple communication instances (e.g., TCP connections) with such destination.
- o As noted in Section 4, SIIT [RFC6145] is the only technology which currently makes use of atomic fragments. Unfortunately, an IPv6

node cannot easily limit its exposure to the aforementioned attack vector by only generating IPv6 atomic fragments towards IPv4 destinations behind a stateless translator. This is due to the fact that Section 3.3 of RFC6052 [RFC6052] encourages operators to use a Network-Specific Prefix (NSP) that maps the IPv4 address space into IPv6. When an NSP is being used, IPv6 addresses representing IPv4 nodes (reached through a stateless translator) are indistinguishable from native IPv6 addresses.

4. Additional Considerations

Besides the security assessment provided in Section 3, it is interesting to evaluate the pros and cons of having an IPv6-to-IPv4 translating router rely on the generation of IPv6 atomic fragments.

Relying on the generation of IPv6 atomic fragments implies a reliance on:

1. ICMPv6 packets arriving from the translator to the IPv6 node
2. The ability of the nodes receiving ICMPv6 PTB messages reporting an MTU smaller than 1280 bytes to actually produce atomic fragments
3. Support for IPv6 fragmentation on the IPv6 side of the translator

Unfortunately,

- o There exists a fair share of evidence of ICMPv6 Packet Too Big messages being dropped on the public Internet (for instance, that is one of the reasons for which PLPMTUD [RFC4821] was produced). Therefore, relying on such messages being successfully delivered will affect the robustness of the protocol that relies on them.
- o A number of IPv6 implementations have been known to fail to generate IPv6 atomic fragments in response to ICMPv6 PTB messages reporting an MTU smaller than 1280 bytes (see Appendix A for a small survey). Additionally, results included in Section 6 of [RFC6145] note that 57% of the tested web servers failed to produce IPv6 atomic fragments in response to ICMPv6 PTB messages reporting an MTU smaller than 1280 bytes. Thus, any protocol relying on IPv6 atomic fragment generation for proper functioning will have interoperability problems with the aforementioned IPv6 stacks.
- o IPv6 atomic fragment generation represents a case in which fragmented traffic is produced where otherwise it would not be needed. Since there is widespread filtering of IPv6 fragments in

the public Internet [I-D.gont-v6ops-ipv6-ehs-in-real-world], this would mean that the (unnecessary) use of IPv6 fragmentation might result, unnecessarily, in a Denial of Service situation even in legitimate cases.

Finally, we note that SIIT essentially employs the Fragment Header of IPv6 atomic fragments to signal the translator how to set the DF bit of IPv4 datagrams (the DF bit is cleared when the IPv6 packet contains a Fragment Header, and is otherwise set to 1 when the IPv6 packet does not contain an IPv6 Fragment Header). Additionally, the translator will employ the low-order 16-bits of the IPv6 Fragment Identification for setting the IPv4 Fragment Identification. At least in theory, this is expected to reduce the Fragment ID collision rate in the following specific scenario:

1. An IPv6 node communicates with an IPv4 node (through SIIT)
2. The IPv4 node is located behind an IPv4 link with an MTU < 1260
3. ECMP routing [RFC2992] with more than one translator are employed for e.g., redundancy purposes

In such a scenario, if each translator were to select the IPv4 Fragment Identification on its own (rather than selecting the IPv4 Fragment ID from the low-order 16-bits of the Fragment Identification of atomic fragments), this could possibly lead to IPv4 Fragment ID collisions. However, since a number of implementations set IPv6 Fragment ID according to the output of a Pseudo-Random Number Generator (PRNG) (see Appendix B of [I-D.ietf-6man-predictable-fragment-id]) and the translator only employs the low-order 16-bits of such value, it is very unlikely that relying on the Fragment ID of the IPv6 atomic fragment will result in a reduced Fragment ID collision rate (when compared to the case where the translator selects each IPv4 Fragment ID on its own).

Finally, we note that [RFC6145] is currently the only "consumer" of IPv6 atomic fragments, and it correctly and diligently notes (in Section 6) the possible interoperability problems of relying on IPv6 atomic fragments, proposing as a workaround something very similar to what we propose in Section 6. We believe that, by making the more robust behavior the default behavior of the "IP/ICMP Translation Algorithm", robustness is improved, and the corresponding code is simplified.

5. Updating RFC2460

The following text from Section 5 of [RFC2460]:

"In response to an IPv6 packet that is sent to an IPv4 destination (i.e., a packet that undergoes translation from IPv6 to IPv4), the originating IPv6 node may receive an ICMP Packet Too Big message reporting a Next-Hop MTU less than 1280. In that case, the IPv6 node is not required to reduce the size of subsequent packets to less than 1280, but must include a Fragment header in those packets so that the IPv6-to-IPv4 translating router can obtain a suitable Identification value to use in resulting IPv4 fragments. Note that this means the payload may have to be reduced to 1232 octets (1280 minus 40 for the IPv6 header and 8 for the Fragment header), and smaller still if additional extension headers are used."

is formally replaced with:

"An IPv6 node that receives an ICMPv6 Packet Too Big error message that reports a Next-Hop MTU smaller than 1280 bytes (the minimum IPv6 MTU) MUST NOT include a Fragment header in subsequent packets sent to the corresponding destination. That is, IPv6 nodes MUST NOT generate IPv6 atomic fragments."

6. Updating RFC6145

The following text from Section 4 (Translating from IPv4 to IPv6) of [RFC6145]:

----- cut here ----- cut here -----
When the IPv4 sender does not set the DF bit, the translator SHOULD always include an IPv6 Fragment Header to indicate that the sender allows fragmentation. The translator MAY provide a configuration function that allows the translator not to include the Fragment Header for the non-fragmented IPv6 packets.

The rules in Section 4.1 ensure that when packets are fragmented, either by the sender or by IPv4 routers, the low-order 16 bits of the fragment identification are carried end-to-end, ensuring that packets are correctly reassembled. In addition, the rules in Section 4.1 use the presence of an IPv6 Fragment Header to indicate that the sender might not be using path MTU discovery (i.e., the packet should not have the DF flag set should it later be translated back to IPv4).
----- cut here ----- cut here -----

is formally replaced with:

----- cut here ----- cut here -----
The rules in Section 4.1 ensure that when packets are fragmented, either by the sender or by IPv4 routers, the low-order 16 bits of the fragment identification are carried end-to-end, ensuring that packets are correctly reassembled.

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

The following text from Section 4.1 ("Translating IPv4 Headers into IPv6 Headers") of [RFC6145]:

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

If there is a need to add a Fragment Header (the DF bit is not set or the packet is a fragment), the header fields are set as above with the following exceptions:

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

is formally replaced with:

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

If there is a need to add a Fragment Header (the packet is a fragment), the header fields are set as above with the following exceptions:

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

The following text from Section 4.2 ("Translating ICMPv4 Headers into ICMPv6 Headers") of [RFC6145]:

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

Code 4 (Fragmentation Needed and DF was Set): Translate to an ICMPv6 Packet Too Big message (Type 2) with Code set to 0. The MTU field MUST be adjusted for the difference between the IPv4 and IPv6 header sizes, i.e., $\text{minimum}(\text{advertised MTU}+20, \text{MTU_of_IPv6_nexthop}, (\text{MTU_of_IPv4_nexthop})+20)$. Note that if the IPv4 router set the MTU field to zero, i.e., the router does not implement [RFC1191], then the translator MUST use the plateau values specified in [RFC1191] to determine a likely path MTU and include that path MTU in the ICMPv6 packet. (Use the greatest plateau value that is less than the returned Total Length field.)

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

is formally replaced with:

----- cut here ----- cut here -----
Code 4 (Fragmentation Needed and DF was Set): Translate to an ICMPv6 Packet Too Big message (Type 2) with Code set to 0. The MTU field MUST be adjusted for the difference between the IPv4 and IPv6 header sizes, but MUST NOT be set to a value smaller than the minimum IPv6 MTU (1280 bytes). That is, it should be set to maximum(1280, minimum(advertised MTU+20, MTU_of_IPv6_nexthop, (MTU_of_IPv4_nexthop)+20)). Note that if the IPv4 router set the MTU field to zero, i.e., the router does not implement [RFC1191], then the translator MUST use the plateau values specified in [RFC1191] to determine a likely path MTU and include that path MTU in the ICMPv6 packet. (Use the greatest plateau value that is less than the returned Total Length field, but that is larger than or equal to 1280.)
----- cut here ----- cut here -----

The following text from Section 5 ("Translating from IPv6 to IPv4") of [RFC6145]:

----- cut here ----- cut here -----
There are some differences between IPv6 and IPv4 (in the areas of fragmentation and the minimum link MTU) that affect the translation. An IPv6 link has to have an MTU of 1280 bytes or greater. The corresponding limit for IPv4 is 68 bytes. Path MTU discovery across a translator relies on ICMP Packet Too Big messages being received and processed by IPv6 hosts, including an ICMP Packet Too Big that indicates the MTU is less than the IPv6 minimum MTU. This requirement is described in Section 5 of [RFC2460] (for IPv6's 1280-octet minimum MTU) and Section 5 of [RFC1883] (for IPv6's previous 576-octet minimum MTU).

In an environment where an ICMPv4 Packet Too Big message is translated to an ICMPv6 Packet Too Big message, and the ICMPv6 Packet Too Big message is successfully delivered to and correctly processed by the IPv6 hosts (e.g., a network owned/operated by the same entity that owns/operates the translator), the translator can rely on IPv6 hosts sending subsequent packets to the same IPv6 destination with IPv6 Fragment Headers. In such an environment, when the translator receives an IPv6 packet with a Fragment Header, the translator SHOULD generate the IPv4 packet with a cleared Don't Fragment bit, and with its identification value from the IPv6 Fragment Header, for all of the IPv6 fragments (MF=0 or MF=1).

In an environment where an ICMPv4 Packet Too Big message is filtered (by a network firewall or by the host itself) or not correctly processed by the IPv6 hosts, the IPv6 host will never generate an IPv6 packet with the IPv6 Fragment Header. In such an environment, the translator SHOULD set the IPv4 Don't Fragment bit. While setting the Don't Fragment bit may create PMTUD black holes [RFC2923] if there are IPv4 links smaller than 1260 octets, this is considered safer than causing IPv4 reassembly errors [RFC4963].

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

is formally replaced with:

----- cut here ----- cut here -----
There are some differences between IPv6 and IPv4 (in the areas of fragmentation and the minimum link MTU) that affect the translation. An IPv6 link has to have an MTU of 1280 bytes or greater. The corresponding limit for IPv4 is 68 bytes. Path MTU discovery across a translator relies on ICMP Packet Too Big messages being received and processed by IPv6 hosts.

The difference in the minimum MTUs of IPv4 and IPv6 is accommodated as follows:

- o When translating an ICMPv4 "Fragmentation Needed" packet, the indicated MTU in the resulting ICMPv6 "Packet Too Big" will never be set to a value lower than 1280. This ensures that the IPv6 nodes will never have to encounter or handle Path MTU values lower than the minimum IPv6 link MTU of 1280. See Section 4.2.
- o When the resulting IPv4 packet is smaller than or equal to 1260 bytes, the translator MUST send the packet with a cleared Don't Fragment bit. Otherwise, the packet MUST be sent with the Don't Fragment bit set. See Section 5.1.

This approach allows Path MTU Discovery to operate end-to-end for paths whose MTU are not smaller than minimum IPv6 MTU of 1280 (which corresponds to MTU of 1260 in the IPv4 domain). On paths that have IPv4 links with MTU < 1260, the IPv4 router(s) connected to those links will fragment the packets in accordance with Section 2.3 of [RFC0791].

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

The following text from Section 5.1 ("Translating IPv6 Headers into IPv4 Headers") of [RFC6145]:

----- cut here ----- cut here -----
Identification: All zero. In order to avoid black holes caused by ICMPv4 filtering or non-[RFC2460]-compatible IPv6 hosts (a workaround is discussed in Section 6), the translator MAY provide a function to generate the identification value if the packet size is greater than 88 bytes and less than or equal to 1280 bytes. The translator SHOULD provide a method for operators to enable or disable this function.

Flags: The More Fragments flag is set to zero. The Don't Fragment (DF) flag is set to one. In order to avoid black holes caused by ICMPv4 filtering or non-[RFC2460]-compatible IPv6 hosts (a workaround is discussed in Section 6), the translator MAY provide a function as follows. If the packet size is greater than 88 bytes and less than or equal to 1280 bytes, it sets the DF flag to zero; otherwise, it sets the DF flag to one. The translator SHOULD provide a method for operators to enable or disable this function.

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

is formally replaced with:

----- cut here ----- cut here -----
Identification: Set according to a Fragment Identification generator at the translator.

Flags: The More Fragments flag is set to zero. The Don't Fragment (DF) flag is set as follows: If the packet size is less than or equal to 1260 bytes, it is set to zero; otherwise, it is set to one.

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

The following text from Section 5.1.1 ("IPv6 Fragment Processing") of [RFC6145]:

----- cut here ----- cut here -----
If a translated packet with DF set to 1 will be larger than the MTU of the next-hop interface, then the translator MUST drop the packet and send the ICMPv6 Packet Too Big (Type 2, Code 0) error message to the IPv6 host with an adjusted MTU in the ICMPv6 message.
----- cut here ----- cut here -----

is formally replaced with:

----- cut here ----- cut here -----
If an IPv6 packet that is smaller than or equal to 1280 bytes results (after translation) in an IPv4 packet that is larger than the MTU of the next-hop interface, then the translator MUST perform IPv4 fragmentation on that packet such that it can be transferred over the constricting link.
----- cut here ----- cut here -----

Finally, the following text from 6 ("Special Considerations for ICMPv6 Packet Too Big") of [RFC6145]:

----- cut here ----- cut here -----
Two recent studies analyzed the behavior of IPv6-capable web servers on the Internet and found that approximately 95% responded as expected to an IPv6 Packet Too Big that indicated MTU = 1280, but only 43% responded as expected to an IPv6 Packet Too Big that indicated an MTU < 1280. It is believed that firewalls violating Section 4.3.1 of [RFC4890] are at fault. Both failures (the 5% wrong response when MTU = 1280 and the 57% wrong response when MTU < 1280) will cause PMTUD black holes [RFC2923]. Unfortunately, the translator cannot improve the failure rate of the first case (MTU = 1280), but the translator can improve the failure rate of the second case (MTU < 1280). There are two approaches to resolving the problem with sending ICMPv6 messages indicating an MTU < 1280. It SHOULD be possible to configure a translator for either of the two approaches.

The first approach is to constrain the deployment of the IPv6/IPv4 translator by observing that four of the scenarios intended for stateless IPv6/IPv4 translators do not have IPv6 hosts on the Internet (Scenarios 1, 2, 5, and 6 described in [RFC6144], which refer to "An IPv6 network"). In these scenarios, IPv6 hosts, IPv6-host-based firewalls, and IPv6 network firewalls can be administered in compliance with Section 4.3.1 of [RFC4890] and therefore avoid the problem witnessed with IPv6 hosts on the Internet.

The second approach is necessary if the translator has IPv6 hosts, IPv6-host-based firewalls, or IPv6 network firewalls that do not (or cannot) comply with Section 5 of [RFC2460] -- such as IPv6 hosts on the Internet. This approach requires the translator to do the following:

1. In the IPv4-to-IPv6 direction: if the MTU value of ICMPv4 Packet Too Big (PTB) messages is less than 1280, change it to 1280. This is intended to cause the IPv6 host and IPv6 firewall to process the ICMP PTB message and generate subsequent packets to this destination with an IPv6 Fragment Header.

Note: Based on recent studies, this is effective for 95% of IPv6

hosts on the Internet.

2. In the IPv6-to-IPv4 direction:

- A. If there is a Fragment Header in the IPv6 packet, the last 16 bits of its value MUST be used for the IPv4 identification value.
- B. If there is no Fragment Header in the IPv6 packet:
 - a. If the packet is less than or equal to 1280 bytes:
 - The translator SHOULD set DF to 0 and generate an IPv4 identification value.
 - To avoid the problems described in [RFC4963], it is RECOMMENDED that the translator maintain 3-tuple state for generating the IPv4 identification value.
 - b. If the packet is greater than 1280 bytes, the translator SHOULD set the IPv4 DF bit to 1.

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

is formally replaced with:

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

A number of studies (see e.g.) indicate that it not unusual for networks to drop ICMPv6 Packet Too Big error messages. Such packet drops will result in PMTUD blackholes [RFC2923], which can only be overcome with PLPMTUD [RFC4821].

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

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 describes a Denial of Service (DoS) attack vector that leverages the widespread filtering of IPv6 fragments in the public Internet by means of ICMPv6 PTB error messages. Additionally, it formally updates [RFC2460] such that this attack vector is eliminated, and also formally updated [RFC6145] such that it does not rely on IPv6 atomic fragments.

9. Acknowledgements

The authors would like to thank (in alphabetical order) Bob Briscoe, Brian Carpenter, Tatuya Jinmei, Jeroen Massar, and Erik Nordmark, for providing valuable comments on earlier versions of this document.

Fernando Gont would like to thank Jan Zorz and Go6 Lab <<http://go6lab.si/>> for providing access to systems and networks that were employed to produce some of tests that resulted in the publication of this document. Additionally, he would like to thank SixXS <<https://www.sixxs.net>> for providing IPv6 connectivity.

10. References

10.1. Normative References

- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [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.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.

10.2. Informative References

- [RFC2923] Lahey, K., "TCP Problems with Path MTU Discovery", RFC 2923, September 2000.
- [RFC2992] Hopps, C., "Analysis of an Equal-Cost Multi-Path Algorithm", RFC 2992, November 2000.
- [RFC5927] Gont, F., "ICMP Attacks against TCP", RFC 5927, July 2010.

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.

[RFC6946] Gont, F., "Processing of IPv6 "Atomic" Fragments", RFC 6946, May 2013.

[I-D.ietf-6man-predictable-fragment-id]
Gont, F., "Security Implications of Predictable Fragment Identification Values", draft-ietf-6man-predictable-fragment-id-01 (work in progress), April 2014.

[I-D.gont-v6ops-ipv6-ehs-in-real-world]
Gont, F., Linkova, J., Chown, T., and W. Will, "IPv6 Extension Headers in the Real World", draft-gont-v6ops-ipv6-ehs-in-real-world-00 (work in progress), August 2014.

[Morbitzer]
Morbitzer, M., "TCP Idle Scans in IPv6", Master's Thesis. Thesis number: 670. Department of Computing Science, Radboud University Nijmegen. August 2013, <https://www.ru.nl/publish/pages/578936/m_morbitzer_masterthesis.pdf>.

Appendix A. Small Survey of OSes that Fail to Produce IPv6 Atomic Fragments

[This section will probably be removed from this document before it is published as an RFC].

This section includes a non-exhaustive list of operating systems that *fail* to produce IPv6 atomic fragments. It is based on the results published in [RFC6946] and [Morbitzer].

The following Operating Systems fail to generate IPv6 atomic fragments in response to ICMPv6 PTB messages that report an MTU smaller than 1280 bytes:

- o FreeBSD 8.0
- o Linux kernel 2.6.32
- o Linux kernel 3.2
- o Mac OS X 10.6.7
- o NetBSD 5.1

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

Tore Anderson
Redpill Linpro
Vitaminveien 1A
NO-0485 Oslo
NORWAY

Phone: +47 959 31 212
Email: tore@redpill-linpro.com

6man
Internet-Draft
Updates: 2460 (if approved)
Intended status: Best Current Practice
Expires: March 1, 2015

F. Gont
UTN-FRH / SI6 Networks
W. Liu
Huawei Technologies
R. Bonica
Juniper Networks
August 28, 2014

Transmission and Processing of IPv6 Options
draft-gont-6man-ipv6-opt-transmit-00.txt

Abstract

Various IPv6 options have been standardized since the core IPv6 standard was first published. This document updates RFC 2460 to clarify how nodes should deal with such IPv6 options and with any options that are defined in the future. It complements [RFC7045], which offers a similar clarification regarding IPv6 Extension Headers.

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 1, 2015.

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 and Problem Statement	2
2. Terminology and Conventions Used in This Document	3
2.1. Terminology	3
2.2. Conventions	3
3. Considerations for All IPv6 Options	4
4. Processing of currently-defined IPv6 Options	5
4.1. Hop-by-Hop Options Header	5
4.2. Destination Options Header	7
5. IANA Considerations	8
6. Security Considerations	10
7. Acknowledgements	10
8. References	10
8.1. Normative References	10
8.2. Informative References	13
Authors' Addresses	13

1. Introduction and Problem Statement

Various IPv6 options have been standardized since the core IPv6 standard [RFC2460] was first published. Except for the padding options (Pad1 and PadN), all the options that have so far been specified are meant to be employed with specific IPv6 extension header types. Additionally, some options have specific requirements such as, for example, only allowing a single instance of the option in the corresponding IPv6 extension header (EH). This establishes some criteria for validating packets that employ IPv6 options.

[RFC2460] specifies that IPv6 extension headers (with the exception of the Hop-by-Hop Options extension header) are not examined or processed by any node along a packet's delivery path, until the packet reaches the node (or each of the set of nodes, in the case of multicast) identified in the Destination Address field of the IPv6 header. However, in practice this is not really the case: some routers, and a variety of middleboxes such as firewalls, load balancers, or packet classifiers, might inspect other parts of each packet [RFC7045]. Hence both end-nodes and intermediate nodes may end up inspecting the contents of extension headers and discard packets based on the presence of specific IPv6 options.

This document clarifies the default processing of IPv6 options. In those cases in which the specifications add additional constraints/

requirements regarding IPv6 options, such additional constraints/requirements are also taken into account.

2. Terminology and Conventions Used in This Document

2.1. Terminology

In the remainder of this document, the term "forwarding node" refers to any router, firewall, load balancer, prefix translator, or any other device or middlebox that forwards IPv6 packets with or without examining the packet in any way.

In this document, "standard" IPv6 options are those specified in detail by IETF Standards Actions [RFC5226]. "Experimental" options include those defined by any Experimental RFC and the option types 0x1E, 0x3E, 0x5E, 0x7E, 0x9E, 0xBE, 0xDE, and 0xFE, defined by [RFC3692] and [RFC4727] when used as experimental options. "Defined" options are the "standard" options plus the "experimental" ones.

The terms "permit" (allow the traffic), "drop" (drop with no notification to sender), and "reject" (drop with appropriate notification to sender) are employed as defined in [RFC3871]. Throughout this document we also employ the term "discard" as a generic term to indicate the act of discarding a packet, irrespective of whether the sender is notified of such packet drops.

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].

2.2. Conventions

This document clarifies some basic validation of IPv6 options, and specifies the default processing of them. We recommend that a configuration option is made available to govern the processing of each IPv6 option type, on a per-EH-type granularity. Such configuration options may include the following possible settings:

- o Permit this IPv6 Option type
- o Drop (and log) packets containing this IPv6 option type
- o Reject (and log) packets containing this IPv6 option type (where the packet drop is signaled with an ICMPv6 error message)
- o Rate-limit the processing of packets containing this IPv6 option type

- o Ignore this IPv6 option type (forwarding packets that contain them)

We note that special care needs to be taken when devices log packet drops/rejects. Devices should count the number of packets dropped/rejected, but the logging of drop/reject events should be limited so as to not overburden device resources.

Finally, we note that when discarding packets, it is generally desirable that the sender be signaled of the packet drop, since this is of use for trouble-shooting purposes. However, throughout this document (when recommending that packets be discarded) we generically refer to the action as "discard" without specifying whether the sender is signaled of the packet drop.

3. Considerations for All IPv6 Options

Forwarding nodes that discard packets (by default) based on the presence of IPv6 options are known to cause connectivity failures and deployment problems. Any forwarding node along an IPv6 packet's path, which forwards the packet for any reason, SHOULD do so regardless of any IPv6 Destination Options that are present, as required by [RFC2460]. Exceptionally, if a forwarding node is designed to examine IPv6 Destination Options for any reason, such as firewalling, it MUST recognise and deal appropriately with all standard IPv6 options types and SHOULD recognise and deal appropriately with all experimental IPv6 options. The list of standard and experimental option types is maintained by IANA (see [IANA-IPV6-PARAM]), and implementors are advised to check this list regularly for updates.

In the case of some options meant to be included in IPv6 extension headers other than Hop-by-Hop Options, [RFC2460] requires destination hosts to discard the corresponding packet if the option is unrecognised. However, intermediate forwarding nodes SHOULD NOT do this, since that might cause them to inadvertently discard traffic using a recently standardised IPv6 option not yet recognised by the intermediate node. The exceptions to this rule are discussed next.

If a forwarding node discards a packet containing a standard IPv6 option, it MUST be the result of a configurable policy and not just the result of a failure to recognise such an option. This means that the discard policy for each standard type of IPv6 option MUST be individually configurable. The default configuration SHOULD allow all standard IPv6 options.

Experimental IPv6 options SHOULD be treated in the same way as standard IPv6 options, including an individually configurable discard policy.

A node that processes the contents of an extension header MUST discard the corresponding packet if it contains any defined options that are not meant for the extension header being processed.

A node that processes the contents of an IPv6 extension header SHOULD discard the corresponding packet if it contains any options that have become deprecated.

A node that processes the contents of an extension header and encounters an undefined (unrecognised) IPv6 option MUST react to such option according to the highest-order two bits of the option type, as specified by Section 4.2 of [RFC2460].

A node that processes an IPv6 extension header MAY discard a packet containing any experimental IPv6 options.

4. Processing of currently-defined IPv6 Options

The following subsections provide advice on how to process the IPv6 options that have been defined at the time of this writing, according to the rules specified in the previous sections.

4.1. Hop-by-Hop Options Header

A node that processes the Hop-by-Hop Options extension header MUST discard the corresponding packet if it contains any of the following options in that header:

- o Type 0x04: Tunnel Encapsulation Limit [RFC2473]
- o Type 0xC9: Home Address [RFC6275]
- o Type 0x8B: ILNP Nonce [RFC6744]
- o Type 0x8C: Line-Identification Option [RFC6788]
- o Type 0x8A: Endpoint Identification [nimrod-eid] [NIMROD-DOC]

NOTE: The rationale for discarding packets containing these options is that these options are meant to be used only with the Destination Options header

A node that processes the Hop-by-Hop Options extension header MUST discard a packet containing multiple instances (i.e., more than one) of this option in the Hop-by-Hop Options extension header:

- o Type 0x05: Router Alert [RFC2711]

NOTE: The rationale for discarding the packet is that [RFC2711] forbids multiple instances of this option.

A node that processes the Hop-by-Hop Options extension header MUST discard a packet that carries a Fragment Header and also contains this option in the Hop-by-Hop Options extension header:

- o Type 0xC2: Jumbo Payload [RFC2675]

NOTE: The rationale for discarding the packet is that [RFC2675] forbids the use of the Jumbo Payload Option in packets that carry a Fragment Header.

A node that processes the Hop-by-Hop Options extension header SHOULD discard a packet containing any of the following options in that header:

- o Type=0x4D: Deprecated

NOTE: The rationale for discarding the packet is that the aforementioned option has been deprecated.

A node that processes the Hop-by-Hop Options extension header MAY discard a packet containing any of the following options in that header:

- o Type 0x1E: RFC3692-style Experiment [RFC4727]
- o Type 0x3E: RFC3692-style Experiment [RFC4727]
- o Type 0x5E: RFC3692-style Experiment [RFC4727]
- o Type 0x7E: RFC3692-style Experiment [RFC4727]
- o Type 0x9E: RFC3692-style Experiment [RFC4727]
- o Type 0xBE: RFC3692-style Experiment [RFC4727]
- o Type 0xDE: RFC3692-style Experiment [RFC4727]
- o Type 0xFE: RFC3692-style Experiment [RFC4727]

NOTE: This is in line with the corresponding specification in [RFC7045] for experimental extension headers.

4.2. Destination Options Header

A node that processes the Destination Options header MUST discard a packet containing any of the following options in that header:

- o Type 0x05: Router Alert [RFC2711]
- o Type 0xC2: Jumbo Payload [RFC2675]
- o Type 0x63: RPL Option [RFC6553]
- o Type 0x08: SMF_DPD [RFC6621]
- o Type 0x6D: MPL Option [I-D.ietf-roll-trickle-mcast]
- o Type 0xEE: IPv6 DFF Header [RFC6971]
- o Type 0x26: Quick-Start [RFC4782]
- o Type 0x07: CALIPSO [RFC5570]

NOTE: The rationale for discarding packets containing these options is that these options are meant to be used only with the Hop by Hop Options header.

A node that processes the Destination Options extension header SHOULD discard a packet containing any of the following options in that header:

- o Type 0x8A: Endpoint Identification [nimrod-eid] [NIMROD-DOC]
- o Type 0x4D: Deprecated

NOTE: The rationale for discarding the packet is that the aforementioned options have been deprecated.

A node that processes the Destination Options extension header MAY discard a packet containing any of the following options in that header:

- o Type 0x1E: RFC3692-style Experiment [RFC4727]
- o Type 0x3E: RFC3692-style Experiment [RFC4727]
- o Type 0x5E: RFC3692-style Experiment [RFC4727]

- o Type 0x7E: RFC3692-style Experiment [RFC4727]
- o Type 0x9E: RFC3692-style Experiment [RFC4727]
- o Type 0xBE: RFC3692-style Experiment [RFC4727]
- o Type 0xDE: RFC3692-style Experiment [RFC4727]
- o Type 0xFE: RFC3692-style Experiment [RFC4727]

NOTE: This is in line with the corresponding specification in [RFC7045] for experimental extension headers.

5. IANA Considerations

IANA is requested to add an extra column entitled "Extension Header Type" to the "Destination Options and Hop-by-Hop Options" registry [IANA-IPV6-PARAM], to clearly mark the IPv6 Extension Header for which each option (defined by IETF Standards Action or IESG Approval) is valid (see the list below). This also applies to Destination Options and Hop-by-Hop Options defined in the future.

What follows is the initial list of IPv6 options and the corresponding marks that indicate which Extension Header type(s) these IPv6 options are valid for:

Hex Value	Description	Reference	EH Type
0x00	Pad1	[RFC2460]	DH
0x01	PadN	[RFC2460]	DH
0xC2	Jumbo Payload	[RFC2675]	H
0x63	RPL Option	[RFC6553]	H
0x04	Tunnel Encapsulation Limit	[RFC2473]	D
0x05	Router Alert	[RFC2711]	H
0x26	Quick-Start	[RFC4782]	H
0x07	CALIPSO	[RFC5570]	H

0x08	SMF_DPD	[RFC6621]	H	
+-----+		+-----+		+-----+
0xC9	Home Address	[RFC6275]	D	
+-----+		+-----+		+-----+
0x8A	Endpoint Identification	[nimrod-eid][NIMROD-DOC]	D	
+-----+		+-----+		+-----+
0x8B	ILNP Nonce	[RFC6744]	D	
+-----+		+-----+		+-----+
0x8C	Line-Identification Option	[RFC6788]	D	
+-----+		+-----+		+-----+
0x4D	Deprecated		U	
+-----+		+-----+		+-----+
0x6D	MPL Option	[I-D.ietf-roll-trickle-mcast]	H	
+-----+		+-----+		+-----+
0xEE	IPv6 DFF Header	[RFC6971]	H	
+-----+		+-----+		+-----+
0x1E	RFC3692-style Experiment	[RFC4727]	DH	
+-----+		+-----+		+-----+
0x3E	RFC3692-style Experiment	[RFC4727]	DH	
+-----+		+-----+		+-----+
0x5E	RFC3692-style Experiment	[RFC4727]	DH	
+-----+		+-----+		+-----+
0x7E	RFC3692-style Experiment	[RFC4727]	DH	
+-----+		+-----+		+-----+
0x9E	RFC3692-style Experiment	[RFC4727]	DH	
+-----+		+-----+		+-----+
0xBE	RFC3692-style Experiment	[RFC4727]	DH	
+-----+		+-----+		+-----+
0xDE	RFC3692-style Experiment	[RFC4727]	DH	
+-----+		+-----+		+-----+
0xFE	RFC3692-style Experiment	[RFC4727]	DH	
+-----+		+-----+		+-----+

Additionally, the following legend should be added to the registry:

D: Destination Options Header
H: Hop-by-Hop Options Header
U: Unknown

6. Security Considerations

Forwarding nodes that operate as firewalls MUST conform to the requirements in this document. In particular, packets containing standard IPv6 options are only to be discarded as a result of an intentionally configured policy.

These requirements do not affect a firewall's ability to filter out traffic containing unwanted or suspect IPv6 options, if configured to do so. However, the changes do require firewalls to be capable of permitting any or all IPv6 options, if configured to do so. The default configurations are intended to allow normal use of any standard IPv6 option, avoiding the interoperability issues described in Section 1 and Section 3.

As noted above, the default configuration might discard packets containing experimental IPv6 options.

7. Acknowledgements

This document is heavily based on [RFC7045], authored by Brian Carpenter and Sheng Jiang.

The authors of this document would like to thank (in alphabetical order) Mike Heard, for providing valuable comments on earlier versions of this document.

8. References

8.1. Normative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, December 1998.

- [RFC2675] Borman, D., Deering, S., and R. Hinden, "IPv6 Jumbograms", RFC 2675, August 1999.
- [RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.
- [RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option", RFC 2711, October 1999.
- [RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, January 2004.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, December 2005.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
- [RFC4304] Kent, S., "Extended Sequence Number (ESN) Addendum to IPsec Domain of Interpretation (DOI) for Internet Security Association and Key Management Protocol (ISAKMP)", RFC 4304, December 2005.
- [RFC4727] Fenner, B., "Experimental Values In IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers", RFC 4727, November 2006.
- [RFC4782] Floyd, S., Allman, M., Jain, A., and P. Sarolahti, "Quick-Start for TCP and IP", RFC 4782, January 2007.
- [RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, December 2007.
- [RFC5201] Moskowitz, R., Nikander, P., Jokela, P., and T. Henderson, "Host Identity Protocol", RFC 5201, April 2008.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, June 2009.
- [RFC5570] StJohns, M., Atkinson, R., and G. Thomas, "Common Architecture Label IPv6 Security Option (CALIPSO)", RFC 5570, July 2009.

- [RFC6275] Perkins, C., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, July 2011.
- [RFC6398] Le Faucheur, F., "IP Router Alert Considerations and Usage", BCP 168, RFC 6398, October 2011.
- [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.
- [RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, March 2012.
- [RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, March 2012.
- [RFC6621] Macker, J., "Simplified Multicast Forwarding", RFC 6621, May 2012.
- [RFC6740] Atkinson,, RJ., "Identifier-Locator Network Protocol (ILNP) Architectural Description", RFC 6740, November 2012.
- [RFC6744] Atkinson,, RJ., "IPv6 Nonce Destination Option for the Identifier-Locator Network Protocol for IPv6 (ILNPv6)", RFC 6744, November 2012.
- [RFC6788] Krishnan, S., Kavanagh, A., Varga, B., Ooghe, S., and E. Nordmark, "The Line-Identification Option", RFC 6788, November 2012.
- [RFC6971] Herberg, U., Cardenas, A., Iwao, T., Dow, M., and S. Cespedes, "Depth-First Forwarding (DFF) in Unreliable Networks", RFC 6971, June 2013.
- [RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, December 2013.
- [RFC7112] Gont, F., Manral, V., and R. Bonica, "Implications of Oversized IPv6 Header Chains", RFC 7112, January 2014.

8.2. Informative References

[Biondi2007]

Biondi, P. and A. Ebalard, "IPv6 Routing Header Security", CanSecWest 2007 Security Conference, 2007, <http://www.secdev.org/conf/IPv6_RH_security-csw07.pdf>.

[I-D.gont-v6ops-ipv6-ehs-in-real-world]

Gont, F., Linkova, J., Chown, T., and W. Will, "IPv6 Extension Headers in the Real World", draft-gont-v6ops-ipv6-ehs-in-real-world-00 (work in progress), August 2014.

[I-D.ietf-roll-trickle-mcast]

Hui, J. and R. Kelsey, "Multicast Protocol for Low power and Lossy Networks (MPL)", draft-ietf-roll-trickle-mcast-09 (work in progress), April 2014.

[IANA-IPV6-PARAM]

Internet Assigned Numbers Authority, "Internet Protocol Version 6 (IPv6) Parameters", December 2013, <<http://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml>>.

[NIMROD-DOC]

Nimrod Documentation Page, ,
"http://ana-3.lcs.mit.edu/~jnc/nimrod/", .

[RFC3871] Jones, G., "Operational Security Requirements for Large Internet Service Provider (ISP) IP Network Infrastructure", RFC 3871, September 2004.

[RFC7126] Gont, F., Atkinson, R., and C. Pignataro, "Recommendations on Filtering of IPv4 Packets Containing IPv4 Options", BCP 186, RFC 7126, February 2014.

[nimrod-eid]

Lynn, C., "Endpoint Identifier Destination Option", IETF Internet Draft, draft-ietf-nimrod-eid-00.txt, November 1995.

Authors' Addresses

Fernando Gont
UTN-FRH / SI6 Networks
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

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

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

6man
Internet-Draft
Updates: 2460 (if approved)
Intended status: Best Current Practice
Expires: February 22, 2016

F. Gont
UTN-FRH / SI6 Networks
W. Liu
Huawei Technologies
R. Bonica
Juniper Networks
August 21, 2015

Transmission and Processing of IPv6 Options
draft-gont-6man-ipv6-opt-transmit-02.txt

Abstract

Various IPv6 options have been standardized since the core IPv6 standard was first published. This document updates RFC 2460 to clarify how nodes should deal with such IPv6 options and with any options that are defined in the future. It complements [RFC7045], which offers a similar clarification regarding IPv6 Extension Headers.

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 22, 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 and Problem Statement	2
2. Terminology and Conventions Used in This Document	3
2.1. Terminology	3
2.2. Conventions	3
3. Considerations for All IPv6 Options	4
4. Processing of currently-defined IPv6 Options	5
4.1. Hop-by-Hop Options Header	5
4.2. Destination Options Header	7
5. IANA Considerations	7
6. Security Considerations	9
7. Acknowledgements	10
8. References	10
8.1. Normative References	10
8.2. Informative References	13
Authors' Addresses	14

1. Introduction and Problem Statement

Various IPv6 options have been standardized since the core IPv6 standard [RFC2460] was first published. Except for the padding options (Pad1 and PadN), all the options that have so far been specified are meant to be employed with specific IPv6 Extension Header (EH) types. Additionally, some options have specific requirements such as, for example, only allowing a single instance of the option in the corresponding IPv6 extension header. This establishes some criteria for validating packets that employ IPv6 options.

[RFC2460] specifies that IPv6 extension headers (with the exception of the Hop-by-Hop Options extension header) are not examined or processed by any node along a packet's delivery path, until the packet reaches the node (or each of the set of nodes, in the case of multicast) identified in the Destination Address field of the IPv6 header. However, in practice this is not really the case: some routers, and a variety of middleboxes such as firewalls, load balancers, or packet classifiers, might inspect other parts of each packet [RFC7045]. Hence both end-nodes and intermediate nodes may end up inspecting the contents of extension headers and discard packets based on the presence of specific IPv6 options.

This document clarifies the default processing of IPv6 options. In those cases in which the specifications add additional constraints/requirements regarding IPv6 options, such additional constraints/requirements are also taken into account.

2. Terminology and Conventions Used in This Document

2.1. Terminology

In the remainder of this document, the term "forwarding node" refers to any router, firewall, load balancer, prefix translator, or any other device or middlebox that forwards IPv6 packets with or without examining the packet in any way.

In this document, "standard" IPv6 options are those specified in detail by IETF Standards Actions [RFC5226]. "Experimental" options include those defined by any Experimental RFC and the option types 0x1E, 0x3E, 0x5E, 0x7E, 0x9E, 0xBE, 0xDE, and 0xFE, defined by [RFC3692] and [RFC4727] when used as experimental options. "Defined" options are the "standard" options plus the "experimental" ones.

The terms "permit" (allow the traffic), "drop" (drop with no notification to sender), and "reject" (drop with appropriate notification to sender) are employed as defined in [RFC3871]. Throughout this document we also employ the term "discard" as a generic term to indicate the act of discarding a packet, irrespective of whether the sender is notified of such packet drops.

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].

2.2. Conventions

This document clarifies some basic validation of IPv6 options, and specifies the default processing of them. We recommend that a configuration option is made available to govern the processing of each IPv6 option type, on a per-EH-type granularity. Such configuration options may include the following possible settings:

- o Permit this IPv6 Option type
- o Drop (and log) packets containing this IPv6 option type
- o Reject (and log) packets containing this IPv6 option type (where the packet drop is signaled with an ICMPv6 error message)

- o Rate-limit the processing of packets containing this IPv6 option type
- o Ignore this IPv6 option type (forwarding packets that contain them)

We note that special care needs to be taken when devices log packet drops/rejects. Devices should count the number of packets dropped/rejected, but the logging of drop/reject events should be limited so as to not overburden device resources.

Finally, we note that when discarding packets, it is generally desirable that the sender be signaled of the packet drop, since this is of use for trouble-shooting purposes. However, throughout this document (when recommending that packets be discarded) we generically refer to the action as "discard" without specifying whether the sender is signaled of the packet drop.

3. Considerations for All IPv6 Options

Forwarding nodes that discard packets (by default) based on the presence of IPv6 options are known to cause connectivity failures and deployment problems. Any forwarding node along an IPv6 packet's path, which forwards the packet for any reason, SHOULD do so regardless of any IPv6 Destination Options that are present, as required by [RFC2460]. Exceptionally, if a forwarding node is designed to examine IPv6 Destination Options for any reason, such as firewalling, it MUST recognise and deal appropriately with all standard IPv6 options types and SHOULD recognise and deal appropriately with all experimental IPv6 options. The list of standard and experimental option types is maintained by IANA (see [IANA-IPV6-PARAM]), and implementors are advised to check this list regularly for updates.

In the case of some options meant to be included in IPv6 extension headers other than Hop-by-Hop Options, [RFC2460] requires destination hosts to discard the corresponding packet if the option is unrecognised. However, intermediate forwarding nodes SHOULD NOT do this, since doing so might cause them to inadvertently discard traffic using a recently standardised IPv6 option not yet recognised by the intermediate node. The exceptions to this rule are discussed next.

If a forwarding node discards a packet containing a standard IPv6 option, it MUST be the result of a configurable policy and not just the result of a failure to recognise such an option. This means that the discard policy for each standard type of IPv6 option MUST be

individually configurable. The default configuration SHOULD allow all standard IPv6 options.

Experimental IPv6 options SHOULD be treated in the same way as standard IPv6 options, including an individually configurable discard policy.

A node that processes the contents of an extension header MUST discard the corresponding packet if it contains any defined options that are not meant for the extension header being processed. This document requests IANA to add a new column to [IANA-IPV6-PARAM] to clearly mark the IPv6 Extension Header type(s) for which each option (defined by IETF Standards Action or IESG Approval) is valid.

A node that processes the contents of an IPv6 extension header MAY discard the corresponding packet if it contains any options that have become deprecated. Whether or not such packets are dropped SHOULD be configurable, and the default setting MUST be to not drop such packets.

A node that processes the contents of an extension header and encounters an undefined (unrecognised) IPv6 option MUST react to such option according to the highest-order two bits of the option type, as specified by Section 4.2 of [RFC2460].

A node that processes an IPv6 extension header MAY discard a packet containing any experimental IPv6 options.

4. Processing of currently-defined IPv6 Options

The following subsections provide advice on how to process the IPv6 options that have been defined at the time of this writing, according to the rules specified in the previous sections.

4.1. Hop-by-Hop Options Header

A node that processes the Hop-by-Hop Options extension header MUST discard the corresponding packet if it contains any options that are not valid for the Hop-by-Hop Options extension header [IANA-IPV6-PARAM].

A node that processes the Hop-by-Hop Options extension header MUST discard a packet containing multiple instances (i.e., more than one) of this option in the Hop-by-Hop Options extension header:

- o Type 0x05: Router Alert [RFC2711]

NOTE: The rationale for discarding the packet is that [RFC2711] forbids multiple instances of this option.

A node that processes the Hop-by-Hop Options extension header MUST discard a packet that carries a Fragment Header and also contains this option in the Hop-by-Hop Options extension header:

- o Type 0xC2: Jumbo Payload [RFC2675]

NOTE: The rationale for discarding the packet is that [RFC2675] forbids the use of the Jumbo Payload Option in packets that carry a Fragment Header.

A node that processes the Hop-by-Hop Options extension header MAY discard a packet containing any of the following options in that header:

- o Type=0x4D: Deprecated

NOTE: The rationale for discarding the packet is that the aforementioned option has been deprecated.

A node that processes the Hop-by-Hop Options extension header MAY discard a packet containing any of the following options in that header:

- o Type 0x1E: RFC3692-style Experiment [RFC4727]
- o Type 0x3E: RFC3692-style Experiment [RFC4727]
- o Type 0x5E: RFC3692-style Experiment [RFC4727]
- o Type 0x7E: RFC3692-style Experiment [RFC4727]
- o Type 0x9E: RFC3692-style Experiment [RFC4727]
- o Type 0xBE: RFC3692-style Experiment [RFC4727]
- o Type 0xDE: RFC3692-style Experiment [RFC4727]
- o Type 0xFE: RFC3692-style Experiment [RFC4727]

NOTE: This is in line with the corresponding specification in [RFC7045] for experimental extension headers.

4.2. Destination Options Header

A node that processes the Destination Options header MUST discard a packet containing any options that are not valid for the Destination Options header [IANA-IPV6-PARAM].

A node that processes the Destination Options extension header MAY discard a packet containing any of the following options in that header:

- o Type 0x8A: Endpoint Identification [nimrod-eid] [NIMROD-DOC]
- o Type 0x4D: Deprecated

NOTE: The rationale for discarding the packet is that the aforementioned options have been deprecated.

A node that processes the Destination Options extension header MAY discard a packet containing any of the following options in that header:

- o Type 0x1E: RFC3692-style Experiment [RFC4727]
- o Type 0x3E: RFC3692-style Experiment [RFC4727]
- o Type 0x5E: RFC3692-style Experiment [RFC4727]
- o Type 0x7E: RFC3692-style Experiment [RFC4727]
- o Type 0x9E: RFC3692-style Experiment [RFC4727]
- o Type 0xBE: RFC3692-style Experiment [RFC4727]
- o Type 0xDE: RFC3692-style Experiment [RFC4727]
- o Type 0xFE: RFC3692-style Experiment [RFC4727]

NOTE: This is in line with the corresponding specification in [RFC7045] for experimental extension headers.

5. IANA Considerations

IANA is requested to add an extra column entitled "Extension Header Types" to the "Destination Options and Hop-by-Hop Options" registry [IANA-IPV6-PARAM], to clearly mark the IPv6 Extension Header types for which each option (defined by IETF Standards Action or IESG Approval) is valid (see the list below). This also applies to Destination Options and Hop-by-Hop Options defined in the future.

What follows is the initial list of IPv6 options and the corresponding marks that indicate which Extension Header type(s) these IPv6 options are valid for:

Hex Value	Description	Reference	EH Types
0x00	Pad1	[RFC2460]	DH
0x01	PadN	[RFC2460]	DH
0xC2	Jumbo Payload	[RFC2675]	H
0x63	RPL Option	[RFC6553]	H
0x04	Tunnel Encapsulation Limit	[RFC2473]	D
0x05	Router Alert	[RFC2711]	H
0x26	Quick-Start	[RFC4782]	H
0x07	CALIPSO	[RFC5570]	H
0x08	SMF_DPD	[RFC6621]	H
0xC9	Home Address	[RFC6275]	D
0x8A	Endpoint Identification	[nimrod-eid][NIMROD-DOC]	D
0x8B	ILNP Nonce	[RFC6744]	D
0x8C	Line-Identification Option	[RFC6788]	D
0x4D	Deprecated		U
0x6D	MPL Option	[I-D.ietf-roll-trickle-mcast]	H
0xEE	IPv6 DFF Header	[RFC6971]	H
0x1E	RFC3692-style Experiment	[RFC4727]	DH

0x3E	RFC3692-style Experiment	[RFC4727]	DH
0x5E	RFC3692-style Experiment	[RFC4727]	DH
0x7E	RFC3692-style Experiment	[RFC4727]	DH
0x9E	RFC3692-style Experiment	[RFC4727]	DH
0xBE	RFC3692-style Experiment	[RFC4727]	DH
0xDE	RFC3692-style Experiment	[RFC4727]	DH
0xFE	RFC3692-style Experiment	[RFC4727]	DH

Additionally, the following legend should be added to the registry:

D: Destination Options Header

H: Hop-by-Hop Options Header

U: Unknown

6. Security Considerations

Forwarding nodes that operate as firewalls MUST conform to the requirements in this document. In particular, packets containing standard IPv6 options are only to be discarded as a result of an intentionally configured policy.

These requirements do not affect a firewall's ability to filter out traffic containing unwanted or suspect IPv6 options, if configured to do so. However, the changes do require firewalls to be capable of permitting any or all IPv6 options, if configured to do so. The default configurations are intended to allow normal use of any standard IPv6 option, avoiding the interoperability issues described in Section 1 and Section 3.

As noted above, the default configuration might discard packets containing experimental IPv6 options.

7. Acknowledgements

This document is heavily based on [RFC7045], authored by Brian Carpenter and Sheng Jiang.

The authors of this document would like to thank (in alphabetical order) Brian Carpenter, Mike Heard, and Jen Linkova, for providing valuable comments on earlier versions of this document.

8. References

8.1. Normative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<http://www.rfc-editor.org/info/rfc1034>>.
- [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>>.
- [RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <<http://www.rfc-editor.org/info/rfc2205>>.
- [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>>.
- [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <<http://www.rfc-editor.org/info/rfc2473>>.
- [RFC2675] Borman, D., Deering, S., and R. Hinden, "IPv6 Jumbograms", RFC 2675, DOI 10.17487/RFC2675, August 1999, <<http://www.rfc-editor.org/info/rfc2675>>.
- [RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, DOI 10.17487/RFC2710, October 1999, <<http://www.rfc-editor.org/info/rfc2710>>.
- [RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option", RFC 2711, DOI 10.17487/RFC2711, October 1999, <<http://www.rfc-editor.org/info/rfc2711>>.

- [RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, DOI 10.17487/RFC3692, January 2004, <<http://www.rfc-editor.org/info/rfc3692>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<http://www.rfc-editor.org/info/rfc4302>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<http://www.rfc-editor.org/info/rfc4303>>.
- [RFC4304] Kent, S., "Extended Sequence Number (ESN) Addendum to IPsec Domain of Interpretation (DOI) for Internet Security Association and Key Management Protocol (ISAKMP)", RFC 4304, DOI 10.17487/RFC4304, December 2005, <<http://www.rfc-editor.org/info/rfc4304>>.
- [RFC4727] Fenner, B., "Experimental Values In IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers", RFC 4727, DOI 10.17487/RFC4727, November 2006, <<http://www.rfc-editor.org/info/rfc4727>>.
- [RFC4782] Floyd, S., Allman, M., Jain, A., and P. Sarolahti, "Quick-Start for TCP and IP", RFC 4782, DOI 10.17487/RFC4782, January 2007, <<http://www.rfc-editor.org/info/rfc4782>>.
- [RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, DOI 10.17487/RFC5095, December 2007, <<http://www.rfc-editor.org/info/rfc5095>>.
- [RFC5201] Moskowitz, R., Nikander, P., Jokela, P., Ed., and T. Henderson, "Host Identity Protocol", RFC 5201, DOI 10.17487/RFC5201, April 2008, <<http://www.rfc-editor.org/info/rfc5201>>.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.
- [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, DOI 10.17487/RFC5533, June 2009, <<http://www.rfc-editor.org/info/rfc5533>>.

- [RFC5570] StJohns, M., Atkinson, R., and G. Thomas, "Common Architecture Label IPv6 Security Option (CALIPSO)", RFC 5570, DOI 10.17487/RFC5570, July 2009, <<http://www.rfc-editor.org/info/rfc5570>>.
- [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <<http://www.rfc-editor.org/info/rfc6275>>.
- [RFC6398] Le Faucheur, F., Ed., "IP Router Alert Considerations and Usage", BCP 168, RFC 6398, DOI 10.17487/RFC6398, October 2011, <<http://www.rfc-editor.org/info/rfc6398>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., 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, DOI 10.17487/RFC6550, March 2012, <<http://www.rfc-editor.org/info/rfc6550>>.
- [RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, DOI 10.17487/RFC6553, March 2012, <<http://www.rfc-editor.org/info/rfc6553>>.
- [RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, DOI 10.17487/RFC6554, March 2012, <<http://www.rfc-editor.org/info/rfc6554>>.
- [RFC6621] Macker, J., Ed., "Simplified Multicast Forwarding", RFC 6621, DOI 10.17487/RFC6621, May 2012, <<http://www.rfc-editor.org/info/rfc6621>>.
- [RFC6740] Atkinson, RJ. and SN. Bhatti, "Identifier-Locator Network Protocol (ILNP) Architectural Description", RFC 6740, DOI 10.17487/RFC6740, November 2012, <<http://www.rfc-editor.org/info/rfc6740>>.
- [RFC6744] Atkinson, RJ. and SN. Bhatti, "IPv6 Nonce Destination Option for the Identifier-Locator Network Protocol for IPv6 (ILNPv6)", RFC 6744, DOI 10.17487/RFC6744, November 2012, <<http://www.rfc-editor.org/info/rfc6744>>.

- [RFC6788] Krishnan, S., Kavanagh, A., Varga, B., Ooghe, S., and E. Nordmark, "The Line-Identification Option", RFC 6788, DOI 10.17487/RFC6788, November 2012, <<http://www.rfc-editor.org/info/rfc6788>>.
- [RFC6971] Herberg, U., Ed., Cardenas, A., Iwao, T., Dow, M., and S. Cespedes, "Depth-First Forwarding (DFF) in Unreliable Networks", RFC 6971, DOI 10.17487/RFC6971, June 2013, <<http://www.rfc-editor.org/info/rfc6971>>.
- [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>>.
- [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>>.

8.2. Informative References

- [Biondi2007]
Biondi, P. and A. Ebalard, "IPv6 Routing Header Security", CanSecWest 2007 Security Conference, 2007, <http://www.secdev.org/conf/IPv6_RH_security-csw07.pdf>.
- [I-D.ietf-roll-trickle-mcast]
Hui, J. and R. Kelsey, "Multicast Protocol for Low power and Lossy Networks (MPL)", draft-ietf-roll-trickle-mcast-12 (work in progress), June 2015.
- [I-D.ietf-v6ops-ipv6-ehs-in-real-world]
Gont, F., Linkova, J., Chown, T., and S. LIU, "Observations on IPv6 EH Filtering in the Real World", draft-ietf-v6ops-ipv6-ehs-in-real-world-00 (work in progress), April 2015.
- [IANA-IPV6-PARAM]
Internet Assigned Numbers Authority, "Internet Protocol Version 6 (IPv6) Parameters", December 2013, <<http://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml>>.
- [NIMROD-DOC]
Nimrod Documentation Page, , <<http://ana-3.lcs.mit.edu/~jnc/nimrod/>>.

[nimrod-eid]

Lynn, C., "Endpoint Identifier Destination Option", IETF Internet Draft, draft-ietf-nimrod-eid-00.txt, November 1995.

[RFC3871] Jones, G., Ed., "Operational Security Requirements for Large Internet Service Provider (ISP) IP Network Infrastructure", RFC 3871, DOI 10.17487/RFC3871, September 2004, <<http://www.rfc-editor.org/info/rfc3871>>.

[RFC7126] Gont, F., Atkinson, R., and C. Pignataro, "Recommendations on Filtering of IPv4 Packets Containing IPv4 Options", BCP 186, RFC 7126, DOI 10.17487/RFC7126, February 2014, <<http://www.rfc-editor.org/info/rfc7126>>.

Authors' Addresses

Fernando Gont
UTN-FRH / SI6 Networks
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

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

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

IPv6 maintenance Working Group (6man)
Internet-Draft
Updates: 4861 (if approved)
Intended status: Standards Track
Expires: April 24, 2015

F. Gont
SI6 Networks / UTN-FRH
R. Bonica
Juniper Networks
W. Liu
Huawei Technologies
October 21, 2014

Validation of IPv6 Neighbor Discovery Options
draft-gont-6man-nd-opt-validation-01

Abstract

This memo specifies validation rules for IPv6 Neighbor Discovery (ND) Options. In order to avoid pathological outcomes, IPv6 implementations validate incoming ND options using these rules.

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 April 24, 2015.

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. Methodology	3
4. The Source Link-Layer Address (SLLA) Option	4
5. The Target Link-Layer Address (TLLA) Option	5
6. The Prefix Information Option	5
7. The Redirected Header Option	7
8. The MTU Option	7
9. The Route Information Option	8
10. The Recursive DNS Server (RDNSS) Option	9
11. The DNS Search List (DNSSL) Option	10
12. IANA Considerations	11
13. Security Considerations	11
14. Acknowledgements	12
15. References	12
15.1. Normative References	12
15.2. Informative References	12
Appendix A. Mapping an IPv6 Address to a Local Router's Own Link-layer Address	13
Appendix B. Mapping a Unicast IPv6 Address to A Broadcast Link- Layer Address	14
Authors' Addresses	15

1. Introduction

IPv6 [RFC2460] nodes use Neighbor Discovery (ND) [RFC4861] to discover their neighbors and to learn their neighbors' link-layer addresses. IPv6 hosts also use ND to find neighboring routers that can forward packets on their behalf. Finally, IPv6 nodes use ND to verify neighbor reachability, and to detect link-layer address changes.

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 messages can include options that convey additional information. Currently, the following ND options are specified:

- o Source link-layer address (SLLA) [RFC4861]
- o Target link-layer address (TLLA) [RFC4861]
- o Prefix information [RFC4861]
- o Redirected header [RFC4861]
- o MTU [RFC4861]
- o Route Information [RFC4191]
- o Recursive DNS Server (RDNSS) [RFC6106]
- o DNS Search List (DNSSL) [RFC6106]

This memo specifies validation rules for the ND options mentioned above. In order to avoid pathological outcomes (such as [FreeBSD-rtssold]), IPv6 implementations validate incoming ND options using these rules.

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. Methodology

Section 4 through Section 11 of this document define validation rules for ND options. These sections also specify actions that are to be taken when an implementation encounters an invalid option. Possible actions are:

- o The entire option MUST be ignored, However, the rest of the ND message MAY be processed.
- o The entire ND message MUST be ignored

In the spirit of "being liberal in what you receive", the first action is always preferred. However, when an option length attribute is invalid, it is not possible to parse the rest of the ND message. In these cases, subsequent ND options should be ignored.

4. The Source Link-Layer Address (SLLA) Option

NS, RS, and RA messages MAY contain an SLLA Option. If any other ND message contains an SLLA Option, the SLLA Option MUST be ignored. However, the rest of the ND message MAY be processed. (As per [RFC4861]).

Figure 1 illustrates the SLLA Option:

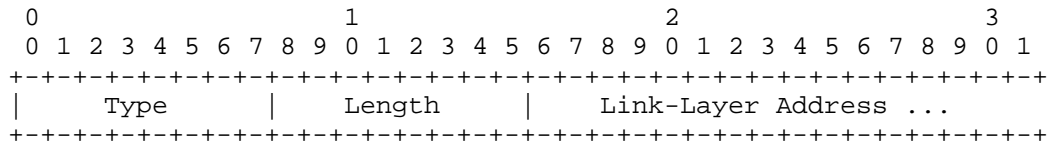


Figure 1: Source Link-Layer Address Option

The Type field is set to 1.

The Length field specifies the length of the option (including the Type and Length fields) in units of 8 octets. The Length field MUST be valid for the underlying link layer. For example, for IEEE 802 addresses the Length field MUST be 1 [RFC2464]. If an incoming ND message does not pass this validation check, the entire ND message MUST be discarded.

The Link-Layer Address field specifies the link-layer address of the packet's originator. It MUST NOT be any of the following:

- o a broadcast address (see Appendix B for rationale)
- o a multicast address (see Appendix B for rationale)
- o an address belonging to the receiving node (see Appendix A for rationale)

If an incoming ND message does not pass this validation check, the SLLA Option MUST be ignored. However, the rest of the ND message MAY be processed.

An ND message that carries the SLLA Option MUST have a source address other than the unspecified address (0:0:0:0:0:0:0:0). If an incoming ND message does not pass this validation check, the SLLA Option MUST be ignored. However, the rest of the ND message MAY be processed. (As per [RFC4861]).

5. The Target Link-Layer Address (TLLA) Option

NA and Redirect messages MAY contain a TLILA Option. If any other ND message contains an TLILA Option, the TLILA Option MUST be ignored. However, the rest of the ND message MAY be processed. (As per [RFC4861]).

Figure 2 illustrates the Target link-layer address:

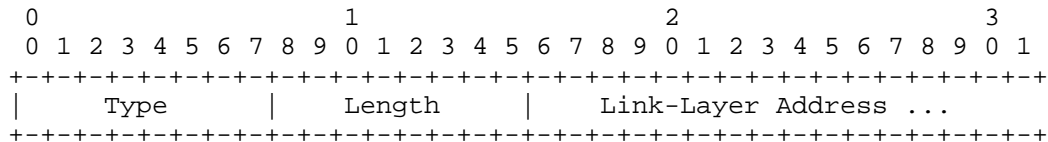


Figure 2: Target link-layer address option format

The Type field is set to 2.

The Length field specifies the length of the option (including the Type and Length fields) in units of 8 octets. The Length field MUST be valid for the underlying link layer. For example, for IEEE 802 addresses the Length field MUST be 1 [RFC2464]. If an incoming ND message does not pass this validation check, the entire ND message MUST be discarded.

An ND message that carries the TLILA option also includes a Target Address. The TLILA Option Link-Layer Address maps to the Target Address. The TLILA Option Link-Layer Address MUST NOT be any of the following:

- o a broadcast address (see Appendix B for rationale)
- o a multicast address (see Appendix B for rationale)
- o an address belonging to the receiving node (see Appendix A for rationale)

If an incoming ND message does not pass this validation check, the TLILA Option MUST be ignored. However, the rest of the ND message MAY be processed.

6. The Prefix Information Option

The RA message MAY contain a Prefix Information Option. If any other ND message contains an Prefix Information Option, the Prefix Information Option MUST be ignored. However, the rest of the ND message MAY be processed. (As per [RFC4861]).

Figure 3 illustrates the Prefix Information Option:

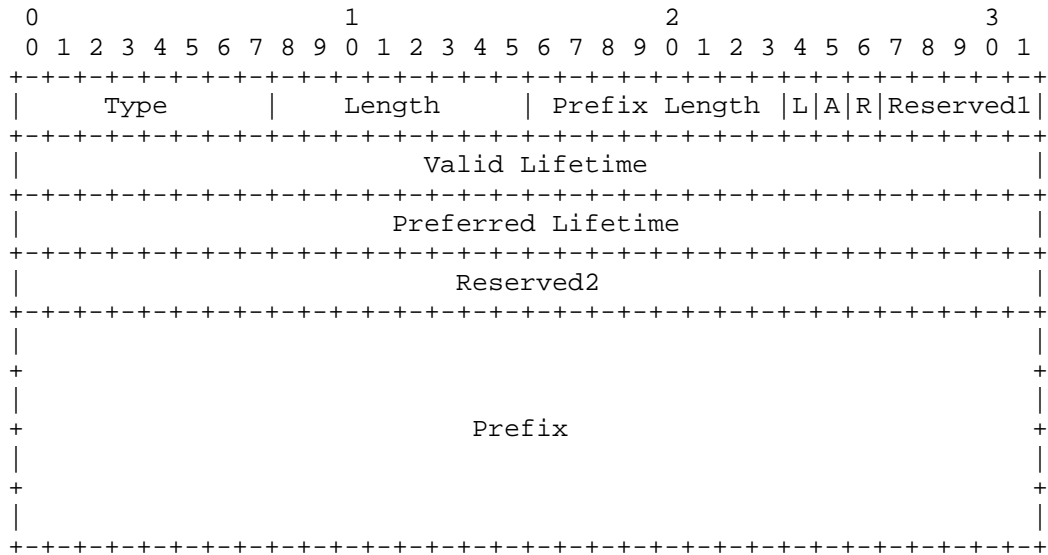


Figure 3: Prefix Information option format

The Type field is set to 3.

The Length field MUST be set to 4. If an incoming ND message does not pass this validation check, the entire ND message MUST be discarded.

As stated in [RFC4861] the Preferred Lifetime MUST be less than or equal to the Valid Lifetime. If an incoming ND message does not pass this validation check, the Prefix Information Option MUST be ignored. However, the rest of the ND message MAY be processed.

The Prefix Length contains the number of leading bits in the prefix that are to be considered valid. It MUST be greater than or equal to 0, and smaller than or equal to 128. If the field does not pass this check, the Prefix Information Option MUST be ignored. However, the rest of the ND message MAY be processed.

The Prefix field MUST NOT contain a link-local or multicast prefix. If an incoming ND message does not pass this validation check, the Prefix Information Option MUST be ignored. However, the rest of the ND message MAY be processed.

7. The Redirected Header Option

The Redirect message MAY contain a Redirect Header Option. If any other ND message contains a Redirect Header Option, the Redirect Header Option MUST be ignored. However, the rest of the ND message MAY be processed. (As per [RFC4861]).

Figure 4 illustrates the Redirected Header option:

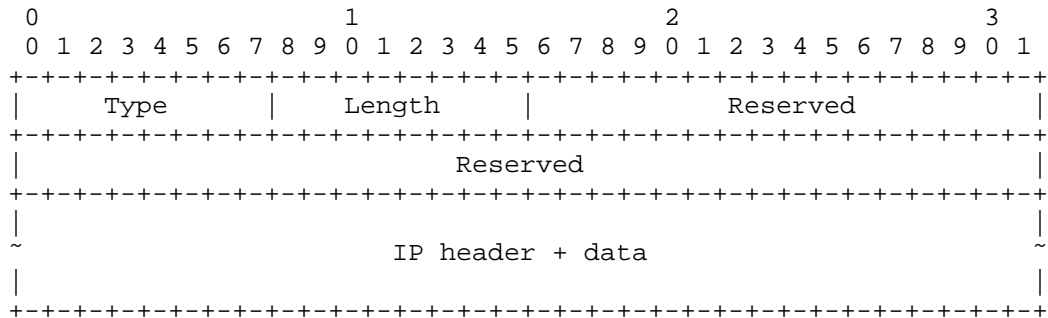


Figure 4: Redirected Header Option format

The Type field is 4.

The Length field specifies the option size (including the Type and Length fields) in units of 8 octets. Its value MUST be greater than or equal to 6. If an incoming ND message does not pass this validation check, the entire ND message MUST be discarded.

The value 6 was chosen to accommodate mandatory fields (8 octets) plus the base IPv6 header (40 octets).

8. The MTU Option

The RA message MAY contain an MTU Option. If any other ND message contains an MTU Option, the MTU Option MUST be ignored. However, the rest of the ND message MAY be processed. (As per [RFC4861]).

Figure 5 illustrates the MTU option:

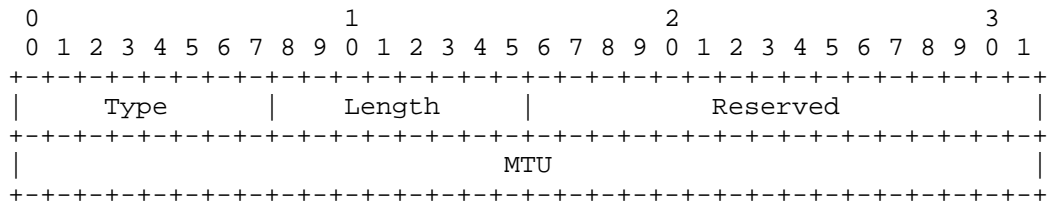


Figure 5: MTU Option Format

The Type field identifies the kind of option and is set to 5.

The Length field MUST BE set to 1 by the sender. If an incoming ND message does not pass this validation check, the entire ND message MUST be discarded.

The MTU field is a 32-bit unsigned integer that specifies the MTU value that should be used for this link. [RFC2460] specifies that the minimum IPv6 MTU is 1280 octets. Therefore, the MTU MUST be greater than or equal to 1280. If an incoming ND message does not pass this validation check, the MTU Option MUST be ignored. However, the rest of the ND message MAY be processed.

Additionally, the advertised MTU MUST NOT exceed the maximum MTU specified for the link-type (e.g., [RFC2464] for Ethernet networks). If an incoming ND message does not pass this validation check, the MTU Option MUST be ignored. However, the rest of the ND message MAY be processed.

9. The Route Information Option

The RA message MAY contain a Route Information Option. If any other ND message contains a Route Information Option, the Route Information Option MUST be ignored. However, the rest of the ND message MAY be processed.

Figure 6 illustrates Route Information option:

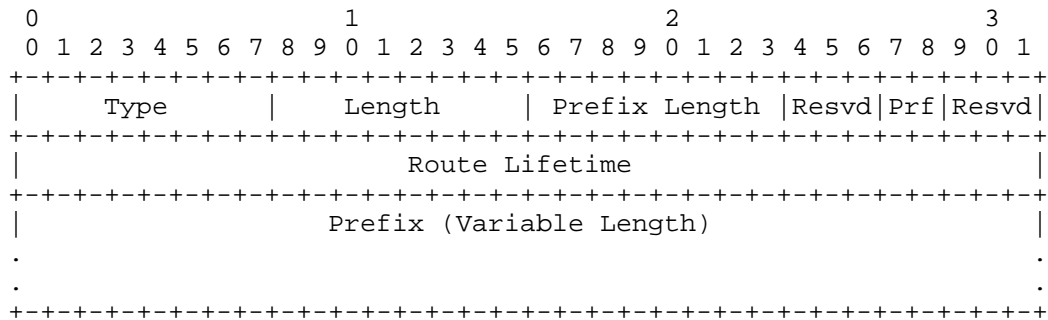


Figure 6: Route Information Option Format

The Type field is 24.

The Length field contains the length of the option (including the Type and Length fields) in units of 8 octets. Its value **MUST** be at least 1 and at most 3. If an incoming ND message does not pass this validation check, the entire ND message **MUST** be discarded.

The Prefix Length field indicates the number of significant bits in the Prefix field that are significant. Its value **MUST** be less than or equal to 128. If the field does not pass this check, the Route Information Option **MUST** be ignored.

The Length field and the Prefix Length field are closely related, as the Length field constrains the possible values of the Prefix Length field. If the Prefix Length is equal to 0, the Length **MUST** be equal to 1. If the Prefix Length is greater than 0 and less than 65, the Length **MUST** be equal to 2. If the Prefix Length is greater than 65 and less than 129, the Length **MUST** be equal to 3. If an incoming ND message does not pass this validation check, the entire ND message **MUST** be discarded.

The Prefix field **MUST NOT** contain a link-local unicast prefix (fe80::/10) or a link-local multicast prefix (e.g., ff02::0/64). If an incoming ND message does not pass this validation check, the Route Information Option **MUST** be ignored. However, the rest of the ND message **MAY** be processed.

10. The Recursive DNS Server (RDNSS) Option

The RA message **MAY** contain a Recursive DNS Server (RDNSS) Option. If any other ND message contains an RDNSS Option, the RDNSS Option **MUST** be ignored. However, the rest of the ND message **MAY** be processed.

Figure 7 illustrates the syntax of the RDNSS option:

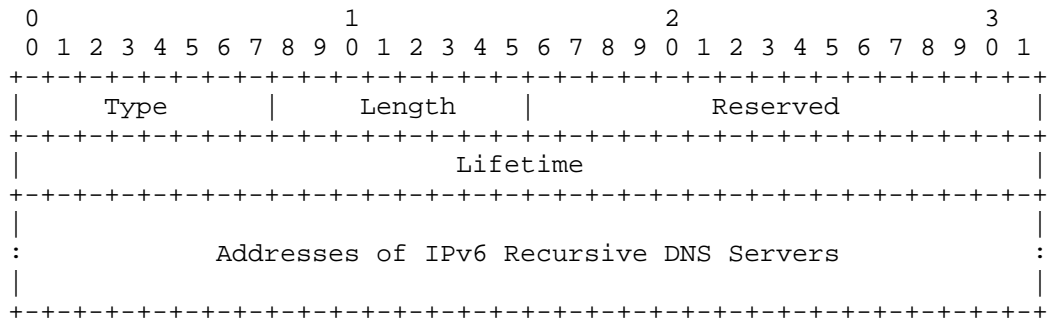


Figure 7: Recursive DNS Server Option Format

The Type field is 25.

The Length field specifies the length of the option (including the Type and Length fields) in units of 8 octets. Its value MUST be greater than or equal to 3. Additionally the Length field MUST pass the following check:

$$(\text{Length} - 1) \% 2 == 0$$

Figure 8

If the option does not pass these validation checks, the entire ND message MUST be discarded.

The Lifetime field specifies the maximum time in seconds that a node may use the IPv6 addresses included in the option for name resolution, with a value of 0 indicating that they can no longer be used. If the Lifetime field is not equal to 0, it MUST be at least 1800 (MinRtrAdvInterval) and at most 3600 (2*MaxRtrAdvInterval). If the RDNSS option does not pass this validation check, it MUST be ignored. However, the rest of the ND message MAY be processed.

The RDNSS address list MUST NOT contain multicast addresses or the unspecified address. If an incoming ND message does not pass this validation check, the RDNSS Option MUST be ignored. However, the rest of the ND message MAY be processed.

11. The DNS Search List (DNSSL) Option

The RA message MAY contain a DNS Search List (DNSSL) Option. If any other ND message contains a DNSSL Option, the DNSSL Option MUST be ignored. However, the rest of the ND message MAY be processed.

Figure 9 illustrates the syntax of the DNSSL option:

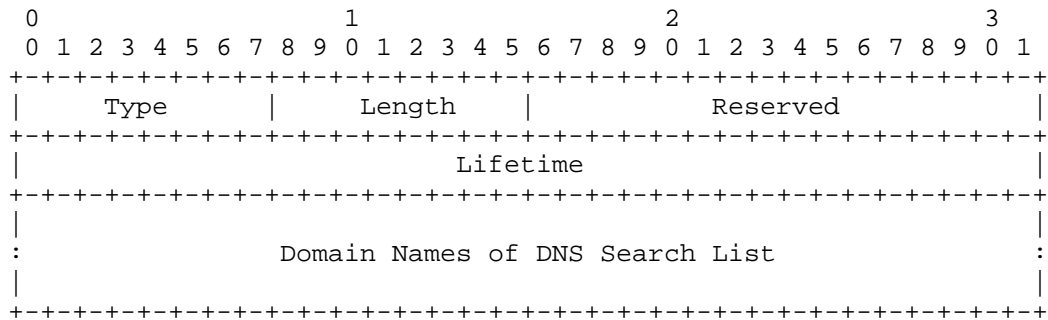


Figure 9: DNS Search List Option Format

The Type field is 31.

The Length field specifies the length of the option (including the Type and Length fields) in units of 8 octets. Its value MUST be greater than or equal to 2. If an incoming ND message does not pass these validation checks, the entire ND message MUST be discarded.

The Lifetime field specifies the maximum time, in seconds (relative to the time the packet is sent), over which this DNSSL domain name may be used for name resolution, with a value of 0 indicating that it can no longer be used. If the Lifetime field is not equal to 0, it MUST be at least 1800 (MinRtrAdvInterval) and at most 3600 (2*MaxRtrAdvInterval). If an incoming ND message does not pass this validation check, the DNSSL Option MUST be ignored. However, the rest of the ND message MAY be processed.

The domain suffixes included in this option MUST be encoded with the simple encoding specified in Section 3.1 of [RFC1035]. Therefore, if any of the labels of a domain does not have the first two bits set to zero, the corresponding DNSSL option MUST be ignored.

12. 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.

13. Security Considerations

This document specifies sanity checks to be performed on Neighbor Discovery options. By enforcing the checks specified in this document, a number of pathological behaviors (including some leading to Denial of Service scenarios) are eliminated.

14. Acknowledgements

Thanks to Jinmei Tatuya for his careful review and comments.

15. References

15.1. Normative References

- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.
- [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.
- [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.
- [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.
- [RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.

15.2. Informative References

- [FreeBSD-rtsold]
FreeBSD, , "rtsold(8) remote buffer overflow vulnerability", 2014,
<<https://www.freebsd.org/security/advisories/FreeBSD-SA-14:20.rtsold.asc>>.

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

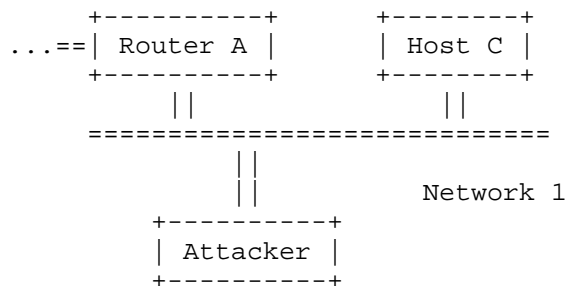


Figure 10: Unicast Forwarding Loop

In Figure 10, an off-net attacker sends Router A a crafted ND message. The ND message contains the following:

- o A Target Address, set the IPv6 address of Host C
- o A TLLA Option, set to link-layer address of Router A's interface to Network 1

The ND message causes Router A to map Host C's IPv6 address to the link layer address of its own 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 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)

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

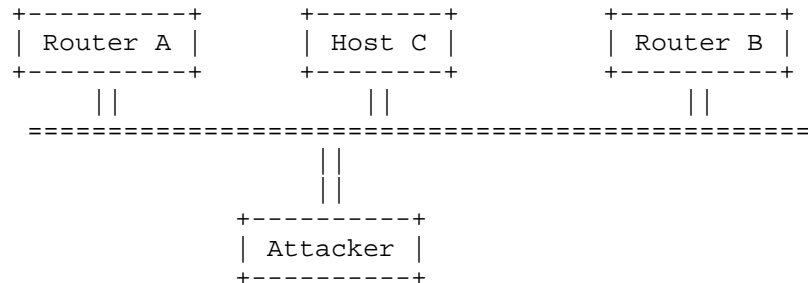


Figure 11: Broadcast Forwarding Loop

In Figure 11, 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.

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 12 describes the amplification factor for this scenario :

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

Figure 12: 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

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.

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

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: April 11, 2015

F. Gont
SI6 Networks / UTN-FRH
A. Cooper
Cisco
D. Thaler
Microsoft
W. Liu
Huawei Technologies
October 8, 2014

Recommendation on Stable IPv6 Interface Identifiers
draft-ietf-6man-default-iids-01

Abstract

The IPv6 addressing architecture defines Modified EUI-64 format Interface Identifiers, and the existing IPv6 over various link-layers specify how such identifiers are derived from the underlying link-layer address (e.g., an IEEE LAN MAC address) when employing IPv6 Stateless Address Autoconfiguration (SLAAC). The security and privacy implications of embedding hardware addresses in the Interface Identifier have been known and understood for some time now, and some popular IPv6 implementations have already deviated from such schemes to mitigate these issues. This document changes the recommended default Interface Identifier generation scheme to that specified in RFC7217, and recommends against embedding hardware addresses in IPv6 Interface Identifiers. It formally updates RFC2464, RFC2467, RFC2470, RFC2491, RFC2492, RFC2497, RFC2590, RFC3146, RFC3572, RFC4291, RFC4338, RFC4391, RFC5072, and RFC5121, which require IPv6 Interface Identifiers to be derived from the underlying link-layer address.

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 April 11, 2015.

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. Generation of IPv6 Interface Identifiers	3
4. Future Work	4
5. IANA Considerations	4
6. Security Considerations	4
7. Acknowledgements	4
8. References	5
8.1. Normative References	5
8.2. Informative References	6
Authors' Addresses	7

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 hardware address (e.g., an IEEE LAN MAC address).

The security and privacy implications of embedding a hardware address in an IPv6 Interface ID have been known for some time now, and are discussed in great detail in

[I-D.ietf-6man-ipv6-address-generation-privacy]; they include:

- o Network activity correlation
- o Location tracking

- o Address scanning
- o Device-specific vulnerability exploitation

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

As a result of the aforementioned issues, this document recommends the implementation of an alternative scheme ([RFC7217]) as the default stable Interface-ID generation scheme, such that the aforementioned issues are mitigated.

NOTE: [RFC4291] defines the "Modified EUI-64 format" for Interface identifiers. 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 interface identifier in the Modified EUI-64 format.

2. Terminology

Stable address:

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

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

3. Generation of IPv6 Interface Identifiers

Nodes SHOULD NOT employ IPv6 address generation schemes that embed the underlying hardware address in the Interface Identifier. Namely, nodes SHOULD NOT generate Interface Identifiers with the schemes specified in [RFC2464], [RFC2467], [RFC2470], [RFC2491], [RFC2492], [RFC2497], [RFC2590], [RFC3146], [RFC3572], [RFC4338], [RFC4391], [RFC5121], and [RFC5072].

Nodes SHOULD implement and employ [RFC7217] as the default scheme for generating stable IPv6 addresses with SLAAC.

Future specifications SHOULD NOT specify IPv6 address generation schemes that embed the underlying hardware address in the Interface Identifier.

4. Future Work

At the time of this writing, the mechanisms specified in the following documents are not compatible with the recommendations in this document:

- o RFC 6282 [RFC6282]
- o RFC 4944 [RFC4944]
- o RFC 6755 [RFC6775]

It is expected that that future revisions or updates of these documents will address the aforementioned issues such that the requirements in this documents can be enforced.

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 document recommends [RFC7217] as the default scheme for generating IPv6 stable addresses with SLAAC, such that the security and privacy issues of Interface IDs that embed hardware addresses are mitigated.

7. Acknowledgements

The authors would like to thank Erik Nordmark and Ray Hunter for providing a detailed review of this document.

The authors would like to thank (in alphabetical order) Fred Baker, 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, Bob Hinden, Jahangir Hossain, Jonathan Hui, Ray Hunter, Sheng Jiang, Roger Jorgensen, Dan Luedtke, George Mitchel, Erik Nordmark, Simon Perreault, Tom Petch, Alexandru Petrescu, Michael Richardson, Arturo Servin, Mark Smith, Tom Taylor, Ole Troan, Tina Tsou, and Randy Turner, 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, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [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., Schuler, P., and M. Jork, "IPv6 over ATM Networks", RFC 2492, January 1999.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, April 2014.
- [RFC2491] Armitage, G., Schuler, P., Jork, M., and G. Harter, "IPv6 over Non-Broadcast Multiple Access (NBMA) networks", RFC 2491, January 1999.
- [RFC2497] Souvatzis, I., "Transmission of IPv6 Packets over ARCnet Networks", RFC 2497, January 1999.
- [RFC2590] Conta, A., Malis, A., and M. Mueller, "Transmission of IPv6 Packets over Frame Relay Networks Specification", RFC 2590, May 1999.
- [RFC3146] Fujisawa, K. and A. Onoe, "Transmission of IPv6 Packets over IEEE 1394 Networks", RFC 3146, October 2001.

- [RFC3572] Ogura, T., Maruyama, M., and T. Yoshida, "Internet Protocol Version 6 over MAPOS (Multiple Access Protocol Over SONET/SDH)", RFC 3572, July 2003.
- [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.
- [RFC4391] Chu, J. and V. Kashyap, "Transmission of IP over InfiniBand (IPoIB)", RFC 4391, April 2006.
- [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.
- [RFC5072] Varada, S., Haskins, D., and E. Allen, "IP Version 6 over PPP", RFC 5072, September 2007.
- [RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, September 2011.
- [RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, 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.

8.2. Informative References

- [I-D.ietf-6man-ipv6-address-generation-privacy]
Cooper, A., Gont, F., and D. Thaler, "Privacy Considerations for IPv6 Address Generation Mechanisms", draft-ietf-6man-ipv6-address-generation-privacy-01 (work in progress), February 2014.
- [Microsoft]
Davies, J., "Understanding IPv6, 3rd. ed", page 83, Microsoft Press, 2012, <<http://it-ebooks.info/book/1022/>>.

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: April 13, 2015

A. Cooper
Cisco
F. Gont
Huawei Technologies
D. Thaler
Microsoft
October 10, 2014

Privacy Considerations for IPv6 Address Generation Mechanisms
draft-ietf-6man-ipv6-address-generation-privacy-02.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 April 13, 2015.

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. 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	6
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	12
5.1. Geographic Location	12
5.2. Network Operation	12
5.3. Compliance	13
5.4. Intellectual Property Rights (IPRs)	13
6. Security Considerations	13
7. IANA Considerations	13
8. Acknowledgements	13
9. Informative References	13
Authors' Addresses	15

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. 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 [RFC1972][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
 - * IPv4 address and port [RFC4380]

Deriving the IID from a globally unique IEEE identifier [RFC2462] was one of the earliest mechanisms developed. A number of privacy and security issues related to the interface IDs 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, 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 network. 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 network.

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 network 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 network (meaning that the Interface ID will remain unchanged as long as the node stays on the same network, but may change when the node moves from one network 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 target host's IID. This can be achieved by a number of different attackers: 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 network as the target (such as a conference network or any public network); 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 because MAC addresses are much more permanent than, say, DHCP leases. MAC addresses tend to last roughly the lifetime of a device's network interface, allowing correlation on the order of years, compared to days for DHCP.

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 networks (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 network segment, 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 Address Unreachable 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 was supposed to 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., in DNS advertisements).

[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 handsets 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.

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 network	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 Interface ID. For example, one would expect "low byte" Interface IDs to be more widely reused than, for example, Interface IDs where the whole 64-bits follow some pattern that is unique to a specific organization. Widely reused Interface IDs 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 networks 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 network, 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 a mechanism that generates a unique random IID for each network. A host that stays connected to the same network 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 networks even if the host only makes use of temporary addresses on those other networks; the attacker can actively probe the other networks 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 the specific DHCPv6 server software being employed. We note that 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."

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].

5. Miscellaneous Issues with IPv6 addressing

5.1. Geographic Location

Since IPv6 subnets have unique prefixes, they reveal some information about the location of the subnet, just as IPv4 addresses do. Hiding this information is one motivation for using NAT in IPv6 (see RFC 5902 section 2.4).

5.2. Network Operation

It is generally agreed that IPv6 addresses that vary over time in a specific network 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].

5.3. Compliance

Some IPv6 compliance testing suites required (and might still require) implementations to support MAC-derived suffixes 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.

5.4. 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, Tim Chown, Rich Draves, Robert Moskowitz, Erik Nordmark, and James Woodyatt for providing valuable comments on earlier versions of this document.

9. 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.

[I-D.ietf-opsec-ipv6-host-scanning]

Gont, F. and T. Chown, "Network Reconnaissance in IPv6 Networks", draft-ietf-opsec-ipv6-host-scanning-04 (work in progress), June 2014.

- [KAME-CGA] KAME, "The KAME IPR policy and concerns of some technologies which have IPR claims", 2005.
- [Microsoft] Microsoft, "IPv6 interface identifiers", 2013.
- [Panopticlick] Electronic Frontier Foundation, "Panopticlick", 2011.
- [RFC1972] Crawford, M., "A Method for the Transmission of IPv6 Packets over Ethernet Networks", RFC 1972, August 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2462] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.
- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.
- [RFC3041] Narten, T. and R. Draves, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 3041, January 2001.
- [RFC3314] Wasserman, M., "Recommendations for IPv6 in Third Generation Partnership Project (3GPP) Standards", RFC 3314, September 2002.
- [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.
- [RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003.
- [RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.
- [RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, February 2006.

- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, September 2007.
- [RFC5991] Thaler, D., Krishnan, S., and J. Hoagland, "Teredo Security Updates", RFC 5991, September 2010.
- [RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, April 2011.
- [RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, July 2013.
- [RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, April 2014.

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

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

IPv6 maintenance Working Group (6man)
Internet-Draft
Updates: 2460 (if approved)
Intended status: Best Current Practice
Expires: November 1, 2014

F. Gont
SI6 Networks / UTN-FRH
April 30, 2014

Security Implications of Predictable Fragment Identification Values
draft-ietf-6man-predictable-fragment-id-01

Abstract

IPv6 specifies the Fragment Header, which is employed for the fragmentation and reassembly mechanisms. The Fragment Header contains an "Identification" field which, together with the IPv6 Source Address and the IPv6 Destination Address of a packet, identifies fragments that correspond to the same original datagram, such that they can be reassembled together at the receiving host. The only requirement for setting the "Identification" value is that it must be different than that employed for any other fragmented packet sent recently with the same Source Address and Destination Address. Some implementations use simple a global counter for setting the Identification field, thus leading to predictable values. This document analyzes the security implications of predictable Identification values, and updates RFC 2460 specifying additional requirements for setting the Identification field of the Fragment Header, such that the aforementioned security implications are mitigated.

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 November 1, 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. Security Implications of Predictable Fragment Identification values	3
4. Updating RFC 2460	6
5. Constraints for the selection of Fragment Identification Values	6
6. Algorithms for Selecting Fragment Identification Values . . .	7
6.1. Per-destination counter (initialized to a random value) .	7
6.2. Randomized Identification values	8
6.3. Hash-based Fragment Identification selection algorithm .	9
7. IANA Considerations	11
8. Security Considerations	11
9. Acknowledgements	11
10. References	11
10.1. Normative References	11
10.2. Informative References	12
Appendix A. Information leakage produced by vulnerable implementations	13
Appendix B. Survey of Fragment Identification selection algorithms employed by popular IPv6 implementations	15
Author's Address	16

1. Introduction

IPv6 specifies the Fragment Header, which is employed for the fragmentation and reassembly mechanisms. The Fragment Header contains an "Identification" field which, together with the IPv6 Source Address and the IPv6 Destination Address of a packet, identifies fragments that correspond to the same original datagram, such that they can be reassembled together at the receiving host.

The only requirement for setting the "Identification" value is that it must be different than that employed for any other fragmented packet sent recently with the same Source Address and Destination Address.

The most trivial algorithm to avoid reusing Fragment Identification values too quickly is to maintain a global counter that is incremented for each fragmented packet that is transmitted. However, this trivial algorithm leads to predictable Identification values, which can be leveraged to performing a variety of attacks.

Section 3 of this document analyzes the security implications of predictable Identification values. Section 4 updates RFC 2460 by adding the requirement that IPv6 Fragment Identification values must not be predictable by an off-path attacker. Section 5 discusses constraints in the possible algorithms for selecting Fragment Identification values. Section 6 specifies a number of algorithms that could be used for generating Identification values. Finally, Appendix B contains a survey of the Fragment Identification algorithms employed by popular IPv6 implementations.

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. Security Implications of Predictable Fragment Identification values

Predictable Identification values result in an information leakage that can be exploited in a number of ways. Among others, they may potentially be exploited to:

- o determine the packet rate at which a given system is transmitting information,
- o perform stealth port scans to a third-party,
- o uncover the rules of a number of firewalls,
- o count the number of systems behind a middle-box,
- o perform Denial of Service (DoS) attacks, or,
- o perform data injection attacks against transport or application protocols

[CPNI-IPv6] contains a detailed analysis of possible vulnerabilities introduced by predictable Fragment Identification values. In summary, their security implications are very similar to those of predictable Identification values in IPv4.

[Sanfilippo1998a] originally pointed out how the IPv4 Identification field could be examined to determine the packet rate at which a given system is transmitting information. Later, [Sanfilippo1998b] describes how a system with such an implementation could be used to perform a stealth port scan to a third (victim) host. [Sanfilippo1999] explains how to exploit this implementation strategy to uncover the rules of a number of firewalls. [Bellovin2002] explains how the IPv4 Identification field can be exploited to count the number of systems behind a NAT. [Fyodor2004] is an entire paper on most (if not all) the ways to exploit the information provided by the Identification field of the IPv4 header (and these results apply in a similar way to IPv6). [Zalewski2003] originally envisioned the exploitation of IP fragmentation for performing data injection attacks against upper-layer protocols. [Herzberg2013] explores the use of IPv4/IPv6 fragmentation and predictable Identification values for performing DNS cache poisoning attacks in great detail. [RFC6274] covers the security implications of the IPv4 case in detail.

One key difference between the IPv4 case and the IPv6 case is that in IPv4 the Identification field is part of the fixed IPv4 header (and thus usually set for all packets), while in IPv6 the Identification field is present only in those packets that carry a Fragment Header. As a result, successful exploitation of the IPv6 Fragment Identification field depends on two different factors:

- o vulnerable IPv6 Fragment Identification generators, and,
- o the ability of an attacker to trigger the use of IPv6 fragmentation for packets sent from/to the victim node

As noted in the previous section, some implementations have been known to use predictable Fragment Identification values. For instance, Appendix B of this document shows that recent versions of a number of popular IPv6 implementations have employed predictable values for the IPv6 Fragment Identification.

Additionally, we note that RFC 1981 [RFC1981] states that when an ICMPv6 Packet Too Big error message advertising an MTU smaller than 1280 bytes is received, the receiving host is not required to reduce the Path-MTU for the corresponding destination address, but must simply include a Fragment Header in all subsequent packets sent to that destination. This triggers the use of the so-called IPv6

"atomic fragments" [RFC6946]: IPv6 fragments with a Fragment Offset equal to 0, and the "M" ("More fragments") bit clear.

Thus, an attacker can usually cause a victim host to "fragment" its outgoing packets by sending it a forged ICMPv6 'Packet Too Big' (PTB) error message that advertises a Next-Hop MTU smaller than 1280 bytes.

There are a number of aspects that should be considered, though:

- o All the implementations the author is aware of record the Path-MTU information on a per-destination basis. Thus, an attacker can only cause the victim to enable fragmentation for those packets sent to the Source Address of IPv6 packet embedded in the payload of the ICMPv6 PTB message. However, we note that Section 5.2 of [RFC1981] notes that an implementation could maintain a single system-wide PMTU value to be used for all packets originating from that nodes. Clearly, such an implementations would exacerbate the problem of any attacks based on PMTUD [RFC5927] or IPv6 fragmentation.
- o If the victim node implements some of the counter-measures for ICMP attacks described in RFC 5927 [RFC5927], it might be difficult for an attacker to cause the victim node to use fragmentation for its outgoing packets. However, many current implementations fail to enforce these validation checks. For example, Linux 2.6.38-8 does not even require received ICMPv6 error messages to correspond to ongoing communication instances.

Implementations that employ predictable Identification values and also fail to enforce validation checks on ICMPv6 error messages become vulnerable to the same type of attacks that can be exploited with IPv4 fragmentation, discussed earlier in this section.

One possible way in which predictable Identification values could be leveraged for performing a Denial of Service (DoS) attack is as follows: Let us assume that Host A is communicating with Host B, and that an attacker wants to DoS such communication. The attacker would learn the the Identification value currently in use by Host A, possibly by sending any packet that would elicit a fragmented response (e.g., an ICCPMv6 echo request with a large payload). The attacker would then send a forged ICMPv6 Packet Too Big error message to Host A (with the IPv6 Destination Address of the embedded IPv6 packet set to the IPv6 address of a Host B), such that any subsequent packets sent by Host A to Host B include a Fragment Header. Finally, the attacker send forged IPv6 fragments to the Host B, with their IPv6 Source Address set to that of Host A, and Identification values that would result in collisions with the Identification values employed for the legitimate traffic sent by Host A to Host B. If Host

B discards fragments that result in collisions of Identification values (e.g., such fragments overlap, and the host implements [RFC5722]), the attacker could simply trash the Identification space by sending multiple forged fragments with different Identification values, such that any subsequent packets from Host A to Host B are discarded at Host B as a result of the malicious fragments sent by the attacker.

NOTES:

For example, Linux 2.6.38-10 is vulnerable to the aforementioned issue.

[RFC6946] describes an improved processing of these packets that would eliminate this specific attack vector, at least in the case of TCP connections that employ the Path-MTU Discovery mechanism.

The previous attack scenario is simply included to illustrate the problem of employing predictable fragment Identification values. We note that regardless of the attacker's ability to cause a victim host to employ fragmentation when communicating with third-parties, use of predictable Identification values makes communication flows that employ fragmentation vulnerable to any fragmentation-based attacks.

4. Updating RFC 2460

Hereby we update RFC 2460 [RFC2460] as follows:

The Identification value of the Fragment Header MUST NOT be predictable by an off-path attacker.

5. Constraints for the selection of Fragment Identification Values

The "Identification" field of the Fragmentation Header is 32-bits long. However, when translators [RFC6145] are employed, the "effective" length of the IPv6 Fragment Identification field is 16 bits.

NOTE: [RFC6145] notes that, when translating in the IPv6-to-IPv4 direction, "if there is a Fragment Header in the IPv6 packet, the last 16 bits of its value MUST be used for the IPv4 identification value". This means that the high-order 16 bits are effectively ignored.

As a result, at least during the IPv6/IPv4 transition/co-existence phase, it is probably safer to assume that only the low-order 16 bits of the IPv6 Fragment Identification are of use to the destination system.

Regarding the selection of Fragment Identification values, the only requirement specified in [RFC2460] is that the Fragment Identification must be different than that of any other fragmented packet sent recently with the same Source Address and Destination Address. Failure to comply with this requirement could lead to the interoperability problems discussed in [RFC4963].

From a security standpoint, unpredictable Identification values are desirable. However, this is somewhat at odds with the "re-use" requirements specified in [RFC2460].

Finally, since Fragment Identification values need to be selected for each outgoing datagram that requires fragmentation, the performance aspect should be considered when choosing an algorithm for the selection of Fragment Identification values.

6. Algorithms for Selecting Fragment Identification Values

This section specifies a number of algorithms that MAY be used for selecting Fragment Identification values.

6.1. Per-destination counter (initialized to a random value)

1. Whenever a packet must be sent with a Fragment Header, the sending host should perform a look-up in the Destinations Cache an entry corresponding to the Destination Address of the packet.
2. If such an entry exists, it contains the last Fragment Identification value used for that Destination. Therefore, such value should be incremented by 1, and used for setting the Fragment Identification value of the outgoing packet. Additionally, the updated value should be recorded in the corresponding entry of the Destination Cache.
3. If such an entry does not exist, it should be created, and the "Identification" value for that destination should be initialized with a random value (e.g., with a pseudorandom number generator), and used for setting the Identification field of the Fragment Header of the outgoing packet.

The advantages of this algorithm are:

- o It is simple to implement, with the only complexity residing in the Pseudo-Random Number Generator (PRNG) used to initialize the "Identification" value contained in each entry of the Destinations Cache.

- o The "Identification" re-use frequency will typically be lower than that achieved by a global counter (when sending traffic to multiple destinations), since this algorithm uses per-destination counters (rather than a single system-wide counter).
- o It has good performance properties (once the corresponding entry in the Destinations Cache has been created, each subsequent "Identification" value simply involves the increment of a counter).

The possible drawbacks of this algorithm are:

- o If as a result of resource management an entry of the Destinations Cache must be removed, the last Fragment Identification value used for that Destination will be lost. Thus, subsequent traffic to that destination would cause that entry to be re-created and re-initialized to random value, thus possibly leading to Fragment Identification "collisions".
- o Since the Fragment Identification values are predictable by the destination host, a vulnerable host might possibly leak to third-parties the Fragment Identification values used by other hosts to send traffic to it (i.e., Host B could leak to Host C the Fragment Identification values that Host A is using to send packets to Host B). Appendix A describes one possible scenario for such leakage in detail.

6.2. Randomized Identification values

Clearly, use of a Pseudo-Random Number Generator for selecting the Fragment Identification would be desirable from a security standpoint. With such a scheme, the Fragment Identification of each fragmented datagram would be selected as:

```
Identification = random()
```

where "random()" is the PRNG.

The specific properties of such scheme would clearly depend on the specific PRNG algorithm used. For example, some PRNGs may result in higher Fragment Identification reuse frequencies than others, in the same way as some PRNGs may be more expensive (in terms of processing requirements and/or implementation complexity) than others.

Discussion of the properties of possible PRNGs is considered out of the scope of this document. However, we do note that some PRNGs employed in the past by some implementations have been found to be

predictable [Klein2007]. Please see [RFC4086] for randomness requirements for security.

6.3. Hash-based Fragment Identification selection algorithm

Another alternative is to implement a hash-based algorithm similar to that specified in [RFC6056] for the selection of transport port numbers. With such a scheme, the Fragment Identification value of each fragment datagram would be selected with the expression:

$$\text{Identification} = F(\text{Src IP}, \text{Dst IP}, \text{secret1}) + \text{counter}[G(\text{src IP}, \text{Dst Pref}, \text{secret2})]$$

where:

Identification:

Identification value to be used for the fragmented datagram

F():

Hash function

Src IP:

IPv6 Source Address of the datagram to be fragmented

Dst IP:

IPv6 Destination Address of the datagram to be fragmented

secret1:

Secret data unknown to the attacker

counter[]:

System-wide array of 32-bit counters (e.g. with 8K elements or more)

G():

Hash function. May or may not be the same hash function as that used for F()

Dst Pref:

IPv6 "Destination Prefix" of datagram to be fragmented (can be assumed to be the first eight bytes of the Destination Address of such packet). Note: the "Destination Prefix" (rather than Destination Address) is used, such that the ability of an attacker of searching the "increments" space by using multiple addresses of the same subnet is reduced.

secret1:

Secret data unknown to the attacker

NOTE: counter[G(src IP, Dst Pref, secret2)] should be incremented by one each time an Identification value is selected.

The advantages of this algorithm are:

- o The "Identification" re-use frequency will typically be lower than that achieved by a global counter (when sending traffic to multiple destinations), since this algorithm uses multiple system-wide counters (rather than a single system-wide counter). The extent to which the re-use frequency will be lower will depend on the number of elements in counter[], and the number of other active flows that result in the same value of G() (and hence cause the same counter to be incremented for each fragmented datagram that is sent).
- o It is possible to implement the algorithm such that good performance is achieved. For example, the result of F() could be stored in the Destinations Cache (such that it need not be recomputed for each packet that must be sent) along with the computed "index"/argument for counter[].

NOTE: If this implementation approach is followed, and an entry of the Destinations Cache must be removed as a result of resource management, the last Fragment Identification value used for that Destination will **not** be lost. This is an improvement over the algorithm specified in Section 6.1.

The possible drawbacks of this algorithm are:

- o Since the Fragment Identification values are predictable by the destination host, a vulnerable host could possibly leak to third-parties the Fragment Identification values used by other hosts to send traffic to it (i.e., Host B could leak to Host C the Fragment Identification values that Host A is using to send packets to Host B). Appendix A describes a possible scenario in which that information leakage could take place. We note, however, that this algorithm makes the aforementioned attack less reliable for the attacker, since each counter could be possibly shared by multiple traffic flows (i.e., packets destined to other destinations might cause the same counter to be incremented).

This algorithm might be preferable (over the one specified in Section 6.1) in those scenarios in which a node is expected to communicate with a large number of destinations, and thus it is desirable to limit the amount of information to be maintained in memory.

NOTE: In such scenarios, if the algorithm specified in Section 6.1 were implemented, entries from the Destinations Cache might need to be pruned frequently, thus increasing the risk of fragment Identification collisions.

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 the security implications of predictable Fragment Identification values, and updates RFC 2460 such that Fragment Identification values are required to be unpredictable by off-path attackers, hence mitigating the aforementioned security implications.

A number of possible algorithms are specified, to provide some implementation alternatives to implementers. However, the selection of a specific algorithm is left to implementers. We note that the selection of such an algorithm usually implies a number of trade-offs (security, performance, implementation complexity, interoperability properties, etc.).

9. Acknowledgements

The author would like to thank Ivan Arce for proposing the attack scenario described in Appendix A.

The author would like to thank Ivan Arce and Dave Thaler for providing valuable comments on earlier versions of this document.

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

- [RFC1981] McCann, J., Deering, S., and J. Mogul, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.
- [RFC5722] Krishnan, S., "Handling of Overlapping IPv6 Fragments", RFC 5722, December 2009.
- [RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-Protocol Port Randomization", BCP 156, RFC 6056, January 2011.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.
- [RFC6946] Gont, F., "Processing of IPv6 "Atomic" Fragments", RFC 6946, May 2013.

10.2. Informative References

- [RFC4963] Heffner, J., Mathis, M., and B. Chandler, "IPv4 Reassembly Errors at High Data Rates", RFC 4963, July 2007.
- [RFC5927] Gont, F., "ICMP Attacks against TCP", RFC 5927, July 2010.
- [RFC6274] Gont, F., "Security Assessment of the Internet Protocol Version 4", RFC 6274, July 2011.
- [Bellovin2002] Bellovin, S., "A Technique for Counting NATted Hosts", IMW'02 Nov. 6-8, 2002, Marseille, France, 2002.
- [CPNI-IPv6] Gont, F., "Security Assessment of the Internet Protocol version 6 (IPv6)", UK Centre for the Protection of National Infrastructure, (available on request).
- [Fyodor2004] Fyodor, , "Idle scanning and related IP ID games", 2004, <<http://www.insecure.org/nmap/idlescan.html>>.
- [Herzberg2013] Herzberg, A. and H. Shulman, "Fragmentation Considered Poisonous", Technical Report 13-03, March 2013, <<http://u.cs.biu.ac.il/~herzbea/security/13-03-frag.pdf>>.

[Klein2007]

Klein, A., "OpenBSD DNS Cache Poisoning and Multiple O/S Predictable IP ID Vulnerability", 2007,
<http://www.trusteer.com/files/OpenBSD_DNS_Cache_Poisoning_and_Multiple_OS_Predictable_IP_ID_Vulnerability.pdf>.

[Sanfilippo1998a]

Sanfilippo, S., "about the ip header id", Post to Bugtraq mailing-list, Mon Dec 14 1998,
<<http://www.kyuzz.org/antirez/papers/ipid.html>>.

[Sanfilippo1998b]

Sanfilippo, S., "Idle scan", Post to Bugtraq mailing-list, 1998, <<http://www.kyuzz.org/antirez/papers/dumbscan.html>>.

[Sanfilippo1999]

Sanfilippo, S., "more ip id", Post to Bugtraq mailing-list, 1999,
<<http://www.kyuzz.org/antirez/papers/moreipid.html>>.

[SI6-IPv6]

"SI6 Networks' IPv6 toolkit",
<<http://www.si6networks.com/tools/ipv6toolkit>>.

[Zalewski2003]

Zalewski, M., "A new TCP/IP blind data injection technique?", Post to Bugtraq mailing-list, Thu, 11 Dec 2003 00:28:28 +0100 (CET), 2003,
<<http://lcamtuf.coredump.cx/ipfrag.txt>>.

Appendix A. Information leakage produced by vulnerable implementations

Section 3 provides a number of references describing a number of ways in which a vulnerable implementation may reveal the Fragment Identification values to be used in subsequent packets, thus opening the door to a number of attacks. In all of those scenarios, a vulnerable implementation leaks/reveals its own Identification number.

This section presents a different case, in which a vulnerable implementation leaks/reveals the Identification number of a non-vulnerable implementation. That is, a vulnerable implementation (Host A) leaks the current Fragment Identification value in use by a third-party host (Host B) to send fragmented datagrams from Host B to Host A.

For the most part, this section is included to illustrate how a vulnerable implementation might be leveraged to leak-out the

Fragment Identification value of an otherwise non-vulnerable implementation. This section might be removed in future revisions of this document.

The following scenarios assume:

Host A:

Is an IPv6 host that implements the recommended Fragment Identification algorithm (Section 6.1), implements [RFC5722], but does not implement [RFC6946].

Host B:

Victim node. Selected the Fragment Identification values from a global counter.

Host C:

Attacker. Can forge the IPv6 Source Address of his packets at will.

In the following scenarios, large ICMPv6 Echo Request packets are employed to "sample" the Fragment Identification value of a host. We note that while the figures show only one packet for the ICMPv6 Echo Request and the ICMPv6 Echo Response, each of those packets will typically comprise two fragments, such that the resulting datagram is larger than the MTU of the networks to which Host B and Host C are attached.

In the lines #1-#2 (and lines #8-#9), the attacker samples the current Fragment Identification value. In line #3, the attacker sends a forged TCP SYN segment to Host A. If corresponding TCP port is closed, and the attacker fails when trying to produce a collision of Fragment Identifications (see line #4), the following packet exchange might take place:

A	B	C
#1	<----- Echo Req #1 ----->	
#2	--- Echo Resp #1, FID=5000 --->	
#3	<----- SYN #1, src= B ----->	
#4	<--- SYN/ACK, FID=42 src = A---	
#5	---- SYN/ACK, FID=9000 ---->	
#6	<----- RST, FID= 5001 ----->	
#7	<----- RST, FID= 5002 ----->	
#8	<----- Echo Req #2 ----->	
#9	--- Echo Resp #2, FID=5003 --->	

On the other hand, if the attacker succeeds to produce a collision of Fragment Identification values, the following packet exchange could take place:

```

      A                               B                               C

#1                                     <----- Echo Req #1 ----->
#2                                     --- Echo Resp #1, FID=5000 --->
#3 <----- SYN #1, src= B ----->
#4 <-- SYN/ACK, FID=9000 src=A ---
#5 ---- SYN/ACK, FID=9000 ---->
      ... (RFC5722) ...
#6                                     <----- Echo Req #2 ----->
#7                                     ---- Echo Resp #2, FID=5001 -->

```

Clearly, the Fragment Identification value sampled by from the second ICMPv6 Echo Response packet ("Echo Resp #2") implicitly indicates whether the Fragment Identification in the forged SYN/ACK (see line #4 in both figures) was the current Fragment Identification in use by Host A.

As a result, the attacker could employ this technique to learn the current Fragment Identification value used by host A to send packets to host B, even when Host A itself has a non-vulnerable implementation.

Appendix B. Survey of Fragment Identification selection algorithms employed by popular IPv6 implementations

This section includes a survey of the Fragment Identification selection algorithms employed in some popular operating systems.

The survey was produced with the SI6 Networks IPv6 toolkit [SI6-IPv6].

Operating System	Algorithm
FreeBSD 9.0	Unpredictable (Random)
Linux 3.0.0-15	Predictable (Global Counter, Init=0, Incr=1)
Linux-current	Unpredictable (Per-dest Counter, Init=random, Incr=1)
NetBSD 5.1	Unpredictable (Random)
OpenBSD-current	Random (SKIP32)
Solaris 10	Predictable (Per-dst Counter, Init=0, Incr=1)
Windows XP SP2	Predictable (Global Counter, Init=0, Incr=2)
Windows Vista (Build 6000)	Predictable (Global Counter, Init=0, Incr=2)
Windows 7 Home Premium	Predictable (Global Counter, Init=0, Incr=2)

Table 1: Fragment Identification algorithms employed by different OSes

In the text above, "predictable" should be taken as "easily guessable by an off-path attacker, by sending a few probe packets".

Author's Address

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>

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2015

S. Jiang
D. Zhang
Huawei Technologies Co., Ltd
S. Krishnan
Ericsson
July 4, 2014

CGA SEC Option for Secure Neighbor Discovery Protocol
draft-jiang-6man-cga-sec-option-00

Abstract

A Cryptographically Generated Address is an IPv6 addresses binding with a public/private key pair. It is a vital component of Secure Neighbor Discovery (SeND) protocol. The current SeND specifications are lack of procedures to specify the Sec bits. A new SEC option is defined accordingly to address this issue.

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 January 5, 2015.

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. Requirements Language	2
3. CGA SEC Option	3
4. Host Behavior	3
5. Security Considerations	3
6. IANA Considerations	3
7. Acknowledgements	4
8. References	4
8.1. Normative References	4
8.2. Informative References	4
Authors' Addresses	4

1. Introduction

Cryptographically Generated Addresses (CGA, [RFC3972]) are used to make sure that the sender of a Neighbor Discovery message is the "owner" of the claimed address. Although it is not mandatory, it is a vital component of Secure Neighbor Discovery (SeND, [RFC3971]) protocol. After CGA has been defined, as an independent security property, many other CGA usages have been proposed and defined, such as Enhanced Route Optimization for Mobile IPv6 [RFC4866], Site Multihoming by IPv6 Intermediation (SHIM6) [RFC5533], etc.

SEC bits are an important parameter in the generation of CGAs. Particularly, SEC values are used to artificially introduce additional difficulty in the CGA generation process in order to provide additional protection against brute force attacks. Therefore, in different environments, host may be required to use different SEC bits in the generation of their CGAs. However, the base SeND protocol fails to distribute the SEC values to the hosts. As a result, the network administration cannot propagate any requirements regarding to SEC value of host-generated CGA addresses. In order to fill this gap, a new CGA SEC Option, is defined in this document.

2. Requirements Language

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

3. CGA SEC Option

CGA SEC Option is used to indicate on link hosts the lowest CGA SEC value they SHOULD use.

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   OPTION_CGA_SEC_OPTION   |   option-len   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   SEC bits   |
+---+---+---+---+---+

```

option-code OPTION_CGA_SEC_OPTION (TBA1)

option-len 1.

SEC bits The value of SEC bits is specified in [RFC3972].

4. Host Behavior

On receiving the CGA SEC Option with a recommended SEC value, a host SHOULD use a CGA with the recommended or higher SEC value. If choosing a CGA with a SEC value lower than the recommended, the host MAY take the risk that it is not able to use full network capabilities. The network may consider the hosts that use CGAs with lower SEC values as unsecure users and decline some or all network services.

5. Security Considerations

This document extends SeND with a CGA SEC Option to transport SEC bits used in the generation of GCAs, which enables administrators to specify and adjust the security level of the CGAs used in the network. Apart from that, this approach does not introduce any significant changes to the underlying security issues considered in Section 9 of [RFC3971].

6. IANA Considerations

This document defines a new Neighbor Discovery Protocol options, which must be assigned an Option Type value within the option numbering space for Neighbor Discovery Protocol messages:

- o The CGA SEC option (TBA1), described in Section 3.

7. Acknowledgements

The authors would like to thanks the valuable comments made by members of 6man WG.

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

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.
- [RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.

8.2. Informative References

- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [RFC4866] Arkko, J., Vogt, C., and W. Haddad, "Enhanced Route Optimization for Mobile IPv6", RFC 4866, May 2007.
- [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, June 2009.

Authors' Addresses

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

Email: jiangsheng@huawei.com

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

Email: zhangdacheng@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

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: July 18, 2015

S. Jiang
Huawei Technologies Co., Ltd
D. Zhang
Alibaba Co., Ltd
S. Krishnan
Ericsson
January 14, 2015

CGA SEC Option for Secure Neighbor Discovery Protocol
draft-jiang-6man-cga-sec-option-01

Abstract

A Cryptographically Generated Address is an IPv6 addresses binding with a public/private key pair. It is a vital component of Secure Neighbor Discovery (SeND) protocol. The current SeND specifications are lack of procedures to specify the Sec bits. A new SEC option is defined accordingly to address this issue.

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 July 18, 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
2. Requirements Language	2
3. CGA SEC Option	3
4. Host Behavior	3
5. Security Considerations	3
6. IANA Considerations	3
7. Acknowledgements	4
8. References	4
8.1. Normative References	4
8.2. Informative References	4
Authors' Addresses	4

1. Introduction

Cryptographically Generated Addresses (CGA, [RFC3972]) are used to make sure that the sender of a Neighbor Discovery message is the "owner" of the claimed address. Although it is not mandatory, it is a vital component of Secure Neighbor Discovery (SeND, [RFC3971]) protocol. After CGA has been defined, as an independent security property, many other CGA usages have been proposed and defined, such as Enhanced Route Optimization for Mobile IPv6 [RFC4866], Site Multihoming by IPv6 Intermediation (SHIM6) [RFC5533], etc.

SEC bits are an important parameter in the generation of CGAs. Particularly, SEC values are used to artificially introduce additional difficulty in the CGA generation process in order to provide additional protection against brute force attacks. Therefore, in different environments, host may be required to use different SEC bits in the generation of their CGAs. However, the base SeND protocol fails to distribute the SEC values to the hosts. As a result, the network administration cannot propagate any requirements regarding to SEC value of host-generated CGA addresses. In order to fill this gap, a new CGA SEC Option, is defined in this document.

2. Requirements Language

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

3. CGA SEC Option

CGA SEC Option is used to indicate on link hosts the lowest CGA SEC value they SHOULD use. It SHOULD be contained in and only in the Router Advertisement Message [RFC4861].

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|      OPTION_CGA_SEC_OPTION      |      option-len      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      SEC bits      |
+-----+-----+-----+-----+

```

option-code OPTION_CGA_SEC_OPTION (TBA1)

option-len 1.

SEC bits The value of SEC bits is specified in [RFC3972].

4. Host Behavior

On receiving the CGA SEC Option with a recommended SEC value, a host SHOULD use a CGA with the recommended or higher SEC value. If choosing a CGA with a SEC value lower than the recommended, the host MAY take the risk that it is not able to use full network capabilities. The network may consider the hosts that use CGAs with lower SEC values as unsecure users and decline some or all network services.

5. Security Considerations

This document extends SeND with a CGA SEC Option to transport SEC bits used in the generation of GCAs, which enables administrators to specify and adjust the security level of the CGAs used in the network. Apart from that, this approach does not introduce any significant changes to the underlying security issues considered in Section 9 of [RFC3971].

6. IANA Considerations

This document defines a new Neighbor Discovery Protocol options, which must be assigned an Option Type value within the IPv6 Neighbor Discovery Option Formats table of Internet Control Message Protocol version 6 (ICMPv6) Parameters (<http://www.iana.org/assignments/icmpv6-parameters>):

Type	Description	Reference
TBA1	CGA SEC option	This document

7. Acknowledgements

The authors would like to thanks the valuable comments made by members of 6man WG.

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

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.
- [RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

8.2. Informative References

- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [RFC4866] Arkko, J., Vogt, C., and W. Haddad, "Enhanced Route Optimization for Mobile IPv6", RFC 4866, May 2007.
- [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, June 2009.

Authors' Addresses

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

Dacheng Zhang
Alibaba Co., Ltd
9th Floor, A Area, Wentelai World Finance Centre, 1 West Dawang Road
Chaoyang District, Beijing, 100095 100025
P.R. China

Email: dacheng.zdc@alibaba-inc.com

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

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

Network Working Group
Internet Draft
Intended status: Stand Track
Expires: April 24, 2014

B. Liu
Huawei Technologies
October 21, 2013

IPv6 ND Option for Network Management Server Discovery
draft-liu-6man-nd-nms-discovery-00.txt

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 April 24, 2014.

Copyright Notice

Copyright (c) 2013 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.

Abstract

This document introduces a mechanism for devices to actively learn the NMS server address from the neighbors through IPv6 ND protocol extension. It is a good leverage of IPv6 automatic features.

This document only discusses problem/solution within the IPv6-only networks/plane.

Table of Contents

1. Introduction	3
2. Basic Approach	3
3. Scenario Description.....	3
3.1. New Devices Getting Online	3
3.2. Regarding Connectivity	4
4. Neighbor Discovery Extension for Supporting NMS Discovery	4
4.1. Option Definition	5
4.2. Sub-Options Definition	5
4.3. Option Carried in Router Advertisement Messages	6
4.4. Option Carried in Neighbor Solicit/Advertisement Messages	7
5. Security Considerations	7
6. IANA Considerations	7
7. Acknowledgments	7
8. References	7
8.1. Normative References	7

1. Introduction

NMS (Network Management System) has become a must-have component in modern networks. It could be utilized to benefit various aspects of a network. For example, the emerging router auto-configuration solutions are mostly based on NMS. If the devices could successfully connect to the NMS server(s), then auto-configuration won't be a problem.

So there is a key problem of how to discover the NMS server for the devices when they get online. Currently there are mainly two methods to solve the problem. One is to set the NMS server's IP/URL into the devices before shipping to the customer premises; the other one is the NMS actively discovering the devices through some polling mechanisms. The former one is easy to be implemented and deployed, but it lacks flexibility due to the static pre-configuration and might be error-prone for configuration when the different networks have different NMS servers; the latter one lacks the instantaneity due to the polling mechanisms need the intermediate nodes to integrate supporting features which introduce complex functions and protocols.

This document introduces a mechanism for devices to actively learn the NMS server from the neighbors through IPv6 ND protocol extension. It is a good leverage of IPv6 automatic features.

This document only discusses problem/solution within the IPv6-only scope.

2. Basic Approach

When a device gets online, we could assume that its neighbors who have already got online have learnt the NMS server's address. So it is quite easy for the new device to learn the information from its neighbor.

This document is based on the above Neighbor-Learning approach.

3. Scenario Description

3.1. New Devices Getting Online

- Adding a New Device into an Existing Network

For adding a new device into an existing network, it is very reasonable to assume that the neighbors have already connected to the NMS server. So it is obvious that the new device could easily learn the NMS server's address from neighbors.

- Deploying a New Network

In the case of deploying a new network, the NMS server address needs to be propagated to the whole network, then some kind of flooding mechanism is needed if the propagation also relies on above mentioned neighbor-learning approach. This is applicable through careful plan which might need proper order for the devices to get online successively.

The detail of the flooding mechanism is out of the scope of this document. We treat it as an assumption for the application of neighbor-learning NMS discovery.

3.2. Regarding Connectivity

- Connecting NMS after Getting Global Connectivity

Normally, address assignment is not coupled with NMS processings. Before connected to the NMS server, the devices could obtain global connectivity either through SLAAC or DHCPv6.

In this case, once the devices have learnt the NMS server address, they could directly connect to get more configurations.

- Connecting NMS before Getting Global Connectivity

In contrast, address assignment might be done through NMS in some situations. For example, the device is a backbone router, and the address has been carefully planned and pre-configured in the NMS server, when the device connect to the server, it will be assigned global address through network management processing.

In this case, after learning the NMS server address, the device might need a proxy to communicate with the server or configuring itself a ULA address and utilizing the NPTv6 processing on its neighbor or uplink router. The details are out of the scope of this document.

4. Neighbor Discovery Extension for Supporting NMS Discovery

Since ND is a basic protocol in IPv6, every router supports IPv6 would support ND, we utilize ND extension to achieve the above mentioned neighbor-learning NMS server discovery.

- o Length: 3
- o IPv6 Address: 128bit IPv6 address with zero padding behind

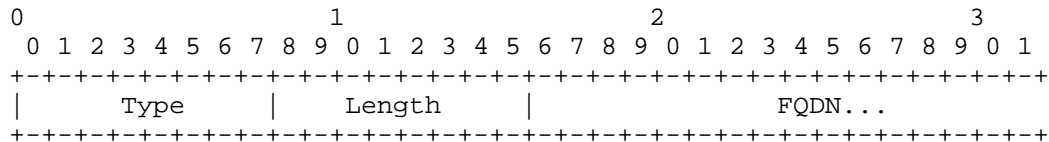


Figure 3: FQDN Sub-option of NMS Server Location

- o Type: TBD (to be assigned by IANA)
- o Length: The length of the option (including the type and length fields) in units of 8 octets.
- o FQDN: FQDN of the NMS server, variable length

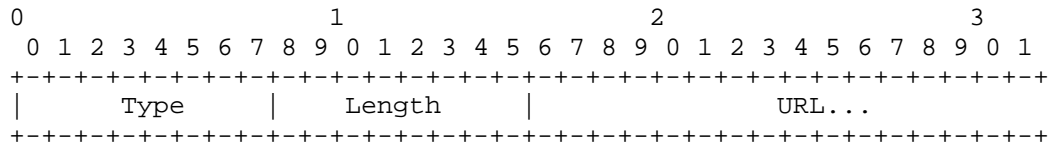


Figure 3: FQDN Sub-option of NMS Server Location

- o Type: TBD (to be assigned by IANA)
- o Length: The length of the option (including the type and length fields) in units of 8 octets.
- o URL: URL of the NMS server, variable length

4.3. Option Carried in Router Advertisement Messages

- RA-only Mode

A device discovers NMS server's address through received Router Advertisement messages which include a new option defined for carrying NMS server's address.

Since RA messages are usually generated by the gateway on a link, this approach is suitable for a hub-and-spoke subnet in which a new device joins in.

After having learnt the NMS server's address, then the device could directly connect to the server

4.4. Option Carried in Neighbor Solicit/Advertisement Messages

A device discovers NMS server's address through actively initiating Neighbor Solicit message and receiving Neighbor Advertisement messages which include the new option carrying the NMS server's address.

This approach is suitable for point-to-point or non-broad circuits.

5. Security Considerations

- Device authentication for NMS Servers

With applying the mechanism described in this document, the devices would actively connect to the NMS servers. So there might be stronger desire for the NMS servers to authenticate the devices.

- ND security

This document doesn't introduce more threats than original Neighbor Discovery protocol, so generally it aligns with the security considerations described in [RFC4861].

6. IANA Considerations

The newly defined options need IANA to assign type codes.

7. Acknowledgments

Many useful comments and contributions were made by Sheng Jiang.

8. References

8.1. Normative References

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

Authors' Addresses

Bing Liu
Huawei Technologies Co., Ltd
Q14, Huawei Campus
No.156 Beiqing Rd.
Hai-Dian District, Beijing 100095
P.R. China

Email: leo.liubing@huawei.com

Network Working Group
Internet Draft
Intended status: Stand Track
Expires: April 30, 2015

B. Liu
Huawei Technologies
October 27, 2014

IPv6 ND Option for Network Management Server Discovery
draft-liu-6man-nd-nms-discovery-01.txt

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 April 30, 2015.

Copyright Notice

Copyright (c) 2013 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.

Abstract

This document introduces a mechanism for devices to actively learn the NMS server address from the neighbors through IPv6 ND protocol extension. It is a good leverage of IPv6 automatic features.

This document only discusses problem/solution within the IPv6-only networks/plane.

Table of Contents

1. Introduction	3
2. Basic Approach	3
3. Scenario Description.....	3
3.1. New Devices Getting Online	3
3.2. Regarding Connectivity	4
4. Neighbor Discovery Extension for Supporting NMS Discovery	4
4.1. Option Definition	5
4.2. Sub-Options Definition	5
4.3. Option Carried in Router Advertisement Messages	6
4.4. Option Carried in Neighbor Solicit/Advertisement Messages	7
5. Security Considerations	7
6. IANA Considerations	7
7. Acknowledgments	7
8. References	7
8.1. Normative References	7

1. Introduction

NMS (Network Management System) has become a must-have component in modern networks. It could be utilized to benefit various aspects of a network. For example, the emerging router auto-configuration solutions are mostly based on NMS. If the devices could successfully connect to the NMS server(s), then auto-configuration won't be a problem.

So there is a key problem of how to discover the NMS server for the devices when they get online. Currently there are mainly two methods to solve the problem. One is to set the NMS server's IP/URL into the devices before shipping to the customer premises; the other one is the NMS actively discovering the devices through some polling mechanisms. The former one is easy to be implemented and deployed, but it lacks flexibility due to the static pre-configuration and might be error-prone for configuration when the different networks have different NMS servers; the latter one lacks the instantaneity due to the polling mechanisms need the intermediate nodes to integrate supporting features which introduce complex functions and protocols.

This document introduces a mechanism for devices to actively learn the NMS server from the neighbors through IPv6 ND protocol extension. It is a good leverage of IPv6 automatic features.

This document only discusses problem/solution within the IPv6-only scope.

2. Basic Approach

When a device gets online, we could assume that its neighbors who have already got online have learnt the NMS server's address. So it is quite easy for the new device to learn the information from its neighbor.

This document is based on the above Neighbor-Learning approach.

3. Scenario Description

3.1. New Devices Getting Online

- Adding a New Device into an Existing Network

For adding a new device into an existing network, it is very reasonable to assume that the neighbors have already connected to the NMS server. So it is obvious that the new device could easily learn the NMS server's address from neighbors.

- Deploying a New Network

In the case of deploying a new network, the NMS server address needs to be propagated to the whole network, then some kind of flooding mechanism is needed if the propagation also relies on above mentioned neighbor-learning approach. This is applicable through careful plan which might need proper order for the devices to get online successively.

The detail of the flooding mechanism is out of the scope of this document. We treat it as an assumption for the application of neighbor-learning NMS discovery.

3.2. Regarding Connectivity

- Connecting NMS after Getting Global Connectivity

Normally, address assignment is not coupled with NMS processing. Before connected to the NMS server, the devices could obtain global connectivity either through SLAAC or DHCPv6.

In this case, once the devices have learnt the NMS server address, they could directly connect to get more configurations.

- Connecting NMS before Getting Global Connectivity

In contrast, address assignment might be done through NMS in some situations. For example, the device is a backbone router, and the address has been carefully planned and pre-configured in the NMS server, when the device connect to the server, it will be assigned global address through network management processing.

In this case, after learning the NMS server address, the device might need a proxy to communicate with the server or configuring itself a ULA address and utilizing the NPTv6 processing on its neighbor or uplink router. The details are out of the scope of this document.

4. Neighbor Discovery Extension for Supporting NMS Discovery

Since ND is a basic protocol in IPv6, every router supports IPv6 would support ND, we utilize ND extension to achieve the above mentioned neighbor-learning NMS server discovery.

- o Length: 3
- o IPv6 Address: 128bit IPv6 address with zero padding behind

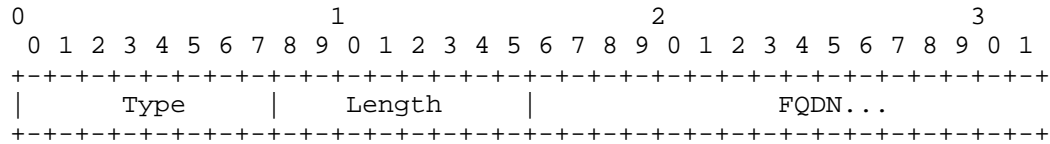


Figure 3: FQDN Sub-option of NMS Server Location

- o Type: TBD (to be assigned by IANA)
- o Length: The length of the option (including the type and length fields) in units of 8 octets.
- o FQDN: FQDN of the NMS server, variable length

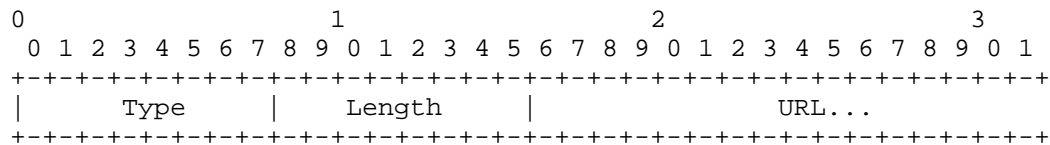


Figure 3: FQDN Sub-option of NMS Server Location

- o Type: TBD (to be assigned by IANA)
- o Length: The length of the option (including the type and length fields) in units of 8 octets.
- o URL: URL of the NMS server, variable length

4.3. Option Carried in Router Advertisement Messages

- RA-only Mode

A device discovers NMS server's address through received Router Advertisement messages which include a new option defined for carrying NMS server's address.

Since RA messages are usually generated by the gateway on a link, this approach is suitable for a hub-and-spoke subnet in which a new device joins in.

After having learnt the NMS server's address, then the device could directly connect to the server

4.4. Option Carried in Neighbor Solicit/Advertisement Messages

A device discovers NMS server's address through actively initiating Neighbor Solicit message and receiving Neighbor Advertisement messages which include the new option carrying the NMS server's address.

This approach is suitable for point-to-point or non-broad circuits.

5. Security Considerations

- Device authentication for NMS Servers

With applying the mechanism described in this document, the devices would actively connect to the NMS servers. So there might be stronger desire for the NMS servers to authenticate the devices.

- ND security

This document doesn't introduce more threats than original Neighbor Discovery protocol, so generally it aligns with the security considerations described in [RFC4861].

6. IANA Considerations

The newly defined options need IANA to assign type codes.

7. Acknowledgments

Many useful comments and contributions were made by Sheng Jiang.

8. References

8.1. Normative References

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

Authors' Addresses

Bing Liu
Huawei Technologies Co., Ltd
Q14, Huawei Campus
No.156 Beiqing Rd.
Hai-Dian District, Beijing 100095
P.R. China

Email: leo.liubing@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 27, 2015

S. Previdi, Ed.
C. Filsfils
Cisco Systems, Inc.
B. Field
Comcast
I. Leung
Rogers Communications
October 24, 2014

IPv6 Segment Routing Header (SRH)
draft-previdi-6man-segment-routing-header-03

Abstract

Segment Routing (SR) allows a node to steer a packet through a controlled set of instructions, called segments, by prepending a SR header to the packet. A segment can represent any instruction, topological or service-based. SR allows to enforce a flow through any path (topological, or application/service based) while maintaining per-flow state only at the ingress node to the SR domain.

Segment Routing can be applied to the IPv6 data plane with the addition of a new type of Routing Extension Header. This draft describes the Segment Routing Extension Header Type and how it is used by SR capable nodes.

Requirements Language

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

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 April 27, 2015.

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. Structure of this document	3
2. Segment Routing Documents	3
3. Introduction	3
3.1. Data Planes supporting Segment Routing	4
3.2. Illustration	4
4. Abstract Routing Model	7
4.1. Segment Routing Global Block (SRGB)	8
4.2. Traffic Engineering with SR	9
4.3. Segment Routing Database	10
5. IPv6 Instantiation of Segment Routing	10
5.1. Segment Identifiers (SIDs) and SRGB	10
5.1.1. Node-SID	11
5.1.2. Adjacency-SID	11
5.2. Segment Routing Extension Header (SRH)	11
5.2.1. SRH and RFC2460 behavior	14
5.2.2. SRH Optimization	15
6. SRH Procedures	16
6.1. Segment Routing Operations	16
6.2. Segment Routing Node Functions	16
6.2.1. Ingress SR Node	17
6.2.2. Transit Non-SR Capable Node	18
6.2.3. SR Intra Segment Transit Node	18
6.2.4. SR Segment Endpoint Node	18
6.3. FRR Flag Settings	19
7. SR and Tunneling	19
8. Example Use Case	19
9. IANA Considerations	22
10. Manageability Considerations	22
11. Security Considerations	22

12. Contributors	22
13. Acknowledgements	22
14. References	22
14.1. Normative References	22
14.2. Informative References	22
Authors' Addresses	23

1. Structure of this document

Section 3 gives an introduction on SR for IPv6 networks.

Section 4 describes the Segment Routing abstract model.

Section 5 defines the Segment Routing Header (SRH) allowing instantiation of SR over IPv6 dataplane.

Section 6 details the procedures of the Segment Routing Header.

2. Segment Routing Documents

Segment Routing terminology is defined in [I-D.filsfils-spring-segment-routing].

Segment Routing use cases are described in [I-D.filsfils-spring-segment-routing-use-cases].

Segment Routing IPv6 use cases are described in [I-D.ietf-spring-ipv6-use-cases].

Segment Routing protocol extensions are defined in [I-D.ietf-isis-segment-routing-extensions], and [I-D.psenak-ospf-segment-routing-ospfv3-extension].

The security mechanisms of the Segment Routing Header (SRH) are described in [I-D.vyncke-6man-segment-routing-security].

3. Introduction

Segment Routing (SR), defined in [I-D.filsfils-spring-segment-routing], allows a node to steer a packet through a controlled set of instructions, called segments, by prepending a SR header to the packet. A segment can represent any instruction, topological or service-based. SR allows to enforce a flow through any path (topological or service/application based) while maintaining per-flow state only at the ingress node to the SR domain. Segments can be derived from different components: IGP, BGP, Services, Contexts, Locators, etc. The list of segment forming the path is called the Segment List and is encoded in the packet header.

SR allows the use of strict and loose source based routing paradigms without requiring any additional signaling protocols in the infrastructure hence delivering an excellent scalability property.

The source based routing model described in [I-D.filsfils-spring-segment-routing] is inherited from the ones proposed by [RFC1940] and [RFC2460]. The source based routing model offers the support for explicit routing capability.

3.1. Data Planes supporting Segment Routing

Segment Routing (SR), can be instantiated over MPLS ([I-D.filsfils-spring-segment-routing-mpls]) and IPv6. This document defines its instantiation over the IPv6 data-plane based on the use-cases defined in [I-D.ietf-spring-ipv6-use-cases].

Segment Routing for IPv6 (SR-IPv6) is required in networks where MPLS data-plane is not used or, when combined with SR-MPLS, in networks where MPLS is used in the core and IPv6 is used at the edge (home networks, datacenters).

This document defines a new type of Routing Header (originally defined in [RFC2460]) called the Segment Routing Header (SRH) in order to convey the Segment List in the packet header as defined in [I-D.filsfils-spring-segment-routing]. Mechanisms through which segment are known and advertised are outside the scope of this document.

3.2. Illustration

In the context of Figure 1 where all the links have the same IGP cost, let us assume that a packet P enters the SR domain at an ingress edge router I and that the operator requests the following requirements for packet P:

The local service S offered by node B must be applied to packet P.

The links AB and CE cannot be used to transport the packet P.

Any node N along the journey of the packet should be able to determine where the packet P entered the SR domain and where it will exit. The intermediate node should be able to determine the paths from the ingress edge router to itself, and from itself to the egress edge router.

Per-flow State for packet P should only be created at the ingress edge router.

The operator can forbid, for security reasons, anyone outside the operator domain to exploit its intra-domain SR capabilities.

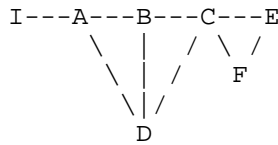


Figure 1: An illustration of SR properties

All these properties may be realized by instructing the ingress SR edge router I to push the following abstract SR header on the packet P.

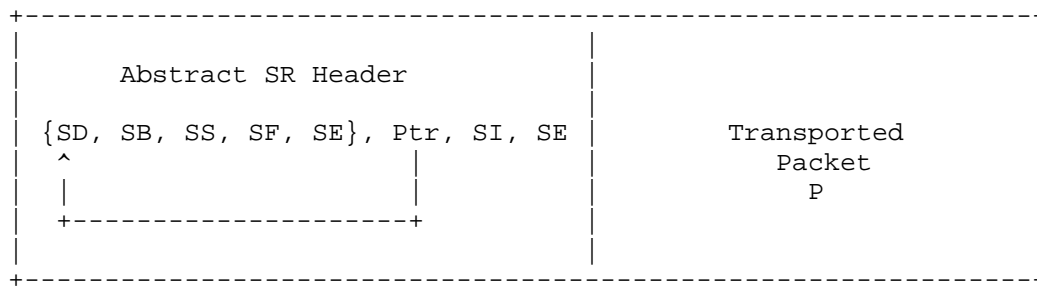


Figure 2: Packet P at node I

The abstract SR header contains a source route encoded as a list of segments {SD, SB, SS, SF, SE}, a pointer (Ptr) and the identification of the ingress and egress SR edge routers (segments SI and SE).

A segment identifies a topological instruction or a service instruction. A segment can either be global or local. The instruction associated with a global segment is recognized and executed by any SR-capable node in the domain. The instruction associated with a local segment is only supported by the specific node that originates it.

Let us assume some IGP (i.e.: ISIS and OSPF) extensions to define a "Node Segment" as a global instruction within the IGP domain to forward a packet along the shortest path to the specified node. Let us further assume that within the SR domain illustrated in Figure 1, segments SI, SD, SB, SE and SF respectively identify IGP node segments to I, D, B, E and F.

Let us assume that node B identifies its local service S with local segment SS.

With all of this in mind, let us describe the journey of the packet P.

The packet P reaches the ingress SR edge router. I pushes the SR header illustrated in Figure 2 and sets the pointer to the first segment of the list (SD).

SD is an instruction recognized by all the nodes in the SR domain which causes the packet to be forwarded along the shortest path to D.

Once at D, the pointer is incremented and the next segment is executed (SB).

SB is an instruction recognized by all the nodes in the SR domain which causes the packet to be forwarded along the shortest path to B.

Once at B, the pointer is incremented and the next segment is executed (SS).

SS is an instruction only recognized by node B which causes the packet to receive service S.

Once the service applied, the next segment is executed (SF) which causes the packet to be forwarded along the shortest path to F.

Once at F, the pointer is incremented and the next segment is executed (SE).

SE is an instruction recognized by all the nodes in the SR domain which causes the packet to be forwarded along the shortest path to E.

E then removes the SR header and the packet continues its journey outside the SR domain.

All of the requirements are met.

First, the packet P has not used links AB and CE: the shortest-path from I to D is I-A-D, the shortest-path from D to B is D-B, the shortest-path from B to F is B-C-F and the shortest-path from F to E is F-E, hence the packet path through the SR domain is I-A-D-B-C-F-E and the links AB and CE have been avoided.

Second, the service S supported by B has been applied on packet P.

Third, any node along the packet path is able to identify the service and topological journey of the packet within the SR domain. For example, node C receives the packet illustrated in Figure 3 and hence is able to infer where the packet entered the SR domain (SI), how it

got up to itself {SD, SB, SS, SE}, where it will exit the SR domain (SE) and how it will do so {SF, SE}.

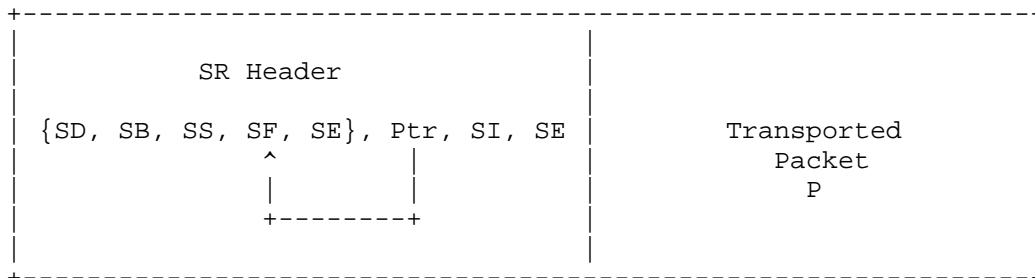


Figure 3: Packet P at node C

Fourth, only node I maintains per-flow state for packet P. The entire program of topological and service instructions to be executed by the SR domain on packet P is encoded by the ingress edge router I in the SR header in the form of a list of segments where each segment identifies a specific instruction. No further per-flow state is required along the packet path. The per-flow state is in the SR header and travels with the packet. Intermediate nodes only hold states related to the IGP global node segments and the local IGP adjacency segments. These segments are not per-flow specific and hence scale very well. Typically, an intermediate node would maintain in the order of 100's to 1000's global node segments and in the order of 10's to 100 of local adjacency segments. Typically the SR IGP forwarding table is expected to be much less than 10000 entries.

Fifth, the SR header is inserted at the entrance to the domain and removed at the exit of the operator domain. For security reasons, the operator can forbid anyone outside its domain to use its intra-domain SR capability.

4. Abstract Routing Model

At the entrance of the SR domain, the ingress SR edge router pushes the SR header on top of the packet. At the exit of the SR domain, the egress SR edge router removes the SR header.

The abstract SR header contains an ordered list of segments, a pointer identifying the next segment to process and the identifications of the ingress and egress SR edge routers on the path of this packet. The pointer identifies the segment that MUST be used by the receiving router to process the packet. This segment is called the active segment.

A property of SR is that the entire source route of the packet, including the identity of the ingress and egress edge routers is always available with the packet. This allows for interesting accounting and service applications.

We define three SR-header operations:

"PUSH": an SR header is pushed on an IP packet, or additional segments are added at the head of the segment list. The pointer is moved to the first entry of the added segments.

"NEXT": the active segment is completed, the pointer is moved to the next segment in the list.

"CONTINUE": the active segment is not completed, the pointer is left unchanged.

In the future, other SR-header management operations may be defined.

As the packet travels through the SR domain, the pointer is incremented through the ordered list of segments and the source route encoded by the SR ingress edge node is executed.

A node processes an incoming packet according to the instruction associated with the active segment.

Any instruction might be associated with a segment: for example, an intra-domain topological strict or loose forwarding instruction, a service instruction, etc.

At minimum, a segment instruction must define two elements: the identity of the next-hop to forward the packet to (this could be the same node or a context within the node) and which SR-header management operation to execute.

Each segment is known in the network through a Segment Identifier (SID). The terms "segment" and "SID" are interchangeable.

4.1. Segment Routing Global Block (SRGB)

In the SR abstract model, a segment is identified by a Segment Routing Identifier (SID). The SR abstract model doesn't mandate a specific format for the SID (IPv6 address or other formats).

In Segment Routing IPv6 the SID is an IPv6 address. Therefore, the SRGB is materialized by the global IPv6 address space which represents the set of IPv6 routable addresses in the SR domain. The following rules apply:

- o Each node of the SR domain MUST be configured with the Segment Routing Global Block (SRGB).
- o All global segments must be allocated from the SRGB. Any SR capable node MUST be able to process any global segment advertised by any other node within the SR domain.
- o Any segment outside the SRGB has a local significance and is called a "local segment". An SR-capable node MUST be able to process the local segments it originates. An SR-capable node MUST NOT support the instruction associated with a local segment originated by a remote node.

4.2. Traffic Engineering with SR

An SR Traffic Engineering policy is composed of two elements: a flow classification and a segment-list to prepend on the packets of the flow.

In SR, this per-flow state only exists at the ingress edge node where the policy is defined and the SR header is pushed.

It is outside the scope of the document to define the process that leads to the instantiation at a node N of an SR Traffic Engineering policy.

[I-D.filsfils-spring-segment-routing-use-cases] illustrates various alternatives:

N is deriving this policy automatically (e.g. FRR).

N is provisioned explicitly by the operator.

N is provisioned by a controller or server (e.g.: SDN Controller).

N is provisioned by the operator with a high-level policy which is mapped into a path thanks to a local CSPF-based computation (e.g. affinity/SRLG exclusion).

N could also be provisioned by other means.

[I-D.filsfils-spring-segment-routing-use-cases] explains why the majority of use-cases require very short segment-lists, hence minimizing the performance impact, if any, of inserting and transporting the segment list.

A SDN controller, which desires to instantiate at node N an SR Traffic Engineering policy, collects the SR capability of node N such as to ensure that the policy meets its capability.

4.3. Segment Routing Database

The Segment routing Database (SRDB) is a set of entries where each entry is identified by a SID. The instruction associated with each entry at least defines the identity of the next-hop to which the packet should be forwarded and what operation should be performed on the SR header (PUSH, CONTINUE, NEXT).

Segment	Next-Hop	SR Header operation
Sk	M	CONTINUE
Sj	N	NEXT
Sl	NAT Srvc	NEXT
Sm	FW srvc	NEXT
Sn	Q	NEXT
etc.	etc.	etc.

Figure 4: SR Database

Each SR-capable node maintains its local SRDB. SRDB entries can either derive from local policy or from protocol segment advertisement.

5. IPv6 Instantiation of Segment Routing

5.1. Segment Identifiers (SIDs) and SRGB

Segment Routing, as described in [I-D.filsfils-spring-segment-routing], defines Node-SID and Adjacency-SID. When SR is used over IPv6 data-plane the following applies.

The SRGB is the global IPv6 address space which represents the set of IPv6 routable addresses in the SR domain.

Node SIDs are IPv6 addresses part of the SRGB (i.e.: routable addresses). Adjacency-SIDs are IPv6 addresses which may not be part of the global IPv6 address space.

5.1.1.1. Node-SID

The Node-SID identifies a node. With SR-IPv6 the Node-SID is an IPv6 prefix that the operator configured on the node and that is used as the node identifier. Typically, in case of a router, this is the IPv6 address of the node loopback interface. Therefore, SR-IPv6 does not require any additional SID advertisement for the Node Segment. The Node-SID is in fact the IPv6 address of the node.

5.1.1.2. Adjacency-SID

In the SR architecture defined in [I-D.filsfils-spring-segment-routing] the Adjacency-SID (or Adj-SID) identifies a given interface and may be local or global (depending on how it is advertised). A node may advertise one (or more) Adj-SIDs allocated to a given interface so to force the forwarding of the packet (when received with that particular Adj-SID) into the interface regardless the routing entry for the packet destination. The semantic of the Adj-SID is:

Send out the packet to the interface this prefix is allocated to.

When SR is applied to IPv6, any SID is in a global IPv6 address and therefore, an Adj-SID has a global significance (i.e.: the IPv6 address representing the SID is a global address). In other words, a node that advertises the Adj-SID in the form of a global IPv6 address representing the link/adjacency the packet has to be forwarded to, will apply to the Adj-SID a global significance.

Advertisement of Adj-SID may be done using multiple mechanisms among which the ones described in ISIS and OSPF protocol extensions:

[I-D.ietf-isis-segment-routing-extensions] and [I-D.psenak-ospf-segment-routing-ospfv3-extension]. The distinction between local and global significance of the Adj-SID is given in the encoding of the Adj-SID advertisement.

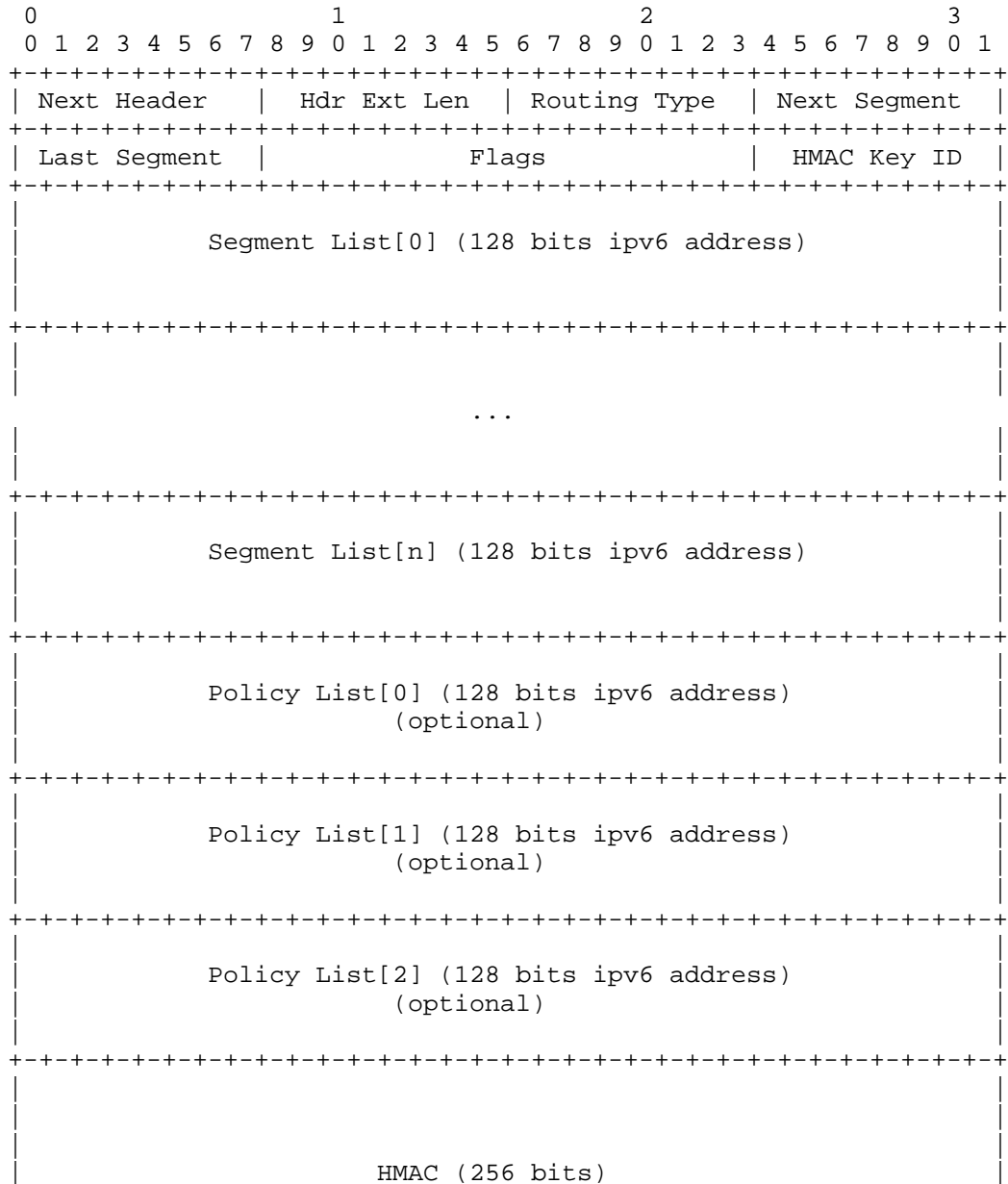
5.2. Segment Routing Extension Header (SRH)

A new type of the Routing Header (originally defined in [RFC2460]) is defined: the Segment Routing Header (SRH) which has a new Routing Type, (suggested value 4) to be assigned by IANA.

As an example, if an explicit path is to be constructed across a core network running ISIS or OSPF, the segment list will contain SIDs representing the nodes across the path (loose or strict) which, usually, are the IPv6 loopback interface address of each node. If the path is across service or application entities, the segment list

contains the IPv6 addresses of these services or application instances.

The Segment Routing Header (SRH) is defined as follows:



(optional)

where:

- o Next Header: 8-bit selector. Identifies the type of header immediately following the SRH.
- o Hdr Ext Len: 8-bit unsigned integer, is the length of the SRH header in 8-octet units, not including the first 8 octets.
- o Routing Type: TBD, to be assigned by IANA (suggested value: 4).
- o Next Segment (originally defined as "Segments Left" in [RFC2460]): index, in the Segment List, of the next active segment (according to terminology defined in [I-D.filsfils-spring-segment-routing]) in the SRH. Note that this differs from the semantic defined in the Routing Header specification ([RFC2460] defines it as "Segments Left"). Therefore, in the Segment Routing context, the "Segments Left" field is renamed as "Next Segment".
- o Last Segment: index, in the Segment List, of the next active segment of the last segment of the path in the SRH.
- o Flags: 16 bits of flags. Following flags are defined:

[illegible]

C-flag: Clean-up flag. Set when the SRH has to be removed from the packet when packet reaches the last segment.

P-flag: Protected flag. Set when the packet has been rerouted through FRR mechanism by a SR endpoint node. See Section 6.3 for more details.

R-flags. Reserved and for future use.

Policy List flags. Define the type of the IPv6 addresses encoded into the Policy List (see below). The following have been defined:

Bits 4-6: determine the type of the first element after the segment list.

Bits 7-9: determine the type of the second element.

Bits 10-12: determine the type of the third element.

Bits 13-15: determine the type of the fourth element.

The following values are used for the type:

0x0: Not present. If value is set to 0x0, it means the element represented by these bits is not present.

0x1: Ingress SR PE address.

0x2: Egress SR PE address.

0x3: Original Source Address.

- o HMAC Key ID and HMAC field, and their use are defined in [I-D.vyncke-6man-segment-routing-security].
- o Segment List[n]: 128 bit IPv6 addresses representing the nth segment of the path.
- o Policy List. Optional addresses representing specific nodes in the SR path such as:

Ingress SR PE: IPv6 address representing the SR node which has imposed the SRH (SR domain ingress).

Egress SR PE: IPv6 address representing the egress SR domain node.

Original Source Address: IPv6 address originally present in the SA field of the packet.

The segments in the Policy List are encoded after the segment list and they are optional. If none are in the SRH, all bits of the Policy List Flags MUST be set to 0x0.

5.2.1. SRH and RFC2460 behavior

The SRH being a new type of the Routing Header, it also has the same properties:

SHOULD only appear once in the packet.

Only the router whose address is in the DA field of the packet header MUST inspect the SRH.

Therefore, Segment Routing in IPv6 networks implies that the segment identifier (i.e.: the IPv6 address of the segment) is moved into the DA of the packet.

The DA of the packet changes at each segment termination/completion and therefore the original DA of the packet MUST be encoded as the last segment of the path.

As illustrated in Section 3.2, nodes that are within the path of a segment will forward packets based on the DA of the packet without inspecting the SRH. This ensures full interoperability between SR-capable and non-SR-capable nodes.

5.2.2. SRH Optimization

In order to optimize the way the SRH and, more precisely, the Segment List is processed by SR nodes, it is desirable that most of the necessary information of the SL is placed at the top of the list so to avoid reading the whole content of the SRH prior to make forwarding decisions.

With this in mind, when the SRH is created and the segment list is inserted, the order of the segments in the segment list is as follows:

- o The Next Segment field points to the next segment to be examined (offset within the SRH).
- o The first segment being encoded in the DA by the ingress node, it doesn't need to sit in the first position of the list.
- o Hence, the first element of the segment list is the second segment of the path so that, when the packet reaches the end of the first segment, the node inspecting the SRH will find the second segment at the beginning of the segment list.
- o The other segments of the path are encoded sequentially after the second segment.
- o The last segment of the path is the original DA address.
- o The last segment in the Segment List is used to encode the first segment. This segment will never be inspected anyway (at least not for forwarding purposes).

6. SRH Procedures

In this section we describe the different procedures on the SRH.

6.1. Segment Routing Operations

When Segment Routing is instantiated over the IPv6 data plane the following applies:

- o The segment list is encoded in the SRH.
- o The active segment is in the destination address of the packet.
- o The Segment Routing CONTINUE operation (as described in [I-D.filsfils-spring-segment-routing]) is implemented as a regular/plain IPv6 operation consisting of DA based forwarding.
- o The NEXT operation is implemented through the update of the DA with the value represented by the Next Segment field in the SRH.
- o The PUSH operation is implemented through the insertion of the SRH or the insertion of additional segments in the SRH segment list.

6.2. Segment Routing Node Functions

SR packets are forwarded to segments endpoints (i.e.: nodes whose address is in the DA field of the packet). The segment endpoint, when receiving a SR packet destined to itself, does:

- o Inspect the SRH.
- o Determine the next segment.
- o Update the SRH (or, if requested, remove the SRH from the packet).
- o Update the DA.
- o Send the packet to the next segment.

The procedures applied to the SRH are related to the node function. Following nodes functions are defined:

Ingress SR Node.

Transit Non-SR Node.

Transit SR Intra Segment Node.

SR Endpoint Node.

6.2.1. Ingress SR Node

Ingress Node can be a router at the edge of the SR domain or a SR-capable host. The ingress SR node may obtain the segment list by either:

- Local path computation.

- Local configuration.

- Interaction with an SDN controller delivering the path as a complete SRH.

- Any other mechanism (mechanisms through which the path is acquired are outside the scope of this document).

When creating the SRH (either at ingress node or in the SDN controller) the following is done:

- Next Header and Hdr Ext Len fields are set according to [RFC2460].

- Routing Type field is set as TBD (SRH).

- The DA of the packet is set with the address of the FIRST segment of the path.

- Next Segment field contains the offset of the SECOND segment of the path which is encoded in the FIRST position of the segment list. The segment list is encoded as follows:

 - The first element of the list contains the second segment (as stated above).

 - All subsequent segments are encoded following the second segment.

 - The original DA of the packet is encoded as the last segment of the path (which is NOT the last segment of the segment list).

 - The last segment of the segment list is the FIRST segment of the path.

- Last Segment field contains the offset of the last segment of the path (i.e.: the original DA of the packet).

- The packet is sent out to the first segment.

6.2.1.1. Security at Ingress

The procedures related to the Segment Routing security are detailed in [I-D.vyncke-6man-segment-routing-security].

In the case where the SR domain boundaries are not under control of the network operator (e.g.: when the SR domain edge is in a home network), it is important to authenticate and validate the content of any SRH being received by the network operator. In such case, the security procedure described in [I-D.vyncke-6man-segment-routing-security] is to be used.

The ingress node (e.g.: the host in the home network) requests the SRH from a control system (e.g.: an SDN controller) which delivers the SRH with its HMAC signature on it.

Then, the home network host can send out SR packets (with an SRH on it) that will be validated at the ingress of the network operator infrastructure.

The ingress node of the network operator infrastructure, is configured in order to validate the incoming SRH HMACs in order to allow only packets having correct SRH according to their SA/DA addresses.

6.2.2. Transit Non-SR Capable Node

SR is interoperable with plain IPv6 forwarding. Any non SR-capable node will forward SR packets solely based on the DA. There's no SRH inspection. This ensures full interoperability between SR and non-SR nodes.

6.2.3. SR Intra Segment Transit Node

Only the node whose address is in DA inspects and processes the SRH (according to [RFC2460]). An intra segment transit node is not in the DA and its forwarding is based on DA and its SR-IPv6 FIB.

6.2.4. SR Segment Endpoint Node

The SR segment endpoint node is the node whose address is in the DA. The segment endpoint node inspects the SRH and does:

1. IF DA = myself (segment endpoint)
2. IF Next Segment <> Last Segment THEN
 update DA with Next Segment
 increment Next Segment
3. ELSE IF Last Segment <> DA THEN
 update DA with Next Segment
 IF Clean-up bit is set THEN remove the SRH
4. ELSE give the packet to next PID (application)
 End of processing.
5. Forward the packet out

6.3. FRR Flag Settings

A node supporting SR and doing Fast Reroute (as described in [I-D.filsfils-spring-segment-routing-use-cases], when rerouting packets through FRR mechanisms, SHOULD inspect the rerouted packet header and look for the SRH. If the SRH is present, the rerouting node SHOULD set the Protected bit on all rerouted packets.

7. SR and Tunneling

Encapsulation can be realized in two different ways with SR-IPv6:

Outer encapsulation.

SRH with SA/DA original addresses.

Outer encapsulation tunneling is the traditional method where an additional IPv6 header is prepended to the packet. The original IPv6 header being encapsulated, everything is preserved and the packet is switched/routed according to the outer header (that could contain a SRH).

SRH allows encoding both original SA and DA and therefore, hence an operator may decide to change the SA/DA at ingress and restore them at egress. This can be achieved without outer encapsulation, by changing SA/DA and encoding the original values in the Segment List (the last segment of the path being the original DA) and in the Policy List (original SA).

8. Example Use Case

A more detailed description of use cases are available in [I-D.ietf-spring-ipv6-use-cases]. In this section, a simple SR-IPv6 example is illustrated.

In the topology described in Figure 6 it is assumed an end-to-end SR deployment. Therefore SR is supported by all nodes from A to J.

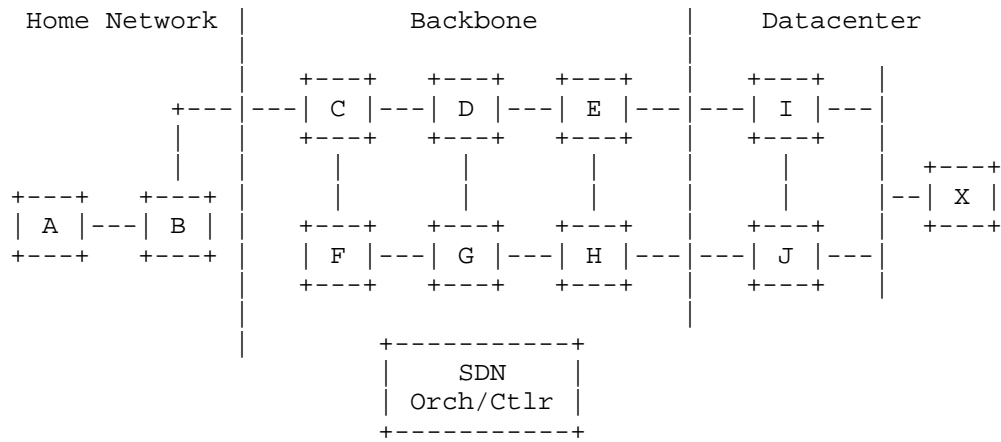


Figure 6: Sample SR topology

The following workflow applies to packets sent by host A and destined to server X.

- . Host A sends a request for a path to server X to the SDN controller or orchestration system.
- . The SDN controller/orchestrator builds a SRH with:
 - . Segment List: C, F, J, X
 - . HMACthat satisfies the requirements expressed in the request by host A and based on policies applicable to host A.
- . Host A receives the SRH and insert it into the packet. The packet has now:
 - . SA: A
 - . DA: C
 - . SRH with
 - . SL: F,J,X,C
 - . PL: C (ingress), J (egress)Note that X is the last segment and C is the first segment (encoded at the end of the SL).
- . When packet arrives in C (first segment), C does:
 - . Validate the HMAC of the SRH.
 - . Update the DA with the next segment (found in SRH):
 - DA is set to F.
 - . Forward the packet to F.
- . Packet arrives in F which inspects the SRH and find the next segment:
 - . DA is set to J.
- . Packet travels across G and H nodes which do plain IPv6 forwarding based on DA. No inspection of SRH needs to be done in these nodes. However, any SR capable node is allowed to set the Protected bit in case of FRR protection.
- . Packet arrives in J where two options are available depending on the settings of the cleanup bit set in the SRH:
 - . If the cleanup bit is set, then node J will strip out the SRH from the packet, set the DA as X and send the packet out.
 - . If the clean-up bit is not set, the DA is set to X and the packet is sent out with the SRH.

The packet arrives in the server that may or may not support SR. The return traffic, from server to host, may be sent using the same procedures.

9. IANA Considerations

TBD

10. Manageability Considerations

TBD

11. Security Considerations

Security mechanisms applied to Segment Routing over IPv6 networks are detailed in [I-D.vyncke-6man-segment-routing-security].

12. Contributors

The authors would like to thank Dave Barach, John Leddy, John Brzozowski, Pierre Francois, Nagendra Kumar, Mark Townsley, Christian Martin, Roberta Maglione, James Connolly and David Lebrun for their contribution to this document.

13. Acknowledgements

TBD

14. References

14.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

14.2. Informative References

- [I-D.filsfils-spring-segment-routing]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-filsfils-spring-segment-routing-04 (work in progress), July 2014.

- [I-D.filsfils-spring-segment-routing-mpls]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., and E. Crabbe, "Segment Routing with MPLS data plane", draft-filsfils-spring-segment-routing-mpls-03 (work in progress), August 2014.
- [I-D.filsfils-spring-segment-routing-use-cases]
Filsfils, C., Francois, P., Previdi, S., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., Kini, S., and E. Crabbe, "Segment Routing Use Cases", draft-filsfils-spring-segment-routing-use-cases-01 (work in progress), October 2014.
- [I-D.ietf-isis-segment-routing-extensions]
Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-02 (work in progress), June 2014.
- [I-D.ietf-spring-ipv6-use-cases]
Brzozowski, J., Leddy, J., Leung, I., Previdi, S., Townsley, W., Martin, C., Filsfils, C., and R. Maglione, "IPv6 SPRING Use Cases", draft-ietf-spring-ipv6-use-cases-01 (work in progress), July 2014.
- [I-D.psenak-ospf-segment-routing-ospfv3-extension]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPFv3 Extensions for Segment Routing", draft-psenak-ospf-segment-routing-ospfv3-extension-02 (work in progress), July 2014.
- [I-D.vyncke-6man-segment-routing-security]
Vyncke, E. and S. Previdi, "IPv6 Segment Routing Header (SRH) Security Considerations", July 2014.
- [RFC1940] Estrin, D., Li, T., Rekhter, Y., Varadhan, K., and D. Zappala, "Source Demand Routing: Packet Format and Forwarding Specification (Version 1)", RFC 1940, May 1996.

Authors' Addresses

Stefano Previdi (editor)
Cisco Systems, Inc.
Via Del Serafico, 200
Rome 00142
Italy

Email: sprevidi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Brussels
BE

Email: cfilsfil@cisco.com

Brian Field
Comcast
4100 East Dry Creek Road
Centennial, CO 80122
US

Email: Brian_Field@cable.comcast.com

Ida Leung
Rogers Communications
8200 Dixie Road
Brampton, ON L6T 0C1
CA

Email: Ida.Leung@rci.rogers.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 4, 2016

S. Previdi, Ed.
C. Filsfils
Cisco Systems, Inc.
B. Field
Comcast
I. Leung
Rogers Communications
J. Linkova
Google
E. Aries
Facebook
T. Kosugi
NTT
E. Vyncke
Cisco Systems, Inc.
D. Lebrun
Universite Catholique de Louvain
October 2, 2015

IPv6 Segment Routing Header (SRH)
draft-previdi-6man-segment-routing-header-08

Abstract

Segment Routing (SR) allows a node to steer a packet through a controlled set of instructions, called segments, by prepending a SR header to the packet. A segment can represent any instruction, topological or service-based. SR allows to enforce a flow through any path (topological, or application/service based) while maintaining per-flow state only at the ingress node to the SR domain.

Segment Routing can be applied to the IPv6 data plane with the addition of a new type of Routing Extension Header. This draft describes the Segment Routing Extension Header Type and how it is used by SR capable nodes.

Requirements Language

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

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 April 4, 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. Segment Routing Documents	3
2. Introduction	3
2.1. Data Planes supporting Segment Routing	4
2.2. Segment Routing (SR) Domain	4
2.2.1. SR Domain in a Service Provider Network	5
2.2.2. SR Domain in a Overlay Network	6
2.3. Illustration	8
3. IPv6 Instantiation of Segment Routing	10
3.1. Segment Identifiers (SIDs)	10
3.1.1. Node-SID	10
3.1.2. Adjacency-SID	11
3.2. Segment Routing Extension Header (SRH)	11
3.2.1. SRH and RFC2460 behavior	14
4. SRH Procedures	15
4.1. Segment Routing Node Functions	15
4.1.1. Source SR Node	16
4.1.2. SR Domain Ingress Node	17
4.1.3. Transit Node	17
4.1.4. SR Segment Endpoint Node	17

5.	Security Considerations	18
5.1.	Threat model	19
5.1.1.	Source routing threats	19
5.1.2.	Applicability of RFC 5095 to SRH	19
5.1.3.	Service stealing threat	20
5.1.4.	Topology disclosure	20
5.1.5.	ICMP Generation	20
5.2.	Security fields in SRH	21
5.2.1.	Selecting a hash algorithm	22
5.2.2.	Performance impact of HMAC	22
5.2.3.	Pre-shared key management	23
5.3.	Deployment Models	23
5.3.1.	Nodes within the SR domain	23
5.3.2.	Nodes outside of the SR domain	24
5.3.3.	SR path exposure	24
5.3.4.	Impact of BCP-38	25
6.	IANA Considerations	25
7.	Manageability Considerations	25
8.	Contributors	25
9.	Acknowledgements	26
10.	References	26
10.1.	Normative References	26
10.2.	Informative References	26
	Authors' Addresses	28

1. Segment Routing Documents

Segment Routing terminology is defined in
[I-D.ietf-spring-segment-routing].

Segment Routing use cases are described in
[I-D.ietf-spring-problem-statement] and
[I-D.ietf-spring-ipv6-use-cases].

Segment Routing protocol extensions are defined in
[I-D.ietf-isis-segment-routing-extensions], and
[I-D.ietf-ospf-ospfv3-segment-routing-extensions].

2. Introduction

Segment Routing (SR), defined in [I-D.ietf-spring-segment-routing], allows a node to steer a packet through a controlled set of instructions, called segments, by prepending a SR header to the packet. A segment can represent any instruction, topological or service-based. SR allows to enforce a flow through any path (topological or service/application based) while maintaining per-flow state only at the ingress node to the SR domain. Segments can be derived from different components: IGP, BGP, Services, Contexts,

Locators, etc. The list of segment forming the path is called the Segment List and is encoded in the packet header.

SR allows the use of strict and loose source based routing paradigms without requiring any additional signaling protocols in the infrastructure hence delivering an excellent scalability property.

The source based routing model described in [I-D.ietf-spring-segment-routing] is inherited from the ones proposed by [RFC1940] and [RFC2460]. The source based routing model offers the support for explicit routing capability.

2.1. Data Planes supporting Segment Routing

Segment Routing (SR), can be instantiated over MPLS ([I-D.ietf-spring-segment-routing-mpls]) and IPv6. This document defines its instantiation over the IPv6 data-plane based on the use-cases defined in [I-D.ietf-spring-ipv6-use-cases].

This document defines a new type of Routing Header (originally defined in [RFC2460]) called the Segment Routing Header (SRH) in order to convey the Segment List in the packet header as defined in [I-D.ietf-spring-segment-routing]. Mechanisms through which segment are known and advertised are outside the scope of this document.

A segment is materialized by an IPv6 address. A segment identifies a topological instruction or a service instruction. A segment can be either:

- o global: a global segment represents an instruction supported by all nodes in the SR domain and it is instantiated through an IPv6 address globally known in the SR domain.
- o local: a local segment represents an instruction supported only by the node who originates it and it is instantiated through an IPv6 address that is known only by the local node.

2.2. Segment Routing (SR) Domain

We define the concept of the Segment Routing Domain (SR Domain) as the set of nodes participating into the source based routing model. These nodes may be connected to the same physical infrastructure (e.g.: a Service Provider's network) as well as nodes remotely connected to each other (e.g.: an enterprise VPN or an overlay).

A non-exhaustive list of examples of SR Domains is:

- o The network of an operator, service provider, content provider, enterprise including nodes, links and Autonomous Systems.
- o A set of nodes connected as an overlay over one or more transit providers. The overlay nodes exchange SR-enabled traffic with segments belonging solely to the overlay routers (the SR domain). None of the segments in the SR-enabled packets exchanged by the overlay belong to the transit networks

The source based routing model through its instantiation of the Segment Routing Header (SRH) defined in this document equally applies to all the above examples.

While the source routing model defined in [RFC2460] doesn't mandate which node is allowed to insert (or modify) the SRH, it is assumed in this document that the SRH is inserted in the packet by its source. For example:

- o At the node originating the packet (host, server).
- o At the ingress node of a SR domain where the ingress node receives an IPv6 packet and encapsulates it into an outer IPv6 header followed by a Segment Routing header.

2.2.1. SR Domain in a Service Provider Network

The following figure illustrates an SR domain consisting of an operator's network infrastructure.

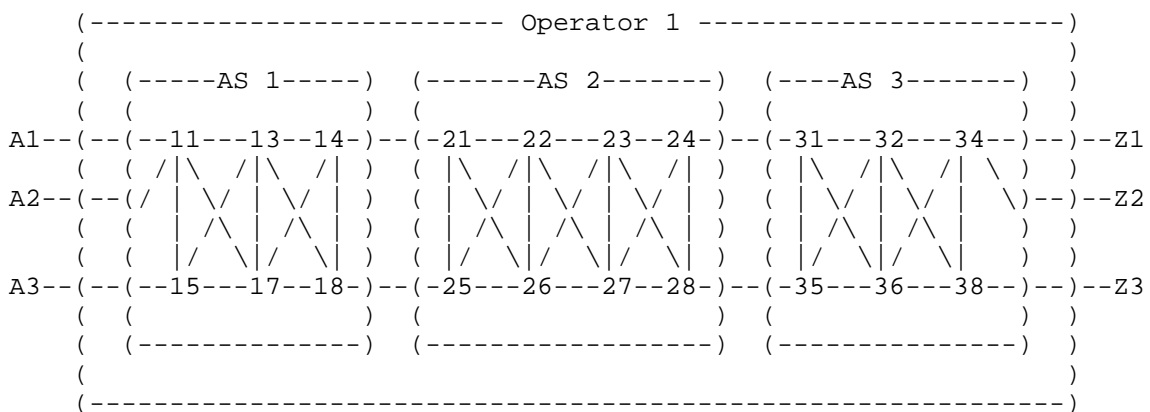


Figure 1: Service Provider SR Domain

Figure 1 describes an operator network including several ASes and delivering connectivity between endpoints. In this scenario, Segment

Routing is used within the operator networks and across the ASes boundaries (all being under the control of the same operator). In this case segment routing can be used in order to address use cases such as end-to-end traffic engineering, fast re-route, egress peer engineering, data-center traffic engineering as described in [I-D.ietf-spring-problem-statement], [I-D.ietf-spring-ipv6-use-cases] and [I-D.ietf-spring-resiliency-use-cases].

Typically, an IPv6 packet received at ingress (i.e.: from outside the SR domain), is classified according to network operator policies and such classification results into an outer header with an SRH applied to the incoming packet. The SRH contains the list of segment representing the path the packet must take inside the SR domain. Thus, the SA of the packet is the ingress node, the DA (due to SRH procedures described in Section 4) is set as the first segment of the path and the last segment of the path is the egress node of the SR domain.

The path may include intra-AS as well as inter-AS segments. It has to be noted that all nodes within the SR domain are under control of the same administration. When the packet reaches the egress point of the SR domain, the outer header and its SRH are removed so that the destination of the packet is unaware of the SR domain the packet has traversed.

The outer header with the SRH is no different from any other tunneling encapsulation mechanism and allows a network operator to implement traffic engineering mechanisms so to efficiently steer traffic across his infrastructure.

2.2.2. SR Domain in a Overlay Network

The following figure illustrates an SR domain consisting of an overlay network over multiple operator's networks.

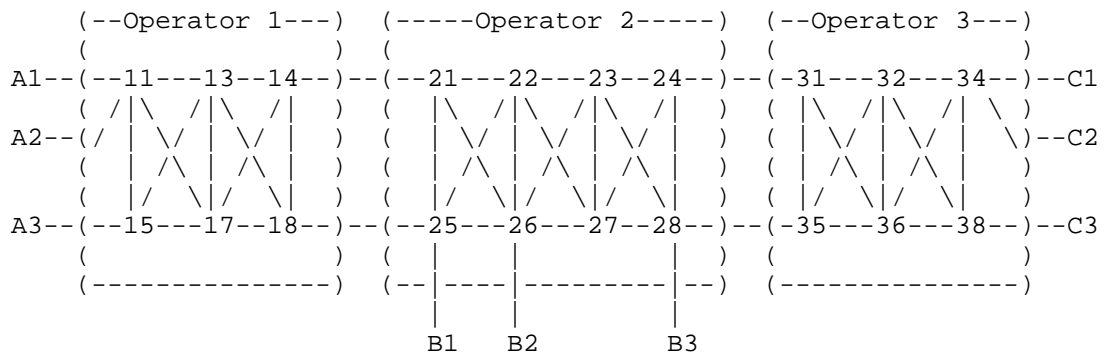


Figure 2: Overlay SR Domain

Figure 2 describes an overlay consisting of nodes connected to three different network operators and forming a single overlay network where Segment routing packets are exchanged.

The overlay consists of nodes A1, A2, A3, B1, B2, B3, C1, C2 and C3. These nodes are connected to their respective network operator and form an overlay network.

Each node may originate packets with an SRH which contains, in the segment list of the SRH or in the DA, segments identifying other overlay nodes. This implies that packets with an SRH may traverse operator's networks but, obviously, these SRHs cannot contain an address/segment of the transit operators 1, 2 and 3. The SRH originated by the overlay can only contain address/segment under the administration of the overlay (e.g. address/segments supported by A1, A2, A3, B1, B2, B3, C1, C2 or C3).

In this model, the operator network nodes are transit nodes and, according to [RFC2460], MUST NOT inspect the routing extension header since there are not the DA of the packet.

It is a common practice in operators networks to filter out, at ingress, any packet whose DA is the address of an internal node and it is also possible that an operator would filter out any packet destined to an internal address and having an extension header in it.

This common practice does not impact the SR-enabled traffic between the overlay nodes as the intermediate transit networks do never see a destination address belonging to their infrastructure. These SR-enabled overlay packets will thus never be filtered by the transit operators.

In all cases, transit packets (i.e.: packets whose DA is outside the domain of the operator's network) will be forwarded accordingly without introducing any security concern in the operator's network. This is similar to tunneled packets.

2.3. Illustration

In the context of Figure 3 we illustrate an example of how segment routing can be used within a SR domain in order to engineer traffic. Let's assume that the SR domain is configured as a single AS and the IGP (OSPF or IS-IS) is configured using the same cost on every link. Let's also assume that a packet P enters the SR domain at an ingress edge router I and that the operator requests the following requirements for packet P:

- o The local service S offered by node B must be applied to packet P.
- o The links AB and CE cannot be used to transport the packet P.
- o Any node N along the journey of the packet should be able to determine where the packet P entered the SR domain and where it will exit. The intermediate node should be able to determine the paths from the ingress edge router to itself, and from itself to the egress edge router.
- o Per-flow State for packet P should only be created at the ingress edge router.
- o The operator can forbid, for security reasons, anyone outside the operator domain to exploit its intra-domain SR capabilities.

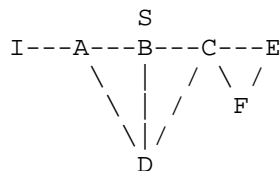


Figure 3: An illustration of SR properties

All these properties may be realized by instructing the ingress SR edge router I to create a SRH with the list of segments the packet must traverse: D, B, S, F, E. Therefore, the ingress router I creates an outer header where:

- o the SA is the IPv6 address of I

- o the final destination of the packet is the SR egress node E however, D being the first segment of the path, the DA is set to D IPv6 address.
- o the SRH is inserted with the segment list consisting of following IPv6 addresses: D, B, S, F, E

The SRH contains a source route encoded as a list of segments (D, B, S, F, E). The ingress and egress nodes are identified in the packet respectively by the SA and the last segment of the segment list.

The packet P reaches the ingress SR node I. Node I pushes the newly created outer header and SRH with the Segment List as illustrated above (D, B, S, F, E)

D is the IPv6 address of node D and it is recognized by all nodes in the SR domain as the forwarding instruction "forward to D according to D route in the IPv6 routing table". The routing table being built through IGPs (OSPF or IS-IS) it is equivalent to say "forward according to shortest path to D".

Once at D, the next segment is inspected and executed (segment B).

B is an instruction recognized by all the nodes in the SR domain which causes the packet to be forwarded along the shortest path to B.

Once at B, the next segment is executed (segment S).

S is an instruction only recognized by node B which causes the packet to receive service S.

Once the service S is applied, the next segment is executed (segment F) which causes the packet to be forwarded along the shortest path to F.

Once at F, the next segment is executed (segment E).

E is an instruction recognized by all the nodes in the SR domain which causes the packet to be forwarded along the shortest path to E.

E being the destination of the packet, removes the outer header and the SRH. Then, it inspects the inner packet header and forwards the packet accordingly.

All of the requirements are met:

- o First, the packet P has not used links AB and CE: the shortest-path from I to D is I-A-D, the shortest-path from D to B is D-B,

the shortest-path from B to F is B-C-F and the shortest-path from F to E is F-E, hence the packet path through the SR domain is I-A-D-B-C-F-E and the links AB and CE have been avoided.

- o Second, the service S supported by B has been applied on packet P.
- o Third, any node along the packet path is able to identify the service and topological journey of the packet within the SR domain by inspecting the SRH and SA/DA fields of the packet header.
- o Fourth, only node I maintains per-flow state for packet P. The entire program of topological and service instructions to be executed by the SR domain on packet P is encoded by the ingress edge router I in the SR header in the form of a list of segments where each segment identifies a specific instruction. No further per-flow state is required along the packet path. Intermediate nodes only hold states related to the global node segments and their local segments. These segments are not per-flow specific and hence scale very well. Typically, an intermediate node would maintain in the order of 100's to 1000's global node segments and in the order of 10's to 100 of local segments.
- o Fifth, the SR header (and its outer header) is inserted at the entrance to the domain and removed at the exit of the operator domain. For security reasons, the operator can forbid anyone outside its domain to use its intra-domain SR capability (e.g. configuring ACL that deny any packet with a DA towards its infrastructure segment).

3. IPv6 Instantiation of Segment Routing

3.1. Segment Identifiers (SIDs)

Segment Routing, as described in [I-D.ietf-spring-segment-routing], defines Node-SID and Adjacency-SID. When SR is used over IPv6 data-plane the following applies.

3.1.1. Node-SID

The Node-SID identifies a node. With SR-IPv6 the Node-SID is an IPv6 address that the operator configured on the node and that is used as the node identifier. Typically, in case of a router, this is the IPv6 address of the node loopback interface. Therefore, SR-IPv6 does not require any additional SID advertisement for the Node Segment. The Node-SID is in fact the IPv6 address of the node.

3.1.2. Adjacency-SID

Adjacency-SIDs can be either globally scoped IPv6 addresses or IPv6 addresses known locally by the node but not advertised in any control plane (in other words an Adjacency-SID may well be any 128-bit identifier). Obviously, in the latter case, the scope of the Adjacency-SID is local to the router and any packet with the a such Adjacency-SID would need first to reach the node through the node's Segment Identifier (i.e.: Node-SID) prior for the node to process the Adjacency-SID. In other words, two segments (SIDs) would then be required: the first is the node's Node-SID that brings the packet to the node and the second is the Adjacency-SID that will make the node to forward the packet through the interface the Adjacency-SID is allocated to.

In the SR architecture defined in [I-D.ietf-spring-segment-routing] a node may advertise one (or more) Adj-SIDs allocated to the same interface as well as a node can advertise the same Adj-SID for multiple interfaces. Use cases of Adj-SID advertisements are described in [I-D.ietf-spring-segment-routing] The semantic of the Adj-SID is:

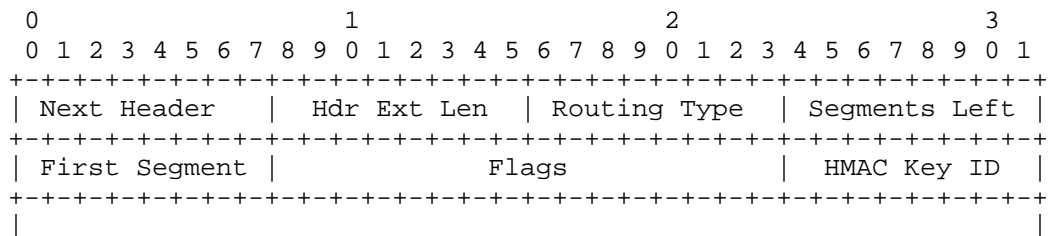
Send out the packet to the interface this Adj-SID is allocated to.

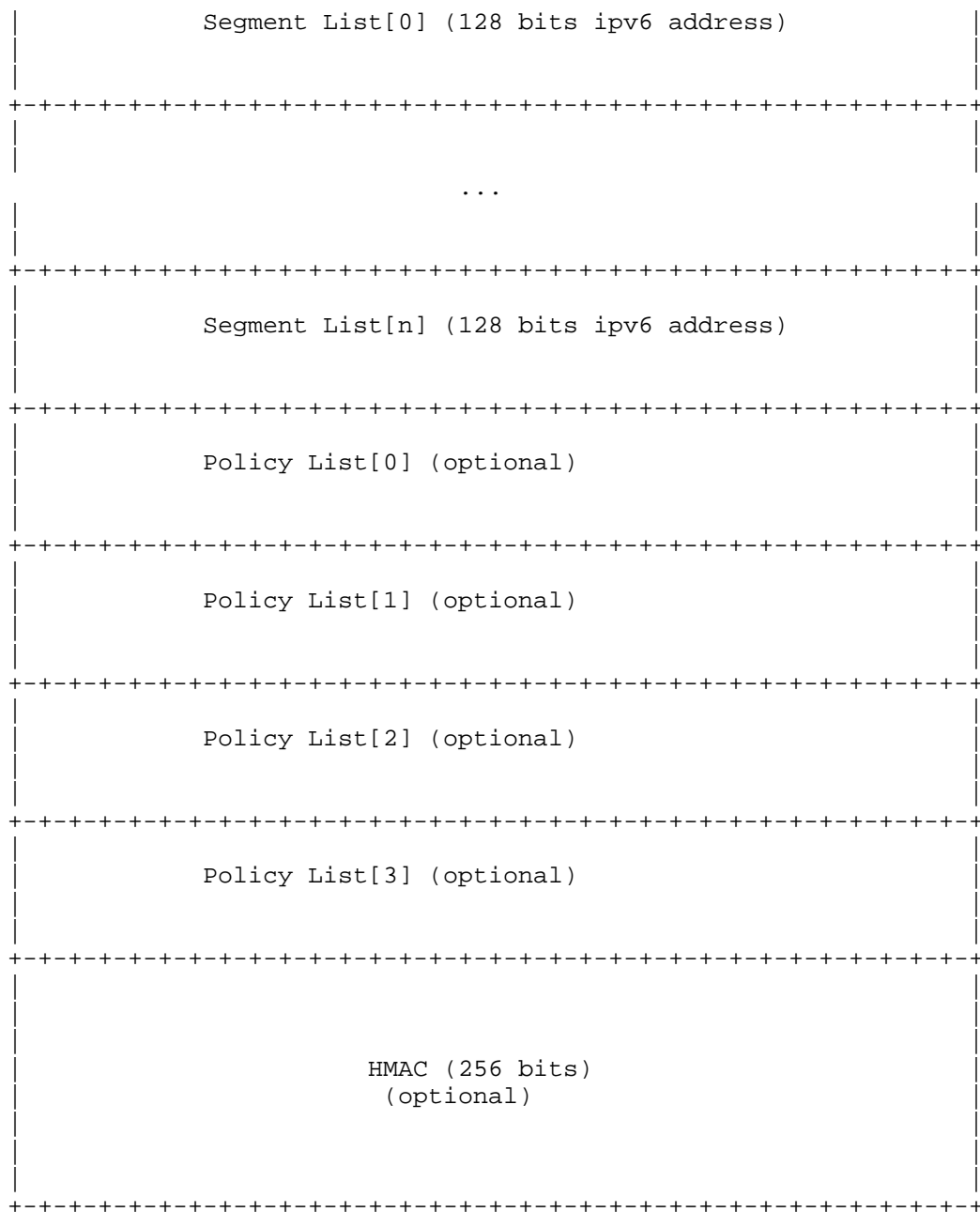
Advertisement of Adj-SID may be done using multiple mechanisms among which the ones described in ISIS and OSPF protocol extensions: [I-D.ietf-isis-segment-routing-extensions] and [I-D.ietf-ospf-ospfv3-segment-routing-extensions]. The distinction between local and global significance of the Adj-SID is given in the encoding of the Adj-SID advertisement.

3.2. Segment Routing Extension Header (SRH)

A new type of the Routing Header (originally defined in [RFC2460]) is defined: the Segment Routing Header (SRH) which has a new Routing Type, (suggested value 4) to be assigned by IANA.

The Segment Routing Header (SRH) is defined as follows:





where:

- o Next Header: 8-bit selector. Identifies the type of header immediately following the SRH.
- o Hdr Ext Len: 8-bit unsigned integer, is the length of the SRH header in 8-octet units, not including the first 8 octets.
- o Routing Type: TBD, to be assigned by IANA (suggested value: 4).
- o Segments Left. Defined in [RFC2460], it contains the index, in the Segment List, of the next segment to inspect. Segments Left is decremented at each segment.
- o First Segment: contains the index, in the Segment List, of the first segment of the path which is in fact the last element of the Segment List.
- o Flags: 16 bits of flags. Following flags are defined:

```

                                1
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+---+---+---+---+---+---+---+---+
|C|P|R|R|   Policy Flags   |
+---+---+---+---+---+---+---+---+

```

C-flag: Clean-up flag. Set when the SRH has to be removed from the packet when packet reaches the last segment.

P-flag: Protected flag. Set when the packet has been rerouted through FRR mechanism by a SR endpoint node.

R-flags. Reserved and for future use.

Policy Flags. Define the type of the IPv6 addresses encoded into the Policy List (see below). The following have been defined:

Bits 4-6: determine the type of the first element after the segment list.

Bits 7-9: determine the type of the second element.

Bits 10-12: determine the type of the third element.

Bits 13-15: determine the type of the fourth element.

The following values are used for the type:

0x0: Not present. If value is set to 0x0, it means the element represented by these bits is not present.

0x1: SR Ingress.

0x2: SR Egress.

0x3: Original Source Address.

0x4 to 0x7: currently unused and SHOULD be ignored on reception.

- o HMAC Key ID and HMAC field, and their use are defined in Section 5.
- o Segment List[n]: 128 bit IPv6 addresses representing the nth segment in the Segment List. The Segment List is encoded starting from the last segment of the path. I.e., the first element of the segment list (Segment List [0]) contains the last segment of the path while the last segment of the Segment List (Segment List[n]) contains the first segment of the path. The index contained in "Segments Left" identifies the current active segment.
- o Policy List. Optional addresses representing specific nodes in the SR path such as:

SR Ingress: a 128 bit generic identifier representing the ingress in the SR domain (i.e.: it needs not to be a valid IPv6 address).

SR Egress: a 128 bit generic identifier representing the egress in the SR domain (i.e.: it needs not to be a valid IPv6 address).

Original Source Address: IPv6 address originally present in the SA field of the packet.

The segments in the Policy List are encoded after the segment list and they are optional. If none are in the SRH, all bits of the Policy List Flags MUST be set to 0x0.

3.2.1. SRH and RFC2460 behavior

The SRH being a new type of the Routing Header, it also has the same properties:

SHOULD only appear once in the packet.

Only the router whose address is in the DA field of the packet header MUST inspect the SRH.

Therefore, Segment Routing in IPv6 networks implies that the segment identifier (i.e.: the IPv6 address of the segment) is moved into the DA of the packet.

The DA of the packet changes at each segment termination/completion and therefore the original DA of the packet MUST be encoded as the last segment of the path.

As illustrated in Section 2.3, nodes that are within the path of a segment will forward packets based on the DA of the packet without inspecting the SRH. This ensures full interoperability between SR-capable and non-SR-capable nodes.

4. SRH Procedures

In this section we describe the different procedures on the SRH.

4.1. Segment Routing Node Functions

SR packets are forwarded to segments endpoints (i.e.: the segment endpoint is the node representing the segment and whose address is in the segment list and in the DA of the packet when traveling in the segment). The segment endpoint, when receiving a SR packet destined to itself, does:

- o Inspect the SRH.
- o Determine the next active segment.
- o Update the Segments Left field (or, if requested, remove the SRH from the packet).
- o Update the DA.
- o Forward the packet to the next segment.

The procedures applied to the SRH are related to the node function. Following nodes functions are defined:

Source SR Node.

SR Domain Ingress Node.

Transit Node.

SR Endpoint Node.

4.1.1.1. Source SR Node

A Source SR Node can be any node originating an IPv6 packet with its IPv6 and Segment Routing Headers. This include either:

A host originating an IPv6 packet

A SR domain ingress router encapsulating a received IPv6 packet into an outer IPv6 header followed by a SRH

The mechanism through which a Segment List is derived is outside of the scope of this document. As an example, the Segment List may be obtained through:

Local path computation.

Local configuration.

Interaction with a centralized controller delivering the path.

Any other mechanism.

The following are the steps of the creation of the SRH:

Next Header and Hdr Ext Len fields are set according to [RFC2460].

Routing Type field is set as TBD (SRH).

The Segment List is built with the FIRST segment of the path encoded in the LAST element of the Segment List. Subsequent segments are encoded on top of the first segment. Finally, the LAST segment of the path is encoded in the FIRST element of the Segment List. In other words, the Segment List is encoded in the reverse order of the path.

The original DA of the packet is encoded as the last segment of the path (encoded in the first element of the Segment List).

The DA of the packet is set with the value of the first segment (found in the last element of the segment list).

The Segments Left field is set to n-1 where n is the number of elements in the Segment List.

The First Segment field is set to n-1 where n is the number of elements in the Segment List.

The packet is sent out towards the first segment (i.e.: represented in the packet DA).

HMAC and HMAC Key ID may be set according to Section 5.

4.1.2. SR Domain Ingress Node

The SR Domain Ingress Node is the node where ingress policies are applied and where the packet path (and processing) is determined.

After policies are applied and packet classification is done, the result may be instantiated into a Segment List representing the path the packet should take. In such case, the SR Domain Ingress Node instantiate a new outer IPv6 header to which the SRH is appended (with the computed Segment List). The procedures for the creation and insertion of the new SRH are described in Section 4.1.1.

4.1.3. Transit Node

According to [RFC2460], the only node who is allowed to inspect the Routing Extension Header (and therefore the SRH), is the node corresponding to the DA of the packet. Any other transit node **MUST NOT** inspect the underneath routing header and **MUST** forward the packet towards the DA and according to the IPv6 routing table.

In the example case described in Section 2.2.2, when SR capable nodes are connected through an overlay spanning multiple third-party infrastructure, it is safe to send SRH packets (i.e.: packet having a Segment Routing Header) between each other overlay/SR-capable nodes as long as the segment list does not include any of the transit provider nodes. In addition, as a generic security measure, any service provider will block any packet destined to one of its internal routers, especially if these packets have an extended header in it.

4.1.4. SR Segment Endpoint Node

The SR segment endpoint node is the node whose address is in the DA. The segment endpoint node inspects the SRH and does:

1. IF DA = myself (segment endpoint)
2. IF Segments Left > 0 THEN
 decrement Segments Left
 update DA with Segment List[Segments Left]
3. IF Segments Left == 0 THEN
 IF Clean-up bit is set THEN remove the SRH
4. ELSE give the packet to next PID (application)
 End of processing.
5. Forward the packet out

5. Security Considerations

This section analyzes the security threat model, the security issues and mitigation techniques of SRH.

SRH is simply another type of the routing header as described in RFC 2460 [RFC2460] and is:

- o added to a new outer IP header by the ingress router when entering the SR domain or by the originating node itself. The source host can be outside the SR domain;
- o inspected and acted upon when reaching the destination address of the IP header per RFC 2460 [RFC2460].

Per RFC2460 [RFC2460], routers on the path that simply forward an IPv6 packet (i.e. the IPv6 destination address is none of theirs) will never inspect and process the content of any routing header (including SRH). Routers whose one interface IPv6 address equals the destination address field of the IPv6 packet MUST to parse the SRH and, if supported and if the local configuration allows it, MUST act accordingly to the SRH content.

According to RFC2460 [RFC2460], non SR-capable (or non SR-configured) router upon receipt of an IPv6 packet with SRH destined to an address of its:

- o must ignore the SRH completely if the Segment Left field is 0 and proceed to process the next header in the IPv6 packet;
- o must discard the IPv6 packet if Segment Left field is greater than 0 and send a Parameter Problem ICMP message back to the Source Address.

5.1. Threat model

5.1.1. Source routing threats

Using a SRH is a specific case of loose source routing, therefore it has some well-known security issues as described in RFC4942 [RFC4942] section 2.1.1 and RFC5095 [RFC5095]:

- o amplification attacks: where a packet could be forged in such a way to cause looping among a set of SR-enabled routers causing unnecessary traffic, hence a Denial of Service (DoS) against bandwidth;
- o reflection attack: where a hacker could force an intermediate node to appear as the immediate attacker, hence hiding the real attacker from naive forensic;
- o bypass attack: where an intermediate node could be used as a stepping stone (for example in a De-Militarized Zone) to attack another host (for example in the datacenter or any back-end server).

5.1.2. Applicability of RFC 5095 to SRH

First of all, the reader must remember this specific part of section 1 of RFC5095 [RFC5095], "A side effect is that this also eliminates benign RH0 use-cases; however, such applications may be facilitated by future Routing Header specifications.". In short, it is not forbidden to create new secure type of Routing Header; for example, RFC 6554 (RPL) [RFC6554] also creates a new Routing Header type for a specific application confined in a single network.

The main use case for SR consists of the single administrative domain (or cooperating administrative domains) where only trusted nodes with SR enabled and explicitly configured participate in SR: this is the same model as in RFC6554 [RFC6554]. All non-trusted nodes do not participate as either SR processing is not enabled by default or because they only process SRH from nodes within their domain.

Moreover, all SR routers SHOULD ignore SRH created by outsiders based on topology information (received on a peering or internal interface) or on presence and validity of the HMAC field. Therefore, if intermediate SR routers ONLY act on valid and authorized SRH (such as within a single administrative domain), then there is no security threat similar to RH-0. Hence, the RFC 5095 [RFC5095] attacks are not applicable.

5.1.3. Service stealing threat

Segment routing is used for added value services, there is also a need to prevent non-participating nodes to use those services; this is called 'service stealing prevention'.

5.1.4. Topology disclosure

The SRH may also contains IPv6 addresses of some intermediate SR routers in the path towards the destination, this obviously reveals those addresses to the potentially hostile attackers if those attackers are able to intercept packets containing SRH. On the other hand, if the attacker can do a traceroute whose probes will be forwarded along the SR path, then there is little learned by intercepting the SRH itself. The clean-bit of SRH can help by removing the SRH before forwarding the packet to potentially a non-trusted part of the network; if the attacker can force the generation of an ICMP message during the transit in the SR domain, then the ICMP will probably contain the SRH header (totally or partially) depending on the ICMP-generating router behavior.

5.1.5. ICMP Generation

Per section 4.4 of RFC2460 [RFC2460], when destination nodes (i.e. where the destination address is one of theirs) receive a Routing Header with unsupported Routing Type, the required behavior is:

- o If Segments Left is zero, the node must ignore the Routing header and proceed to process the next header in the packet.
- o If Segments Left is non-zero, the node must discard the packet and SHOULD send an ICMP Parameter Problem, Code 0, message to the packet's Source Address, pointing to the unrecognized Routing Type.

This required behavior could be used by an attacker to force the generation of ICMP message by any node. The attacker could send packets with SRH (with Segment Left different than 0) destined to a node not supporting SRH. Per RFC2460 [RFC2460], the destination node must then generate an ICMP message per RFC 2460, causing a local CPU utilization and if the source of the offending packet with SRH was spoofed could lead to a reflection attack without any amplification.

It must be noted that this is a required behavior for any unsupported Routing Type and not limited to SRH packets. So, it is not specific to SRH and the usual rate limiting for ICMP generation is required anyway for any IPv6 implementation and has been implemented and deployed for many years.

5.2. Security fields in SRH

This section summarizes the use of specific fields in the SRH. They are based on a key-hashed message authentication code (HMAC).

The security-related fields in SRH are:

- o HMAC Key-id, 8 bits wide;
- o HMAC, 256 bits wide (optional, exists only if HMAC Key-id is not 0).

The HMAC field is the output of the HMAC computation (per RFC 2104 [RFC2104]) using a pre-shared key and hashing algorithm identified by HMAC Key-id and of the text which consists of the concatenation of:

- o the source IPv6 address;
- o First Segment field;
- o an octet whose bit-0 is the clean-up bit flag and others are 0;
- o HMAC Key-id;
- o all addresses in the Segment List.

The purpose of the HMAC field is to verify the validity, the integrity and the authorization of the SRH itself. If an outsider of the SR domain does not have access to a current pre-shared secret, then it cannot compute the right HMAC field and the first SR router on the path processing the SRH and configured to check the validity of the HMAC will simply reject the packet.

The HMAC field is located at the end of the SRH simply because only the router on the ingress of the SR domain needs to process it, then all other SR nodes can ignore it (based on local policy) because they trust the upstream router. This is to speed up forwarding operations because SR routers which do not validate the SRH do not need to parse the SRH until the end.

The HMAC Key-id field allows for the simultaneous existence of several hash algorithms (SHA-256, SHA3-256 ... or future ones) as well as pre-shared keys. This allows for pre-shared key roll-over when two pre-shared keys are supported for a while when all SR nodes converged to a fresher pre-shared key. The HMAC Key-id field is opaque, i.e., it has neither syntax nor semantic except as an index to the right combination of pre-shared key and hash algorithm and except that a value of 0 means that there is no HMAC field. It could

also allow for interoperation among different SR domains if allowed by local policy and assuming a collision-free Key Id allocation which is out of scope of this memo.

When a specific SRH is linked to a time-related service (such as turbo-QoS for a 1-hour period), then it is important to refresh the shared-secret frequently as the HMAC validity period expires only when the HMAC Key-id and its associated shared-secret expires.

5.2.1. Selecting a hash algorithm

The HMAC field in the SRH is 256 bits wide. Therefore, the HMAC MUST be based on a hash function whose output is at least 256 bits. If the output of the hash function is 256, then this output is simply inserted in the HMAC field. If the output of the hash function is larger than 256 bits, then the output value is truncated to 256 by taking the least-significant 256 bits and inserting them in the HMAC field.

SRH implementations can support multiple hash functions but MUST implement SHA-2 [FIPS180-4] in its SHA-256 variant.

NOTE: SHA-1 is currently used by some early implementations used for quick interoperations testing, the 160-bit hash value must then be right-hand padded with 96 bits set to 0. The authors understand that this is not secure but is ok for limited tests.

5.2.2. Performance impact of HMAC

While adding a HMAC to each and every SR packet increases the security, it has a performance impact. Nevertheless, it must be noted that:

- o the HMAC field SHOULD be used only when SRH is inserted by a device (such as a home set-up box) which is outside of the segment routing domain. If the SRH is added by a router in the trusted segment routing domain, then, there is no need for a HMAC field, hence no performance impact.
- o when present, the HMAC field MUST be checked and validated only by the first router of the segment routing domain, this router is named 'validating SR router'. Downstream routers may not inspect the HMAC field.
- o this validating router can also have a cache of <IPv6 header + SRH, HMAC field value> to improve the performance. It is not the same use case as in IPsec where HMAC value was unique per packet, in SRH, the HMAC value is unique per flow.

- o Last point, hash functions such as SHA-2 have been optimized for security and performance and there are multiple implementations with good performance.

With the above points in mind, the performance impact of using HMAC is minimized.

5.2.3. Pre-shared key management

The field HMAC Key-id allows for:

- o key roll-over: when there is a need to change the key (the hash pre-shared secret), then multiple pre-shared keys can be used simultaneously. The validating routing can have a table of <HMAC Key-id, pre-shared secret, hash algorithm> for the currently active and future keys.
- o different algorithm: by extending the previous table to <HMAC Key-id, hash function, pre-shared secret>, the validating router can also support simultaneously several hash algorithms (see section Section 5.2.1)

The pre-shared secret distribution can be done:

- o in the configuration of the validating routers, either by static configuration or any SDN oriented approach;
- o dynamically using a trusted key distribution such as [RFC6407]

The intent of this document is NOT to define yet-another-key-distribution-protocol.

5.3. Deployment Models

5.3.1. Nodes within the SR domain

The routers inside a SR domain can be trusted to generate the outer IP header and the SRH and to process SRH received on interfaces that are part of the SR domain. These nodes MUST drop all SRH packets received on any interface that is not part of the SR domain and containing a SRH whose HMAC field cannot be validated by local policies. This includes obviously packet with a SRH generated by a non-cooperative SR domain.

If the validation fails, then these packets MUST be dropped, ICMP error messages (parameter problem) SHOULD be generated (but rate limited) and SHOULD be logged.

5.3.2. Nodes outside of the SR domain

Nodes outside of the SR domain cannot be trusted for physical security; hence, they need to obtain by some trusted means (outside of the scope of this document) a complete SRH for each new connection (i.e. new destination address). The received SRH MUST include a HMAC Key-id and HMAC field which has been computed correctly (see Section 5.2).

When a outside the SR domain sends a packet with a SRH and towards a SR domain ingress node, the packet MUST contain the HMAC Key-id and HMAC field and the destination address MUST be an address of a SR domain ingress node .

The ingress SR router, i.e., the router with an interface address equals to the destination address, MUST verify the HMAC field with respect to the HMAC Key-id.

If the validation is successful, then the packet is simply forwarded as usual for a SR packet. As long as the packet travels within the SR domain, no further HMAC check needs to be done. Subsequent routers in the SR domain MAY verify the HMAC field when they process the SRH (i.e. when they are the destination).

If the validation fails, then this packet MUST be dropped, an ICMP error message (parameter problem) SHOULD be generated (but rate limited) and SHOULD be logged.

5.3.3. SR path exposure

As the intermediate SR nodes addresses appears in the SRH, if this SRH is visible to an outsider then he/she could reuse this knowledge to launch an attack on the intermediate SR nodes or get some insider knowledge on the topology. This is especially applicable when the path between the source node and the first SR domain ingress router is on the public Internet.

The first remark is to state that 'security by obscurity' is never enough; in other words, the security policy of the SR domain SHOULD assume that the internal topology and addressing is known by the attacker.

IPsec Encapsulating Security Payload [RFC4303] cannot be use to protect the SRH as per RFC4303 the ESP header must appear after any routing header (including SRH).

When the SRH is not generated by the actual source node but by an SR domain ingress router, it is added after a new outer IP header, this

means that a normal traceroute will not reveal the routers in the SR domain (pretty much like in a MPLS network) and that if ICMP are generated by routers in the SR domain they will be sent to the ingress router of the SR domain without revealing anything to the outside of the SR domain.

To prevent a user to leverage the gained knowledge by intercepting SRH, it is recommended to apply an infrastructure Access Control List (iACL) at the edge of the SR domain. This iACL will drop all packets from outside the SR-domain whose destination is any address of any router inside the domain. This security policy should be tuned for local operations.

5.3.4. Impact of BCP-38

BCP-38 [RFC2827], also known as "Network Ingress Filtering", checks whether the source address of packets received on an interface is valid for this interface. The use of loose source routing such as SRH forces packets to follow a path which differs from the expected routing. Therefore, if BCP-38 was implemented in all routers inside the SR domain, then SR packets could be received by an interface which is not expected one and the packets could be dropped.

As a SR domain is usually a subset of one administrative domain, and as BCP-38 is only deployed at the ingress routers of this administrative domain and as packets arriving at those ingress routers have been normally forwarded using the normal routing information, then there is no reason why this ingress router should drop the SRH packet based on BCP-38. Routers inside the domain commonly do not apply BCP-38; so, this is not a problem.

6. IANA Considerations

TBD but should at least require a new type for routing header

7. Manageability Considerations

TBD should we talk about traceroute? about SRH in ICMP replies?

8. Contributors

The authors would like to thank Dave Barach, John Leddy, John Brzozowski, Pierre Francois, Nagendra Kumar, Mark Townsley, Christian Martin, Roberta Maglione, James Connolly, Aloys Augustin and Fred Baker for their contribution to this document.

9. Acknowledgements

TBD

10. References

10.1. Normative References

[FIPS180-4]

National Institute of Standards and Technology, "FIPS 180-4 Secure Hash Standard (SHS)", March 2012, <<http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>>.

[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>>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<http://www.rfc-editor.org/info/rfc4303>>.

[RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, DOI 10.17487/RFC5095, December 2007, <<http://www.rfc-editor.org/info/rfc5095>>.

[RFC6407] Weis, B., Rowles, S., and T. Hardjono, "The Group Domain of Interpretation", RFC 6407, DOI 10.17487/RFC6407, October 2011, <<http://www.rfc-editor.org/info/rfc6407>>.

10.2. Informative References

[I-D.ietf-isis-segment-routing-extensions]

Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-05 (work in progress), June 2015.

- [I-D.ietf-ospf-ospfv3-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H.,
Shakir, R., Henderickx, W., and J. Tantsura, "OSPFv3
Extensions for Segment Routing", draft-ietf-ospf-ospfv3-
segment-routing-extensions-03 (work in progress), June
2015.
- [I-D.ietf-spring-ipv6-use-cases]
Brzozowski, J., Leddy, J., Leung, I., Previdi, S.,
Townsend, W., Martin, C., Filsfils, C., and R. Maglione,
"IPv6 SPRING Use Cases", draft-ietf-spring-ipv6-use-
cases-05 (work in progress), September 2015.
- [I-D.ietf-spring-problem-statement]
Previdi, S., Filsfils, C., Decraene, B., Litkowski, S.,
Horneffer, M., and R. Shakir, "SPRING Problem Statement
and Requirements", draft-ietf-spring-problem-statement-04
(work in progress), April 2015.
- [I-D.ietf-spring-resiliency-use-cases]
Francois, P., Filsfils, C., Decraene, B., and R. Shakir,
"Use-cases for Resiliency in SPRING", draft-ietf-spring-
resiliency-use-cases-01 (work in progress), March 2015.
- [I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Decraene, B., Litkowski, S.,
and r. rjs@rob.sh, "Segment Routing Architecture", draft-
ietf-spring-segment-routing-05 (work in progress),
September 2015.
- [I-D.ietf-spring-segment-routing-mpls]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B.,
Litkowski, S., Horneffer, M., Shakir, R., Tantsura, J.,
and E. Crabbe, "Segment Routing with MPLS data plane",
draft-ietf-spring-segment-routing-mpls-01 (work in
progress), May 2015.
- [RFC1940] Estrin, D., Li, T., Rekhter, Y., Varadhan, K., and D.
Zappala, "Source Demand Routing: Packet Format and
Forwarding Specification (Version 1)", RFC 1940,
DOI 10.17487/RFC1940, May 1996,
<<http://www.rfc-editor.org/info/rfc1940>>.
- [RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-
Hashing for Message Authentication", RFC 2104,
DOI 10.17487/RFC2104, February 1997,
<<http://www.rfc-editor.org/info/rfc2104>>.

- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <<http://www.rfc-editor.org/info/rfc2827>>.
- [RFC4942] Davies, E., Krishnan, S., and P. Savola, "IPv6 Transition/ Co-existence Security Considerations", RFC 4942, DOI 10.17487/RFC4942, September 2007, <<http://www.rfc-editor.org/info/rfc4942>>.
- [RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, DOI 10.17487/RFC6554, March 2012, <<http://www.rfc-editor.org/info/rfc6554>>.

Authors' Addresses

Stefano Previdi (editor)
Cisco Systems, Inc.
Via Del Serafico, 200
Rome 00142
Italy

Email: sprevidi@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Brussels
BE

Email: cfilsfil@cisco.com

Brian Field
Comcast
4100 East Dry Creek Road
Centennial, CO 80122
US

Email: Brian_Field@cable.comcast.com

Ida Leung
Rogers Communications
8200 Dixie Road
Brampton, ON L6T 0C1
CA

Email: Ida.Leung@rci.rogers.com

Jen Linkova
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043
US

Email: furry@google.com

Ebben Aries
Facebook
US

Email: exa@fb.com

Tomoya Kosugi
NTT
3-9-11, Midori-Cho Musashino-Shi,
Tokyo 180-8585
JP

Email: kosugi.tomoya@lab.ntt.co.jp

Eric Vyncke
Cisco Systems, Inc.
De Kleetlaann 6A
Diegem 1831
Belgium

Email: evyncke@cisco.com

David Lebrun
Universite Catholique de Louvain
Place Ste Barbe, 2
Louvain-la-Neuve, 1348
Belgium

Email: david.lebrun@uclouvain.be

Network Working Group
INTERNET-DRAFT
Updates RFC 3972 (if approved)
Intended Status: Standards Track
Expires: February 11, 2015

H.Rafiee
D. Zhang
Huawei Technologies
August 11, 2014

CGA Security Improvement
<draft-rafiee-rfc3972-bis-00.txt>

Abstract

This document addresses the security problems existing in the current CGA specification. It also explain the changes that is needed to take into consideration when the prefix length needs to be variable.

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 11, 2015.

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	3
2. Sec Value Solution	3
3. CGA and Challenges in Variable Length Prefix	4
4. Security Considerations	4
5. IANA Considerations	4
6. References	4
6.1. Normative	4
Authors' Addresses	6

1. Introduction

In the Cryptographically Generated Addresses (CGA) specification [RFC3972], the 64 rightmost bits of an IPv6 address is securely generated with a public key. This solution is able to provides the proof of IP address ownership and then prevent source IP spoofing by finding a binding between the public key and the node's IP address. Unfortunately, during the verification step as explained in [cga-attack], the verifier nodes ignore the 3 bits sec value in the interface ID (IID) and there is no check between the source and target IP address. This problem lead to the case where an attacker can calculate a new CGA address which is identical to the address of the victim node except its sec value field is zero. This document tries to explain how to address this problem.

This document also tries to explain how CGA specification needs to be changed when it is expected to support variable prefix.

2. Sec Value Solution

Sec value in CGA algorithm is the value between 0 to 7. This value shows the strengthen of the algorithm against brute-force attacks. As higher this value is, the more expensive and complicated the algorithm is for the attacker.

As explained in [cga-attack], since there is no check between the source and target addresses and the node ignores 3 bits sec values during verification process, an attacker can try to perform brute-force attacks without being detected. In other words, it does not matter what sec value the legitimate node uses, the attacker can always generate a new CGA address identical to the address of the victim except of the sec value field, and use the address to impersonate the legal node without being detected. To address this problem, we propose the changes in the following section of RFC 3972:

- Section 5. new step MUST be placed before step 1 of verification.

- 1- If the sender's source address is not a multicast IP address, then the verifier node MUST compare the sender's source address with its own local and global IP addresses. If there is a match it starts the other verification steps. Otherwise, it discards the message silently.

If the sender's source address is a multicast IP address but the target address is a unicast IP address, then the verifier node MUST compare the target address with its own local and global IP addresses. If there is a match then it MUST process the other verification steps. If there is no match, it should discard the

message silently.

3. CGA and Challenges in Variable Length Prefix

CGA algorithm, by default, uses a 64-bit prefix. The output of this algorithm is a 64-bit IID. This value is the result of hashing function on CGA parameters and taking only 64 bits of the hashing result (digest). To conform CGA with a dynamic prefix length, the number of bits which are taken from the hashing value should be the same size. Having a dynamic prefix, as explained in [cga-attack], might lead to the case where the attacker claim the address ownership of other legitimate nodes with different prefix values. This is specially true and feasible when prefixes are longer than 64 bits. In other words, less bits are available for Interface ID.

4. Security Considerations

There is no security consideration

5. IANA Considerations

There is no IANA consideration

6. References

6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.
- [RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.
- [RFC7136] Carpenter, B., Jiang, S., "Significance of IPv6 Interface Identifiers", RFC 7136, February 2014.
- [cga-attack] Rafiee, H., Meinel, C., "Possible Attack on Cryptographically Generated Addresses (CGA)", <http://tools.ietf.org/html/draft-rafiee-6man-cga-attack>, Augst 2014
- [variableprefix] Carpenter, B., Chown, T, Gont, F., Jiang, S., Petrescu, A., Yourtchenko, A., "Analysis

of the 64-bit Boundary in IPv6 Addressing",
<http://tools.ietf.org/html/draft-ietf-6man-why64> ,
April 2014

Authors' Addresses

Hosnieh Rafiee
HUAWEI TECHNOLOGIES Duesseldorf GmbH
Riesstrasse 25, 80992,
Munich, Germany
Phone: +49 (0)162 204 74 58
Email: hosnieh.rafiiee@huawei.com

Dacheng Zhang
HUAWEI TECHNOLOGIES
Q14 huawei campus, Beiqing Rd., Haidian Dist.,
Beijing, China
E-mail: zhangdacheng@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 5, 2014

B. Sarikaya
Huawei USA
June 3, 2014

IPv6 RA Options for Next Hop Routes
draft-sarikaya-6man-next-hop-ra-02

Abstract

This document proposes new Router Advertisement options for configuring next hop routes on the mobile or fixed nodes. Using these options, an operator can easily configure nodes with multiple interfaces (or otherwise multi-homed) to enable them to select the routes to a destination. Each option is defined together with definitions of host and router behaviors. This document also proposes the router advertisement extensions for source address dependent routing.

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 December 5, 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. Default Route Configuration	3
4. Source Address Dependent Routing	4
5. Host Configuration	4
6. Router Configuration	5
7. RA Packet Size and Router Issues	6
8. Route Prefix option	7
9. Next Hop Address option	7
10. Source Address/Prefix option	8
11. Next Hop Address with Route Prefix option	8
12. Next Hop Address with Source Address and Route Prefix option	9
13. Security Considerations	10
14. IANA Considerations	11
15. Acknowledgements	11
16. References	11
16.1. Normative References	11
16.2. Informative References	12
Author's Address	12

1. Introduction

IPv6 Neighbor Discovery and IPv6 Stateless Address Autoconfiguration protocols can be used to configure fixed and mobile nodes with various parameters related to addressing and routing [RFC4861], [RFC4862], [RFC4191]. DNS Recursive Server Addresses and Domain Name Search Lists are additional parameters that can be configured using router advertisements [RFC6106].

Router Advertisements can also be used to configure fixed and mobile nodes in multi-homed scenarios with route information and next hop address. Different scenarios exist such as the node is simultaneously connected to multiple access network of e.g. WiFi and 3G. The node may also be connected to more than one gateway. Such connectivity may be realized by means of dedicated physical or logical links that may also be shared with other users nodes such as in residential access networks.

Host configuration can be done using DHCPv6 or using router advertisements. A comparison of DHCPv6 and RA based host configuration approaches is presented in [I-D.yourtchenko-ra-dhcpv6-comparison].

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 [RFC2119].

3. Default Route Configuration

A host, usually a mobile host interested in obtaining routing information usually sends a Router Solicitation (RS) message on the link. The router, when configured to do so, provides the route information using zero, one or more Next Hop Address and Route Information options in the router advertisement (RA) messages sent in response.

The route options are extensible, as well as convey detailed information for routes.

RS and RA exchange is for next hop address and route information determination and not for determining the link-layer address of the router. Subsequent Neighbor Solicitation and Neighbor Advertisement exchange can be used to determine link-layer address of the router.

It should be noted that the proposed options in this document will need a central site-wide configuration mechanism. The required values can not automatically be derived from routing tables.

Next hop address and related route information may be provided by some other means such as directly by the next hop routers. In this document we assume that next hop routers are not able to provide this information. One solution would be to develop an inter-router protocol to instigate the next hop routers to provide this information. However, such a solution has been singled out due to the complexities involved.

A non-trustworthy network may be available at the same time as a trustworthy network, with the risk of bad consequences if the host gets confused between the two. These are basically the two models for hosts with multiple interfaces, both of which are valid, but which are incompatible with each other. In the first model, an interface is connected to something like a corporate network, over a Virtual Private Network (VPN). This connection is trusted because it has been authenticated. Routes obtained over such a connection can probably be trusted, and indeed it may be important to use those routes. This is because in the VPN case, you may also be connected to a network that's offered you a default route, and you could be attacked over that connection if you attempt to connect to resources on the enterprise network over it.

On the other, non-trustworthy network scenario, none of the networks to which the host is connected are meaningfully more or less trustworthy. In this scenario, the untrustworthy network may hand out routes to other hosts, e.g. those in the VPN going through some malicious nodes. This will have bad consequences because the host's traffic intended for the corporate VPN may be hijacked by the intermediate nodes.

Router advertisement extensions described in this document can be used to install the routes. However, the use of such a technique makes sense only in the former case above, i.e. trusted network. So the host MUST have an authenticated connection to the network it connects so that the router advertisements can be trusted before establishing routes.

4. Source Address Dependent Routing

In multihomed networks there is a need to do source address based routing if some providers are performing the ingress filtering defined in BCP38 [RFC2827]. This requires the routers to consider the source addresses as well as the destination addresses in determining the next hop to send the packet to.

The routers may be informed about the source addresses to use in routing using extensions to the routing protocols like IS-IS defined in [ISO.10589.1992] [I-D.baker-ipv6-isis-dst-src-routing] and OSPF defined in [RFC5340] [I-D.baker-ipv6-ospf-dst-src-routing]. In this document we define the router advertisement extensions for source address dependent routing.

Routing protocol extensions for source address dependent routing does not avoid a host using a source address that may be subject to ingress filtering when sending a packet to one of the next hops. In that case the host receives an ICMP source address failed ingress/egress policy error message in which case the host must resend the packet trying a different source address. The extensions defined in this document aims at avoiding this inefficiency in packet forwarding at the host.

5. Host Configuration

Router advertisement options defined in this document are used by Type C hosts.

As defined in [RFC4191] Type C host uses a Routing Table instead of a Default Router List.

The hosts set up their routing tables based on the router advertisement extensions defined in this document. The routes established are used in forwarding the packets to a next hop based on the destination prefix/address using the longest match algorithm.

In case the host receives Next Hop Address with Source Address and Route Prefix option, the host uses source and destination prefix/address using the longest match algorithm in order to select the next hop to forward the packet to.

6. Router Configuration

The router MAY send one or more Next Hop Address that specify the IPv6 next hop addresses. Each Next Hop Address may be associated with one or more Route Prefix options that represent the IPv6 destination prefixes reachable via the given next hop. Router includes Route Prefix option in message to indicate that given prefix is available directly on-link. When router sends Next Hop Address that is associated with Router Prefix option, the router MUST use Next Hop Address with Route Prefix option defined in Section 11. The Route Prefix MAY contain `::/0`, i.e. with Prefix Length set to zero to indicate available default route.

The router MAY send one or more Next Hop Address options that specify the IPv6 next hop addresses and source address. Each Next Hop Address may be associated with zero, one or more Source Prefix that represent the source addresses that are assigned from the prefixes that belong to this next hop. The option MAY contain Route Prefix options that represent the IPv6 destination prefixes reachable via the given next hop as defined in Figure 4. Router includes Next Hop Address with Route Prefix option and Source Prefix in the message to indicate that given prefix is available directly on-link and that any source addresses derived from the source prefix will not be subject to ingress filtering on these routes supported by these next hops.

The router MAY send one or more Next Hop Address that specify the IPv6 next hop addresses and source address. Each Next Hop Address option may be associated with zero, one or more Source Address that represent the source addresses that are assigned from the prefixes that belong to this next hop. The option MAY contain Route Prefix options that represent the IPv6 destination prefixes reachable via the given next hop defined in Figure 5. Router includes Next Hop Address with Source Address and Route Prefix option in the message to indicate that given prefix is available directly on-link and that the source address will not be subject to ingress filtering. For the Source Address, Source Prefix option is used with prefix length set to 128.

Each Next Hop Address may be associated with zero, one or more Source Prefix that represent the source addresses that are assigned from the prefixes that belong to this next hop. The option MAY contain Route Prefix options that represent the IPv6 destination prefixes reachable via the given next hop. Router includes Next Hop Address with Route Prefix option defined in Section 11 in the message to indicate that given prefix is available directly on-link. Next Hop Address with Route Prefix option MUST be followed by a Source Prefix option defined in Section 10 to indicate that any source addresses derived from the source prefix will not be subject to ingress filtering on these routes supported by these next hops.

7. RA Packet Size and Router Issues

The options defined in this document are to be used on multi-homed hosts. A mobile host would typically have two interfaces, Wi-Fi and 3G but hosts with 3 or 4 interfaces may also exist. Configuring such hosts using the options defined in this document brings up the RA packet size issue, i.e. the packet size should not exceed the maximum transmission unit (MTU) of the link.

Total size of all options defined in this document is 160 octets. Considering that 1500 bytes is the minimum MTU configured by the vast majority of links in the Internet the hosts with 3-4 interfaces or links can be easily configured by a single router advertisement message carrying the options defined here.

The router before sending the RA SHOULD check if it fits in one frame, i.e. the size does not exceed the path MTU, the router should send a single RA message. If it does not then sending the options in consecutive RA messages should be considered, avoiding any re-assembly issues.

The routes advertised have route lifetime values. The host considers the routes in its routing table stale when the lifetime expires. The router MUST refresh these routes periodically in order to avoid stale routing table entries in the hosts.

In some cases the mobile devices with multiple interfaces become routers. Such devices may configure their routing tables using routing protocols such as RIPng or OSPFv3 [RFC7157]. RA based approach described in this document can also be used to configure such hosts.

8. Route Prefix option

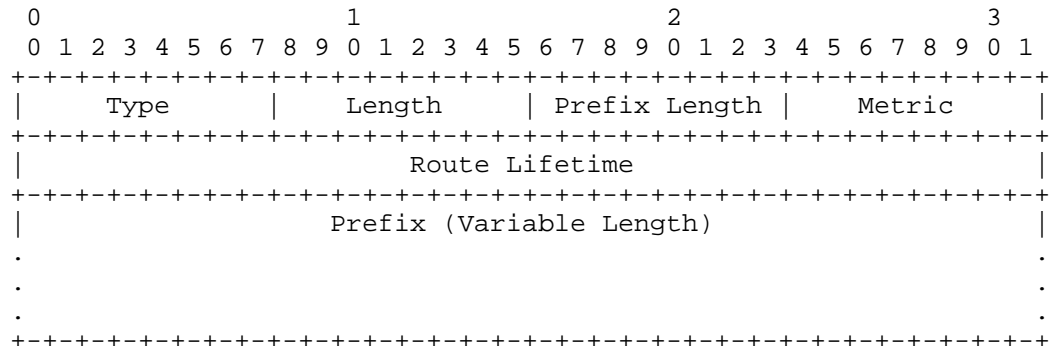


Figure 1: Route Prefix option

Fields:

Type: TBD.

Length: The length of the option (including the Type and Length fields) in units of 8 octets.

Other fields are as in [RFC4191] except:

Metric: Route Metric. 8-bit signed integer. The Route Metric indicates whether to prefer the next hop associated with this prefix over others, when multiple identical prefixes (for different next hops) have been received.

9. Next Hop Address option

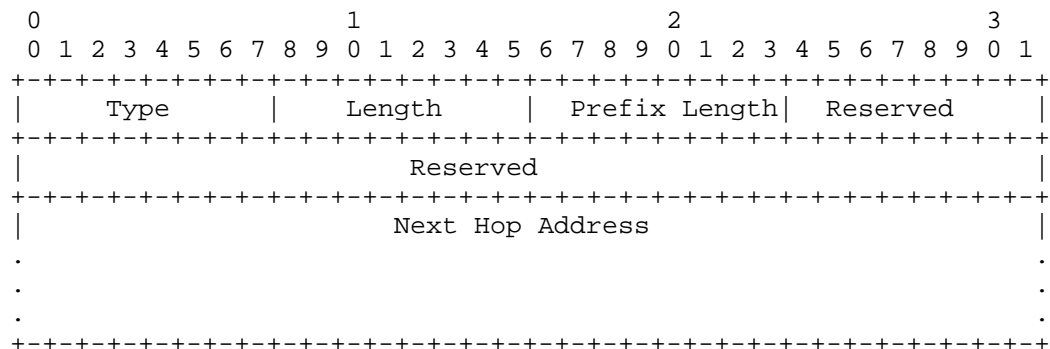


Figure 2: Next Hop Address option

Fields:

Type: TBD.

Length: The length of the option (including the type and length fields) in units of 8 octets. It's value is 3.

Prefix Length: 128

Next Hop Address: An IPv6 address that specifies IPv6 address of the next hop. It is 16 octets in length.

10. Source Address/Prefix option

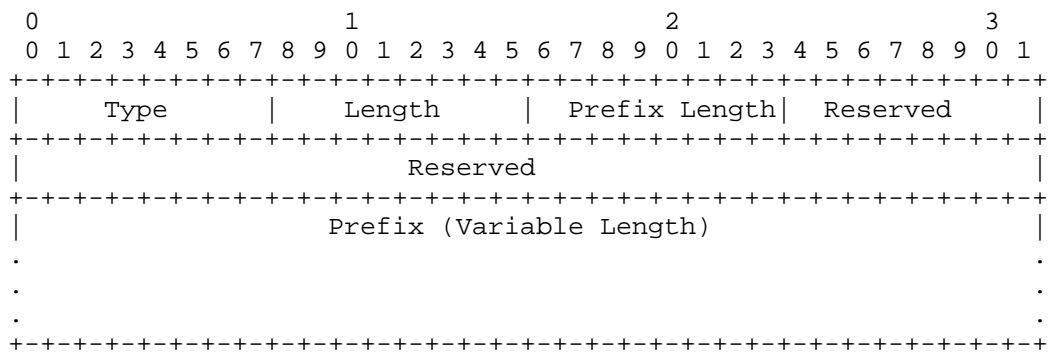


Figure 3: Source Address/Prefix option

Fields:

Type: TBD.

Length: The length of the option (including the type and length fields) in units of 8 octets. It's value is 3.

Prefix Length: An IPv6 prefix length in bits, from 0 to 128.

Prefix: An IPv6 prefix that specifies the source IPv6 prefix. It is 16 octets or less in length. Note that when the prefix length is set to 128, this option becomes a source address option.

11. Next Hop Address with Route Prefix option

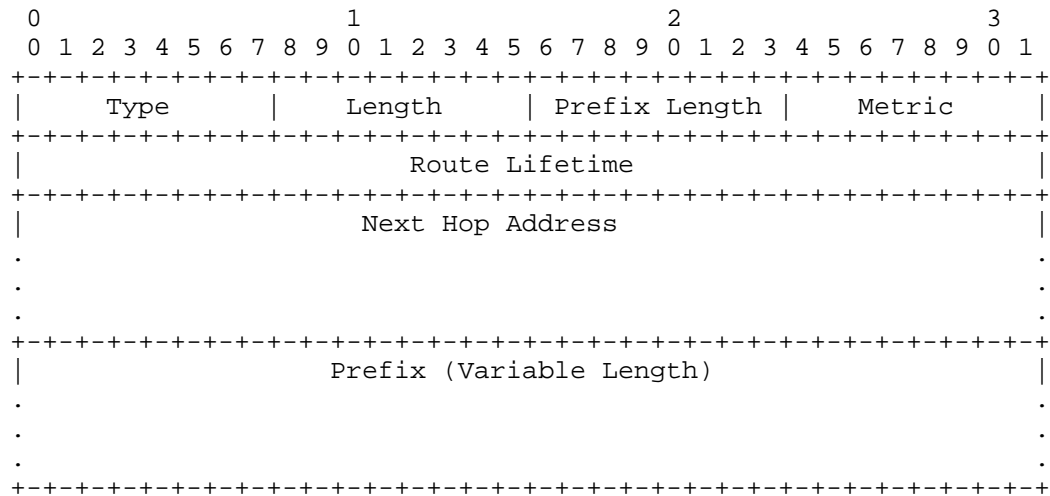


Figure 4: Next Hop Address with Route Prefix option

Fields:

Type: TBD.

Length: The length of the option (including the type and length fields) in units of 8 octets. For example, the length for a prefix of length 16 is 5.

Other fields are as in Section 8 and Section 9.

12. Next Hop Address with Source Address and Route Prefix option

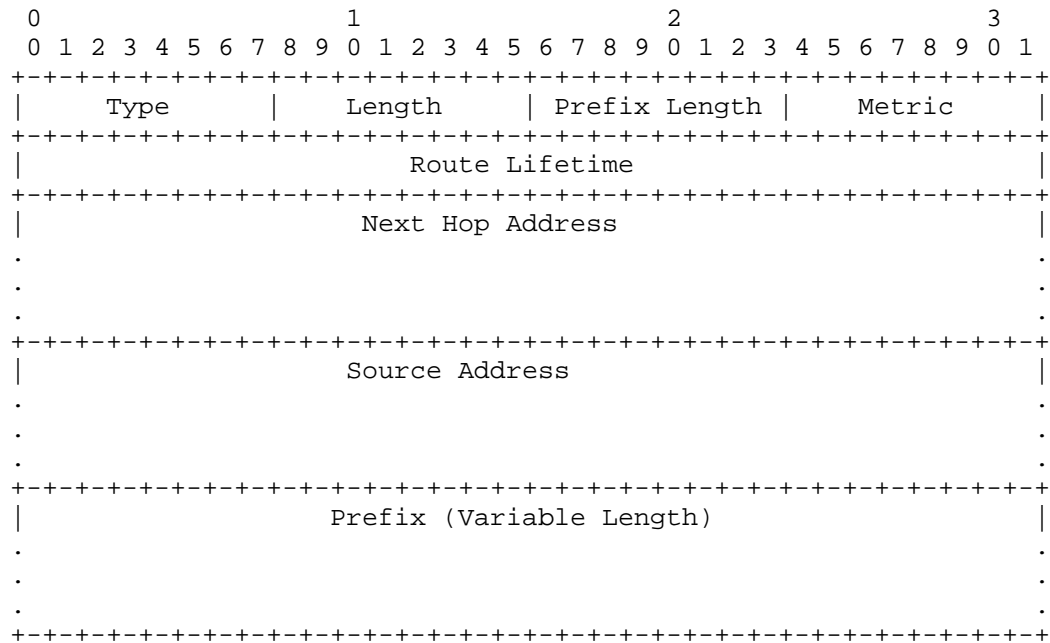


Figure 5: Next Hop Address with Source Address and Route Prefix option

Fields:

Type: TBD.

Length: The length of the option (including the type and length fields) in units of 8 octets. For example, the length for a prefix of length 16 is 7.

Other fields are as in Section 8, Section 9 and Section 10. Note that when prefix length is set to 128, the source prefix field refers to the source address.

13. Security Considerations

Neighbor Discovery is subject to attacks that cause IP packets to flow to unexpected places. Because of this, neighbor discovery messages SHOULD be secured, possibly using Secure Neighbor Discovery (SEND) protocol [RFC3971].

14. IANA Considerations

Authors of this document request IANA to assign the following new RA options:

Option Name	Type
Route Prefix	
Next Hop Address	
Source Address/Prefix	
Next Hop Address and Route Prefix	
Next Hop Address with Source Address and Route Prefix	

Table 1:

15. Acknowledgements

Dan Luedtke, Brian Carpenter, Ray Hunter provided many comments that have been incorporated into the document.

16. References

16.1. Normative References

- [ISO.10589.1992] International Organization for Standardization, "Intermediate system to intermediate system intra-domain-routing routine information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473), ISO Standard 10589", ISO ISO.10589.1992, 1992.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

- [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.
- [RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 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.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.
- [RFC7157] Troan, O., Miles, D., Matsushima, S., Okimoto, T., and D. Wing, "IPv6 Multihoming without Network Address Translation", RFC 7157, March 2014.

16.2. Informative References

- [I-D.baker-ipv6-isis-dst-src-routing]
Baker, F., "IPv6 Source/Destination Routing using IS-IS", draft-baker-ipv6-isis-dst-src-routing-01 (work in progress), August 2013.
- [I-D.baker-ipv6-ospf-dst-src-routing]
Baker, F., "IPv6 Source/Destination Routing using OSPFv3", draft-baker-ipv6-ospf-dst-src-routing-03 (work in progress), August 2013.
- [I-D.yourtchenko-ra-dhcpv6-comparison]
Yourtchenko, A., "A comparison between the DHCPv6 and RA based host configuration", draft-yourtchenko-ra-dhcpv6-comparison-00 (work in progress), November 2013.
- [RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.

Author's Address

Behcet Sarikaya
Huawei USA
5340 Legacy Dr. Building 175
Plano, TX 75024

Email: sarikaya@ieee.org

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 11, 2015

B. Sarikaya
Huawei USA
December 8, 2014

IPv6 RA Options for Next Hop Routes
draft-sarikaya-6man-next-hop-ra-04

Abstract

This document proposes new Router Advertisement options for configuring next hop routes on the mobile or fixed nodes. Using these options, an operator can easily configure nodes with multiple interfaces (or otherwise multi-homed) to enable them to select the routes to a destination. Each option is defined together with definitions of host and router behaviors. This document also proposes the router advertisement extensions for source address dependent routing.

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 June 11, 2015.

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. Default Route Configuration	3
4. Source Address Dependent Routing	4
5. Host Configuration	5
6. Router Configuration	5
7. RA Packet Size and Router Issues	6
8. Route Prefix option	7
9. Next Hop Address option	8
10. Source Address/Prefix option	8
11. Next Hop Address with Route Prefix option	9
12. Next Hop Address with Source Address and Route Prefix option	9
13. Route Prefix with Source Address/Prefix Option	10
14. Security Considerations	11
15. IANA Considerations	11
16. Acknowledgements	12
17. References	12
17.1. Normative References	12
17.2. Informative References	13
Author's Address	14

1. Introduction

IPv6 Neighbor Discovery and IPv6 Stateless Address Autoconfiguration protocols can be used to configure fixed and mobile nodes with various parameters related to addressing and routing [RFC4861], [RFC4862], [RFC4191]. DNS Recursive Server Addresses and Domain Name Search Lists are additional parameters that can be configured using router advertisements [RFC6106].

Router Advertisements can also be used to configure fixed and mobile nodes in multi-homed scenarios with route information and next hop address. Different scenarios exist such as the node is simultaneously connected to multiple access network of e.g. WiFi and 3G. The node may also be connected to more than one gateway. Such connectivity may be realized by means of dedicated physical or logical links that may also be shared with other users nodes such as in residential access networks.

Host configuration can be done using DHCPv6 or using router advertisements. A comparison of DHCPv6 and RA based host

configuration approaches is presented in [I-D.yourtchenko-ra-dhcpv6-comparison].

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 [RFC2119].

3. Default Route Configuration

A host, usually a mobile host interested in obtaining routing information usually sends a Router Solicitation (RS) message on the link. The router, when configured to do so, provides the route information using zero, one or more Next Hop Address and Route Information options in the router advertisement (RA) messages sent in response.

The route options are extensible, as well as convey detailed information for routes.

RS and RA exchange is for next hop address and route information determination and not for determining the link-layer address of the router. Subsequent Neighbor Solicitation and Neighbor Advertisement exchange can be used to determine link-layer address of the router.

It should be noted that the proposed options in this document will need a central site-wide configuration mechanism. The required values can not automatically be derived from routing tables.

Next hop address and related route information may be provided by some other means such as directly by the next hop routers. In this document we assume that next hop routers are not able to provide this information. One solution would be to develop an inter-router protocol to instigate the next hop routers to provide this information. However, such a solution has been singled out due to the complexities involved.

A non-trustworthy network may be available at the same time as a trustworthy network, with the risk of bad consequences if the host gets confused between the two. These are basically the two models for hosts with multiple interfaces, both of which are valid, but which are incompatible with each other. In the first model, an interface is connected to something like a corporate network, over a Virtual Private Network (VPN). This connection is trusted because it has been authenticated. Routes obtained over such a connection can probably be trusted, and indeed it may be important to use those routes. This is because in the VPN case, you may also be connected

to a network that's offered you a default route, and you could be attacked over that connection if you attempt to connect to resources on the enterprise network over it.

On the other, non-trustworthy network scenario, none of the networks to which the host is connected are meaningfully more or less trustworthy. In this scenario, the untrustworthy network may hand out routes to other hosts, e.g. those in the VPN going through some malicious nodes. This will have bad consequences because the host's traffic intended for the corporate VPN may be hijacked by the intermediate nodes.

Router advertisement extensions described in this document can be used to install the routes. However, the use of such a technique makes sense only in the former case above, i.e. trusted network. So the host MUST have an authenticated connection to the network it connects so that the router advertisements can be trusted before establishing routes.

4. Source Address Dependent Routing

In multihomed networks there is a need to do source address based routing if some providers are performing the ingress filtering defined in BCP38 [RFC2827]. This requires the routers to consider the source addresses as well as the destination addresses in determining the next hop to send the packet to.

The routers may be informed about the source addresses to use in routing using extensions to the routing protocols like IS-IS defined in [ISO.10589.1992] [I-D.baker-ipv6-isis-dst-src-routing] and OSPF defined in [RFC5340] [I-D.baker-ipv6-ospf-dst-src-routing]. In this document we define the router advertisement extensions for source address dependent routing.

Routing protocol extensions for source address dependent routing does not avoid a host using a source address that may be subject to ingress filtering when sending a packet to one of the next hops. In that case the host receives an ICMP source address failed ingress/egress policy error message in which case the host must resend the packet trying a different source address. The extensions defined in this document aims at avoiding this inefficiency in packet forwarding at the host.

More information on the scenarios, their analysis and why host based approach to source address dependent routing is needed, are presented in [I-D.sarikaya-6man-sadr-overview].

5. Host Configuration

Router advertisement options defined in this document are used by Type C hosts.

As defined in [RFC4191] Type C host uses a Routing Table instead of a Default Router List.

The hosts set up their routing tables based on the router advertisement extensions defined in this document. The routes established are used in forwarding the packets to a next hop based on the destination prefix/address using the longest match algorithm. The hosts MUST keep Route Prefix that it received together with Next Hop Address, Source Address options in a stable storage. This will enable the host to consistently use these options as described next.

In case the host receives Next Hop Address with Source Address and Route Prefix option, the host uses source and destination prefix/address using the longest match algorithm in order to select the next hop to forward the packet to.

6. Router Configuration

The router MAY send one or more Next Hop Address that specify the IPv6 next hop addresses. Each Next Hop Address may be associated with one or more Route Prefix options that represent the IPv6 destination prefixes reachable via the given next hop. Router includes Route Prefix option in message to indicate that given prefix is available directly on-link. When router sends Next Hop Address that is associated with Route Prefix option, the router MUST use Next Hop Address with Route Prefix option defined in Section 11. The Route Prefix MAY contain `::/0`, i.e. with Prefix Length set to zero to indicate available default route.

The router MAY send one or more Next Hop Address options that specify the IPv6 next hop addresses and source address. Each Next Hop Address may be associated with zero, one or more Source Prefix that represent the source addresses that are assigned from the prefixes that belong to this next hop. The option MAY contain Route Prefix options that represent the IPv6 destination prefixes reachable via the given next hop as defined in Figure 4. Router includes Next Hop Address with Route Prefix option and Source Prefix in the message to indicate that given prefix is available directly on-link and that any source addresses derived from the source prefix will not be subject to ingress filtering on these routes supported by these next hops.

The router MAY send one or more Next Hop Address that specify the IPv6 next hop addresses and source address. Each Next Hop Address

option may be associated with zero, one or more Source Address that represent the source addresses that are assigned from the prefixes that belong to this next hop. The option MAY contain Route Prefix options that represent the IPv6 destination prefixes reachable via the given next hop defined in Figure 5. Router includes Next Hop Address with Source Address and Route Prefix option in the message to indicate that given prefix is available directly on-link and that the source address will not be subject to ingress filtering. For the Source Address, Source Prefix option is used with prefix length set to 128.

Each Next Hop Address may be associated with zero, one or more Source Prefix that represent the source addresses that are assigned from the prefixes that belong to this next hop. The option MAY contain Route Prefix options that represent the IPv6 destination prefixes reachable via the given next hop. Router includes Next Hop Address with Route Prefix option defined in Section 11 in the message to indicate that given prefix is available directly on-link. Next Hop Address with Route Prefix option MUST be followed by a Source Prefix option defined in Section 10 to indicate that any source addresses derived from the source prefix will not be subject to ingress filtering on these routes supported by these next hops.

In home networks, there is possibility of configuring each interface of the host using Router Advertisements sent from their next hop routers. This brings the need for a new option, Router Prefix with Source Address Option defined in Figure 6 to indicate that any source addresses derived from the source prefix will not be subject to ingress filtering on these routes supported by this router.

7. RA Packet Size and Router Issues

The options defined in this document are to be used on multi-homed hosts. A mobile host would typically have two interfaces, Wi-Fi and 3G but hosts with 3 or 4 interfaces may also exist. Configuring such hosts using the options defined in this document brings up the RA packet size issue, i.e. the packet size should not exceed the maximum transmission unit (MTU) of the link.

Total size of all options defined in this document is 160 octets. Considering that 1500 bytes is the minimum MTU configured by the vast majority of links in the Internet the hosts with 3-4 interfaces or links can be easily configured by a single router advertisement message carrying the options defined here.

The router before sending the RA SHOULD check if it fits in one frame, i.e. the size does not exceed the path MTU, the router should send a single RA message. If it does not then sending the options in

consecutive RA messages should be considered, avoiding any re-assembly issues.

The routes advertised have route lifetime values. The host considers the routes in its routing table stale when the lifetime expires. The router MUST refresh these routes periodically in order to avoid stale routing table entries in the hosts.

In some cases the mobile devices with multiple interfaces become routers. Such devices may configure their routing tables using routing protocols such as RIPng or OSPFv3 [RFC7157]. RA based approach described in this document can also be used to configure such hosts.

8. Route Prefix option

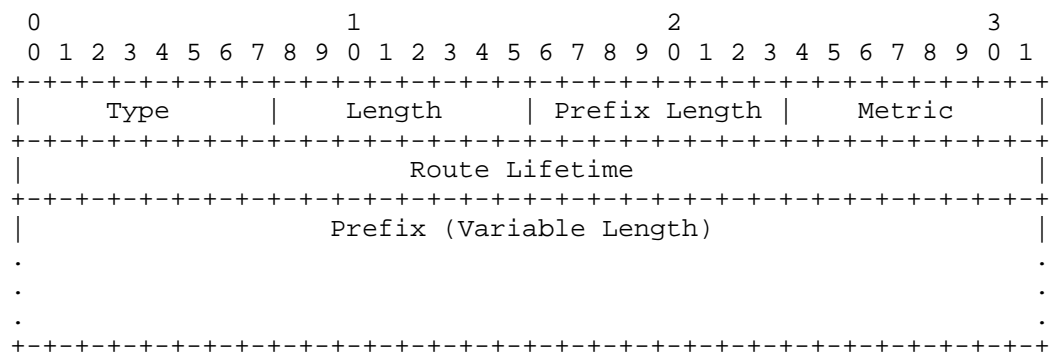


Figure 1: Route Prefix option

Fields:

Type: TBD.

Length: The length of the option (including the Type and Length fields) in units of 8 octets.

Other fields are as in [RFC4191] except:

Metric: Route Metric. 8-bit signed integer. The Route Metric indicates whether to prefer the next hop associated with this prefix over others, when multiple identical prefixes (for different next hops) have been received.

9. Next Hop Address option

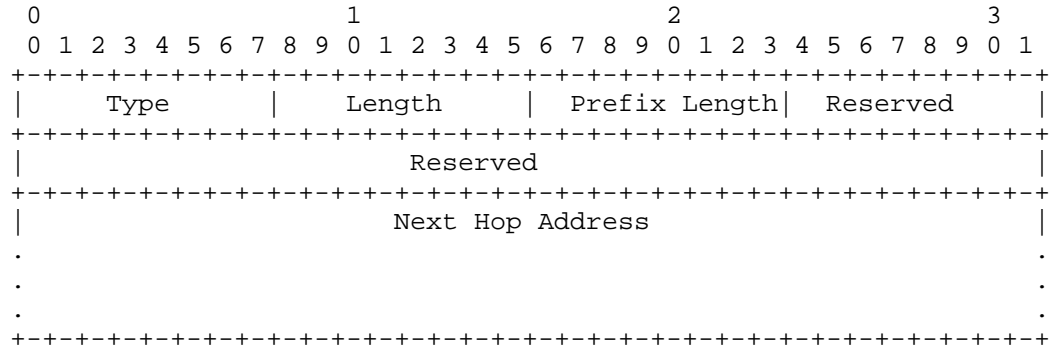


Figure 2: Next Hop Address option

Fields:

Type: TBD.

Length: The length of the option (including the type and length fields) in units of 8 octets. It's value is 3.

Prefix Length: 128

Next Hop Address: An IPv6 address that specifies IPv6 address of the next hop. It is 16 octets in length.

10. Source Address/Prefix option

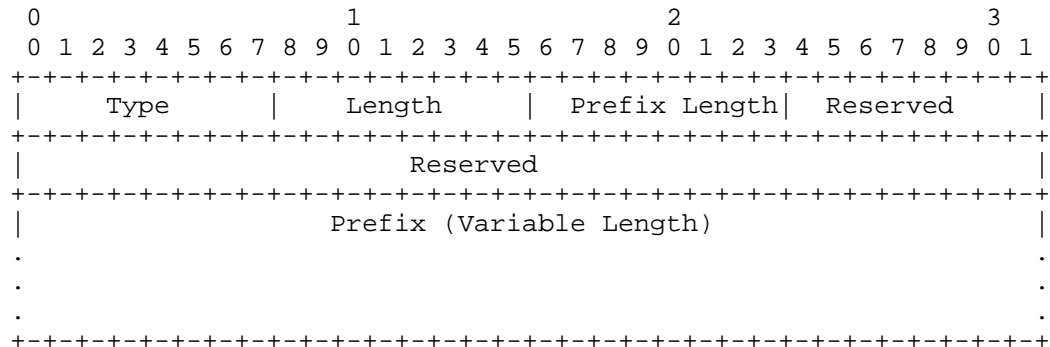


Figure 3: Source Address/Prefix option

Fields:

Type: TBD.

Length: The length of the option (including the type and length fields) in units of 8 octets. It's value is 3.

Prefix Length: An IPv6 prefix length in bits, from 0 to 128.

Prefix: An IPv6 prefix that specifies the source IPv6 prefix. It is 16 octets or less in length. Note that when the prefix length is set to 128, this option becomes a source address option.

11. Next Hop Address with Route Prefix option

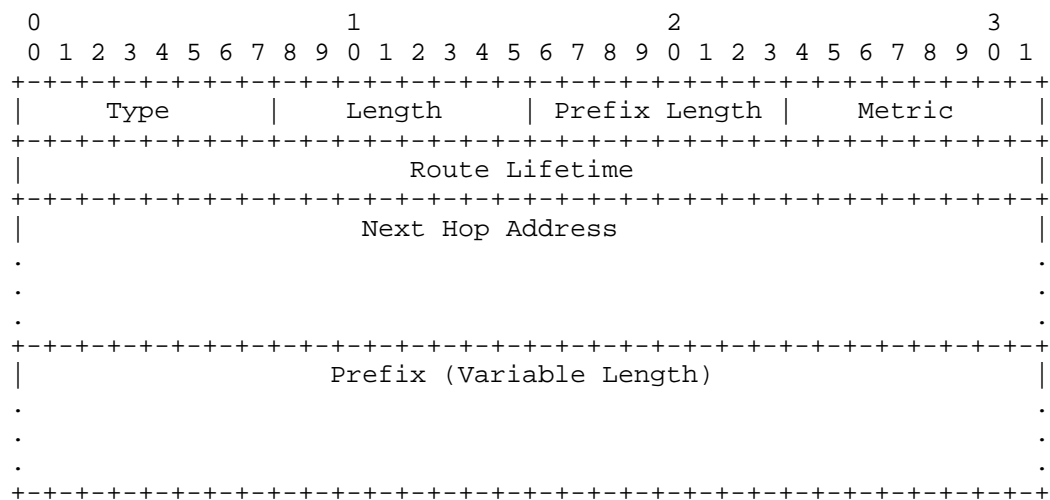


Figure 4: Next Hop Address with Route Prefix option

Fields:

Type: TBD.

Length: The length of the option (including the type and length fields) in units of 8 octets. For example, the length for a prefix of length 16 is 5.

Other fields are as in Section 8 and Section 9.

12. Next Hop Address with Source Address and Route Prefix option

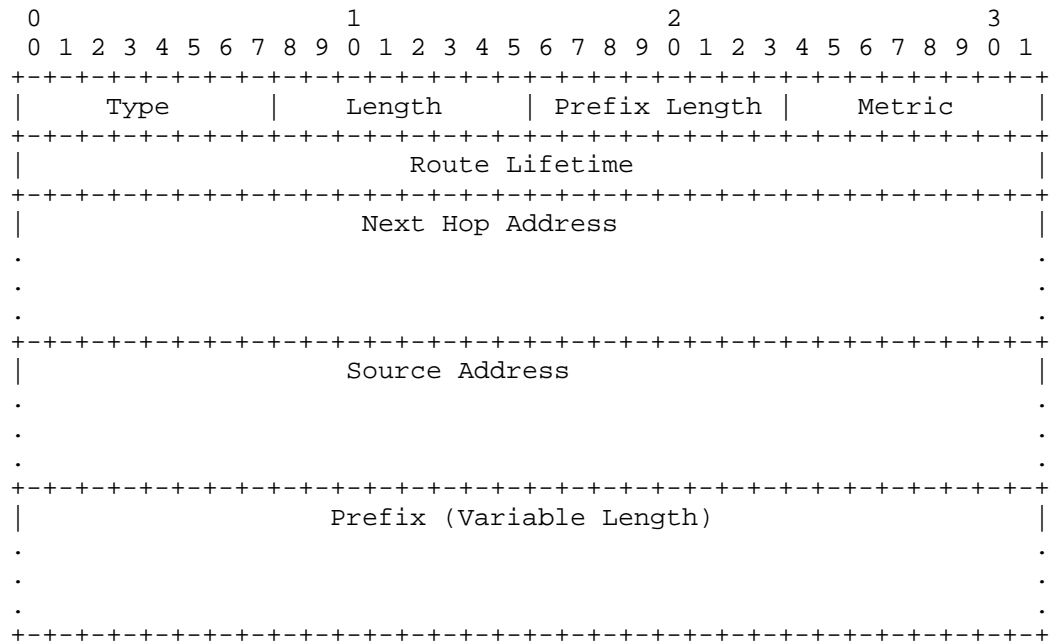


Figure 5: Next Hop Address with Source Address and Route Prefix option

Fields:

Type: TBD.

Length: The length of the option (including the type and length fields) in units of 8 octets. For example, the length for a prefix of length 16 is 7.

Other fields are as in Section 8, Section 9 and Section 10. Note that when prefix length is set to 128, the source prefix field refers to the source address.

13. Route Prefix with Source Address/Prefix Option

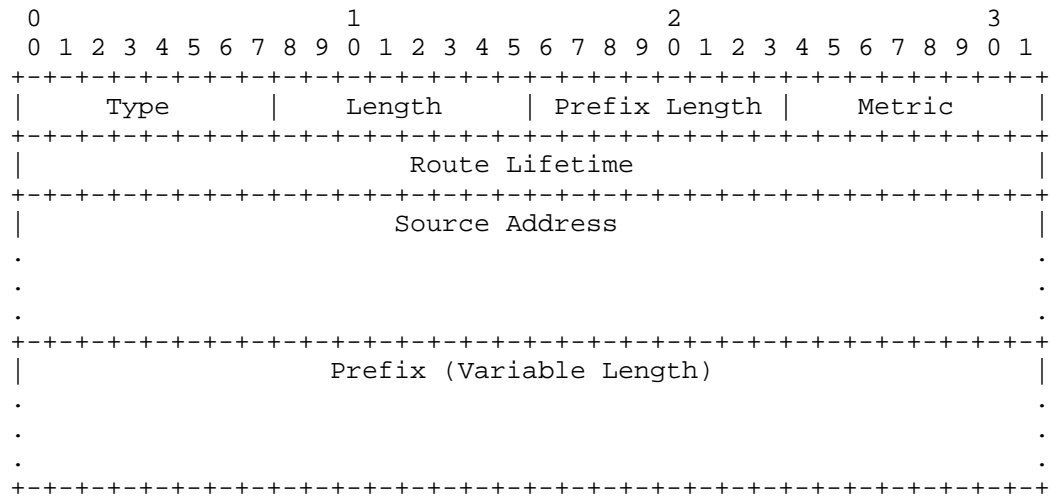


Figure 6: Route Prefix with Source Address option

Fields:

Type: TBD.

Length: The length of the option (including the type and length fields) in units of 8 octets. For example, the length for a prefix of length 16 is 5.

Other fields are as in Section 8 and Section 10.

14. Security Considerations

Neighbor Discovery is subject to attacks that cause IP packets to flow to unexpected places. Because of this, neighbor discovery messages SHOULD be secured, possibly using Secure Neighbor Discovery (SEND) protocol [RFC3971].

15. IANA Considerations

Authors of this document request IANA to assign the following new RA options:

Option Name	Type
Route Prefix	
Next Hop Address	
Source Address/Prefix	
Next Hop Address and Route Prefix	
Next Hop Address with Source Address and Route Prefix	
Route Prefix with Source Address	

Table 1:

16. Acknowledgements

Dan Luedtke, Brian Carpenter, Ray Hunter, Pierre Pfister provided many comments that have been incorporated into the document. Comments from Lorenzo Colitti, Ole Troan are much appreciated.

17. References

17.1. Normative References

- [ISO.10589.1992] International Organization for Standardization, "Intermediate system to intermediate system intra-domain-routing routine information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473), ISO Standard 10589", ISO ISO.10589.1992, 1992.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.
- [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.

- [RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 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.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.
- [RFC7157] Troan, O., Miles, D., Matsushima, S., Okimoto, T., and D. Wing, "IPv6 Multihoming without Network Address Translation", RFC 7157, March 2014.

17.2. Informative References

- [I-D.baker-ipv6-isis-dst-src-routing]
Baker, F. and D. Lamparter, "IPv6 Source/Destination Routing using IS-IS", draft-baker-ipv6-isis-dst-src-routing-02 (work in progress), October 2014.
- [I-D.baker-ipv6-ospf-dst-src-routing]
Baker, F., "IPv6 Source/Destination Routing using OSPFv3", draft-baker-ipv6-ospf-dst-src-routing-03 (work in progress), August 2013.
- [I-D.sarikaya-6man-sadr-overview]
Sarikaya, B., "Overview of Source Address Dependent Routing", draft-sarikaya-6man-sadr-overview-02 (work in progress), October 2014.
- [I-D.yourtchenko-ra-dhcpv6-comparison]
Yourtchenko, A., "A comparison between the DHCPv6 and RA based host configuration", draft-yourtchenko-ra-dhcpv6-comparison-00 (work in progress), November 2013.
- [RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 6106, November 2010.

Author's Address

Behcet Sarikaya
Huawei USA
5340 Legacy Dr. Building 175
Plano, TX 75024

Email: sarikaya@ieee.org

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 23, 2015

B. Sarikaya
Huawei USA
October 20, 2014

Overview of Source Address Dependent Routing
draft-sarikaya-6man-sadr-overview-02

Abstract

This document presents an overview of source address dependent routing from the host perspective. Multihomed hosts and hosts with multiple interfaces are considered. Different architectures are introduced and with their help, why source address selection and next hop resolution in view of source address dependent routing is needed is explained. The document concludes with a discussion on the standardization work that is needed.

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 April 23, 2015.

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. SADR Scenarios	4
4. Analysis of Source Address Dependent Routing	6
4.1. Scenarios Analysis	6
4.2. Provisioning Domains and SADR	7
5. What Needs to be Done	8
6. Security Considerations	9
7. IANA Considerations	9
8. Acknowledgements	9
9. References	9
9.1. Normative References	9
9.2. Informative References	11
Author's Address	12

1. Introduction

BCP 38 recommends ingress traffic routing to prohibit Denial of Service (DoS) attacks, i.e. datagrams which have source addresses that do not match with the network where the host is attached are discarded [RFC2827]. Avoiding packets to be dropped because of ingress filtering is difficult especially in multihomed networks where the host receives more than one prefix from the connected Internet Service Providers (ISP) and may have more than one source addresses. Based on BCP 38, BCP 84 introduced recommendations on the routing system for multihomed networks [RFC3704].

Recommendations on the routing system for ingress filtering such as in BCP 84 inevitably involve source address checks. This leads us to the source address dependent routing. Source address dependent routing is an issue especially when the host is connected to a multihomed network and is communicating with another host in another multihomed network. In such a case, the communication can be broken in both directions if ISPs apply ingress filtering and the datagrams contain wrong source addresses [I-D.huitema-multi6-ingress-filtering].

Hosts with simultaneously active interfaces receive multiple prefixes and have multiple source addresses. Datagrams originating from such hosts carry great risks to be dropped due to ingress filtering. Source address selection algorithm needs to be careful to try to avoid ingress filtering on the next-hop router [RFC6724].

Many use cases have been reported for source/destination routing in [I-D.baker-rtgwg-src-dst-routing-use-cases]. These use cases clearly indicate that the multihomed host or Customer Premises Equipment (CPE) router needs to be configured with correct source prefixes/addresses so that it can route packets upstream correctly to avoid ingress filtering applied by an upstream ISP to drop the packets.

In multihomed networks there is a need to do source address based routing if some providers are performing the ingress filtering defined in BCP38 [RFC2827]. This requires the routers to consider the source addresses as well as the destination addresses in determining the next hop to send the packet to.

Based on the use cases defined in [I-D.baker-rtgwg-src-dst-routing-use-cases], the routers may be informed about the source addresses to use in routing using extensions to the routing protocols like IS-IS defined in [ISO.10589.1992] [I-D.baker-ipv6-isis-dst-src-routing] and OSPF defined in [RFC5340] [I-D.baker-ipv6-ospf-dst-src-routing]. In this document we describe the use cases for source address dependent routing from the host perspective.

There are two cases. A host may have a single interface with multiple addresses (from different prefixes or /64s). Each address or prefix is connected to or coming from different exit routers, and this case can be called multi-prefix multihoming (MPMH). A host may have simultaneously connected multiple interfaces where each interface is connected to a different exit router and this case can be called multi-prefix multiple interface (MPMI).

It should be noted that Network Address and Port Translation (NAPT) [RFC3022] in IPv4 and IPv6-to-IPv6 Network Prefix Translation (NPTv6) [RFC6296] in IPv6 implement the functions of source address selection and next-hop resolution and as such they address multihoming (and hosts with multiple interfaces) requirements arising from source address dependent routing [RFC7157]. In this case, the gateway router or CPE router does the source address and next hop selection for all the hosts connected to the router. However, for end-to-end connectivity, NAPT and NPTv6 should be avoided and because of this, NAPT and NPTv6 are left out of scope 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 [RFC2119].

3. SADR Scenarios

Source address dependent routing can be facilitated at the host with proper next hop and source address selection. For this, each router connected to different interfaces of the host uses Router Advertisements to distribute default route, next hop as well as source address/prefix information to the host.

The use case shown in Figure 1 is multi-prefix multi interface use case where rtr1 and rtr2 represent customer premises equipment/routers (CPE) and there are exit routers in both network 1 and network 2. The issue in this case is ingress filtering. If the packets from the host communicating with a remote destination are routed to the wrong exit router, i.e. carry wrong source address, they will get dropped.

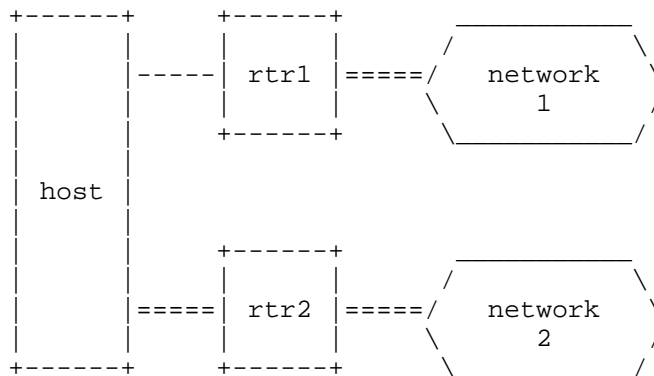


Figure 1: Multihomed Host with Two CPE Routers

Our next use case is shown in Figure 2. This use case is a multi-prefix multihoming use case. rtr is CPE router which is connected to two ISPs each advertising their own prefixes. In this case, the host may have a single interface but it receives multiple prefixes from the connected ISPs. Assuming that ISPs apply ingress filtering policy the packets for any external communication from the host should follow source address dependent routing in order to avoid getting dropped.

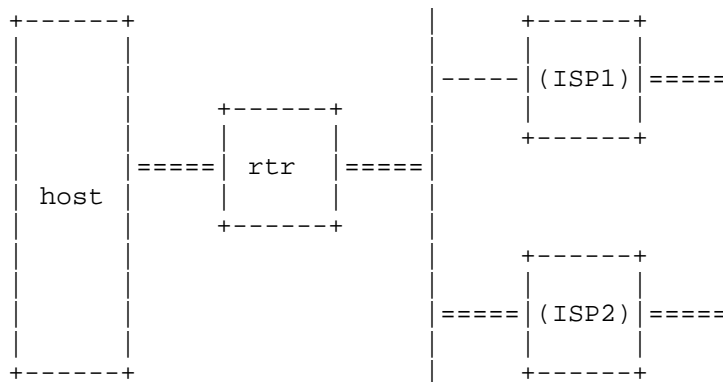


Figure 2: Multihomed Host with Multiple CPE Routers

A variation of this use case is specialized egress routing. Upstream networks offer different services with specific requirements, e.g. video service. The hosts using this service need to use the service's source and destination addresses. No other service will accept this source address, i.e. those packets will be dropped [I-D.baker-rtgwg-src-dst-routing-use-cases].

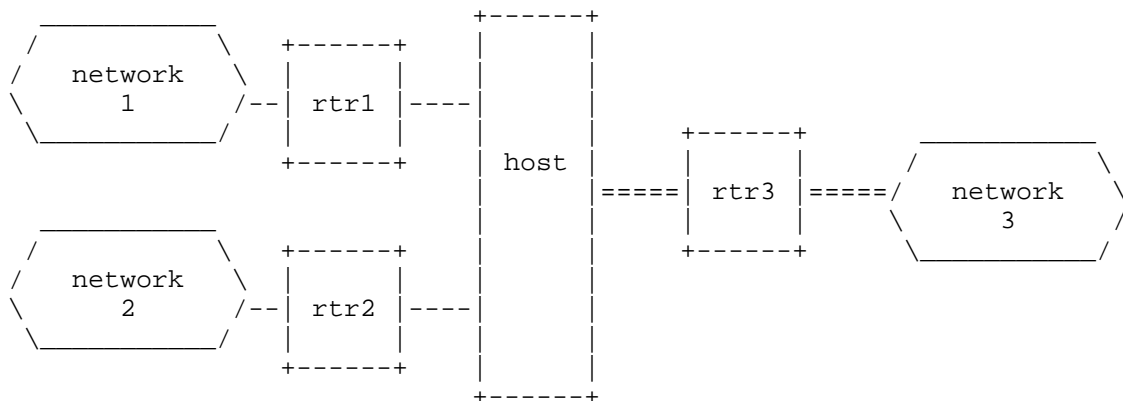


Figure 3: Multihomed Host with Three CPE Routers

Next use case is shown in Figure 3. It is a variation of multi-prefix multi interface use case above. rtr1, rtr2 and rtr3 are CPE Routers. The networks apply ingress routing. Source address dependent routing should be used to avoid any external communications be dropped.

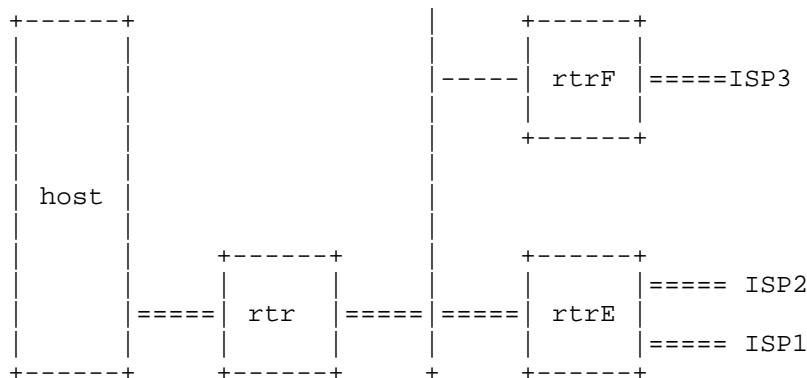


Figure 4: Shim6 Host with Two Routers

The last use case in Figure 4 is also a variation of multi-prefix multihoming use case above. In this case `rtrE` is connected to two ISPs. All ISPs are assumed to apply ingress routing. The host receives prefixes from each ISP and starts communicating with external hosts, e.g. `H1`, `H2`, etc. `H1` and `H2` may be accessible both from `ISP1` and `ISP3`.

The host receives multiple provider-allocated IPv6 address prefixes, e.g. `P1`, `P2` and `P3` for `ISP1`, `ISP2` and `ISP3` and supports shim6 protocol [RFC5533]. `rtr` is a CPE router and the default router for the host. `rtr` receives OSPF routes and has a default route for `rtrE` and `rtrF`.

4. Analysis of Source Address Dependent Routing

In this section we present an analysis of the scenarios of Section 3 and then discuss the relevance of SADR to the provisioning domains.

4.1. Scenarios Analysis

As in [RFC7157] we assume that the routers in Section 3 use Router Advertisements to distribute default route, next hop and source address prefixes supported in each next hop to the hosts or the gateway/CPE router relays this information to the hosts.

Referring to the scenario in Figure 1, source address dependent routing can present a solution to the problem of the host wishes to reach a destination in network 2 and the host may choose `rtr1` as the default router. The solution should start with the correct configuration of the host. The host should be configured with the next hop addresses and the prefixes supported in these next hops. This way the host having received many prefixes will have the correct

knowledge in selecting the right source address and next hop when sending packets to remote destinations.

Note that similar considerations apply to the scenario in Figure 3.

In the configuration of the scenario in Figure 2 also it is useful to configure the host with the next hop addresses and the prefixes and source address prefixes they support. This will enable the host to select the right prefix when sending packets to the right next hop and avoid any ingress filtering.

Source address dependent routing in the use case of specialized egress routing may work as follows. The specialized service router advertizes one or more specific prefixes with appropriate source prefixes, e.g. to the CPE Router, rtr in Figure 2. The CPE router in turn advertizes the specific service's prefixes and source prefixes to the host. This will allow proper configuration at the host so that the host can use the service by sending the packets with the correct source and destination addresses.

Finally, the use case in Figure 4 shows that even though all the routers may have source address dependent routing support, the packets still may get dropped.

The host in Figure 4 starts external communication with H1 and sends the first packet with source address P3::iid. Since rtr has a default route to rtrE it will use this default route in sending the host's packet out towards rtrE. rtrE will route this packet to ISP1 and the packet will be dropped due to the ingress filtering.

A solution to this issue could be that rtrE having multiple routes to H1 could use the path through rtrF and could direct the packet to the other route, i.e. rtrF which would reach H1 in ISP3 without being subject to ingress routing
[I-D.baker-6man-multiprefix-default-route].

4.2. Provisioning Domains and SADR

Consistent set of network configuration information is called provisioning domain (PvD). In case of multi-prefix multihoming (MPMH), more than one provisioning domain is present on a single link. In case of multi-prefix multiple interface (MPMI) environments, elements of the same domain may be present on multiple links. PvD aware nodes support association of configuration information into PvDs and use these PvDs to serve requests for network connections, e.g. choosing the right source address for the packets. PvDs can be constructed from one of more DHCP or Router Advertisement (RA) options carrying such information as PvD identity

and PvD container [I-D.ietf-mif-mpvd-ndp-support], [I-D.ietf-mif-mpvd-dhcp-support]. PvDs constructed based on such information are called explicit PvDs [I-D.ietf-mif-mpvd-arch].

Apart from PvD identity, PvD content may be defined in separate RA or DHCP options. Examples of such content are defined in [I-D.sarikaya-6man-next-hop-ra] and [I-D.sarikaya-dhc-dhcpv6-raoptions-sadr]. They constitute the content or parts of the content of explicit PvD.

Explicit PvDs may be received from different interfaces. Single PvD may be accessible over one interface or simultaneously accessible over multiple interfaces. Explicit PvDs may be scoped to a configuration related to a particular interface, however in general this may not apply. What matters is PvD ID provided that PvD ID is authenticated by the node even in cases where the node has a single connected interface. Single PvD information may be received over multiple interfaces as long as PvD ID is the same. This applies to the router advertisements (RAs) in which case a multi-homed host (that is, with multiple interfaces) should trust a message from a router on one interface to install a route to a different router on another interface.

5. What Needs to be Done

We presented many topologies in which a host with multiple interfaces or a multihomed host is connected to various networks or ISPs which in turn may apply ingress routing. Our scenario analysis showed that in order to avoid packets getting dropped due to ingress routing, source address dependent routing is needed.

One possible solution is the default source address selection Rule 5.5 in [RFC6724] which recommends to select source addresses advertized by the next hop. Source address selection rules can be distributed by DHCP server using DHCP Option OPTION_ADDRSEL_TABLE defined in [RFC7078].

However, it is known that IPv6 implementations are not required to remember which next-hops advertised which prefixes. Also in case of DHCP, DHCP server can configure only the interface of the host to which it is directly connected. In order for it to apply on other interfaces the option has to be sent on those interfaces as well.

There is a need to configure the host not only with the next hops and their prefixes but also with the source prefixes they support. Such a configuration may avoid the host getting ingress/egress policy error messages such as ICMP source address failure message.

If host configuration is done using router advertisement messages then there is a need to define new router advertisement options for source address dependent routing. These options include Next Hop Address with Route Prefix option and Next Hop Address with Source Address and Route Prefix option.

If host configuration is done using DHCP then there is a need to define new DHCP options for source address dependent routing. As mentioned above, DHCP server configuration is interface specific. New DHCP options for source address dependent routing such as route prefix, next hop address and source prefix need to be configured for each interface separately.

6. Security Considerations

This document describes some use cases and thus brings no new security risks to the Internet.

7. IANA Considerations

None.

8. Acknowledgements

In writing this document, the author benefited from face to face discussions he had with Brian Carpenter and Ole Troan.

9. References

9.1. Normative References

- [ISO.10589.1992] International Organization for Standardization, "Intermediate system to intermediate system intra-domain-routing routine information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473), ISO Standard 10589", ISO ISO.10589.1992, 1992.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, January 2001.
- [RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, March 2004.
- [RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.
- [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, November 2005.
- [RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, August 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.
- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, July 2008.
- [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, June 2009.
- [RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.
- [RFC6724] Thaler, D., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, September 2012.
- [RFC7078] Matsumoto, A., Fujisaki, T., and T. Chown, "Distributing Address Selection Policy Using DHCPv6", RFC 7078, January 2014.
- [RFC7157] Troan, O., Miles, D., Matsushima, S., Okimoto, T., and D. Wing, "IPv6 Multihoming without Network Address Translation", RFC 7157, March 2014.

9.2. Informative References

- [I-D.baker-6man-multiprefix-default-route]
Baker, F., "Multiprefix IPv6 Routing for Ingress Filters", draft-baker-6man-multiprefix-default-route-00 (work in progress), November 2007.
- [I-D.baker-ipv6-isis-dst-src-routing]
Baker, F., "IPv6 Source/Destination Routing using IS-IS", draft-baker-ipv6-isis-dst-src-routing-01 (work in progress), August 2013.
- [I-D.baker-ipv6-ospf-dst-src-routing]
Baker, F., "IPv6 Source/Destination Routing using OSPFv3", draft-baker-ipv6-ospf-dst-src-routing-03 (work in progress), August 2013.
- [I-D.baker-rtgwg-src-dst-routing-use-cases]
Baker, F., "Requirements and Use Cases for Source/Destination Routing", draft-baker-rtgwg-src-dst-routing-use-cases-00 (work in progress), August 2013.
- [I-D.huitema-multi6-ingress-filtering]
Huitema, C., "Ingress filtering compatibility for IPv6 multihomed sites", draft-huitema-multi6-ingress-filtering-00 (work in progress), October 2004.
- [I-D.ietf-mif-mpvd-arch]
Anipko, D., "Multiple Provisioning Domain Architecture", draft-ietf-mif-mpvd-arch-07 (work in progress), October 2014.
- [I-D.ietf-mif-mpvd-dhcp-support]
Krishnan, S., Korhonen, J., and S. Bhandari, "Support for multiple provisioning domains in DHCPv6", draft-ietf-mif-mpvd-dhcp-support-00 (work in progress), August 2014.
- [I-D.ietf-mif-mpvd-ndp-support]
Korhonen, J., Krishnan, S., and S. Gundavelli, "Support for multiple provisioning domains in IPv6 Neighbor Discovery Protocol", draft-ietf-mif-mpvd-ndp-support-00 (work in progress), August 2014.
- [I-D.sarikaya-6man-next-hop-ra]
Sarikaya, B., "IPv6 RA Options for Next Hop Routes", draft-sarikaya-6man-next-hop-ra-02 (work in progress), June 2014.

[I-D.sarikaya-dhc-dhcpv6-raoptions-sadr]

Sarikaya, B., "DHCPv6 Route Options for Source Address
Dependent Routing", draft-sarikaya-dhc-dhcpv6-raoptions-
sadr-00 (work in progress), June 2014.

[RFC6106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli,
"IPv6 Router Advertisement Options for DNS Configuration",
RFC 6106, November 2010.

Author's Address

Behcet Sarikaya
Huawei USA
5340 Legacy Dr. Building 175
Plano, TX 75024

Email: sarikaya@ieee.org

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 1, 2017

B. Sarikaya
Huawei USA
M. Boucadair
Orange
September 28, 2016

Source Address Dependent Routing and Source Address Selection for IPv6
Hosts: Problem Space Overview
draft-sarikaya-6man-sadr-overview-12

Abstract

This document presents the source address dependent routing (SADR) problem space from the host perspective. Both multihomed hosts and hosts with multiple interfaces are considered. Several network architectures are presented to illustrate why source address selection and next hop resolution in view of source address dependent routing is needed.

The document is scoped on identifying a set of scenarios for source address dependent routing from the host perspective and analyze a set of solutions to mitigate encountered issues. The document does not make any solution recommendations.

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 April 1, 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
1.1. Overall Context	2
1.2. Scope	3
2. Source Address Dependent Routing (SADR) Scenarios	4
2.1. Multi-Prefix Multihoming	4
2.2. Multi-Prefix Multi-Interface	5
2.3. Home Network (Homenet)	6
2.4. Service-specific Egress Routing	7
3. Analysis of Source Address Dependent Routing	8
3.1. Scenarios Analysis	8
3.2. Provisioning Domains and SADR	10
4. Discussion on Alternate Solutions	11
4.1. Router Advertisement Option	11
4.2. Router Advertisement Option Set	12
4.3. Source Address Selection Rule 5.5	12
5. Security Considerations	13
6. Acknowledgements	13
7. References	13
7.1. Normative References	13
7.2. Informative References	14
Authors' Addresses	16

1. Introduction

1.1. Overall Context

BCP 38 recommends ingress traffic routing to prohibit Denial-of-Service (DoS) attacks. As such, datagrams which have source addresses that do not match with the network where the host is attached are discarded [RFC2827]. Avoiding packets to be dropped because of ingress filtering is difficult especially in multihomed networks where the host receives more than one prefix from the networks it is connected to, and consequently may have more than one source addresses. Based on BCP 38, BCP 84 introduced recommendations on the routing system for multihomed networks [RFC3704].

Recommendations on the routing system for ingress filtering such as in BCP 84 inevitably involve source address checks. This leads to the source address dependent routing (SADR). Source address dependent routing is an issue especially when the host is connected to a multihomed network and is communicating with another host in another multihomed network. In such a case, the communication can be broken in both directions if Network Providers apply ingress filtering and the datagrams contain wrong source addresses (see for more details [I-D.huitema-multi6-ingress-filtering]).

Hosts with simultaneously active interfaces receive multiple prefixes and have multiple source addresses. Datagrams originating from such hosts are likely to be dropped due to ingress filtering policies. Source address selection algorithm needs to be careful to try to avoid ingress filtering on the next-hop router [RFC6724].

Many use cases have been reported for source/destination routing, for example [I-D.baker-rtgwg-src-dst-routing-use-cases]. These use cases clearly indicate that the multihomed host or Customer Premises Equipment (CPE) router needs to be configured with correct source prefixes/addresses so that it can forward packets upstream correctly to avoid ingress filtering applied by an upstream Network Provider to drop the packets.

In multihomed networks there is a need to enforce source address based routing if some providers are performing the ingress filtering. This requires the routers to consider the source addresses as well as the destination addresses in determining the next hop to send the packet to.

1.2. Scope

Based on the use cases defined in [I-D.baker-rtgwg-src-dst-routing-use-cases], the routers may be informed about the source addresses to use for forwarding using extensions to the routing protocols like IS-IS [ISO.10589.1992] [I-D.baker-ipv6-isis-dst-src-routing] or OSPF [RFC5340] [I-D.baker-ipv6-ospf-dst-src-routing].

In this document, we describe the scenarios for source address dependent routing from the host perspective. Two flavors can be considered:

1. A host may have a single interface with multiple addresses (from different prefixes or /64s). Each prefix is delegated from different exit routers, and this case can be called multi-prefix multihoming (MPMH). In such case, source address selection is

performed by the host while source-depending routing is to be enforced by an upstream router.

2. A host may have simultaneously connected multiple interfaces where each interface is connected to a different exit router and this case can be called multi-prefix multiple interface (MPMI). For this case, the host requires to support both source address selection and source-depending routing to avoid the need to rewrite the IPv6 prefix by an upstream router.

Several limitations arise in such NAT- and NPTv6-based ([RFC6296]) multihoming contexts (see for example [RFC4116]). NPTv6 is left out of scope of this document.

This document was initially written to inform the community about the SADR problem space. It was updated to record the various set of alternate solutions to address that problem space. The 6man consensus is documented in [I-D.ietf-6man-multi-homed-host].

2. Source Address Dependent Routing (SADR) Scenarios

This section describes a set of scenarios to illustrate the SADR problem. Scenarios are listed following a complexity order.

2.1. Multi-Prefix Multihoming

The scenario shown in Figure 1 is a multi-prefix multihoming use case. "rtr" is a CPE router which is connected to two Network Providers, each advertising their own prefixes. In this case, the host may have a single interface but it receives multiple prefixes from the upstream Network Providers. Assuming that providers apply ingress filtering policy the packets for any external communication from the host should follow source address dependent routing in order to avoid getting dropped.

In this scenario, the host does not need to perform source-depending routing; it does only need to perform source address selection.

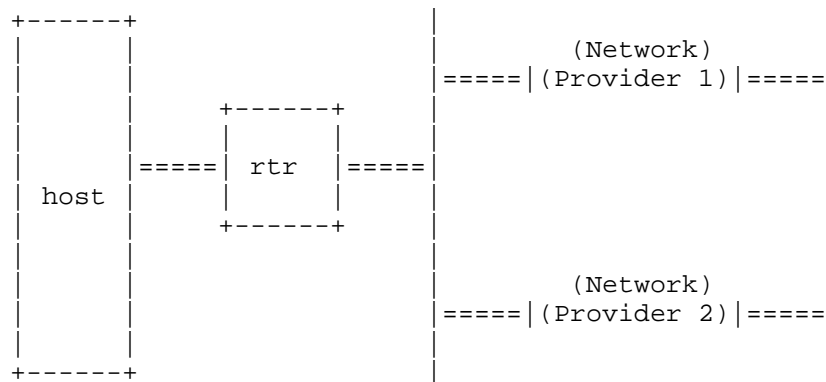


Figure 1: Multihomed Host with Multiple CPE Routers

2.2. Multi-Prefix Multi-Interface

The scenario shown in Figure 2 is multi-prefix multi interface, where "rtr1" and "rtr2" represent CPE routers and there are exit routers in both "network 1" and "network 2". If the packets from the host communicating with a remote destination are routed to the wrong exit router, i.e., carry wrong source address, they will get dropped due to ingress filtering.

In order to avoid complications to send packets and avoid a need to rewrite the source IPv6 prefix, the host requires to perform both source address selection and source-depending routing so that appropriate next-hop is selected taking into account the source address.

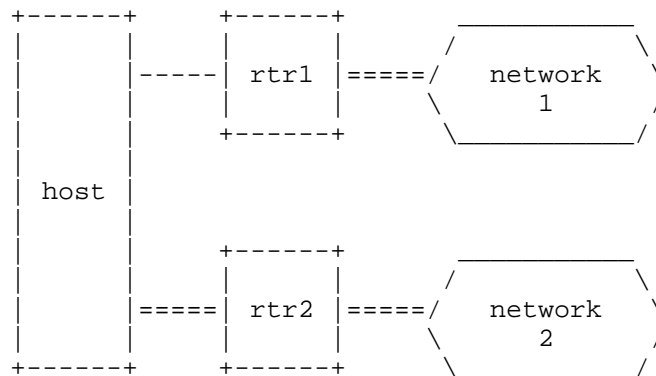


Figure 2: Multiple Interfaced Host with Two CPE Routers

There is a variant of Figure 2 that is often referred to as a corporate VPN, i.e., a secure tunnel from the host to a router attached to a corporate network. In this case "rtr2" gives access directly to the corporate network, and the link from the host to "rtr2" is a secure tunnel (for example an IPsec tunnel). The interface is therefore a virtual interface, with its own IP address/prefix assigned by the corporate network.

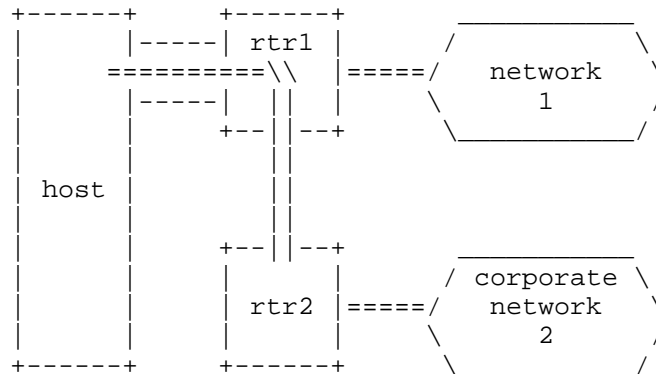


Figure 3: VPN case

There are at least two sub-cases:

- a. Dedicated forwarding entries are created in the host such that only traffic directed to the corporate network is sent to "rtr2"; everything else is sent to "rtr1".
- b. All traffic is sent to "rtr2" and then routed to the Internet if necessary. This case doesn't need host routes but leads to unnecessary traffic and latency because of the path stretch via rtr2.

2.3. Home Network (Homenet)

In the homenet scenario depicted in Figure 4, representing a simple home network, there is a host connected to a local network that is serviced with two CPEs which are connected to providers 1 and 2, respectively. Each network delegates a different prefix. Also each router provides a different prefix to the host. The issue in this scenario is also ingress filtering used by each provider. This scenario can be considered as a variation of the scenario described in Section 2.2.

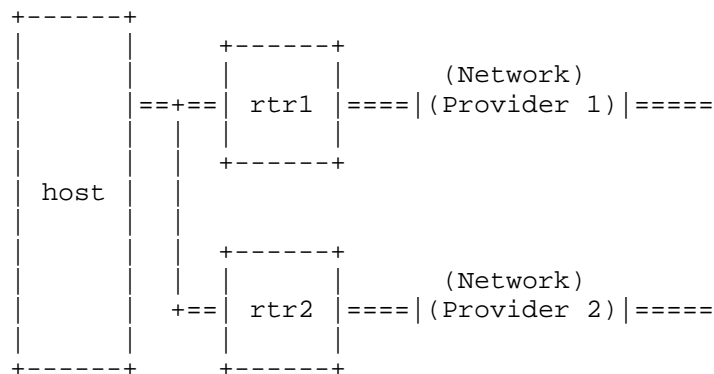


Figure 4: Simple Home Network with Two CPE Routers

The host has to select the source address from the prefixes of Providers 1 or 2 when communicating with other hosts in Provider 1 or 2. The next issue is to select the correct next hop router, rtr1 or rtr2 that can reach the correct provider, "Network Provider 1" or "Network Provider 2".

2.4. Service-specific Egress Routing

A variation of the scenario in Section 2.1 is: specialized egress routing. Upstream networks offer different services with specific requirements, e.g., VoIP or IPTV. The hosts using this service need to use the service's source and destination addresses. No other service will accept this source address, i.e., those packets will be dropped [I-D.baker-rtgwg-src-dst-routing-use-cases].

Both source address selection and source-dependent routing are required to be performed by the host.

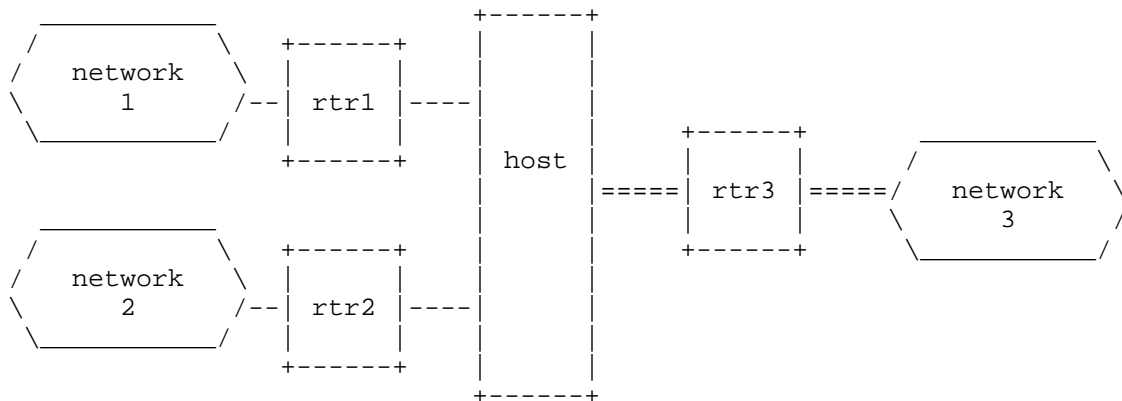


Figure 5: Multiple Interfaced Host with Three CPE Routers

The scenario shown in Figure 5 is a variation of multi-prefix multi interface scenario (Section 2.2). "rtr1", "rtr2" and "rtr3" are CPE routers. The networks apply ingress routing. Source address dependent routing should be used to avoid any external communications be dropped.

3. Analysis of Source Address Dependent Routing

SADR can be facilitated at the host with proper source address and next-hop selection. For this, each router connected to different interfaces of the host uses Router Advertisements (RAs, [RFC4861]) to distribute a default route, next hop as well as source address/prefix information to the host. As a reminder, while Prefix Information Option (PIO) is defined in [RFC4861], Route Information Option (RIO) is defined in [RFC4191].

Section 3.1 presents an analysis of the scenarios of Section 2 and then Section 3.2 discusses the relevance of SADR to the provisioning domains.

3.1. Scenarios Analysis

As in [RFC7157] we assume that the routers in Section 2 use Router Advertisements (RAs) to distribute default route and source address prefixes supported in each next hop to the hosts or the gateway/CPE router relays this information to the hosts.

Referring to Section 2.1, source address selection is undertaken by the host while source-dependent routing must be followed by "rtr" to avoid packets drop. No particular modification is required for next-hop selection at the host.

Referring to the scenario in Figure 2, source address dependent routing can present a solution to the problem of the host wishes to reach a destination in network 2 and the host may choose rtr1 as the default router. The solution assumes the host is correctly configured. The host should be configured with the prefixes supported in these next hops. This way the host having received many prefixes will have the correct knowledge in selecting the right source address and next hop when sending packets to remote destinations.

Note that similar considerations apply to the scenario in Figure 5.

In the configuration of the scenario (Figure 1) it is also useful to configure the host with the prefixes and source address prefixes they support. This will enable the host to select the right prefix when sending packets to the right next hop and avoid any ingress filtering issues.

Let us analyze the scenario in Section 2.3. If a source address dependent routing protocol is used, the two routers (rtr1 and rtr2) are both able to route traffic correctly, no matter which next-hop router and source address the host selects. In case the host chooses the wrong next hop router, e.g., for provider 2 rtr1 is selected, rtr1 will forward the traffic to rtr2 to be sent to network provider 2 and no ingress filtering will happen.

Note that home networks are expected to comply with requirements for source address dependent routing and the routers will be configured accordingly, no matter which routing protocol is used [RFC7788].

This would work but with issues. The host traffic to provider 2 will have to go over two links instead of one, i.e., the link bandwidth will be halved. Another possibility is rtr1 can send an ICMPv6 Redirect message to the host to direct the traffic to rtr2. Host would redirect provider 2 traffic to rtr2.

The problem with redirects is that ICMPv6 Redirect message can only convey two addresses, i.e., in this case the router address, or rtr2 address and the destination address, or the destination host in provider 2. That means the source address will not be communicated. As a result, the host would send packets to the same destination using both source addresses which causes rtr2 to send a redirect message to rtr1, resulting in ping-pong redirects sent by rtr1 and rtr2.

A solution to these issues is to configure the host with the source address prefixes that the next hop supports. In a homenet context, each interface of the host can be configured by its next hop router,

so that all that is needed is to add the information on source address prefixes. This results in the hosts to select the right router no matter what.

Source address dependent routing in the use case of specialized egress routing (Section 2.4) may work as follows. The specialized service router advertises one or more specific prefixes with appropriate source prefixes, e.g., to the CPE router, rtr in Figure 1. The CPE router in turn advertises the specific service's prefixes and source prefixes to the host. This will allow proper configuration at the host so that the host can use the service by sending the packets with the correct source and destination addresses.

3.2. Provisioning Domains and SADR

Consistent set of network configuration information is called provisioning domain (PvD). In case of multi-prefix multihoming (MPMH), more than one provisioning domain is present on a single link. In case of multi-prefix multiple interface (MPMI) environments, elements of the same domain may be present on multiple links. PvD aware nodes support association of configuration information into PvDs and use these PvDs to serve requests for network connections, e.g., choosing the right source address for the packets. PvDs can be constructed from one of more DHCP or Router Advertisement (RA) options carrying such information as PvD identity and PvD container [I-D.ietf-mif-mpvd-ndp-support], [I-D.ietf-mif-mpvd-dhcp-support]. PvDs constructed based on such information are called explicit PvDs [RFC7556].

Apart from PvD identity, PvD content may be encapsulated in separate RA or DHCP options called PvD Container Option. These options are placed in the container options of an explicit PvD.

Explicit PvDs may be received from different interfaces. Single PvD may be accessible over one interface or simultaneously accessible over multiple interfaces. Explicit PvDs may be scoped to a configuration related to a particular interface, however in general this may not apply. What matters is PvD ID provided that PvD ID is authenticated by the node even in cases where the node has a single connected interface. The authentication of the PvD ID should meet the level required by the node policy. Single PvD information may be received over multiple interfaces as long as PvD ID is the same. This applies to the router advertisements (RAs) in which case a multi-homed host (that is, with multiple interfaces) should trust a message from a router on one interface to install a route to a different router on another interface.

4. Discussion on Alternate Solutions

We presented many topologies in which a host with multiple interfaces or a multihomed host is connected to various networks or Network Providers which in turn may apply ingress routing. The scenario analysis in Section 3.1 shows that in order to avoid packets getting dropped due to ingress routing, source address dependent routing is needed. Also, source address dependent routing should be supported by routers throughout a site that has multiple egress points.

In this section, we provide some alternate solutions vis a vis the scenarios presented in Section 2. We start with source address selection rule 5.5 ([RFC6724]) and the scenarios it solves and continue with solutions that state exactly what information hosts need in terms of new router advertisement options for correct source address selection in those scenarios. No recommendation is made in this section.

4.1. Router Advertisement Option

There is a need to configure the host not only with the prefixes but also with the source prefixes the next hop routers support. Such a configuration may avoid the host getting ingress/egress policy error messages such as ICMP source address failure message.

If host configuration is done using router advertisement messages then there is a need to define new router advertisement options for source address dependent routing. These options include Route Prefix with Source Address/Prefix Option. Other options such as Next Hop Address with Route Prefix option and Next Hop Address with Source Address and Route Prefix option will be considered in Section 4.2.

As discussed in Section 3.1, the scenario in Figure 4 can be solved by defining a new router advertisement option.

If host configuration is done using DHCP then there is a need to define new DHCP options for Route Prefix with Source Address/Prefix. As mentioned above, DHCP server configuration is interface specific. New DHCP options for source address dependent routing such as route prefix and source prefix need to be configured for each interface separately.

The scenario in Figure 4 can be solved by defining a new DHCP option.

4.2. Router Advertisement Option Set

The source address selection rule 5.5 may possibly be a solution for selecting the right source addresses for each next hop but there are cases where the next hop routers on each interface of the host are not known by the host initially. Such use cases are out of scope. Guidelines for use cases that require router advertisement option set involving third party next hop addresses are also out of scope.

4.3. Source Address Selection Rule 5.5

One possible solution is the default source address selection Rule 5.5 in [RFC6724] which recommends to select source addresses advertised by the next hop. Considering the above scenarios, we can state that this rule can solve the problem in Figure 2, Figure 1 and Figure 5.

Source address selection rules can be distributed by DHCP server using DHCP Option `OPTION_ADDRSEL_TABLE` defined in [RFC7078].

In case of DHCP based host configuration, DHCP server can configure only the interface of the host to which it is directly connected. In order for Rule 5.5 to apply on other interfaces the option should be sent on those interfaces as well using [RFC7078].

The default source address selection Rule 5.5 solves that problem when an application sends a packet with an unspecified source address. In the presence of two default routes, one route will be chosen, and Rule 5.5 will make sure the right source address is used.

When the application selects a source address, i.e., the source address is chosen before next-hop selection, even though the source address is a way for the application to select the exit point, in this case that purpose will not be served. In the presence of multiple default routes, one will be picked, ignoring the source address which was selected by the application because it is known that IPv6 implementations are not required to remember which next-hops advertised which prefixes. Therefore, the next-hop router may not be the correct one, and the packets may be filtered.

This implies that the hosts should register which next-hop router announced each prefix. It is required that RAs be sent by the routers and that they contain PIO on all links. It is also required that the hosts remember the source addresses of the routers that sent PIOs together with the prefixes advertised. This can be achieved by updating redirect rules specified in [RFC4861].

[I-D.ietf-6man-multi-homed-host] further elaborates this to specify to which router a host should present its transmission.

Source address dependent routing solution is not complete without support from the edge routers. All routers in edge networks need to be required to support routing based on not only the destination address but also the source address. All edge routers need to be required to satisfy BCP 38 filters.

5. Security Considerations

This document describes some use cases and thus brings no additional security risks. Solution documents should further elaborate on specific security considerations.

6. Acknowledgements

In writing this document, we benefited from the ideas expressed by the electronic mail discussion participants on 6man Working Group: Brian Carpenter, Ole Troan, Pierre Pfister, Alex Petrescu, Ray Hunter, Lorenzo Colitti and others.

Pierre Pfister proposed the scenario in Figure 4 as well as some text for Rule 5.5.

The text on corporate VPN in Section 3 was provided by Brian Carpenter.

7. References

7.1. Normative References

- [I-D.ietf-6man-multi-homed-host]
Marcon, J. and B. Carpenter, "First-hop router selection by hosts in a multi-prefix network", draft-ietf-6man-multi-homed-host-09 (work in progress), August 2016.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <<http://www.rfc-editor.org/info/rfc2827>>.
- [RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <<http://www.rfc-editor.org/info/rfc3704>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<http://www.rfc-editor.org/info/rfc4861>>.

- [RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", RFC 5340, DOI 10.17487/RFC5340, July 2008, <<http://www.rfc-editor.org/info/rfc5340>>.
- [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, DOI 10.17487/RFC5533, June 2009, <<http://www.rfc-editor.org/info/rfc5533>>.
- [RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, DOI 10.17487/RFC6296, June 2011, <<http://www.rfc-editor.org/info/rfc6296>>.
- [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>>.
- [RFC7078] Matsumoto, A., Fujisaki, T., and T. Chown, "Distributing Address Selection Policy Using DHCPv6", RFC 7078, DOI 10.17487/RFC7078, January 2014, <<http://www.rfc-editor.org/info/rfc7078>>.

7.2. Informative References

- [I-D.baker-6man-multiprefix-default-route]
Baker, F., "Multiprefix IPv6 Routing for Ingress Filters", draft-baker-6man-multiprefix-default-route-00 (work in progress), November 2007.
- [I-D.baker-ipv6-isis-dst-src-routing]
Baker, F. and D. Lamparter, "IPv6 Source/Destination Routing using IS-IS", draft-baker-ipv6-isis-dst-src-routing-05 (work in progress), April 2016.
- [I-D.baker-ipv6-ospf-dst-src-routing]
Baker, F., "IPv6 Source/Destination Routing using OSPFv3", draft-baker-ipv6-ospf-dst-src-routing-03 (work in progress), August 2013.
- [I-D.baker-rtwgw-src-dst-routing-use-cases]
Baker, F., Xu, M., Yang, S., and J. Wu, "Requirements and Use Cases for Source/Destination Routing", draft-baker-rtwgw-src-dst-routing-use-cases-02 (work in progress), April 2016.

- [I-D.huitema-multi6-ingress-filtering]
Huitema, C., "Ingress filtering compatibility for IPv6 multihomed sites", draft-huitema-multi6-ingress-filtering-00 (work in progress), October 2004.
- [I-D.ietf-mif-mpvd-dhcp-support]
Krishnan, S., Korhonen, J., and S. Bhandari, "Support for multiple provisioning domains in DHCPv6", draft-ietf-mif-mpvd-dhcp-support-02 (work in progress), October 2015.
- [I-D.ietf-mif-mpvd-ndp-support]
Korhonen, J., Krishnan, S., and S. Gundavelli, "Support for multiple provisioning domains in IPv6 Neighbor Discovery Protocol", draft-ietf-mif-mpvd-ndp-support-03 (work in progress), February 2016.
- [I-D.naderi-ipv6-probing]
Naderi, H. and B. Carpenter, "Experience with IPv6 path probing", draft-naderi-ipv6-probing-01 (work in progress), April 2015.
- [ISO.10589.1992]
International Organization for Standardization, "Intermediate system to intermediate system intra-domain-routing routine information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473), ISO Standard 10589", ISO ISO.10589.1992, 1992.
- [RFC4116] Abley, J., Lindqvist, K., Davies, E., Black, B., and V. Gill, "IPv4 Multihoming Practices and Limitations", RFC 4116, DOI 10.17487/RFC4116, July 2005, <<http://www.rfc-editor.org/info/rfc4116>>.
- [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <<http://www.rfc-editor.org/info/rfc4191>>.
- [RFC7157] Troan, O., Ed., Miles, D., Matsushima, S., Okimoto, T., and D. Wing, "IPv6 Multihoming without Network Address Translation", RFC 7157, DOI 10.17487/RFC7157, March 2014, <<http://www.rfc-editor.org/info/rfc7157>>.
- [RFC7556] Anipko, D., Ed., "Multiple Provisioning Domain Architecture", RFC 7556, DOI 10.17487/RFC7556, June 2015, <<http://www.rfc-editor.org/info/rfc7556>>.

[RFC7788] Stenberg, M., Barth, S., and P. Pfister, "Home Networking Control Protocol", RFC 7788, DOI 10.17487/RFC7788, April 2016, <<http://www.rfc-editor.org/info/rfc7788>>.

Authors' Addresses

Behcet Sarikaya
Huawei USA
5340 Legacy Dr. Building 175
Plano, TX 75024

Email: sarikaya@ieee.org

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 22, 2014

B. Sarikaya
Huawei USA
June 20, 2014

DHCPv6 Route Options for Source Address Dependent Routing
draft-sarikaya-dhc-dhcpv6-raoptions-sadr-00

Abstract

This document describes DHCPv6 Route Options for provisioning IPv6 routes on DHCPv6 client nodes for source address dependent routing. Using these options, an operator can configure multi-homed nodes where other means of route configuration may be impractical.

Requirements Language

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

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 December 22, 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. DHCPv6 Based Solution	3
2.1. Default route configuration	3
2.2. Configuring on-link routes	4
2.3. Deleting obsolete route	4
2.4. Applicability to routers	5
2.5. Updating Routing Information	5
2.6. Limitations	6
3. DHCPv6 Route Options	6
3.1. Route Prefix Option Format	7
3.2. Next Hop Option Format	8
3.3. Source Address/Prefix Option Format	9
4. DHCPv6 Server Behavior	10
5. DHCPv6 Client Behavior	11
5.1. Conflict resolution	12
6. IANA Considerations	13
7. Security Considerations	13
8. Acknowledgements	14
9. References	14
9.1. Normative References	14
9.2. Informative References	14
Author's Address	15

1. Introduction

The Neighbor Discovery (ND) protocol [RFC4861] provides a mechanism for hosts to discover one or more default routers on a directly connected network segment. Extensions to the Router Advertisement (RA) protocol defined in [RFC4191] allow hosts to discover the preferences for multiple default routers on a given link, as well as any specific routes advertised by these routers. This provides network administrators with a new set of tools to handle multi-homed host topologies and influence the route selection by the host. This ND based mechanism however is sub optimal or impractical in some multi-homing scenarios, e.g. source address dependent routing. Both Router Advertisement options [I-D.sarikaya-6man-next-hop-ra] and DHCPv6 can be used. In networks that deployed DHCPv6, the use of DHCPv6 [RFC3315] is seen to be more viable.

DHCPv6 Route Options defined in this document can be used to configure fixed and mobile nodes in multi-homed scenarios with route information and next hop address. Different scenarios exist such as the node is simultaneously connected to multiple access network of e.g. WiFi and 3G. The node may also be connected to more than one gateway. Such connectivity may be realized by means of dedicated physical or logical links that may also be shared with other users nodes such as in residential access networks.

A document defining topologies and in general providing an overview of the issue of source address dependent routing is TBD.

The solution presented in this document is part of the network configuration information. A consistent set of network configuration is defined as Provisioning Domain (PvD) [I-D.ietf-mif-mpvd-arch]. PvDs or so-called explicit PvDs may include information related to more than one interfaces as is the case in this document. It is important to note that the node has a trust relationship with the PvD, in such a case, it is called trusted PvD. The trust is established using authorization and authentication between the node that is using the PvD configuration and the source that provided that configuration. In this document, we assume that DHCP server can provide trusted PvDs to the hosts.

2. DHCPv6 Based Solution

A DHCPv6 based solution allows an operator an on demand and node specific means of configuring static routing information. Such a solution also fits into network environments where the operator prefers to manage Residential Gateway (RG) configuration information from a centralized DHCP server. [RFC7157] provides additional background to the need for a DHCPv6 solution to the problem.

In terms of the high level operation of the solution defined in this draft, a DHCPv6 client interested in obtaining routing information requests the route options using the DHCPv6 Option Request Option (ORO) sent to a server. A Server, when configured to do so, provides the requested route information as part of a nested options structure covering; the next-hop address; the destination prefix; the route metric; any additional options applicable to the destination or next-hop.

2.1. Default route configuration

A non-trustworthy network may be available at the same time as a trustworthy network, with the risk of bad consequences if the host gets confused between the two. These are basically the two models for hosts with multiple interfaces, both of which are valid, but

which are incompatible with each other. In the first model, an interface is connected to something like a corporate network, over a Virtual Private Network (VPN). This connection is trusted because it has been authenticated. Routes obtained over such a connection can probably be trusted, and indeed it may be important to use those routes. This is because in the VPN case, you may also be connected to a network that's offered you a default route, and you could be attacked over that connection if you attempt to connect to resources on the enterprise network over it.

On the other, non-trustworthy network scenario, none of the networks to which the host is connected are meaningfully more or less trustworthy. In this scenario, the untrustworthy network may hand out routes to other hosts, e.g. those in the VPN going through some malicious nodes. This will have bad consequences because the host's traffic intended for the corporate VPN may be hijacked by the intermediate nodes.

DHCPv6 options described in this document can be used to install the routes. However, the use of such a technique makes sense only in the former case above, i.e. trusted network. So the host **MUST** have an authenticated connection to the network it connects so that DHCPv6 route options can be trusted before establishing routes.

Server **MUST NOT** define more than one default route.

2.2. Configuring on-link routes

Server may also configure on-link routes, i.e. routes that are available directly over the link, not via routers. To specify on-link routes, server **MAY** include RTPREFIX option directly in Advertise and Reply messages.

2.3. Deleting obsolete route

There are two mechanisms that allow removing a route. Each defined route has a route lifetime. If specific route is not refreshed and its timer reaches 0, client **MUST** remove corresponding entry from routing table.

In cases, where faster route removal is needed, server **SHOULD** return RT_PREFIX option with route lifetime set to 0. Client that receives RT_PREFIX with route lifetime set to 0 **MUST** remove specified route immediately, even if its previous lifetime did not expire yet.

2.4. Applicability to routers

Contrary to Router Advertisement mechanism, defined in [RFC4861] that explicitly limits configuration to hosts, routing configuration over DHCPv6 defined in this document may be used by both hosts and routers. (This limitation of RA mechanism was partially lifted by W-1 requirement formulated in [RFC6204].)

One of the envisaged usages for this solution are residential gateways (RG) or Customer Premises Equipment (CPE). Those devices very often perform routing. It may be useful to configure routing on such devices over DHCPv6. One example of such use may be a class of premium users that are allowed to use dedicated router that is not available to regular users.

2.5. Updating Routing Information

Network configuration occasionally changes, due to failure of existing hardware, migration to newer equipment or many other reasons. Therefore there a way to inform clients that routing information have changed is required.

There are several ways to inform clients about new routing information. Every client SHOULD periodically refresh its configuration, according to Information Refresh Time Option, so server may send updated information the next time client refreshes its information. New routes may be configured at that time. As every route has associated lifetime, client is required to remove its routes when this timer expires. This method is particularly useful, when migrating to new router is undergoing, but old router is still available.

Server MAY also announce routes via soon to be removed router with lifetimes set to 0. This will cause the client to remove its routes, despite the fact that previously received lifetime may not yet expire.

Aforementioned methods are useful, when there is no urgent need to update routing information. Bound by timer set by value of Information Refresh Time Option, clients may use outdated routing information until next scheduled renewal. Depending on configured value this delay may be not acceptable in some cases. In such scenarios, administrators are advised to use RECONFIGURE mechanism, defined in [RFC3315]. Server transmits RECONFIGURE message to each client, thus forcing it to immediately start renewal process.

See also Section 2.6 about limitations regarding dynamic routing.

2.6. Limitations

Defined mechanism is not intended to be used as a dynamic routing protocol. It should be noted that proposed mechanism cannot automatically detect routing changes. In networks that use dynamic routing and also employ this mechanism, clients may attempt using routes configured over DHCPv6 even though routers or specific routes ceased to be available. This may cause black hole routing problem. Therefore it is not recommended to use this mechanism in networks that use dynamic routing protocols. This mechanism **SHOULD NOT** be used in such networks, unless network operator can provide a way to update DHCP server information in case of router availability changes.

Discussion: It should be noted that DHCPv6 server is not able to monitor health of existing routers. As there are currently more than 60 options defined for DHCPv6, it is infeasible to implement mechanism that would monitor huge set of services and stop announcing its availability in case of service outage. Therefore in case of prolonged unavailability human intervention is required to change DHCPv6 server configuration. If that is considered a problem, network administrators should consider using other alternatives, like RA and ND mechanisms (see [RFC4861]).

3. DHCPv6 Route Options

A DHCPv6 client interested in obtaining routing information includes the NEXT_HOP and RT_PREFIX options as part of its Option Request Option (ORO) in messages directed to a server (as allowed by [RFC3315], i.e. Solicit, Request, Renew, Rebind or Information-request messages). A Server, when configured to do so, provides the requested route information using zero, one or more NEXT_HOP options in messages sent in response (Advertise, and Reply). So as to allow the route options to be both extensible, as well as conveying detailed info for routes, use is made of a nested options structure. Server sends one or more NEXT_HOP options that specify the IPv6 next hop addresses. Each NEXT_HOP option conveys in turn zero, one or more RT_PREFIX options that represents the IPv6 destination prefixes reachable via the given next hop. Server includes RT_PREFIX directly in message to indicate that given prefix is available directly on-link. Server MAY send a single NEXT_HOP without any RT_PREFIX suboptions or with RT_PREFIX that contains ::/0 to indicate available default route. The Formats of the NEXT_HOP and RT_PREFIX options are defined in the following sub-sections.

The DHCPv6 Route Options format borrows from the principles of the Route Information Option defined in [RFC4191].

3.1. Route Prefix Option Format

The Route Prefix Option is used to convey information about a single prefix that represents the destination network. The Route Prefix Option is used as a sub-option in the previously defined Next Hop Option. It may also be sent directly in message to indicate that route is available directly on-link.

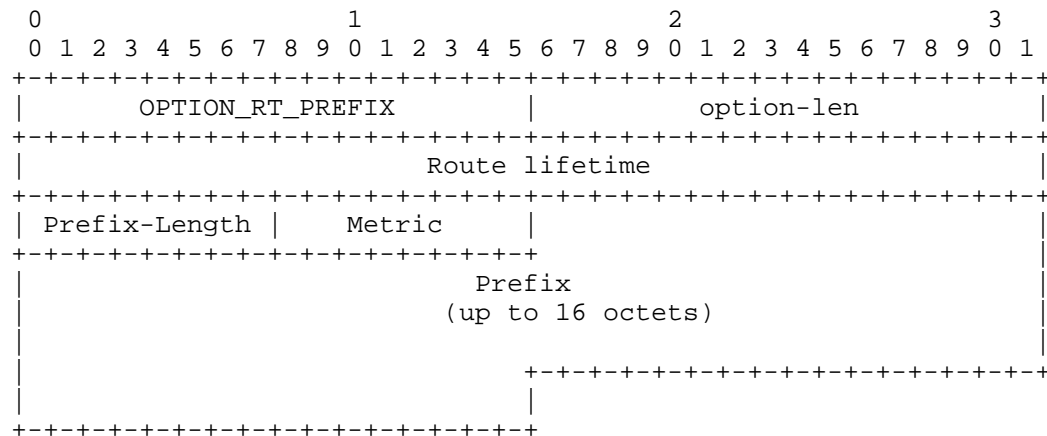


Figure 1: Route Prefix Option Format

```
option-code:  OPTION_RT_PREFIX (TBD2).
```

option-len: Length of the Route Prefix option including all its sub-options.

Route lifetime 32-bit unsigned integer. Specifies lifetime of the route information, expressed in seconds (relative to the time the packet is sent). There are 2 special values defined. 0 means that route is no longer valid and must be removed by clients. A value of all one bits (0xffffffff) represents infinity. means infinity.

Prefix Length: 8-bit unsigned integer. The length in bits of the IP Prefix. The value ranges from 0 to 128. This field represents the number of valid leading bits in the prefix.

Resvd: Reserved field. Server MUST set this value to zero and client MUST ignore its content.

Metric: Route Metric. 8-bit signed integer. The Route Metric indicates whether to prefer the next hop associated with

this prefix over others, when multiple identical prefixes (for different next hops) have been received.

Prefix: a variable size field that specifies Rule IPv6 prefix. Length of the field is defined by prefix6-len field and is rounded up to the nearest octet boundary (if case when Prefix Length is not divisible by 8). In such case additional padding bits must be zeroed.

Values for metric field have meaning based on the value, i.e. higher value indicates higher preference.

3.2. Next Hop Option Format

Each IPv6 route consists of an IPv6 next hop address, an IPv6 destination prefix (a.k.a. the destination subnet), and a host preference value for the route. Elements of such route (e.g. Next hops and prefixes associated with them) are conveyed in NEXT_HOP option that contains RT_PREFIX suboptions.

The Next Hop Option defines the IPv6 address of the next hop, usually corresponding to a specific next-hop router. For each next hop address there can be zero, one or more prefixes reachable via that next hop.

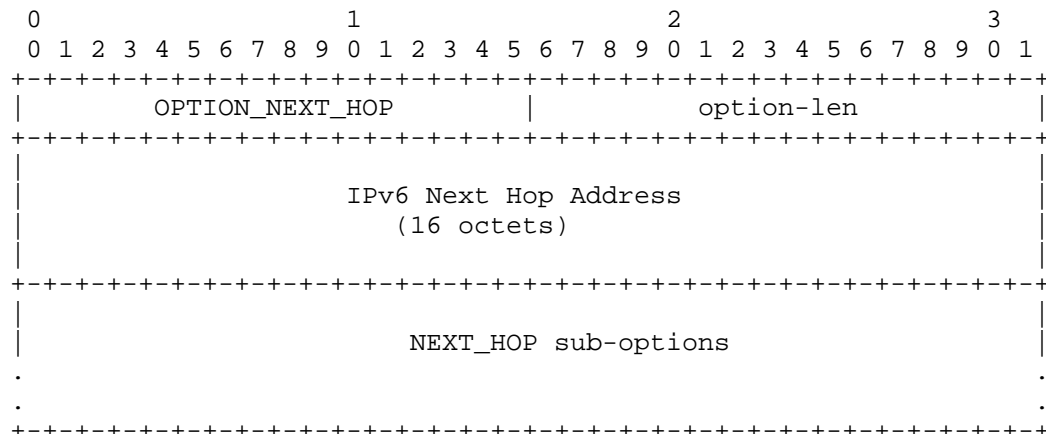


Figure 2: IPv6 Next Hop Option Format

option-code: OPTION_NEXT_HOP (TBD1).

option-len: 16 + Length of NEXT_HOP options field.

IPv6 Next Hop Address: 16 octet long field that specified IPv6 address of the next hop.

NEXT_HOP options: Options associated with this Next Hop. This includes, but is not limited to, zero, one or more RT_PREFIX options that specify prefixes reachable through the given next hop.

NEXT_HOP options: Options associated with this Next Hop. This includes, but is not limited to, zero, one or more **SOURCE_AP** and **RT_PREFIX** options that specify prefixes reachable through the given next hop.

3.3. Source Address/Prefix Option Format

Each IPv6 route consists of an IPv6 next hop address, an IPv6 destination prefix (a.k.a. the destination subnet), and a host preference value for the route. Elements of such route (e.g. Next hops and prefixes associated with them) are conveyed in NEXT_HOP option that contains RT_PREFIX suboptions.

The Source Address/Prefix Option defines the source IPv6 prefix/ address that are assigned from the prefixes that belong to this next hop.

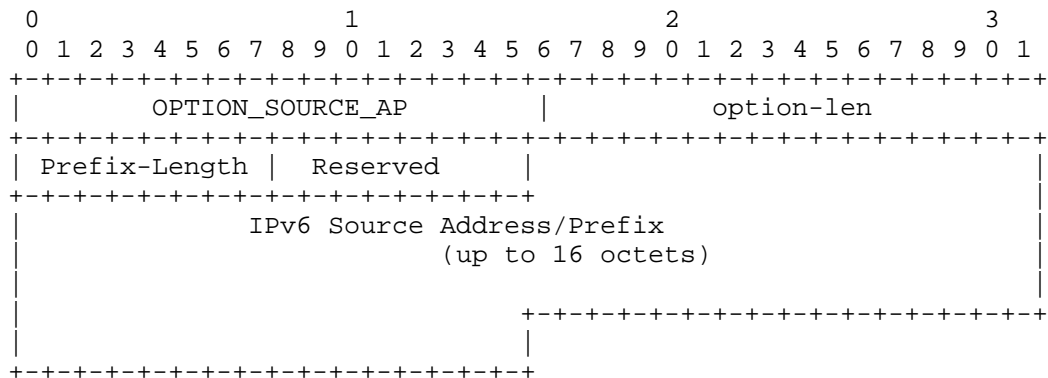


Figure 3: IPv6 Source Address/Prefix Option Format

```
option-code:  OPTION SOURCE AP (TBD1).
```

option-len: 16 + Length of SOURCE_AP options field.

Prefix Length: 8-bit unsigned integer. The length in bits of the IP Prefix. The value ranges from 0 to 128. This field

represents the number of valid leading bits in the prefix.
In case of source address this field is set to 132.

Resvd: Reserved field. Server MUST set this value to zero and client MUST ignore its content.

IPv6 Source Address/Prefix: 16 octet long field that specified IPv6 source address or source prefix.

4. DHCPv6 Server Behavior

When configured to do so, a DHCPv6 server shall provide the NEXT_HOP and RT_PREFIX Options in ADVERTISE and REPLY messages sent to a client that requested the route option. Each Next Hop Option sent by the server must convey at least one Route Prefix Option.

Server includes NEXT_HOP option with possible RT_PREFIX suboptions to designate that specific routes are available via routers. Server includes RT_PREFIX options in Next Hop sub-options directly in Advertise and Reply messages to inform that specific routes are available directly on-link.

If there is more than one route available via specific next hop, server MUST send only one NEXT_HOP for that next hop, which contains multiple RT_PREFIX options. Server MUST NOT send more than one identical (i.e. with equal next hop address field) NEXT_HOP option.

When configured to do so, a DHCPv6 server shall send one or more NEXT_HOP options that contain one or more source addresses Figure 3 included in the Next Hop sub-options field. Each Next Hop Address may be associated with zero, one or more Source Prefix that represent the source addresses that are assigned from the prefixes that belong to this next hop. The Next Hop sub-options field MAY contain Route Prefix options that represent the IPv6 destination prefixes reachable via the given next hop as defined in Figure 2. When configured to do so, a DHCPv6 server shall send NEXT_HOP option with Route Prefix option and Source Prefix in the message in the Next Hop sub-options field to indicate that given prefix is available directly on-link and that any source addresses derived from the source prefix will not be subject to ingress filtering on these routes supported by these next hops.

When configured to do so, a DHCPv6 server shall send one or more NEXT_HOP option that specify the IPv6 next hop addresses and source address. Each Next Hop Address option may be associated with zero, one or more Source Address that represent the source addresses that are assigned from the prefixes that belong to this next hop. The Next Hop sub-options field shall contain Source Address Figure 3 and

Route Prefix options Figure 1 that represent the IPv6 destination prefixes reachable via the given next hop. DHCPv6 server shall include Next Hop Address with Source Address and Route Prefix option in Next Hop sub-options field in the message to indicate that given prefix is available directly on-link and that the source address will not be subject to ingress filtering. For the Source Address, Source Address/Prefix option Figure 3 is used with prefix length set to 128.

Each Next Hop Address may be associated with zero, one or more Source Prefix that represent the source addresses that are assigned from the prefixes that belong to this next hop. The option MAY contain Route Prefix options that represent the IPv6 destination prefixes reachable via the given next hop. DHCP server shall include Next Hop Address with Route Prefix option in Next Hop sub-option field defined in Figure 2 in the message to indicate that given prefix is available directly on-link. To indicate that any source addresses derived from the source prefix will not be subject to ingress filtering on these routes supported by these next hops DHCPv6 server shall send two options, Next Hop option with Route Prefix option in Next Hop options field and a Source Prefix option defined in Figure 3.

Servers SHOULD NOT send NEXT_HOP or RT_PREFIX to clients that did not explicitly requested it, using the ORO.

Servers MUST NOT send NEXT_HOP or RT_PREFIX in messages other than ADVERTISE or REPLY.

Servers MAY also include Status Code Option, defined in Section 22.13 of the [RFC3315] to indicate the status of the operation.

Servers MUST include the Status Code Option, if the requested routing configuration was not successful and SHOULD use status codes as defined in [RFC3315] and [RFC3633].

The maximum number of routing information in one DHCPv6 message depend on the maximum DHCPv6 message size defined in [RFC3315]

5. DHCPv6 Client Behavior

A DHCPv6 client compliant with this specification MUST request the NEXT_HOP and RT_PREFIX Options in an Option Request Option (ORO) in the following messages: Solicit, Request, Renew, Rebind, and Information-Request. The messages are to be sent as and when specified by [RFC3315].

When processing a received Route Options a client MUST substitute a received 0::0 value in the Next Hop Option with the source IPv6 address of the received DHCPv6 message. It MUST also associate a

received Link Local next hop addresses with the interface on which the client received the DHCPv6 message containing the route option. Such a substitution and/or association is useful in cases where the DHCPv6 server operator does not directly know the IPv6 next-hop address, other than knowing it is that of a DHCPv6 relay agent on the client LAN segment. DHCPv6 Packets relayed to the client are sourced by the relay using this relay's IPv6 address, which could be a link local address.

The Client SHOULD refresh assigned route information periodically. The generic DHCPv6 Information Refresh Time Option, as specified in [RFC4242], can be used when it is desired for the client to periodically refresh of route information.

The routes conveyed by the Route Option should be considered as complimentary to any other static route learning and maintenance mechanism used by, or on the client with one modification: The client MUST flush DHCPv6 installed routes following a link flap event on the DHCPv6 client interface over which the routes were installed. This requirement is necessary to automate the flushing of routes for clients that may move to a different network.

Client MUST confirm that routers announced over DHCPv6 are reachable, using one of methods suitable for specific network type. The most common mechanism is Neighbor Unreachability Detection (NUD), specified in [RFC4861]. Client SHOULD use NUD to verify that received routers are reachable before adjusting its routing tables. Client MAY use other reachability verification mechanisms specific to used network technology. To avoid potential long-lived routing black holes, client MAY periodically confirm that router is still reachable.

5.1. Conflict resolution

Information received via Route Options over DHCPv6 MUST be treated equally to routing information obtained via other sources. In particular, from the RA perspective, DHCPv6 provisioning should be treated as if yet another RA was received. Preference field should be taken into consideration during route information processing. In particular, administrators are encouraged to read [RFC4191], Section 4.1 for guidance.

To facilitate information merge between DHCPv6 and RA, DHCPv6 options in this document convey the same information specified in [I-D.sarikaya-6man-next-hop-ra].

To facilitate information merge between DHCPv6 and RA, DHCPv6 option RT_PREFIX conveys the same information specified in [RFC4191] albeit on-wire format is slightly different. The differences are:

Metric field is an 8-bit field that conveys the route metric.

RIO uses 128-length prefix field, while DHCPv6 option uses variable prefix length. That difference is used to minimize packet size as it avoid transmitting zeroed octets. Despite slightly different encoding, delivered information is exactly the same.

If prefix is available directly on-link, Route Prefix option is conveyed directly in DHCPv6 message, not within Next Hop option. That feature is considered a superset, compared to RIO.

In short, when DHCPv6 RT_PREFIX option is used alone this specification works in compatibility mode with [RFC4191].

6. IANA Considerations

IANA is kindly requested to allocate DHCPv6 option code TBD1 to the OPTION_NEXT_HOP, TBD2 to OPTION_RT_PREFIX, TBD3 to OPTION_SOURCE_AP. All values should be added to the DHCPv6 option code space defined in Section 24.3 of [RFC3315].

7. Security Considerations

The overall security considerations discussed in [RFC3315] apply also to this document. The Route option could be used by malicious parties to misdirect traffic sent by the client either as part of a denial of service or man-in-the-middle attack. An alternative denial of service attack could also be realized by means of using the route option to overflowing any known memory limitations of the client, or to exceed the client's ability to handle the number of next hop addresses.

Neither of the above considerations are new and specific to the proposed route option. The mechanisms identified for securing DHCPv6 as well as reasonable checks performed by client implementations are deemed sufficient in addressing these problems.

It is essential that clients verify that announced routers are indeed reachable, as specified in Section 5. Failing to do so may create black hole routing problem.

This mechanism may introduce severe problems if deployed in networks that use dynamic routing protocols. See Section 2.6 for details.

DHCPv6 becomes a complete provisioning protocol with this mechanism, i.e. all necessary configuration parameters may be delivered using DHCPv6 only. It was suggested that in some cases this may lead to decision of disabling RA. While RA-less networks could offer lower operational expenses and protection against rogue RAs, they would not work with nodes that do not support this feature. Therefore such decision is not recommended, unless all effects are carefully analyzed. It is worth noting that disabling RA support in hosts would solve rogue RA problem, it would in fact only change the issue into rogue DHCPv6 problem. That is somewhat beneficial, however, as rogue RA may affect all nodes immediately while rogue DHCPv6 server will affect only new nodes, that boot up after rogue server manifests itself.

Reader is also encouraged to read DHCPv6 security considerations document [I-D.ietf-dhc-sedhcpv6].

8. Acknowledgements

The author acknowledges the work done by his co-authors in MIF WG draft entitled DHCPv6 Route Options.

9. References

9.1. 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.
- [RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

9.2. Informative References

- [I-D.ietf-dhc-sedhcpv6]
Jiang, S., Shen, S., Zhang, D., and T. Jinmei, "Secure DHCPv6 with Public Key", draft-ietf-dhc-sedhcpv6-03 (work in progress), June 2014.
- [I-D.ietf-mif-mpvd-arch]
Anipko, D., "Multiple Provisioning Domain Architecture", draft-ietf-mif-mpvd-arch-01 (work in progress), May 2014.

- [I-D.sarikaya-6man-next-hop-ra]
Sarikaya, B., "IPv6 RA Options for Next Hop Routes",
draft-sarikaya-6man-next-hop-ra-02 (work in progress),
June 2014.
- [RFC3442] Lemon, T., Cheshire, S., and B. Volz, "The Classless
Static Route Option for Dynamic Host Configuration
Protocol (DHCP) version 4", RFC 3442, December 2002.
- [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and
More-Specific Routes", RFC 4191, November 2005.
- [RFC4242] Venaas, S., Chown, T., and B. Volz, "Information Refresh
Time Option for Dynamic Host Configuration Protocol for
IPv6 (DHCPv6)", RFC 4242, November 2005.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman,
"Neighbor Discovery for IP version 6 (IPv6)", RFC 4861,
September 2007.
- [RFC6204] Singh, H., Beebee, W., Donley, C., Stark, B., and O.
Troan, "Basic Requirements for IPv6 Customer Edge
Routers", RFC 6204, April 2011.
- [RFC7157] Troan, O., Miles, D., Matsushima, S., Okimoto, T., and D.
Wing, "IPv6 Multihoming without Network Address
Translation", RFC 7157, March 2014.

Author's Address

Behcet Sarikaya
Huawei USA
5340 Legacy Dr.
Plano, TX 75024
United States

Phone: +1 972-509-5599
Email: sarikaya@ieee.org

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 12, 2015

B. Skeen
Boeing Phantom Works
E. King
Boeing EO&T IT
F. Templin, Ed.
Boeing Research & Technology
September 08, 2014

Including Geolocation Information in IPv6 Packet Headers (IPv6 GEO)
draft-skeen-6man-ipv6geo-01.txt

Abstract

This document provides a specification for including geolocation information in the headers of IPv6 packets (IPv6 GEO). The information is intended to be included in packets for which the location of the source node is to be conveyed via the network to the destination node or nodes.

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 12, 2015.

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. Requirements	4
4. Motivation and Applicability	4
5. IPv6 GEO Specification	6
5.1. IPv6 GEO Destination Option Format	6
5.2. IPv6 GEO Option Encoding Algorithm	8
5.3. IPv6 Node Requirements	9
6. IANA Considerations	9
7. Security Considerations	9
8. Related Work in the IETF	9
9. Implementation Status	10
10. Contributors	10
11. Acknowledgments	10
12. References	10
12.1. Normative References	10
12.2. Informative References	11
Authors' Addresses	11

1. Introduction

Internet Protocol, version 4 (IPv4) [RFC0791] provides limited capabilities for including additional information in the headers of packets. The maximum IPv4 header length is 60 bytes including any IP options, and options are not widely used due to incompatibilities with network middleboxes. On the other hand, Internet Protocol, version 6 (IPv6) [RFC2460] includes an extensible header format whereby additional information can be inserted between the IPv6 header and the transport layer header. These extensions can be included on a per-packet basis, and not necessarily for all packets of the same flow. This document specifies a format for including geolocation information within the headers of individual IPv6 packets (IPv6 GEO).

IPv6 GEO information is included at the discretion of source nodes for the benefit of destination nodes and/or network elements that may need to examine the headers of packets in transit. Legacy destination nodes that do not recognize the IPv6 GEO information must ignore it and process the rest of the packet as if it were not present. The IPv6 specification defines several extension header types, including the Destination Options header. Section 4.6 of [RFC2460] describes conditions under which new information should be

encoded as either a new extension header or as a new destination option:

"Note that there are two possible ways to encode optional destination information in an IPv6 packet: either as an option in the Destination Options header, or as a separate extension header. The Fragment header and the Authentication header are examples of the latter approach. Which approach can be used depends on what action is desired of a destination node that does not understand the optional information:"

Section 3 of [RFC6564] further states that:

"The base IPv6 standard [RFC2460] allows the use of both extension headers and destination options in order to encode optional destination information in an IPv6 packet. The use of destination options to encode this information provides more flexible handling characteristics and better backward compatibility than using extension headers. Because of this, implementations SHOULD use destination options as the preferred mechanism for encoding optional destination information, and use a new extension header only if destination options do not satisfy their needs. The request for creation of a new IPv6 extension header MUST be accompanied by a specific explanation of why destination options could not be used to convey this information."

Our first interpretation of this guidance and the supporting text that follows suggests that, since IPv6 GEO information must be ignored by legacy destination nodes, encoding as a Destination Option is indicated. Further investigation and community input may indicate that a new extension header type is instead warranted. In either case, future versions of this document will adopt the encoding approach indicated by community consensus.

2. Terminology

The following terms are defined within the scope of this document:

IPv6 Geolocation (IPv6 GEO)

a means for identifying the location of the source of an IPv6 packet based on geographical coordinates, altitude, timestamp and/or other information conveyed from the source to the destination(s).

3. Requirements

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]. When used in lower case (e.g., must, must not, etc.), these words MUST NOT be interpreted as described in [RFC2119], but are rather interpreted as they would be in common English.

4. Motivation and Applicability

Traditionally, a given source node will include a set of identifying criteria that can be used to help determine the relative location of that node on the network. Such criteria include, but are not limited to, IP address, Ethernet MAC addresses, 802.11 or Bluetooth MAC addresses, Wifi and RFID tags, or other user-defined variables that may be specific to a given implementation. However, these variables are often unreliable in determining the physical location of a source node as modern networks are typically implemented with a logical "layer 2" structure without emphasis on the node's physical location. Furthermore, variables such as IP address and Wifi RFID tags are commonly defined by a network administrator and are subject to the implementation criteria of a given network, and therefore are susceptible to error in identifying the location of a given node since there is no common mechanism for associating these criteria to a given physical location. In addition, the proliferation of portable and handheld mobile devices makes it increasingly likely that nodes will at some point change the point of attachment to a given network and will need to be identified and likely authenticated against a set of reliable location-based criteria.

In the absence of location-based authentication criteria, a host will typically be configured to require either local parameters, i.e., username and password, or a strong "two-factor" authentication mechanism, or both. Whereas the merit and applicability of these methods is outside the scope of this document, some implementations require an additional layer of authentication control based on the physical location of a given source node. As a result, a means for identifying the location of the source node based on the geographical coordinates, altitude, timestamp and/or other information is needed.

Numerous use cases can be identified for location-based authentication control that would require the source node to provide its current location to one or more destination node(s). The source node to be geolocated can be defined as any IPv6 GEO node capable of encoding the geolocation data within the IPv6 Destination Options header; for example, an airplane, a remote corporate user, a ground soldier, or an unmanned aerial vehicle, to name a few. The

destination node can be any IPv6 node that can interpret the IPv6 GEO encoded data contained in the Destination Options header; for example, an authentication server responsible for deriving the geolocation criteria received from the source node and authenticating it against a location-based access policy.

Potential use cases for IPv6 GEO include:

- o A remote corporate user that requires an encrypted tunnel connection to a corporate VPN server must provide authentic location information. In addition to a two-factor authentication request, an IPv6 source node using IPv6 GEO would also encode its geolocation data into the authentication request to be sent to the corporate VPN server. The corporate VPN server would authenticate the specified location of the source node to the corporate policy that includes the list of approved locations for the source node on the corporate authentication server in order to accept the connection request.
- o An expeditionary team may want to relay geolocation data to a mission control center in order to provide emergency response coordinates, humanitarian support vectors, new terrain characteristics, or as a means to coordinate the search of a large geographic region. Further, a method to authenticate the control messages sent from the expedition team leader to the control center may require that the geolocation authenticity of the messages be verified
- o A first responder may require a rapidly deployable means of providing geolocation data to emergency teams engaged in rescuing lost or injured personnel or in coordinating the location of support personnel conducting a search over wide geographic areas. The ability to provide location awareness could provide the critical communication needed to reduce the time to contact in life-threatening emergency situations.
- o Civil aviation Air Traffic Management (ATM) systems require a means for tracking the location of aircraft in their various phases of flight (both on the ground and in the sky). As ATM becomes increasingly dependent on data communications, the ability to associate an aircraft's location with its communications messaging can augment and in some instances replace mechanisms such as Automatic Dependent Surveillance - Broadcast (ADS-B).
- o Unmanned Air Systems (UAS) are envisioned in a wide variety of use cases. IPv6 GEO information sharing for both ground control and UAS-to-UAS communications will naturally result in more effective fleet coordination and tracking.

- o Space exploration vehicles must be tracked by control stations and other vehicles throughout all mission phases. Especially for deep space applications, an extraterrestrial location coordinate system may be needed.
- o Convergence of dynamic routing protocols in a wide variety of mobile networks can benefit greatly from knowledge of the geographical locations of prospective neighbors. This information is best conveyed in the headers of IPv6 packets used for routing protocol control message exchanges.
- o The networks that make up the greater "Internet," including all various forms of Intranets (Enterprises, small businesses, Service Providers, etc...) all need to manage those assets that constitute their administrative domain. Sometimes these networks are millions of dollars and all of the time are critical to business value. Being able to locate and place where these devices are located mean actual dollar value to the businesses bottom line because of various tax and depreciation details that are variable, depending on which taxing authority these devices are located (City, State (Province), Country or any other various taxing authority in which the business provides value with those assets. Having a clear location, at any time has distinct advantages to the business as to where exactly those devices are, at any one time.

In these cases, the actual implementation of a geolocation authentication layer in a multi-layered security scheme is considered outside the scope of this document. This document seeks to specify a method for including the geolocation data in the IPv6 Destination Options header in order for it to be utilized in the manner specified by a set of given implementation criteria.

In the final analysis, if a subject node that willingly submits itself for surveillance sends only a single IPv6 packet or fragment before falling silent, then any tracking node(s) should be able to determine where the packet came from.

5. IPv6 GEO Specification

5.1. IPv6 GEO Destination Option Format

The IPv6 GEO "Type 0" Destination Option is formatted as shown in Figure 1:

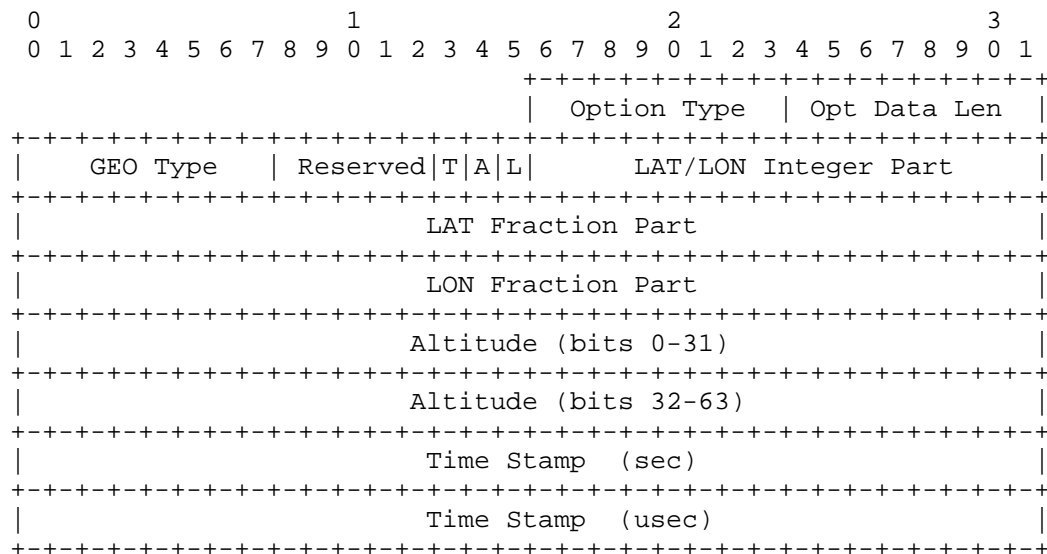


Figure 1: IPv6 GEO Type 0 Destination Option Format

The fields of the option are defined as follows:

Option Type (8)

the IPv6 Option Type code for IPv6 GEO; to be assigned by IANA. The high order three bits of the Option Type encode the value '000' to indicate that the option is to be skipped over if not recognized, and that the data must not change en route (see: Section 4.2 of [RFC2460]).

Opt Data Len (8)

the length of the data portion of the IPv6 GEO Option.

GEO Type (8)

the IPv6 GEO encoding type; set to 0 for the encapsulation format specified in this section.

Flags (8)

an 8-bit flags field. Contains a 5-bit Reserved field that is set to 0 on transmission and ignored on reception. The following three bits (T, A, L) are set to 1 if the corresponding GEO information fields are included and set to 0 otherwise.

LAT/LON Integer Part (16)

a 16 bit field that encodes the integer part of the Latitude and Longitude coordinates (see below). Included when 'L' is 1 and omitted when 'L' is 0.

LAT Fraction Part (32)

a 32 bit field that encodes the fractional part of the Latitude coordinate (see below). Included when 'L' is 1 and omitted when 'L' is 0.

LON Fractional Part (32)

a 32 bit field that encodes the fractional part of the Longitude coordinate (see below). Included when 'L' is 1 and omitted when 'L' is 0.

Altitude (64)

two 32-bit fields that together encode the altitude (in centimeters). Included when 'A' is 1 and omitted when 'A' is 0.

Time Stamp (sec) (32)

a 32 bit field that encodes the time that the IPv6 GEO data was generated in seconds since the epoch (00:00:00 UTC on 1 January 1970). Included when 'T' is 1 and omitted when 'T' is 0.

Time Stamp (usec) (32)

a 32 bit field that encodes the microseconds at the time that the IPv6 GEO data was generated. Included when 'T' is 1 and omitted when 'T' is 0.

In the language of Section 4.2 of [RFC2460], the option has alignment requirement '4n+2' when the 'L' flag is set and '4n' when the 'L' flag is clear. Future specifications may include new IPv6 GEO types to encode alternate formats.

5.2. IPv6 GEO Option Encoding Algorithm

The Latitude (LAT) and Longitude (LON) coordinate values are treated as floating point numbers with 10^{-10} precision. LAT values range from 0 at the equator to +90 northward and -90 southward. LON values range from 0 at the IERS Reference Meridian [WGS-84] to +180 eastward and -180 westward. The LAT/LON coordinates are then encoded as follows:

LAT/LON Integer Part = $\text{int}(\text{LAT}+90)*360 + \text{int}(\text{LON}+180)$

LAT Fraction Part = $\text{fra}(\text{LAT})*1,000,000,000$

LON Fraction Part = $\text{fra}(\text{LON})*1,000,000,000$

where "int()" returns the integer part of the floating point number and "fra()" returns the fractional part of the floating point number. This encoding scheme is similar to one proposed in "Efficient WGS84 (aka GPS) coordinates compression" [WGS-ENCODE].

5.3. IPv6 Node Requirements

IPv6 source hosts MAY insert the IPv6 GEO destination option in any IPv6 packets they send to IPv6 destinations (unicast, multicast or anycast). Any IPv6 packet is eligible, including a minimal packet that includes only an (extended) IPv6 header with the value "No Next Header" in the final "Next Header" field.

If the host inserts the IPv6 GEO destination option, it MUST construct the option using the format specified in Section 5.1 and using the encoding algorithm specified in Section 5.2. The host MUST further ensure that the geolocation information encoded in the option is current and accurate.

IPv6 destinations that do not recognize the IPv6 GEO destination option MUST ignore it and continue to process the IPv6 destination options extension header as though the IPv6 GEO option were not present.

6. IANA Considerations

IANA is requested to allocate an IPv6 Option number for the IPv6 GEO Option in the "Destination Options and Hop-by-Hop Options" registry.

7. Security Considerations

Packets with IPv6 GEO options that are sent in the clear without encryption risk exposure of sensitive information to unauthorized eavesdroppers. When location privacy is desired, Internet security protocols (e.g., IPsec [RFC4301], etc.) and/or link layer security SHOULD be used to ensure confidentiality.

A spoofing attack is exposed when a source includes forged IPv6 GEO information that is incorrect for its current location and/or time. Destinations SHOULD therefore authenticate the source of IPv6 packets before accepting any IPv6 GEO information they may include.

User agents MUST NOT send geolocation information to unauthorized correspondents (e.g., Web sites, etc.) without the express permission of the user.

8. Related Work in the IETF

The IETF GEOPRIV working group is chartered to "continue to develop and refine representations of location in Internet protocols, and to analyze the authorization, integrity, and privacy requirements that must be met when these representations of location are created, stored, and used". However, the group is located within the Real-

time Applications and Infrastructure area, and as such it is not clear whether the Internet layer approach proposed in this document would fit within the area focus. The GEOPRIV working group has published a BCP on "An Architecture for Location and Location Privacy in Internet Applications" [RFC6280].

A BoF on "Internet-wide Geo-Networking (geonet)" was held at IETF88 in November 2013. A Problem Statement related to the BoF states that: "Internet-based applications use IP addresses to address a node that can be a host, a server or a router. Scenarios and use cases exist where nodes are being addressed using their geographical location instead of their IP address" [I-D.karagiannis-problem-statement-geonetworking]. This BoF was held within the Internet area and concerns geolocation at the Internet layer.

9. Implementation Status

A prototype implementation has been developed and tested, but not yet available for public release. The prototype implementation uses the Option Type value reserved for experimentation [RFC3692].

10. Contributors

The authors greatly appreciate the efforts of Jin Fang, who jointly developed the IPv6 GEO message format and was the primary author of the prototype implementation. We wish Jin the best of success in his future endeavors.

11. Acknowledgments

The following individuals are acknowledged for helpful comments and suggestions: Jeff Ahrenholz, Kerry Hu.

12. References

12.1. Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

- [RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, January 2004.
- [RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, April 2012.

12.2. Informative References

- [I-D.karagiannis-problem-statement-geonetworking] Karagiannis, G., Heijenk, G., Festag, A., Petrescu, A., and A. Chaiken, "Internet-wide Geo-networking Problem Statement", draft-karagiannis-problem-statement-geonetworking-01 (work in progress), November 2013.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.
- [RFC6280] Barnes, R., Lepinski, M., Cooper, A., Morris, J., Tschofenig, H., and H. Schulzrinne, "An Architecture for Location and Location Privacy in Internet Applications", BCP 160, RFC 6280, July 2011.
- [WGS-84] Wikipedia, W., "World Geodetic System (http://en.wikipedia.org/wiki/World_Geodetic_System)", November 2013.
- [WGS-ENCODE] Dupuis, L., "Efficient WGS84 (aka GPS) Coordinates Compression (<http://www.dupuis.me/node/35>)", August 2013.

Authors' Addresses

Brian Skeen
Boeing Phantom Works
P.O. Box 3707
Seattle, WA 98124
USA

Email: brian.l.skeen@boeing.com

Edwin King
Boeing EO&T IT
P.O. Box 3707
Seattle, WA 98124
USA

Email: edwin.e.king@boeing.com

Fred L. Templin (editor)
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
USA

Email: fltemplin@acm.org

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 4, 2017

B. Skeen
Boeing Phantom Works
E. King
Boeing EO&T IT
F. Templin, Ed.
Boeing Research & Technology
October 31, 2016

Including Geolocation Information in IPv6 Packet Headers (IPv6 GEO)
draft-skeen-6man-ipv6geo-03.txt

Abstract

This document provides a specification for including geolocation information in the headers of IPv6 packets (IPv6 GEO). The information is intended to be included in packets for which the location of the source node is to be conveyed via the network to the destination node or nodes.

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 May 4, 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	3
3. Requirements	4
4. Motivation and Applicability	4
5. IPv6 GEO Specification	7
5.1. IPv6 GEO Destination Option Format	7
5.2. IPv6 GEO Option Encoding Algorithm	9
5.3. IPv6 Node Requirements	9
6. IANA Considerations	9
7. Security Considerations	10
8. Related Work in the IETF	10
9. Implementation Status	11
10. Contributors	11
11. Acknowledgments	11
12. References	11
12.1. Normative References	11
12.2. Informative References	12
Authors' Addresses	12

1. Introduction

Internet Protocol, version 4 (IPv4) [RFC0791] provides limited capabilities for including additional information in the headers of packets. The maximum IPv4 header length is 60 bytes including any IP options, and options are not widely used due to incompatibilities with network middleboxes. On the other hand, Internet Protocol, version 6 (IPv6) [RFC2460] includes an extensible header format whereby additional information can be inserted between the IPv6 header and the transport layer header. These extensions can be included on a per-packet basis, and not necessarily for all packets of the same flow. This document specifies a format for including geolocation information within the headers of individual IPv6 packets (IPv6 GEO).

IPv6 GEO information is included at the discretion of source nodes for the benefit of destination nodes and/or network elements that may need to examine the headers of packets in transit. Legacy destination nodes that do not recognize the IPv6 GEO information must ignore it and process the rest of the packet as if it were not present. The IPv6 specification defines several extension header types, including the Destination Options header. Section 4.6 of [RFC2460] describes conditions under which new information should be

encoded as either a new extension header or as a new destination option:

"Note that there are two possible ways to encode optional destination information in an IPv6 packet: either as an option in the Destination Options header, or as a separate extension header. The Fragment header and the Authentication header are examples of the latter approach. Which approach can be used depends on what action is desired of a destination node that does not understand the optional information:"

Section 3 of [RFC6564] further states that:

"The base IPv6 standard [RFC2460] allows the use of both extension headers and destination options in order to encode optional destination information in an IPv6 packet. The use of destination options to encode this information provides more flexible handling characteristics and better backward compatibility than using extension headers. Because of this, implementations SHOULD use destination options as the preferred mechanism for encoding optional destination information, and use a new extension header only if destination options do not satisfy their needs. The request for creation of a new IPv6 extension header MUST be accompanied by a specific explanation of why destination options could not be used to convey this information."

Our first interpretation of this guidance and the supporting text that follows suggests that, since IPv6 GEO information must be ignored by legacy destination nodes, encoding as a Destination Option is indicated. Further investigation and community input may indicate that a new extension header type is instead warranted. In either case, future versions of this document will adopt the encoding approach indicated by community consensus.

2. Terminology

The following terms are defined within the scope of this document:

IPv6 Geolocation (IPv6 GEO)

a means for identifying the location of the source of an IPv6 packet based on geographical coordinates, altitude, timestamp and/or other information conveyed from the source to the destination(s).

3. Requirements

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]. When used in lower case (e.g., must, must not, etc.), these words MUST NOT be interpreted as described in [RFC2119], but are rather interpreted as they would be in common English.

IPv6 forwarding nodes must not discard packets that include the destination options specified herein unless by explicit administrative policy. General forwarding considerations for packets that contain IPv6 options are discussed in [I-D.ietf-opsec-ipv6-eh-filtering].

4. Motivation and Applicability

Traditionally, a given source node will include a set of identifying criteria that can be used to help determine the relative location of that node on the network. Such criteria include, but are not limited to, IP address, Ethernet MAC addresses, 802.11 or Bluetooth MAC addresses, Wifi and RFID tags, or other user-defined variables that may be specific to a given implementation. However, these variables are often unreliable in determining the physical location of a source node as modern networks are typically implemented with a logical "layer 2" structure without emphasis on the node's physical location. Furthermore, variables such as IP address and Wifi RFID tags are commonly defined by a network administrator and are subject to the implementation criteria of a given network, and therefore are susceptible to error in identifying the location of a given node since there is no common mechanism for associating these criteria to a given physical location. In addition, the proliferation of portable and handheld mobile devices makes it increasingly likely that nodes will at some point change the point of attachment to a given network and will need to be identified and likely authenticated against a set of reliable location-based criteria.

In the absence of location-based authentication criteria, a host will typically be configured to require either local parameters, i.e., username and password, or a strong "two-factor" authentication mechanism, or both. Whereas the merit and applicability of these methods is outside the scope of this document, some implementations require an additional layer of authentication control based on the physical location of a given source node. As a result, a means for identifying the location of the source node based on the geographical coordinates, altitude, timestamp and/or other information is needed.

Numerous use cases can be identified for location-based authentication control that would require the source node to provide its current location to one or more destination node(s). The source node to be geolocated can be defined as any IPv6 GEO node capable of encoding the geolocation data within the IPv6 Destination Options header; for example, an airplane, an automobile, a remote corporate user, a ground soldier, or an unmanned aerial vehicle, to name a few. The destination node can be any IPv6 node that can interpret the IPv6 GEO encoded data contained in the Destination Options header; for example, an authentication server responsible for deriving the geolocation criteria received from the source node and authenticating it against a location-based access policy.

Potential use cases for IPv6 GEO include:

- o A remote corporate user that requires an encrypted tunnel connection to a corporate VPN server must provide authentic location information. In addition to a two-factor authentication request, an IPv6 source node using IPv6 GEO would also encode its geolocation data into the authentication request to be sent to the corporate VPN server. The corporate VPN server would authenticate the specified location of the source node to the corporate policy that includes the list of approved locations for the source node on the corporate authentication server in order to accept the connection request.
- o An expeditionary team may want to relay geolocation data to a mission control center in order to provide emergency response coordinates, humanitarian support vectors, new terrain characteristics, or as a means to coordinate the search of a large geographic region. Further, a method to authenticate the control messages sent from the expedition team leader to the control center may require that the geolocation authenticity of the messages be verified
- o A first responder may require a rapidly deployable means of providing geolocation data to emergency teams engaged in rescuing lost or injured personnel or in coordinating the location of support personnel conducting a search over wide geographic areas. The ability to provide location awareness could provide the critical communication needed to reduce the time to contact in life-threatening emergency situations.
- o Civil aviation Air Traffic Management (ATM) systems require a means for tracking the location of aircraft in their various phases of flight (both on the ground and in the sky). As ATM becomes increasingly dependent on data communications, the ability to associate an aircraft's location with its communications

messaging can augment and in some instances replace mechanisms such as Automatic Dependent Surveillance - Broadcast (ADS-B).

- o Unmanned Air Systems (UAS) are envisioned in a wide variety of use cases. IPv6 GEO information sharing for both ground control and UAS-to-UAS communications will naturally result in more effective fleet coordination and tracking.
- o Automobiles and vehicles of all types are increasingly connected to the Internet. Comfort-enhancing entertainment applications, road safety applications using bidirectional data flows, and connected automated driving are but a few new features expected in automobiles to hit the roads from now to year 2020. Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) use-cases where IP is well-suited as a networking technology, supporting also applications that involve exchanges of safety-related messages between vehicles and infrastructure if necessary.
- o Space exploration vehicles must be tracked by control stations and other vehicles throughout all mission phases. Especially for deep space applications, an extraterrestrial location coordinate system may be needed.
- o Convergence of dynamic routing protocols in a wide variety of mobile networks can benefit greatly from knowledge of the geographical locations of prospective neighbors. This information is best conveyed in the headers of IPv6 packets used for routing protocol control message exchanges.
- o The networks that make up the greater "Internet," including all various forms of Intranets (Enterprises, small businesses, Service Providers, etc.) all need to manage those assets that constitute their administrative domain. Sometimes these networks are millions of dollars and all of the time are critical to business value. Being able to locate and place where these devices are located may mean actual dollar value to the businesses bottom line because of various tax and depreciation details that are variable, depending on which taxing authority these devices are located (City, State (Province), Country or any other various taxing authority in which the business provides value with those assets. Having a clear location, at any time has distinct advantages to the business as to where exactly those devices are, at any one time.

In these cases, the actual implementation of a geolocation authentication layer in a multi-layered security scheme is considered outside the scope of this document. This document seeks to specify a method for including the geolocation data in the IPv6 Destination

Options header in order for it to be utilized in the manner specified by a set of given implementation criteria.

In the final analysis, if a subject node that willingly submits itself for surveillance sends only a single IPv6 packet or fragment before falling silent, then any tracking node(s) should be able to determine where the packet came from.

5. IPv6 GEO Specification

5.1. IPv6 GEO Destination Option Format

The IPv6 GEO "Type 0" Destination Option is formatted as shown in Figure 1:

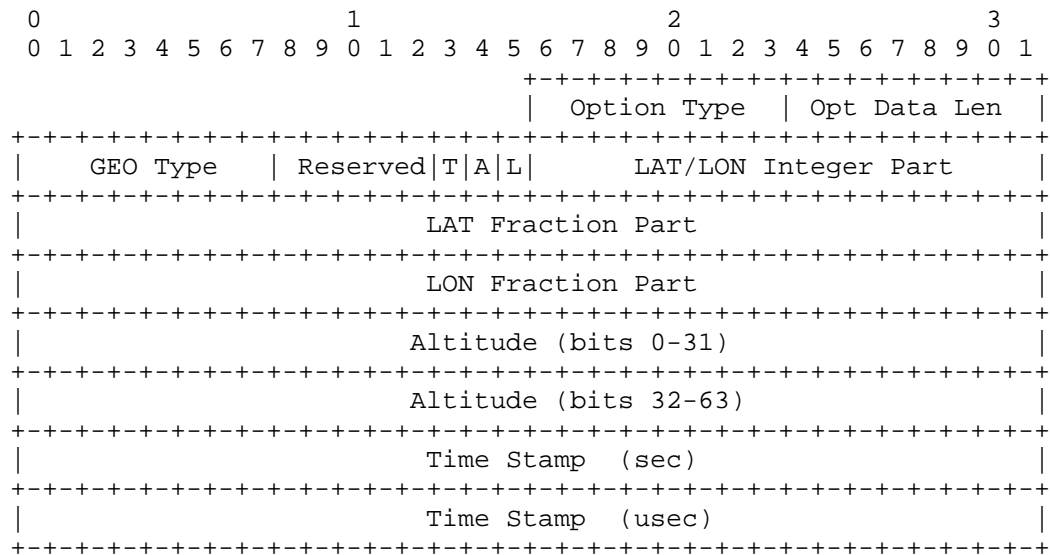


Figure 1: IPv6 GEO Type 0 Destination Option Format

The fields of the option are defined as follows:

Option Type (8)

the IPv6 Option Type code for IPv6 GEO; to be assigned by IANA. The high order three bits of the Option Type encode the value '000' to indicate that the option is to be skipped over if not recognized, and that the data must not change en route (see: Section 4.2 of [RFC2460]).

Opt Data Len (8)

the length of the data portion of the IPv6 GEO Option.

GEO Type (8)

the IPv6 GEO encoding type; set to 0 for the encapsulation format specified in this section.

Flags (8)

an 8-bit flags field. Contains a 5-bit Reserved field that is set to 0 on transmission and ignored on reception. The following three bits (T, A, L) are set to 1 if the corresponding GEO information fields are included and set to 0 otherwise.

LAT/LON Integer Part (16)

a 16 bit field that encodes the integer part of the Latitude and Longitude coordinates (see below). Included when 'L' is 1 and omitted when 'L' is 0.

LAT Fraction Part (32)

a 32 bit field that encodes the fractional part of the Latitude coordinate (see below). Included when 'L' is 1 and omitted when 'L' is 0.

LON Fractional Part (32)

a 32 bit field that encodes the fractional part of the Longitude coordinate (see below). Included when 'L' is 1 and omitted when 'L' is 0.

Altitude (64)

two 32-bit fields that together encode the altitude (in centimeters). Included when 'A' is 1 and omitted when 'A' is 0.

Time Stamp (sec) (32)

a 32 bit field that encodes the time that the IPv6 GEO data was generated in seconds since the epoch (00:00:00 UTC on 1 January 1970). Included when 'T' is 1 and omitted when 'T' is 0.

Time Stamp (usec) (32)

a 32 bit field that encodes the microseconds at the time that the IPv6 GEO data was generated. Included when 'T' is 1 and omitted when 'T' is 0.

In the language of Section 4.2 of [RFC2460], the option has alignment requirement '4n+2' when the 'L' flag is set and '4n' when the 'L' flag is clear. Future specifications may include new IPv6 GEO types to encode alternate formats.

5.2. IPv6 GEO Option Encoding Algorithm

The Latitude (LAT) and Longitude (LON) coordinate values are treated as floating point numbers with 10^{-10} precision. LAT values range from 0 degrees at the equator to +90 degrees northward and -90 degrees southward. LON values range from 0 degrees at the IERS Reference Meridian [WGS-84] to +180 degrees eastward and -180 degrees westward. The LAT/LON coordinates are then encoded as follows:

$$\text{LAT/LON Integer Part} = \text{int}(\text{LAT}+90)*360 + \text{int}(\text{LON}+180)$$
$$\text{LAT Fraction Part} = \text{fra}(\text{LAT})*1,000,000,000$$
$$\text{LON Fraction Part} = \text{fra}(\text{LON})*1,000,000,000$$

where "int()" returns the integer part of the floating point number and "fra()" returns the fractional part of the floating point number. This encoding scheme is similar to one proposed in "Efficient WGS84 (aka GPS) coordinates compression" [WGS-ENCODE].

5.3. IPv6 Node Requirements

IPv6 source hosts MAY insert the IPv6 GEO destination option in any IPv6 packets they send to IPv6 destinations (unicast, multicast or anycast). Any IPv6 packet is eligible, including a minimal packet that includes only an (extended) IPv6 header with the value "No Next Header" in the final "Next Header" field.

If the host inserts the IPv6 GEO destination option, it MUST construct the option using the format specified in Section 5.1 and using the encoding algorithm specified in Section 5.2. The host MUST further ensure that the geolocation information encoded in the option is current and accurate.

IPv6 destinations that do not recognize the IPv6 GEO destination option MUST ignore it and continue to process the IPv6 destination options extension header as though the IPv6 GEO option were not present.

6. IANA Considerations

IANA is requested to allocate an IPv6 Option number for the IPv6 GEO Option in the "Destination Options and Hop-by-Hop Options" registry.

7. Security Considerations

Packets with IPv6 GEO options that are sent in the clear without encryption risk exposure of sensitive information to unauthorized eavesdroppers. When location privacy is desired, Internet security protocols (e.g., IPsec [RFC4301], etc.) and/or link layer security SHOULD be used to ensure confidentiality.

A spoofing attack is exposed when a source includes forged IPv6 GEO information that is incorrect for its current location and/or time. Destinations SHOULD therefore authenticate the source of IPv6 packets before accepting any IPv6 GEO information they may include.

User agents MUST NOT send geolocation information to unauthorized correspondents (e.g., Web sites, etc.) without the express permission of the user.

8. Related Work in the IETF

The IETF GEOPRIV working group is chartered to "continue to develop and refine representations of location in Internet protocols, and to analyze the authorization, integrity, and privacy requirements that must be met when these representations of location are created, stored, and used". However, the group is located within the Real-time Applications and Infrastructure area, and as such it is not clear whether the Internet layer approach proposed in this document would fit within the area focus. The GEOPRIV working group has published a BCP on "An Architecture for Location and Location Privacy in Internet Applications" [RFC6280].

A BoF on "Internet-wide Geo-Networking (geonet)" was held at IETF88 in November 2013. A Problem Statement related to the BoF states that: "Internet-based applications use IP addresses to address a node that can be a host, a server or a router. Scenarios and use cases exist where nodes are being addressed using their geographical location instead of their IP address" [I-D.karagiannis-problem-statement-geonetworking]. This BoF was held within the Internet area and concerns geolocation at the Internet layer.

As a result of the geonet BoF, a new working group known as 'Intelligent Transportation Systems (its)' is undergoing chartering activities. It is expected that IPv6GEO will be closely related to the its charter.

9. Implementation Status

A prototype implementation has been developed and tested, but not yet available for public release. The prototype implementation uses the Option Type value reserved for experimentation [RFC3692].

10. Contributors

The authors greatly appreciate the efforts of Jin Fang, who jointly developed the IPv6 GEO message format and was the primary author of the prototype implementation. We wish Jin the best of success in his future endeavors.

11. Acknowledgments

The following individuals are acknowledged for helpful comments and suggestions: Jeff Ahrenholz, Kerry Hu.

12. References

12.1. Normative References

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<http://www.rfc-editor.org/info/rfc791>>.
- [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>>.
- [RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, DOI 10.17487/RFC3692, January 2004, <<http://www.rfc-editor.org/info/rfc3692>>.
- [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>>.

12.2. Informative References

- [I-D.ietf-opsec-ipv6-eh-filtering]
Gont, F., LIU, S., and R. Bonica, "Recommendations on Filtering of IPv6 Packets Containing IPv6 Extension Headers", draft-ietf-opsec-ipv6-eh-filtering-01 (work in progress), July 2016.
- [I-D.karagiannis-problem-statement-geonetworking]
Karagiannis, G., Heijenk, G., Festag, A., Petrescu, A., and A. Chaiken, "Internet-wide Geo-networking Problem Statement", draft-karagiannis-problem-statement-geonetworking-01 (work in progress), November 2013.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<http://www.rfc-editor.org/info/rfc4301>>.
- [RFC6280] Barnes, R., Lepinski, M., Cooper, A., Morris, J., Tschofenig, H., and H. Schulzrinne, "An Architecture for Location and Location Privacy in Internet Applications", BCP 160, RFC 6280, DOI 10.17487/RFC6280, July 2011, <<http://www.rfc-editor.org/info/rfc6280>>.
- [WGS-84] Wikipedia, W., "World Geodetic System (http://en.wikipedia.org/wiki/World_Geodetic_System)", November 2013.
- [WGS-ENCODE]
Dupuis, L., "Efficient WGS84 (aka GPS) Coordinates Compression (<http://www.dupuis.me/node/35>)", August 2013.

Authors' Addresses

Brian Skeen
Boeing Phantom Works
P.O. Box 3707
Seattle, WA 98124
USA

Email: brian.l.skeen@boeing.com

Edwin King
Boeing EO&T IT
P.O. Box 3707
Seattle, WA 98124
USA

Email: edwin.e.king@boeing.com

Fred L. Templin (editor)
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
USA

Email: fltemplin@acm.org

6MAN
Internet-Draft
Intended status: Standards Track
Expires: February 24, 2015

P. Thubert, Ed.
Cisco
August 25, 2014

The IPv6 Flow Label within a LLN domain
draft-thubert-6man-flow-label-for-rpl-05

Abstract

This document presents how the Flow Label can be used inside a LLN domain such as a RPL domain or an ISA100.11a D-subnet, and provides updated rules for a domain Border Router to set and reset the Flow Label when forwarding between inside the domain and the larger Internet in both direction. Rules for routers inside the domain are also provided.

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 24, 2015.

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. Requirements for LLN Flows	3
4. On Compatibility With Existing Standards	4
5. Updated Rules	5
6. Security Considerations	6
7. IANA Considerations	6
8. Acknowledgements	6
9. References	7
9.1. Normative References	7
9.2. Informative References	7
Author's Address	8

1. Introduction

The design of Lowpower Lossy Networks (LLNs) is generally focussed on saving energy, which is typically the most constrained resource of all. Other classical constraints, such as memory capacity, frame size, as well as the duty cycling of the LLN devices, derive from that primary concern.

In isolated devices, energy is typically available from batteries that are expected to last for years, or scavenged from the environment in very limited quantities. Any protocol that is intended for use in LLNs must be designed with the primary concern of saving energy as a strict requirement.

The IEEE802.15.4 [IEEE802154] was designed to offer the Physical (PHY) and Medium Access Control (MAC) layers for low-cost, low-speed, low-power Wireless Personal Area Networks (WPANs), which are a wireless form of LLNs.

With the traditional IEEE802.15.4 PHY, frames are limited to 127 octets. In order to adapt IPv6 [RFC2460] over IEEE802.15.4, 6LoWPAN [RFC4944] introduced a fragmentation mechanism under IP, which in turn causes even more energy spending and other issues as discussed in LLN Fragment Forwarding and Recovery [I-D.thubert-6lo-forwarding-fragments].

The IEEE802.15.4e Task Group further defined the TimeSlotted Channel Hopping [I-D.ietf-6tisch-tsch] (TSCH) mode of operation as an update to the MAC specification in order to address Time Sensitive applications.

The 6TiSCH architecture [I-D.ietf-6tisch-architecture] specifies the operation of IPv6 over IEEE802.15.4e TSCH networks attached and synchronized by backbone routers. 6TiSCH was created to simplify the adoption of IETF technology by other Standard Defining Organizations (SDOs), in particular in the Industrial Automation space, which already relies on variations of IEEE802.15.4e TSCH for Wireless Sensor Networking.

The ISA100.11a [ISA100.11a] specification provides an example of such an industrial WSN standard, using a precursor to IEEE802.15.4e over the classical IEEE802.14.5 PHY. In that case, after security is applied, roughly 80 octets are available per frame for IP and Payload. In order to 1) avoid fragmentation and 2) conserve energy, the ISA100 WG in charge of that specification did scrutinize the use of every bit in the frame and rejected any perceived waste.

The challenge to obtain the adoption of IPv6 in the original standard was thus to save all possible bits in the frames, including the UDP checksum which was an interesting discussion on its own. This work was actually one of the roots for the 6LoWPAN Header Compression [RFC6282] work, which goes down to the individual bits to save space in the frames for actual data, and allowed ISA100.11a to adopt IPv6.

ISA100.11a (now IEC62734) uses IPv6 over UDP, and conforms to a number of other IETF RFCs including the IPv6 Flow Label Specification [RFC3697] that was the reference at the time the standard was elaborated, but fails to conform to the newer IPv6 Flow Label Specification [RFC6437] that obsoleted it.

The bone of contention is the use of the Flow Label as an index called a contract ID, and the capability for the Backbone Router, that is the Border Router of a ISA100.11a WSN (also called a D-subnet), to modify the Flow Label. There is work at ROLL that indicates that RPL nodes may benefit from similar abilities to also transport flow-related information in the Flow Label.

This document adds an exception to the rules in [RFC6437], for application within a well-defined LLN domain, whereby the Border Routers would be in a position to ensure that from an external viewpoint, the domain complies to the new Flow Label specification even though the internal use of the Flow Label does not.

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 [RFC2119].

This document uses Terminology defined in Terminology in Low power And Lossy Networks [RFC7102], as well as [RFC6550] and [RFC6553].

3. Requirements for LLN Flows

In Industrial Automation and Control Systems (IACS) [RFC5673], a packet loss is usually acceptable but jitter and latency must be strictly controlled as they can play a critical role in the interpretation of the measured information. Sensory systems are often distributed, and the control information can in fact be originated from multiple sources and aggregated. In such cases, related packets from multiple sources should not be load-balanced

along their path in the Internet.

In a typical LLN application, the bulk of the traffic consists of small chunks of data (in the order few bytes to a few tens of bytes) at a time. 4Hz is a typical loop frequency in Process Control, though it can be a lot slower than that in, say, environmental monitoring. The granularity of traffic from a single source is too small to make a lot of sense in load balancing application.

As a result, it can be a requirement for related measurements from multiple sources to be treated as a single flow following a same path over the Internet so as to experience similar jitter and latency. The traditional tuple of source, destination and ports might then not be the proper indication to isolate a consistent flow. On the other hand, the flow integrity can be preserved in a simple manner if the setting of the Flow Label in the IPv6 header of packets outgoing a LLN domain, is centralized to the Border Router, such as the root of a RPL DODAG structure, or an ISA100.11a Backbone Router, as opposed to distributed across the actual sources.

Considering that the goal for setting the Flow Label as prescribed in the IPv6 Flow Label Specification [RFC6437] is to improve load balancing in the core of the Internet, it is unlikely that LLN devices will consume energy to generate and then transmit a Flow Label to serve outside interests and the Flow Label is generally left to zero so as to be elided in the 6LoWPAN [RFC6282] compression. So in a general manner the interests of the core are better served if the RPL roots systematically rewrite the flow label rather than if they never do.

For packets coming into the RPL domain from the Internet, the value for setting the Flow Label as prescribed in [RFC6437] is consumed once the packet has traversed the core and reaches the LLN. Then again, there is little value but a high cost for the LLN in spending 20 bits to transport a Flow Label, that was set by a peer or a router in the Internet, over the constrained network to a destination node that has no use of it.

On a PHY layer with super-short frames such as IEEE802.15.4, compliance with those rules will simply not happen, and the rules will become an bone of contention for IPv6 adoption at a time where great progress is happening towards that goal, as illustrated by the activity at 6lo on multiple LLN Link-layers.

4. On Compatibility With Existing Standards

All the packets from all the nodes in a same DODAG that are leaving a RPL domain towards the Internet will transit via a same RPL root. The RPL root segregates the Internet and the RPL domain, which enables the capability to reuse the Flow Label within the RPL domain. The ISA100.11a Backbone Router plays a similar role and interfaces an ISA100.11a WSN D-subnet with a larger IPv6 network.

This specification enables the operation of resetting or reusing the IPv6 Flow Label at the border of a LLN domain. This is a deviation from the IPv6 Flow Label Specification [RFC6437], in that the LLN border router is neither the source nor the first hop router that sets the final Flow Label for use outside the LLN domain.

But if we consider the whole RPL domain as a large virtual host from the standpoint of the rest of the Internet, the interests that lead to [RFC6437], and in particular load balancing in the core of the Internet, are probably better served if the root guarantees that the Flow Label is set in a compliant fashion than if we rely on each individual sensor that may not use it at all, or use it slightly differently such as done in ISA100.11a.

Additionally, LLN flows can be compound flows aggregating information from multiple sources. The Border Router is an ideal place to rewrite the Flow Label to a same value for a same flow across multiple sources, ensuring compliance with the rules defined by [RFC6437] for use outside of the RPL domain and in particular in the core of the Internet.

This document specifies how the Flow Label can be reused within a LLN domain such as a RPL domain and an ISA100.11a D-subnet, in which a Border Router delineates the limit of the domain and may rewrite the Flow Label on all packets. In a RPL domain, it will become acceptable to use the Flow Label as replacement to the RPL option, though whether that operation gets standardized is left to be discussed. That use of the Flow Label within a RPL domain would be an instance of the stateful scenarios as discussed in [RFC6437] where the flow state in the node is indexed by the RPLInstanceID that identifies the routing topology. ISA100.11a would be another instance where the 16bit Contract ID in the Flow Label identifies a state in a node that is specific to a particular flow.

5. Updated Rules

This specification applies to a constrained LLN domain that forms a stub and is connected to the Internet by and only by its Border Routers. In the case of a RPL domain, the RPL root is such a bottleneck for all the traffic between the Internet and the Destination-Oriented Directed Acyclic Graph (DODAG) that it serves. This specification also covers other LLN domains with the same properties of having strict constraints in energy and/or frame size, such as an ISA100.11a [ISA100.11a] Industrial Wireless Sensor Network, but does not generalize to any arbitrary domain. This updates the IPv6 Flow Label Specification [RFC6437], which does not allow any specific rule in any particular domain, and updates it only in the context of constrained LLN domains.

In that context, a LLN domain Border Router MAY rewrite the Flow Label of all packets entering or leaving the RPL domain in both directions, from and towards the Internet, regardless of its original setting. For the limited context of a constrained LLN domain, this updates the IPv6 Flow Label Specification [RFC6437] which stipulates that once it is set, the Flow Label is left unchanged; but the RFC also indicates a violation to the rule can be accepted for compelling reasons related to security. This specification adds that energy-saving is another compelling reason for a violation to the aforementioned rule, though applicable only inside a constrained LLN.

In particular, the Border Router of a LLN domain MAY set the Flow Label of IPv6 packets that exit the LLN domain. It SHOULD do it if the LLN domain operations do not conform [RFC6437], and if it does modify the Flow Label, then it MUST do it in a manner that conforms [RFC6437] from the perspective of a Node outside the LLN.

It results that a Node in a constrained LLN domain MUST NOT assume that the setting of the Flow Label will be preserved end-to-end, and that an intermediate router inside a constrained LLN MAY alter a non-zero Flow Label between the source in the LLN and the LLN Border Router. This does not modify the expectations on end Nodes but extends the updated rules from [RFC6437] to arbitrary routers in the LLN.

For instance, a RPL root MAY reset the Flow Label of IPv6 packets entering the RPL domain to zero for an optimal Header Compression by 6LoWPAN [RFC6282]. A RPL root MAY also reuse the Flow Label towards the LLN for other purposes, such as to carry the RPL Information [RFC6553]. An ISA100.11s Backbone Router MAY reuse the Flow Label to carry local flow information, such as the Contract ID specified in ISA100.11a [ISA100.11a].

6. Security Considerations

Because the flow label is not protected by IPSec, it is expected that Layer-2 security is deployed in the LLN where is specification is applied. This is the actual best practice in LLNs, which serves in particular to avoid forwarding of untrusted packets over the constrained network.

The specification insists that the LLN Node should not expect that the Flow Label is conserved end-to-end and rather reduces the risk of misinterpretation in case of a rewrite by a router in the middle.

7. IANA Considerations

No IANA action is required for this specification.

8. Acknowledgements

The author wishes to thank Brian Carpenter for his in-depth review and constructive approach to the problem resolution.

9. References

9.1. Normative References

- [IEEE802154]
IEEE standard for Information Technology, "IEEE std. 802.15.4, Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks", June 2011.
- [ISA100.11a]
ISA/ANSI, "Wireless Systems for Industrial Automation: Process Control and Related Applications - ISA100.11a-2011 - IEC 62734", 2011, <<http://www.isa.org/Community/SP100WirelessSystemsforAutomation>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S.E. and R.M. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC3697] Rajahalme, J., Conta, A., Carpenter, B. and S. Deering, "IPv6 Flow Label Specification", RFC 3697, March 2004.
- [RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, September 2011.
- [RFC6437] Amante, S., Carpenter, B., Jiang, S. and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, November 2011.
- [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.
- [RFC6552] Thubert, P., "Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6552, March 2012.
- [RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, March 2012.

9.2. Informative References

- [I-D.ietf-6tisch-architecture]
Thubert, P., Watteyne, T. and R. Assimiti, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", Internet-Draft draft-ietf-6tisch-architecture-01, February 2014.

[I-D.ietf-6tisch-tsch]

Watteyne, T., Palattella, M. and L. Grieco, "Using IEEE802.15.4e TSCH in an LLN context: Overview, Problem Statement and Goals", Internet-Draft draft-ietf-6tisch-tsch-00, November 2013.

[I-D.thubert-6lo-forwarding-fragments]

Thubert, P. and J. Hui, "LLN Fragment Forwarding and Recovery", Internet-Draft draft-thubert-6lo-forwarding-fragments-01, February 2014.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J. and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.

[RFC5673] Pister, K., Thubert, P., Dwars, S. and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks", RFC 5673, October 2009.

[RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and Lossy Networks", RFC 7102, January 2014.

Author's Address

Pascal Thubert, editor
Cisco Systems
Village d'Entreprises Green Side
400, Avenue de Roumanille
Batiment T3
Biot - Sophia Antipolis, 06410
FRANCE

Phone: +33 4 97 23 26 34
Email: pthubert@cisco.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 4, 2015

E. Vyncke
S. Previdi
Cisco Systems, Inc.
B. Field
Comcast
I. Leung
Rogers Communications
July 3, 2014

IPv6 Segment Routing Header (SRH) Security Considerations
draft-vyncke-6man-segment-routing-security-00

Abstract

Segment Routing (SR) allows a node to steer a packet through a controlled set of instructions, called segments, by prepending a SR header to the packet. A segment can represent any instruction, topological or service-based. SR allows to enforce a flow through any path (topological, or application/service based) while maintaining per-flow state only at the ingress node to the SR domain.

Segment Routing can be applied to the IPv6 data plane with the addition of a new type of Routing Extension Header. This draft analyses the security aspects the Segment Routing Extension Header Type and how it is used by SR capable nodes to deliver a secure service.

Requirements Language

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

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 January 4, 2015.

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. Segment Routing Documents	2
2. Introduction	3
3. Threat model	3
3.1. Source routing threat	3
3.2. Applicability of RFC 5095 to SRH	4
3.3. Service stealing threat	4
3.4. Topology disclosure	4
4. Security fields in SRH	5
4.1. Selecting a hash algorithm	6
4.2. Performance impact of HMAC	6
4.3. Pre-shared key management	7
5. Deployment Models	7
5.1. Nodes within the SR domain	7
5.2. Nodes outside of the SR domain	7
5.3. SR path exposure	8
6. IANA Considerations	9
7. Manageability Considerations	9
8. Security Considerations	9
9. Acknowledgements	9
10. References	9
10.1. Normative References	9
10.2. Informative References	9
Authors' Addresses	10

1. Segment Routing Documents

Segment Routing terminology is defined in [I-D.filsfils-spring-segment-routing].

Segment Routing use cases are described in [I-D.filsfils-spring-segment-routing-use-cases].

Segment Routing IPv6 use cases are described in [I-D.ietf-spring-ipv6-use-cases].

Segment Routing protocol extensions are defined in [I-D.ietf-isis-segment-routing-extensions], and [I-D.psenak-ospf-segment-routing-ospfv3-extension].

2. Introduction

This section analyses the security threat model as well as the security issues and proposed solutions related to the new routing header for segment routing.

The SRH is simply another version of the routing header as described in [RFC2460] and is:

- o inserted when entering the segment routing domain which could be done by a node or by a router;
- o inspected and acted upon when reaching the destination address of the IP header.

Routers on the path that simply forward an IPv6 packet (i.e. the IPv6 destination address is none of theirs) will never inspect and process the SRH. Routers whose one interface IPv6 address equals the destination address field of the SRH will have to parse the SRH and, if supported and if the local configuration allows it, will act on the SRH.

3. Threat model

3.1. Source routing threat

Using a SRH, which is basically source routing, has some well-known security issues as described in [RFC4942] section 2.1.1 and [RFC5095]:

- o amplification attacks: where a packet could be forged in such a way to cause looping among a set of SR-enabled routers causing unnecessary traffic, hence a denial of service against bandwidth;
- o reflection attack: where a hacker could force an intermediate node to appear as the immediate attacker, hence hiding the real attacker from naive forensic;

- o bypass attack: where an intermediate node could be used as a stepping stone (for example in a DMZ) to attack another host (for example in the datacenter or any back-end server).

These security issues did lead to obsoleting the routing-header type 0, RH-0, with [RFC5095] because:

- o it was assumed to be inspected and acted upon by default by each and every router on the Internet;
- o it contained multiple segments in the payload.

Therefore, if intermediate nodes ONLY act on valid and authorized SRH, then there is no security threat similar to RH-0.

3.2. Applicability of RFC 5095 to SRH

In the segment routing architecture described in [I-D.filsfils-spring-segment-routing] there are basically two kinds of nodes (routers and hosts):

- o nodes within the segment routing domain, which is within one single administrative domain, i.e., where all nodes are trusted anyway else the damage caused by those nodes could be worse than amplification attacks: traffic interception and man-in-the-middle attacks, more server DoS by dropping packets, and so on.
- o Nodes outside of the segment routing domain, which is outside of the administrative segment routing domain hence they cannot be trusted because there is no physical security for those nodes, i.e., they can be replaced by hostile nodes or can be coerced in wrong behaviors.

3.3. Service stealing threat

SR is used for added value services, there is also a need to prevent non-participating nodes to use those services; this is called 'service stealing prevention'.

3.4. Topology disclosure

The SRH also contains all IPv6 addresses of intermediate SR-nodes, this obviously reveals those addresses to the potentially hostile attackers if those attackers are on the path.

4. Security fields in SRH

This section summarizes the use of specific fields in the SRH; they are integral part of [I-D.previdi-6man-segment-routing-header] and they are again described here for reader's sake.

The security-related fields in SRH are:

- o HMAC Key-id, 8 bits wide, if HMAC key-id is null, then there is no HMAC field;
- o HMAC, 256 bits wide.

The HMAC field is the output of the hash of the concatenation of:

- o the source IPv6 address;
- o last segment field, an octet whose bit-0 is the clean-up bit flag and others are 0, HMAC key-id, all addresses in the Segment List;
- o a pre-shared secret between SR nodes in the SR domain (routers, controllers, ...);
- o if required by the hash algorithm a pad field filled with 0.

The purpose of the HMAC field is to verify the validity, the integrity and the authorization of the SRH itself. If an outsider of the SR domain does not have access to a current pre-shared secret, then it cannot compute the right HMAC field and the first SR router on the path processing the SRH and configured to check the validity of the HMAC will simply reject the packet.

The HMAC field is located at the end of the SRH simply because only the router on the ingress of the SR domain needs to process it, then all other SR nodes can ignore it (based on local policy) because they can trust the upstream router. This is to speed up forwarding operations because some hardware platforms can only parse in hardware so many bytes.

The HMAC Key-id field allows for the simultaneous existence of several hash algorithms (SHA-256, SHA3-256 ... or future ones) as well as pre-shared keys. This allows for pre-shared key roll-over when two pre-shared keys are supported for a while when all SR nodes converged to a fresher pre-shared key. The HMAC key-id is opaque, i.e., it has no syntax except as an index to the right combination of pre-shared key and hash algorithm. It also allows for interoperation among different SR domains if allowed by local policy.

When a specific SRH is linked to a time-related service (such as turbo-QoS for a 1-hour period) where the DA, SID are identical, then it is important to refresh the shared-secret frequently as the HMAC validity period expires only when the HMAC key-id and its associated shared-secret expires. How HMAC key-id and pre-shared secret are synchronized between participating nodes in the SR domain is outside of the scope of this document ([RFC6407] GDOI could be a basis).

4.1. Selecting a hash algorithm

The HMAC field in the SRH is 256 bit wide. Therefore, the HMAC MUST be based on a hash function whose output is at least 256 bits. If the output of the hash function is 256, then this output is simply inserted in the HMAC field. If the output of the hash function is larger than 256 bits, then the output value is truncated to 256 by taking the least-significant 256 bits and inserting them in the HMAC field.

SRH implementations can support multiple hash functions but MUST implement SHA-2 [FIPS180-4] in its SHA-256 variant.

4.2. Performance impact of HMAC

While adding a HMAC to each and every SR packet increases the security, it has a performance impact. Nevertheless, it must be noted that:

- o the HMAC field is used only when SRH is inserted by a device (such as a home set-up box) which is outside of the segment routing domain. If the SRH is added by a router in the trusted segment routing domain, then, there is no need for a HMAC field, hence no performance impact.
- o when present, the HMAC field MUST only be checked and validated by the first router of the segment routing domain, this router is named 'validating router'. Downstream routers SHOULD NOT inspect the HMAC field.
- o this validating router can also have a cache of <IPv6 header + SRH, HMAC field value> to improve the performance. It is not the same use case as in IPsec where HMAC value was unique per packet, in SRH, the HMAC value is unique per flow.
- o Last point, hash functions such as SHA-2 have been optimized for security and performance and there are multiple implementations with good performance.

With the above points in mind, the performance impact of using HMAC is minimized.

4.3. Pre-shared key management

The field HMAC key-id allows for:

- o key roll-over: when there is a need to change the key (the hash pre-shared secret), then multiple pre-shared keys can be used simultaneously. The validating routing can have a table of <key-id, pre-shared secret> for the current and future keys.
- o different algorithm: by extending the previous table to <key-id, hash function, pre-shared secret>, the validating router can also support simultaneously several hash algorithm (see section Section 4.1)

The pre-shared secret distribution can be done:

- o in the configuration of the validating routers, either by static configuration or any SDN oriented approach;
- o dynamically using a trusted key distribution such as [RFC6407]

NOTE: this section needs more work but the intent is NOT to define yet-another-key-distribution-protocol.

5. Deployment Models

5.1. Nodes within the SR domain

Those nodes can be trusted to generate SRH and to process SRH received on interfaces that are part of the SR domain. These nodes MUST drop all packets received on an interface that is not part of the SR domain and containing a SRH whose HMAC field cannot be validated by local policies. This includes obviously packet with a SRH generated by a non-cooperative SR domain.

If the validation fails, then these packets MUST be dropped, ICMP error messages (parameter problem) SHOULD be generated (but rate limited) and SHOULD be logged.

5.2. Nodes outside of the SR domain

Nodes outside of the SR domain cannot be trusted for physical security; hence, they need to request by some means (outside of the scope of this document) a complete SRH for each new connection (i.e.

new destination address). The SRH MUST include a HMAC key-id and HMAC field which is computed correctly (see Section 4).

When an outside node sends a packet with an SRH and towards a SR ingress node, the packet MUST contain the HMAC key-id and HMAC field and the SR ingress node MUST be the destination address.

The ingress SR router, i.e., the router with an interface address equals to the destination address, MUST verify the HMAC field with respect to the HMAC key-id.

If the validation is successful, then the packet is simply forwarded as usual for a SR packet. As long as the packet travels within the SR domain, no further HMAC check needs to be done. Subsequent routers in the SR domain MAY verify the HMAC field when they process the SRH (i.e. when they are the destination).

If the validation fails, then this packet MUST be dropped, an ICMP error message (parameter problem) SHOULD be generated (but rate limited) and SHOULD be logged.

5.3. SR path exposure

As the intermediate SR nodes addresses appears in the SRH, if this SRH is visible to an outside then he/she could reuse this knowledge to launch an attack on the intermediate SR nodes or get some insider knowledge on the topology. This is especially applicable when the path between the source node and the first SR-node in the domain is on the public Internet.

The first remark is to state that 'security by obscurity' is never enough; in other words, the security policy of the SR domain MUST assume that the internal topology and addressing is known by the attacker. A simple traceroute will also give the same information (with even more information as all intermediate nodes between SID will also be exposed). IPsec Encapsulating Security Payload (RFC 4303) cannot be used to protect the SRH as per RFC 4303 the ESP header must appear after any routing header (including SRH).

To prevent a user to leverage the gained knowledge by intercepting SRH, it is recommended to apply an infrastructure Access Control List (iACL) at the edge of the SR domain. This iACL will drop all packets from outside the SR-domain whose destination is any address of any router inside the domain. This security policy should be tuned for local operations.

6. IANA Considerations

There are no IANA request or impact in this document.

7. Manageability Considerations

TBD

8. Security Considerations

This document describes the security mechanisms applied to the Segment Routing Header defined in [I-D.previdi-6man-segment-routing-header]

9. Acknowledgements

The authors would like to thank Dave Barach for his contribution to this document.

10. References

10.1. Normative References

- [FIPS180-4] National Institute of Standards and Technology, "FIPS 180-4 Secure Hash Standard (SHS)", March 2012, <<http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, December 2007.
- [RFC6407] Weis, B., Rowles, S., and T. Hardjono, "The Group Domain of Interpretation", RFC 6407, October 2011.

10.2. Informative References

[I-D.filsfils-spring-segment-routing]

Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-filsfils-spring-segment-routing-03 (work in progress), June 2014.

[I-D.filsfils-spring-segment-routing-use-cases]

Filsfils, C., Francois, P., Previdi, S., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., Kini, S., and E. Crabbe, "Segment Routing Use Cases", draft-filsfils-spring-segment-routing-use-cases-00 (work in progress), March 2014.

[I-D.ietf-isis-segment-routing-extensions]

Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-02 (work in progress), June 2014.

[I-D.ietf-spring-ipv6-use-cases]

Brzozowski, J., Leddy, J., Leung, I., Previdi, S., Townsley, W., Martin, C., Filsfils, C., and R. Maglione, "IPv6 SPRING Use Cases", draft-ietf-spring-ipv6-use-cases-00 (work in progress), May 2014.

[I-D.previdi-6man-segment-routing-header]

Previdi, S., Filsfils, C., Field, B., and I. Leung, "IPv6 Segment Routing Header (SRH)", draft-previdi-6man-segment-routing-header-01 (work in progress), June 2014.

[I-D.psenak-ospf-segment-routing-ospfv3-extension]

Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPFv3 Extensions for Segment Routing", draft-psenak-ospf-segment-routing-ospfv3-extension-02 (work in progress), July 2014.

[RFC4942] Davies, E., Krishnan, S., and P. Savola, "IPv6 Transition/Co-existence Security Considerations", RFC 4942, September 2007.

Authors' Addresses

Eric Vyncke
Cisco Systems, Inc.
De Kleetlaan 6A
Diegem 1831
Belgium

Email: evyncke@cisco.com

Stefano Previdi
Cisco Systems, Inc.
Via Del Serafico, 200
Rome 00142
Italy

Email: sprevidi@cisco.com

Brian Field
Comcast
4100 East Dry Creek Road
Centennial, CO 80122
US

Email: Brian_Field@comcast.com

Ida Leung
Rogers Communications
8200 Dixie Road
Brampton, ON L6T 0C1
CA

Email: Ida.Leung@rci.rogers.com

6man Group
Internet-Draft
Intended status: Standards Track
Expires: August 29, 2015

E. Vyncke, Ed.
S. Previdi
Cisco Systems, Inc.
D. Lebrun
Universite Catholique de Louvain
February 25, 2015

IPv6 Segment Routing Security Considerations
draft-vyncke-6man-segment-routing-security-02

Abstract

Segment Routing (SR) allows a node to steer a packet through a controlled set of instructions, called segments, by prepending a SR header to the packet. A segment can represent any instruction, topological or service-based. SR allows to enforce a flow through any path (topological, or application/service based) while maintaining per-flow state only at the ingress node to the SR domain.

Segment Routing can be applied to the IPv6 data plane with the addition of a new type of Routing Extension Header. This document analyzes the security aspects of the Segment Routing Extension Header (SRH) and how it is used by SR capable nodes to deliver a secure service.

Requirements Language

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

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 29, 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. Segment Routing Documents	3
2. Threat model	3
2.1. Source routing threats	4
2.2. Applicability of RFC 5095 to SRH	4
2.3. Service stealing threat	5
2.4. Topology disclosure	5
2.5. ICMP Generation	5
3. Security fields in SRH	6
3.1. Selecting a hash algorithm	7
3.2. Performance impact of HMAC	7
3.3. Pre-shared key management	8
4. Deployment Models	9
4.1. Nodes within the SR domain	9
4.2. Nodes outside of the SR domain	9
4.3. SR path exposure	10
4.4. Impact of BCP-38	10
5. IANA Considerations	10
6. Manageability Considerations	11
7. Security Considerations	11
8. Acknowledgements	11
9. References	11
9.1. Normative References	11
9.2. Informative References	11
Authors' Addresses	13

1. Introduction

This document analyzes the security threat model, the security issues and proposed solutions related to the new routing header for segment routing with an IPv6 data plane.

The Segment Routing Header (SRH) is simply another type of the routing header as described in RFC 2460 [RFC2460] and is:

- o inserted by a SR edge router when entering the segment routing domain or by the originating host itself. The source host can even be outside the SR domain;
- o inspected and acted upon when reaching the destination address of the IP header per RFC 2460 [RFC2460].

Per RFC2460 [RFC2460], routers on the path that simply forward an IPv6 packet (i.e. the IPv6 destination address is none of theirs) will never inspect and process the content of SRH. Routers whose one interface IPv6 address equals the destination address field of the IPv6 packet MUST to parse the SRH and, if supported and if the local configuration allows it, MUST act accordingly to the SRH content.

According to RFC2460 [RFC2460], the default behavior of a non SR-capable router upon receipt of an IPv6 packet with SRH destined to an address of its, is to:

- o ignore the SRH completely if the Segment Left field is 0 and proceed to process the next header in the IPv6 packet;
- o discard the IPv6 packet if Segment Left field is greater than 0, it MAY send a Parameter Problem ICMP message back to the Source Address.

1.1. Segment Routing Documents

Segment Routing terminology is defined in [I-D.ietf-spring-segment-routing] and in [I-D.ietf-spring-problem-statement]. Segment Routing use cases are described in [I-D.filsfils-spring-segment-routing-use-cases]. Segment Routing protocol extensions are defined in [I-D.ietf-isis-segment-routing-extensions], and [I-D.ietf-ospf-ospfv3-segment-routing-extensions].

Segment Routing IPv6 use cases are described in [I-D.ietf-spring-ipv6-use-cases]. And the IPv6 Segment Routing header is described in [I-D.previdi-6man-segment-routing-header].

2. Threat model

2.1. Source routing threats

Using a SRH is similar to source routing, therefore it has some well-known security issues as described in RFC4942 [RFC4942] section 2.1.1 and RFC5095 [RFC5095]:

- o amplification attacks: where a packet could be forged in such a way to cause looping among a set of SR-enabled routers causing unnecessary traffic, hence a Denial of Service (DoS) against bandwidth;
- o reflection attack: where a hacker could force an intermediate node to appear as the immediate attacker, hence hiding the real attacker from naive forensic;
- o bypass attack: where an intermediate node could be used as a stepping stone (for example in a De-Militarized Zone) to attack another host (for example in the datacenter or any back-end server).

2.2. Applicability of RFC 5095 to SRH

First of all, the reader must remember this specific part of section 1 of RFC5095 [RFC5095], "A side effect is that this also eliminates benign RH0 use-cases; however, such applications may be facilitated by future Routing Header specifications.". In short, it is not forbidden to create new secure type of Routing Header; for example, RFC 6554 (RPL) [RFC6554] also creates a new Routing Header type for a specific application confined in a single network.

In the segment routing architecture described in [I-D.ietf-spring-segment-routing] there are basically two kinds of nodes (routers and hosts):

- o nodes within the SR domain, which is within one single administrative domain, i.e., where all nodes are trusted anyway else the damage caused by those nodes could be worse than amplification attacks: traffic interception, man-in-the-middle attacks, more server DoS by dropping packets, and so on.
- o nodes outside of the SR domain, which is outside of the administrative segment routing domain hence they cannot be trusted because there is no physical security for those nodes, i.e., they can be replaced by hostile nodes or can be coerced in wrong behaviors.

The main use case for SR consists of the single administrative domain where only trusted nodes with SR enabled and configured participate

in SR: this is the same model as in RFC6554 [RFC6554]. All non-trusted nodes do not participate as either SR processing is not enabled by default or because they only process SRH from nodes within their domain.

Moreover, all SR nodes ignore SRH created by outsiders based on topology information (received on a peering or internal interface) or on presence and validity of the HMAC field. Therefore, if intermediate nodes ONLY act on valid and authorized SRH (such as within a single administrative domain), then there is no security threat similar to RH-0. Hence, the RFC 5095 [RFC5095] attacks are not applicable.

2.3. Service stealing threat

Segment routing is used for added value services, there is also a need to prevent non-participating nodes to use those services; this is called 'service stealing prevention'.

2.4. Topology disclosure

The SRH may also contains IPv6 addresses of some intermediate SR-nodes in the path towards the destination, this obviously reveals those addresses to the potentially hostile attackers if those attackers are able to intercept packets containing SRH. On the other hand, if the attacker can do a traceroute whose probes will be forwarded along the SR path, then there is little learned by intercepting the SRH itself. Also the clean-bit of SRH can help by removing the SRH before forwarding the packet to potentially a non-trusted part of the network.

2.5. ICMP Generation

Per section 4.4 of RFC2460 [RFC2460], when destination nodes (i.e. where the destination address is one of theirs) receive a Routing Header with unsupported Routing Type, the required behavior is:

- o If Segments Left is zero, the node must ignore the Routing header and proceed to process the next header in the packet.
- o If Segments Left is non-zero, the node must discard the packet and send an ICMP Parameter Problem, Code 0, message to the packet's Source Address, pointing to the unrecognized Routing Type.

This required behavior could be used by an attacker to force the generation of ICMP message by any node. The attacker could send packets with SRH (with Segment Left set to 0) destined to a node not supporting SRH. Per RFC2460 [RFC2460], the destination node could

generate an ICMP message, causing a local CPU utilization and if the source of the offending packet with SRH was spoofed could lead to a reflection attack without any amplification.

It must be noted that this is a required behavior for any unsupported Routing Type and not limited to SRH packets. So, it is not specific to SRH and the usual rate limiting for ICMP generation is required anyway for any IPv6 implementation and has been implemented and deployed for many years.

3. Security fields in SRH

This section summarizes the use of specific fields in the SRH; they are integral part of [I-D.previdi-6man-segment-routing-header] and they are again described here for reader's sake. They are based on a key-hashed message authentication code (HMAC).

The security-related fields in SRH are:

- o HMAC Key-id, 8 bits wide;
- o HMAC, 256 bits wide (optional, exists only if HMAC Key-id is not 0).

The HMAC field is the output of the HMAC computation (per RFC 2104 [RFC2104]) using a pre-shared key identified by HMAC Key-id and of the text which consists of the concatenation of:

- o the source IPv6 address;
- o First Segment field;
- o an octet whose bit-0 is the clean-up bit flag and others are 0;
- o HMAC Key-id;
- o all addresses in the Segment List.

The purpose of the HMAC field is to verify the validity, the integrity and the authorization of the SRH itself. If an outsider of the SR domain does not have access to a current pre-shared secret, then it cannot compute the right HMAC field and the first SR router on the path processing the SRH and configured to check the validity of the HMAC will simply reject the packet.

The HMAC field is located at the end of the SRH simply because only the router on the ingress of the SR domain needs to process it, then all other SR nodes can ignore it (based on local policy) because they

trust the upstream router. This is to speed up forwarding operations because SR routers which do not validate the SRH do not need to parse the SRH until the end.

The HMAC Key-id field allows for the simultaneous existence of several hash algorithms (SHA-256, SHA3-256 ... or future ones) as well as pre-shared keys. This allows for pre-shared key roll-over when two pre-shared keys are supported for a while when all SR nodes converged to a fresher pre-shared key. The HMAC Key-id field is opaque, i.e., it has neither syntax nor semantic except as an index to the right combination of pre-shared key and hash algorithm and except that a value of 0 means that there is no HMAC field. It could also allow for interoperation among different SR domains if allowed by local policy and assuming a collision-free Key Id allocation.

When a specific SRH is linked to a time-related service (such as turbo-QoS for a 1-hour period) where the DA, Segment ID (SID) are identical, then it is important to refresh the shared-secret frequently as the HMAC validity period expires only when the HMAC Key-id and its associated shared-secret expires.

3.1. Selecting a hash algorithm

The HMAC field in the SRH is 256 bit wide. Therefore, the HMAC MUST be based on a hash function whose output is at least 256 bits. If the output of the hash function is 256, then this output is simply inserted in the HMAC field. If the output of the hash function is larger than 256 bits, then the output value is truncated to 256 by taking the least-significant 256 bits and inserting them in the HMAC field.

SRH implementations can support multiple hash functions but MUST implement SHA-2 [FIPS180-4] in its SHA-256 variant.

NOTE: SHA-1 is currently used by some early implementations used for quick interoperations testing, the 160-bit hash value must then be right-hand padded with 96 bits set to 0. The authors understand that this is not secure but is ok for limited tests.

3.2. Performance impact of HMAC

While adding a HMAC to each and every SR packet increases the security, it has a performance impact. Nevertheless, it must be noted that:

- o the HMAC field is used only when SRH is inserted by a device (such as a home set-up box) which is outside of the segment routing domain. If the SRH is added by a router in the trusted segment

routing domain, then, there is no need for a HMAC field, hence no performance impact.

- o when present, the HMAC field MUST only be checked and validated by the first router of the segment routing domain, this router is named 'validating SR router'. Downstream routers MAY NOT inspect the HMAC field.
- o this validating router can also have a cache of <IPv6 header + SRH, HMAC field value> to improve the performance. It is not the same use case as in IPsec where HMAC value was unique per packet, in SRH, the HMAC value is unique per flow.
- o Last point, hash functions such as SHA-2 have been optimized for security and performance and there are multiple implementations with good performance.

With the above points in mind, the performance impact of using HMAC is minimized.

3.3. Pre-shared key management

The field HMAC Key-id allows for:

- o key roll-over: when there is a need to change the key (the hash pre-shared secret), then multiple pre-shared keys can be used simultaneously. The validating routing can have a table of <HMAC Key-id, pre-shared secret> for the currently active and future keys.
- o different algorithm: by extending the previous table to <HMAC Key-id, hash function, pre-shared secret>, the validating router can also support simultaneously several hash algorithms (see section Section 3.1)

The pre-shared secret distribution can be done:

- o in the configuration of the validating routers, either by static configuration or any SDN oriented approach;
- o dynamically using a trusted key distribution such as [RFC6407]

The intent of this document is NOT to define yet-another-key-distribution-protocol.

4. Deployment Models

4.1. Nodes within the SR domain

A SR domain is defined as a set of interconnected routers where all routers at the perimeter are configured to insert and act on SRH. Some routers inside the SR domain can also act on SRH or simply forward IPv6 packets.

The routers inside a SR domain can be trusted to generate SRH and to process SRH received on interfaces that are part of the SR domain. These nodes MUST drop all SRH packets received on an interface that is not part of the SR domain and containing a SRH whose HMAC field cannot be validated by local policies. This includes obviously packet with a SRH generated by a non-cooperative SR domain.

If the validation fails, then these packets MUST be dropped, ICMP error messages (parameter problem) SHOULD be generated (but rate limited) and SHOULD be logged.

4.2. Nodes outside of the SR domain

Nodes outside of the SR domain cannot be trusted for physical security; hence, they need to request by some trusted means (outside of the scope of this document) a complete SRH for each new connection (i.e. new destination address). The received SRH MUST include a HMAC Key-id and HMAC field which is computed correctly (see Section 3).

When an outside node sends a packet with an SRH and towards a SR domain ingress node, the packet MUST contain the HMAC Key-id and HMAC field and the the destination address MUST be an address of a SR domain ingress node .

The ingress SR router, i.e., the router with an interface address equals to the destination address, MUST verify the HMAC field with respect to the HMAC Key-id.

If the validation is successful, then the packet is simply forwarded as usual for a SR packet. As long as the packet travels within the SR domain, no further HMAC check needs to be done. Subsequent routers in the SR domain MAY verify the HMAC field when they process the SRH (i.e. when they are the destination).

If the validation fails, then this packet MUST be dropped, an ICMP error message (parameter problem) SHOULD be generated (but rate limited) and SHOULD be logged.

4.3. SR path exposure

As the intermediate SR nodes addresses appears in the SRH, if this SRH is visible to an outsider then he/she could reuse this knowledge to launch an attack on the intermediate SR nodes or get some insider knowledge on the topology. This is especially applicable when the path between the source node and the first SR domain ingress router is on the public Internet.

The first remark is to state that 'security by obscurity' is never enough; in other words, the security policy of the SR domain MUST assume that the internal topology and addressing is known by the attacker. A simple traceroute will also give the same information (with even more information as all intermediate nodes between SID will also be exposed). IPsec Encapsulating Security Payload [RFC4303] cannot be use to protect the SRH as per RFC4303 the ESP header must appear after any routing header (including SRH).

To prevent a user to leverage the gained knowledge by intercepting SRH, it is recommended to apply an infrastructure Access Control List (iACL) at the edge of the SR domain. This iACL will drop all packets from outside the SR-domain whose destination is any address of any router inside the domain. This security policy should be tuned for local operations.

4.4. Impact of BCP-38

BCP-38 [RFC2827], also known as "Network Ingress Filtering", checks whether the source address of packets received on an interface is valid for this interface. The use of loose source routing such as SRH forces packets to follow a path which differs from the expected routing. Therefore, if BCP-38 was implemented in all routers inside the SR domain, then SR packets could be received by an interface which is not expected one and the packets could be dropped.

As a SR domain is usually a subset of one administrative domain, and as BCP-38 is only deployed at the ingress routers of this administrative domain and as packets arriving at those ingress routers have been normally forwarded using the normal routing information, then there is no reason why this ingress router should drop the SRH packet based on BCP-38. Routers inside the domain commonly do not apply BCP-38; so, this is not a problem.

5. IANA Considerations

There are no IANA request or impact in this document.

6. Manageability Considerations

TBD

7. Security Considerations

Security mechanisms applied to Segment Routing over IPv6 networks are detailed in Section 3.

8. Acknowledgements

The authors would like to thank Dave Barach and Stewart Bryant for their contributions to this document.

9. References

9.1. Normative References

- [FIPS180-4] National Institute of Standards and Technology, "FIPS 180-4 Secure Hash Standard (SHS)", March 2012, <<http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
- [RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, December 2007.
- [RFC6407] Weis, B., Rowles, S., and T. Hardjono, "The Group Domain of Interpretation", RFC 6407, October 2011.

9.2. Informative References

- [I-D.filsfils-spring-segment-routing-use-cases]
Filsfils, C., Francois, P., Previdi, S., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., Kini, S., and E. Crabbe, "Segment Routing Use Cases", draft-filsfils-spring-segment-routing-use-cases-01 (work in progress), October 2014.
- [I-D.ietf-isis-segment-routing-extensions]
Previdi, S., Filsfils, C., Bashandy, A., Gredler, H., Litkowski, S., Decraene, B., and J. Tantsura, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-03 (work in progress), October 2014.
- [I-D.ietf-ospf-ospfv3-segment-routing-extensions]
Psenak, P., Previdi, S., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPFv3 Extensions for Segment Routing", draft-ietf-ospf-ospfv3-segment-routing-extensions-02 (work in progress), February 2015.
- [I-D.ietf-spring-ipv6-use-cases]
Brzozowski, J., Leddy, J., Leung, I., Previdi, S., Townsley, W., Martin, C., Filsfils, C., and R. Maglione, "IPv6 SPRING Use Cases", draft-ietf-spring-ipv6-use-cases-03 (work in progress), November 2014.
- [I-D.ietf-spring-problem-statement]
Previdi, S., Filsfils, C., Decraene, B., Litkowski, S., Horneffer, M., and R. Shakir, "SPRING Problem Statement and Requirements", draft-ietf-spring-problem-statement-03 (work in progress), October 2014.
- [I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Shakir, R., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", draft-ietf-spring-segment-routing-01 (work in progress), February 2015.
- [I-D.previdi-6man-segment-routing-header]
Previdi, S., Filsfils, C., Field, B., and I. Leung, "IPv6 Segment Routing Header (SRH)", draft-previdi-6man-segment-routing-header-05 (work in progress), January 2015.
- [RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [RFC4942] Davies, E., Krishnan, S., and P. Savola, "IPv6 Transition/ Co-existence Security Considerations", RFC 4942, September 2007.
- [RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, March 2012.

Authors' Addresses

Eric Vyncke (editor)
Cisco Systems, Inc.
De Kleetlaann 6A
Diegem 1831
Belgium

Email: evyncke@cisco.com

Stefano Previdi
Cisco Systems, Inc.
Via Del Serafico, 200
Rome 00142
Italy

Email: sprevidi@cisco.com

David Lebrun
Universite Catholique de Louvain
Place Ste Barbe, 2
Louvain-la-Neuve, 1348
Belgium

Email: david.lebrun@uclouvain.be

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: September 9, 2015

A. Wang
China Telecom
S. Jiang
Huawei Technologies Co., Ltd
March 8, 2015

IPv6 Flow Label Reflection
draft-wang-6man-flow-label-reflection-01

Abstract

The current definition of the IPv6 Flow Label focuses mainly on how the packet source forms the value of this field and how the forwarder in-path treats it. In network operations, there are needs to correlate an upstream session and the corresponding downstream session together. This document propose a flow label reflection mechanism that network devices copy the flow label value from received packets to the corresponding flow label field in return packets. This mechanism could simplify the network traffic recognition process in network operations and make the policy for both directions of traffic of one session consistent.

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 September 9, 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. Summary of the current usage for IPv6 Flow Label	3
2. Requirements Language	4
3. Potential Benefit of Flow Label Reflection	4
4. Flow Label Reflection Behaviors on Network Devices	4
5. Applicable Scenarios	5
5.1. Flow Label Reflection on CP servers	5
5.2. Flow Label Reflection for Bi-direction Tunnels	6
5.3. Flow Label Reflection on edge devices	7
5.4. Misc Possible Scenarios	7
5.4.1. Aid to mitigate the ND cache DDoS Attack	7
5.4.2. Improve the efficiency of PTB problem solution in load-balance environment	8
6. Deployment Consideration	8
7. Security Considerations	9
8. IANA Considerations	9
9. Acknowledgements	9
10. References	9
10.1. Normative References	9
10.2. Informative References	10
Authors' Addresses	10

1. Introduction

The IPv6 flow label [RFC6437] in the fixed IPv6 header is designed to differentiate the various flow session of IPv6 traffic; it can accelerate the clarification and treatment of IPv6 traffic by the network devices in its forwarding path. In practice, many current implementations use the 5-tuple {dest addr, source addr, protocol, dest port, source port} as the identifier of network flows. However, transport-layer information, such as the port numbers, is not always in a fixed position, since it follows any IPv6 extension headers that may be present; in contrast, the flow label is at a fixed position in every IPv6 packet and easier to access. In fact, the logic of finding the transport header is always more complex for IPv6 than for IPv4, due to the absence of an Internet Header Length field in IPv6. Additionally, if packets are fragmented, the flow label will be present in all fragments, but the transport header will only be in one packet. Therefore, within the lifetime of a given transport-

layer connection, the flow label can be a more convenient "handle" than the port number for identifying that particular connection.

The usages of IPv6 flow label, so far as briefly summarized in Section 1.1, only exploit the characteristic of IPv6 flow label in one direction.

In current practice, an application session is often recognized as two separated IP traffics, in two opposite directions. However, from the point view of a service provider, the upstream and downstream of one session should be handled together, particularly, when application-aware operations are placed in the network. A ubiquitous example is that end user initiates a request, with small-scale data transmitted, towards a content server, then the server responds with a large set of follow-up packets. The bi-directional flows should be correlated together and handled with the same policy. Ideally, the request embeds a flow recognition identifier that is accessible and the follow-up response packets carry the same identifier. The flow label is a good choice for the flow recognition identifier.

This document proposes a flow label reflection mechanism so that network devices copy the flow label value from received packets to the corresponding flow label field in return packets. By having the same flow label value in the downstream and upstream of one IPv6 traffic session, the network traffic recognition process and the traffic policy deployment in network operations could be simplified. It may also increase the accuracy of network traffic recognition.

Several applicable scenarios of the IPv6 flow label reflection are also given, in Section 5. For now, this document only considers the scenario in a single administrative domain, although the IPv6 flow label reflection mechanism may also bring benefits into cross domain scenarios.

1.1. Summary of the current usage for IPv6 Flow Label

[RFC6438] describe the usage of IPv6 Flow Label for ECMP and link aggregation in Tunnels; it mainly utilizes the random distribution characteristic of IPv6 flow label. [RFC7098] also describes similar usage in server farms.

All these usage scenarios consider only the usage of IPv6 flow label in one direction, while many bi-directional network traffics need to be treated together.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when they appear in ALL CAPS. When these words are not in ALL CAPS (such as "should" or "Should"), they have their usual English meanings, and are not to be interpreted as [RFC2119] key words.

Flow Label Reflection A mechanism/behavior so that a network device copies the value of flow label from a IPv6 flow into a corresponding return IPv6 flow.

Flow Label Reflection Device A network device that applies the flow label reflection mechanism. It is the end of an IPv6 flow and the initiation node of the corresponding return IPv6 flow.

3. Potential Benefit of Flow Label Reflection

With flow label reflection mechanism, the IPv6 Flow Label could be used to correlate the upstream and downstream packets of bi-directional traffics:

- o It makes the downstream and upstream of one session be easily recognized. It makes the correlation of traffic and then the recognition of various traffics easier.
- o The network operator can easily apply the same policy to the bi-directional traffic of one interested session
- o The traffic analyzer can also easily correlate the upstream and downstream of one session to find the symptoms of various internet protocols.

4. Flow Label Reflection Behaviors on Network Devices

To fulfill the flow label reflection mechanism, the below proposed behaviors on network devices:

- o The generation method of IPv6 flow label in source IPv6 node SHOULD follow the guidelines in [RFC6437], that is the IPv6 flow label should be generated randomly and distributed enough.
- o On the Flow Label Reflection Device, the value of IPv6 Flow Label from received packets SHOULD be copied into the corresponding flow label field in return packets by the flow label reflection devices.

- o The forwarding nodes within the management domain SHOULD follow the specification in [RFC6437], that is the IPv6 flow label SHOULD NOT be modified in the path, unless flow label value in arriving packets is zero. The forwarding nodes MAY follow the specification in [RFC6438] when using the flow label for load balancing by equal cost multipath (ECMP) routing and for link aggregation, particularly for IPv6-in-IPv6 tunneled traffic.
- o The network traffic recognition devices, or devices that may have differentiated operations per flow, SHOULD recognize and analyze network traffics based on 3-tuple of {dest addr, source addr, flowlabel}. It SHOULD consider the traffics that have same flow label value and reversed source/dest addr as upstream and downstream of the same flow, match them together to accomplish the traffic recognition process.
- o Other network operations MAY also be based on 3-tuple of {dest addr, source addr, flowlabel}.

5. Applicable Scenarios

This section describes some applicable scenarios, which network operators can benefit from deploying the flow label reflection mechanism. It is not a complete enumeration. More scenarios may be introduced in the future.

5.1. Flow Label Reflection on CP servers

There is rapidly increasing requirement from service providers (SP) to cooperate with the content providers (CP) to provide more accurate services and charging policies based on accurate traffic recognition. The service providers need to recognize the CP/SP's bi-directional traffics at the access edge devices of the network, such as BRAS/PDSN/P-GW devices.

Normally, the burden for these edge devices to recognize the subscriber's upstream traffic is light, because request messages are typically small. But they often need more resource to recognize downstream traffics, which normally contain large data. With flow label reflection on CP servers, recognition based on the 3-tuple of {dest addr, source addr, flowlabel} would reduce the consumption of recognition and make the correlation process much easier.

In this scenario, the CP servers would be the Flow Label Reflection Devices. They copy the flow label value from received upstream user request packets to the corresponding flow label field in return downstream packets.

The access edge devices of service provider scrutinize the subscriber's upstream IPv6 traffic and record the binding of 3-tuple and traffic-specific policy. If the flow label is zero, the access edge devices must rewrite the flow label value according to local policy. With the recorded binding information, the access edge devices can easily recognize and match the downstream packet to the previous recognized upstream packet, by just accessing 3-tuple. The edge devices can then apply the corresponding traffic policy to the upstream/downstream of the session to the cooperated CP.

Note: this mechanism may not reliable when the CP servers are not directly connected to the service provider, because there is no guarantee the flow label would not be changed by intermediate devices in other administrative domains.

5.2. Flow Label Reflection for Bi-direction Tunnels

Tunnel is ubiquitous within service provider networks. It is very difficult (important if the tunnel is encrypted) for intermediate network devices to recognize the inner encapsulated packet, although such recognition could be very helpful in some scenarios, such as traffic statistics, network diagnoses, etc. Furthermore, such recognition normally requires to correlate bi-direction traffic together. The flow label reflection mechanism could provide help in such requirement scenarios.

In this scenario, the tunnel end devices would be the Flow Label Reflection Devices. They record the flow label value from received tunnel packets, and copy it to the corresponding flow label field in return packets, which can be recognized by 5-tuple or 3-tuple of the inner packet at the tunnel end devices.

The tunnel initiating devices should generate different flow label values for different inner flow traffics based on their 5-tuple or 3-tuple in accordance with [RFC6437]. Note: if the inner flow is encryption in ESP model [RFC4303], the transport-layer port numbers are inaccessible. In such case, 5-tuple is not available.

Then the intermediate network device can easily distinguish the different flow within the same tunnel transport link and correlate bi-direction traffics of same flow together. This can also increase the service provider's traffic control capabilities.

This mechanism can also work when the encapsulated traffics are IPv4 traffics, such as DS-Lite scenario [RFC6333]. The IPv4 5-tuple may be used as the input for the flow label generation.

5.3. Flow Label Reflection on edge devices

If the flow label reflection mechanisms have been applied on peer host, the service provider could always use it for bi-directional traffic recognition. However, there is no guarantee the flow label would not be changed by intermediate devices in other administrative domains. Therefore, to make the flow label value trustful, the edge devices need to validate the flow label reflection.

In this scenario, the edge devices would be the (backup) Flow Label Reflection Devices. They record the flow label value from the packets that leave the domain. When the corresponding flow label field in return packets are recognized by 5-tuple or 3-tuple at the edge devices, the edge devices should check the flow label as below:

- o if the flow label matches the record value, it remains;
- o if the flow label is zero, the edge devices copy the record value into it;
- o if the flow label is non-zero, but does not matches the record value, the edge devices can decide the flow label are modified by other intermediate devices (with the assumption the peer host has reflect the original flow label), then restore the flow label using the record value.

Then the network recognition devices located anywhere within the service provider network could easily correlate bi-directional traffics together, and apply traffic-specific policy accordingly.

5.4. Misc Possible Scenarios

In the below scenarios, the flow label reflection mechanism needs to be combined with other mechanisms in order to achieve the design goals.

5.4.1. Aid to mitigate the ND cache DDoS Attack

Neighbor Discovery Protocol [RFC4861][RFC4861] is vulnerable for the possible DDoS attack to the device's ND cache, see section 11.1, [RFC4861]. There are many proposals are aiming to mitigate this problem, but none of them are prevalent now. It is mainly because that there is no obvious mechanism to assure the validation of the NS/RS packet on the first arrival, the receiving node by default will cache the link-layer address of the NA packet. Reverse detection mechanisms can be added to solve this issue. However, for reverse detection mechanisms, there would be another issue: how to pair the return NA/RA packet with the NS/RS packet on the sending node. It

can be solved by applying the flow label reflection mechanism in the return NA/RA packet. Then the sending node can pair the reverse detect NS/RS packet with original NA/RA packet and response to the reverse detect NS/RS packet correctly. Only the NS/RS packet that passed the reverse detection validation will be accept by the node and the link-layer address within it will be cached.

5.4.2. Improve the efficiency of PTB problem solution in load-balance environment

[I-D.v6ops-pmtud-ecmp-problem] introduces the Packet Too Big [RFC4443] problem in load-balance environment. The downstream packet from a server, which responses to a client request message, may meet a forwarding node that rejects the packet for "too big" reason. The PTB error ICMPv6 message should be returned to the original server. However, it requests the load balancer to distribute the PTB error ICMPv6 message based on the information of the invoking packet within the ICMPv6 packet, not the ICMPv6 packet itself. The load balancer needs to obtain the source IP address and transport port information within the invoking packet.

However, if both the server and the forwarding node that generates the PTB message apply the flow label reflection mechanism, the PTB error ICMPv6 message would have the same flow label with the original client request message. Then, the load balancer, that follows [RFC7098], could easily forward the PTB packet to same server without parsing the transport port in the invoking packet, thus increases the efficiency.

6. Deployment Consideration

The IPv6 flow label reflection mechanism requires the "Flow Label Reflection Device" to be stateful, store the flow label value and copy it to the corresponding return packet. Such change cannot be accomplished within a short term, and therefore the deployment of this mechanism will be accomplished gradually. During the incremental deployment period, the traditional recognition mechanisms, which are more expensive, would coexist. The traffics that could not be recognized by 3-tuple of {dest addr, source addr, flowlabel} could fall back to the traditional process or be skipped over by advanced services. The more devices support the flow label reflection mechanism, the less consumption for traffic recognition from the network management perspective, or the better coverage of advanced services that are based on the traffic recognition.

7. Security Considerations

Security aspects of the flow label are discussed in [RFC6437]. A malicious source or man-in-the-middle could disturb the traffic recognition by manipulating flow labels. However, the worst case is that fall back to the current practice that an application session is often recognized as two separated IP traffics. The flow label does not significantly alter this situation.

Specifically, the IPv6 flow label specification [RFC6437] states that "stateless classifiers should not use the flow label alone to control load distribution." This is answered by also using the source and destination addresses with flow label.

8. IANA Considerations

This draft does not request any IANA action.

9. Acknowledgements

The authors would like to thanks Brian Carpenter, who gave many useful advices. The authors would also like to thanks the valuable comments made by Fred Baker, Lee Howard, Mark ZZZ Smith, Jeroen Massar, Florent Fourcot and other members of V6OPS WG. Also, special thanks for Florent Fourcot, who have implemented the flow label reflection mechanims in the Linux.

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

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.
- [RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.
- [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.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.

- [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.
- [RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, November 2011.
- [RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, November 2011.

10.2. Informative References

- [I-D.v6ops-pmtud-ecmp-problem] Byerly, M., Hite, M., and J. Jaeggli, "Close encounters of the ICMP type 2 kind (near misses with ICMPv6 PTB)", draft-v6ops-pmtud-ecmp-problem-00 (work in progress), August 2014.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, August 2011.
- [RFC7098] Carpenter, B., Jiang, S., and W. Tarreau, "Using the IPv6 Flow Label for Load Balancing in Server Farms", RFC 7098, January 2014.

Authors' Addresses

Aijun Wang
China Telecom
Beijing Research Institute, China Telecom Cooperation Limited
No.118, Xizhimenneidajie, Xicheng District, Beijing 100035
China

Phone: 86-10-58552347
Email: wangaj@ctbri.com.cn

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

Email: jiangsheng@huawei.com