

6MAN
Internet-Draft
Obsoletes: 6874 (if approved)
Updates: 3986, 3987 (if approved)
Intended status: Standards Track
Expires: 12 August 2022

B. Carpenter
Univ. of Auckland
S. Cheshire
Apple Inc.
R. Hinden
Check Point Software
8 February 2022

Representing IPv6 Zone Identifiers in Address Literals and Uniform
Resource Identifiers
draft-car Carpenter-6man-rfc6874bis-03

Abstract

This document describes how the zone identifier of an IPv6 scoped address, defined as <zone_id> in the IPv6 Scoped Address Architecture (RFC 4007), can be represented in a literal IPv6 address and in a Uniform Resource Identifier that includes such a literal address. It updates the URI Generic Syntax and Internationalized Resource Identifier specifications (RFC 3986, RFC 3987) accordingly, and obsoletes RFC 6874.

Discussion Venue

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the 6MAN mailing list (ipv6@ietf.org), which is archived at <https://mailarchive.ietf.org/arch/browse/ipv6/> (<https://mailarchive.ietf.org/arch/browse/ipv6/>).

Status of This Memo

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

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

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

This Internet-Draft will expire on 12 August 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
2. Issues with Implementing RFC 6874	4
3. Specification	4
4. URI Parsers	6
5. Security Considerations	7
6. Acknowledgements	7
7. References	7
7.1. Normative References	7
7.2. Informative References	8
Appendix A. Options Considered	9
Appendix B. Change log	10
Authors' Addresses	11

1. Introduction

The Uniform Resource Identifier (URI) syntax specification [RFC3986] defined how a literal IPv6 address can be represented in the "host" part of a URI. Two months later, the IPv6 Scoped Address Architecture specification [RFC4007] extended the text representation of limited-scope IPv6 addresses such that a zone identifier may be concatenated to a literal address, for purposes described in that specification. Zone identifiers are especially useful in contexts in which literal addresses are typically used, for example, during fault diagnosis, when it may be essential to specify which interface is used for sending to a link-local address. It should be noted that zone identifiers have purely local meaning within the node in which they are defined, often being the same as IPv6 interface names. They are completely meaningless for any other node. Today, they are meaningful only when attached to addresses with less than global scope, but it is possible that other uses might be defined in the future.

The IPv6 Scoped Address Architecture specification [RFC4007] does not specify how zone identifiers are to be represented in URIs.

Practical experience has shown that this feature is useful or necessary, in at least three use cases:

1. When using a web browser for simple debugging actions involving link-local addresses on a host with more than one active link interface.
2. When using a web browser to configure or reconfigure a device which only has a link local address and whose only configuration tool is a web server, again from a host with more than one active link interface.
3. When using an HTTP-based protocol for establishing link-local relationships, such as the Apple CUPS printing mechanism [CUPS].

It should be noted that whereas some operating systems and network APIs support a default zone identifier as described in [RFC4007], others do not, and for them an appropriate URI syntax is particularly important.

In the past, some browser versions directly accepted the IPv6 Scoped Address syntax [RFC4007] for scoped IPv6 addresses embedded in URIs, i.e., they were coded to interpret a "%" sign following the literal address as introducing a zone identifier [RFC4007], instead of introducing two hexadecimal characters representing some percent-encoded octet [RFC3986]. Clearly, interpreting the "%" sign as introducing a zone identifier is very convenient for users, although it is not supported by the URI syntax [RFC3986] or the Internationalized Resource Identifier (IRI) syntax [RFC3987]. Therefore, this document updates RFC 3986 and RFC 3987 by adding syntax to allow a zone identifier to be included in a literal IPv6 address within a URI.

It should be noted that in contexts other than a user interface, a zone identifier is mapped into a numeric zone index or interface number. The MIB textual convention InetZoneIndex [RFC4001] and the socket interface [RFC3493] define this as a 32-bit unsigned integer. The mapping between the human-readable zone identifier string and the numeric value is a host-specific function that varies between operating systems. The present document is concerned only with the human-readable string.

Several alternative solutions were considered while this document was developed. Appendix A briefly describes the various options and their advantages and disadvantages.

This document obsoletes its predecessor [RFC6874] by greatly simplifying its recommendations and requirements for URI parsers. Its effect on the formal URI syntax [RFC3986] is different from that of RFC 6874.

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

2. Issues with Implementing RFC 6874

Several issues prevented RFC 6874 being implemented in browsers:

1. There was some disagreement with requiring percent-encoding of the "%" sign preceding a zone identifier. This requirement is dropped in the present document.
2. The requirement to delete any zone identifier before emitting a URI from the host in an HTTP message was considered both too complex to implement and in violation of normal HTTP practice [RFC7230]. This requirement has been dropped from the present document.
3. The suggestion to pragmatically allow a bare "%" sign when this would be unambiguous was considered both too complex to implement and confusing for users. This suggestion has been dropped from the present document since it is now irrelevant.

3. Specification

According to IPv6 Scoped Address syntax [RFC4007], a zone identifier is attached to the textual representation of an IPv6 address by concatenating "%" followed by <zone_id>, where <zone_id> is a string identifying the zone of the address. However, the IPv6 Scoped Address Architecture specification gives no precise definition of the character set allowed in <zone_id>. There are no rules or de facto standards for this. For example, the first Ethernet interface in a host might be called %0, %1, %en1, %eth0, or whatever the implementer happened to choose. Also, %25 would be valid.

In a URI, a literal IPv6 address is always embedded between "[" and "]". This document specifies how a <zone_id> can be appended to the address. According to the text in Section 2.4 of [RFC3986], "%" must be percent-encoded as "%25" to be used as data within a URI. However, in the formal ABNF syntax of RFC 3986, this only applies where the "pct-encoded" element appears. For this reason, it is

possible to extend the ABNF such that the scoped address `fe80::abcd%en1` would appear in a URI as `http://[fe80::abcd%en1]` or `https://[fe80::abcd%en1]`.

A `<zone_id>` MUST contain only ASCII characters classified as "unreserved" for use in URIs [RFC3986]. This excludes characters such as `]` or even `%` that would complicate parsing. The `<zone_id>` `"25"` cannot be forbidden since it is valid in some operating systems, so a parser MUST NOT apply percent decoding to a URI such as `http://[fe80::abcd%25]`.

If an operating system uses any other characters in zone or interface identifiers that are not in the "unreserved" character set, they cannot be used in a URI.

We now present the corresponding formal syntax.

The URI syntax specification [RFC3986] formally defines the IPv6 literal format in ABNF [RFC5234] by the following rule:

```
IP-literal = "[" ( IPv6address / IPvFuture  ) "]"
```

To provide support for a zone identifier, the existing syntax of IPv6address is retained, and a zone identifier may be added optionally to any literal address. This syntax allows flexibility for unknown future uses. The rule quoted above from [RFC3986] is replaced by three rules:

```
IP-literal = "[" ( IPv6address / IPv6addrz / IPvFuture  ) "]"
```

```
ZoneID = 1*( unreserved )
```

```
IPv6addrz = IPv6address "%" ZoneID
```

This change also applies to [RFC3987].

This syntax fills the gap that is described at the end of Section 11.7 of the IPv6 Scoped Address Architecture specification [RFC4007]. It replaces and obsoletes the syntax in Section 2 of [RFC6874].

The established rules for textual representation of IPv6 addresses [RFC5952] SHOULD be applied in producing URIs.

The URI syntax specification [RFC3986] states that URIs have a global scope, but that in some cases their interpretation depends on the end-user's context. URIs including a ZoneID are to be interpreted only in the context of the host at which they originate, since the ZoneID is of local significance only.

The IPv6 Scoped Address Architecture specification [RFC4007] offers guidance on how the ZoneID affects interface/address selection inside the IPv6 stack. Note that the behaviour of an IPv6 stack, if it is passed a non-null zone index for an address other than link-local, is undefined.

4. URI Parsers

This section discusses how URI parsers, such as those embedded in web browsers, might handle this syntax extension. Unfortunately, there is no formal distinction between the syntax allowed in a browser's input dialogue box and the syntax allowed in URIs. For this reason, no normative statements are made in this section.

In practice, although parsers respect the established syntax, they are coded pragmatically rather than being formally syntax-driven. Typically, IP address literals are handled by an explicit code path. Parsers have been inconsistent in providing for ZoneIDs. Most have no support, but there have been examples of ad hoc support. For example, some versions of Firefox allowed the use of a ZoneID preceded by a bare "%" character, but this feature was removed for consistency with established syntax [RFC3986]. As another example, some versions of Internet Explorer allowed use of a ZoneID preceded by a "%" character encoded as "%25", still beyond the syntax allowed by the established rules [RFC3986]. This syntax extension is in fact used internally in the Windows operating system and some of its APIs.

It is desirable for all URI parsers to recognise a ZoneID according to the syntax defined in Section 3.

URIs including a ZoneID have no meaning outside the originating HTTP client node. However, in some use cases, such as CUPS mentioned above, the URI will be reflected back to the client.

The various use cases for the ZoneID syntax will cause it to be entered in a browser's input dialogue box. Thus, URIs including a ZoneID are unlikely to occur in HTML documents. However, if they do (for example, in a diagnostic script coded in HTML), it would be appropriate to treat them exactly as above.

5. Security Considerations

The security considerations from the URI syntax specification [RFC3986] and the IPv6 Scoped Address Architecture specification [RFC4007] apply. In particular, this URI format creates a specific pathway by which a deceitful zone index might be communicated, as mentioned in the final security consideration of the Scoped Address Architecture specification.

To limit this risk, implementations MUST NOT allow use of this format except for well-defined usages, such as sending to link-local addresses under prefix fe80::/10. At the time of writing, this is the only well-defined usage known.

6. Acknowledgements

The lack of this format was first pointed out by Margaret Wasserman and later by Kerry Lynn. A previous draft document by Bill Fenner and Martin Dürst [LITERAL-ZONE] discussed this topic but was not finalised. Michael Sweet and Andrew Cady explained some of the difficulties caused by RFC 6874. The ABNF syntax proposed above was drafted by Andrew Cady.

Valuable comments and contributions were made by Karl Auer, Carsten Bormann, Benoit Claise, Martin Dürst, Stephen Farrell, Brian Haberman, Ted Hardie, Philip Homburg, Tatuya Jinmei, Yves Lafon, Barry Leiba, Radia Perlman, Tom Petch, Michael Richardson, Tomoyuki Sahara, Juergen Schoenwaelder, Nico Schottelius, Dave Thaler, Martin Thomson, Ole Troan, and others.

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC3987] Duerst, M. and M. Suignard, "Internationalized Resource Identifiers (IRIs)", RFC 3987, DOI 10.17487/RFC3987, January 2005, <<https://www.rfc-editor.org/info/rfc3987>>.

- [RFC4007] Deering, S., Haberman, B., Jinmei, T., Nordmark, E., and B. Zill, "IPv6 Scoped Address Architecture", RFC 4007, DOI 10.17487/RFC4007, March 2005, <<https://www.rfc-editor.org/info/rfc4007>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.
- [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <<https://www.rfc-editor.org/info/rfc5952>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

7.2. Informative References

- [CUPS] Apple, "CUPS open source printing system", 2021, <<https://www.cups.org/>>.
- [LITERAL-ZONE] Fenner, B. and M. Dürst, "Formats for IPv6 Scope Zone Identifiers in Literal Address Formats", Work in Progress, October 2005.
- [RFC3493] Gilligan, R., Thomson, S., Bound, J., McCann, J., and W. Stevens, "Basic Socket Interface Extensions for IPv6", RFC 3493, DOI 10.17487/RFC3493, February 2003, <<https://www.rfc-editor.org/info/rfc3493>>.
- [RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, DOI 10.17487/RFC4001, February 2005, <<https://www.rfc-editor.org/info/rfc4001>>.
- [RFC6874] Carpenter, B., Cheshire, S., and R. Hinden, "Representing IPv6 Zone Identifiers in Address Literals and Uniform Resource Identifiers", RFC 6874, DOI 10.17487/RFC6874, February 2013, <<https://www.rfc-editor.org/info/rfc6874>>.
- [RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <<https://www.rfc-editor.org/info/rfc7230>>.

Appendix A. Options Considered

The syntax defined above allows a ZoneID to be added to any IPv6 address. The 6man WG discussed and rejected an alternative in which the existing syntax of IPv6address would be extended by an option to add the ZoneID only for the case of link-local addresses. It was felt that the solution presented in this document offers more flexibility for future uses and is more straightforward to implement.

The various syntax options considered are now briefly described.

1. Leave the problem unsolved.

This would mean that per-interface diagnostics would still have to be performed using ping or ping6:

```
ping fe80::abcd%en1
```

Advantage: works today.

Disadvantage: less convenient than using a browser. Leaves some use cases unsatisfied.

2. Simply use the percent character:

```
http://[fe80::abcd%en1]
```

Advantage: allows use of browser; allows cut and paste.

Disadvantage: requires code changes to all URI parsers.

This is the option chosen for standardisation.

3. Simply use an alternative separator:

```
http://[fe80::abcd-en1]
```

Advantage: allows use of browser; simple syntax.

Disadvantage: Requires all IPv6 address literal parsers and generators to be updated in order to allow simple cut and paste; inconsistent with existing tools and practice.

Note: The initial proposal for this choice was to use an underscore as the separator, but it was noted that this becomes effectively invisible when a user interface automatically underlines URLs.

4. Simply use the "IPvFuture" syntax left open in RFC 3986:

`http://[v6.fe80::abcd_en1]`

Advantage: allows use of browser.

Disadvantage: ugly and redundant; doesn't allow simple cut and paste.

5. Retain the percent character already specified for introducing zone identifiers for IPv6 Scoped Addresses [RFC4007], and then percent-encode it when it appears in a URI, according to the already-established URI syntax rules [RFC 3986]:

`http://[fe80::abcd%25en1]`

Advantage: allows use of browser; consistent with general URI syntax.

Disadvantage: somewhat ugly and confusing; doesn't allow simple cut and paste.

Appendix B. Change log

This section is to be removed before publishing as an RFC.

* draft-carpenter-6man-rfc6874bis-03, 2022-02-08:

- Changed to bare % signs.
- Added IRIs, RFC3987
- Editorial fixes

* draft-carpenter-6man-rfc6874bis-02, 2021-18-12:

- Give details of open issues
- Update authorship
- Editorial fixes

* draft-carpenter-6man-rfc6874bis-01, 2021-07-11:

- Added section on issues with RFC6874
- Removed suggested heuristic for bare % signs

- Editorial fixes
- * draft-carpenter-6man-rfc6874bis-00, 2021-07-05:
- Initial version

Authors' Addresses

Brian Carpenter
School of Computer Science
University of Auckland
PB 92019
Auckland 1142
New Zealand

Email: brian.e.carpenter@gmail.com

Stuart Cheshire
Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
United States of America

Email: cheshire@apple.com

Robert M. Hinden
Check Point Software
959 Skyway Road
San Carlos, CA 94070
United States of America

Email: bob.hinden@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 27, 2022

J. Dong
Z. Li
Huawei Technologies
C. Xie
C. Ma
China Telecom
G. Mishra
Verizon Inc.
October 24, 2021

Carrying Virtual Transport Network (VTN) Identifier in IPv6 Extension
Header
draft-dong-6man-enhanced-vpn-vtn-id-06

Abstract

Virtual Private Networks (VPNs) provide different customers with logically separated connectivity over a common network infrastructure. With the introduction and evolvement of 5G and other network scenarios, some existing or new customers may require connectivity services with advanced characteristics comparing to traditional VPNs. Such kind of network service is called enhanced VPNs (VPN+).

A Virtual Transport Network (VTN) is a virtual underlay network which consists of a set of dedicated or shared network resources allocated from the physical underlay network, and is associated with a customized logical network topology. VPN+ services can be delivered by mapping one or a group of overlay VPNs to the appropriate VTNs as the virtual underlay. In packet forwarding, some fields in the data packet needs to be used to identify the VTN the packet belongs to, so that the VTN-specific processing can be performed on each node the packet traverses.

This document proposes a new Hop-by-Hop option of IPv6 extension header to carry the VTN Resource ID, which is used to identify the set of network resources allocated to a VTN for packet processing. The procedure for processing the VTN option is also specified.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute

working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on April 27, 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
1.1. Requirements Language	4
2. New IPv6 Extension Header Option for VTN	4
3. Procedures	5
3.1. VTN Option Insertion	5
3.2. VTN based Packet Forwarding	5
4. Operational Considerations	6
5. IANA Considerations	6
6. Security Considerations	6
7. Contributors	6
8. Acknowledgements	7
9. References	7
9.1. Normative References	7
9.2. Informative References	7
Authors' Addresses	8

1. Introduction

Virtual Private Networks (VPNs) provide different customers with logically isolated connectivity over a common network infrastructure. With the introduction and evolvement of 5G and other network

scenarios, some existing or new customers may require connectivity services with advanced characteristics comparing to traditional VPNs, such as resource isolation from other services or guaranteed performance. Such kind of network service is called enhanced VPN (VPN+). VPN+ service requires the coordination and integration between the overlay VPNs and the network characteristics of the underlay.

[I-D.ietf-teas-enhanced-vpn] describes a framework and the candidate component technologies for providing VPN+ services. It also introduces the concept of Virtual Transport Network (VTN). A Virtual Transport Network (VTN) is a virtual underlay network which consists of a set of dedicated or shared network resources allocated from the physical underlay network, and is associated with a customized logical network topology. VPN+ services can be delivered by mapping one or a group of overlay VPNs to the appropriate VTNs as the underlay, so as to provide the network characteristics required by the customers. In packet forwarding, traffic of different VPN+ services need to be processed separately based on the network resources and the logical topology associated with the corresponding VTN.

[I-D.dong-teas-enhanced-vpn-vtn-scalability] describes the scalability considerations and the possible optimizations for providing a relatively large number of VTNs for VPN+ services. One approach to improve the data plane scalability of VTN is to introduce a dedicated VTN Resource Identifier (VTN Resource ID) in the data packet to identify the set of network resources allocated to a VTN, so that VTN-specific packet processing can be performed using that set of resources, which avoids the possible resource competition with services in other VTNs. This is called Resource Independent (RI) VTN. A VTN Resource ID represents a subset of the resources (e.g. bandwidth, buffer and queuing resources) allocated on a given set of links and nodes which constitute a logical network topology. The logical topology associated with a VTN could be defined using mechanisms such as Multi-Topology [RFC4915], [RFC5120] or Flex-Algo [I-D.ietf-lsr-flex-algo], etc.

This document proposes a mechanism to carry the VTN resource ID in a new Hop-by-Hop option of IPv6 extension header [RFC8200] of IPv6 packet, so that on each network node along the packet forwarding path, the VTN option in the packet is parsed, and the obtained VTN Resource ID is used to instruct the network node to use the set of network resources allocated to the corresponding VTN to process and forward the packet. The procedure for processing the VTN Resource ID is also specified. This provides a scalable solution to support a relatively large number of VTNs in an IPv6 network.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14 RFC 2119 [RFC2119] RFC 8174 [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. New IPv6 Extension Header Option for VTN

A new Hop-by-Hop option type "VTN" is defined to carry the VTN related Identifier in an IPv6 packet. Its format is shown as below:

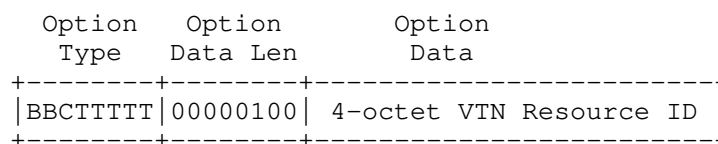


Figure 1. The format of VTN Option

Option Type: 8-bit identifier of the type of option. The type of VTN option is to be assigned by IANA. The highest-order bits of the type field are defined as below:

- o BB 00 The highest-order 2 bits are set to 00 to indicate that a node which does not recognize this type will skip over it and continue processing the header.
- o C 0 The third highest-order bit are set to 0 to indicate this option does not change en route.

Opt Data Len: 8-bit unsigned integer indicates the length of the option Data field of this option, in octets. The value of Opt Data Len of VTN option SHOULD be set to 4.

VTN Resource ID: 4-octet identifier which uniquely identifies the set of network resources allocated to a VTN.

Editor's note: The length of the VTN Resource ID is defined as 4-octet in correspondence to the 4-octet Single Network Slice Selection Assistance Information (S-NSSAI) defined in 3GPP [TS23501].

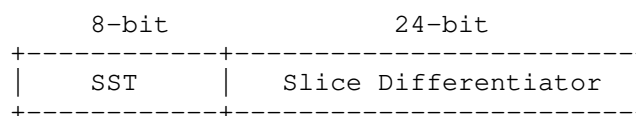


Figure 2. The format of S-NSSAI

3. Procedures

As the VTN option needs to be processed by each node along the path for VTN-specific forwarding, it SHOULD be carried in IPv6 Hop-by-Hop options header when the Hop-by-Hop options header can be either processed or ignored in forwarding plane by all the nodes along the path.

3.1. VTN Option Insertion

When an ingress node of an IPv6 domain receives a packet, according to the traffic classification or mapping policy, the packet is steered into one of the VTNs in the network, then the packet SHOULD be encapsulated in an outer IPv6 header, and the Resource ID of the VTN which the packet is mapped to SHOULD be carried in the VTN option of the Hop-by-Hop options header associated with the outer IPv6 header.

3.2. VTN based Packet Forwarding

On receipt of a packet with the VTN option, each network node which can process the VTN option in fast path SHOULD use the VTN Resource ID to determine the set of local network resources allocated to the VTN for packet processing. The packet forwarding behavior is based on both the destination IP address and the VTN Resource ID. More specifically, the destination IP address is used to determine the next-hop and the outgoing interface, and VTN Resource ID is used to determine the set of network resources on the outgoing interface which are reserved to the VTN for processing and sending the packet. The Traffic Class field of the outer IPv6 header MAY be used to provide Diffserv treatment for packets which belong to the same VTN. The egress node of the IPv6 domain SHOULD decapsulate the outer IPv6 header which includes the VTN option.

In the forwarding plane, there can be different approaches of partitioning the local network resources and allocating them to different VTNs. For example, on one physical interface, a subset of the forwarding plane resources (e.g. the bandwidth and the associated buffer and queuing resources) can be allocated to a particular VTN and represented as a virtual sub-interface with reserved bandwidth resource. In packet forwarding, the IPv6 destination address of the received packet is used to identify the next-hop and the outgoing layer-3 interface, and the VTN Resource ID is used to further identify the virtual sub-interface which is associated with the VTN on the outgoing interface.

Network nodes which do not support the processing of Hop-by-Hop options header SHOULD ignore the Hop-by-Hop options header and

forward the packet only based on the destination IP address. Network nodes which support Hop-by-Hop Options header, but do not support the VTN option SHOULD ignore the VTN option and continue to forward the packet based on the destination IP address and MAY also based on the rest of the Hop-by-Hop Options.

4. Operational Considerations

As described in [RFC8200], network nodes may be configured to ignore the Hop-by-Hop Options header, and in some implementations a packet containing a Hop-by-Hop Options header may be dropped or assigned to a slow processing path. The proposed modification to the processing of IPv6 Hop-by-Hop options header is specified in [I-D.hinden-6man-hbh-processing]. Operator needs to make sure that all the network nodes involved in a VTN can either process Hop-by-Hop Options header in the fast path, or ignore the Hop-by-Hop Option header. Since a VTN is associated with a logical network topology, it is practical to ensure that all the network nodes involved in that logical topology support the processing of the HBH options header and the VTN option. In other word, packets steered into a VTN MUST NOT be dropped due to the existence of the Hop-by-Hop Options header. It is RECOMMENDED to configure all the network nodes involved in a VTN to process the Hop-by-Hop Options header and the VTN option if there is a nob for this.

5. IANA Considerations

This document requests IANA to assign a new option type from "Destination Options and Hop-by-Hop Options" registry.

Value	Description	Reference
TBD	VTN Option	this document

6. Security Considerations

The security considerations with IPv6 Hop-by-Hop options header are described in [RFC8200], [RFC7045] and [I-D.hinden-6man-hbh-processing]. This document introduces a new IPv6 Hop-by-Hop option which is either processed in the fast path or ignored by network nodes, thus it does not introduce additional security issues.

7. Contributors

Zhibo Hu
Email: huzhibo@huawei.com

Lei Bao
Email: baolei7@huawei.com

8. Acknowledgements

The authors would like to thank Juhua Xu, James Guichard, Joel Halpern and Tom Petch for their review and valuable comments.

9. References

9.1. Normative References

- [I-D.ietf-teas-enhanced-vpn]
Dong, J., Bryant, S., Li, Z., Miyasaka, T., and Y. Lee, "A Framework for Enhanced Virtual Private Network (VPN+) Services", draft-ietf-teas-enhanced-vpn-08 (work in progress), July 2021.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

9.2. Informative References

- [I-D.dong-teas-enhanced-vpn-vtn-scalability]
Dong, J., Li, Z., Gong, L., Yang, G., Guichard, J. N., Mishra, G., and F. Qin, "Scalability Considerations for Enhanced VPN (VPN+)", draft-dong-teas-enhanced-vpn-vtn-scalability-03 (work in progress), July 2021.
- [I-D.hinden-6man-hbh-processing]
Hinden, R. M. and G. Fairhurst, "IPv6 Hop-by-Hop Options Processing Procedures", draft-hinden-6man-hbh-processing-01 (work in progress), June 2021.

- [I-D.ietf-lsr-flex-algo]
Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and
A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-
algo-17 (work in progress), July 2021.
- [RFC4915] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., and P.
Pillay-Esnault, "Multi-Topology (MT) Routing in OSPF",
RFC 4915, DOI 10.17487/RFC4915, June 2007,
<<https://www.rfc-editor.org/info/rfc4915>>.
- [RFC5120] Przygienda, T., Shen, N., and N. Sheth, "M-ISIS: Multi
Topology (MT) Routing in Intermediate System to
Intermediate Systems (IS-IS)", RFC 5120,
DOI 10.17487/RFC5120, February 2008,
<<https://www.rfc-editor.org/info/rfc5120>>.
- [RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing
of IPv6 Extension Headers", RFC 7045,
DOI 10.17487/RFC7045, December 2013,
<<https://www.rfc-editor.org/info/rfc7045>>.
- [TS23501] "3GPP TS23.501", 2016,
<[https://portal.3gpp.org/desktopmodules/Specifications/
SpecificationDetails.aspx?specificationId=3144](https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144)>.

Authors' Addresses

Jie Dong
Huawei Technologies
Huawei Campus, No. 156 Beiqing Road
Beijing 100095
China

Email: jie.dong@huawei.com

Zhenbin Li
Huawei Technologies
Huawei Campus, No. 156 Beiqing Road
Beijing 100095
China

Email: lizhenbin@huawei.com

Chongfeng Xie
China Telecom
China Telecom Beijing Information Science & Technology, Beiqijia
Beijing 102209
China

Email: xiechf@chinatelecom.cn

Chenhao Ma
China Telecom
China Telecom Beijing Information Science & Technology, Beiqijia
Beijing 102209
China

Email: machh@chinatelecom.cn

Gyan Mishra
Verizon Inc.

Email: gyan.s.mishra@verizon.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 24, 2021

T. Herbert
SiPanda
June 22, 2021

Limits on Sending and Processing IPv6 Extension Headers
draft-herbert-6man-eh-limits-00

Abstract

This specification defines various limits that may be applied to receiving, sending, and otherwise processing packets that contain IPv6 extension headers. The need for such limits is pragmatic to facilitate interoperability amongst hosts and routers in the presence of extension headers and thereby increasing the feasibility of deployment of 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 <https://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on December 24, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction	2
1.1. Related work	3
1.2. Adherence to the Robustness Principle	4
1.2.1. Be conservative in what you send	4
1.2.2. Be liberal in what you receive	4
2. Overview of extension header limits	5
2.1. Types of nodes	5
2.2. Types of limits	6
2.2.1. Limits on extension header length	6
2.2.2. Limits on option length	6
2.2.3. Limits on number of extension headers	6
2.2.4. Limits on number of options	6
2.2.5. Limits on padding options	7
2.2.6. Limit on IPv6 header chain length	8
2.3. Requirements for extension header limits	11
3. Requirements	12
3.1. Host requirements	12
3.1.1. Sending extension headers	12
3.1.2. Receiving extension headers	13
3.2. Intermediate node and intermediate destination requirements	15
3.3. Intermediate destination requirements	16
4. References	17
4.1. Normative References	17
4.2. Informative References	18
Author's Address	18

1. Introduction

Extension headers are a core component of the IPv6 protocol as specified in [RFC8200]. IPv6 extension headers were originally defined with few restrictions. For instance, there is no specified limit on the number of extension headers a packet may have, nor is there a limit on the length in bytes of extension headers in a packet (other than being limited by the MTU). Similarly, variable length extension headers typically do not have prescribed limits such as limits on the number of Hop-by-Hop or Destination options in a packet. The lack of limits essentially requires implementations to handle every conceivable usage of the protocol, including a myriad of use cases those are obviously outside the realm of ever being realistic or useful in real world deployment.

The lack of limits and the requirements for supporting virtually open-ended protocol have led to a significant lack of support and deployment of extension headers [RFC7872]. Instead of attempting to satisfy the protocol requirements concerning extension headers, some router and middlebox vendors have opted to either invent and apply their own ad hoc limits, relegate packets with extension headers to slow path processing, or have gone so far as to summarily discard all packets with extension headers. The net result of this situation is that deployment and use of extension headers is underwhelming to the extent that they are often considered unusable, and hence IPv6 extension headers have not lived up to their potential as the extensibility mechanism of IPv6.

As an example, consider that Hop-by-Hop Options and Destination Options have no limit on how many options may be placed in a packet nor any limits as to how many options a receiver must process. A single 1500 byte MTU sized packet could legally contain a Hop-by-Hop Options extension header with over seven hundred two byte options. There is no use case for this other than being a Denial of Service attack where an attacker simply creates packets with hundreds of small unknown Hop-by-Hop Options with the two high order bits in the option type set to 00 meaning to skip the unknown option. Any node in that path that attempts to dutifully process all these options per the requirements of [RFC8200] would be easily overwhelmed by the processing needed to parse these options (this is true for both hardware or software implementations).

This specification describes various limits that hosts and intermediate nodes may apply to the processing of extension headers. The goal of establishing limits is to narrow the requirements to better match reasonable use cases thereby facilitating practical implementation. Subsequently, this increases the viability of extension headers as the extensibility mechanism of IPv6.

1.1. Related work

Some of the problems of unlimited extension headers have been addressed in certain aspects.

[RFC8200] relaxed the requirement that all nodes in the path must process Hop-by-Hop Options to be:

NOTE: While [RFC2460] required that all nodes must examine and process the Hop-by-Hop Options header, it is now expected that nodes along a packet's delivery path only examine and process the Hop-by-Hop Options header if explicitly configured to do so.

Section 5.3 of [RFC8504] defines a number of limits that hosts may apply to processing extensions. For instance:

A host MAY set a limit on the maximum number of non-padding options allowed in the destination options and Hop-by-Hop extension headers. If this feature is supported, the maximum number SHOULD be configurable, and the default value SHOULD be set to 8.

[RFC8883] defines a set of ICMP errors that may be sent if a limit concerning extension headers is exceeded and a node discards a packet as a result. This RFC allows both hosts and routers to send such messages (effectively acknowledging that some routers drop packets with extension headers even though such behavior is non-conformant).

[RFC7872] presents real-world data regarding the extent to which packets with IPv6 Extension Headers (EHs) are dropped in the Internet, and [I-D.gont-v6ops-ipv6-ehs-packet-drops] summarizes the operational implications of IPv6 extension headers, and attempts to analyze reasons why packets with IPv6 extension headers are often dropped in the public Internet.

1.2. Adherence to the Robustness Principle

The robustness principle, or Postel's Law, can be stated as "Be conservative in what you send, liberal in what you receive". This section considers the limits defined in this specification with respect to the robustness principle.

1.2.1. Be conservative in what you send

The limits on sending extension headers are well aligned with the send clause of the robustness principle. A sender of extension headers is generally constrained in its use of extension headers. Most of these limits are assumed to be the default to apply in an arbitrary environment such as the public Internet, that is they can be considered "baseline limits". These limits may be relaxed if a sender has a priori information that all possible nodes in path will properly handle packets that exceed the baseline limits. In particular, if a sender is sending in a limited domain, it might be known that all nodes in the limited domain have sufficient capabilities to handle packets exceeding the baseline limits.

1.2.2. Be liberal in what you receive

Considering the receive clause of the robustness principle, this specification recommends that receivers accept all packets with extension headers, however they may ignore extension headers or

options within extension headers. In particular, the philosophy of this specification is that intermediate nodes should not drop packets on the basis that they don't have sufficient capabilities to process all the headers in a packet. As such, intermediate nodes may define arbitrarily restrictive limits on what they process with regards to extension headers as long as the action taken when those limits are exceeded is to ignore items beyond the limit. Hosts are more constrained in this regard since they generally can't correctly process a packet without processing all the headers, so when limits are exceeded on a host, packets are dropped. It should be noted that hosts stacks inherently have more processing capabilities than intermediate nodes, so it is expected that they should be able to support higher limits.

This specification does specify one hard requirement for receiving nodes, namely nodes must be able to properly handle packets having an IPv6 header chain length up to 104 bytes. This requirement acknowledges that some intermediate nodes perform deep packet inspection at least to extract information from the transport layer headers. In this case, the data exceeding the limit may contain information that the node considers critical for correct processing, so that data cannot be ignored.

2. Overview of extension header limits

This specification considers extension header limits in three dimensions: 1) The types of nodes that may process extension headers and the requirements specific to each type, 2) The types of limits that may be applied, 3) The action taken when a limit is exceeded.

2.1. Types of nodes

For the purposes of describing handling of extension headers this specification considers three types of node in an IPv6 network:

- * Hosts: The source of an IPv6 packet, as addressed by the source address, or the final destination node of a packet as addressed by the destination address in a packet with no Routing header or as addressed by final segment in a Routing header.
- * Intermediate destination: An intermediate destination node in a Routing header as addressed by the destination address of a packet with a Routing header where the address is not the final destination in the Routing header
- * Intermediate nodes: A router on the path that is not addressed by the packet's destination address.

2.2. Types of limits

The limits and requirements for handling extension headers defined in this specification fall in the following categories:

- * Limits on extension header length
- * Limits on option length
- * Limits on number of extension headers
- * Limits on number of options
- * Limits on padding for extension headers with options
- * Limits on the length of the IPv6 header chain

2.2.1. Limits on extension header length

[RFC8504] defines limits that may be defined for the length of an extension header. Those limits are extended to be applicable to intermediate nodes. [RFC8883] defines ICMP Parameter Problem codes that may be sent when an extension header is exceeded.

2.2.2. Limits on option length

A node may establish a limit on the size Hop-by-Hop or Destination options. Conceivably, such a limit could apply to all option types, or length limits may be specific to individual options. [RFC8883] defines ICMP Parameter Problem codes that may be sent when an option length limit is exceeded.

2.2.3. Limits on number of extension headers

A node may define a limit on the number of extension headers it will process. Although [RFC8200] only defines four types of extension headers, it does not preclude the same type of extension header being present multiple times. A limit on the number of extension headers could be useful to disallow packets that contain multiple instances of the same extension header.

2.2.4. Limits on number of options

Limits may be established for the number of options sent or received (specifically applicable to Hop-by-Hop options and Destination options). The need for this limit arises from the fact that [RFC8200] does not specify a limit. Requiring nodes to process packets with tens or hundreds of options has no foreseeable use cases

in deployment except as a denial of service attack. [RFC8504] has proposed such a limit for host processing of Hop-by-Hop and Destination options with a default of eight options. This specification extends that limit to be applicable to intermediate nodes. Specific limits may be established for number of non-padding options or the number of all options including padding.

To derive a limit on all options, one can assume that at most one padding option is used between two non-padding options (an explicit limit on consecutive padding options is described below). With this assumption, we can extrapolate a reasonable limit on the number of all options that should be twice the limit of the number of non-padding options. Per [RFC8504], the recommended default limit for number of non-padding options is eight, so this specification establishes a maximum default limit of sixteen options including padding options. The choice of sixteen options as a default limit attempts to strike a balance between allowing extensibility and maintaining reasonable expectations for node processing requirements.

With regards to extensibility, it is observed that in the almost thirty year history of IPv6 there are only thirteen defined non-deprecated Destination options and Hop-by-Hop options and three temporary assigned options. Current evidence suggests that having more than one Destination option or Hop-by-Hop option in a packet is rare, and extrapolating that point with the rate of new options being defined suggests a limit of eight non-padding options allows for sufficient extensibility in the foreseeable future.

With regards to processing requirements, TLVs, e.g. Hop-by-Hop options and Destination options, have historically been considered difficult to process efficiently due to their serial processing requirements and combinatorial nature. TLV processing has been a particularly acute problem for ASIC devices. Recently, there is a strong trend in programmable implementation even in high performance routers using emerging programming frameworks such as PANDA and P4. Programmable implementations are better equipped to handle TLVs, at least for a reasonably small number. It might also be pointed out that the need to efficiently process TLVs exists in other protocols, for instance processing TCP requires processing of TLVs which are an intrinsic part of the protocol.

2.2.5. Limits on padding options

[RFC8200] defines PAD1 and PADN options that respectively provide one byte or N bytes of padding in an extension header. The purpose of padding is to properly align the following non-padding option to its expected alignment, or to add padding after the last Destination or Hop-by-Hop option so that the length of the extension header is a

multiple of eight bytes as required by [RFC8200]. [RFC8504] defines limits on number of bytes used for consecutive padding where the amount of padding between options or at the end of the extension header is no more than eight bytes; this limit is sufficient to align any following data after the padding to eight bytes. These limits are extended to be applicable to intermediate nodes.

This specification allows a receiving node to set a requirement that consecutive padding options are not present in a packet; which in turn requires a sender to not place consecutive padding options in a packet. The rationale for this limit is that a PAD1 or PADN option is able to provide one to 257 bytes of padding, so a single padding option is sufficient for any expected use case of padding. When the sender creates options, it can compute the amount of padding necessary to satisfy the alignment requirements of the following data. If one byte of padding is needed a PAD1 option is used, if more than one byte of padding is needed then an appropriate PADN options.

2.2.6. Limit on IPv6 header chain length

Intermediate nodes often perform deep packet inspection (DPI) in order to implement various functions in the network. Routers perform DPI when they inspect packets beyond the IPv6 header or beyond Hop-by-Hop options if they are present. Some router implementations must inspect the transport layer headers in order to process and forward the packet, and if the transport layer headers are not readable a packet may be dropped. Even if a transport layer header is in plain text within a packet, some devices may not be capable of reading it if the header is too deep in the packet.

Hardware devices often have constraints on how much of the headers in a packet can be parsed for DPI. A typical design is that some portion of the beginning of a received packet is loaded into a memory buffer for header parsing (i.e. the parsing buffer). The size of this parsing buffer is often fixed per device.

To derive a size limit on the IPv6 header chain, we need to take into account headers in a packet that might be subject to DPI which include the link layer header through at least the pertinent fields of the transport layer header. The most common required information is the transport layer port numbers which typically occupy the first four bytes of the transport headers (e.g. TCP, UDP, SCTP, DCCP, etc.). Inspection of port numbers may be needed for stateless load balancing as well as port filtering. There are middleboxes that may need to inspect more of transport layer headers or the transport payload, however those can be considered specialized devices that

perform work beyond simple packet forwarding and filtering and hence should have more capabilities for DPI.

In addition to limits on the length of the IP header chain, it is conceivable that there could be a limit on the length of the whole header chain. The whole header chain would comprise the IPv6 header chain as well as any headers that are part of network encapsulation that precede the innermost transport layer. The definition of such a limit is out of scope for this document, however [RFC8883] defines an ICMP error to send when a limit on size of an aggregate header chain is exceeded.

This document specifies that the minimum supported limit for IPv6 header chains is 104 bytes. The value is derived by assuming that nodes have the ability to process at least the first 128 bytes of a packet (that is they have a parsing buffer that can contain at least 128 bytes). The 128 byte parsing buffer would be expected to at least contain:

- * 16 bytes for a Layer 2 header (e.g. Ethernet header)
- * 40 bytes for the IPv6 header
- * 64 bytes for the extension headers
- * 8 bytes for the transport layer (i.e the first eight bytes of the transport layer header)

This scheme thus establishes a requirement that all Internet devices are capable of correctly processing packets with up to sixty-four bytes of extension headers, and subsequently it establishes a requirement that a host shouldn't send packets with more than sixty-four bytes of extension headers. Note that this establishes a global baseline requirement across the Internet, within a limited domain higher limits could be applied.

128 bytes is likely the minimal useful parsing buffer size in deployment today. Devices performing a very narrow DPI could conceptually use a smaller parsing buffer, for instance that could be as small as sixty-four bytes which accommodates an L2 header, IPv6 header, and eight bytes of transport header; however, such a device would be extremely limited in capabilities and if they do exist they are likely legacy devices that will eventually be decommissioned. Many routers now have the capability to perform DPI into encapsulation headers which implies they already have a larger parsing buffer than this baseline minimum.

Similar to limiting the number of options allowing in a packet, setting a limit for IP header length chain is a tradeoff between extensibility and feasible implementation.

For extensibility, the pertinent extension headers contributing to the sixty-four byte limit are mostly Hop-by-Hop and Destination options. The Routing Header extension header is really intended for limited domains and not the Internet (e.g. SRv6 Routing Header is confined to a Segment Routing Domain) and therefore would be subject to a domain specific limit for IP header chain length. Encryption Header may be used on the Internet, however encryption obfuscates the encapsulated transport headers such that intermediate nodes can't inspect them regardless of their position in a packet. Fragmentation may be used in the Internet, however only the first fragment of a fragmented packet might contain transport layer headers that could be read by an Intermediate node. In any case, the Fragment Header is only four bytes so that would not be a particularly large portion of a sixty-four byte limit.

The Authentication Header is usable on the Internet and does allow the transport layer headers to be in readable in plain text. However, Authentication Header is relatively large, typically thirty-two bytes or more, so it would contribute significantly to a limit on IP header chain length. On the other hand, the use of Authentication Header, without encryption, is currently rare on the Internet.

Individual Hop-by-Hop Destination Options may also be categorized as being intended for use over the Internet or just in limited domains. For instance, the IOAM Hop-by-Hop option is intended for use in limited domains.

Paring this down, the types extension headers and Destination and Hop-by-Hop options that might be used outside of limited domains are fairly limited. Options that are intended for use over the public Internet could be defined to be small and compact to promote not exceeding a sixty-four byte limit on extension headers, whereas options constrained to a limited domain could be larger since larger limits can be assumed.

2.2.6.1. Action when limit is exceeded

For each limit that is defined, an action is specified for when the limit is exceeded. The appropriate action depends on whether the processing node is the destination host, an intermediate destination, or an intermediate node. For a destination host, the typical action to take when a limit is exceeded is to discard the packet. This is appropriate since the destination host is required to process all of

the headers in a packet, and if a limit is exceeded then it cannot process the packet so there is no other alternative but to discard.

For intermediate nodes, the typical action to take when a limit is exceeded is to stop processing headers at the point the limit is reached and to forward the packet on. [RFC8200] allows that an intermediate may not process the Hop-by-Hop Options extension headers therefore an intermediate node may ignore all of the Hop-by-Hop options in a packet. This specification expands on that requirement to allow an intermediate node to process some arbitrary subset of consecutive Hop-by-Hop options in the TLV list and to ignore the following ones. In the case of an egregious violation of a limit, for instance an attacker sends three hundred options in a packet, the destination host can decide if the appropriate response is to drop (the destination host must process all options). Note that this provision motivates the sender to place Hop-by-Hop Options in the packet so that those considered more important are placed first. It should also be noted that [RFC8200] sets a default limit of eight; this specification adds a counterpart for sending hosts that they shouldn't send more than eight Hop-by-Hop options.

Intermediate destinations have characteristics of both hosts and intermediate nodes. If a limit is exceeded related to Hop-by-Hop options then the suggested action in this specification is to assume the same processing of limits as intermediate nodes. If limits are exceeded that affect the processing specific to an intermediate destination, such as limits on Destination options before the Routing header, then the action should be to discard packet.

2.3. Requirements for extension header limits

The set of limits that a node may apply when processing extension headers include:

- * Too many non-padding or padding options
- * Extension header too big
- * Option too big
- * Too many consecutive padding options
- * Too many consecutive bytes of padding
- * Extension header chain too long
- * Aggregate header chain too long

- * Too many extension headers

3. Requirements

This section lists the normative requirements related to sending and processing extension headers.

3.1. Host requirements

3.1.1. Sending extension headers

The requirements are:

- * A host MUST NOT send more than 8 non-padding options in Destination Options in a packet unless it has explicit knowledge that the destination, or all intermediate destinations in the case of Destination Options before the routing header, are able to process a greater number of options.
- * A host MUST NOT send more than 8 non-padding options in Hop-by-Hop Options in a packet unless it has explicit knowledge that the final destination host is able to process a greater number of options.
- * A host SHOULD NOT send more than 8 non-padding options in Hop-by-Hop Options in a packet unless it has explicit knowledge that all possible intermediate nodes are able to process a greater number of options or will ignore options that exceeds their limit.
- * A host MUST NOT send a packet with an extension header larger than 64 bytes unless it has explicit knowledge that all nodes that might process the extension header are capable of processing a larger header.
- * A host MUST NOT send a packet with a Destination option or Hop-by-Hop option with Data Length greater than 60 bytes unless it has explicit knowledge that all nodes that might process the option are capable of processing ones with a larger Data Length.
- * A host node MUST NOT send a packet with an IPv6 header chain larger than 104 bytes unless it has explicit knowledge that all nodes in the path are capable of properly handling packets with larger header chains. This requirements is equivalently stated as a host MUST NOT send a packet with more than 64 bytes of aggregate extension headers.
- * A host MUST NOT set more than one consecutive pad option, either PAD1 or PADN, in Destination options or Hop-by-Hop options.

- * A host MUST NOT send a PadN option in Hop-by-Hop Options or Destination Options with total length of more than seven bytes.
- * A host node MUST NOT send more than 16 options (padding or non-padding) Destination options in a packet unless it has explicit knowledge that the destination, or all intermediate destinations in the case of Destination Options before the routing header, are able to process a greater number of options. Note that if the above requirements on a host sending non-padding Destination options and requirements on option padding are met, then this requirement is implicitly satisfied.
- * A host node MUST NOT send more than 16 options (padding or non-padding) in Hop-by-Hop Options in a packet unless it has explicit knowledge that the final destination host is able to process a greater number of options. Note that if the above requirements on a host sending non-padding Hop-by-Hop options and requirements on padding are met, then this requirement is implicitly satisfied.

3.1.2. Receiving extension headers

Per [RFC8200], a host node that receives a packet with extension headers must process all the extension headers in the packet before accepting the payload and processing the payload.

As described in [RFC8504] a host may establish limits on the processing of extension headers. This specification reiterates and updates those requirements to allow for a host to send an RFC8883 error if a limit has been exceeded.

- * A host MAY set a limit on the maximum number of non-padding options allowed in the Destination Options or Hop-by-Hop Options extension headers. If this limit is supported then the maximum number SHOULD be configurable, the limit MUST be greater than or equal to 8, and the default value SHOULD be set to 8. The limits for Destination options and Hop-by-Hop options MAY be separately configurable. If a packet is received and the number of Destination or Hop-by-Hop options exceeds the limit, then the packet SHOULD be discarded and an ICMP Parameter Problem with code 9 MAY be sent to the packet's source address.
- * A host MAY set a limit on the maximum number of options (padding or non-padding) allowed in Destination Options or Hop-by-Hop Options extension headers. If this limit is supported then the maximum number SHOULD be configurable and the limit MUST be greater than or equal to 16. The limits for Destination options and Hop-by-Hop options MAY be separately configurable. If a packet is received and the number of destination or Hop-by-Hop

options exceeds the limit, then the packet SHOULD be discarded and an ICMP Parameter Problem with code 9 MAY be sent to the packet's source address

- * A host node MAY set a limit on the length of an extension header. If this limit is supported then the limit SHOULD be configurable and the limit MUST be greater than or equal to 64 bytes. The length limits for different extension headers MAY be separately configurable.
- * A host node MAY set a limit on the Data Length of a Hop-by-Hop or Destination option. If this limit is supported then the limit SHOULD be configurable, and the limit MUST be greater than or equal to 60 bytes. The limits for Destination options and Hop-by-Hop options MAY be separately configurable. If a packet is received and a Hop-by-Hop or destination option has a length that exceeds the limit, then the packet SHOULD be discarded and an ICMP Parameter Problem with code 10 MAY be sent to the packet's source address.
- * A host MAY limit the number of consecutive PAD1 options in destination options or Hop-by-Hop options to 7. In this case, if there are more than 7 consecutive PAD1 options present, the packet SHOULD be discarded and an ICMP Parameter Problem with code 10 MAY be sent to the packet's source address
- * A host MAY limit the number of bytes in a PADN option to be less than 8. In such a case, if a PADN option is present that has a length greater than 7, the packet SHOULD be discarded and an ICMP Parameter Problem with code 10 MAY be sent to the packet's source address.
- * A host MAY set a limit on the maximum length of Destination Options or Hop-by-Hop Options extension headers. This value SHOULD be configurable, and if the limit is used then the limit MUST be greater than or equal to 64 bytes. If a packet is received and the length of the Destination or Hop-by-Hop Options extension header exceeds the length limit, then the packet SHOULD be discarded and an ICMP Parameter Problem with code 6 MAY be sent to the packet's source address.
- * A host node MAY set a limit on the maximum length of the IPv6 header chain, or equivalently a host MAY set a limit on the aggregate length of extension headers in a packet. If the limit is used then it MUST be greater than or equal to 104 bytes, or, equivalently, the limit on aggregate header extension length MUST be greater than or equal to 64 bytes. If a packet is received and the aggregate length of the IPv6 header chain exceeds the limit

then the packet SHOULD be discarded and an ICMP Parameter Problem with code 7 MAY be sent to the packet's source address.

Additional host requirements for receive.

- * A host MAY disallow consecutive padding options, either PAD1 or PADN, to be present in a packet. If consecutive padding options are received and disallowed by the host, the then packet SHOULD be discarded and an ICMP Parameter Problem with code 9 MAY be sent to the packet's source address.

3.2. Intermediate node and intermediate destination requirements

The following requirements are established for intermediate nodes and intermediate destination nodes that receive and process packets with extension header.

- * An intermediate node MUST be able to correctly forward packets that contain an IPv6 header chain of 104 or fewer bytes, or equivalently an intermediate node MUST be able to process a packet with an aggregate length of extension headers less than or equal to 64 bytes.
- * Per [RFC8200] an intermediate node MAY be configured to not process Hop-by-Hop Options. If a node is configured as such and a packet with Hop-by-Hop options is received, the extension header MUST be skipped and the packet MUST otherwise be properly processed and forwarded.
- * An intermediate node MAY limit the number of non-padding Hop-by-Hop options that it processes. If a limit is exceeded, that is a packet contains more non-padding options than are configured to process, the intermediate SHOULD stop processing the Hop-by-Hop Option and ignore any options in the chain beyond the limit. It is NOT RECOMMENDED that an intermediate node discards the packet because the limit is exceeded, however if it does so then the intermediate node MAY send an ICMP Parameter Problem with code 10 MAY be sent to the packet's source address.
- * An intermediate node MAY limit the number of Hop-by-Hop options (padding or non-padding) that it processes. If a limit is exceeded, that is a packet contains more non-padding options than are configured to process, the intermediate SHOULD stop processing the Hop-by-Hop options and ignore any options in the chain beyond the limit. It is NOT RECOMMENDED that the intermediate node discards the packet because the limit is exceeded, however if it does so then the intermediate node MAY send an ICMP Parameter Problem with code 10 MAY be sent to the packet's source address.

- * If an intermediate node encounters an unknown Hop-by-Hop option and the two high order bits are not 00 then the node SHOULD immediately stop processing the option chain and ignore any options in the chain beyond the unknown option. An intermediate node MAY either elect to discard the packet and MAY send an ICMP Parameter Problem per the requirements of [RFC8200]; or the intermediate node MAY forward the packet.
- * An intermediate node MAY set a limit on the maximum length of Hop-by-Hop Options extension headers. This value SHOULD be configurable. If this limit is exceeded, that is a packet has an extension header larger than the limit, then the intermediate SHOULD stop processing the Hop-by-Hop Option and ignore any options in the chain beyond the limit. It is NOT RECOMMENDED that the intermediate node discards the packet because the limit is exceeded, however if it does so then the intermediate node MAY send an ICMP Parameter Problem with code 10 MAY be sent to the packet's source address.

3.3. Intermediate destination requirements

The following are requirements specific to intermediate destinations pertaining to the processing of Destination Options before the Routing header.

- * An intermediate destination MAY set a limit on the maximum length of Destination Options extension header before the Routing header. This value SHOULD be configurable, and the default is to accept options of any length. If a limit is defined is MUST be at least 64 bytes. If the limit is exceeded then the intermediate destination SHOULD discard the packet and MAY send an ICMP Parameter Problem with code 6 to the packet's source address.
- * An intermediate destination node MAY limit the number of non-padding options in Destination Options before the Routing header. If a limit is exceeded, that is a packet contains more non-padding options than are configured to process, the intermediate destination node SHOULD discard the packet and MAY send an ICMP Parameter Problem with code 10 to the packet's source address.
- * An intermediate destination node MAY limit the number of options (padding or non-padding) in Destination Options before the Routing header. If a limit is exceeded, that is a packet contains more non-padding options than are configured to process, the intermediate destination node SHOULD discard the packet and MAY send an ICMP Parameter Problem with code 10 to the packet's source address.

- * An intermediate destination MAY limit the total number bytes in consecutive PAD1 options in destination options before the Routing Header 7. If the limit is exceeded, that is there are more than seven bytes in consecutive PAD1 or PADN options present, the intermediate destination node SHOULD discard the packet and MAY send an ICMP Parameter Problem with code 10 to the packet's source address.
- * A intermediate destination MAY limit the number of bytes in a PADN option in Destination Options before the Routing header to be less than 8. In such a case, if a PADN option is present that has a length greater than 7, the packet SHOULD be discarded and the intermediate destination node SHOULD discard the packet and MAY send an ICMP Parameter Problem with code 10 to the packet's source address.
- * A intermediate destination MAY set a limit on the maximum number of non-padding options allowed in Destination options before the Routing header. If this feature is supported, the maximum number SHOULD be configurable, and the default value SHOULD be set to 8. If a packet is received and the number of Destination options before the Routing header exceeds the limit, the intermediate destination node SHOULD discard the packet and MAY send an ICMP Parameter Problem with code 10 to the packet's source address.
- * A intermediate MAY set a limit on the maximum length of Destination Options extension header before the Routing header. This value SHOULD be configurable, and the default is to accept options of any length. If a packet is received and the length of the Destination or Hop-by-Hop Options extension header exceeds the length limit, the intermediate destination node SHOULD discard the packet and MAY send an ICMP Parameter Problem with code 10 to the packet's source address.

4. References

4.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

- [RFC8504] Chown, T., Loughney, J., and T. Winters, "IPv6 Node Requirements", BCP 220, RFC 8504, DOI 10.17487/RFC8504, January 2019, <<https://www.rfc-editor.org/info/rfc8504>>.
- [RFC8883] Herbert, T., "ICMPv6 Errors for Discarding Packets Due to Processing Limits", RFC 8883, DOI 10.17487/RFC8883, September 2020, <<https://www.rfc-editor.org/info/rfc8883>>.

4.2. Informative References

- [I-D.gont-v6ops-ipv6-ehs-packet-drops]
Gont, F., Hilliard, N., Doering, G., Kumari, W., and G. Huston, "Operational Implications of IPv6 Packets with Extension Headers", draft-gont-v6ops-ipv6-ehs-packet-drops-04 (work in progress), July 2020.
- [RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <<https://www.rfc-editor.org/info/rfc7872>>.

Author's Address

Tom Herbert
SiPanda
Santa Clara, CA
USA

Email: tom@sipanda.io

Network Working Group
Internet-Draft
Updates: 8200 (if approved)
Intended status: Standards Track
Expires: 4 December 2021

R. Hinden
Check Point Software
G. Fairhurst
University of Aberdeen
2 June 2021

IPv6 Hop-by-Hop Options Processing Procedures
draft-hinden-6man-hbh-processing-01

Abstract

This document specifies procedures for how IPv6 Hop-by-Hop options are processed. It modifies the procedures specified in the IPv6 Protocol Specification (RFC8200) to make processing of IPv6 Hop-by-Hop options practical with the goal of making IPv6 Hop-by-Hop options useful to deploy and use in the Internet. When published, this document updates RFC8200.

Status of This Memo

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

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

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

This Internet-Draft will expire on 4 December 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Requirements Language	2
3. Terminology	2
4. Background	3
5. Hop-by-Hop Header Processing Procedures	5
5.1. Hop-by-Hop Options Per Packet	6
5.2. Hop-by-Hop Headers Processing	6
5.3. Router Alert Option	7
5.4. Configuration	8
6. New Hop-by-Hop Options	9
7. IANA Considerations	9
8. Security Considerations	10
9. Acknowledgments	11
10. Change log [RFC Editor: Please remove]	11
11. Normative References	11
12. Informative References	12
Authors' Addresses	12

1. Introduction

This document specifies procedures for how IPv6 Hop-by-Hop options are processed. It modifies the procedures specified in the IPv6 Protocol Specification (RFC8200) to make processing of IPv6 Hop-by-Hop options practical with the goal of making IPv6 Hop-by-Hop options useful to deploy and use in the Internet.

When published this document updates [RFC8200].

The current list of defined Hop-by-Hop options can be found at [IANA-HBH].

2. Requirements Language

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

3. Terminology

This document uses the following loosely defined terms:

- * Forwarding Plane: IPv6 hosts exchange user data through the forwarding plane. User data is processed by its recipient (i.e., an IPv6 host). User data can traverse intermediate nodes (i.e., routers) between its source and its destination. These intermediate nodes process metadata contained in packet headers. However, they do not process information contained in packet payloads.
- * Control Plane: IPv6 routers exchange management and routing information with controllers. They also exchange routing information with one another. Management and routing information is processed by its recipient (i.e., an IPv6 router or controller). Management and control information can traverse intermediate nodes (i.e., routers) between its source and its destination. These intermediate nodes process metadata contained in packet headers. However, they do not process information contained in packet payloads. So, from their perspective, this information is user data.
- * Fast Path: A path through a router that is optimized for forwarding packets without processing their payloads. The Fast Path may be supported by Application Specific Integrated Circuits (ASICs), Network Processor (NP), or other special purpose hardware. This is the usual processing path within a router taken by the forwarding plane.
- * Slow Path: A path through a router that is capable of general purpose processing and is not optimized for any particular function. This processing path is used for packets that require special processing or differ from assumptions made in Fast Path heuristics, or to process router control protocols used by the control plane.

NOTE: This distinct separation between hardware and software processing from [RFC6398] does not apply to all router architectures. However, a router that performs all or most processing in software might still incur more processing cost when providing special processing (aka Slow Path).

[RFC6192] is an example of how designs can separate control plane (Slow Path) and forwarding plane (Fast Path) functions.

4. Background

In the first version of the IPv6 specification, Hop-by-Hop options were required to be processed by all nodes: routers and hosts. This proved to not be practical in high speed routers due to several factors, including:

- * Inability to process the hop-by-hop options at wire speed on the Fast Path.
- * Hop-by-Hop options would be sent to the Slow Path. This could degrade the a router's performance and it's ability to process important control traffic.
- * A mechanism that forces packets from any source to the routers "Slow Path" could be exploited as a Denial of Service attack against the router.
- * Packets could contain multiple Hop-by-Hop options making the previous issues worse by increasing the complexity required to process them.

When the IPv6 Specification was updated and published in July 2017 as [RFC8200], the procedures relating to hop-by-hop options were as follows:

Extension headers (except for the Hop-by-Hop Options header) are not processed, inserted, or deleted 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.

The Hop-by-Hop Options header is not inserted or deleted, but may be 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. The Hop-by-Hop Options header, when present, must immediately follow the IPv6 header. Its presence is indicated by the value zero in the Next Header field of the IPv6 header.

NOTE: While [RFC2460] required that all nodes must examine and process the Hop-by-Hop Options header, it is now expected that nodes along a packet's delivery path only examine and process the Hop-by-Hop Options header if explicitly configured to do so.

The changes meant that an implementation complied with the IPv6 specification even if it did not process hop-by-hop options, and that it was expected that routers would add configuration information to control which hop-by-hop options they would process.

Unfortunately, this did not improve the processing of Hop-by-Hop options and did not significantly improve deployment and use in the Internet. Essentially, it only documented how they were being used in the Internet at the time RFC8200 was published.

The main issues remain:

- * Routers are commonly configured to drop transit packets containing hop-by-hop options that would have be processed in the Slow Path. This behavior is seen as protecting against a denial of service attack on the router. This is discussed in [I-D.ietf-v6ops-ipv6-ehs-packet-drops].
- * Allowing multiple hop-by-hop options in a single packet makes it even more expensive in router resources to process these packets. It adds complexity to the number of permutations that might need to be processed.
- * Any mechanism that can be used to force packets into the router's Slow Path can be exploited as a denial of service attack on a transit router by saturating the resources needed for router management protocols (e.g., routing protocols, network management protocols, etc.) that may cause the router to fail. This issue for the Router Alert option, which intentionally places packets on the Slow Path, is discussed in [RFC6398]. Section 3 of that RFC includes a good summary:

"In a nutshell, the IP Router Alert Option does not provide a convenient universal mechanism to accurately and reliably distinguish between IP Router Alert packets of interest and unwanted IP Router Alert packets. This, in turn, creates a security concern when the IP Router Alert Option is used, because, short of appropriate router-implementation-specific mechanisms, the router Slow Path is at risk of being flooded by unwanted traffic."

There has been research that discussed the general problem with dropping packets containing IPv6 extension headers, including the Hop-by-Hop Options header. For example [Hendriks] states that "dropping all packets with Extension Headers, is a bad practice", and that "The share of traffic containing more than one EH however, is very small. For the design of hardware able to handle the dynamic nature of EHs, we therefore recommend to support at least one EH".

This document defines a set of procedures for the hop-by-hop option header that make the processing of hop-by-hop options practical in modern transit routers.

5. Hop-by-Hop Header Processing Procedures

This section describes several changes to [RFC8200].

5.1. Hop-by-Hop Options Per Packet

The Hop-by-Hop Option Header as defined in Section 4.3 of [RFC8200] is identified by a Next Header value of 0 in the IPv6 header. Section 4.1 of [RFC8200] requires a Hop-by-Hop Options header to appear immediately after the IPv6 header. [RFC8200] also requires that a Hop-by-Hop Options header can only appear once in a packet.

The Hop-by-Hop Options Header as defined in [RFC8200] can contain one or more Hop-by-Hop options. This document updates [RFC8200] that a node **MUST** process the first Option in the Hop-by-Hop Header in the Fast Path and **MAY** process additional Hop-by-Hop Options if configured to do so. The motivation for this change is to simplify the processing of Hop-by-Hop options in the Fast Path.

Nodes creating packets with a Hop-by-Hop option headers **SHOULD** include a single Hop-by-Hop Option in the packet and **MAY** include more based on local configuration.

If there are more than one Hop-by-Hop options in the Hop-by-Hop Options header, the node **MAY** skip the rest of the options without having to examine these options using the "Hdr Ext Len" field in the Hop-by-Hop Options header. This field specifies the length of the Option Header in 8-octet units. The additional options do not need to be processed or verified.

5.2. Hop-by-Hop Headers Processing

Nodes that implement a differentiation between a Fast Path and a Slow Path **MUST** process all (with one exception noted below) Hop-by-Hop options in the Fast Path. The one exception to this is the Router Alert Option [RFC2711]. See Section 5.3 for discussion of the Router Alert.

If the node can not process an option in the Fast Path, it **MUST** behave as if it does not recognize the Option Type (as described in the next paragraph).

Section 4.2 of [RFC8200] defines the Option Type identifiers as internally encoded such that their highest-order 2 bits specify the action that must be taken if the processing IPv6 node does not recognize the Option Type. The text is:

- 00 - skip over this option and continue processing the header.
- 01 - discard the packet.
- 10 - discard the packet and, regardless of whether or not the packet's Destination Address was a multicast address, send an ICMP Parameter Problem, Code 2, message to the packet's Source Address, pointing to the unrecognized Option Type.
- 11 - discard the packet and, only if the packet's Destination Address was not a multicast address, send an ICMP Parameter Problem, Code 2, message to the packet's Source Address, pointing to the unrecognized Option Type.

This document modifies this behaviour for the "10" and "11" values that the node MAY send an ICMP Parameter Problem, Code 2, message to the packet's Source Address, pointing to the unrecognized Option Type. The modified text for "10" and "11" values is:

- 10 - discard the packet and, regardless of whether or not the packet's Destination Address was a multicast address, MAY send an ICMP Parameter Problem, Code 2, message to the packet's Source Address, pointing to the unrecognized Option Type.
- 11 - discard the packet and, only if the packet's Destination Address was not a multicast address, MAY send an ICMP Parameter Problem, Code 2, message to the packet's Source Address, pointing to the unrecognized Option Type.

The motivation for this change is to loosen the requirement to send ICMPv6 Parameter Problem messages to simplify what the router needs to do in the Fast Path when it does not recognize the Option Type.

When an ICMP Parameter Problem, Code 2, message is delivered to the source, the source can become aware that at least one node on the path has failed to recognize the option.

5.3. Router Alert Option

The Router Alert option [RFC2711] purpose is to tell the node that the packet needs additional processing on the Slow Path.

The Router Alert option includes a two octet Value field that describes the protocol that is carried in the packet. The current values can be found in the IANA Router Alert Value registry [IANA-RA].

DISCUSSION

The Router Alert Option is a problem since it's function is to do what this specification is proposing to eliminate, that is, process the packet in the Slow Path. One approach would be to deprecate it as it's usage appears to be limited and packets containing Hop-by-Hop options are frequently dropped. Deprecation would allow current implementations to continue and it's use could be phased out over time.

The authors current thinking is that the Router Alert function may have reasonable potential use for new functions that have to be processed in the Slow Path. We think that keeping it as the single exception for Slow Path processing with the following restrictions is a reasonable compromise to allow future flexibility. These are compatible with Section 5 of [RFC6398].

A Fast Path implementation SHOULD verify that a Router Alert contains a protocol, as indicated by the Value field in the Router Alert option, that is configured as a protocol of interest to that router. A verified packet SHOULD be sent on the Slow Path for processing [RFC6398]. Otherwise, the router implementation SHOULD forward within the Fast Path (subject to all normal policies and forwarding rules). As specified in [RFC2711] the top two bits of Option Type for the Router Alert option are always set to "00" indicating the node should skip over this option and continue processing the header in this case.

Implementations of the IP Router Alert Option SHOULD offer the configuration option to simply ignore the presence of "IP Router Alert" in IPv4 and IPv6 packets" [RFC6398].

A node that is configured to process a Router Alert option using the Slow Path MUST protect itself from infrastructure attack that could result from processing on the Slow Path. This might include some combination of access control list to only permit from trusted nodes, rate limiting of processing, or other methods [RFC6398].

5.4. Configuration

Section 4 of [RFC8200] allows for a router to control it's processing of IPv6 Hop-by-Hop options by local configuration. The text is:

NOTE: While [RFC2460] required that all nodes must examine and process the Hop-by-Hop Options header, it is now expected that nodes along a packet's delivery path only examine and process the Hop-by-Hop Options header if explicitly configured to do so.

A possible approach to implementing this is to maintain a lookup table based on Option Type of the IPv6 options that are supported in the Fast Path. This would allow for a node to quickly determine if an option is supported and can be processed. If the option is not supported, then the node processes it as described in Section 5.2 of this document.

A node configured not to process HBH options, MUST drop the packet if the top two bits of the Option Type field of the first HBH option is non-zero.

The actions of the lookup table SHOULD be configurable by the operator of the router.

6. New Hop-by-Hop Options

Any new IPv6 Hop-by-Hop option designed in the future should be designed to be processed in the Fast Path. New options MUST NOT be defined that require Slow Path processing. New Hop-by-Hop options SHOULD have the following characteristics:

- * Straight forward to process. That is, they should be designed to keep the time to process low.
- * Fixed size in 8-octet units. Specifically any new Hop-by-Hop options should not be variable size that could extend beyond what can be executed in the Fast Path.

Any new Hop-by-Hop option that is standardized that does not meet these criteria needs to explain in detail in its specification why this can not be accomplished and that there is a reasonable expectation that it can be proceed in most Fast Path implementations.

7. IANA Considerations

There are no actions required for IANA defined in this document.

8. Security Considerations

Security issues with IPv6 Hop-by-Hop options are well known and have been documented in several places, including [RFC6398], [RFC6192], and [I-D.ietf-v6ops-ipv6-ehs-packet-drops]. The main issue, as noted in Section 4, is that any mechanism that can be used to force packets into the router's Slow Path can be exploited as a denial of service attack on a transit router by saturating the resources need for router management protocols (e.g., routing protocols, network management protocols, etc.) that may cause the router to fail. Due to this it's common for transit routers to drop packets with Hop-by-Hop options headers.

While Hop-by-Hop options are not required to be processed in the Slow Path, the Router Alert options is designed to do just that.

This document changes the way Hop-by-Hop options are processed in several ways that significantly reduces the attack surface. These changes include:

- * All Hop-by-Hop options (with one exception) must be processed in the Fast Path. Only one HBH Option MUST be processed and additional HBH Options MAY be processed based on local configuration.
- * Only the Router Alert option can be processed in the Slow Path, and the router must be configured to do so.
- * Added criteria to allow control over how Router Alert options are processed and that a node configured to support these options must protect itself from attacks using the Router Alert.
- * Limited the default number of Hop-by-Hop options that that can be in a packet to a single Hop-by-Hop option.
- * Additional Hop-by-Hop options MAY be included, based on local configuration. Although nodes only process these additional Hop-by-Hop Options if configured to do so.
- * Added restrictions to any future new Hop-by-Hop options that limit their size and computational requirements.

The authors believe that these changes significantly reduces the security issues relating to IPv6 Hop-by-Hop options and will enable them to be used safely in the Internet.

9. Acknowledgments

Helpful comments were received from Brian Carpenter, Ron Bonica, Ole Troan, Mark Heard, Tom Herbert, [your name here], and other members of the 6MAN working group.

10. Change log [RFC Editor: Please remove]

draft-hinden-6man-hbh-processing-01, 2021-June-2:

- * Expanded terminology section to include Forwarding Plane and Control Plane.
- * Changed draft that only one HBH Option MUST be processed and additional HBH Options MAY be processed based on local configuration.
- * Clarified that all HBH options (with one exception) must be processed on the Fast Path.
- * Kept the Router Alert options as the single exception for Slow Path processing.
- * Rewrote and expanded section on New Hop-by-Hop Options.
- * Removed requirement for HBH Option size and alignment.
- * Removed sections evaluating currently defined HBH Options.
- * Added content to the Security Considerations section.
- * Added people to the acknowledgements section.
- * Numerous editorial changes

draft-hinden-6man-hbh-processing-00, 2020-Nov-29:

- * Initial draft.

11. Normative References

- [IANA-HBH] "Destination Options and Hop-by-Hop Options",
<<https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml#ipv6-parameters-2>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200,

DOI 10.17487/RFC8200, July 2017,
<<https://www.rfc-editor.org/info/rfc8200>>.

12. Informative References

- [Hendriks] Hendriks, L., Velan, P., Schmidt, R.O., Boer, P., and A. Aiko, "Threats and Surprises behind IPv6 Extension Headers", August 2017, <http://dl.ifip.org/db/conf/tma/tma2017/tma2017_paper22.pdf>.
- [I-D.ietf-v6ops-ipv6-ehs-packet-drops] Gont, F., Hilliard, N., Doering, G., Kumari, W., Huston, G., and W. (. Liu, "Operational Implications of IPv6 Packets with Extension Headers", Work in Progress, Internet-Draft, draft-ietf-v6ops-ipv6-ehs-packet-drops-06, 8 April 2021, <<https://tools.ietf.org/html/draft-ietf-v6ops-ipv6-ehs-packet-drops-06>>.
- [IANA-RA] "IPv6 Router Alert Option Values", <<https://www.iana.org/assignments/ipv6-routeralert-values/ipv6-routeralert-values>>.
- [RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option", RFC 2711, DOI 10.17487/RFC2711, October 1999, <<https://www.rfc-editor.org/info/rfc2711>>.
- [RFC6192] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the Router Control Plane", RFC 6192, DOI 10.17487/RFC6192, March 2011, <<https://www.rfc-editor.org/info/rfc6192>>.
- [RFC6398] Le Faucheur, F., Ed., "IP Router Alert Considerations and Usage", BCP 168, RFC 6398, DOI 10.17487/RFC6398, October 2011, <<https://www.rfc-editor.org/info/rfc6398>>.

Authors' Addresses

Robert M. Hinden
Check Point Software
959 Skyway Road
San Carlos, CA 94070
United States of America

Email: bob.hinden@gmail.com

Godred Fairhurst
University of Aberdeen

School of Engineering, Fraser Noble Building
Aberdeen
AB24 3UE
United Kingdom

Email: gorry@erg.abdn.ac.uk

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: 11 November 2022

R. Hinden
Check Point Software
G. Fairhurst
University of Aberdeen
10 May 2022

IPv6 Minimum Path MTU Hop-by-Hop Option
draft-ietf-6man-mtu-option-15

Abstract

This document specifies a new IPv6 Hop-by-Hop option that is used to record the minimum Path MTU along the forward path between a source host to a destination host. The recorded value can then be communicated back to the source using the return Path MTU field in the option.

Status of This Memo

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

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

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

This Internet-Draft will expire on 11 November 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction	2
1.1. Example Operation	3
1.2. Use of the IPv6 Hop-by-Hop Options Header	4
2. Motivation and Problem Solved	5
3. Requirements Language	6
4. Applicability Statements	6
5. IPv6 Minimum Path MTU Hop-by-Hop Option	6
6. Router, Host, and Transport Layer Behaviors	8
6.1. Router Behavior	8
6.2. Host Operating System Behavior	8
6.3. Transport Layer Behavior	9
6.3.1. Including the Option in an Outgoing Packet	10
6.3.2. Validation of the Packet that includes the Option	12
6.3.3. Receiving the Option	12
6.3.4. Using the Rtn-PMTU Field	13
6.3.5. Detecting Path Changes	14
6.3.6. Detection of Dropping Packets that include the Option	14
7. IANA Considerations	14
8. Security Considerations	14
8.1. Router Option Processing	15
8.2. Network Layer Host Processing	15
8.3. Validating use of the Option Data	16
8.4. Direct use of the Rtn-PMTU Value	16
8.5. Using the Rtn-PMTU Value as a Hint for Probing	17
8.6. Impact of Middleboxes	17
9. Experiment Goals	17
10. Implementation Status	18
11. Acknowledgments	18
12. Change log [RFC Editor: Please remove]	18
13. References	21
13.1. Normative References	21
13.2. Informative References	22
Appendix A. Examples of Usage	24
Authors' Addresses	26

1. Introduction

This document specifies a new IPv6 Hop-by-Hop (HBH) Option to record the minimum Maximum Transmission Unit (MTU) along the forward path between a source and a destination host. The source host creates a packet with this option and initializes the Min-PMTU field with the value of the MTU for the outbound link that will be used to forward the packet towards the destination host.

At each subsequent hop where the option is processed, the router compares the value of the Min-PMTU Field in the option and the MTU of its outgoing link. If the MTU of the link is less than the Min-PMTU, it rewrites the value in the option data with the smaller value. When the packet arrives at the destination host, the host can send the value of the minimum reported MTU for the path back to the source host using the Rtn-PMTU field in the option. The source host can then use this value as input to the method that sets the Path MTU (PMTU) used by upper layer protocols.

The IPv6 Minimum Path MTU Hop-by-Hop (MinPMTU HBH) Option is designed to work with packet sizes that can be specified in the IPv6 header. The maximum packet size that can be specified in an IPv6 header is 65,535 octets (2^{16}).

This method has the potential to complete Path MTU discovery in a single round trip time, even over paths that have successive links each with a lower MTU.

The mechanism defined in this document is focused on Unicast, it does not describe Multicast. That is left for future work.

1.1. Example Operation

The figure below illustrates the operation of the method. In this case, the path between the source host and the destination host comprises three links, the source has a link MTU of size MTU-S, the link between routers R1 and R2 has an MTU of size 9000 bytes, and the final link to the destination has an MTU of size MTU-D.

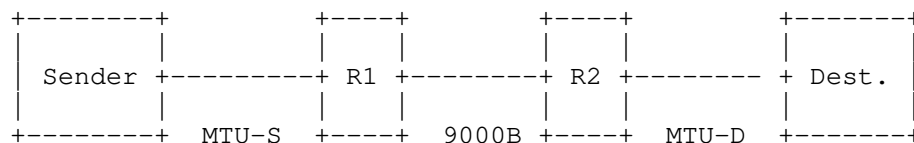


Figure 1

Three scenarios are described:

- * Scenario 1, considers all links to have an 9000 byte MTU and the method is supported by both routers. The initial Min-PMTU is not modified along the path, and therefore the PMTU is 9000 bytes.
- * Scenario 2, considers the link between R2 and destination host (MTU-D) to have an MTU of 1500 bytes. This is the smallest MTU, router R2 updates the Min-PMTU to 1500 bytes and the method

correctly updates the PMTU to 1500 bytes. Had there been another smaller MTU at a link further along the path that also supports the method, the lower MTU would also have been detected.

- * Scenario 3, considers the case where the router preceding the smallest link (R2) does not support the method, and the link to the destination host (MTU-D) has an MTU of 1500 bytes. Therefore, router R2 does not update the Min-PMTU to 1500 bytes. The method then fails to detect the actual PMTU.

In Scenarios 2 and 3, a lower PMTU would also fail to be detected in the case where PMTUD had been used and an ICMPv6 Packet Too Big (PTB) message had not been delivered to the sender [RFC8201].

These scenarios are summarized in the table below. "H" in R1 and/or R2 columns means the router understands the MinPMTU HBH option.

	MTU-S	MTU-D	R1	R2	Rec PMTU	Note
1	9000B	9000B	H	H	9000 B	Endpoints attempt to use a 9000 B PMTU.
2	9000B	1500B	H	H	1500 B	Endpoints attempt to use a 1500 B PMTU.
3	9000B	1500B	H	-	9000 B	Endpoints attempt to use a 9000 B PMTU, but need to implement a method to fall back to discover and use a 1500 B PMTU.

Figure 2

1.2. Use of the IPv6 Hop-by-Hop Options Header

IPv6 as specified in [RFC8200] allows nodes to optionally process the Hop-by-Hop header. Specifically, from Section 4:

- * The Hop-by-Hop Options header is not inserted or deleted, but may be 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. The Hop-by-Hop Options header, when present, must immediately follow the IPv6 header. Its presence is indicated by the value zero in the Next Header field of the IPv6 header.
- * NOTE: While [RFC2460] required that all nodes must examine and process the Hop-by-Hop Options header, it is now expected that nodes along a packet's delivery path only examine and process the Hop-by-Hop Options header if explicitly configured to do so.

The Hop-by-Hop Option defined in this document is designed to take advantage of this property of how Hop-by-Hop options are processed. Nodes that do not support this Option SHOULD ignore them. This can mean that the Min-PMTU value does not account for all links along a path.

2. Motivation and Problem Solved

The current state of Path MTU Discovery on the Internet is problematic. The mechanisms defined in [RFC8201] are known to not work well in all environments. It fails to work in various cases, including when nodes in the middle of the network do not send ICMPv6 PTB messages, or rate-limited ICMPv6 messages, or do not have a return path to the source host.

This results in many transport layer connections being configured to use smaller packets (e.g., 1280 bytes) by default and makes it difficult to take advantage of paths with a larger PMTU where they do exist. Applications that send large packets are forced to use IPv6 Fragmentation [RFC8200], which can reduce the reliability of Internet communication [RFC8900].

Encapsulations and network-layer tunnels further reduce the payload size available for a transport protocol to use. Also, some use-cases increase packet overhead, for example, Network Virtualization Using Generic Routing Encapsulation (NVGRE) [RFC7637] encapsulates L2 packets in an outer IP header and does not allow IP Fragmentation.

Sending larger packets can improve host performance, e.g., avoiding limits to packet processing by the packet rate. For example, the packet per second rate required to reach wire speed on a 10G link with 1280 byte packets is about 977K packets per second (pps), vs. 139K pps for 9000 byte packets.

The purpose of this document is to improve the situation by defining a mechanism that does not rely on reception of ICMPv6 Packet Too Big messages from nodes in the middle of the network. Instead, this provides information to the destination host about the minimum Path MTU, and sends this information back to the source host. This is expected to work better than the current RFC8201-based mechanisms.

A similar mechanism was proposed in 1988 for IPv4 in [RFC1063] by Jeff Mogul, C. Kent, Craig Partridge, and Keith McCloghrie. It was later obsoleted in 1990 by [RFC1191], the current deployed approach to Path MTU Discovery. In contrast, the method described in this document uses the Hop-by-Hop option of IPv6. It does not replace PMTUD [RFC8201], PLPPMTUD [RFC4821] or Datagram PLPMTUD [RFC8899], but rather is designed to compliment these methods.

3. Requirements Language

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

4. Applicability Statements

The Path MTU option is designed for environments where there is control over the hosts and nodes that connect them, and where there is more than one MTU size in use. For example, in Data Centers and on paths between Data Centers, to allow hosts to better take advantage of a path that is able to support a large PMTU.

The design of the option is sufficiently simple that it can be executed on a router's fast path. A successful experiment depends on both implementation by host and router vendors and deployment by operators. The contained use-case of connections within and between Data Centers could be a driver for deployment.

The method could also be useful in other environments, including the general Internet, and offers advantage when this Hop-by-Hop Option is supported on all paths. The method is more robust when used to probe the path using packets that do not carry application data and when also paired with a method such as Packetization Layer PMTUD [RFC4821] or Datagram PLPMTUD [RFC8899].

5. IPv6 Minimum Path MTU Hop-by-Hop Option

The Minimum Path MTU Hop-by-Hop Option has the following format:

Option Type	Option Data Len	Option Data		
BBCTTTTT	00000100	Min-PMTU	Rtn-PMTU	R

Option Type (see Section 4.2 of [RFC8200]):

BB 00 Skip over this option and continue processing.

C 1 Option data can change en route to the packet's final destination.

TTTTT 10000 Option Type assigned from IANA [IANA-HBH].

Length: 4 The size of the value field in Option Data field supports PMTU values from 0 to 65,534 octets, the maximum size represented by the Path MTU option.

Min-PMTU: n 16-bits. The minimum MTU recorded along the path in octets, reflecting the smallest link MTU that the packet experienced along the path. A value less than the IPv6 minimum link MTU [RFC8200] MUST be ignored.

Rtn-PMTU: n 15-bits. The returned Path MTU field, carrying the 15 most significant bits of the latest received Min-PMTU field for the forward path. The value zero means that no Reported MTU is being returned.

R n 1-bit. R-Flag. Set by the source to signal that the destination host should include the received Rtn-PMTU field updated by the reported Min-PMTU value when the destination host is to send a PMTU Option back to the source host.

Figure 3

NOTE: The encoding of the final two octets (Rtn-PMTU and R-Flag) could be implemented by a mask of the latest received Min-PMTU value with 0xFFFE, discarding the right-most bit and then performing a logical 'OR' with the R-Flag value of the sender. This encoding fits in the minimum-sized Hop-by-Hop Option header.

6. Router, Host, and Transport Layer Behaviors

6.1. Router Behavior

Routers that are not configured to support Hop-by-Hop Options are not expected to examine or process the contents of this option [RFC8200].

Routers that support Hop-by-Hop Options, but are not configured to support this option SHOULD skip over this option and continue to processing the header [RFC8200].

Routers that support this option MUST compare the value of the Min-PMTU field with the MTU configured for the outgoing link. If the MTU of the outgoing link is less than the Min-PMTU, the router rewrites the Min-PMTU in the Option to use the smaller value. (The router processing is performed without checking the valid range of the Min-PMTU or the Rtn-PMTU fields.)

A router MUST ignore and MUST NOT change the Rtn-PMTU field or the R-Flag in the option.

6.2. Host Operating System Behavior

The PMTU entry associated with the destination in the host's destination cache [RFC4861] SHOULD be updated after detecting a change using the IPv6 Minimum Path MTU Hop-by-Hop Option. This cached value can be used by other flows that share the host's destination cache.

The value in the host destination cache SHOULD be used by PLPMTUD to select an initial PMTU for a flow. The cached PMTU is only increased by PLPMTUD when the Packetization Layer determines the path actually supports a larger PMTU [RFC4821] [RFC8899].

When requested to send an IPv6 packet with the MinPMTU HBH option, the source host includes the option in an outgoing packet. The source host MUST fill the Min-PMTU field with the MTU configured for the link over which it will send the packet on the next hop towards the destination host.

When a host includes the option in a packet it sends, the host SHOULD set the Rtn-PMTU field to the previously cached value of the received Minimum Path MTU for the flow in the Rtn-PMTU field (see Section 6.3.3). If this value is not set (for example, because there is no cached reported Min-PMTU value), the Rtn-PMTU field value MUST be set to zero.

The source host MAY request the destination host to return the reported Min-PMTU value by setting the R-Flag in the option of an outgoing packet. The R-Flag SHOULD NOT be set when the MinPMTU HBH Option was sent solely to provide requested feedback on the return Path MTU to avoid each response generating another response.

The destination host controls when to send a packet with this option in response to an R-flag, as well as which packets to include it in. The destination host MAY limit the rate at which it sends these packets.

A destination host only sets the R Flag if it wishes the source host to also return the discovered PMTU value for the path from the destination to the source.

The normal sequence of operation of the R-Flag using the terminology from the diagram in Figure 1 is:

1. The source sends a probe to the destination. The sender sets the R-Flag.
2. The destination responds by sending a probe including the received Min-PMTU as the Rtn-PMTU. A destination that does not wish to probe the return path sets the R-Flag to 0.

6.3. Transport Layer Behavior

This Hop-by-Hop option is intended to be used with a path MTU discovery method.

PLPMTUD [RFC9000] uses probe packets for two distinct functions:

- * Probe packets are used to confirm connectivity. Such probes can be of any size up to the PLPMTU. These probe packets are sent to solicit a response use the path to the remote node. These probe packets can carry the Hop-by-Hop PMTU option, providing the final size of the packet does not exceed the current PLPMTU. After validating that the packet originates from the path (section 4.6.1), the PLPMTUD method can use the reported size from the Hop-by-Hop option as the next search point when it resumes the search algorithm. (This use resembles the use of the PTB_SIZE information in section 4.6.2 of [RFC8899])
- * A second use of probe packets is to explore if a path supports a packet size greater than the current PLPMTU. If this probe packet is successfully delivered (as determined by the source host), then the PLPMTU is raised to the size of the successful probe. These probe packets do not usually set the Path MTU Hop-by-Hop option.

See section 1.2 of [RFC8899]. Section 4.1 of [RFC8899] also describes ways that a Probe Packet can be constructed, depending on whether the probe packets carry application data.

- * The PMTU Hop-by-Hop Option Probe can be sent on packets that include application data, but needs to be robust to potential loss of the packet (i.e., with the possibility that retransmission might be needed if the packet is lost).
- * Using a PMTU Probe on packets that do not carry application data will avoid the need for loss recovery if a router on the path drops packets that set this option. (This avoids the transport needing to retransmit a lost packet that includes this option.) This is the normal default format for both uses of probes.

6.3.1. Including the Option in an Outgoing Packet

The upper layer protocol can request the MinPMTU HBH option to be included in an outgoing IPv6 packet. A transport protocol (or upper layer protocol) can include this option only on specific packets used to test the path. This option does not need to be included in all packets belonging to a flow.

NOTE: Including this option in a large packet (e.g., one larger than the present PMTU) is not likely to be useful, since the large packet would itself be dropped by any link along the path with a smaller MTU, preventing the Min-PMTU information from reaching the destination host.

Discussion:

- * In the case of TCP, the option could be included in a packet that carries a TCP segment sent after the connection is established. A segment without data could be used, to avoid the need to retransmit this data if the probe packet is lost. The discovered value can be used to inform PLPMTUD [RFC4821].

NOTE: A TCP SYN can also negotiate the Maximum Segment Size (MSS), which acts as an upper limit to the packet size that can be sent by a TCP sender. If this option were to be included in a TCP SYN, it could increase the probability that the SYN segment is lost when routers on the path drop packets with this option (see Section 6.3.6), which could have an unwanted impact on the result of racing options [I-D.ietf-taps-arch] or feature negotiation.

- * The use with datagram transport protocols (e.g., UDP) is harder to characterize because applications using datagram transports range from very short-lived (low data-volume applications) exchanges, to longer (bulk) exchanges of packets between the source and destination hosts [RFC8085].
- * Simple-exchange protocols (i.e., low data-volume applications [RFC8085] that only send one or a few packets per transaction), might assume that the PMTU is symmetrical. That is, the PMTU is the same in both directions, or at least not smaller for the return path. This optimization does not hold when the paths are not symmetric.
- * The MinPMTU HBH option can be used with ICMPv6 [RFC4443]. This requires a response from the remote node and therefore is restricted to use with ICMPv6 echo messages. The MinPMTU HBH option could provide additional information about the PMTU that might be supported by a path. This could be use as a diagnostic tool to measure the PMTU of a path. As with other uses, the actual supported PMTU is only confirmed after receiving a response to a subsequent probe of the PMTU size.
- * A datagram transport can utilise DPLPMTUD [RFC8899]. For example, QUIC (see section 14.3 of [RFC9000]), can use DPLPMTUD to determine whether the path to a destination will support a desired maximum datagram size. When using the IPv6 MinPMTU HBH option, the option could be added to an additional QUIC PMTU Probe that is of minimal size (or one no larger than the currently supported PMTU size). Once the return Path MTU value in the MinPMTU HBH option has been learned, DPLPMTUD can be triggered to test for a larger PLPMTU using an appropriately sized PLPMTU Probe Packet (see section 5.3.1 of [RFC8899]).
- * The use of this option with DNS and DNSSEC over UDP is expected to work for paths where the PMTU is symmetric. The DNS server will learn the PMTU from the DNS query messages. If the Rtn-PMTU value is smaller, then a large DNSSEC response might be dropped and the known problems with PMTUD will then occur. DNS and DNSSEC over transport protocols that can carry the PMTU ought to work.
- * This method also can be used with Anycast to discover the PMTU of the path, but the use needs to be aware that the Anycast binding might change.

6.3.2. Validation of the Packet that includes the Option

An upper layer protocol (e.g., transport endpoint) using this option needs to provide protection from data injection attacks by off-path devices [RFC8085]. This requires a method to assure that the information in the Option Data is provided by a node on the path. This validates that the packet forms a part of an existing flow, using context available at the upper layer. For example, a TCP connection or UDP application that maintains the related state and uses a randomized ephemeral port would provide this basic validation to protect from off-path data injection, see Section 5.1 of [RFC8085]. IPsec [RFC4301] and TLS [RFC8446] provide greater assurance.

The upper layer discards any received packet when the packet validation fails. When packet validation fails, the upper layer **MUST** also discard the associated Option Data from the MinPMTU HBH option without further processing.

6.3.3. Receiving the Option

For a connection-oriented upper layer protocol, caching of the received Min-PMTU could be implemented by saving the value in the connection context at the transport layer. A connection-less upper layer (e.g., one using UDP), requires the upper layer protocol to cache the value for each flow it uses.

A destination host that receives a MinPMTU HBH Option with the R-Flag **SHOULD** include the MinPMTU HBH option in the next outgoing IPv6 packet for the corresponding flow.

A simple mechanism could only include this option (with the Rtn-PMTU field set) the first time this option is received or when it notifies a change in the Minimum Path MTU. This limits the number of packets including the option packets that are sent. However, this does not provide robustness to packet loss or recovery after a sender loses state.

Discussion:

- * Some upper layer protocols send packets less frequently than the rate at which the host receives packets. This provides less frequent feedback of the received Rtn-PMTU value. However, a host always sends the most recent Rtn-PMTU value.

6.3.4. Using the Rtn-PMTU Field

The Rtn-PMTU field provides an indication of the PMTU from on-path routers. It does not necessarily reflect the actual PMTU between the source and destination hosts. Care therefore needs to be exercised in using the Rtn-PMTU value. Specifically:

- * The actual PMTU can be lower than the Rtn-PMTU value because the Min-PMTU field was not updated by a router on the path that did not process the option.
- * The actual PMTU may be lower than the Rtn-PMTU value because there is a layer-2 device with a lower MTU.
- * The actual PMTU may be larger than the Rtn-PMTU value because of a corrupted, delayed or mis-ordered response. A source host **MUST** ignore a Rtn-PMTU value larger than the MTU configured for the outgoing link.
- * The path might have changed between the time when the probe was sent and when the Rtn-PMTU value received.

IPv6 requires that every link in the Internet have an MTU of 1280 octets or greater. A node **MUST** ignore a Rtn-PMTU value less than 1280 octets [RFC8200].

To avoid unintentional dropping of packets that exceed the actual PMTU (e.g., Scenario 3 in Section 1.1), the source host can delay increasing the PMTU until a probe packet with the size of the Rtn-PMTU value has been successfully acknowledged by the upper layer, confirming that the path supports the larger PMTU. This probing increases robustness, but adds one additional path round trip time before the PMTU is updated. This use resembles that of PTB messages in section 4.6 of DPLPMTUD [RFC8899] (with the important difference that a PTB message can only seek to lower the PMTU, whereas this option could trigger a probe packet to seek to increase the PMTU.)

Section 5.2 of [RFC8201] provides guidance on the caching of PMTU information and also the relation to IPv6 flow labels. Implementations should consider the impact of Equal Cost Multipath (ECMP) [RFC6438]. Specifically, whether a PMTU ought to be maintained for each transport endpoint, or for each network address.

6.3.5. Detecting Path Changes

Path characteristics can change and the actual PMTU could increase or decrease over time. For instance, following a path change when packets are forwarded over a link with a different MTU than that previously used. To bound the delay in discovering an increase in the actual PMTU, a host with a link MTU larger than the current PMTU SHOULD periodically send the MinPMTU HBH Option with the R-bit set. DPLPMTUD provides recommendations concerning how this could be implemented (see Section 5.3 of [RFC8899]). Since the option consumes less capacity than a full-sized probe packet, there can be advantage in using this to detect a change in the path characteristics.

6.3.6. Detection of Dropping Packets that include the Option

There is evidence that some middleboxes drop packets that include Hop-by-Hop options. For example, a firewall might drop a packet that carries an unknown extension header or option. This practice is expected to decrease as an option becomes more widely used. It could result in generation of an ICMPv6 message indicating the problem. This could be used to (temporarily) suspend use of this option.

A middlebox that silently discards a packet with this option results in dropping of any packet using the option. This dropping can be avoided by appropriate configuration in a controlled environment, such as within a data centre, but needs to be considered for Internet usage. Section 6.2 recommends that this option is not used on packets where loss might adversely impact performance.

7. IANA Considerations

IANA has assigned and registered an IPv6 Hop-by-Hop Option type with Temporary status from the "Destination Options and Hop-by-Hop Options" registry [IANA-HBH]. This assignment is shown in Section 5.

IANA is requested to update this registry to point to this document and remove the Temporary status.

8. Security Considerations

This section discusses the security considerations. It first reviews router option processing. It then reviews host processing when receiving this option at the network layer. It then considers two ways in which the Option Data can be processed, followed by two approaches for using the Option Data. Finally, it discusses middlebox implications related to use in the general Internet.

8.1. Router Option Processing

This option shares the characteristics of all other IPv6 Hop-by-Hop Options, in that if not supported at line rate it could be used to degrade the performance of a router. This option, while simple, is no different to other uses of IPv6 Hop-by-Hop options.

It is common for routers to ignore the Hop-by-Hop Option header or drop packets containing a Hop-by-Hop Option header. Routers implementing IPv6 according to [RFC8200] only examine and process the Hop-by-Hop Options header if explicitly configured to do so.

8.2. Network Layer Host Processing

A malicious attacker can forge a packet directed at a host that carries the MinPMTU HBH option. By design, the fields of this IP option can be modified by the network.

For comparison, the ICMPv6 Packet Too Big message used in [RFC8201] Path MTU Discovery, the source host has an inherent trust relationship with the destination host including this option. This trust relationship can be used to help verify the option. ICMPv6 Packet Too Big messages are sent from any router on the path to the destination host, the source host has no prior knowledge of these routers (except for the first hop router).

Reception of this packet will require processing as the network stack parses the packet before the packet is delivered to the upper layer protocol. This network layer option processing is normally completed before any upper layer protocol delivery checks are performed.

The network layer does not normally have sufficient information to validate that the packet carrying an option originated from the destination (or an on-path node). It also does not typically have sufficient context to demultiplex the packet to identify the related transport flow. This can mean that any changes resulting from reception of the option applies to all flows between a pair of endpoints.

These considerations are no different to other uses of Hop-by-Hop options, and this is the use case for PMTUD. The following section describes a mitigation for this attack.

8.3. Validating use of the Option Data

Transport protocols should be designed to provide protection from data injection attacks by off-path devices and mechanisms should be described in the Security Considerations for each transport specification (see Section 5.1 of the UDP Guidelines [RFC8085]). For example, a TCP or UDP application that maintains the related state and uses a randomized ephemeral port would provide basic protection. TLS [RFC8446] or IPsec [RFC4301] provide cryptographic authentication. An upper layer protocol that validates each received packet discards any packet when this validation fails. In this case, the host **MUST** also discard the associated Option Data from the MinPMTU HBH option without further processing (Section 6.3).

A network node on the path has visibility of all packets it forwards. By observing the network packet payload, the node might be able to construct a packet that might be validated by the destination host. Such a node would also be able to drop or limit the flow in other ways that could be potentially more disruptive. Authenticating the packet, for example, using IPsec [RFC4301] or TLS [RFC8446] mitigates this attack. Note that AH style authentication [RFC4302] while authenticating the payload and outer IPv6 header, does not check Hop-by-Hop options that change on route.

8.4. Direct use of the Rtn-PMTU Value

The simplest way to utilize the Rtn-PMTU value is to directly use this to update the PMTU. This approach results in a set of security issues when the option carries malicious data:

- * A direct update of the PMTU using the Rtn-PMTU value could result in an attacker inflating or reducing the size of the host PMTU for the destination. Forcing a reduction in the PMTU can decrease the efficiency of network use, might increase the number of packets/fragments required to send the same volume of payload data, and prevents sending an unfragmented datagram larger than the PMTU. Increasing the PMTU can result in black-holing (see Section 1.1 of [RFC8899]) when the source host sends packets larger than the actual PMTU. This persists until the PMTU is next updated.
- * The method can be used to solicit a response from the destination host. A malicious attacker could forge a packet that causes the destination to add the option to a packet sent to the source host. A forged value of Rtn-PMTU in the Option Data might also impact the remote endpoint, as described in the previous bullet. This persists until a valid MinPMTU HBH option is received. This attack could be mitigated by limiting the sending of the MinPMTU HBH option in reply to incoming packets that carry the option.

8.5. Using the Rtn-PMTU Value as a Hint for Probing

Another way to utilize the Rtn-PMTU value is to indirectly trigger a probe to determine if the path supports a PMTU of size Rtn-PMTU. This approach needs context for the flow, and hence assumes an upper layer protocol that validates the packet that carries the option (see Section 8.3). This is the case when used in combination with DPLPMTUD [RFC8899]. A set of security considerations result when an option carries malicious data:

- * If the forged packet carries a validated option with a non-zero Rtn-PMTU field, the upper layer protocol could utilize the information in the Rtn-PMTU field. A Rtn-PMTU larger than the current PMTU can trigger a probe for a new size.
- * If the forged packet carries a non-zero Min-PMTU field, the upper layer protocol would change the cached information about the path from the source. The cached information at the destination host will be overwritten when the host receives another packet that includes a MinPMTU HBH option corresponding to the flow.
- * Processing of the option could cause a destination host to add the MinPMTU HBH option to a packet sent to the source host. This option will carry a Rtn-PMTU value that could have been updated by the forged packet. The impact of the source host receiving this resembles that discussed previously.

8.6. Impact of Middleboxes

There is evidence that some middleboxes drop packets that include Hop-by-Hop options. For example, a firewall might drop a packet that carries an unknown extension header or option. This practice is expected to decrease as the option becomes more widely used. Methods to address this are discussed in Section 6.3.6.

When a forged packet causes a packet to be sent including the MinPMTU HBH option, and the return path does not forward packets with this option, the packet will be dropped Section 6.3.6. This attack is mitigated by validating the option data before use and by limiting the rate of responses generated. An upper layer could further mitigate the impact by responding to an R-Flag by including the option in a packet that does not carry application data.

9. Experiment Goals

This section describes the experimental goals of this specification.

A successful deployment of the method depends upon several components being implemented and deployed:

- * Support in the sending node (see Section 6.2). This also requires corresponding support in upper layer protocols (see Section 6.3).
- * Router support in nodes (see Section 6.1). The IETF continues to provide recommendations on the use of IPv6 Hop-by-Hop options, for example Section 2.2.2 of [RFC9099]. This document does not update the way router implementations configure support for Hop-by-Hop options.
- * Support in the receiving node (see Section 6.3.3).

Experience from deployment is an expected input to any decision to progress this specification from Experimental to IETF Standards Track. Appropriate inputs might include:

- * Reports of implementation experience;
- * Measurements of the number paths where the method can be used;
- * Measurements showing the benefit realized or the implications of using specific methods over specific paths.

10. Implementation Status

At the time this document was published there are two known implementations of the Path MTU Hop-by-Hop option. These are:

- * Wireshark dissector. This is shipping in production in Wireshark version 3.2 [WIRESHARK].
- * A prototype in the open source version of the FD.io Vector Packet Processing (VPP) technology [VPP]. At the time this document was published, the source code can be found [VPP_SRC].

11. Acknowledgments

Helpful comments were received from Tom Herbert, Tom Jones, Fred Templin, Ole Troan, Tianran Zhou, Jen Linkova, Brian Carpenter, Peng Shuping, Mark Smith, Fernando Gont, Michael Dougherty, Erik Kline, and other members of the 6MAN working group.

12. Change log [RFC Editor: Please remove]

draft-ietf-6man-mtu-option-15, 2022-May-10

- * Correcting an editing mistake in Appendix A.
- * Editorial Change.

draft-ietf-6man-mtu-option-14, 2022-April-15

- * Area Director Reviews:
 - Lars Eggert's Review: Fixed "nits".
 - Eric Vyncke's Review: Added that this work is focused on Unicast, removed Discussion from Section 6.1, revised text on PLPMTUD probing, changed SHOULD to MUST in Section 6.3.4, and fixed several NITs.
 - Alvaro Retana's Review: Changed SHOULD language to more general text in Section 6.1
 - ARTART Review: Added new Appendix "Examples of Usage" with diagrams showing examples of use.
 - Zaheduzzaman Sarker's Review: Fixed some editorial issues, and updated SHOULD language.
- * Editorial Changes.

draft-ietf-6man-mtu-option-13, 2022-February-28

- * Area Directorate Reviews:
 - SECDIR Review: Fixed "nit".
 - TSVART Review: Restructured Section 6 including making Transport Behavior more prominent, added text about ICMPv6 to Section 6.3.1, moved the text about prior work in RFC1063 to Section 2.
 - GENART Review: Added text to Section 1 that this option was designed to work with packet sizes that can be specified in the IPv6 Header.
- * Editorial Changes.

draft-ietf-6man-mtu-option-12, 2022-January-26

- * Clarified a few issues raised by AD review by Erik Kline AD review.

draft-ietf-6man-mtu-option-11, 2021-September-30

- * Clarifications and editorial changes to the Security Considerations section based on early AD review by Erik Kline.

draft-ietf-6man-mtu-option-10, 2021-September-27

- * Clarifications and editorial changes based on second chair review by Ole Troan.
- * Editorial changes.

draft-ietf-6man-mtu-option-09, 2021-September-23

- * Clarifications and editorial changes based on review by Michael Dougherty.

draft-ietf-6man-mtu-option-08, 2021-September-7

- * Clarifications and editorial changes based on chair review by Ole Troan.
- * Correction and clarifications based on review by Fernando Gont.

draft-ietf-6man-mtu-option-07, 2021-August-31

- * Added Experiment Goals section.
- * Added Implementation Status section.
- * Updated the IANA Considerations section to point to this document and remove Temporary status.
- * Clarifications and editorial changes based on review by Mark Smith.

draft-ietf-6man-mtu-option-06, 2021-August-7

- * Transport usage of the mechanism clarified in response to feedback and suggestions from Jen Linkova.
- * Restructured Section 6 to improve readability.
- * Editorial changes.

draft-ietf-6man-mtu-option-05, 2021-April-28

- * Editorial changes.

draft-ietf-6man-mtu-option-04, 2020-Oct-23

- * Fixes for typos.

draft-ietf-6man-mtu-option-03, 2020-Sept-14

- * Rewrite to make text and terminology more consistent.
- * Added the notion of validating the packet before use of the HBH option data.
- * Method aligned with the way common APIs send/receive HBH option data.
- * Added reference to DPLPMTUD and clarified upper layer usage.
- * Completed security considerations section.

draft-ietf-6man-mtu-option-02, 2020-March-9

- * Editorial changes to make text and terminology more consistent.

- * Added reference to DPLPMTUD.

draft-ietf-6man-mtu-option-01, 2019-September-13

- * Changes to show IANA assigned code point.
- * Editorial changes to make text and terminology more consistent.
- * Added a reference to RFC8200 in Section 2 and a reference to RFC6438 in Section 6.3.

draft-ietf-6man-mtu-option-00, 2019-August-9

- * First 6man w.g. draft version.
- * Changes to request IANA allocation of code point.
- * Editorial changes.

draft-hinden-6man-mtu-option-02, 2019-July-5

- * Changed option format to also include the Returned PMTU value and Return flag and made related text changes in Section 6.2 to describe this behavior.
- * ICMPv6 Packet Too Big messages are no longer used for feedback to the source host.
- * Added to Acknowledgements Section that a similar mechanism was proposed for IPv4 in 1988 in [RFC1063].
- * Editorial changes.

draft-hinden-6man-mtu-option-01, 2019-March-05

- * Changed requested status from Standards Track to Experimental to allow use of experimental option type (11110) to allow for experimentation. Removed request for IANA Option assignment.
- * Added Section 2 "Motivation and Problem Solved" section to better describe what the purpose of this document is.
- * Added appendix describing planned experiments and how the results will be measured.
- * Editorial changes.

draft-hinden-6man-mtu-option-00, 2018-Oct-16

- * Initial draft.

13. References

13.1. Normative References

[IANA-HBH] "Destination Options and Hop-by-Hop Options",
<<https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml#ipv6-parameters-2>>.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <<https://www.rfc-editor.org/info/rfc8201>>.

13.2. Informative References

- [I-D.ietf-taps-arch] Pauly, T., Trammell, B., Brunstrom, A., Fairhurst, G., and C. Perkins, "An Architecture for Transport Services", Work in Progress, Internet-Draft, draft-ietf-taps-arch-12, 3 January 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-taps-arch-12>>.
- [RFC1063] Mogul, J., Kent, C., Partridge, C., and K. McCloghrie, "IP MTU discovery options", RFC 1063, DOI 10.17487/RFC1063, July 1988, <<https://www.rfc-editor.org/info/rfc1063>>.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <<https://www.rfc-editor.org/info/rfc1191>>.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<https://www.rfc-editor.org/info/rfc2460>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <<https://www.rfc-editor.org/info/rfc4302>>.

- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <<https://www.rfc-editor.org/info/rfc4821>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <<https://www.rfc-editor.org/info/rfc6438>>.
- [RFC7637] Garg, P., Ed. and Y. Wang, Ed., "NVGRE: Network Virtualization Using Generic Routing Encapsulation", RFC 7637, DOI 10.17487/RFC7637, September 2015, <<https://www.rfc-editor.org/info/rfc7637>>.
- [RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <<https://www.rfc-editor.org/info/rfc8085>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8899] Fairhurst, G., Jones, T., Tüxen, M., Rüngeler, I., and T. Völker, "Packetization Layer Path MTU Discovery for Datagram Transports", RFC 8899, DOI 10.17487/RFC8899, September 2020, <<https://www.rfc-editor.org/info/rfc8899>>.
- [RFC8900] Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", BCP 230, RFC 8900, DOI 10.17487/RFC8900, September 2020, <<https://www.rfc-editor.org/info/rfc8900>>.
- [RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <<https://www.rfc-editor.org/info/rfc9000>>.

- [RFC9099] Vyncke, É., Chittimaneni, K., Kaeo, M., and E. Rey,
"Operational Security Considerations for IPv6 Networks",
RFC 9099, DOI 10.17487/RFC9099, August 2021,
<<https://www.rfc-editor.org/info/rfc9099>>.
- [VPP] "VPP/What is VPP?",
<https://wiki.fd.io/view/VPP/What_is_VPP%3F>.
- [VPP_SRC] "VPP Source", <<https://gerriet.fd.io/r/c/vpp/+21948>>.
- [WIRESHARK] "Wireshark Network Protocol Analyzer",
<<https://www.wireshark.org>>.

Appendix A. Examples of Usage

This section provides examples that illustrate a use of the MinPMTU HBH option by a source using DPLPMTUD to discover the PLPMTU supported by a path. They consider a path where the on-path router has been configured with an outgoing MTU of d' . The source starts by transmission of packets of size a , and then uses DPLPMTUD to seek to increase the size in steps resulting in sizes of b, c, d, e , etc., (chosen by the search algorithm used by DPLPMTUD). The search algorithm terminates with a PLPMTU that is at least d and is less than or equal to d' .

The first example considers DPLPMTUD without using the MinPMTU HBH option. In this case, DPLPMTUD searches using an increasing size of probe packet. Probe packets of size (e) are sent, which are larger than the actual PMTU. In this example, PTB messages are not received from the routers and repeated unsuccessful probes result in the search phase completing. Packets of data are never sent with a size larger than the size of the last confirmed probe packet. ACKs of data packets are not shown.

```

----Packets of data size (a) ----->
----Probe size (b) ----->
<----- ACK of probe -----
----Packets of data size (b) ----->
----Probe size (c) ----->
<----- ACK of probe -----
----Packets of data size (c) ----->
----Probe size (d) ----->
<----- ACK of probe -----
----Packets of data size (d) ----->
<----- ACK of probe -----
...
----Probe size (e) -----X
      X----ICMPv6 PTB (d') --|
----Packets of data size (d) ----->
----Probe size (e) -----X (again)
      X----ICMPv6 PTB (d') --|
----Packets of data size (d) -----
...
etc, until MaxProbes are unsuccessful and search phase completes.
----Packets of data size (d) ----->

```

Figure 4

The second example considers DPLPMTUD with the MinPMTU HBH option set on a connectivity probe packet.

The IPv6 option is sent end-to-end, and the Min-PMTU is updated by a router on the path to d', which is returned in a response that also sets the MinPMTU HBH option. Upon receiving Rtn-PMTU value is received, DPLPMTUD immediately sends a probe packet of the target size (d'). If the probe packet is confirmed for the path, the PLPMTU is updated, allowing the source to use data packets up to size d'. (The search algorithm is allowed to continue to probe to see if the path supports a larger size.) Packets of data are never sent with a size larger than the last confirmed probe size, d'.

```

----Packets of data size (a) ----->
----Connectivity probe with MinPMTU-
      +--updated to minPMTU=d'----->
<-----ACK with Rtn-PMTU=d'-----
----Packets of data size (a) ----->
----Probe size (d') ----->
<----- ACK of probe -----
----Packets of data size (d') ----->
Search phase completes.
----Packets of data size (d') ----->

```

Figure 5

The final example considers DPLPMTUD with the MinPMTU HBH option set on a connectivity probe packet, but shows the effect when this connectivity probe packet is dropped.

In this case, the packet with the MinPMTU HBH option is not received. DPLPMTUD searches using probe packets of increasing size, increasing the PLPMTU when the probes are confirmed. An ICMPv6 PTB message is received when the probed size exceeds the actual PMTU, indicating a PTB_SIZE of d'. DPLPMTUD immediately sends a probe packet of the target size (d'). If the probe packet is confirmed for the path, the PLPMTU is updated, allowing the source to use data packets up to size d'. If the ICMPv6 PTB message is not received, the DPLPMTU will be the last confirmed probe size, d.

```

----Packets of data size (a) ----->
----Connectivity probe with MinPMTU -----X
----Packets of data size (a) ----->
----Probe size (b) ----->
<----- ACK of probe -----
----Packets of data size (b) ----->
----Probe size (c) ----->
<----- ACK of probe -----
----Packets of data size (c) ----->
----Probe size (d) ----->
<----- ACK of probe -----
----Packets of data size (d) ----->
----Probe size (e) -----X
<--ICMPv6 PTB PTB_SIZE(d') -|
----Packets of data size (d) ----->
----Probe size (d') using target set by PTB_SIZE ----->
<----- ACK of probe -----
Search phase completes.
----Packets of data size (d') ----->

```

Figure 6

The number of probe rounds depends on the number of steps needed by the search algorithm, and is typically larger for a larger PMTU.

Authors' Addresses

Robert M. Hinden
 Check Point Software
 959 Skyway Road
 San Carlos, CA 94070
 United States of America

Email: bob.hinden@gmail.com

Godred Fairhurst
University of Aberdeen
School of Engineering
Fraser Noble Building
Aberdeen
AB24 3UE
United Kingdom
Email: gorrry@erg.abdn.ac.uk

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 25 April 2022

T. Winters
QA Cafe
O. Troan
cisco
22 October 2021

The Universal IPv6 Configuration Option
draft-troan-6man-universal-ra-option-06

Abstract

One of the original intentions for the IPv6 host configuration, was to configure the network-layer parameters only with IPv6 ND, and use service discovery for other configuration information. Unfortunately that hasn't panned out quite as planned, and we are in a situation where all kinds of configuration options are added to RAs. This document proposes a new universal option for RA in a self-describing data format, with the list of elements maintained in an IANA registry, with greatly relaxed rules for registration.

Status of This Memo

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

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

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

This Internet-Draft will expire on 25 April 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

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

Table of Contents

1. Introduction	2
2. Conventions	3
3. Introduction	3
4. The Universal IPv6 Configuration option	3
5. CBOR encoding	5
6. Implementation Guidance	5
7. Implementation Status	5
8. Security Considerations	5
9. IANA Considerations	6
9.1. Universal configuration option	6
9.2. Initial objects in the registry	6
9.3. Initial objects in the registry	6
9.3.1. CDDL/JSON Mapping Parameters to CBOR	6
9.3.2. Key Registry	7
10. Normative References	8
11. Informative References	9
Appendix A. Acknowledgements	9
Authors' Addresses	9

1. Introduction

This document proposes a new universal option for the Router Advertisement IPv6 ND message [RFC4861]. Its purpose is to use the RA messages as opaque carriers for configuration information between an agent on a router and a host.

DHCP is suited to give per-client configuration information, while the RA mechanism advertises configuration information to all hosts on the link. There is a long running history of "conflict" between the two. The arguments go; there is less fate-sharing in DHCP, DHCP doesn't deal with multiple sources of information, or make it more difficult to change information independent of the lifetimes, RA cannot be used to configure different information to different clients and so on. And of course some options are only available in RAs and some options are only available in DHCP.

While this proposal does not resolve the DHCP vs RA debate, it proposes a solution to the problem of a very slow process of standardizing new Router Advertisement options, and the IETF spending an inordinate amount of time arguing over new configuration options in Router Advertisements. It is possible in the future to use the new universal option in DHCP, since this would lead to additional conflict resolution an additional document will need to be considered for that.

2. Conventions

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].

Additionally, the key words "*MIGHT*", "*COULD*", "*MAY WISH TO*", "*WOULD PROBABLY*", "*SHOULD CONSIDER*", and "*MUST (BUT WE KNOW YOU WON'T)*" in this document are to be interpreted as described in RFC 6919 [RFC6919].

3. Introduction

This document specifies a new "self-describing" universal configuration option. Currently new configuration option requires "standards action". The proposal is that no future IETF document will be required. The configuration option is described directly in the universal configuration IANA registry.

4. The Universal IPv6 Configuration option

The option data is described using the schema language CDDL [RFC8610], encoded in CBOR [RFC7049].

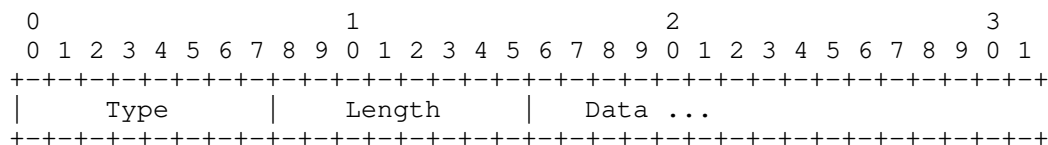


Figure 1: IPv6 Configuration Option Format

Fields:

Type: 42 for Universal IPv6 Configuration Option

Length: The length of the option (including the type and length fields) in units of 8 octets.

Data: CBOR encoded data.

The Option is zero-padded to nearest 8-octet boundary.

Example of an JSON instance of the option:

```
{
  "ietf": {
    "dns": {
      "dnssl": [
        "example.com"
      ],
      "rdnss": [
        "2001:db8::1",
        "2001:db8::2"
      ]
    },
    "nat64": {
      "prefix": "64:ff9b::/96"
    },
    "rio": [
      {
        "prefix": "::/0",
        "next-hop": "fe80::1"
      },
      {
        "prefix": "2001:db8::/32",
        "next-hop": "fe80::2"
      }
    ]
  }
}
```

The universal IPv6 Configuration option MUST be small enough to fit within a single IPv6 ND packet. It then follows that a single element in the dictionary cannot be larger than what fits within a single option. Different elements can be split across multiple universal configuration options (in separate packets). All IANA registered elements are under the "ietf" key in the dictionary. Private configuration information can be included in the option using different keys.

If information learnt via this option conflicts with other configuration information learnt via Router Advertisement messages, that is considered a configuration error. How those conflicts should be resolved is left up to the implementation.

5. CBOR encoding

It is recommended that the user can configure the option using JSON. Likewise an application registering interest in an option SHOULD be able to use string keys. The CBOR encoding to save space, uses integers for map keys. The mapping table between integer and string map keys are part of the IANA registry for the option.

Values -23-23 encodes to a single byte in CBOR, and these values are reserved for IETF used map keys.

6. Implementation Guidance

The purpose of this option is to allow users to use the RA as an opaque carrier for configuration information without requiring code changes in the option carrying infrastructure.

On the router there should be an API allowing a user to add an element, e.g. a JSON object [RFC8259] or a pre-encoded CBOR string to RAs sent on a given interface.

On the host side, an API SHOULD be available allowing applications to subscribe to received configuration elements. It SHOULD be possible to subscribe to configuration object by dictionary key.

The contents of any elements that are not recognized, either in whole or in part, by the receiving host MUST be ignored and the remainder of option's contents MUST be processed as normal.

An implementation SHOULD provide a "JSON interface" for configuring the option.

7. Implementation Status

The Universal IPv6 configuration option sending side is implemented in VPP (<https://wiki.fd.io/view/VPP> (<https://wiki.fd.io/view/VPP>)).

The implementation is a prototype released under Apache license and available at: <https://github.com/vpp-dev/vpp/commit/156db316565e77de30890f6e9b2630bd97b0d61d> (<https://github.com/vpp-dev/vpp/commit/156db316565e77de30890f6e9b2630bd97b0d61d>).

8. Security Considerations

Unless there is a security relationship between the host and the router (e.g. SEND), and even then, the consumer of configuration information can put no trust in the information received.

9. IANA Considerations

IANA is requested to add a new registry for the Universal IPv6 Configuration option. The registry should be named "IPv6 Universal Configuration Information Option".

The schema field follows the CDDL schema definition in [RFC8610].

Changes and additions to the registry follow the policies below [RFC8126]:

Range	Registration Procedure
-23-23	Standards Action
24-32767	Specification Required
32768-18446744073709551615	Expert Review

Table 1

A new registration requires a new CBOR key to parameter name assignment and a CDDL definition.

9.1. Universal configuration option

The IANA is requested to add the universal option to the "IPv6 Neighbor Discovery Option Formats" registry with the value of 42.

9.2. Initial objects in the registry

The PVD [RFC8801] elements and DNS [RFC8106]) are included to provide an alternative representation for the proposed new options in that draft.

9.3. Initial objects in the registry

9.3.1. CDDL/JSON Mapping Parameters to CBOR

Parameter Name / JSON key	CBOR Key
ietf	-23
pio	-22

mtu	-21
rio	-20
dns	-19
nat64	-18
ipv6-only	-17
pvd	-16
prefix	-15
preferred-lifetime	-14
valid-lifetime	-13
lifetime	-12
a-flag	-11
l-flag	-10
preference	-9
nexthop	-8
nssl	-7
dnss	-6
fqdn	-5
uri	-4

Table 2

9.3.2. Key Registry

CDDL	Reference
<pre> ietf = { ? pio : [+ pio] ? rio : [+ rio] ? dns : dns ? nat64: nat64 ? ipv6-only: bool ? pvd : pvd } dns = { nssl : [* tstr] dnss : [+ ipv6-address] lifetime : uint .size 4 } nat64 = { prefix : ipv6-prefix } ipv6-only : bool pvd = { fqdn : tstr uri : tstr ? dns : dns ? nat64: nat64 ? pio : [+ pio] ? rio : [+ rio] } </pre>	<p>RFC8106</p> <p>RFC7050</p> <p>[v6only]</p>

10. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.

- [RFC6919] Barnes, R., Kent, S., and E. Rescorla, "Further Key Words for Use in RFCs to Indicate Requirement Levels", RFC 6919, DOI 10.17487/RFC6919, April 2013, <<https://www.rfc-editor.org/info/rfc6919>>.
- [RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049, October 2013, <<https://www.rfc-editor.org/info/rfc7049>>.
- [RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <<https://www.rfc-editor.org/info/rfc8610>>.

11. Informative References

- [RFC8106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 8106, DOI 10.17487/RFC8106, March 2017, <<https://www.rfc-editor.org/info/rfc8106>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/info/rfc8259>>.
- [RFC8801] Pfister, P., Vyncke, É., Pauly, T., Schinazi, D., and W. Shao, "Discovering Provisioning Domain Names and Data", RFC 8801, DOI 10.17487/RFC8801, July 2020, <<https://www.rfc-editor.org/info/rfc8801>>.

Appendix A. Acknowledgements

Many thanks to Dave Thaler for feedback and suggestions of a more effective CBOR encoding. Thank you very much to Carsten Bormann for CBOR and CDDL help.

Authors' Addresses

T. Winters
QA Cafe

Email: tim@qacafe.com

O. Troan
cisco

Email: ot@cisco.com

IPv6 Maintenance (6man) Working Group
Internet Draft
Updates: 4861, 4862 (if approved)
Intended status: Standards Track
Expires: September 2022

E. Vasilenko
P. Volpato
Huawei Technologies
Olorunloba Olopade
Virgin Media
March 4, 2022

ND Prefix Robustness Improvements
draft-vv-6man-nd-prefix-robustness-02

Abstract

IPv6 prefixes could become invalid abruptly as a result of outages, network administrator actions, or particular product shortcomings.

That could lead to connectivity problems for the hosts attached to the subtended network.

This document has two targets: on one hand, to analyze the cases that may lead to network prefix invalidity; on the other to develop a root cause analysis for those cases and propose a solution.

This may bring to extensions of the protocols used to convey prefix information and other options.

Status of this Memo

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

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

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

This Internet-Draft will expire on September 2022.

Copyright Notice

Copyright (c) 2022 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. Terminology and pre-requisite.....	3
2. Introduction.....	4
3. Problem Scenarios.....	4
3.1. Reference architectures.....	5
3.2. Discussion on the scenarios.....	5
3.2.1. Non-graceful reload due to unexpected events	5
3.2.2. Graceful reload without precautions	6
3.2.3. Abrupt hardware replacement without the possibility for graceful prefix deprecation.....	7
3.2.4. Non-graceful configuration change	8
3.2.5. An uplink breaks connectivity without a relevant notification to the connected hosts.....	8
4. Root cause analysis.....	10
4.1. What to protect.....	10
4.2. Where to protect.....	12
4.3. When to protect: technology scenarios.....	12
5. Solutions.....	13
5.1. Multi-homing multi-prefix (MHMP) environment.....	13
5.2. A provider is not reachable in MHMP environment.....	16
5.3. Administrator abruptly replaces PA prefix.....	17
5.4. Planned router outage.....	18
5.5. Prefix information lost because of abrupt router outage..	19
5.6. Prefix information lost after hardware replacement.....	19
5.7. Link layer address of the router should be changed.....	20
5.8. Dependency between solutions and extensions.....	20
6. Extensions of the existing standards.....	20

6.1. Default router choice by host.....	20
6.2. Prefixes become dynamic.....	21
6.3. Do not forget to deprecate prefixes on renumbering.....	22
6.4. Do not forget to deprecate prefixes on shutdown.....	23
6.5. Store prefixes in non-volatile memory.....	23
6.6. Find lost information by "Synchronization".....	24
6.7. Default router announcement rules.....	26
6.8. Faster detection of the stale default router.....	26
6.9. Clean orphaned prefixes after default router list change.....	27
7. Interoperability analysis.....	27
8. Applicability analysis.....	28
9. Security Considerations.....	28
10. IANA Considerations.....	29
11. References.....	29
11.1. Normative References.....	29
11.2. Informative References.....	30
12. Acknowledgments.....	31

1. Terminology and pre-requisite

[ND] and [SLAAC] are pre-requisite to understand this document.
The terms are inherited from these standards.

Additional terms:

Home Gateway - a small consumer-grade router that provides network access between hosts on the local area network (LAN) and the Internet behind the wide area network (WAN)

PA - Provider-Aggregatable addresses leased to the client or subscriber

MHMP - Multi-Homing Multi-Prefix. An environment with hosts connected to different PA providers (multi-homing) through different address spaces announced from different providers (multi-prefix)

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

2. Introduction

It has been reported that some number of cases could lead to loss of information (primarily prefixes) by [ND]. Current [ND] protocol's default timers may lead to many days of outage for hosts. This is not acceptable.

This document analyses all potential cases when an outage could happen and proposes solutions. Discussion is restricted to potential [ND] extensions only.

MHMP environment has been considered. It has been discovered that [ND] problems could be isolated from the overall complex [MHMP] environment, and could be fixed separately.

The document is organized to introduce, in section 3, the scenarios where the issue of prefix invalidity may happen and the cases of invalidity.

Section 4 provides a root cause analysis for the cases of invalidity and identifies the corner-cases which are subject of our discussion.

Section 5 proposes a solution for the cases identified.

Section 6 brings the discussion forward, proposing extensions to [ND].

3. Problem Scenarios

[ND] distributes prefixes as Prefix Information Options (PIOs) in Router Advertisements (RA) messages from routers.

Once a router assigns a prefix to a host, this prefix is assumed to be stable so that hosts can employ it to configure the IPv6 addresses associated with their interfaces [SLAAC] or to forward packets to the network.

Prefix changes may happen and are governed by the rules of [ND], [SLAAC].

Yet, cases exist where prefix instability may happen. An example is provided by the so-called "flash-renumbering" event: when flash-renumbering happens a network prefix in use suddenly becomes invalid because it is replaced by a new prefix.

The router causing or forced to cause the network renumbering may not be able to cope with the effects of this sudden change (for example, deprecating the previously assigned prefixes). Another possibility is that the subtended hosts do not have the means of overcoming the effects of renumbering.

This section describes problems that were found in live networks. Most of the information in this section comes from [Flash-Renumbering], [SLAAC Robustness]. Their contributions are greatly acknowledged.

3.1. Reference architectures

Home broadband networks, SOHO (Small Office Home Office) networks are the typical scenarios affected by renumbering. Some problems discussed below applicable on the more general basis.

In typical case a router (e.g. Home Gateway, Customer Premise Equipment (CPE), Customer Edge (CE), etc.) is deployed to provide connectivity to a Service Provider network for the attached devices. A second router may be deployed for redundancy, especially for business scenarios.

Two reference architecture can be considered:

Architecture #1. Hosts are directly connected to the router. For example, a Home Gateway embeds the functions of L2 device (Ethernet switch, WiFi AP) and L3 device (router).

Architecture #2. Hosts connect to an intermediate L2 device (e.g. a wired Ethernet switch or a Wi-Fi access point) that, in turn, connects to the router (or routers, if uplink redundancy is requested).

3.2. Discussion on the scenarios

The discussion provided here is introductory to both the root cause analysis provided in section 4. and the solutions proposed in section 5.

3.2.1. Non-graceful reload due to unexpected events

A router could be reloaded abruptly for many reasons: hardware or software bug, power outage, manual intervention. This last one is very probable for home broadband subscribers that tend to fix every problem with power recycle.

Usually, it does not create additional problems for [ND] and [SLAAC] because the same PIO information would be advertised by the router in RA messages after each reload. In such cases, a Home Gateway would initialize its Ethernet and WiFi connections, clearing all stale information on directly connected hosts.

It should not create problems for proper home network design where all CPEs are routers - see [HomeNet Architecture]. The delegated prefix would not be changed in the case of subtended CPE reload. Prefix change in the case of upstream CPE reload should be properly discontinued by subtended CPE. There is the need for a special protocol for prefix distribution that is out of the scope of this document - see [HNCP].

For architecture #2 implemented in home environments, there is a corner case when Home Gateway's abrupt reload would not be visible to hosts connected to subtended "bridged" CPE. If it would coincide with the situation when a different prefix would be delegated from Carrier (at 37% probability according to [Residential practices]), it would lead to the situation that hosts would receive a new prefix without deprecation of the previous one. Hosts do not have any standard mechanism to choose only the new prefix for communication. That would lead to a connectivity problem.

How long a non-preferred prefix would be kept in a stale state on the host is not important (default AdvValidLifetime is 30 days in section 6.2.1 of [ND]), because according to [Default Address] section 5 rule#3, it should have a lower priority to be chosen. [SLAAC] section 5.5.4 is another good reference highlighting that address should be avoided after it would reach the deprecated status.

How long an address would stay in the preferred state is important. [ND] instructs hosts to prefer certain prefix for 7 days - see default AdvPreferredLifetime in section 6.2.1.

It is not realistic for the subscriber to wait for 7 days.

It practically means that the subscriber in this corner case would have a few options to fix the problem: (1) reload all hosts, or (2) reconnect the physical link of every host, or (3) reload subtended bridge, or (4) manually delete the prefix on the hosts to clear stale information.

3.2.2. Graceful reload without precautions

Specifically this scenario may happen when developers don't apply precautions in case previous prefixes are not deprecated. It may happen in both architectures.

The router could be reloaded by graceful procedure (reboot or shutdown that would use "init 6" in Unix). It is still possible that software would not send RA with prefix Preferred Lifetime zero to inform hosts about prefix deprecation. This practice prevails because IPv4's centralized address assignments by DHCP does not need similar precautions.

Again, like in the previous section, it would not create a problem in the majority of the cases for directly connected hosts (architecture #1) because link layer would be reinitialized too. The same corner case (architecture #2) would lead to the same result: a connectivity problem that could be resolved only by 4 types of manual intervention mentioned in the previous section.

3.2.3. Abrupt hardware replacement without the possibility for graceful prefix deprecation

Such type of an outage is again may happen only for architecture #2. It would lead to up to 30 minutes (including time for hardware replacement) outage in all cases (to detect missing router) and up to 1 week additional outage if a different prefix would be announced after the hardware replacement.

The hardware could fail or be replaced with an abrupt power disconnect. The latter is very probable for the home environment. Graceful notification of hosts may not happen.

The new hardware may have a different link layer address and a different link local address as a result. The router would look like a new one on the link. Any communication with it could not be the reason to deprecate announcements made early by the router perceived as a different one.

[ND] section 6.2.1 has recommended the AdvDefaultLifetime as $3 * \text{MaxRtrAdvInterval}$. Hosts would send traffic to a non-existent router for up to 30 minutes.

According to section 4.2 of [ND] "Router Lifetime" is related only to router default status. PIO announced early may be preferred up to 7 days according to AdvPreferredLifetime in section 6.2.1 of [ND] even after the router default status is deprecated. The probability for such a situation is the same low as discussed in section 3.2.1. because a different prefix should be announced after hardware reload and a switch should be present between the host and the router. The same corner case would lead to the same result: a connectivity

problem that could be resolved only by 4 types of manual intervention mentioned in section 3.2.1. .

3.2.4. Non-graceful configuration change

This situation may happen due to abrupt prefix change on the router (in both architectures) or VLAN change on the switch (it may happen in architecture #2).

Router configuration could be changed manually, by automation tools, or by protocols (for example, prefix distribution).

Additionally for architecture #2, L2 domain could be abruptly changed by configuration (for example, VLAN change from "quarantine" to "production" without any chance for the router to send a message).

It could lead to the situation that prefix would change abruptly, without any notification to hosts about the necessity to deprecate the previous prefix. Hosts should be notified by prefix announcement with Preferred Lifetime set to zero.

It should not happen for residential CPE because [CPE Requirements] section 4.3 requirement L-13 clearly instructs: "If the delegated prefix changes, i.e., the current prefix is replaced with a new prefix without any overlapping period of time, then the IPv6 CE router MUST immediately advertise the old prefix with a Preferred Lifetime of zero".

But it is perfectly possible for other environments (except residential CPEs) because other routers are not required to do the same: [Node Requirements] does not clarify the exact router behavior in the case of abrupt prefix change. [SLAAC] does not have any recommendations either.

3.2.5. An uplink breaks connectivity without a relevant notification to the connected hosts

It may happen in both architectures #1 and #2.

A router could lose uplink. The probability for such an event is much bigger for a mobile uplink (modem). It would invalidate the possibility to use a PA prefix advertised from this carrier even in the case that another carrier uplink is available on this or redundant router (connectivity to the Internet is not lost). Some mechanism is needed to inform hosts not to use address space from

the disconnected carrier because another carrier would filter it out by anti-spoofing security protection.

The multi-homing multi-prefix PA environment has been properly explained in [MHMP]. The discussion of how traffic should be source-routed by routers in [MHMP] environment is not relevant to our [ND] discussion. Unfortunately, an improper address used as a source would cause a traffic drop as soon as traffic gets to the different carrier.

[Default Address] section 5 (source address selection) rule 5 (for different interfaces on the host) and rule 5.5 (for the same interface) partially prepare hosts for such situation: "Prefer addresses in a prefix advertised by the next-hop. If SA or SA's prefix is assigned by the selected next-hop that will be used to send to D [...] then prefer SA". This algorithm has an assumption that the source address should be chosen after the next hop.

Unfortunately, the rules mentioned above in [Default Address] section 5 would work only if the default router would cease to be default after it loses route to its carrier. It would work only in simplified topology where all hosts connect by L2 to different CPEs, each leading to its separate carrier prefix. It could be called a "common-link environment for all hosts and routers". It is not possible in practice because hosts on the most popular link layer technology (WiFi) are rooted to only one CPE (with AP inside) - they would not switch automatically to different CPE where the Internet connectivity may be still available.

[CPE Requirements] have G-3/4/5 specifically for this simplified multi-homing residential design. It recommends announcing Router Lifetime as zero on LAN if CPE does not have "default router from the uplink" - it would push the host to use another source address by the mentioned above source address selection algorithm.

It is not explained in [CPE Requirements] what should happen with PA delegated prefix after the respective uplink is disconnected. Probably, this is because it was not needed to deprecate stale prefix for the above mentioned mechanism (based on default router withdrawal) to work.

The local residential network could be left without any default router as a result of using the above mechanism - it is especially probable in the single CPE environment. Hence, [CPE Requirements] promotes [ULA] addresses for local connectivity. Default router functionality is returned specifically for [ULA] addresses by

requirement L-3: use "Route Information Option" from [Route Preferences]. It needs hosts' participation in routing through the RIO option.

Unfortunately, this long chain of fixes explained above is strictly optimized for the environment "common-link for all hosts and routers". It is not the case for single WiFi inside any CPE or other topologies.

Neither [ND] nor [SLAAC] instruct the router what to do when the PA delegated prefix is withdrawn abruptly.

[Multi-Homing] section 3 has a good discussion about the proper relationship between default routers and prefixes advertised by respective routers in a stable situation. This would be discussed in more details in section 5.1. . [Multi-Homing] does not discuss what to do in the situation when the router is available, but some uplinks (with delegated prefixes) are lost.

[MHMP] discusses the problem in deep detail with two tools proposed to regulate [ND] behavior: [Policy by DHCP] to change [Default Address] algorithm and [Route Preferences] to inform about appropriate exit points. There are more details later in section 5.1.

4. Root cause analysis

Let's further analyze to be sure that all corner cases are found.

It is assumed in all discussions below that [RA-Guard] is implemented, and all messages are from routers under legitimate administrative control. Security issues are considered as resolved by [RA-Guard], and possibly with extensions in [RA-Guard+].

DHCP is almost as vulnerable as SLAAC for cases found below. DHCP's typical lease time (hours) is shorter than SLAAC's prefix lifetime (days), but is too long for users to accept self-repairing time. Root cause analysis below applies to all possible environments: DHCP, SLAAC, and mixed.

4.1. What to protect

[ND] Router Advertisements deliver configuration information to hosts. Such information could become inaccurate in two different periods of time:

- a) "Recoverable". Time is needed for some process to finish and update information (example: router reload or uplink re-connect).
- b) "Non-recoverable". Time, dependent on some timer expiration (example: complete loss of prefix or default router).

A careful look at the information distributed by RA would give us the understanding that the most problematic is the information that is already protected by deprecation timers: Prefix Information Option and Default Router. Section 3 discusses that the handling of this information is still susceptible to recoverable and non-recoverable periods of inaccuracy.

For example, in the case of abrupt router reload described in sections 3.2.1. -3.2.3. , the recoverable part is the time spent by router and hosts to update their cache after the router reload. The non-recoverable part is related to the setting of the AdvPreferredLifetime timer which would probably force a user to solve the issue with manual intervention.

The next problematic case is the abrupt change of source link-layer address. This problem is not discovered yet in production because it has a low probability. Indeed, a router with a different link-layer address would be treated as a new router, the old router would just disappear from the link. It would affect primarily default router information because all other information should be immediately re-advertised from the new link layer address. Section 6.2.8 of [ND] already discusses how to properly deprecate the default router status of the old link layer address, but no recommendation is given in [ND] for prefix deprecation in this situation. A corner case is possible that software would not treat the new virtual interface as identical concerning the prefix information that should be announced. Different prefixes may be announced. Some additional precautions are needed.

Other information in RA (Hop Limit, MTU, DHCP flags, Reachable timer, and Retransmit timer) are not so sensitive because (1) it is typically static and (2) it does not affect connectivity for respective parameters change in the wide range.

Flag "A" in PIO deserves special attention. It could be cleared abruptly (signaling that hosts should not use this prefix for [SLAAC] anymore). That should not create any problem, because the prefix is still available from a respected PA provider - traffic could be routed to the global Internet. Therefore, it is not vitally important for the host to immediately deprecate the address from

this prefix.

A similar situation is with flag "M" in RA: DHCP address should be deprecated. It should not create a connectivity problem because prefixes could be routed to the global Internet.

4.2. Where to protect

[ND] is the protocol for first-hop connection between host and router. It is designed for one link only. One link could have more than one router.

It is assumed below that a more complex topology (many other routers) is shielded from this link by some other protocol that would deliver all necessary information to those routers.

[HomeNet Architecture] discusses many types of information that should be distributed to every home router. Let's focus on delegated prefixes for our discussion.

The number of uplinks on every router is not important, as long as proper information about prefixes is up to date on the router.

Hence, all our topologies could be simplified into the following scenarios:

- I. L2 device (switch, WiFi AP) and L3 device (router) are in the same device (sharing the fate for power, reboot) (refer to architecture #1 in section 3.1.).
- II. Separate L2 device (probably a switch) and an arbitrary number of L3 devices (routers) are connected to the same IPv6 link (refer to architecture #2 in section 3.1.).

4.3. When to protect: technology scenarios

Let's reorder scenarios discussed in section 3. in the way that it would be better to map to the technology modifications and account for some corner cases found in root cause analysis:

1. Proper prefix usage for Multi-Homing Multi-Prefix environment.
Hosts should be capable of choosing in a coordinated way
 - (1) a source address (from proper PA prefix) and (2) a next hop:
 - A.1. In a normal situation: all providers and prefixes are available

A.2. In a faulty situation: one provider is not reachable, but some hosts and links on the routed path to this provider may still be reachable

A.3. In the case when an administrator abruptly replaces delegated prefix

2. Proper prefix usage for the case of router outage that:

A.4. Planned for this interface
(reboot, shutdown, or ceasing to be a router)

A.5. Abrupt (power outage, software or hardware bug)

A.6. Abrupt (power outage, hardware fault) with hardware replacement

3. Proper prefix usage for the case of link layer address of the router.

These cases are discussed from section 5.1. to section 5.6.

There is no big difference for [ND] between ULA and GUA at the considered link because both could be disjoined at any routed hop upstream. It would need the same invalidation mechanisms on the link. ULA could be invalidated too for the case that ULA spans many sites in a big company. The residential network would probably have a separate ULA for every household that would decrease the probability of ULA prefixes invalidation. It is the responsibility of another protocol (for example, [HNCP]) to decide when ULA should be invalidated, if ever.

5. Solutions

Let's look at the solutions for scenarios listed in section 4.3.

5.1. Multi-homing multi-prefix (MHMP) environment

Let's consider here host capability to choose a proper PA prefix and next hop router in a stable multi-homing multi-prefix (MHMP) environment.

The complex MHMP situation is properly discussed in [MHMP] section 3.1 - it is critical to read it to understand the rest of this section. Our discussion is restricted to [ND] protocol only (one link) - it would cut the number of topologies discussed in section 4.2. MHMP may need additional complex routing interactions that are out of the scope of this document.

It is possible to introduce one additional classification to clearly separate what it is possible to implement now from what needs additional standardization efforts:

1. Case "equal prefixes": Announced prefixes are fully equal by scope and value, all resources interested for hosts could be reachable through any announced PA prefix; additionally, traffic distribution between carriers could be round-robin (no any traffic engineering or policing).
2. Case "non-equal prefixes": Announced prefixes are not equal because (1) some resources could be accessed only through a particular prefix (for example walled garden of one carrier) or (2) it is desirable to have some policy for traffic distribution between PA prefixes (cost of traffic, delay, packet loss, jitter, proportional load).

There are two reminders before the discussion of the above cases:

- o [ND] section 6.3.6 recommends next hop choice between default routers in a round-robin style. Traffic policy or even reachability of particular resources through a particular default router is not considered at the [ND] level.
- o [Default Address] section 7 assumes that source and destination address selection should happen after the next hop (or interface) choice by [ND] or routing, source address is chosen after this.

Case "equal prefixes" does not create any requirement on what prefix should be used for the source address. It is only needed that the source address would be chosen to be compatible with the next hop that should be in the direction of the respective carrier.

No problem is possible for the topology with only one router on the link. The router itself may need source routing to choose next hop properly but it is out of the scope of ND protocol and this document.

Host on a multi-homing link would better be compliant to [Default Address] section 5 (source address selection) rule 5 (for different interfaces on the host) or rule 5.5 (for different next hops on the

same interface). It would help to properly choose a source address compliant to the next hop chosen first. Moreover, if the source address would be chosen wrongly then it is still possible to reroute the packet later by source routing. Hence, it is possible to satisfy the "equal prefixes" case on the current level of standardization developed.

Case "non-equal prefixes" is more complicated. It would be too late to try to solve this problem on a router, because the wrong source address may be already chosen by the host - it would not be possible to contact the appropriate resource in the "walled garden". Only NAT could be left as an option, but that is not a valid choice for IPv6.

There are 2 methods to resolve the case of "non-equal prefixes":

1. The same policies could be formatted differently and fed to the host by two mechanisms: 1) "Routing Information Options" of [Route Preferences] and 2) [Policy by DHCP] to modify policies in [Default Address] selection algorithm. Then current priority of mechanisms could be preserved the same: initially [ND] or routing would choose the next hop, then [Default Address] would choose a source address (and destination if multiple answers from DNS are available). It is the method that is assumed in [MHMP].
2. Alternatively, policies could be supplied only by [Policy by DHCP] to [Default Address] selection algorithm. [Default Address] discusses potential capability in section 7 to reverse algorithm's order: source address may be chosen first, only then to choose next hop (default router). Source address selected from proper carrier is potentially the complete information needed for the host to choose the next hop, but not for the default round-robin distribution between available routers that specified in [ND]. [ND] extension is needed for this method for the host to prioritize default routers that have announced prefixes used for the source address of the considered flow. It is this method that is assumed in [Multi-Homing] section 3.2. This document is different in that the same rules are formulated not as the general advice, but as the particular extension to [ND] - see section 6.1 of this document.

The second method has the advantage that there is no need to download RIO policies by [Route Preferences]. It would simplify the implementation of the MHMP environment. Only the second method is universal and extendable because some policies may not be translated as RIO of [Route Preferences]. For example, dynamic policies (packet loss, delay, and jitter) could

be measured on hosts. Hence, the decision about source address and next hop should be local.

5.2. A provider is not reachable in MHMP environment

Let's assume the fault situation when one provider is not reachable in the [MHMP] environment. A prefix may be very dynamic for a few reasons. It could be received from some protocols (DHCP-PD, HNCP). The prefix could become invalid (at least for the global Internet connectivity) as a result of the abrupt link loss in the upstream direction to the carrier that distributed this prefix.

Additionally, consider the more complicated case when some hosts on the upstream routed path to this provider may still be reachable using a particular prefix but Internet connectivity is broken later.

Let's consider the last problem. Because Internet connectivity is lost for this prefix, it should be announced to hosts by zero Preferred Lifetime. [Route Preferences] gives the possibility to inform hosts that particular a prefix (RIO) is still available on-site but it would be an automation challenge to dynamically calculate and announce prefix. Additionally, [Route Preferences] should be supported by hosts.

In general, it is not a good idea to involve [ND] in routing. Hence, it is better to support on-site connectivity by PI GUA or ULA that may not be invalidated. There are many reasons to promote [ULA] for internal site connectivity: (1) hosts may not have GUA address at all without initial connection to the provider, (2) PA addresses would be invalidated in 30 days of disconnect anyway, (3) it is not a good idea to use addresses from PA pool that is disconnected from global Internet - hosts may have a better option to get global reachability. ULA has better security (open transport ports that are not accessible from the Internet) which is an additional bonus. It is effectively the request to join current [CPE Requirements] and [HomeNet Architecture] requirements in sections 2.2, 2.4, 3.4.2 that subscriber's network should have local ULA addresses.

Prefix deprecation should be done by RA with zero Lifetime for this prefix. It will put the prefix on hosts to the deprecated status that according to many standards ([ND], [SLAAC], and [Default Address]) would prioritize other addresses. Global communication would be disrupted for this prefix anyway. Local communication for deprecated addresses would continue till normal resolution because the default Valid Lifetime is 30 days. Moreover, if it would happen that this delegated prefix was the only one in the local network (no [ULA] for the same reason), then new sessions would be opened on

deprecated prefix because it is the only address available. If connectivity would be re-established and the same prefix would be delegated to the link - it would be announced again with proper preferred lifetime. If a different prefix could be delegated by the PA provider, then the old prefix would stay in deprecated status. It is an advantage for the host that would know about global reachability on this prefix (by deprecated status) because the host may use other means for communication at that time.

Such dynamic treatment of prefixes may have the danger of [ND] messages flood if the link on the path to PA provider would be oscillating.

[HNCP] section 1.1 states: "it is desirable for ISPs to provide large enough valid and preferred lifetimes to avoid unnecessary HNCP state churn in homes".

It makes sense to introduce dampening for the rate of prefix announcements.

Such conceptual change in the treatment of prefixes would not affect current enterprise installations where prefixes are static.

It is important to mention again that it is the responsibility of the respective protocol (that has delivered prefix to the considered router) to inform the router that prefix is not routed anymore to the respective carrier. It is easy to do it in the simplified topology when the only router could correlate uplink status with the DHCP-PD prefix delegated early. Some additional protocols like [HNCP] are needed for a more complex topology.

There is nothing in [ND] or [SLAAC] that prevents us from treating prefixes as something more dynamic than "renumbering" to reflect the dynamic path status to the PA provider. Section 6.2. proposes extensions to [CPE Requirements] and [SLAAC] that follow the logic of this section.

5.3. Administrator abruptly replaces PA prefix

This is the case when the network administrator (maybe from another domain) replaces prefix much faster than 2 hours or the remaining preferred lifetime (as per section 5.5.3 of [SLAAC] on router advertisement processing). The reason for abrupt replacement is probably not related to networking.

Abrupt prefix change may be caused by improper configuration, for example, VLAN change at the switch.

Standards recommend deprecating old prefixes but do not recommend for developers and system designers to additionally check abrupt

configuration changes to mitigate human mistakes. IPv4 cannot mitigate such type of mistake, IPv6 has an advantage here.

Section 6.3. proposes a recommendation for the additional check to make sure that prefix would be deprecated.

This problem could be exacerbated by the low reliability of multicast delivery in a wireless environment - the only packet sent (for example before VLAN change) could be lost. A long-term solution for this problem is proposed in section 6.6 that permits synchronizing host states with a new flag in router announcements.

5.4. Planned router outage

A router could be planned to be put out of service for a link (reboot, shutdown, or ceasing to be a router).

The primary Operating System for routers is LINUX. The following discussion is based on LINUX as an example - other developers can find an analogy for their operating system.

Some LINUX shutdown commands are not graceful in principle (like Halt or Poweroff). It would need extraordinary efforts to send messages discussed in this section before the system would be stopped. It is better to restrict network administrators from such tools on routers.

Other LINUX shutdown commands are safe (Reboot is safe for a long time, Shutdown and "Init 6" have been safe). It would execute shutdown scripts that would give the developer the chance to comply with requirements in this section.

It is up to the developer how reboot and shutdown should be mapped to particular OS commands in graphical user interface (GUI), command line interface (CLI), or automation interface (Netconf/YANG), and what particular actions should be taken. It SHOULD guarantee that section 6.2.5 of [ND] with updates in section 6.4 of this document properly inform hosts that the router is going out of service.

The same procedure SHOULD be automatically activated for cases when an administrator tries manually (via CLI or GUI) or automatically (via Netcong/YANG/Other) to change Link Layer Address on this router interface or disable router functionality in [ND] for this link.

5.5. Prefix information lost because of abrupt router outage

PIO could be lost because of the abrupt reload - the router may not have a chance to warn hosts, but the router could receive a different prefix after reload. Reasons could be (1) power outage, (2) software bug, or (3) hardware problem.

[HomeNet Architecture] section 3.4.3 (Delegated Prefixes) has already recommended usage of non-volatile memory:

"Provisioning such persistent prefixes may imply the need for stable storage on routing devices and also a method for a home user to 'reset' the stored prefix should a significant reconfiguration be required (though ideally the home user should not be involved at all)".

[SLAAC] section 5.7 has recommended storing acquired addresses on hosts in non-volatile memory too.

This document joins these requests and propose adding similar requirements to [CPE Requirements] and [SLAAC] - see section 6.5.

The best long-term solution is to inform the host by [ND] protocol that RA has all information in one announcement. Any missing information SHOULD be considered deprecated. It is possible to do it with the new flag in RA - see section 6.6.

"Complete" flag would become useful only when implemented on both: host and router. It is proposed to rely on storage improvements in non-volatile memory till the "Complete" flag would be supported on many hosts.

5.6. Prefix information lost after hardware replacement

Hardware fault or power outage may follow by hardware replacement.

Prefix storage in non-volatile memory and a "complete" flag would not protect in such a situation. The new router would not have the old prefix information and the "complete" flag would be sourced from a different LLA.

Initially, it would be good to speed up the detection of hardware replacement to delete the stale hardware from the default router list of hosts. It is proposed to request all routers availability by RS all-routers multicast address after new router detection on the link- see section 6.8. It would permit to detect that old hardware is not active in 13 seconds (see section 6.3.7 of [ND] for timers $\text{MAX_RTR_SOLICITATIONS} * \text{RTR_SOLICITATION_INTERVAL} + \text{MAX_RTR_SOLICITATION_DELAY}$). 13 seconds is considered a short enough outage compare to hardware replacement and reload.

Then it is proposed to detect stale prefixes at the event of the respective router deletion from the default router list. If the particular prefix is not announced anymore by any active router on the default router list then the prefix (and all associated addresses) should be deprecated - see section 6.9.

5.7. Link layer address of the router should be changed

Sections 6.3 and 6.4 provide an additional check also in the case of a link layer address change. Hence, additionally resolve LLA change case.

5.8. Dependency between solutions and extensions

It could be useful to map, for quick reference, the dependency between the solutions listed in this section and standard's extensions as presented in section 6.

Solution discussed in		Corresponding extension
5.1.	->	6.1.
5.2.	->	6.2. & 6.7.
5.3.	->	6.3. & 6.6.
5.4.	->	6.4.
5.5.	->	6.5. & 6.6.
5.6.	->	6.8. & 6.9.
5.7.	->	6.3. & 6.4.

6. Extensions of the existing standards

The solution requires a number of standard extensions. They are split into separate sections for better understanding. It is better to read references from section 5. before reading this section, see section 5.8. for cross-reference.

6.1. Default router choice by host

* Section 6.3.6 (Default Router Selection) of [ND], add an initial policy to default router selection:

- 0) For the cases when a particular implementation of ND does know the source address at the time of default router selection (it means that source address was chosen first), then default routers that advertise the prefix for respective source address SHOULD be preferred over routers that do not advertise respective prefix.

6.2. Prefixes become dynamic

* This document joins the request to [CPE Requirements] that has been proposed in section 11 (General Requirements for HNCP Nodes) of [HNCP]:

The requirement L-13 to deprecate prefixes is applied to all delegated prefixes in the network from which assignments have been made on the respective interface. Furthermore, the Prefix Information Options indicating deprecation MUST be included in Router Advertisements for the remainder of the prefixes' respective valid lifetime, but MAY be omitted after at least 2 hours have passed.

* Add section 4.2 into [SLAAC]:

4.2 Dynamic Link Renumbering

Prefix delegation (primarily by DHCP-PD) is adopted by the industry as the primary mechanism of PA address delegation in the fixed and mobile broadband environments, including cases of small business and branches of the big enterprises.

The delegated prefix is tied to dynamic link that has a considerable probability to be disconnected, especially in a mobile environment. The delegated prefix is losing the value if the remote site is disconnected from prefix provider - this fact should be propagated to all nodes on the disconnected site, including hosts. Information Options indicating deprecation (multicast RA with zero Preferred Lifetime) MUST be sent at least one time. It SHOULD be included in Router Advertisements for the remainder of the prefixes' respective valid lifetime but MAY be omitted after 2 hours of deprecation announcements.

There is a high probability that connectivity to the provider would be restored very soon then the prefix could be announced again to all nodes on the site.

There is the probability that in a small period of time the same problem would disconnect the site again (especially for mobile uplink). Such oscillation between available and not available provider could happen frequently that would flood the remote site with [ND] updates.

Dampening mechanism MAY be implemented to suppress oscillation: if the time between a particular prefix announcement and previous deprecation was less than DampeningCheck then delay the next prefix announcement for DampeningDelay and check the need for the prefix announcement after DampeningDelay seconds.

It is recommended for protocol designers to implement a dampening mechanism for protocols (like [HNCP]) that would be used to distribute prefix delegation inside the site to relieve the majority of site routers and the protocol itself from the processing of oscillating messages.

* Section 5.1 (Node Configuration Variables) of [SLAAC], add timers:

DampeningCheck - the time between prefix announcement and previous deprecation is checked against this value to decide about dampening need. The timer should use 16bit unsigned integer measured in seconds. The default value is 10 seconds.

DampeningDelay - the delay (penalty) for the next attempt to announce the same prefix again. The timer should use 16bit unsigned integer measured in seconds. The default value is 10 seconds.

These timers should be configurable like all other timers in [SLAAC] section 5.1.

6.3. Do not forget to deprecate prefixes on renumbering

* Section 4.1 (Site renumbering) of [SLAAC], add at the end:

A network administrator SHOULD avoid the situations when renumbering is done abruptly (with the time of transition that is less than the preferred time for the respective prefix). Situations could happen when it is not possible to archive the above-mentioned goal: (1) the prefix could be withdrawn by the administrator of another domain, (2) there could be the urgent need to change the prefix for reasons not related to networking, (3) prefix could be invalidated after some network event (example: loss of uplink that was used to receive this prefix), (4) L2 connection (VLAN or VPN) could be changed

abruptly by mistake or due to not a proper design. Prefix deprecation MUST be signaled at least one time by multicast RA with Preferred Lifetime set to zero for respective PIO. It SHOULD be included in RA for the remainder of the prefixes' respective valid lifetime but MAY be omitted after 2 hours of deprecation announcements.

It is recommended for developers to check and enforce this rule in router's software: if an administrator, automated system, or other protocol would try to delete a particular prefix from the link and if that prefix has the preferred lifetime bigger than zero, then the software MUST automatically generate deprecation announcements according to the rules explained above.

System designer SHOULD make sure that in the case of abrupt change of logical connectivity at L2 (VLAN, VPN) new default router SHOULD deprecate stale prefixes inherited from the previous default router.

6.4. Do not forget to deprecate prefixes on shutdown

* Section 6.2.5 of [ND] starts from the definition of ceasing cases for the router on [ND] link. One additional reason SHOULD be added to the end of the list:

- Link layer address of the interface should be changed.

* Section 6.2.5 (Ceasing To Be an Advertising Interface) and Section 6.2.8 (Link Local Address Change) of [ND] already discusses requirements of proper ceasing to be [ND] router advertising interface. It has requirements to announce zero for a default router lifetime. It is proposed to add at the end of both sections:

A router MUST also announce in above-mentioned announcements all previously advertised prefixes with zero Preferred LifeTime. Valid LifeTime should not be decreased from originally intended - current hosts sessions should have the possibility to be rerouted to the redundant router (if available).

6.5. Store prefixes in non-volatile memory

Add the same text:

- * [CPE Requirements], new requirement G-6 at the end of section 4.1, and
- * [SLAAC], at the end of section 5.7:

The IPv6 router SHOULD keep in non-volatile memory all prefixes advertised on all links, including prefixes received by dynamic protocols with the reference to the respective protocol (DHCP-PD, HNCP, others).

A router could experience a non-graceful reload.

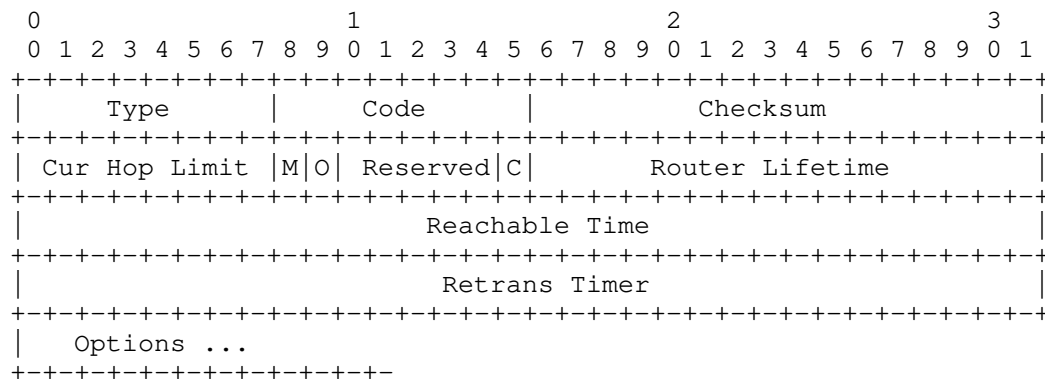
If another protocol would delegate any prefixes for router links then the router SHOULD immediately start announcing them in the normal way.

Additionally, the router should wait until the end of convergence for the respective prefix-delegation protocol. The way for how to decide that convergence is finished is the responsibility of the respective protocol design. It could be a simple timer after uplink would go to "up" or successful exchange by some protocol (like DHCP-PD).

If another protocol would not delegate prefix recorded in non-volatile memory after assumed convergence is achieved, then the old prefix MUST be announced on the link at least one time by multicast RA with the zero Preferred Lifetime. It SHOULD be included in RA for the remainder of the prefixes' respective valid lifetime but MAY be omitted after 2 hours of deprecation announcements.

6.6. Find lost information by "Synchronization"

* Section 4.2 (RA format) of [ND], introduce new flag:



- 0 1-bit "Complete configuration" flag. When set, it indicates that all configuration information has been put inside this RA. The last reserved bit has been chosen to preserve the compatibility with [Route Preferences] that already propose to use the first reserved bit.

* Section 6.2.3 (RA content) of [ND], introduce new flag:

- In the C flag: set if it was possible to put all configuration information into this RA.

* Section 6.2.3 (RA content) of [ND], add at the end:

It is recommended that all configuration information SHOULD be included in one RA (if MTU permits) for multicast and unicast distribution. If successful, then the "Complete" flag SHOULD be set to signal the possibility of synchronization with hosts.

* Section 6.3.4 (RA processing) of [ND], add at the beginning:

After: "the receipt of a Router Advertisement MUST NOT invalidate all information received in a previous advertisement or from another source".

Add: "Except for the case when RA received with "Complete" flag set, then any information from the same router (same Link Local Address) missing in this RA SHOULD be deprecated. Information protected by timers SHOULD be put into the deprecated state. Other information SHOULD be returned to the original state: in compliance to information from other routers or to default configuration if other routers do not announce respected information."

* Section 6.3.4 (RA processing) of [ND], add to the list of PIO processing options:

- If the prefix is missing in RA with the "Complete" flag set, then respective addresses should be put immediately into deprecated state up to the original valid lifetime.

[ND] section 9 mentions: "In order to ensure that future extensions properly coexist with current implementations, all nodes MUST

silently ignore any options they do not recognize in received ND packets and continue processing the packet."

There is a possibility for the gradual introduction of the "Complete" flag:

- o If the host is upgraded to the new functionality first, then the router would send this bit zero (according to the basic [ND]) that would not activate new functionality on the host.
- o If the router is upgraded to the new functionality first, then the host would not pay attention to the flag for Reserved bits.

6.7. Default router announcement rules

* This document joins [HNCP] section 11 (General Requirements for HNCP Nodes) request to [CPE Requirements]:

The generic requirements G-4 and G-5 are relaxed such that any known default router on any interface is sufficient for a router to announce itself as the default router; similarly, only the loss of all such default routers results in self-invalidation.

6.8. Faster detection of the stale default router

* Section 6.3.7 (sending Router Solicitations) of [ND].

The text: "When an interface becomes enabled, a host may be unwilling to wait for the next unsolicited Router Advertisement to locate default routers or learn prefixes. To obtain Router Advertisements quickly, a host SHOULD transmit up to MAX_RTR_SOLICITATIONS Router Solicitation messages, each separated by at least RTR_SOLICITATION_INTERVAL seconds. Router Solicitations may be sent after any of the following events:"

Should be replaced by the text: "

Interface enablement or new router arrival could be the signal of router replacement, a host may be unwilling to wait for the next unsolicited Router Advertisement to locate and invalidate default routers or learn prefixes. To obtain Router Advertisements quickly, a host SHOULD transmit up to MAX_RTR_SOLICITATIONS Router Solicitation messages, each separated by at least RTR_SOLICITATION_INTERVAL seconds. Router Solicitations may be sent after any of the following events:

- the new router is discovered from RA
- . . . <list of other reasons>
- "

* Section 6.3.7 (sending Router Solicitations) of [ND].

After the text: "If a host sends MAX_RTR_SOLICITATIONS solicitations, and receives no Router Advertisements after having waited MAX_RTR_SOLICITATION_DELAY seconds after sending the last solicitation, the host concludes that there are no routers on the link for the purpose of [ADDRCONF]."

Add new text: "If a host sends MAX_RTR_SOLICITATIONS solicitations, and receives no Router Advertisements from the router already present on the default router list after having waited MAX_RTR_SOLICITATION_DELAY seconds, the host concludes that the router SHOULD be deprecated from the default router list."

6.9. Clean orphaned prefixes after default router list change

* Section 6.3.6 (Timing out Prefixes and Default Routers) of [ND] has:

"Whenever the Lifetime of an entry in the Default Router List expires, that entry is discarded. When removing a router from the Default Router list, the node MUST update the Destination Cache in such a way that all entries using the router perform next-hop determination again rather than continue sending traffic to the (deleted) router."

Add at the end:

"All prefixes announced by deprecated default router SHOULD be checked on the announcement from other default routers. If any prefix is not anymore announced from any router - it SHOULD be deprecated."

7. Interoperability analysis

The primary motivation for the proposed changes originated from residential broadband requirements. [ND] extensions proposed in this document should not affect other environments (enterprise WAN,

Campus). Moreover, some precautions proposed could block mistakes originated by humans in some corner cases in all environments.

This document mostly intersects with Homenet working group documents [HomeNet Architecture], [HNCP], and [MHMP]. It was shown that it is possible to isolate [ND] in the context of Homenet to solve specific [ND] problems without any potential impact to the Homenet development and directions.

[CPE Requirements] have the assumption of managing simplified topologies by manipulating routing information injection into [ND]. It has been shown in [MHMP] and in this document that it is better to signal reachability information to [ND] (reachability information to ND sounds strange) by the deprecation of delegated prefixes. This document joins [MHMP] request to change the approach.

[Route Preferences] have been avoided as the mechanism for environments with PA address space because source address is selected first. Then next hop choice can be simplified - see section 5.1 for more details.

[Route Preferences] could still be applicable for PI (Provider-Independent) address environments because only next hops need to be chosen properly.

8. Applicability analysis

Two standard extensions require changes to hosts. Hence, it would take a long time to be implemented in live networks. But workaround exists for the solution to work before it would happen:

- o Absence of implementation for RA information synchronization by C flag on some hosts is not critical because router could use non-volatile memory for prefix storage.
- o Not being capable of excluding a router from the default router list (for the situation when it does not advertise respective prefix) is not critical, because it is needed only for the very advanced MHMP environment with traffic distribution by the policy between different PA providers.
It is for the far future anyway.

9. Security Considerations

This document does not introduce new vulnerabilities.

10. IANA Considerations

This document has no any request to IANA.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [ND] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.
- [SLAAC] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <<https://www.rfc-editor.org/info/rfc4862>>.
- [Route Preferences] R. Draves, D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <<https://www.rfc-editor.org/info/rfc4191>>.
- [Multi-Homing] F. Baker, B. Carpenter, "First-Hop Router Selection by Hosts in a Multi-Prefix Network", RFC 8028, DOI 10.17487/RFC8028, November 2016, <<https://www.rfc-editor.org/info/rfc8028>>.
- [NUD improvement] E. Nordmark, I. Gashinsky, "Neighbor Unreachability Detection Is Too Impatient", RFC 7048, DOI 10.17487/RFC7048, July 2010, <<https://www.rfc-editor.org/info/rfc7048>>.
- [Default Address] D. Thaler, R. Draves, A. Matsumoto, T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <<https://www.rfc-editor.org/info/rfc6724>>.

- [Node Requirements] T. Chown, J. Loughney, T. Winters, "IPv6 Node Requirements", RFC 8504, DOI 10.17487/RFC8504, January 2019, <<https://www.rfc-editor.org/info/rfc8504>>.
- [CPE Requirements] Singh, H., Beebee W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, DOI 10.17487/RFC7084, November 2013, <<https://www.rfc-editor.org/info/rfc7084>>.
- [HomeNet Architecture] T. Chown, J. Arkko, A. Brandt, O. Troan, J. Weil, "IPv6 Home Networking Architecture Principles", RFC 7368, DOI 10.17487/RFC7368, October 2014, <<https://www.rfc-editor.org/info/rfc7368>>.
- [HNCP] M. Stenberg, S. Barth, P. Pfister, "Home Networking Control Protocol", RFC 7788, DOI 10.17487/RFC7788, April 2016, <<https://www.rfc-editor.org/info/rfc7788>>.
- [Policy by DHCP] A. Matsumoto, T. Fujisaki, T. Chown, "Distributing Address Selection Policy Using DHCPv6", RFC 7078 DOI 10.17487/RFC7078, January 2014, <<https://www.rfc-editor.org/info/rfc7078>>.
- [Residential practices] Palet, J., "IPv6 Deployment Survey Residential/Household Services) How IPv6 is being deployed?", UK NOF 39, January 2018, <<https://indico.uknof.org.uk/event/41/contributions/542/attachments/712/866/bcop-ipv6-prefix-v9.pdf>>.
- [SLAAC Robustness] F. Gont, J. Zorz, R. Patterson, "Improving the Robustness of Stateless Address Autoconfiguration (SLAAC) to Flash Renumbering Events", draft-ietf-6man-slaac-renum-02 (work in progress), January 2021

11.2. Informative References

- [RFC8200] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [Flash-Renumbering] F. Gont, J. Zorz, R. Patterson, "Reaction of Stateless Address Autoconfiguration (SLAAC) to Flash-Renumbering Events", RFC 8978, March 2021.

[RA-Guard] E. Levy-Abegnoli, G. Van de Velde, C. Popoviciu, J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, DOI 10.17487/RFC6105, February 2011, <<https://www.rfc-editor.org/info/rfc6105>>.

[RA-Guard+] F. Gont, "Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)", RFC 7113, DOI 10.17487/RFC7113, February 2014, <<https://www.rfc-editor.org/info/rfc7113>>.

[MHMP] O. Troan, D. Miles, S. Matsushima, T. Okimoto, D. Wing, "IPv6 Multihoming without Network Address Translation", RFC 7157, DOI 10.17487/RFC7157, March 2014, <<https://www.rfc-editor.org/info/rfc7157>>.

[ULA] R. Hinden, B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<https://www.rfc-editor.org/info/rfc4193>>.

12. Acknowledgments

Thanks to 6man working group for problem discussion.

Authors' Addresses

Olorunloba Olopade
Virgin Media
270 & 280 Bartley Way, Bartley Wood Business Park, Hook,
Hampshire RG27 9UP
Email: Loba.Olopade@virginmedia.co.uk

Eduard Vasilenko
Huawei Technologies
17/4 Krylatskaya st, Moscow, Russia 121614
Email: vasilenko.eduard@huawei.com

Paolo Volpato
Huawei Technologies
Via Lorenteggio 240, 20147 Milan, Italy
Email: paolo.volpato@huawei.com

