IPv6 maintenance Working Group (6man)
Internet-Draft
Intended status: Informational
Expires: May 3, 2018

F. Gont
SI6 Networks / UTN-FRH
G. Gont
SI6 Networks
M. Garcia Corbo
SITRANS
C. Huitema
Private Octopus Inc.
October 30, 2017

Problem Statement Regarding IPv6 Address Usage
draft-gont-6man-address-usage-recommendations-04

Abstract

This document analyzes the security and privacy implications of IPv6 addresses based on a number of properties (such as address scope, stability, and usage type), and identifies gaps that currently prevent systems and applications from leveraging the increased flexibility and availability of IPv6 addresses.

Status of This Memo

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

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

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

This Internet-Draft will expire on May 3, 2018.

Copyright Notice

Table of Contents

1. Introduction 2
2. Terminology 3
3. Background 3
4. IPv6 Address Properties 4
4.1. Address Scope Considerations 4
4.2. Address Stability Considerations 5
4.3. Usage Type Considerations 6
5. Default Address Selection in IPv6 7
6. Current Possible Approaches for IPv6 Address Usage 8
6.1. Incoming communications 9
6.2. Outgoing communications 9
7. Problem Statement 10
7.1. Issues Associated with Sub-optimal IPv6 Address Usage . . 10
7.1.1. Correlation of Network Activity 10
7.1.2. Testing for the Presence of Node in the Network . . . 10
7.1.3. Unexpected Address Discovery 11
7.1.4. Availability Outside the Expected Scope 11
7.2. Current Limitations in the Address Selection APIs 12
7.3. Sub-optimal IPv6 Address Configuration 12
7.4. Sub-optimal IPv6 Address Usage 13
8. IANA Considerations 14
9. Security Considerations 14
10. Acknowledgements 14
11. References 14
11.1. Normative References 14
11.2. Informative References 15
Authors' Addresses 16

1. Introduction

IPv6 addresses may differ in a number of properties, such as address scope (e.g. link-local vs. global), stability (e.g. stable addresses vs. temporary addresses), and intended usage type (outgoing communications vs. incomming communications). While often overlooked, these properties have impact on areas such as security, privacy, and performance.

IPv6 hosts typically configure a number of IPv6 addresses of different properties. For example, a host may configure one stable and one temporary address per each autoconfiguration prefix

advertised on the local network. Currently, the addresses to be configured typically depend on local system policy, with the aforementioned policy being static and irrespective of the network the host attaches to. This "one size fits all" approach limits the ability of systems and applications of fully-leveraging the increased flexibility and availability of IPv6 addresses.

Each application running on a given system may have its own set of requirements or expectations for the properties of the IPv6 addresses to be employed. For example, an application meaning to offer a public service might expect to employ global stable addresses for such purpose, while a privacy-sensible client application might prefer short-lived temporary addresses, or might even expect to employ single-use ("throw-away") IPv6 addresses when connecting to public servers. However, the subtetlies associated with IPv6 addresses (and associated properties) are often ignored by application programmers and, in any case, current APIs (such as the BSD Sockets API) tend to be very limited in the amount of control they give applications to select the most appropriate IPv6 addresses for a given task, thus limiting a programmer’s ability to leverage IPv6 address availability and properties.

This document analyzes the impact of a number of properties of IPv6 addresses on areas such as security and privacy, and analyzes how IPv6 addresses are curently generated and employed by different operating systems and applications. Finally, it provides a problem statement by identifying and analyzing gaps that prevent systems and applications from fully-leveraging IPv6 addressing capabilities, setting the basis for new work that could fill those gaps.

2. Terminology

This document employs the definitions of "public address", "stable address", and "temporary address" from Section 2 of [RFC7721].

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

3. Background

Predictable IPv6 addresses result in a number of security and privacy implications. For example, [Barnes2012] discusses how patterns in network prefixes can be leveraged for IPv6 address scanning. On the other hand, [RFC7707], [RFC7721] and [RFC7217] discuss the security and privacy implications of predictable IPv6 Interface Identifiers (IIDs).

Given the aforementioned previous work in this area, and the formal specification update produced by [RFC8064], we expect (and assume in the rest of this document) that implementations have replaced any schemes that produce predictable addresses with alternative schemes that avoid such patterns (e.g., RFC7217 in replacement of the traditional SLAAC addresses that embed link-layer addresses).

4. IPv6 Address Properties

There are three parameters that affect the security and privacy properties of an IPv6 address:

- Scope
- Stability
- Usage type (client-like "outgoing connections" vs. server-like "incoming connections")

Section 4.1, Section 4.2, and Section 4.3 discuss the security and privacy implications (and associated tradeoffs) of the scope, stability and usage type properties of IPv6 addresses, respectively.

4.1. Address Scope Considerations

The IPv6 address scope can, in some scenarios, limit the attack exposure of a node as a result of the implicit isolation provided by a non-global address scope. For example, a node that only employs link-local addresses may, in principle, only be exposed to attack from other nodes in the local link. Hosts employing only Unique Local Addresses (ULAs) may be more isolated from attack than those employing Global Unicast Addresses (GUAs), assuming that proper packet filtering is enforced at the network edge.

The potential protection provided by a non-global addresses should not be regarded as a complete security strategy, but rather as a form of "prophylactic" security (see [I-D.gont-opsawg-firewalls-analysis]).

We note that the use of non-global addresses is usually limited to a reduced type of applications/protocols that e.g. are only meant to operate on a reduced scope, and hence their applicability may be limited.

A discussion of ULA usage considerations can be found in [I-D.ietf-v6ops-ula-usage-considerations].

4.2. Address Stability Considerations

The stability of an address has two associated security/privacy implications:

- Ability of an attacker to correlate network activity
- Exposure to attack

For obvious reasons, an address that is employed for multiple communication instances allows the aforementioned network activities to be correlated. The longer an address is employed (i.e., the more stable it is), the longer such correlation will be possible. In the worst-case scenario, a stable address that is employed for multiple communication instances over time will allow all such activities to be correlated. On the other hand, if a host were to generate (and eventually "throw away") one new address for each communication instance (e.g., TCP connection), network activity correlation would be mitigated.

NOTE:
> The use of constant IIDs (as in traditional SLAAC) result in addresses that, while not constant as a whole (since the prefix changes), contain a globally-unique value that leaks out the node "identity". Such addresses result in the worst possible security and privacy implications, and their use has been deprecated by [RFC8064].

Typically, when it comes to attack exposure, the longer an address is employed the longer an attacker is exposed to attacks (e.g. an attacker has more time to find the address in the first place [RFC7707]). While such exposure is traditionally associated with the stability of the address, the usage type of the address (see Section 4.3) may also have an impact on attack exposure.

A popular approach to mitigate network activity correlation is the use of "temporary addresses" [RFC4941]. Temporary addresses are typically configured and employed along with stable addresses, with the temporary addresses employed for outgoing communications, and the stable addresses employed for incoming communications.

NOTE:
> Ongoing work [I-D.gont-6man-non-stable-iids] aims at updating [RFC4941] such that temporary addresses can be employed without the need to configure stable addresses.

We note that the extent to which temporary addresses provide improved mitigation of network activity correlation and/or reduced attack

exposure may be questionable and/or limited in some scenarios. For example, a temporary address that is reachable for, say, a few hours has a questionable "reduced exposure" (particularly when automated attack tools do not typically require such a long period of time to complete their task). Similarly, if network activity can be correlated for the life of such address (e.g., on the order of several hours), such period of time might be long enough for the attacker to correlate all the network activity he is meaning to correlate.

In order to better mitigate network activity correlation and/or possibly reduce host exposure, an implementation might want to either reduce the preferred lifetime of a temporary address, or even better, generate one new temporary address for each new transport protocol instance. However, the associated lifetime/stability of an address may have a negative impact on the network. For example, if a node were to employ "throw away" IPv6 addresses, or employ temporary addresses [RFC4941] with a short preferred lifetime, local nodes might need to maintain too many entries in their Neighbor Cache, and a number of devices (possibly enforcing security policies) might also need to cope with such additional state.

Additionally, enforcing a maximum lifetime on IPv6 addresses may cause long-lived TCP connections to fail. For example, an address becoming "Invalid" (after transitioning through the "Preferred" and "Deprecated" states) would cause the TCP connections employing them to break. This, in turn, would cause e.g. long-lived SSH sessions to break/fail.

In some scenarios, attack exposure may be reduced by limiting the usage of temporary addresses to outgoing connections, and prevent such addresses from being used for incoming connections (please see Section 4.3).

4.3. Usage Type Considerations

A node that employs one of its addresses to communicate with an external server (i.e., to perform an "outgoing connection") may cause such address to become exposed to attack. For example, once the external server receives an incoming connection, the corresponding server might launch an attack against the aforementioned address. A real-world instance of this type of scenario has been documented in [Hein].

However, we note that employing an IPv6 address for outgoing communications need not increase the exposure of local services to other parties. For example, nodes could employ temporary addresses only for outgoing connections, but not for incoming connections.

Thus, external nodes that learn about client's addresses could not really leverage such addresses for actively contacting the clients.

There are multiple ways in which this could possibly be achieved, with different implications. Namely:

- Run a host-based or network-based firewall
- Bind services to specific (explicit) addresses
- Bind services only to stable addresses

A client could simply run a host-based firewall that only allows incoming connections on the stable addresses. This is clearly more of an operational way of achieving the desired functionality, and may require good firewall/host integration (e.g., the firewall should be able to tell stable vs. temporary addresses), may require the client to run additional firewall software for this specific purpose, etc. In other scenarios, a network-based firewall could be configured to allow outgoing communications from all internal addresses, but only allow incoming communications to stable addresses. For obvious reasons, this is generally only applicable to networks where incoming communications are allowed to a limited number of hosts/servers.

Services could be bound to specific (explicit) addresses, rather than to all locally-configured addresses. However, there are a number of short-comings associated with this approach. Firstly, an application would need to be able to learn all of its addresses and associated stability properties, something that tends to be non-trivial and non-portable, and that also makes applications protocol-dependent, unnecessarily. Secondly, the BSD Sockets API does not really allow a socket to be bound to a subset of the node's addresses. That is, sockets can be bound to a single address or to all available addresses (wildcard), but not to a subset of all the configured addresses.

Binding services only to stable addresses provides a clean separation between addresses employed for client-like outgoing connections and server-like incoming connections. However, we currently lack an appropriate API for nodes to be able to specify that a socket should only be bound to stable addresses.

5. Default Address Selection in IPv6

Applications use system API's to select the IPv6 addresses that will be used for incoming and outgoing connections. These choices have consequences in terms of privacy, security, stability and performance.

Default Address Selection for IPv6 is specified in [RFC6724]. The selection starts with a set of potential destination addresses, such as returned by getaddrinfo(), and the set of potential source addresses currently configured for the selected interfaces. For each potential destination address, the algorithm will select the source address that provides the best route to the destination, while choosing the appropriate scope and preferring temporary addresses. The algorithm will then select the destination address, while giving a preference to reachable addresses with the smallest scope. The selection may be affected by system settings. We note that [RFC6724] only applies for outgoing connections, such as those made by clients trying to use services offered by other hosts.

We note that [RFC6724] selects IPv6 addresses from all the currently available addresses on the host, and there is currently no way for an application to indicate expected or desirable properties for the IPv6 source addresses employed for such outgoing communications. For example, a privacy-sensitive application might want that each outgoing communication instance employs a new, single-use IPv6 address, or to employ a new reusable address that is not employed or reusable by any other application on the host. Reuse of an IPv6 address by an application would allow the correlation of all network activities corresponding to such application as being performed by the same host, while reuse of an IPv6 address by multiple different applications would allow the correlation of all such network activities as being performed by the host with such IPv6 address.

When devices provide a service, the common pattern is to just wait for connections over all addresses configured on the device. For example, applications using the BSD Sockets API will commonly bind() the listening socket to the undefined address. This long-established behavior is appropriate for devices providing public services, but may have unexpected results for devices providing semi-private services, such as various forms of peer-to-peer or local-only applications.

This behavior leads to three problems: device tracking, discussed in Section 7.1.2; unexpected address discovery, discussed in Section 7.1.3; and availability outside the expected scope, discussed in Section 7.1.4. These problems are caused in part by the limitations of available address selection API, presented in Section 7.2.

6. Current Possible Approaches for IPv6 Address Usage

6.1. Incoming communications

There are a number of ways in which a system or network may affect which address (and how) may be employed for different services and cases. Namely,

- TCP/IP stack address filtering
- Application-based address filtering
- Firewall-based address filtering

Clearly, the most elegant approach for address selection is for applications to be able to specify the properties of the addresses they are willing to employ by means of an API, such the TCP/IP stack itself can "filter" which addresses are allowed to be employed for the given service/application. This relieves the application from dealing with low level details of networking, improves portability, and avoids duplicate code in applications. However, constraints in the current APIs (see Section 7.2) may limit the ability of application progremmers for leveraging this technique.

Another possible approach is for applications to e.g. bind services to all available addresses, and perform the associated selection/ filtering at the application level. While possible this has a number of drawbacks. Firstly, it would require applications to deal with low-level networking details, require that all the associated code be duplicated in all applications, and also negatively affect portability. Besides, performing address/selection filtering at the application level may not mitigate some possible threats. For example, port scanning will still be possible, since the aforementioned filtering will only be performed e.g. once UDP packets are received or TCP connections are established.

Finally, a firewall may be employed to filter addresses based on their intended usage. For example, a firewall may block incoming requests to all addresses except to some whitelisted addresses (such as the stable addresses of the node). This technique not only requires the use of a firewall (which may or may not be present), but also implies knowledge of the firewall regarding the desired properties of the addresses that each application/service is intended to use.

6.2. Outgoing communications

An application might be able to obtain the list of currently-configured addresses, and subsequently select an address with desired

properties, and explicitly "bind" the address to the socket, to override the default source address selection.

However, this approach is problematic for a number of reasons. Firstly, there is no portable way of obtaining the list of currently-configured addresses on the local node, and even less to check for properties such "valid lifetime". Secondly, as discussed in Section 6.1, it would require application programmers to understand all the subtetiles associated with IPv6 addressing, and would also lead to duplicate code on all applications. Finally, applications would be limited to use already-configured addresses and unable to trigger the generation of new addresses where desirable (e.g. the genration of a new temporary address for this application instance or communication instance).

7. Problem Statement

This section elaborates the problem statement on IPv6 address usage. Section 7.1 describes the security and privacy implications of improper IPv6 address usage, while Section 7.2, Section 7.4, Section 7.3, analyze the possible root of such improper address usage, suggesting possible future work.

7.1. Issues Associated with Sub-optimal IPv6 Address Usage

7.1.1. Correlation of Network Activity

As discussed in [RFC7721], a node that reuses an IPv6 address for multiple communication instances would allow the correlation of such network activities. This could be the case when the same IPv6 address is employed by several instances of the same application (e.g., a browser in "privacy" mode and a browser in "normal" mode), or when the same IPv6 address is employed by two different applications on the same node (e.g., a browser in "privacy" mode, and an email client).

Particularly for privacy-sensitive applications, an application or system might want to limit the usage of a given IPv6 address to a single communication instance, a single application, a single user on the system, etc. However, given current APIs, this is practically impossible.

7.1.2. Testing for the Presence of Node in the Network

The stable addresses recommended in [RFC8064] use stable IIDs defined in [RFC7217]. One key part of that algorithm is that if a device connects to a given network at different times, it will always configure the same IPv6 addresses on that network. If the device

hosts a service ready to accept connections on that stable address, adversaries can test the presence of the device on the network by attempting connections to that stable address. Stable addresses used by listening services will thus enable testing whether a specific device is returning to a particular network, which in a number of cases might be considered a privacy issue.

7.1.3. Unexpected Address Discovery

Systems like DNS-Based Service Discovery [RFC6763] allow clients to discover services within a limited scope, that can be defined by a domain name. These services are not advertised outside of that scope, and thus do not expect to be discovered by random parties on the Internet. However, such services may be easily discoverable if they listen for connections to IPv6 addresses that a client process also uses as source address when connecting to remote servers.

NOTE:
An example of such unexpected discovery is described in [Hein]. A network manager observed scanning traffic directed at the temporary addresses of local devices. The analysis in [Hein] shows that the scanners learned the addresses by observing the device contact an NTP service ([RFC5905]). The remote scanning was possible because the local devices were also accepting connections directed to the temporary addresses.

It is obvious from the example that the "attack surface" of the services is increased because they are bond to the same IPv6 addresses that are also used by clients for outgoing communications with remote systems. But the overlap between "client" and "server" addresses is only one part of the problem. Suppose that a device hosts both a video game and a home automation application. The video game users will be able to discover the IPv6 address of the game server. If the home automation server listens to the same IPv6 addresses, it is now exposed to connection attempts by all these users. That, too, increases the attack surface of the home automation server.

7.1.4. Availability Outside the Expected Scope

The IPv6 addressing architecture [RFC4291] defines multiple address scopes. In practice, devices are often configured with globally reachable unicast addresses, link local addresses, and Unique Local IPv6 Unicast Addresses (ULA) [RFC4193]. Availability outside the expected scope happens when a service is expected to be only available in some local scope, but inadvertently becomes available to remote parties. That could happen for example if a service is meant to be available only on a given link, but becomes reachable through

ULA or through globally reachable addresses, or if a service is meant to be available only inside some organization's perimeter and becomes reachable through globally reachable addresses. It will happen in particular if a service intended for some local scope is programmed to bind to "unspecified" addresses, which in practice means every address configured for the device (please see Section 7.2).

7.2. Current Limitations in the Address Selection APIs

Application developers using the BSD Sockets API can "bind" a listening socket to a specific address, and ensure that the application is only reachable through that address. In theory, careful selection of the binding address could mitigate the problems described in Section 7.1. Binding services to temporary addresses could mitigate the ability of an attacker from testing for the presence of the node in the network. Binding different services to different addresses could mitigate unexpected discovery. Binding services to link local addresses or ULA could mitigate availability outside the expected scope. However, explicitly managing addresses adds significant complexity to the application development. It requires that application developers master addressing architecture subtleties, and implement logic that reacts adequately to connectivity events and address changes. Experience shows that application developers would probably prefer some much simpler solution.

In addition, we should note that many application developers use high level APIs that listen to TLS, HTTP, or some other application protocol. These high level APIs seldom provide detailed access to specific IP addresses, and typically default to listening to all available addresses.

A more advanced API could allow an application programmer to select desired properties in an address (scope, lifespan, etc.), such that the best-suitable addresses are selected, while relieving the application for low-level IPv6 addressing details. Such API might also trigger the generation of new IPv6 addresses when the specified properties would require so.

7.3. Sub-optimal IPv6 Address Configuration

Most operating systems configure the same types of addresses regardless of the current "operating mode" or "profile" of the device (e.g., device connected to enterprise network vs roaming across untrusted networks). For example, many operating systems configure both stable [RFC8064] and temporary [RFC4941] addresses on all network interfaces. However, this "one size fits all" approach tends to be sub-optimal or inappropriate for some scenarios. For example,

enterprise networks typically prefer usage of only stable address, thus meaning that a network administator needs to find the means for disabling the generation of temporary addresses on all those systems that would otherwise generate them. On the other hand, some mobile devices configure both stable and temporary addresses, even when their usage pattern (client-like operation, as opposed to offering services to other nodes) would allow for the more privacy-sensible option of configuring only temporary addresses.

The lack of better tuned address configuration policies has helped the "one size fits all" approach that, as noted, may lead to suboptimal results. Advice in this area might help achieve more optional address generation policies such that IPv6 addressing capabilities are fully leveraged.

NOTE:
One might envision a document that provides advice regarding the address generation for different typical scenarios (e.g., when to configure stable-only, temporary-only, or stable+temporary). In the most simple analysis, one might expect nodes in a typical enterprise network to employ only stable addresses. General-purpose nodes in a home or "trusted" network may want to employ both stable and temporary addresses. Finally, mobile nodes (e.g. when roaming across non-trusted networks) may want to employ only temporary addresses).

7.4. Sub-optimal IPv6 Address Usage

An application programmer, left with the question of which are the most appropriate addresses for a given usage type and application, typically resorts to the Default IPv6 Address Selection for IPv6 (see Section 5) for outgoing communications, and to accepting incoming communications on all available addresses for incoming communications. As discussed throughout this document, this leads to sub-optimal results. Besides, all applications on a node share the same pool of configured addresses, and applications are also prevented from triggering the generation of new addresses (e.g. to be employed for a particular application or communcation instance).

Guidance in this area is warranted such that applications and systems fully-leverage IPv6 addressing.

NOTE:
Such guidance would elaborate, among other things, on the usage of IPv6 addresses when offering network services and when performing client-like communications. For example, for incomming communications, hosts might want to employ only the smallest-scope applicable addresses (if available) and, if stable addresses are

available, they might want to accept incoming connections only on such addresses (but *not* on temporary addresses). For client-like communications, hosts might prefer temporary addresses, unless the coresponding communication instances are expected to be long-lived (e.g., SSH sessions).

8. IANA Considerations

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

9. Security Considerations

The security and privacy implications associated with the predictability and lifetime of IPv6 addresses has been analyzed in [RFC7217] [RFC7721], and [RFC7707]. This document complements and extends the aforementioned analysis by considering other IPv6 properties such as the address scope and address usage type, and the associated tradeoffs. Finally, it describes possible future standards-track work to allow for greater flexibility in IPv6 address usage.

10. Acknowledgements

The authors would like to thank (in alphabetical order) Francis Dupont, Tatuya Jinmei, and Dave Thaler for providing valuable comments on earlier versions of this document.

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>.

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

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <https://www.rfc-editor.org/info/rfc4941>.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <https://www.rfc-editor.org/info/rfc5905>.

[RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <https://www.rfc-editor.org/info/rfc6724>.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <https://www.rfc-editor.org/info/rfc6763>.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <https://www.rfc-editor.org/info/rfc7217>.

[RFC8064] Gont, F., Cooper, A., Thaler, D., and W. Liu, "Recommendation on Stable IPv6 Interface Identifiers", RFC 8064, DOI 10.17487/RFC8064, February 2017, <https://www.rfc-editor.org/info/rfc8064>.

11.2. Informative References

[Barnes2012] Barnes, R., Altmann, R., and D. Kerr, "Mapping the Great Void Smarter scanning for IPv6", ISMA 2012 AIMS-4 - Workshop on Active Internet Measurements, February 2012, <https://www.caida.org/workshops/isma/1202/slides/aims1202_rbarnes.pdf>.

[Hein] Hein, B., "The Rising Sophistication of Network Scanning", January 2016, <http://netpatterns.blogspot.be/2016/01/the-rising-sophistication-of-network.html>.

[I-D.gont-6man-non-stable-iids] Gont, F., Huitema, C., Gont, G., and M. Corbo, "Recommendation on Temporary IPv6 Interface Identifiers", draft-gont-6man-non-stable-iids-01 (work in progress), March 2017.

[I-D.gont-opsawg-firewalls-analysis]
Gont, F. and F. Baker, "On Firewalls in Network Security", draft-gont-opsawg-firewalls-analysis-02 (work in progress), February 2016.

[I-D.ietf-v6ops-ula-usage-considerations]
Liu, B. and S. Jiang, "Considerations For Using Unique Local Addresses", draft-ietf-v6ops-ula-usage-considerations-02 (work in progress), March 2017.

[RFC7707] Gont, F. and T. Chown, "Network Reconnaissance in IPv6 Networks", RFC 7707, DOI 10.17487/RFC7707, March 2016, <https://www.rfc-editor.org/info/rfc7707>.

[RFC7721] Cooper, A., Gont, F., and D. Thaler, "Security and Privacy Considerations for IPv6 Address Generation Mechanisms", RFC 7721, DOI 10.17487/RFC7721, March 2016, <https://www.rfc-editor.org/info/rfc7721>.

Authors' Addresses

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

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

Guillermo Gont
SI6 Networks
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: ggont@si6networks.com
URI: https://www.si6networks.com

Madeleine Garcia Corbo
Servicios de Informacion del Transporte
Neptuno 358
Havana City 10400
Cuba

Email: madelen.garcia16@gmail.com

Christian Huitema
Private Octopus Inc.
Friday Harbor, WA 98250
U.S.A.

Email: huitema@huitema.net
URI: http://privateoctopus.com

IPv6 maintenance Working Group (6man)
Internet-Draft
Intended status: Standards Track
Expires: September 25, 2018

F. Gont
SI6 Networks / UTN-FRH
C. Huitema
Private Octopus Inc.
S. Krishnan
Ericsson Research
G. Gont
SI6 Networks
M. Garcia Corbo
SITRANS
March 24, 2018

Recommendation on Temporary IPv6 Interface Identifiers
draft-gont-6man-non-stable-iids-04

Abstract

This document specifies a set of requirements for generating temporary addresses, and clarifies the stability requirements for IPv6 addresses, allowing for the use of only temporary addresses.

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 25, 2018.

Copyright Notice

Table of Contents

1. Introduction 2
2. Terminology 3
3. Problem statement 3
4. Stability Requirements for IPv6 Addresses 4
5. Requirements for Temporary IPv6 Addresses 4
6. Future Work 6
7. IANA Considerations 6
8. Security Considerations 6
9. Acknowledgements 7
10. References 7
Authors' Addresses 10

1. Introduction

IPv6 StateLess Address AutoConfiguration (SLAAC) [RFC4862] has traditionally resulted in stable addresses, since the Interface Identifier (IID) has been generated by embedding a stable layer-2 numeric identifier (e.g., a MAC address). [RFC4941] originally implied, throughout the specification, that temporary addresses are generated and employed along with stable addresses.

While the use of stable addresses (only) or mixed stable and temporary addresses can be desirable in a number of scenarios, there are other scenarios in which, for security and privacy reasons, a node may want to use only temporary address (e.g., a temporary address).

On the other hand, the lack of a formal set of requirements for temporary addresses led to a number of flaws in popular implementations and in the protocol specification itself, such as allowing for the correlation of network activity carried out with different addresses, reusing randomized identifiers across different networks, etc.

This document clarifies the requirements for stability of IPv6 addresses, such that nodes are not required to configure stable addresses, and may instead employ only temporary addresses. It also specifies a set of requirements for the generation of temporary addresses.

2. Terminology

Statistically different:
When two values are required to be "statistically different", it means that the equality of those values cannot be caused by anything else other than random chance.

This document employs the definitions of "stable address" and "temporary address" from [RFC7721].

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

3. Problem statement

When [RFC4941] was written, its authors wanted to prevent privacy and security attacks enabled by addresses that contain "an embedded interface identifier, which remains constant over time". They observed that "Anytime a fixed identifier is used in multiple contexts, it becomes possible to correlate seemingly unrelated activity using this identifier." They were concerned with both on-path attackers who would observe the IP addresses of packets observed in transit, and attackers that would have access to the logs of servers.

Since the publication of [RFC4941] in September 2007, our understanding of threats and mitigations has evolved. The IETF is now officially concerned with Pervasive Monitoring [RFC7258], as well as the wide spread collection of information for advertising and other purposes, for example through the Real Time Bidding protocol used for advertising auctions [RTB25].

3.1. Privacy requirements

The widespread deployment of encryption advocated in [RFC7624] is a response to Pervasive Monitoring. Encryption of communication reduces the amount of information that can be collected by monitoring data links, but does not prevent monitoring of IPv6 addresses embedded in clear text packet headers. Stable IPv6 addresses enable the correlation of such data over time.

MAC Address Randomization [IETFMACRandom] is another response to pervasive monitoring. In conjunction with DHCP Anonymity [RFC7844], it ensures that devices cannot be tracked by their MAC Address or their DHCP identifiers when they connect to "hot spots". However, the privacy effects of MAC Address Randomization would be nullified

if a device kept using the same IPv6 address before and after a MAC-address randomization event.

Many Web Browsers have options enabling browsing "in private". However, if the web connections during the private mode use the same IPv6 address as those in the public mode, web tracking systems similar to [RTB25] will quickly find the correlation between the public personna of the user and the supposedly private connection. Similarly, many web browsers have options to "delete history", including deleting "cookies" and other persistent data. Again, if the same IPv6 address is used before and after the deletion of cookies, web tracking systems will easily correlate the new activity with the prior data collection.

Using temporary address alone may not be sufficient to prevent all forms of tracking. It is however quite clear that some usage of temporary addresses is necessary to provide user privacy. It is also clear that the usage of temporary addresses needs to be synchronized with other privacy defining event such as moving to a new network, performing MAC Address Randomization, or changing the privacy posture of a node.

4. Stability Requirements for IPv6 Addresses

Nodes are not required to generate addresses with any specific stability properties. That is, the generation of stable addresses is OPTIONAL. This means that a node may end up configuring only stable addresses, only temporary, or both stable and temporary addresses.

5. Requirements for Temporary IPv6 Addresses

The requirements for temporary IPv6 addresses are as follows:

1. Temporary addresses MUST have a limited lifetime (limited "valid lifetime" and "preferred lifetime" from [RFC4862]), that should be statistically different for different addresses. The lifetime of an address essentially limits the extent to which network activity correlation can be performed for such address.

2. The lifetime of an address MUST be further reduced when privacy-meaningful events (such as a node attaching to a new network) takes place.

3. The resulting Interface Identifiers MUST be statistically different when addresses are configured for different prefixes. That is, when temporary addresses are generated for different autoconfiguration prefixes for the same network interface, the resulting Interface Identifiers must be statistically different.

This means that, given two addresses that employ different prefixes, it must be difficult for an outside entity to tell whether the addresses correspond to the same network interface or even whether they have been generated by the same host.

4. It must be difficult for an outside entity to predict the Interface Identifiers that will be employed for temporary addresses, even with knowledge of the algorithm/method employed to generate them and/or knowledge of the Interface Identifiers previously employed for other temporary addresses.

5. The resulting Interface Identifiers MUST be semantically opaque [RFC7136] and MUST NOT follow any specific patterns.

By definition, temporary addresses have a limited lifetime. This is in contrast with e.g. stable addresses [RFC7217], that are not expected to become invalid under normal circumstances. Employing statistically different lifetimes for different addresses prevents an observer from synchronizing with the temporary address regeneration; that is, from being able to predict when a temporary address will become invalid and a new one regenerated, and thus being able to infer that one newly observed address is actually the result of regenerating a previously observed one.

The lifetime of an address should be further reduced by privacy-meaningful events. For example, a host must not employ the same address across network attachment events. That is, a host that de-attaches from a network and subsequently re-attaches to a (possibly different) network should regenerate all of its temporary addresses. Similarly, a host that implements MAC address randomization should regenerate all of its temporary addresses. Failure to regenerate temporary addresses upon such events would allow the correlation of network activity across such events (e.g., correlation of network activity as a host moves from one network to another). Other events, such as those discussed in Section 3.1 should also trigger the regeneration of all temporary addresses.

Temporary addresses configured for different prefixes should employ statistically different interface identifiers. In general, the reuse of identifiers across different contexts or scopes can be detrimental for security and privacy [I-D.gont-predictable-numeric-ids] [RFC6973] [RFC4941]. For example, a node that deterministically employs the same interface identifier for generating temporary addresses for different prefixes will allow the correlation of network activity.

For security and privacy reasons, the IIDs generated for temporary addresses must be unpredictable by an outside entity. Otherwise, the node may be subject to many (if not all) of the security and privacy

issues that temporary addresses are expected to mitigate (please see [RFC7721]).

Any semantics or patterns in an IID might be leveraged by an attacker to e.g. reduce the search space when performing address-scanning attacks (see [RFC7707], infer the identity of the node, etc.

NOTE:
In the above text, where the "lifetime" of different addresses is required to be statistically different, or where the interface identifiers for different temporary addresses is required to be statistically different, the goal is that an implementation must not deterministically employ the same such values for different addresses. For example, where interface identifiers for different temporary addresses are required to be statistically different, the goal is to e.g. prevent an implementation from computing a single random interface identifier and employing such identifier for the generation of temporary addresses for other prefixes for the same network interface (as was the case with the algorithm specified in [RFC4941]). Therefore, a node is neither required nor expected to e.g. enforce that a newly-generated random interface identifier is not currently employed by any other temporary address configured by the node, or that such interface identifier has not been previously employed for any other temporary address configured by the node.

6. Future Work

This document clarifies the requirements for stability requirements for IPv6 addresses, and specifies requirements for temporary addresses. A separate document ([I-D.gont-taps-address-usage-problem-statement]) discusses the trade-offs involved when considering different stability properties of IPv6 addresses.

7. IANA Considerations

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

8. Security Considerations

This document clarifies the stability requirements for IPv6 addresses, and specifies requirements for the generation of temporary addresses.

The security and privacy properties of IPv6 addresses have been discussed in detail in [RFC7721] and [RFC7707].

9. Acknowledgements

The authors would like to thank (in alphabetical order) Brian Carpenter, Lorenzo Colitti, and David Plonka, for providing valuable feedback on earlier versions of this document.

10. References

10.1. Normative References

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

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <https://www.rfc-editor.org/info/rfc4086>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <https://www.rfc-editor.org/info/rfc4862>.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <https://www.rfc-editor.org/info/rfc4941>.

[RFC5453] Krishnan, S., "Reserved IPv6 Interface Identifiers", RFC 5453, DOI 10.17487/RFC5453, February 2009, <https://www.rfc-editor.org/info/rfc5453>.

[RFC7136] Carpenter, B. and S. Jiang, "Significance of IPv6 Interface Identifiers", RFC 7136, DOI 10.17487/RFC7136, February 2014, <https://www.rfc-editor.org/info/rfc7136>.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <https://www.rfc-editor.org/info/rfc7217>.

[RFC8064] Gont, F., Cooper, A., Thaler, D., and W. Liu, "Recommendation on Stable IPv6 Interface Identifiers", RFC 8064, DOI 10.17487/RFC8064, February 2017, <https://www.rfc-editor.org/info/rfc8064>.

10.2. Informative References

[FIPS-SHS] NIST, "Secure Hash Standard (SHS)", FIPS Publication 180-4, March 2012, <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>.

[I-D.gont-predictable-numeric-ids] Gont, F. and I. Arce, "Security and Privacy Implications of Numeric Identifiers Employed in Network Protocols", draft-gont-predictable-numeric-ids-02 (work in progress), February 2018.

[I-D.gont-taps-address-usage-problem-statement] Gont, F., Gont, G., Corbo, M., and C. Huitema, "Problem Statement Regarding IPv6 Address Usage", draft-gont-taps-address-usage-problem-statement-00 (work in progress), February 2018.

[IANA-RESERVED-IID] IANA, "Reserved IPv6 Interface Identifiers", <http://www.iana.org/assignments/ipv6-interface-ids>.

[IETFMACRandom] Zuniga, JC., "MAC Privacy", November 2014, <http://www.ietf.org/blog/2014/11/mac-privacy/>.

[OPEN-GROUP] The Open Group, "The Open Group Base Specifications Issue 7 / IEEE Std 1003.1-2008, 2016 Edition", Section 4.16 Seconds Since the Epoch, 2016, <http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/contents.html>.

[RAID2015] Ullrich, J. and E. Weippl, "Privacy is Not an Option: Attacking the IPv6 Privacy Extension", International Symposium on Recent Advances in Intrusion Detection (RAID), 2015, <https://www.sba-research.org/wp-content/uploads/publications/Ullrich2015Privacy.pdf>.

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, DOI 10.17487/RFC1321, April 1992, <https://www.rfc-editor.org/info/rfc1321>.

[RFC3041] Narten, T. and R. Draves, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 3041, DOI 10.17487/RFC3041, January 2001, <https://www.rfc-editor.org/info/rfc3041>.

[RFC6059] Krishnan, S. and G. Daley, "Simple Procedures for Detecting Network Attachment in IPv6", RFC 6059, DOI 10.17487/RFC6059, November 2010, <https://www.rfc-editor.org/info/rfc6059>.

[RFC6151] Turner, S. and L. Chen, "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms", RFC 6151, DOI 10.17487/RFC6151, March 2011, <https://www.rfc-editor.org/info/rfc6151>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/info/rfc6973>.

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <https://www.rfc-editor.org/info/rfc7258>.

[RFC7624] Barnes, R., Schneier, B., Jennings, C., Hardie, T., Trammell, B., Huitema, C., and D. Borkmann, "Confidentiality in the Face of Pervasive Surveillance: A Threat Model and Problem Statement", RFC 7624, DOI 10.17487/RFC7624, August 2015, <https://www.rfc-editor.org/info/rfc7624>.

[RFC7707] Gont, F. and T. Chown, "Network Reconnaissance in IPv6 Networks", RFC 7707, DOI 10.17487/RFC7707, March 2016, <https://www.rfc-editor.org/info/rfc7707>.

[RFC7721] Cooper, A., Gont, F., and D. Thaler, "Security and Privacy Considerations for IPv6 Address Generation Mechanisms", RFC 7721, DOI 10.17487/RFC7721, March 2016, <https://www.rfc-editor.org/info/rfc7721>.

[RFC7844] Huitema, C., Mrugalski, T., and S. Krishnan, "Anonymity Profiles for DHCP Clients", RFC 7844, DOI 10.17487/RFC7844, May 2016, <https://www.rfc-editor.org/info/rfc7844>.

[RTB25] Interactive Advertising Bureau (IAB), "Real Time Bidding (RTB) project, OpenRTB API Specification Version 2.5", December 2016, <http://www.iab.com/wp-content/uploads/2016/03/OpenRTB-API-Specification-Version-2-5-FINAL.pdf>.

Authors' Addresses

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

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

Christian Huitema
Private Octopus Inc.
Friday Harbor, WA 98250
U.S.A.

Email: huitema@huitema.net
URI: http://privateoctopus.com/

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

Email: suresh.krishnan@ericsson.com

Guillermo Gont
SI6 Networks
Evaristo Carriego 2644
Haedo, Provincia de Buenos Aires 1706
Argentina

Phone: +54 11 4650 8472
Email: ggont@si6networks.com
URI: https://www.si6networks.com

Madeleine Garcia Corbo
Servicios de Informacion del Transporte
Neptuno 358
Havana City 10400
Cuba

Email: madelen.garcia16@gmail.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 14, 2018

L. Han, Ed.
G. Li
B. Tu
X. Tan
F. Li
R. Li
Huawei Technologies
J. Tantsura

K. Smith
Vodafone

October 11, 2017

IPv6 in-band signaling for the support of transport with QoS
draft-han-6man-in-band-signaling-for-transport-qos-00

Abstract

This document proposes a method to support the IP transport service that could guarantee a certain level of service quality in bandwidth and latency. The new transport service is fine-grained and could apply to individual or aggregated TCP/UDP flow(s).

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 14, 2018.

Table of Contents

1. Introduction 3
1.1. IP and Transport Technologies 4
1.2. TCP Solution Analysis 4
1.2.1. TCP Overview and Evolution 4
1.2.2. TCP Solution Variants 5
1.2.3. Throughput Constraint 6
1.2.3.1. By Algorithm 6
1.2.3.2. By Fairness Principle 7
1.2.4. Latency Constraint 7
1.2.5. Summary of TCP Solution 7
1.3. Other Solution Analysis 8
1.4. New approach 8
1.4.1. IP Transport with quality of service 8
1.4.2. Design targets 9
1.4.3. Scope and assumption 9
2. Terminology 10
2.1. Definitions 10
3. Control plane 11
3.1. Sub-layer in IP for transport control 12
3.2. IP In-band signaling 13
3.3. Control mechanism 14
3.4. IPv6 Approach 15
3.4.1. Basic Control Scenarios for TCP 16
3.4.2. Details of In-band Signaling for TCP 17
3.5. Key Messages and Parameters in Control Protocol 20
3.5.1. Setup and Setup State Report messages 20
3.5.2. OAM 21
3.5.3. Forwarding State and Forwarding State Report messages 21
3.5.4. Flow Identifying Methods 21
3.5.5. Hop Number 23
3.5.6. Mapping Index, Size and Mapping Index List 23
3.5.7. QoS State and life of Time 23
3.5.8. Authentication 24
4. Data plane 24
4.1. Basic Capability 24
4.2. Forwarding State and Forwarding State Report 25
4.3. Flow Identification in Packet Forwarding 26
4.4. QoS Forwarding State Detection and Failure Handling . . . 26

5. Other Issues 27
5.1. User and Application driven 27
5.2. Traffic Management in Host 28
5.3. Non-shortest-path 28
5.4. Heterogeneous Network 29
5.5. Proxy Control 29
6. Message Format 29
6.1. Setup Msg 29
6.2. Bandwidth Msg 31
6.3. Burst Msg 31
6.4. Latency Msg 31
6.5. Authentication Msg 32
6.6. OAM Msg 32
6.7. Forwarding State Msg 32
6.8. Setup State Report Msg 33
6.9. Forward State Report Msg 34
7. IANA Considerations 34
8. Security Considerations 35
9. Acknowledgements 35
10. References 36
10.1. Normative References 36
10.2. Informative References 36
Authors' Addresses 39

1. Introduction

Recently, more and more new applications for Internet are emerging. These applications have a common part that is their required bandwidth is very high and/or latency is very low compared with traditional applications like most of web and video applications.

For example, AR or VR applications may need a couple of hundred Mbps bandwidth (throughput) and a low single digit ms latency. Moreover, the difference of mean bit rate and peak bit rate is huge due to the compression algorithm [I-D.han-iccrg-arvr-transport-problem].

Some future applications expect that network can provide a bounded latency service, such as tactile network [Tactile].

With the technology development in 5G and beyond, the wireless access network is also rising the demand for the Ultra-Reliable and Low-Latency Communications (URLLC), this also leads to the question if IP transport can provide such service in Evolved Packet Core (EPC) network. IP is becoming more and more important in EPC when the Multi-access Edge Computing (MEC) for 5G will require the cloud and data service moving closer to eNodeB.

Following sections will brief the current transport and QoS technologies, and analyze the limitations to support above new applications.

A new approach that could provide QoS for transport service will be proposed. The scope and criteria for the new technology will also be summarized.

1.1. IP and Transport Technologies

The traditional IP network can only provide the best-effort service. The transport layer (TCP/UDP) on top of IP are based on this fundamental architecture. The best-effort-only service has influenced the transport evolution for quite long time, and results in some widely accepted assumptions and solutions, such as:

1. The IP layer can only provide the basic P2P (point to point) or P2MP (point to multi-point) end-to-end connectivity in Internet, but the connectivity is not reliable and does not guarantee any quality of service to end-user or application, such as bandwidth, packet loss, latency etc. Due to this assumption, the transport layer or application must have its own control mechanism in congestion and flow to obtain the reliable and satisfactory service to cooperate with the under layer network quality.

2. The transport layer assumes that the IP layer can only process all IP flows equally in the hardware since the best effort service is actually an un-differentiated service. The process includes scheduling, queuing and forwarding. Thus, the transport layer must behave nicely and friendly to make sure all flows will only obtain its own faired share of resource, and no one could consume more and no one could be starved.

1.2. TCP Solution Analysis

As a most popular and widely used transport technology, TCP traffic is dominating in Internet from the born of Internet. It is important to analyze the TCP. This section will brief the TCP, its variation, and some key factors.

1.2.1. TCP Overview and Evolution

The major functionalities of TCP are flow control and congestion control.

The flow control is based on the sliding window algorithm. In each TCP segment, the receiver specifies in the receive window field the amount of additionally received data (in bytes) that it is willing to

buffer for the connection. The sending host can send only up to that amount of data before it must wait for an acknowledgment and window update from the receiving host.

The congestion control is algorithms to prevent the hosts and network device fall into congestion state while trying to achieve the maximum throughput. There are many algorithm variations developed so far.

All congestion control will use some congestion detection scheme to detect the congestion and adjust the rate of source to avoid the congestion.

No matter what congestion control algorithm is used, traditionally, all TCP solutions are pursuing three targets, high efficiency in bandwidth utilization, high fairness in bandwidth allocation, and fast convergence to the equilibrium state. [TCP_Targets]

Recently, with the growth of new TCP applications in data center, more and more solutions were proposed to solve bufferbloat, incast problems typically happened in data center. These solutions include DCTCP, PIE, CoDel, FQ-CoDel, etc. In addition to the three traditional targets mentioned above, these solutions have another target which is to minimize the latency.

1.2.2. TCP Solution Variants

There are many TCP variants and optimization solutions since TCP was introduced 40 years ago. We have collected major TCP variants including typical traditional solution and some new solutions proposed recently.

The traditional solutions:

These solutions are implemented on host only. They use different congestion detection and inference mechanism, either based on packet loss, RTT or both, to dynamically adjust the TCP window to do the congestion control, such as: TCP-reno [RFC2581], TCP-vegas [TCP-vegas], TCP-cubic [TCP-cubic], TCP-compound [I-D.sridharan-tcpm-ctcp], TIMELY [TIMELY], etc

The explicit rate solutions:

These solutions do not use the traditional black box mechanism executed at host to infer the TCP congestion status, instead, they rely on the rate calculation on routers to let host adjust accordingly. Both network devices and hosts must be changed. Typical solutions are: XCP [I-D.falk-xcp-spec], RCP [RCP]. Note, we put XCP and RCP as TCP here is referring to the scenario when XCP and RCP are used with TCP

The AQM solutions:
These solutions use AQM (Active Queue Management) techniques on routers to control the buffer size, thus control the congestion and minimize the latency indirectly. Both network devices and hosts must be changed. They include: DCTCP [I-D.ietf-tcpm-dctcp], PIE [I-D.ietf-aqm-pie], CoDel [I-D.ietf-aqm-codel], FQ-CoDel [I-D.ietf-aqm-fq-codel], etc.

The new concept solutions:
Unlike above categories, these solutions use completely new concepts and methods to either accurately calculate, or figure out the optimized rate and latency of TCP, such as: PERC [PERC], BBR [BBR], PCC [PCC], Fastpass [Fastpass], etc

1.2.3. Throughput Constraint

For the traditional TCP optimization solutions, the efficiency target is to obtain the high bandwidth utilization as much as possible to approach the link capacity. The link utilization is defined as the total throughput of all TCP flows on a network device to the network bandwidth for links.

For individual TCP flow, its actual throughput is not guaranteed at all. It depends on many factors, such as TCP algorithm used, the number of TCP flows sharing the same link, host CPU power, network device congestion status, delay in transmission, etc.

For traditional TCP, the real throughput for a flow is limited by three factors: The 1st one is the available maximum throughput at the physical layer, accounting for maximum theoretical bandwidth, network load, buffering configuration, maximum segment size, signal strength, etc; The another is related to congestion control algorithm; The 3rd is related to the TCP fairness principle. Below we will analyze the last two factors.

1.2.3.1. By Algorithm

No matter what algorithm is used, The TCP throughput is always related to some flow and network characteristics, such as the RTT (Round Trip Time) and PLR (packet loss ratio). For example, TCP-reno throughput is shown in the formula (3) in [Reno_throughput]; And TCP-cubic throughput is expressed in formula (21) in [Cubic_throughput].

This limit will prevent the link capacity to be utilized by all TCP flows. Each TCP flow may only get a few portion of the link bandwidth as the real throughput for application. Even there is one TCP flow in a link, the throughput for the TCP could be way below the link capacity for a network which RTT and PLR are high.

1.2.3.2. By Fairness Principle

TCP fairness is a de facto principle for all TCP solutions. By this rule, each router will process all TCP flows equally and fairly to allocate the required resource to all TCP flows. Different Fair Queuing algorithms were used, such as Packet based Round Robin, Core-Stateless Fair Queuing(CSFQ), WFQ, etc. The targets of all algorithms are to reach the so called max-min fairness [Fairness] of TCP in terms of bandwidth.

TCP fairness played an important and critical role in saving internet from collapse caused by congestions since TCP was introduced.

The analysis [RCP] on page 35 has given the formula of the fair share rate at bottleneck routers, the rate or throughput is capped for applications which required bandwidth are not satisfied under the rule of fairness.

1.2.4. Latency Constraint

TCP fairness will not process some TCP flows differently with others, or there is no TCP micro-flow handling.

As described above, for the traditional solutions and explicit rate solution, the latency is not considered as a target, thus no latency guarantee at all.

For AQM solutions and some new concept solutions which try to control the buffer bloat or flow latency, it can only provide the statistic bounded latency for all TCP flows. The latency is related to the queue size and other factors. And the real latency for specific flow(s) is not deterministic. It could be very small or pretty large due to the long tail effect if the flow is blocked by other slower TCP flows.

1.2.5. Summary of TCP Solution

The bandwidth and latency can hardly be satisfied simultaneously without micro flow handling and management. While trying to get higher bandwidth, it may lead to more queued packet in router and result in longer latency. While approaching shorter latency, it may cause the queue under run, and lead to the lower bandwidth.

As a summary, to support some special TCP applications that are very sensitive to bandwidth and/or latency, we need to handle those TCP flows differently with others, and the TCP fairness must be relaxed for these scenarios.

It must be noted that the fairness based transport service could satisfy most of the applications, and it is the most efficient and economical way for hardware implementation and the network bandwidth efficiency.

When providing some TCP flows with differentiated service, the traditional transport service must be able to coexist with the new service. The resource partitioning between different service is a operation and management job for service provider.

1.3. Other Solution Analysis

DiffServ
: DiffServ [DiffServ] or Differentiated services is a network architecture that specifies a simple, scalable and coarse-grained mechanism for classifying and managing network traffic and providing QoS on modern IP networks. DiffServ is designed to support the QoS of aggregated traffic and normally is deployed in Service Provider networks. End user application cannot directly use DiffServ.

IntServ
: IntServ [IntServ] or integrated services specifies more fine-grained QoS, which is often contrasted with DiffServ's coarse-grained control system. IntServ definitely can support the applications requiring special QoS guarantee if it is deployed in a network, supported by Host OS and integrated with application. However, IntServ works on a small-scale only. When you scale up the network, it is difficult to keep track of all of the reservations and session states. Thus, IntServ is not scalable. Another problem of IntServ is it is not application driven, tedious provisioning cross different network must be done earlier. The provisioning is slow and hard to maintain.

MPLS-TE
: MPLS-TE can provide aggregated QoS or fine-grained QoS service for different class of traffic. Similar to DiffServ, MPLS-TE is majorly used for service providers network. It requires extra protocol sets like LDP, MPLS-TE, etc to operate. It is not practical to extend MPLS-TE to end user's desktop.

1.4. New approach

1.4.1. IP Transport with quality of service

Semiconductor chip technology has advanced a lot for last decades, the widely used network process can not only forward the packet in line speed, but also support fast packet processing for other

features, such as Qos for DiffServ/MPLS, Access Control List (ACL), fire wall, Deep Packet Inspection (DIP), etc. To treat some TCP/IP flows differently with others and give them specified resource are feasible now by using network processor.

Network processor is also able to do the general process to handle the simple control message for traffic management, such as signaling for hardware programming, congestion state report, OAM, etc.

This document proposes a mechanism to provide the capability of IP network to support the transport layer with quality of service. The solution is based on the QoS implemented in network processor. the proposal of the document is composed of two parts:

1. Control plane, it explains a transport control sub-layer for IP, the details of control mechanism.

2. Data plane, the realization of QoS in data forwarding, QoS and error handling.

1.4.2. Design targets

The new transport service is expected to satisfy following criteria:

1. End user or application can directly use and control the new service

2. The new service can coexist with the current transport service and is backward compatible.

3. The service provider can manage the new service.

4. Performance and scalability targets of new service are practical for vendors to achieve.

5. The new service is transport agnostic. Both TCP, UDP and other transport protocols on top of IP can use it

1.4.3. Scope and assumption

The initial aim is to propose a solution for IPv6.

To limit the scope of the document and simplify the design and solution, the following constraints are given.

1. The transport with QoS is aimed to be supplementary to the regular transport service. At the current situation, It is targeted for the applications that are bandwidth and/or latency

sensitive. It is not intended to replace the TCP variants that have been proved to be efficient and successful for current applications.

2. The new service is limited within one administrative domain, even it does not exclude the possibilities to extend the mechanism for inter-domain scenarios. Thus, the security and other inter-domain requirements are not critical. The basic security is good enough, the inter-domain SLA, accounting and other issues are not discussed.

3. Due to high bandwidth requirement of new service for individual flow, the total number of the flows with the new service cannot be high for a port, or a system. From another point of view, the new service is targeted for the application that really needs it, the number of supported applications/users are under controlled and cannot be unlimited. So, the scalability requirement for the new service is limited.

4. The new service must coexist with the regular transport service in the same hardware, and backward compatible. Also, a transport flow can switch without the service interruption between the regular transport support and new service.

2. Terminology

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

2.1. Definitions

E2E
: End-to-end

EH
: IPv6 Extension Header or Extension Option

QoS
: Quality of Service

OAM
: Operation and Management

In-band Signaling
: In telecommunications, in-band signaling is the sending of control information within the same band or channel used for voice or video.

Out-of-band Signaling
: out-of-band signaling is that the control information sent over a different channel, or even over a separate network.

IP flow
: For non-IPSec, a IP flow is identified by the source, destination IP address, the protocol number, the source and destination port number.

IP path
: A IP path is the route that IP flow will traverse. It could be the shortest path determined by routing protocols (IGP or BPG), or the explicit path decided by another management entity, such as a central controller, or Path Computation Element (PCE) Communication Protocol (PCEP), etc

QoS channel
: A forwarding channel that the QoS is guaranteed, it provides an additional QoS service to the normal IP forwarding. A QoS channel can be used for one or multiple IP flows depends on the granularity of in-band signaling.

Cir
: Committed Information Rate, this is the guaranteed bandwidth

Pir
: Peak Information Rate. this is the up limit bandwidth. Whether a flow can reach the PIR depends on the implementation. To use resource more efficiently, the system normally does not guarantee the PIR, but allow the sharing of resource between flows.

HbH-EH
: IPv6 Hop-by-Hop Extension Header

Dst-EH
: IPv6 Destination Extension Header

HbH-EH-aware node
: Network nodes that are configured to process the IPv6 Hop-by-Hop Extension Header

3. Control plane

3.1. Sub-layer in IP for transport control

In order to provide some new features for the upper layer above IP, it is very useful to introduce an additional sub-layer, Transport Control, between layer 3 (IP) and layer 4 (TCP/UDP). The new layer belongs to the IP, and is present only when the system needs to provide extra control for the upper layer, in addition to the normal IP forwarding. Fig 1. illustrates a new stack with the sub-layer.

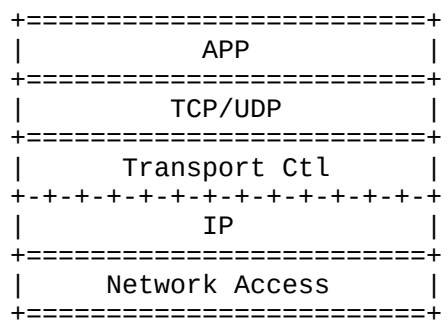


Figure 1: The new stack with a sub-layer in Layer 3

The new sub-layer is always bound with IP layer and can provide a support of the features for upper layer, such as:

In-band Signaling
: The IP header with the new sub-layer can carry the signaling information for the devices on the IP path. The information may include all QoS related parameters used for hardware programming.

Congestion control
: The congestion state in each device on the path can be detected and notified to the source of flows by the sub-layer; The dynamic congestion control instruction can also be carried by the sub-layer and examined by network devices on the IP path.

IP Path OAM
: The OAM instruction can be carried in the sub-layer, and the OAM state can be notified to the source of flows by the sub-layer. The OAM includes the path and device property detection, QoS forwarding diagnosis and report.

IPv6 can realize the sub-layer easily by the IPv6 extension header [RFC8200].

IPv4 could use the IP option for the purpose of the sub-layer. But due to the limit size of the IP option, the functionalities, scalability of the layer is restricted.

The document will focus on the solution for IPv6 by using different IPv6 extension header.

The control plane of the propose comprises of IP in-band signaling, and the detailed control mechanisms.

3.2. IP In-band signaling

There is no definition for IP in-band signaling. From the point of view of similarity to traditional telecommunication technology, the In-band signaling for IP is that the IP control messages are sharing some common header information as the data packet.

In this document, we introduce three types of "in-band signaling" for different signaling granularity:

Flow level In-band Signaling
: The control message and data packet share the same flow identification. The flow identification could be 5 tuples for non IPSec IPv6 packet: the source, destination IP address, protocol number, source and destination port number, and also could be 3 tuples for IPSec IPv6 packet: the source, destination IP address and the flow label. For the flow level in-band signaling, the signaling is for the individual IP flow, and there is no aggregation at all.

Address level In-band Signaling
: The control message and data packet share the same source, destination IP address, but with different protocol number. This is the scenario that the signaling is for the aggregated flows which have the same source, destination address. i.e, All TCP/UDP flows between the same client and same server (only one address for client and one for server)

Transport level In-band Signaling
: The control message and data packet share the same source, destination IP address, protocol number, but with different source or destination port number (non-IPSec) or different flow label (IPSec). This is the situation that the signaling is for the aggregated TCP or UDP flows that started and terminated at the same IP addresses.

Using In-band signaling, the control message can be embedded into any data packet, this can bring up some advantages that other methods can hardly provide:

Diagnosis
: The in-band signaling message takes the same path, same hops, same processing at each hop as the data packet, this will make the diagnosis for both signaling and data path easier.

Simplicity
: The in-band signaling message is forwarded with the normal data packet, it does not need to run a separate protocol. This will dramatically reduce the complexity of the control.

Performance and scalability
: Due to the simplicity of in-band signaling for control, it is easier to provide a better performance and scalability for a new future.

Note, the requirement of IP in-band signaling was proposed before by John Harper [I-D.harper-inband-signalling-requirements]. And the in-band QoS signaling for IPv6 was simply discussed in [I-D.roberts-inband-qos-ipv6]. Unfortunately, both works did not continue.

This document not only gives detailed solution for in-band signaling, but also try to address issues raised for the previous proposal, such as security, scalability and performance. Finally, experiments with proprietary hardware and chips are given in a presentation.

3.3. Control mechanism

The in-band signaling must be cooperated with a control method to achieve the QoS control. There are two categories of control, one is the closed-loop control and another is the open-loop control.

1. Closed-loop control is that the in-band signaling is sent in one direction and the feedback will return in the reverse direction. For example, the closed-loop control can be achieved by inserting the signaling information into a data packet sent in one direction, and the feedback information is carried in the data packet in reverse direction. The transport service with bi-direction data flow can use this mechanism, such as TCP and point-to-point UDP. In closed-loop control, a signaling message in one direction is processed at each router on the path. When the signaling message reaches the destination, the signaling message is processed by the protocol stack in the host, and the report information is generated. The report information is then

embedded into the flow data packet in the reverse direction and return to the host of the signaling source.

2. Open-loop control is that the in-band signaling is sent periodically in one direction without any feedback. The transport service with uni-direction data flow can use this mechanism, such as multicast by UDP. The transport service with bi-directional data flow can also use this mechanism when the simplicity of the control is wanted, i.e. no control feedback needed.

For both closed-loop and open-loop control, the signaling message for one direction is for the QoS programming for the direction. For example, the TCP-SYN or TCP data packet from client to server can carry the in-band signaling message to program the QoS for the direction of client to service. TCP-SYNACK or TCP data packet from server to client can carry the in-band signaling message to program the QoS for the server to client direction

Due to the nature that symmetric IP path between any source and destination cannot be guaranteed, in closed-loop control, the feedback information may take the different path as the in-band signaling path. The in-band signaling must not depend on the feedback information to accomplish the signaling work, such as the programming of hardware. This is one of the difference between in-band signaling and RSVP protocol.

For this document, we will only discuss the detailed mechanism for closed-loop control for TCP.

3.4. IPv6 Approach

The IPv6 In-band signaling could be realized by using the IPv6 extension header.

There are two types of extension header used for the purpose of transport QoS control, one is the hop-by-hop EH (HbH-EH) and another is the destination EH (Dst-EH).

The HbH-EH may be examined and processed by the nodes that are explicitly configured to do so [RFC8200]. We call this nodes as HbH-EH-aware nodes in document below. It is used to carry the QoS requirement for dedicated flow(s) and then the information is intercepted by HbH-EH-aware nodes on the path to program hardware accordingly.

The destination EH will only be examined and processed by the destination device that is associated with the destination IPv6

address in the IPv6 header. This EH is used to send the QoS related report information directly to the source of the signaling at other end.

3.4.1. Basic Control Scenarios for TCP

The finest grained QoS for TCP is flow level, this document will only focus on the solution of the flow level in-band signaling and its data plane. Other two types, address level and transport level QoS for TCP are briefly discussed in section 5.3.

The feature of TCP with flow level QoS comprises following control scenarios:

1. Setup: The setup is combined with the TCP 3-hand shaking, or any two directional TCP packets. When used with TCP 3-hand shaking, the 1st signaling embedded into HbH-EH is sent with TCP-SYN. It will be processed at HbH-EH-aware nodes on the path from source to destination. The signaling message includes the QoS requirements, such as max/min bandwidth, burst size, the latency, and the setup state. The setup state message is updated at HbH-EH-aware nodes to include the QoS programming and provisioning result and the necessary hardware reference information for IP forwarding with QoS. The 2nd signaling message is the TCP-SYNACK from server side, it includes the setup report message encoded as the Dst-EH. The setup report message is from the 1st TCP-SYN which represents the setup results on all HbH-EH-aware nodes on the path. The setup can even be started after TCP is established whenever the QoS service is required.

2. Dynamic control: this scenario is for the situation that previous QoS programming must be refreshed, modified or re-programmed. Normally, the signaling message can be embedded into HbH-EH for any TCP data packet or TCP-ACK packet. There are couple cases that the dynamic control is needed.

   HW state refreshing
      The HW state for QoS programming is data driven (see Section 4.1 for details). Its state will be refreshed if there is a data packet received. If there is no data received for a pre-configured time, the HW programming will be erased and the resource will be released.

   HW programming modification
      The HW QoS parameters can be modified if a new in-band signaling message is received and the embedded parameters are different with the old one that was used to program the HW. Section 3.4.2 will explain more about this scenario.

HW programming repairing
The IP path may be changed due to rerouting, link or node failures. This may result in the HW QoS programming failure. To repair any QoS programming failure, the new in-band signaling message can be embedded into any data packet and sent to the destination. All hops on the new path will be reprogramed with the QoS parameters. Section 4.4 has more detailed discussion.

3. Congestion Control: For TCP protocol, if IP layer can provide a certain level of quality service guarantee, the congestion control algorithm will be impacted a lot. As for what is the new congestion control, it depends on the quality service implementation in hardware and the behavior of the application. This is simply discussed in section 5.2.

3.4.2. Details of In-band Signaling for TCP

This document introduces following type of message for in-band signaling and associated data forwarding, the detailed format of messages is expressed in Section 6,

- Setup: This is for the setup of QoS channel through the IP path.
- Bandwidth: This is the required bandwidth for the QoS channel. It has minimum (CIR) and maximum bandwidth (PIR).
- Latency: This is the required latency for the QoS channel, it is the bounded latency for each hop on the path. This is not the end to end latency.
- Burst: This is the required burst for the QoS channel, it is the maximum burst size.
- Authentication: This is the security message for a in-band signaling.
- OAM: This is the Operation and Management message for the QoS channel.
- Setup State Report: This is the state report of a setup message.
- Forwarding State: This is the forwarding state message used for data packet.
- Forwarding State Report: This is the forwarding state report of a QoS channel.

There are three scenarios of QoS signaling for TCP session setup with QoS

1. Upstream: This is for the direction of client to server. A application decides to open a TCP session with upstream QoS (for uploading), it will call TCP API to open a socket and connect to a server. The client host will form a TCP SYN packet with the HbH-EH in the IPv6 header. The EH includes Setup message and Bandwidth message, and optionally Latency, Burst, Authentication and OAM messages. The packet is forwarded at each hop. Each HbH-EH-aware nodes will process the signaling message to finish the following tasks before forwarding the packet to next hop:

   * Retrieve the QoS parameters to program the Hardware, it includes: FL, Time, Bandwidth, Latency, Burst

   * Update the field in the EH, it includes: Hop_number, Total_latency, and possibly Mapping Index List

   When the server receives the TCP SYN, the Host kernel will also check the HbH-EH while punting the TCP packet to the TCP stack for processing. If the HbH-EH is present and the Report bit is set, the Host kernel must form a new Setup State Report message, all fields in the message must be copied from the Setup message in the HbH-EH. When the TCP stack is sending the TCP-SYNACK to the client, the kernel must add the Setup State Report message as a Dst-EH in the IPv6 header. After this, the IPv6 packet is complete and can be sent to wire; When the client receives the TCP-SYNACK, the Host kernel will check the Dst-EH while punting the TCP packet to the TCP stack for processing. If the Dst-EH is present and the Setup State Report message is valid, the kernel must read the Setup State Report message. Depending on the setup state, the client will operate according to description in section 5.1

2. Downstream: This is for the direction of server to client. A application decides to open a TCP session with downstream QoS (for downloading), it will call TCP API to open a socket and connect to a server. The client host will form a TCP SYN packet with the Dst-EH in the IPv6 header. The EH includes Bandwidth message, and optionally Latency, Burst messages. The packet is forwarded at each hop. Each hop will not process the Dst-EH. When the server receives the TCP SYN, the Host kernel will check the Dst-EH while punting the TCP packet to the TCP stack for processing. If the Dst-EH is present, the Host kernel will retrieve the QoS requirement information from Bandwidth, Latency and Burst message, and check the QoS policy for the user. If the user is allowed to get the service with the expected QoS, the

server will form a Setup message similar to the case of client to server, and add it as the HbH-EH in the IPv6 header, and send the TCP-SYNACK to client. Each HbH-EH-aware nodes on the path from server to client will process the message similar to the case of client to server. After the client receives the TCP-SYNACK, The client will send the Setup State Report message to server as the Dst-EH in the TCP-ACK. Finally the server receives the TC-ACK and Setup State Report message, it can send the data to the established session according to the pre-negotiated QoS requirements.

3. Bi-direction: This is the case that the client wants to setup a session with bi-direction QoS guarantee. The detailed operations are actually a combination of Upstream and Downstream described above.

After a QoS channel is setup, the in-band signaling message can still be exchanged between two hosts, there are two scenarios for this.

1. Modify QoS on the fly: When the pre-set QoS parameters need to be adjusted, the application at source host can re-send a new in-band signaling message, the message can be embedded into any TCP packet as a IPv6 HbH-EH. The QoS modification should not impact the established TCP session and programmed QoS service. Thus, there is no service impcted during the QoS modification. Depending on the hardware performance, the signaling message can be sent with TCP packet with different data size. If the performance is high, the signaling message can be sent with any TCP packet; otherwise, the signaling message should be sent with small size TCP packet or zero-size TCP packet (such as TCP ACK). Modification of QoS on the fly is a very critical feature for the so called "Application adaptive QoS transport service". With this service, an application (or the proxy from a service provider) could setup an optimized CIR for different stage of application for the economical and efficient purpose. For example, in the transport of compressed video, the I-frame has big size and cannot be lost, but P-frame and B-frame both have smaller size and can tolerate some loss. There are much more P-frame and B-frame than I-frame in videos with smooth changes and variations in images [I-D.han-iccrg-arvr-transport-problem]. Based on this characteristics, application can request a relatively small CIR for the time of P-frame and P-frame, and request a big CIR for the time of I-frame.

2. Repairing of the QoS channel: This is the case the QoS channel was broken and need to be repaired, see section 4.4.

3.5. Key Messages and Parameters in Control Protocol

The detailed message format is described in the section 6, the detailed explanation of key messages and parameters are below:

3.5.1. Setup and Setup State Report messages

Setup is the message used for following purpose:

- Setup the QoS channel for a TCP when the TCP session is establishing.

- Dynamic Control of the QoS channel for a established TCP session. See section 3.4.1

Setup message is intended to program the hardware for QoS channel on the IP path from the source to the destination expressed in IPv6 header. It is embedded as the HbH-EH in an appropriate TCP packet and will be processed at each HbH-EH-aware node. For the simplicity, performance and scalability purpose, we can configure some hop to do the processing and some hops do not. For different QoS requirement and scenarios, different criteria can be used for the configuration of the hop to be HbH-EH-aware node, below are some factor to consider:

- Reserved bandwidth is required: The throttle router is the critical point to be configured to process the hop-by-hop EH for the bandwidth reservation. The throttle router is the device that a interested TCP session cannot get the enough bandwidth to support its application. The regular throttle routers include the BRAS (broadband remote access server) in broadband access network, the PGW (PDN Gateway) in LTE network, the TOR (Top of Rack) in data center. In more general case, any routers which aggregate traffic may become as a throttle router. Moreover, the direction of congestion must be considered. Normally, the congestion happens on the direction that more than one flows from multiple ingress links are aggregated and sent to one egress link. For other devices that the interested TCP session can get the enough bandwidth do not need to process the hop-by-hop EH.

- Bounded latency is required: In theory, each router and switch could contribute some delay to the end-to-end latency, but the throttle router will contribute more than non-throttle routers, and slow device will contribute more than fast device. We can use OAM to detect the latency contribution in a network, and configure those worst-cast devices to process the HbH-EH.

Setup State Report message is the message sent from the destination host to the source host (from the point of view of the Setup message). The message is embedded into the Dst-EH in any data packet. The Setup State Report in the message is just a copy from the Setup message received at the destination host for a typical TCP session. The message is used at the source host to forward the packet later and to do the congestion control.

3.5.2. OAM

OAM is a special in-band signaling message used for detection and diagnosis. It can be used before and after a QoS channel is established. Before a QoS channel is established, OAM message can be added as a HbH-EH to any IPv6 packet and used to detect:

- IP path properties: Total hop number that is HbH-EH-aware node; The IP address of each HbH-EH-aware node.

- Static properties at each HbH-EH-aware node: Protocol version; Supported Flow identifying methods; Mapping index size; Supported configuration range of bandwidth, latency, forwarding QoS state time.

- Financial properties at each HbH-EH-aware node: Unit price for bandwidth; Unit price for service duration; Price for different latency.

After a QoS channel is established, OAM message can also be added as a HbH-EH to any IPv6 packet and used to detect and diagnose failures:

- IP path dynamic properties: Total end to end latency

- Dynamic properties at ach HbH-EH-aware node: Queue size; Remained bandwidth; Dropped packet number by different reasons.

- The detailed QoS forwarding failure reason.

3.5.3. Forwarding State and Forwarding State Report messages

Forwarding State and Forwarding State Report messages are used for data plane, See section 4.2.

3.5.4. Flow Identifying Methods

This is a parameter to program the HW for the flow identifying method. It is used for the QoS granularity definition and flow identification for QoS process. The QoS is enforced for a group of flows or a dedicated flow that can be identified by the same flow

identification. The QoS granularity is determined by the flow identification method during the setup and packet forwarding process. There are three levels of QoS granularities: Flow level, Address level and transport level. Each level of QoS granularity is realized by corresponding in-band signaling. The document focus on the flow level in-band signaling, other two level in-band signaling are discussed in the section 5.3.

There are two ways for the flow identifying method. One is by the tuples in IP header, another is by a local significant number (see mapping index) generated and maintained in a router. When "Mapping Index Size" (Mis) is zero, it means the "Flow identification method" (FI) is used for both control plane and data plane. When "Mis" is not zero, it means "FI" is only used in signaling, and the data plane will only use the "Mapping Index".

There are four types for "Flow identification method":

1. Individual Flow: Non-IPSec case: flow is identified by source and destination address, source and destination port number, and protocol number; IPSec case: flow is identified by source and destination address, flow label. For both case, FI = 0; the associated QoS is flow level, and QoS is guaranteed for a dedicated IP flow.

2. TCP flows: flow is identified by source and destination address, and TCP protocol number. The associated QoS is transport level, and QoS is guaranteed for TCP flows that have the same source and destination address. For this case, FI=1.

3. UDP flows: flow is identified by source and destination address, and UDP protocol number. The associated QoS is transport level, and QoS is guaranteed for UDP flows that have the same source and destination address. For this case, FI=2.

4. All flows: flow is identified by source and destination address. The associated QoS is address level, and QoS is guaranteed for all IP flows that have the same source and destination address. For this case, FI=3

The use of local generated number to identify flow is to speed up the flow lookup and QoS process for data plane. The number could be the MPLS label or a local tag for a MPLS capable router. The difference between this method and the MPLS switch is that there is no MPLS LDP protocol running and the IP packet does not need to be encapsulated as MPLS packet at the source host. When the MPLS label is used, the "Mapping Index Size" is 20 bits.

3.5.5. Hop Number

This is a parameter for the total number of hop that is HbH-EH-aware node on the path. it is the field "Hop_num" in Setup message. It is used to locate the bit position for "Setup State" and the "Mapping Index" in "Mapping Index List". The value of "Hop_num" must be decremented at each hop. And at the receive host of the in-band signaling, the Hop_num must be zero.

The source host must know the exact hop number, and setup the initial value in the Setup message. The exact hop number can be detected by the OAM message.

3.5.6. Mapping Index, Size and Mapping Index List

Mapping Index is the local significant number generated and maintained in a router, and The "Mapping Index List" is just a list of "Mapping Index" for all hops that are HbH-EH-aware nodes on the IP path.

Mapping Index Size is the size for each mapping index in the Mapping Index List. The source host must know Mapping Index Size, and setup the initial value in the Setup message. The exact Mapping Index Size can be detected by the OAM message.

When a router receives a HbH-EH, it may generate a mapping index for the flow(s) that is defined by the Flow Identifying Method in "FL". Then the router must attach the mapping index value to the end of the Mapping Index List. After the packet reaches the destination host, the Mapping Index List will be that the 1st router’s mapping index as the list header, and the last router’s mapping index as the list tail.

3.5.7. QoS State and life of Time

After the chip is programmed for a QoS, a QoS state is created. The QoS state life is determined by the "Time" in the Setup message. Whenever there is a packet processed by a QoS state, the associated timer for the QoS state is reset. If the timer of a QoS state is expired, the QoS state will be erased and the associated resource will be released.

In order to keep the QoS state active, a application at source host can send some zero size of data to refresh the QoS state.

When the Time is set to zero, it means the life of the QoS State will be kept until the de-programming message is received.

3.5.8. Authentication

The in-band signaling is designed to have a basic security mechanism to protect the integrity of a signaling message. The Authentication message is to attach to a signaling message, the source host calculates the harsh value of a key and all invariable part of a signaling message (Setup message: ver, FI, R, Mis, P, Time; Bandwidth message, Latency message, Burst message). The key is only known to the hosts and all HbH-EH-aware nodes. The securely distribution of the key is out the scope of the document

4. Data plane

To support the QoS feature, there are couple of important requirements and schemes for implementations. These include the basic capability for the hardware, the scheme for the data forwarding, QoS processing, state report, etc.

Section 4.1 will talk about the basic capability for data plane, and section 4.2 will discuss the messages used for data plane after the QoS channel is established.

4.1. Basic Capability

The document only proposes the protocol used for control, and it is independent of the implementation of the system. However, to achieve the satisfactory targets for performance and scalability, the protocol must be cooperated with capable hardware to provide the desired fine-grained QoS for different transport.

In our experiment to implement the feature for TCP, we used a network processor with traffic management feature. The traffic management can provide the fine-grained QoS for any configured flow(s). Following capabilities are RECOMMENDED:

1. The in-banding signaling is processed in network processor without punting to controller CPU for help

2. The QoS forwarding state is kept and maintained in network processor without the involvement from controller CPU.

3. The QoS state has a life of a pre-configured time and will be automatically deleted if there is no data packet processed by that QoS state. The timer can be changed on the fly.

4. The QoS forwarding does not need to be done at the controller CPU, or so called slow path. It is at the same hardware as the normal IP forwarding. For any IP packet, the QoS forwarding is

executed first. Normal forwarding will be executed if there is no QoS state associated with the identification of the flow.

5. The QoS forwarding and normal forwarding can be switched on the fly.

4.2. Forwarding State and Forwarding State Report

After the QoS is programmed by the in-band signaling, the specified IP flows can be processed and forwarded for the QoS requirement. There are two ways for host to use the QoS channel for associated TCP session:

1. Host directly send the IP packet without any changes to the packet, this is for the following cases:

   * The hardware was programmed to use the tuples in IP header as identification for QoS process (Mis = 0), and

   * The packet does not function to collect the QoS forwarding state on the path.

2. Host add the Forward State message into a data packet’s IP header as HbH-EH and send the packet, this is for the cases:

   * The hardware was programmed to use the mapping index as identification for QoS process (Mis != 0).

   * The hardware was programmed to use the tuples in IP header as identification for QoS process (Mis = 0), and the data packet functions to collect the QoS forwarding state on the path. This is the situation that host wants to detect the QoS forwarding state for the purpose of failure handling (See section 4.3).

Forwarding State message format is shown in the Section 6.7. It is used to notify the mapping index and also update QoS forwarding state for the hops that are HbH-EH-aware nodes.

After Forwarding State message is reaching the destination host, the host is supposed to retrieve it and form a Forwarding State Report message, and carry it in any data packet as the Dst-EH, then send to the host in the reverse direction.

4.3. Flow Identification in Packet Forwarding

Flow identification in Packet Forwarding is same as the QoS channel establishment by Setup message. It is to forward a packet with a specified QoS process if the packet is identified to be belonging to specified flow(s).

There are two method used in data forwarding to identify flows:

1. Hardware was programmed to use tuples in IP header implicitly. This is indicated by that the "Mis" is zero or the Mapping index is not used. When a packet is received, its tuples are looked up according to the value of "FI". If there is a QoS table has match for the packet, the packet will be processed by the QoS state found in the QoS table. This method does not need any EH added into the data packet unless the data packet function to collect the QoS forwarding state on the path. See section 4.3

2. Hardware was programmed to use mapping index to identify flows. This is indicated by that the "Mis" is not zero. When a packet is received, the mapping index associated with the hop is retrieved and looked up for the QoS table. If it has match for the packet, the packet will be processed by the QoS state entry found in the QoS table.

4.4. QoS Forwarding State Detection and Failure Handling

QoS forwarding may be failed due to different reasons:

1. Hardware failure in HbH-EH-aware node.

2. IP path change due to link failure, node failure or routing changes; And the IP path change has impact to the HbH-EH-aware node.

3. Network topology change; and the change leads to the changes of HbH-EH-aware nodes.

Application may need to be aware of the service status of QoS guarantee when the application is using a TCP session with QoS. In order to provide such feature, the TCP stack in the source host can detect the QoS forwarding state by sending TCP data packet with Forwarding State message coded as HbH-EH. After the TCP data packet reaches the destination host, the host will copy the forwarding state into a Forwarding State Report message, and send it with another TCP packet (for example, TCP-ACK) in reverse direction to the source host. Thereafter, the source host can obtain the QoS forwarding state on all HbH-EH-aware nodes.

A host can do the QoS forwarding state detection by three ways: on demand, periodically or constantly.

After a host detects that there is QoS forwarding state failure, it can repair such failure by sending another Setup message embedded into a HbH-EH of any TCP packet. This repairing can handle all failure case mentioned above.

If a failure cannot be repaired, host will be notified, and appropriate action can be taken, see section 5.1

5. Other Issues

Above document only covers the details for the QoS support of individual TCP session by using the flow level in-band signaling. Due to the extensive scope of in-band signaling, there are many other associated issues for IP transport control. Below lists some of them, and we only brief the solution but do not go to details.

The details of each topic can be expressed in other drafts.

5.1. User and Application driven

The QoS transport service is initiated and controlled by end user’s application. Following tasks are done in host

1. The detailed QoS parameters in in-band signaling is set by end user application. New socket option must be added, the option is a place holder for QoS parameters (Setup, Bandwidth, etc), Setup State Report and Forwarding State Report messages.

2. The Setup State Report and Forwarding State Report message received at host are processed by transport service in kernel. The Setup State Report message processed at host can result in the notification to the application whether the setup is successful. If the setup is successful, the application can start to use the socket having the QoS support; If the setup is failed, the application may have three choices:

    * Lower the QoS requirement and re-setup a new QoS channel with new in-band signaling message.

    * Use the TCP session as traditional transport without any QoS support.

    * Lookup the service provider for help to locate the problem in network.

5.2. Traffic Management in Host

In order to accommodate in-band signaling and the QoS transport service, the OS on a host muts be changed in traffic management related areas. There are two parts for traffic management to be changed, One is to manage traffic going out a host's shared links. Another is congestion control for TCP flows:

1. The current traffic management in a host manages traffic from different TCP/UDP session going out host link(s), in the way similar to routers to send traffic out. All TCP/UDP sessions will share the bandwidth for all egress links. For the purpose to work with the differentiated service provided by under layer network in bandwidth and latency, the kernel may allocate expected resource to applications that are using the QoS transport service. For example, kernel can queue different packets from different applications or users to different queue and schedule them in different priority. Only after this change, some application can use more bandwidth and get less queuing delay for a link than others.

2. The congestion control in a host manages the behavior of TCP flow(s). This includes important features like slow start, AIMD, fast retransmit, selective ACK, etc. To accommodate the benefit of the QoS guaranteed transport service, the congestion control will be much simpler. The new congestion control is related to the implementation of QoS guarantee. Following is a simple congestion control algorithm assuming that the CIR is guaranteed and PIR is shared between flows:

    * There is no slow start, the TCP can start the traffic at the rate of CIR.

    * The AIMD is kept, but the range of the sawtooth pattern should be maintained between CIR and PIR.

    * Other congestion control features can be kept.

5.3. Non-shortest-path

The above method for the transport service with QoS is for the normal IP flows passing along the shortest path determined by the IGP or BGP. However, the IP shortest path may not be the best path in terms of the QoS. For example, the original IP path may not have enough bandwidth for a transport QoS service. The latency of the IP path is not the minimum in the network. There are two problems involved. One is how to find the best path for a QoS criteria, bandwidth or

latency. Another is how to setup the transport QoS for a non-shortest-path.

The 1st problem is out of scope of this document and many technologies have be discovered or in research.

The 2nd problem can be solved by combining the segment routing and in-band signaling. The use of the HbH-EH and Dst-EH is independent of the type of IP path, thus can be used with segment routing for any path determined by source. Note, the HbH-EH-aware nodes may not be different as the explicit IPv6 address in the segment routing header.

5.4. Heterogeneous Network

When IP network is crossing a non-IP network, such as MPLS or Ethernet network, the in-band signaling needs to be interworking with that network. The behavior, protocol and rules in the interworking with non-IP network is not the problem this document will address. More study and research need to be done, and new draft should be written to solve the problem.

5.5. Proxy Control

It is expected that for a real service provider network, the in-band signaling will be checked, filtered and managed at a proxy routers. This will serve following purpose:

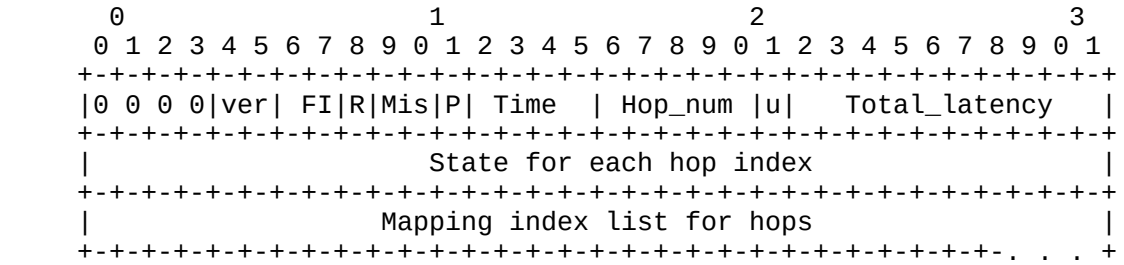
1. Proxy can check if a in-band signaling from end user for the SLA compliance, security and DOS attack prevention.

2. Proxy can collect the statistics for user’s TCP flows and check the in-band signaling for accounting and charging.

3. Proxy can insert and process appropriate in-band signaling for TCP flows that the host does not support the new feature, and this can provide the backward compatibility for host to use the new feature.

6. Message Format

6.1. Setup Msg



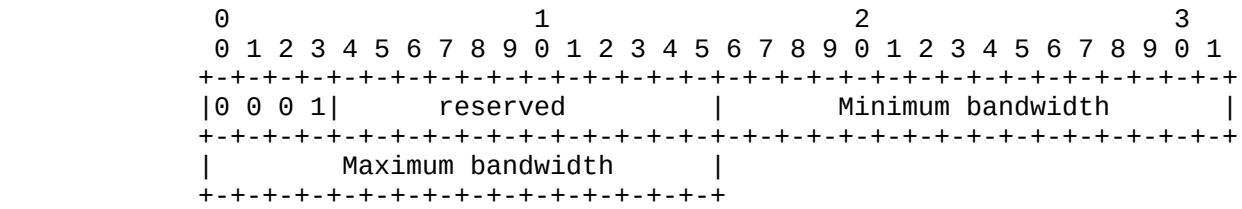
Type = 0, Setup state;
Version: The version of the protocol for the QoS
FI: Flow identification method,
0: 5 tuples; 1: src,dst,TCP; 2: src,dst,UDP; 3: src,dst
R: If the destination host report the received Setup state to the src address by Destination EH. 0: dont report; 1: report
Mis: Mapping index size; 0: 0bits, 1: 16bits, 2: 20bits, 3: 32bits
P: Programming the HW for QoS; 0: program HW for the QoS from src to dst; 1: De-program HW for the QoS from src to dst
Time: The life time of QoS forwarding state in second.
Hop_num: The total hop number on the path set by host. It must be decremented at each hop after the processing.
u: the unit of latency, 0: ms; 1: us
Total_latency : Latency accumulated from each hop, each hop will add the latency in the device to this value.
Setup state for each hop index: each bit is the setup state on each hop on the path, 0: failed; 1: success. The 1st hop is at the most significant bit.
Mapping index list for hops: the mapping index list for all hops on the path, each index bit size is defined in Mis. The 1st mapping index is at the top of the stack. Each hop add its mapping index at the correct position indexed by the current hop number for the router.

Figure 2: The Setup message

The Setup message is embedded into the hop-by-hop EH to setup the QoS in the device on the IP forwarding path. At each hop, if the router is configured to process the header and to enforce the QoS, it must retrieve the hardware required information from the header, and then update some fields in the hader.

To keep the whole setup message size unchanged at each hop, the total hop number must be known at the source host The total hop number can be detected by OAM. The mapping index list is empty before the 1st hop receives the in-band signaling. Each hop then fill up the associated mapping index into the correct place determined by the index of the hop.

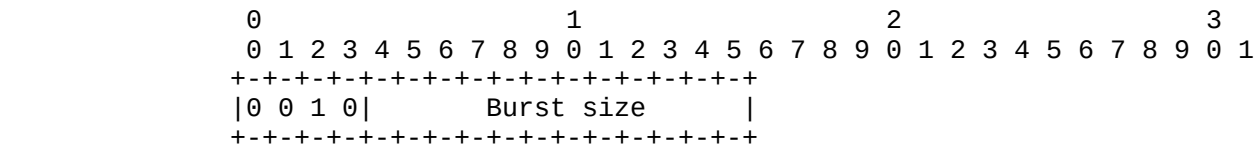
6.2. Bandwidth Msg



Type = 1,
Minimum bandwidth : The minimum bandwidth required, or CIR, unit Mbps
Maximum bandwidth : The maximum bandwidth required, or PIR, unit Mbps

Figure 3: The Bandwidth message

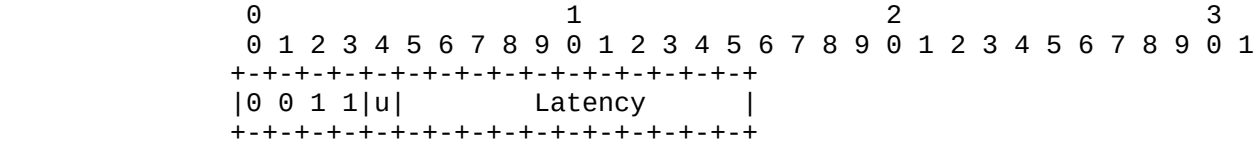
6.3. Burst Msg



Type = 2,
Burst size : The burst size, unit M bytes

Figure 4: The burst message

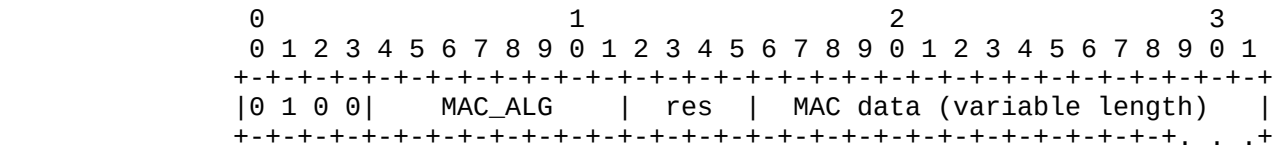
6.4. Latency Msg



Type = 3,
u: the unit of the latency
   0: ms; 1: us
Latency: Expected maximum latency for each hop

Figure 5: The Latency message

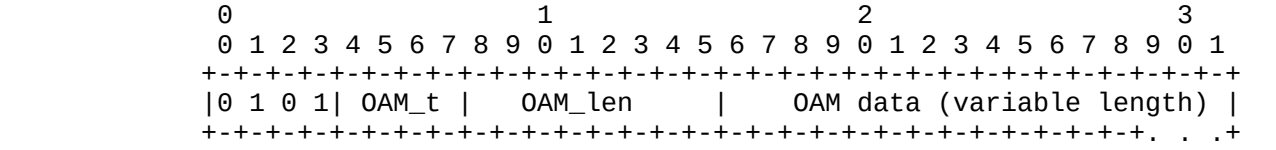
6.5. Authentication Msg



Type = 4,
MAC_ALG: Message Authentication Algorithm
0: MD5; 1:SHA-0; 2: SHA-1; 3: SHA-256; 4: SHA-512
MAC data: Message Authentication Data;
Res: Reserved bits
Size of signaling data (opt_len): Size of MAC data + 2
MD5: 18; SHA-0: 22; SHA-1: 22; SHA-256: 34; SHA-512: 66

Figure 6: The Authentication message

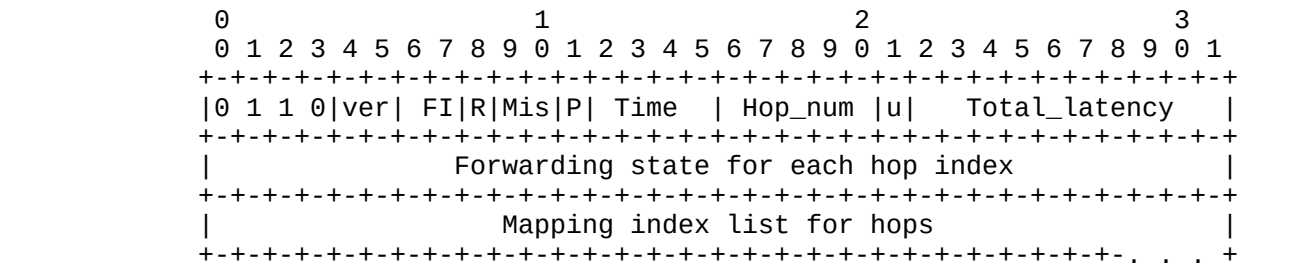
6.6. OAM Msg



Type = 5,
OAM_t : OAM type
OAM_len : 8-bit unsigned integer. Length of the OAM data, in octets;
OAM data: OAM data, details of OAM data are TBD.

Figure 7: The OAM message

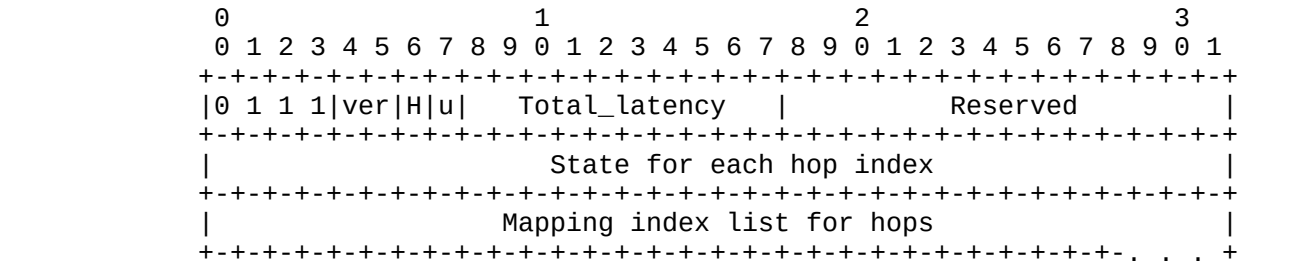
6.7. Forwarding State Msg



Type = 6, Forwarding state;
All parameter definitions and process in the 1st row are same in the setup message.
Forward state for each hop index : each bit is the fwd state on each hop on the path, 0: failed; 1: success; The 1st hop is at the most significant bit.
Mapping index list for hops: the mapping index list for all hops on the path, each index bit size is defined in Mis. The list is from the setup report message.

Figure 8: The Forwarding State message

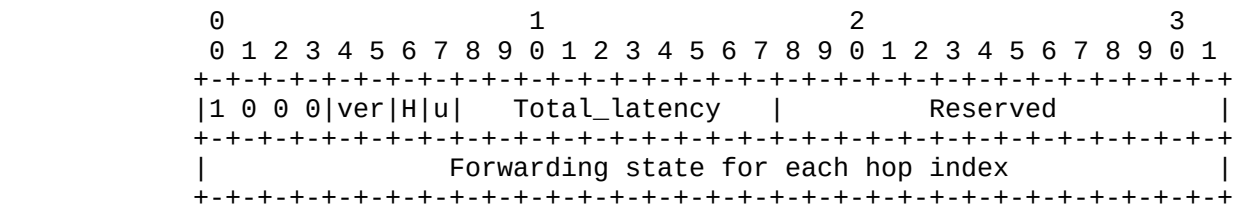
6.8. Setup State Report Msg



Type = 7, Setup state report;
H: Hop number bit. When a host receives a setup message and form a setup report message, it must check if the Hop_num in setup message is zero. If it is zero, the H bit is set to one, and if it is not zero, the H bit is clear. This will notify the source of setup message that if the original Hop_num was correct.
Following are directly copied from the setup message:
u, Total_latency;
State for each hop index
Mapping index list for hops.

Figure 9: The Setup State Report message

6.9. Forward State Report Msg



Type = 8, Forwarding state report;
H: Hop number bit. When a host receives a Forward State message and form a Forward State Report message, it must check if the Hop_num in Forward State message is zero. If it is zero, the H bit is set to one, and if it is not zero, the H bit is clear.
This will notify the source of Forward State message that if the original Hop_num was set correct.
Following are directly copied from the Forward State message:
u, Total_latency;
Forwarding State for each hop index

Figure 10: The Fwd State Report message

7. IANA Considerations

This document defines a new option type for the Hop-by-Hop Options header and the Destination Options header. According to [RFC8200], the detailed value are:

Hex Value	Binary Value			Description	Reference
	act	chg	rest		
0x0	00	0	10000	In-band Signaling	Section 6 in this doc

Figure 11: The New Option Type

1. The highest-order 2 bits: 00, indicating if the processing IPv6 node does not recognize the Option type, skip over this option and continue processing the header.

2. The third-highest-order bit: 0, indicating the Option Data does not change en route.

3. The low-order 5 bits: 10000, assigned by IANA.

This document also defines a 4-bit subtype field, for which IANA will create and will maintain a new sub-registry entitled "In-band signaling Subtypes" under the "Internet Protocol Version 6 (IPv6) Parameters" [IPv6_Parameters] registry. Initial values for the subtype registry are given below

Type	Mnemonic	Description	Reference
0	SETUP	Setup message	Section 6.1
1	BANDWIDTH	Bandwidth message	Section 6.2
2	BURST	Burst message	Section 6.3
3	LATENCY	Latency message	Section 6.4
4	AUTH	Authentication message	Section 6.5
5	OAM	OAM message	Section 6.6
6	FWD STATE	Forward state	Section 6.7
7	SETUP REPORT	Setup state report	Section 6.8
8	FWD REPORT	Forwarding state report	Section 6.9

Figure 12: The In-band Signaling Sub Type

8. Security Considerations

There is no security issue introduced by this document

9. Acknowledgements

We like to thank Huawei’s Nanjing research team leaded by Feng Li to provide the Product on Concept (POC) development and test, the team member includes Fengxin Sun, Xingwang Zhou, Weiguang Wang. We also like to thank other people involved in the discussion of solution: Tao Ma from Future Network Streategy dept.

10. References

10.1. Normative References

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

[RFC2581] Allman, M., Paxson, V., and W. Stevens, "TCP Congestion Control", RFC 2581, DOI 10.17487/RFC2581, April 1999, <https://www.rfc-editor.org/info/rfc2581>.

[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>.

10.2. Informative References

[BBR] Neal Cardwell, et al, Google, "BBR Congestion Control", 2016, <https://www.ietf.org/proceedings/97/slides/slides-97-iccrg-bbr-congestion-control-02.pdf>.

[Cubic_throughput] Wei Bao, et al. The University of British Columbia, Vancouver, Canada, IEEE Globecom 2010 proceedings, "A Model for Steady State Throughput of TCP CUBIC", 2010, <https://www.researchgate.net/publication/224211021_A_Model_for_Steady_State_Throughput_of_TCP_CUBIC>.

[DiffServ] wiki, "Differentiated services", 2016, <https://en.wikipedia.org/wiki/Differentiated_services>.

[Fairness] Jain, R., et al. DEC Research Report TR-301, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems", 1984, <http://www1.cse.wustl.edu/~jain/papers/ftp/fairness.pdf>.

[Fastpass] Jonathan Perry, et al, MIT, "Fastpass: A Centralized ?Zero-Queue? Datacenter Network", 2014, <http://fastpass.mit.edu/Fastpass-SIGCOMM14-Perry.pdf>.

[I-D.falk-xcp-spec]
Falk, A., "Specification for the Explicit Control Protocol (XCP)", draft-falk-xcp-spec-03 (work in progress), July 2007.

[I-D.han-iccrg-arvr-transport-problem]
Han, L. and K. Smith, "Problem Statement: Transport Support for Augmented and Virtual Reality Applications", draft-han-iccrg-arvr-transport-problem-01 (work in progress), March 2017.

[I-D.harper-inband-signalling-requirements]
Harper, J., "Requirements for In-Band QoS Signalling", draft-harper-inband-signalling-requirements-00 (work in progress), January 2007.

[I-D.ietf-aqm-codel]
Nichols, K., Jacobson, V., McGregor, A., and J. Iyengar, "Controlled Delay Active Queue Management", draft-ietf-aqm-codel-06 (work in progress), December 2016.

[I-D.ietf-aqm-fq-codel]
Hoeiland-Joergensen, T., McKenney, P., dave.taht@gmail.com, d., Gettys, J., and E. Dumazet, "The FlowQueue-CoDel Packet Scheduler and Active Queue Management Algorithm", draft-ietf-aqm-fq-codel-06 (work in progress), March 2016.

[I-D.ietf-aqm-pie]
Pan, R., Natarajan, P., Baker, F., and G. White, "PIE: A Lightweight Control Scheme To Address the Bufferbloat Problem", draft-ietf-aqm-pie-10 (work in progress), September 2016.

[I-D.ietf-tcpm-dctcp]
Bensley, S., Eggert, L., Thaler, D., Balasubramanian, P., and G. Judd, "Datacenter TCP (DCTCP): TCP Congestion Control for Datacenters", draft-ietf-tcpm-dctcp-03 (work in progress), November 2016.

[I-D.roberts-inband-qos-ipv6]
Roberts, L. and J. Harford, "In-Band QoS Signaling for IPv6", draft-roberts-inband-qos-ipv6-00 (work in progress), July 2005.

[I-D.sridharan-tcpm-ctcp]
Sridharan, M., Tan, K., Bansal, D., and D. Thaler, "Compound TCP: A New TCP Congestion Control for High-Speed and Long Distance Networks", draft-sridharan-tcpm-ctcp-02 (work in progress), November 2008.

[IntServ] wiki, "Integrated services", 2016, <https://en.wikipedia.org/wiki/Integrated_services>.

[IPv6_Parameters]
IANA, "Internet Protocol Version 6 (IPv6) Parameters", 2015, <https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml#ipv6-parameters-2>.

[PCC] Mo Dong, et al, University of Illinois at Urbana-Champaign, Hebrew University of Jerusalem, "PCC: Re-architecting Congestion Control for Consistent High Performance", 2014, <https://arxiv.org/abs/1409.7092>.

[PERC] Lavanya Jose, et al, Stanford University, MIT, Microsoft, "High Speed Networks Need Proactive Congestion Control", 2016, <http://web.stanford.edu/~lavanyaj/papers/perc-hotnets15.pdf>.

[RCP] Nandita Dukkipati, Ph.D. Thesis, Department of Electrical Engineering, Stanford University, "Rate Control Protocol (RCP): Congestion control to make flows complete quickly", 2007, <http://yuba.stanford.edu/~nanditad/thesis-NanditaD.pdf>.

[Reno_throughput]
Matthew Mathis, et al, Pittsburgh Supercomputing Center, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm", 1997, <https://cseweb.ucsd.edu/classes/wi01/cse222/papers/mathis-tcpmodel-ccr97.pdf>.

[Tactile] JDavid Szabo, et al. Proceedings of European Wireless 2015; 21th European Wireless Conference, "Towards the Tactile Internet: Decreasing Communication Latency with Network Coding and Software Defined Networking", 2015, <http://fastpass.mit.edu/Fastpass-SIGCOMM14-Perry.pdf>.

[TCP-cubic]
Ha, S., Rhee, I., and L. Xu, "CUBIC: A New TCP-Friendly High-Speed TCP Variant", 2008.

[TCP-vegas]
Peterson, L., "TCP Vegas: New Techniques for Congestion Detection and Avoidance - CiteSeer page on the 1994 SIGCOMM paper", 1994.

[TCP_Targets]
Andreas Benthin, Stefan Mischke, University of Paderborn, "Bandwidth Allocation of TCP", 2004.

[TIMELY] Radhika Mittal, et al. Google, Inc., "TIMELY: RTT-based Congestion Control for the Datacenter", 2010, <http://conferences.sigcomm.org/sigcomm/2015/pdf/papers/p537.pdf>.

Authors' Addresses

Lin Han (editor)
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Phone: +10 408 330 4613
Email: lin.han@huawei.com

Guoping Li
Huawei Technologies
Beijing
China

Email: liguoping@huawei.com

Boyan Tu
Huawei Technologies
Beijing
China

Email: tuboyan@huawei.com

Xuefei Tan
Huawei Technologies
Beijing
China

Email: tanxuefei@huawei.com

Frank Li
Huawei Technologies
Nanjing
China

Email: frank.lifeng@huawei.com

Richard Li
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: renwei.li@huawei.com

Jeff Tantsura

Email: jefftant.ietf@gmail.com

Kevin Smith
Vodafone
UK

Email: Kevin.Smith@vodafone.com

Internet Engineering Task Force
Internet-Draft
Obsoletes: 6434 (if approved)
Intended status: Best Current Practice
Expires: January 17, 2019

T. Chown
Jisc
J. Loughney
Intel
T. Winters
UNH-IOL

July 16, 2018

IPv6 Node Requirements
draft-ietf-6man-rfc6434-bis-09

Abstract

This document defines requirements for IPv6 nodes. It is expected that IPv6 will be deployed in a wide range of devices and situations. Specifying the requirements for IPv6 nodes allows IPv6 to function well and interoperate in a large number of situations and deployments.

This document obsoletes RFC 6434, and in turn RFC 4294.

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 January 17, 2019.

Table of Contents

1. Introduction 3
1.1. Scope of This Document 4
1.2. Description of IPv6 Nodes 4
2. Requirements Language 5
3. Abbreviations Used in This Document 5
4. Sub-IP Layer 5
5. IP Layer 6
5.1. Internet Protocol Version 6 - RFC 8200 6
5.2. Support for IPv6 Extension Headers 7
5.3. Protecting a node from excessive EH options 8
5.4. Neighbor Discovery for IPv6 - RFC 4861 9
5.5. SEcure Neighbor Discovery (SEND) - RFC 3971 10
5.6. IPv6 Router Advertisement Flags Option - RFC 5175 11
5.7. Path MTU Discovery and Packet Size 11
5.7.1. Path MTU Discovery - RFC 8201 11
5.7.2. Minimum MTU considerations 12
5.8. ICMP for the Internet Protocol Version 6 (IPv6) - RFC 4443 12
5.9. Default Router Preferences and More-Specific Routes - RFC 4191 12
5.10. First-Hop Router Selection - RFC 8028 12
5.11. Multicast Listener Discovery (MLD) for IPv6 - RFC 3810 . 12
5.12. Explicit Congestion Notification (ECN) - RFC 3168 13
6. Addressing and Address Configuration 13
6.1. IP Version 6 Addressing Architecture - RFC 4291 13
6.2. Host Address Availability Recommendations 13
6.3. IPv6 Stateless Address Autoconfiguration - RFC 4862 . . . 14
6.4. Privacy Extensions for Address Configuration in IPv6 - RFC 4941 15
6.5. Stateful Address Autoconfiguration (DHCPv6) - RFC 3315 . 15
6.6. Default Address Selection for IPv6 - RFC 6724 16
7. DNS 16
8. Configuring Non-Address Information 16
8.1. DHCP for Other Configuration Information 16
8.2. Router Advertisements and Default Gateway 17
8.3. IPv6 Router Advertisement Options for DNS Configuration - RFC 8106 17
8.4. DHCP Options versus Router Advertisement Options for Host Configuration 17
9. Service Discovery Protocols 18

10. IPv4 Support and Transition 18
10.1. Transition Mechanisms 18
10.1.1. Basic Transition Mechanisms for IPv6 Hosts and Routers - RFC 4213 18
11. Application Support 18
11.1. Textual Representation of IPv6 Addresses - RFC 5952 . . 18
11.2. Application Programming Interfaces (APIs) 18
12. Mobility 19
13. Security 20
13.1. Requirements 21
13.2. Transforms and Algorithms 21
14. Router-Specific Functionality 21
14.1. IPv6 Router Alert Option - RFC 2711 22
14.2. Neighbor Discovery for IPv6 - RFC 4861 22
14.3. Stateful Address Autoconfiguration (DHCPv6) - RFC 3315 . 22
14.4. IPv6 Prefix Length Recommendation for Forwarding - BCP 198 23
15. Constrained Devices 23
16. IPv6 Node Management 23
16.1. Management Information Base (MIB) Modules 23
16.1.1. IP Forwarding Table MIB 24
16.1.2. Management Information Base for the Internet Protocol (IP) 24
16.1.3. Interface MIB 24
16.2. YANG Data Models 24
16.2.1. IP Management YANG Model 24
16.2.2. Interface Management YANG Model 24
17. Security Considerations 24
18. IANA Considerations 24
19. Authors and Acknowledgments 24
19.1. Authors and Acknowledgments (Current Document) 25
19.2. Authors and Acknowledgments from RFC 6434 25
19.3. Authors and Acknowledgments from RFC 4294 25
20. Appendix: Changes from RFC 6434 25
21. Appendix: Changes from RFC 4294 27
22. References 28
22.1. Normative References 28
22.2. Informative References 35
Authors' Addresses 40

1. Introduction

This document defines common functionality required by both IPv6 hosts and routers. Many IPv6 nodes will implement optional or additional features, but this document collects and summarizes requirements from other published Standards Track documents in one place.

This document tries to avoid discussion of protocol details and references RFCs for this purpose. This document is intended to be an applicability statement and to provide guidance as to which IPv6 specifications should be implemented in the general case and which specifications may be of interest to specific deployment scenarios. This document does not update any individual protocol document RFCs.

Although this document points to different specifications, it should be noted that in many cases, the granularity of a particular requirement will be smaller than a single specification, as many specifications define multiple, independent pieces, some of which may not be mandatory. In addition, most specifications define both client and server behavior in the same specification, while many implementations will be focused on only one of those roles.

This document defines a minimal level of requirement needed for a device to provide useful internet service and considers a broad range of device types and deployment scenarios. Because of the wide range of deployment scenarios, the minimal requirements specified in this document may not be sufficient for all deployment scenarios. It is perfectly reasonable (and indeed expected) for other profiles to define additional or stricter requirements appropriate for specific usage and deployment environments. For example, this document does not mandate that all clients support DHCP, but some deployment scenarios may deem it appropriate to make such a requirement. For example, NIST has defined profiles for specialized requirements for IPv6 in target environments (see [USGv6]).

As it is not always possible for an implementer to know the exact usage of IPv6 in a node, an overriding requirement for IPv6 nodes is that they should adhere to Jon Postel’s Robustness Principle: "Be conservative in what you do, be liberal in what you accept from others" [RFC0793].

1.1. Scope of This Document

IPv6 covers many specifications. It is intended that IPv6 will be deployed in many different situations and environments. Therefore, it is important to develop requirements for IPv6 nodes to ensure interoperability.

1.2. Description of IPv6 Nodes

From the Internet Protocol, Version 6 (IPv6) Specification [RFC8200], we have the following definitions:

IPv6 node - a device that implements IPv6.
IPv6 router - a node that forwards IPv6 packets not explicitly addressed to itself.
IPv6 host - any IPv6 node that is not a router.

2. Requirements Language

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

3. Abbreviations Used in This Document

AH	Authentication Header
DAD	Duplicate Address Detection
ESP	Encapsulating Security Payload
ICMP	Internet Control Message Protocol
IKE	Internet Key Exchange
MIB	Management Information Base
MLD	Multicast Listener Discovery
MTU	Maximum Transmission Unit
NA	Neighbor Advertisement
NBMA	Non-Broadcast Multiple Access
ND	Neighbor Discovery
NS	Neighbor Solicitation
NUD	Neighbor Unreachability Detection
PPP	Point-to-Point Protocol

4. Sub-IP Layer

An IPv6 node MUST include support for one or more IPv6 link-layer specifications. Which link-layer specifications an implementation should include will depend upon what link-layers are supported by the hardware available on the system. It is possible for a conformant IPv6 node to support IPv6 on some of its interfaces and not on others.

As IPv6 is run over new layer 2 technologies, it is expected that new specifications will be issued. In the following, we list some of the layer 2 technologies for which an IPv6 specification has been developed. It is provided for informational purposes only and may not be complete.

- Transmission of IPv6 Packets over Ethernet Networks [RFC2464]

- Transmission of IPv6 Packets over Frame Relay Networks Specification [RFC2590]

- Transmission of IPv6 Packets over IEEE 1394 Networks [RFC3146]

- Transmission of IPv6, IPv4, and Address Resolution Protocol (ARP) Packets over Fibre Channel [RFC4338]

- Transmission of IPv6 Packets over IEEE 802.15.4 Networks [RFC4944]

- Transmission of IPv6 via the IPv6 Convergence Sublayer over IEEE 802.16 Networks [RFC5121]

- IP version 6 over PPP [RFC5072]

In addition to traditional physical link-layers, it is also possible to tunnel IPv6 over other protocols. Examples include:

- Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs) [RFC4380]

- Section 3 of "Basic Transition Mechanisms for IPv6 Hosts and Routers" [RFC4213]

5. IP Layer

5.1. Internet Protocol Version 6 - RFC 8200

The Internet Protocol Version 6 is specified in [RFC8200]. This specification MUST be supported.

The node MUST follow the packet transmission rules in RFC 8200.

All conformant IPv6 implementations MUST be capable of sending and receiving IPv6 packets; forwarding functionality MAY be supported. Nodes MUST always be able to send, receive, and process fragment headers.

IPv6 nodes MUST not create overlapping fragments. Also, when reassembling an IPv6 datagram, if one or more of its constituent fragments is determined to be an overlapping fragment, the entire datagram (and any constituent fragments) MUST be silently discarded. See [RFC5722] for more information.

As recommended in [RFC8021], nodes MUST NOT generate atomic fragments, i.e., where the fragment is a whole datagram. As per [RFC6946], if a receiving node reassembling a datagram encounters an atomic fragment, it should be processed as a fully reassembled

packet, and any other fragments that match this packet should be processed independently.

To mitigate a variety of potential attacks, nodes SHOULD avoid using predictable fragment Identification values in Fragment Headers, as discussed in [RFC7739].

All nodes SHOULD support the setting and use of the IPv6 Flow Label field as defined in the IPv6 Flow Label specification [RFC6437]. Forwarding nodes such as routers and load distributors MUST NOT depend only on Flow Label values being uniformly distributed. It is RECOMMENDED that source hosts support the flow label by setting the Flow Label field for all packets of a given flow to the same value chosen from an approximation to a discrete uniform distribution.

5.2. Support for IPv6 Extension Headers

RFC 8200 specifies extension headers and the processing for these headers.

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.

Any unrecognized extension headers or options MUST be processed as described in RFC 8200. Note that where Section 4 of RFC 8200 refers to the action to be taken when a Next Header value in the current header is not recognized by a node, that action applies whether the value is an unrecognized Extension Header or an unrecognized upper layer protocol (ULP).

An IPv6 node MUST be able to process these extension headers. An exception is Routing Header type 0 (RH0), which was deprecated by [RFC5095] due to security concerns and which MUST be treated as an unrecognized routing type.

Further, [RFC7045] adds specific requirements for processing of Extension Headers, in particular that any forwarding node along an IPv6 packet's path, which forwards the packet for any reason, SHOULD do so regardless of any extension headers that are present.

As per RFC 8200, when a node fragments an IPv6 datagram, it MUST include the entire IPv6 Header Chain in the first fragment. The Per-Fragment headers MUST consist of the IPv6 header plus any extension headers that MUST be processed by nodes en route to the destination, that is, all headers up to and including the Routing header if

present, else the Hop-by-Hop Options header if present, else no extension headers. On reassembly, if the first fragment does not include all headers through an Upper-Layer header, then that fragment SHOULD be discarded and an ICMP Parameter Problem, Code 3, message SHOULD be sent to the source of the fragment, with the Pointer field set to zero. See [RFC7112] for a discussion of why oversized IPv6 Extension Header chains are avoided.

Defining new IPv6 extension headers is not recommended, unless there are no existing IPv6 extension headers that can be used by specifying a new option for that IPv6 extension header. A proposal to specify a new IPv6 extension header MUST include a detailed technical explanation of why an existing IPv6 extension header can not be used for the desired new function, and in such cases need to follow the format described in Section 8 of RFC 8200. For further background reading on this topic, see [RFC6564].

5.3. Protecting a node from excessive EH options

As per RFC 8200, end hosts are expected to process all extension headers, destination options, and hop-by-hop options in a packet. Given that the only limit on the number and size of extension headers is the MTU, the processing of received packets could be considerable. It is also conceivable that a long chain of extension headers might be used as a form of denial-of-service attack. Accordingly, a host may place limits on the number and sizes of extension headers and options it is willing to process.

A host MAY limit the number of consecutive PAD1 options in destination options or hop-by-hop options to seven. In this case, if the more than seven consecutive PAD1 options are present the packet MAY be silently discarded. The rationale is that if padding of eight or more bytes is required than the PADN option SHOULD be used.

A host MAY limit number of bytes in a PADN option to be less than eight. In such a case, if a PADN option is present that has a length greater than seven then the packet SHOULD be silently discarded. The rationale for this guideline is that the purpose of padding is for alignment and eight bytes is the maximum alignment used in IPv6.

A host MAY disallow unknown options in destination options or hop-by-hop options. This SHOULD be configurable where the default is to accept unknown options and process them per [RFC8200]. If a packet with unknown options is received and the host is configured to disallow them, then the packet SHOULD be silently discarded.

A host MAY impose 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 eight. 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 silently discarded.

A host MAY impose a limit on the maximum length of destination options or hop-by-hop options extension 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 destination or hop-by-hop options extension header exceeds the length limit, then the packet SHOULD be silently discarded.

5.4. Neighbor Discovery for IPv6 - RFC 4861

Neighbor Discovery is defined in [RFC4861]; the definition was updated by [RFC5942]. Neighbor Discovery SHOULD be supported. RFC 4861 states:

> Unless specified otherwise (in a document that covers operating IP over a particular link type) this document applies to all link types. However, because ND uses link-layer multicast for some of its services, it is possible that on some link types (e.g., Non-Broadcast Multi-Access (NBMA) links), alternative protocols or mechanisms to implement those services will be specified (in the appropriate document covering the operation of IP over a particular link type). The services described in this document that are not directly dependent on multicast, such as Redirects, next-hop determination, Neighbor Unreachability Detection, etc., are expected to be provided as specified in this document. The details of how one uses ND on NBMA links are addressed in [RFC2491].

Some detailed analysis of Neighbor Discovery follows:

Router Discovery is how hosts locate routers that reside on an attached link. Hosts MUST support Router Discovery functionality.

Prefix Discovery is how hosts discover the set of address prefixes that define which destinations are on-link for an attached link. Hosts MUST support Prefix Discovery.

Hosts MUST also implement Neighbor Unreachability Detection (NUD) for all paths between hosts and neighboring nodes. NUD is not required for paths between routers. However, all nodes MUST respond to unicast Neighbor Solicitation (NS) messages.

[RFC7048] discusses NUD, in particular cases where it behaves too impatiently. It states that if a node transmits more than a certain number of packets, then it SHOULD use the exponential backoff of the retransmit timer, up to a certain threshold point.

Hosts MUST support the sending of Router Solicitations and the receiving of Router Advertisements. The ability to understand individual Router Advertisement options is dependent on supporting the functionality making use of the particular option.

[RFC7559] discusses packet loss resiliency for Router Solicitations, and requires that nodes MUST use a specific exponential backoff algorithm for RS retransmissions.

All nodes MUST support the sending and receiving of Neighbor Solicitation (NS) and Neighbor Advertisement (NA) messages. NS and NA messages are required for Duplicate Address Detection (DAD).

Hosts SHOULD support the processing of Redirect functionality. Routers MUST support the sending of Redirects, though not necessarily for every individual packet (e.g., due to rate limiting). Redirects are only useful on networks supporting hosts. In core networks dominated by routers, Redirects are typically disabled. The sending of Redirects SHOULD be disabled by default on routers intended to deployed on core networks. They MAY be enabled by default on routers intended to support hosts on edge networks.

"IPv6 Host-to-Router Load Sharing" [RFC4311] includes additional recommendations on how to select from a set of available routers. [RFC4311] SHOULD be supported.

5.5. SEcure Neighbor Discovery (SEND) - RFC 3971

SEND [RFC3971] and Cryptographically Generated Addresses (CGAs) [RFC3972] provide a way to secure the message exchanges of Neighbor Discovery. SEND has the potential to address certain classes of spoofing attacks, but it does not provide specific protection for threats from off-link attackers.

There have been relatively few implementations of SEND in common operating systems and platforms since its publication in 2005, and thus deployment experience remains very limited to date.

At this time, support for SEND is considered optional. Due to the complexity in deploying SEND, and its heavyweight provisioning, its deployment is only likely to be considered where nodes are operating in a particularly strict security environment.

5.6. IPv6 Router Advertisement Flags Option - RFC 5175

Router Advertisements include an 8-bit field of single-bit Router Advertisement flags. The Router Advertisement Flags Option extends the number of available flag bits by 48 bits. At the time of this writing, 6 of the original 8 single-bit flags have been assigned, while 2 remain available for future assignment. No flags have been defined that make use of the new option, and thus, strictly speaking, there is no requirement to implement the option today. However, implementations that are able to pass unrecognized options to a higher-level entity that may be able to understand them (e.g., a user-level process using a "raw socket" facility) MAY take steps to handle the option in anticipation of a future usage.

5.7. Path MTU Discovery and Packet Size

5.7.1. Path MTU Discovery - RFC 8201

"Path MTU Discovery for IP version 6" [RFC8201] SHOULD be supported. From [RFC8200]:

> It is strongly recommended that IPv6 nodes implement Path MTU Discovery [RFC8201], in order to discover and take advantage of path MTUs greater than 1280 octets. However, a minimal IPv6 implementation (e.g., in a boot ROM) may simply restrict itself to sending packets no larger than 1280 octets, and omit implementation of Path MTU Discovery.

The rules in [RFC8200] and [RFC5722] MUST be followed for packet fragmentation and reassembly.

As described in RFC 8201, nodes implementing Path MTU Discovery and sending packets larger than the IPv6 minimum link MTU are susceptible to problematic connectivity if ICMPv6 messages are blocked or not transmitted. For example, this will result in connections that complete the TCP three-way handshake correctly but then hang when data is transferred. This state is referred to as a black-hole connection [RFC2923]. Path MTU Discovery relies on ICMPv6 Packet Too Big (PTB) to determine the MTU of the path (and thus these MUST not be filtered, as per the recommendation in [RFC4890]).

An alternative to Path MTU Discovery defined in RFC 8201 can be found in [RFC4821], which defines a method for Packetization Layer Path MTU Discovery (PLPMTUD) designed for use over paths where delivery of ICMPv6 messages to a host is not assured.

5.7.2. Minimum MTU considerations

While an IPv6 link MTU can be set to 1280 bytes, it is recommended that for IPv6 UDP in particular, which includes DNS operation, the sender use a large MTU if they can, in order to avoid gratuitous fragmentation-caused packet drops.

5.8. ICMP for the Internet Protocol Version 6 (IPv6) - RFC 4443

ICMPv6 [RFC4443] MUST be supported. "Extended ICMP to Support Multi-Part Messages" [RFC4884] MAY be supported.

5.9. Default Router Preferences and More-Specific Routes - RFC 4191

"Default Router Preferences and More-Specific Routes" [RFC4191] provides support for nodes attached to multiple (different) networks, each providing routers that advertise themselves as default routers via Router Advertisements. In some scenarios, one router may provide connectivity to destinations the other router does not, and choosing the "wrong" default router can result in reachability failures. In order to resolve this scenario IPv6 Nodes MUST implement [RFC4191] and SHOULD implement the Type C host role defined in RFC4191.

5.10. First-Hop Router Selection - RFC 8028

In multihomed scenarios, where a host has more than one prefix, each allocated by an upstream network that is assumed to implement BCP 38 ingress filtering, the host may have multiple routers to choose from.

Hosts that may be deployed in such multihomed environments SHOULD follow the guidance given in [RFC8028].

5.11. Multicast Listener Discovery (MLD) for IPv6 - RFC 3810

Nodes that need to join multicast groups MUST support MLDv2 [RFC3810]. MLD is needed by any node that is expected to receive and process multicast traffic and in particular MLDv2 is required for support for source-specific multicast (SSM) as per [RFC4607].

Previous versions of this document only required MLDv1 ([RFC2710]) to be implemented on all nodes. Since participation of any MLDv1-only nodes on a link require that all other nodeas on the link then operate in version 1 compatibility mode, the requirement to support MLDv2 on all nodes was upgraded to a MUST. Further, SSM is now the preferred multicast distribution method, rather than ASM.

Note that Neighbor Discovery (as used on most link types -- see Section 5.4) depends on multicast and requires that nodes join Solicited Node multicast addresses.

5.12. Explicit Congestion Notification (ECN) - RFC 3168

An ECN-aware router sets a mark in the IP header in order to signal impending congestion, rather than dropping a packet. The receiver of the packet echoes the congestion indication to the sender, which can then reduce its transmission rate as if it detected a dropped packet.

Nodes SHOULD support [RFC3168] by implementing an interface for the upper layer to access and set the ECN bits in the IP header. The benefits of using ECN are documented in [RFC8087].

6. Addressing and Address Configuration

6.1. IP Version 6 Addressing Architecture - RFC 4291

The IPv6 Addressing Architecture [RFC4291] MUST be supported.

The current IPv6 Address Architecture is based on a 64-bit boundary for subnet prefixes. The reasoning behind this decision is documented in [RFC7421].

Implementations MUST also support the Multicast flag updates documented in [RFC7371]

6.2. Host Address Availability Recommendations

Hosts may be configured with addresses through a variety of methods, including SLAAC, DHCPv6, or manual configuration.

[RFC7934] recommends that networks provide general-purpose end hosts with multiple global IPv6 addresses when they attach, and it describes the benefits of and the options for doing so. Routers SHOULD support [RFC7934] for assigning multiple address to a host. Host SHOULD support assigning multiple addresses as described in [RFC7934].

Nodes SHOULD support the capability to be assigned a prefix per host as documented in [RFC8273]. Such an approach can offer improved host isolation and enhanced subscriber management on shared network segments.

6.3. IPv6 Stateless Address Autoconfiguration - RFC 4862

Hosts MUST support IPv6 Stateless Address Autoconfiguration. It is RECOMMENDED, as described in [RFC8064], that unless there is a specific requirement for MAC addresses to be embedded in an IID, nodes follow the procedure in [RFC7217] to generate SLAAC-based addresses, rather than using [RFC4862]. Addresses generated through RFC7217 will be the same whenever a given device (re)appears on the same subnet (with a specific IPv6 prefix), but the IID will vary on each subnet visited.

Nodes that are routers MUST be able to generate link-local addresses as described in [RFC4862].

From RFC 4862:

> The autoconfiguration process specified in this document applies only to hosts and not routers. Since host autoconfiguration uses information advertised by routers, routers will need to be configured by some other means. However, it is expected that routers will generate link-local addresses using the mechanism described in this document. In addition, routers are expected to successfully pass the Duplicate Address Detection procedure described in this document on all addresses prior to assigning them to an interface.

All nodes MUST implement Duplicate Address Detection. Quoting from Section 5.4 of RFC 4862:

> Duplicate Address Detection MUST be performed on all unicast addresses prior to assigning them to an interface, regardless of whether they are obtained through stateless autoconfiguration, DHCPv6, or manual configuration, with the following [exceptions noted therein].

"Optimistic Duplicate Address Detection (DAD) for IPv6" [RFC4429] specifies a mechanism to reduce delays associated with generating addresses via Stateless Address Autoconfiguration [RFC4862]. RFC 4429 was developed in conjunction with Mobile IPv6 in order to reduce the time needed to acquire and configure addresses as devices quickly move from one network to another, and it is desirable to minimize transition delays. For general purpose devices, RFC 4429 remains optional at this time.

[RFC7527] discusses enhanced DAD, and describes an algorithm to automate the detection of looped back IPv6 ND messages used by DAD. Nodes SHOULD implement this behaviour where such detection is beneficial.

6.4. Privacy Extensions for Address Configuration in IPv6 - RFC 4941

A node using Stateless Address Autoconfiguration [RFC4862] to form a globally unique IPv6 address using its MAC address to generate the IID will see that IID remain the same on any visited network, even though the network prefix part changes. Thus it is possible for 3rd party device to track the activities of the node they communicate with, as that node moves around the network. Privacy Extensions for Stateless Address Autoconfiguration [RFC4941] address this concern by allowing nodes to configure an additional temporary address where the IID is effectively randomly generated. Privacy addresses are then used as source addresses for new communications initiated by the node.

General issues regarding privacy issues for IPv6 addressing are discussed in [RFC7721].

RFC 4941 SHOULD be supported. In some scenarios, such as dedicated servers in a data center, it provides limited or no benefit, or may complicate network management. Thus devices implementing this specification MUST provide a way for the end user to explicitly enable or disable the use of such temporary addresses.

Note that RFC4941 can be used independently of traditional SLAAC, or of RFC7217-based SLAAC.

Implementers of RFC 4941 should be aware that certain addresses are reserved and should not be chosen for use as temporary addresses. Consult "Reserved IPv6 Interface Identifiers" [RFC5453] for more details.

6.5. Stateful Address Autoconfiguration (DHCPv6) - RFC 3315

DHCPv6 [RFC3315] can be used to obtain and configure addresses. In general, a network may provide for the configuration of addresses through SLAAC, DHCPv6, or both. There will be a wide range of IPv6 deployment models and differences in address assignment requirements, some of which may require DHCPv6 for stateful address assignment. Consequently, all hosts SHOULD implement address configuration via DHCPv6.

In the absence of observed Router Advertisement messages, IPv6 nodes MAY initiate DHCP to obtain IPv6 addresses and other configuration information, as described in Section 5.5.2 of [RFC4862].

Where devices are likely to be carried by users and attached to multiple visisted networks, DHCPv6 client anonymity profiles SHOULD be supported as described in [RFC7844] to minimise the disclosure of

identifying information. Section 5 of RFC7844 describes operational considerations on the use of such anonymity profiles.

6.6. Default Address Selection for IPv6 - RFC 6724

IPv6 nodes will invariably have multiple addresses configured simultaneously, and thus will need to choose which addresses to use for which communications. The rules specified in the Default Address Selection for IPv6 [RFC6724] document MUST be implemented. [RFC8028] updates rule 5.5 from [RFC6724]; implementations SHOULD implement this rule.

7. DNS

DNS is described in [RFC1034], [RFC1035], [RFC3363], and [RFC3596]. Not all nodes will need to resolve names; those that will never need to resolve DNS names do not need to implement resolver functionality. However, the ability to resolve names is a basic infrastructure capability on which applications rely, and most nodes will need to provide support. All nodes SHOULD implement stub-resolver [RFC1034] functionality, as in [RFC1034], Section 5.3.1, with support for:

- AAAA type Resource Records [RFC3596];

- reverse addressing in ip6.arpa using PTR records [RFC3596];

- Extension Mechanisms for DNS (EDNS0) [RFC6891] to allow for DNS packet sizes larger than 512 octets.

Those nodes are RECOMMENDED to support DNS security extensions [RFC4033] [RFC4034] [RFC4035].

A6 Resource Records, which were only ever defined with Experimental status in [RFC3363], are now classified as Historic, as per [RFC6563].

8. Configuring Non-Address Information

8.1. DHCP for Other Configuration Information

DHCP [RFC3315] Specifies a mechanism for IPv6 nodes to obtain address configuration information (see Section 6.5) and to obtain additional (non-address) configuration. If a host implementation supports applications or other protocols that require configuration that is only available via DHCP, hosts SHOULD implement DHCP. For specialized devices on which no such configuration need is present, DHCP may not be necessary.

An IPv6 node can use the subset of DHCP (described in [RFC3736]) to obtain other configuration information.

If an IPv6 node implements DHCP it MUST implement the DNS options [RFC3646] as most deployments will expect these options are available.

8.2. Router Advertisements and Default Gateway

There is no defined DHCPv6 Gateway option.

Nodes using the Dynamic Host Configuration Protocol for IPv6 (DHCPv6) are thus expected to determine their default router information and on-link prefix information from received Router Advertisements.

8.3. IPv6 Router Advertisement Options for DNS Configuration - RFC 8106

Router Advertisement Options have historically been limited to those that are critical to basic IPv6 functionality. Originally, DNS configuration was not included as an RA option, and DHCP was the recommended way to obtain DNS configuration information. Over time, the thinking surrounding such an option has evolved. It is now generally recognized that few nodes can function adequately without having access to a working DNS resolver, and thus a Standards Track document has been published to provide this capability [RFC8106].

Implementations MUST include support for the DNS RA option [RFC8106].

8.4. DHCP Options versus Router Advertisement Options for Host Configuration

In IPv6, there are two main protocol mechanisms for propagating configuration information to hosts: Router Advertisements (RAs) and DHCP. RA options have been restricted to those deemed essential for basic network functioning and for which all nodes are configured with exactly the same information. Examples include the Prefix Information Options, the MTU option, etc. On the other hand, DHCP has generally been preferred for configuration of more general parameters and for parameters that may be client-specific. Generally speaking, however, there has been a desire to define only one mechanism for configuring a given option, rather than defining multiple (different) ways of configuring the same information.

One issue with having multiple ways of configuring the same information is that interoperability suffers if a host chooses one mechanism but the network operator chooses a different mechanism. For "closed" environments, where the network operator has significant influence over what devices connect to the network and thus what

configuration mechanisms they support, the operator may be able to ensure that a particular mechanism is supported by all connected hosts. In more open environments, however, where arbitrary devices may connect (e.g., a WIFI hotspot), problems can arise. To maximize interoperability in such environments, hosts would need to implement multiple configuration mechanisms to ensure interoperability.

9. Service Discovery Protocols

[RFC6762] and [RFC6763] describe multicast DNS (mDNS) and DNS-Based Service Discovery (DNS-SD) respectively. These protocols, collectively commonly referred to as the 'Bonjour' protocols after their naming by Apple, provide the means for devices to discover services within a local link and, in the absence of a unicast DNS service, to exchange naming information.

Where devices are to be deployed in networks where service dicovery would be beneficial, e.g., for users seeking to discover printers or display devices, mDNS and DNS-SD SHOULD be supported.

10. IPv4 Support and Transition

IPv6 nodes MAY support IPv4.

10.1. Transition Mechanisms

10.1.1. Basic Transition Mechanisms for IPv6 Hosts and Routers - RFC 4213

If an IPv6 node implements dual stack and tunneling, then [RFC4213] MUST be supported.

11. Application Support

11.1. Textual Representation of IPv6 Addresses - RFC 5952

Software that allows users and operators to input IPv6 addresses in text form SHOULD support "A Recommendation for IPv6 Address Text Representation" [RFC5952].

11.2. Application Programming Interfaces (APIs)

There are a number of IPv6-related APIs. This document does not mandate the use of any, because the choice of API does not directly relate to on-the-wire behavior of protocols. Implementers, however, would be advised to consider providing a common API or reviewing existing APIs for the type of functionality they provide to applications.

"Basic Socket Interface Extensions for IPv6" [RFC3493] provides IPv6 functionality used by typical applications. Implementers should note that RFC3493 has been picked up and further standardized by the Portable Operating System Interface (POSIX) [POSIX].

"Advanced Sockets Application Program Interface (API) for IPv6" [RFC3542] provides access to advanced IPv6 features needed by diagnostic and other more specialized applications.

"IPv6 Socket API for Source Address Selection" [RFC5014] provides facilities that allow an application to override the default Source Address Selection rules of [RFC6724].

"Socket Interface Extensions for Multicast Source Filters" [RFC3678] provides support for expressing source filters on multicast group memberships.

"Extension to Sockets API for Mobile IPv6" [RFC4584] provides application support for accessing and enabling Mobile IPv6 [RFC6275] features.

12. Mobility

Mobile IPv6 [RFC6275] and associated specifications [RFC3776] [RFC4877] allow a node to change its point of attachment within the Internet, while maintaining (and using) a permanent address. All communication using the permanent address continues to proceed as expected even as the node moves around. The definition of Mobile IP includes requirements for the following types of nodes:

- mobile nodes
- correspondent nodes with support for route optimization
- home agents
- all IPv6 routers

At the present time, Mobile IP has seen only limited implementation and no significant deployment, partly because it originally assumed an IPv6-only environment rather than a mixed IPv4/IPv6 Internet. Recently, additional work has been done to support mobility in mixed-mode IPv4 and IPv6 networks [RFC5555].

More usage and deployment experience is needed with mobility before any specific approach can be recommended for broad implementation in all hosts and routers. Consequently, [RFC6275], [RFC5555], and

associated standards such as [RFC4877] are considered a MAY at this time.

IPv6 for 3GPP [RFC7066] lists a snapshot of required IPv6 Functionalities at the time the document was published that would need to be implemented, going above and beyond the recommendations in this document. Additionally a 3GPP IPv6 Host MAY implement [RFC7278] for delivering IPv6 prefixes on the LAN link.

13. Security

This section describes the specification for security for IPv6 nodes.

Achieving security in practice is a complex undertaking. Operational procedures, protocols, key distribution mechanisms, certificate management approaches, etc., are all components that impact the level of security actually achieved in practice. More importantly, deficiencies or a poor fit in any one individual component can significantly reduce the overall effectiveness of a particular security approach.

IPsec either can provide end-to-end security between nodes or or can provide channel security (for example, via a site-to-site IPsec VPN), making it possible to provide secure communication for all (or a subset of) communication flows at the IP layer between pairs of internet nodes. IPsec has two standard operating modes, Tunnel-mode and Transport-mode. In Tunnel-mode, IPsec provides network-layer security and protects an entire IP packet by encapsulating the orginal IP packet and then pre-pending a new IP header. In Transport-mode, IPsec provides security for the transport-layer (and above) by encapsulating only the transport-layer (and above) portion of the IP packet (i.e., without adding a 2nd IP header).

Although IPsec can be used with manual keying in some cases, such usage has limited applicability and is not recommended.

A range of security technologies and approaches proliferate today (e.g., IPsec, Transport Layer Security (TLS), Secure SHell (SSH), TLS VPNS, etc.) No one approach has emerged as an ideal technology for all needs and environments. Moreover, IPsec is not viewed as the ideal security technology in all cases and is unlikely to displace the others.

Previously, IPv6 mandated implementation of IPsec and recommended the key management approach of IKE. This document updates that recommendation by making support of the IPsec Architecture [RFC4301] a SHOULD for all IPv6 nodes. Note that the IPsec Architecture requires (e.g., Section 4.5 of RFC 4301) the implementation of both

manual and automatic key management. Currently, the recommended automated key management protocol to implement is IKEv2 [RFC7296].

This document recognizes that there exists a range of device types and environments where approaches to security other than IPsec can be justified. For example, special-purpose devices may support only a very limited number or type of applications, and an application-specific security approach may be sufficient for limited management or configuration capabilities. Alternatively, some devices may run on extremely constrained hardware (e.g., sensors) where the full IPsec Architecture is not justified.

Because most common platforms now support IPv6 and have it enabled by default, IPv6 security is an issue for networks that are ostensibly IPv4-only; see [RFC7123] for guidance on this area.

13.1. Requirements

"Security Architecture for the Internet Protocol" [RFC4301] SHOULD be supported by all IPv6 nodes. Note that the IPsec Architecture requires (e.g., Section 4.5 of [RFC4301]) the implementation of both manual and automatic key management. Currently, the default automated key management protocol to implement is IKEv2. As required in [RFC4301], IPv6 nodes implementing the IPsec Architecture MUST implement ESP [RFC4303] and MAY implement AH [RFC4302].

13.2. Transforms and Algorithms

The current set of mandatory-to-implement algorithms for the IPsec Architecture are defined in "Cryptographic Algorithm Implementation Requirements For ESP and AH" [RFC8221]. IPv6 nodes implementing the IPsec Architecture MUST conform to the requirements in [RFC8221]. Preferred cryptographic algorithms often change more frequently than security protocols. Therefore, implementations MUST allow for migration to new algorithms, as RFC 8221 is replaced or updated in the future.

The current set of mandatory-to-implement algorithms for IKEv2 are defined in "Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)" [RFC8247]. IPv6 nodes implementing IKEv2 MUST conform to the requirements in [RFC8247] and/or any future updates or replacements to [RFC8247].

14. Router-Specific Functionality

This section defines general host considerations for IPv6 nodes that act as routers. Currently, this section does not discuss detailed

routing-specific requirements. For the case of typical home routers, [RFC7084] defines basic requirements for customer edge routers.

14.1. IPv6 Router Alert Option - RFC 2711

The IPv6 Router Alert Option [RFC2711] is an optional IPv6 Hop-by-Hop Header that is used in conjunction with some protocols (e.g., RSVP [RFC2205] or Multicast Listener Discovery (MLDv2) [RFC3810]). The Router Alert option will need to be implemented whenever such protocols that mandate its use are implemented. See Section 5.11.

14.2. Neighbor Discovery for IPv6 - RFC 4861

Sending Router Advertisements and processing Router Solicitations MUST be supported.

Section 7 of [RFC6275] includes some mobility-specific extensions to Neighbor Discovery. Routers SHOULD implement Sections 7.3 and 7.5, even if they do not implement Home Agent functionality.

14.3. Stateful Address Autoconfiguration (DHCPv6) - RFC 3315

A single DHCP server ([RFC3315] or [RFC4862]) can provide configuration information to devices directly attached to a shared link, as well as to devices located elsewhere within a site. Communication between a client and a DHCP server located on different links requires the use of DHCP relay agents on routers.

In simple deployments, consisting of a single router and either a single LAN or multiple LANs attached to the single router, together with a WAN connection, a DHCP server embedded within the router is one common deployment scenario (e.g., [RFC7084]). There is no need for relay agents in such scenarios.

In more complex deployment scenarios, such as within enterprise or service provider networks, the use of DHCP requires some level of configuration, in order to configure relay agents, DHCP servers, etc. In such environments, the DHCP server might even be run on a traditional server, rather than as part of a router.

Because of the wide range of deployment scenarios, support for DHCP server functionality on routers is optional. However, routers targeted for deployment within more complex scenarios (as described above) SHOULD support relay agent functionality. Note that "Basic Requirements for IPv6 Customer Edge Routers" [RFC7084] requires implementation of a DHCPv6 server function in IPv6 Customer Edge (CE) routers.

14.4. IPv6 Prefix Length Recommendation for Forwarding - BCP 198

Forwarding nodes MUST conform to BCP 198 [RFC7608] and thus IPv6 implementations of nodes that may forward packets MUST conform to the rules specified in Section 5.1 of [RFC4632].

15. Constrained Devices

The target for this document is general IPv6 nodes. In this Section, we briefly discuss considerations for constrained devices.

In the case of constrained nodes, with limited CPU, memory, bandwidth or power, support for certain IPv6 functionality may need to be considered due to those limitations. While the requirements of this document are RECOMMENDED for all nodes, including constrained nodes, compromises may need to be made in certain cases. Where such compromises are made, the interoperability of devices should be strongly considered, paticularly where this may impact other nodes on the same link, e.g., only supporting MLDv1 will affect other nodes.

The IETF 6LowPAN (IPv6 over Low Power LWPAN) WG defined six RFCs, including a general overview and problem statement ([RFC4919], the means by which IPv6 packets are transmitted over IEEE 802.15.4 networks [RFC4944] and ND optimisations for that medium [RFC6775].

IPv6 nodes that are battery-powered SHOULD implement the recommendations in [RFC7772].

16. IPv6 Node Management

Network management MAY be supported by IPv6 nodes. However, for IPv6 nodes that are embedded devices, network management may be the only possible way of controlling these nodes.

Existing network management protocols include SNMP [RFC3411], NETCONF [RFC6241] and RESTCONF [RFC8040].

16.1. Management Information Base (MIB) Modules

[RFC8096] clarifies the obsoleted status of various IPv6-specific MIB modules.

The following two MIB modules SHOULD be supported by nodes that support a Simple Network Management Protocol (SNMP) agent.

16.1.1. IP Forwarding Table MIB

The IP Forwarding Table MIB [RFC4292] SHOULD be supported by nodes that support an SNMP agent.

16.1.2. Management Information Base for the Internet Protocol (IP)

The IP MIB [RFC4293] SHOULD be supported by nodes that support an SNMP agent.

16.1.3. Interface MIB

The Interface MIB [RFC2863] SHOULD be supported by nodes the support an SNMP agent.

16.2. YANG Data Models

The following YANG data models SHOULD be supported by nodes that support a NETCONF or RESTCONF agent.

16.2.1. IP Management YANG Model

The IP Management YANG Model [I-D.ietf-netmod-rfc7277bis] SHOULD be supported by nodes that support NETCONF or RESTCONF.

16.2.2. Interface Management YANG Model

The Interface Management YANG Model [I-D.ietf-netmod-rfc7223bis] SHOULD be supported by nodes that support NETCONF or RESTCONF.

17. Security Considerations

This document does not directly affect the security of the Internet, beyond the security considerations associated with the individual protocols.

Security is also discussed in Section 13 above.

18. IANA Considerations

This document does not require any IANA actions.

19. Authors and Acknowledgments

19.1. Authors and Acknowledgments (Current Document)

For this version of the IPv6 Node Requirements document, the authors would like to thank Brian Carpenter, Dave Thaler, Tom Herbert, Erik Kline, Mohamed Boucadair, and Michayla Newcombe for their contributions.

19.2. Authors and Acknowledgments from RFC 6434

Ed Jankiewicz and Thomas Narten were named authors of the previous iteration of this document, RFC6434.

For this version of the document, the authors thanked Hitoshi Asaeda, Brian Carpenter, Tim Chown, Ralph Droms, Sheila Frankel, Sam Hartman, Bob Hinden, Paul Hoffman, Pekka Savola, Yaron Sheffer, and Dave Thaler.

19.3. Authors and Acknowledgments from RFC 4294

The original version of this document (RFC 4294) was written by the IPv6 Node Requirements design team, which had the following members: Jari Arkko, Marc Blanchet, Samita Chakrabarti, Alain Durand, Gerard Gastaud, Jun-ichiro Itojun Hagino, Atsushi Inoue, Masahiro Ishiyama, John Loughney, Rajiv Raghunarayan, Shoichi Sakane, Dave Thaler, and Juha Wiljakka.

The authors would like to thank Ran Atkinson, Jim Bound, Brian Carpenter, Ralph Droms, Christian Huitema, Adam Machalek, Thomas Narten, Juha Ollila, and Pekka Savola for their comments. Thanks to Mark Andrews for comments and corrections on DNS text. Thanks to Alfred Hoenes for tracking the updates to various RFCs.

20. Appendix: Changes from RFC 6434

There have been many editorial clarifications as well as significant additions and updates. While this section highlights some of the changes, readers should not rely on this section for a comprehensive list of all changes.

1. Restructured sections

2. Added 6LoWPAN to link layers as it has some deployment.

3. Removed DOD IPv6 Profile as it hasn’t been updated.

4. Updated to MLDv2 support to a MUST since nodes are restricted if MLDv1 is used.

5. Require DNS RA Options so SLAAC-only devices can get DNS, RFC8106 is a MUST.

6. Require RFC3646 DNS Options for DHCPv6 implementations.

7. Added RESTCONF and NETCONF as possible options to Network management.

8. Added section on constrained devices.

9. Added text on RFC7934, address availability to hosts (SHOULD).

10. Added text on RFC7844, anonymity profiles for DHCPv6 clients.

11. mDNS and DNS-SD added as updated service discovery.

12. Added RFC8028 as a SHOULD as a method for solving multi-prefix network

13. Added ECN RFC3168 as a SHOULD

14. Added reference to RFC7123 for Security over IPv4-only networks

15. Removed Jumbograms RFC2675 as they aren’t deployed.

16. Updated Obseleted RFCs to the new version of the RFC including 2460, 1981, 7321, 4307

17. Added RFC7772 for power comsumptions considerations

18. Added why /64 boundries for more detail - RFC 7421

19. Added a Unique IPv6 Prefix per Host to support currently deployed IPv6 networks

20. Clarified RFC7066 was snapshot for 3GPP

21. Updated 4191 as a MUST, SHOULD for Type C Host as it helps solve multi-prefix problem

22. Removed IPv6 over ATM since there aren’t many deployments

23. Added a note in Section 6.6 for RFC6724 Section 5.5/

24. Added MUST for BCP 198 for forwarding IPv6 packets

25. Added reference to RFC8064 for stable address creation.

26. Added text on protection from excessive EH options

27. Added text on dangers of 1280 MTU UDP, esp. wrt DNS traffic

28. Added text to clarify RFC8200 behaviour for unrecognized EHs or unrecognized ULPs

29. Removed dated email addresses from design team acknowledgements for RFC 4294.

21. Appendix: Changes from RFC 4294

There have been many editorial clarifications as well as significant additions and updates. While this section highlights some of the changes, readers should not rely on this section for a comprehensive list of all changes.

1. Updated the Introduction to indicate that this document is an applicability statement and is aimed at general nodes.

2. Significantly updated the section on Mobility protocols, adding references and downgrading previous SHOULDs to MAYs.

3. Changed Sub-IP Layer section to just list relevant RFCs, and added some more RFCs.

4. Added section on SEND (it is a MAY).

5. Revised section on Privacy Extensions [RFC4941] to add more nuance to recommendation.

6. Completely revised IPsec/IKEv2 section, downgrading overall recommendation to a SHOULD.

7. Upgraded recommendation of DHCPv6 to SHOULD.

8. Added background section on DHCP versus RA options, added SHOULD recommendation for DNS configuration via RAs (RFC6106), and cleaned up DHCP recommendations.

9. Added recommendation that routers implement Sections 7.3 and 7.5 of [RFC6275].

10. Added pointer to subnet clarification document [RFC5942].

11. Added text that "IPv6 Host-to-Router Load Sharing" [RFC4311] SHOULD be implemented.

12. Added reference to [RFC5722] (Overlapping Fragments), and made it a MUST to implement.

13. Made "A Recommendation for IPv6 Address Text Representation" [RFC5952] a SHOULD.

14. Removed mention of "DNAME" from the discussion about [RFC3363].

15. Numerous updates to reflect newer versions of IPv6 documents, including [RFC4443], [RFC4291], [RFC3596], and [RFC4213].

16. Removed discussion of "Managed" and "Other" flags in RAs. There is no consensus at present on how to process these flags, and discussion of their semantics was removed in the most recent update of Stateless Address Autoconfiguration [RFC4862].

17. Added many more references to optional IPv6 documents.

18. Made "A Recommendation for IPv6 Address Text Representation" [RFC5952] a SHOULD.

19. Added reference to [RFC5722] (Overlapping Fragments), and made it a MUST to implement.

20. Updated MLD section to include reference to Lightweight MLD [RFC5790].

21. Added SHOULD recommendation for "Default Router Preferences and More-Specific Routes" [RFC4191].

22. Made "IPv6 Flow Label Specification" [RFC6437] a SHOULD.

22. References

22.1. Normative References

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[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>.

[RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, DOI 10.17487/RFC2710, October 1999, <https://www.rfc-editor.org/info/rfc2710>.

[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>.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <https://www.rfc-editor.org/info/rfc2863>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/info/rfc3168>.

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <https://www.rfc-editor.org/info/rfc3315>.

[RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, DOI 10.17487/RFC3411, December 2002, <https://www.rfc-editor.org/info/rfc3411>.

[RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", STD 88, RFC 3596, DOI 10.17487/RFC3596, October 2003, <https://www.rfc-editor.org/info/rfc3596>.

[RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, DOI 10.17487/RFC3736, April 2004, <https://www.rfc-editor.org/info/rfc3736>.

[RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <https://www.rfc-editor.org/info/rfc3810>.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <https://www.rfc-editor.org/info/rfc4033>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/info/rfc4034>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <https://www.rfc-editor.org/info/rfc4035>.

[RFC4213] Nordmark, E. and R. Gilligan, "Basic Transition Mechanisms for IPv6 Hosts and Routers", RFC 4213, DOI 10.17487/RFC4213, October 2005, <https://www.rfc-editor.org/info/rfc4213>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

[RFC4292] Haberman, B., "IP Forwarding Table MIB", RFC 4292, DOI 10.17487/RFC4292, April 2006, <https://www.rfc-editor.org/info/rfc4292>.

[RFC4293] Routhier, S., Ed., "Management Information Base for the Internet Protocol (IP)", RFC 4293, DOI 10.17487/RFC4293, April 2006, <https://www.rfc-editor.org/info/rfc4293>.

[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>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

[RFC4311] Hinden, R. and D. Thaler, "IPv6 Host-to-Router Load Sharing", RFC 4311, DOI 10.17487/RFC4311, November 2005, <https://www.rfc-editor.org/info/rfc4311>.

[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>.

[RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, DOI 10.17487/RFC4607, August 2006, <https://www.rfc-editor.org/info/rfc4607>.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, DOI 10.17487/RFC4632, August 2006, <https://www.rfc-editor.org/info/rfc4632>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <https://www.rfc-editor.org/info/rfc4862>.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <https://www.rfc-editor.org/info/rfc4941>.

[RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, DOI 10.17487/RFC5095, December 2007, <https://www.rfc-editor.org/info/rfc5095>.

[RFC5453] Krishnan, S., "Reserved IPv6 Interface Identifiers", RFC 5453, DOI 10.17487/RFC5453, February 2009, <https://www.rfc-editor.org/info/rfc5453>.

[RFC5722] Krishnan, S., "Handling of Overlapping IPv6 Fragments", RFC 5722, DOI 10.17487/RFC5722, December 2009, <https://www.rfc-editor.org/info/rfc5722>.

[RFC5790] Liu, H., Cao, W., and H. Asaeda, "Lightweight Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Version 2 (MLDv2) Protocols", RFC 5790, DOI 10.17487/RFC5790, February 2010, <https://www.rfc-editor.org/info/rfc5790>.

[RFC5942] Singh, H., Beebee, W., and E. Nordmark, "IPv6 Subnet Model: The Relationship between Links and Subnet Prefixes", RFC 5942, DOI 10.17487/RFC5942, July 2010, <https://www.rfc-editor.org/info/rfc5942>.

[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>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, DOI 10.17487/RFC6564, April 2012, <https://www.rfc-editor.org/info/rfc6564>.

[RFC6724] Thaler, D., Ed., Draves, R., Matsumoto, A., and T. Chown, "Default Address Selection for Internet Protocol Version 6 (IPv6)", RFC 6724, DOI 10.17487/RFC6724, September 2012, <https://www.rfc-editor.org/info/rfc6724>.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <https://www.rfc-editor.org/info/rfc6762>.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <https://www.rfc-editor.org/info/rfc6763>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <https://www.rfc-editor.org/info/rfc6775>.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <https://www.rfc-editor.org/info/rfc6891>.

[RFC6946] Gont, F., "Processing of IPv6 "Atomic" Fragments", RFC 6946, DOI 10.17487/RFC6946, May 2013, <https://www.rfc-editor.org/info/rfc6946>.

[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>.

[RFC7048] Nordmark, E. and I. Gashinsky, "Neighbor Unreachability Detection Is Too Impatient", RFC 7048, DOI 10.17487/RFC7048, January 2014, <https://www.rfc-editor.org/info/rfc7048>.

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

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <https://www.rfc-editor.org/info/rfc7217>.

[I-D.ietf-netmod-rfc7223bis] Bjorklund, M., "A YANG Data Model for Interface Management", draft-ietf-netmod-rfc7223bis-03 (work in progress), January 2018.

[I-D.ietf-netmod-rfc7277bis] Bjorklund, M., "A YANG Data Model for IP Management", draft-ietf-netmod-rfc7277bis-03 (work in progress), January 2018.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

[RFC7527] Asati, R., Singh, H., Beebee, W., Pignataro, C., Dart, E., and W. George, "Enhanced Duplicate Address Detection", RFC 7527, DOI 10.17487/RFC7527, April 2015, <https://www.rfc-editor.org/info/rfc7527>.

[RFC7559] Krishnan, S., Anipko, D., and D. Thaler, "Packet-Loss Resiliency for Router Solicitations", RFC 7559, DOI 10.17487/RFC7559, May 2015, <https://www.rfc-editor.org/info/rfc7559>.

[RFC7608] Boucadair, M., Petrescu, A., and F. Baker, "IPv6 Prefix Length Recommendation for Forwarding", BCP 198, RFC 7608, DOI 10.17487/RFC7608, July 2015, <https://www.rfc-editor.org/info/rfc7608>.

[RFC8021] Gont, F., Liu, W., and T. Anderson, "Generation of IPv6 Atomic Fragments Considered Harmful", RFC 8021, DOI 10.17487/RFC8021, January 2017, <https://www.rfc-editor.org/info/rfc8021>.

[RFC8028] Baker, F. and 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>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8064] Gont, F., Cooper, A., Thaler, D., and W. Liu, "Recommendation on Stable IPv6 Interface Identifiers", RFC 8064, DOI 10.17487/RFC8064, February 2017, <https://www.rfc-editor.org/info/rfc8064>.

[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>.

[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>.

[RFC8221] Wouters, P., Migault, D., Mattsson, J., Nir, Y., and T. Kivinen, "Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 8221, DOI 10.17487/RFC8221, October 2017, <https://www.rfc-editor.org/info/rfc8221>.

[RFC8247] Nir, Y., Kivinen, T., Wouters, P., and D. Migault, "Algorithm Implementation Requirements and Usage Guidance for the Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 8247, DOI 10.17487/RFC8247, September 2017, <https://www.rfc-editor.org/info/rfc8247>.

22.2. Informative References

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <https://www.rfc-editor.org/info/rfc793>.

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <https://www.rfc-editor.org/info/rfc2205>.

[RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, DOI 10.17487/RFC2464, December 1998, <https://www.rfc-editor.org/info/rfc2464>.

[RFC2491] Armitage, G., Schulter, P., Jork, M., and G. Harter, "IPv6 over Non-Broadcast Multiple Access (NBMA) networks", RFC 2491, DOI 10.17487/RFC2491, January 1999, <https://www.rfc-editor.org/info/rfc2491>.

[RFC2590] Conta, A., Malis, A., and M. Mueller, "Transmission of IPv6 Packets over Frame Relay Networks Specification", RFC 2590, DOI 10.17487/RFC2590, May 1999, <https://www.rfc-editor.org/info/rfc2590>.

[RFC2923] Lahey, K., "TCP Problems with Path MTU Discovery", RFC 2923, DOI 10.17487/RFC2923, September 2000, <https://www.rfc-editor.org/info/rfc2923>.

[RFC3146] Fujisawa, K. and A. Onoe, "Transmission of IPv6 Packets over IEEE 1394 Networks", RFC 3146, DOI 10.17487/RFC3146, October 2001, <https://www.rfc-editor.org/info/rfc3146>.

[RFC3363] Bush, R., Durand, A., Fink, B., Gudmundsson, O., and T. Hain, "Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System (DNS)", RFC 3363, DOI 10.17487/RFC3363, August 2002, <https://www.rfc-editor.org/info/rfc3363>.

[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>.

[RFC3542] Stevens, W., Thomas, M., Nordmark, E., and T. Jinmei, "Advanced Sockets Application Program Interface (API) for IPv6", RFC 3542, DOI 10.17487/RFC3542, May 2003, <https://www.rfc-editor.org/info/rfc3542>.

[RFC3646] Droms, R., Ed., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, DOI 10.17487/RFC3646, December 2003, <https://www.rfc-editor.org/info/rfc3646>.

[RFC3678] Thaler, D., Fenner, B., and B. Quinn, "Socket Interface Extensions for Multicast Source Filters", RFC 3678, DOI 10.17487/RFC3678, January 2004, <https://www.rfc-editor.org/info/rfc3678>.

[RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <https://www.rfc-editor.org/info/rfc6275>.

[RFC3776] Arkko, J., Devarapalli, V., and F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents", RFC 3776, DOI 10.17487/RFC3776, June 2004, <https://www.rfc-editor.org/info/rfc3776>.

[RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <https://www.rfc-editor.org/info/rfc3971>.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <https://www.rfc-editor.org/info/rfc3972>.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <https://www.rfc-editor.org/info/rfc4191>.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <https://www.rfc-editor.org/info/rfc4302>.

[RFC4338] DeSanti, C., Carlson, C., and R. Nixon, "Transmission of IPv6, IPv4, and Address Resolution Protocol (ARP) Packets over Fibre Channel", RFC 4338, DOI 10.17487/RFC4338, January 2006, <https://www.rfc-editor.org/info/rfc4338>.

[RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, DOI 10.17487/RFC4380, February 2006, <https://www.rfc-editor.org/info/rfc4380>.

[RFC4429] Moore, N., "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, DOI 10.17487/RFC4429, April 2006, <https://www.rfc-editor.org/info/rfc4429>.

[RFC4584] Chakrabarti, S. and E. Nordmark, "Extension to Sockets API for Mobile IPv6", RFC 4584, DOI 10.17487/RFC4584, July 2006, <https://www.rfc-editor.org/info/rfc4584>.

[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>.

[RFC4877] Devarapalli, V. and F. Dupont, "Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture", RFC 4877, DOI 10.17487/RFC4877, April 2007, <https://www.rfc-editor.org/info/rfc4877>.

[RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, DOI 10.17487/RFC4884, April 2007, <https://www.rfc-editor.org/info/rfc4884>.

[RFC4890] Davies, E. and J. Mohacsi, "Recommendations for Filtering ICMPv6 Messages in Firewalls", RFC 4890, DOI 10.17487/RFC4890, May 2007, <https://www.rfc-editor.org/info/rfc4890>.

[RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, DOI 10.17487/RFC4919, August 2007, <https://www.rfc-editor.org/info/rfc4919>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <https://www.rfc-editor.org/info/rfc4944>.

[RFC5014] Nordmark, E., Chakrabarti, S., and J. Laganier, "IPv6 Socket API for Source Address Selection", RFC 5014, DOI 10.17487/RFC5014, September 2007, <https://www.rfc-editor.org/info/rfc5014>.

[RFC5072] Varada, S., Ed., Haskins, D., and E. Allen, "IP Version 6 over PPP", RFC 5072, DOI 10.17487/RFC5072, September 2007, <https://www.rfc-editor.org/info/rfc5072>.

[RFC5121] Patil, B., Xia, F., Sarikaya, B., Choi, JH., and S. Madanapalli, "Transmission of IPv6 via the IPv6 Convergence Sublayer over IEEE 802.16 Networks", RFC 5121, DOI 10.17487/RFC5121, February 2008, <https://www.rfc-editor.org/info/rfc5121>.

[RFC5555] Soliman, H., Ed., "Mobile IPv6 Support for Dual Stack Hosts and Routers", RFC 5555, DOI 10.17487/RFC5555, June 2009, <https://www.rfc-editor.org/info/rfc5555>.

[RFC6563] Jiang, S., Conrad, D., and B. Carpenter, "Moving A6 to Historic Status", RFC 6563, DOI 10.17487/RFC6563, March 2012, <https://www.rfc-editor.org/info/rfc6563>.

[RFC7066] Korhonen, J., Ed., Arkko, J., Ed., Savolainen, T., and S. Krishnan, "IPv6 for Third Generation Partnership Project (3GPP) Cellular Hosts", RFC 7066, DOI 10.17487/RFC7066, November 2013, <https://www.rfc-editor.org/info/rfc7066>.

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

[RFC7123] Gont, F. and W. Liu, "Security Implications of IPv6 on IPv4 Networks", RFC 7123, DOI 10.17487/RFC7123, February 2014, <https://www.rfc-editor.org/info/rfc7123>.

[RFC7278] Byrne, C., Drown, D., and A. Vizdal, "Extending an IPv6 /64 Prefix from a Third Generation Partnership Project (3GPP) Mobile Interface to a LAN Link", RFC 7278, DOI 10.17487/RFC7278, June 2014, <https://www.rfc-editor.org/info/rfc7278>.

[RFC7371] Boucadair, M. and S. Venaas, "Updates to the IPv6 Multicast Addressing Architecture", RFC 7371, DOI 10.17487/RFC7371, September 2014, <https://www.rfc-editor.org/info/rfc7371>.

[RFC7421] Carpenter, B., Ed., Chown, T., Gont, F., Jiang, S., Petrescu, A., and A. Yourtchenko, "Analysis of the 64-bit Boundary in IPv6 Addressing", RFC 7421, DOI 10.17487/RFC7421, January 2015, <https://www.rfc-editor.org/info/rfc7421>.

[RFC7721] Cooper, A., Gont, F., and D. Thaler, "Security and Privacy Considerations for IPv6 Address Generation Mechanisms", RFC 7721, DOI 10.17487/RFC7721, March 2016, <https://www.rfc-editor.org/info/rfc7721>.

[RFC7739] Gont, F., "Security Implications of Predictable Fragment Identification Values", RFC 7739, DOI 10.17487/RFC7739, February 2016, <https://www.rfc-editor.org/info/rfc7739>.

[RFC7772] Yourtchenko, A. and L. Colitti, "Reducing Energy Consumption of Router Advertisements", BCP 202, RFC 7772, DOI 10.17487/RFC7772, February 2016, <https://www.rfc-editor.org/info/rfc7772>.

[RFC7844] Huitema, C., Mrugalski, T., and S. Krishnan, "Anonymity Profiles for DHCP Clients", RFC 7844, DOI 10.17487/RFC7844, May 2016, <https://www.rfc-editor.org/info/rfc7844>.

[RFC7934] Colitti, L., Cerf, V., Cheshire, S., and D. Schinazi, "Host Address Availability Recommendations", BCP 204, RFC 7934, DOI 10.17487/RFC7934, July 2016, <https://www.rfc-editor.org/info/rfc7934>.

[RFC8087] Fairhurst, G. and M. Welzl, "The Benefits of Using Explicit Congestion Notification (ECN)", RFC 8087, DOI 10.17487/RFC8087, March 2017, <https://www.rfc-editor.org/info/rfc8087>.

[RFC8096] Fenner, B., "The IPv6-Specific MIB Modules Are Obsolete", RFC 8096, DOI 10.17487/RFC8096, April 2017, <https://www.rfc-editor.org/info/rfc8096>.

[RFC8273] Brzozowski, J. and G. Van de Velde, "Unique IPv6 Prefix per Host", RFC 8273, DOI 10.17487/RFC8273, December 2017, <https://www.rfc-editor.org/info/rfc8273>.

[POSIX] IEEE, "IEEE Std. 1003.1-2008 Standard for Information Technology -- Portable Operating System Interface (POSIX), ISO/IEC 9945:2009", <http://www.ieee.org>.

[USGv6] National Institute of Standards and Technology, "A Profile for IPv6 in the U.S. Government - Version 1.0", July 2008, <https://doi.org/10.6028/NIST.SP.500-267Ar1>.

Authors' Addresses

Tim Chown
Jisc
Lumen House, Library Avenue
Harwell Oxford, Didcot OX11 0SG
United Kingdom

Email: tim.chown@jisc.ac.uk

John Loughney
Intel
Santa Clara, CA
USA

Email: john.loughney@gmail.com

Timothy Winters
University of New Hampshire, Interoperability Lab (UNH-IOL)
Durham, NH
United States

Email: twinters@iol.unh.edu

opsec
Internet-Draft
Intended status: Informational
Expires: January 3, 2019

F. Gont
UTN-FRH / SI6 Networks
W. Liu
Huawei Technologies

July 2, 2018

Recommendations on the Filtering of IPv6 Packets Containing IPv6 Extension Headers
draft-ietf-opsec-ipv6-eh-filtering-06

Abstract

It is common operator practice to mitigate security risks by enforcing appropriate packet filtering. This document analyzes both the general security implications of IPv6 Extension Headers and the specific security implications of each Extension Header and Option type. Additionally, it discusses the operational and interoperability implications of discarding packets based on the IPv6 Extension Headers and IPv6 options they contain. Finally, it provides advice on the filtering of such IPv6 packets at transit routers for traffic *not* directed to them, for those cases in which such filtering is deemed as necessary.

Table of Contents

1. Introduction 2
2. Terminology and Conventions Used in This Document 3
2.1. Terminology 4
2.2. Applicability Statement 4
2.3. Conventions 4
3. IPv6 Extension Headers 5
3.1. General Discussion 5
3.2. General Security Implications 6
3.3. Summary of Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers 6
3.4. Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers 7
3.5. Advice on the Handling of Packets with Unknown IPv6 Extension Headers 16
4. IPv6 Options 17
4.1. General Discussion 17
4.2. General Security Implications of IPv6 Options 17
4.3. Advice on the Handling of Packets with Specific IPv6 Options 17
4.4. Advice on the handling of Packets with Unknown IPv6 Options 29
5. IANA Considerations 29
6. Security Considerations 30
7. Acknowledgements 30
8. References 30
8.1. Normative References 30
8.2. Informative References 34
Authors' Addresses 35

1. Introduction

Recent studies (see e.g. [RFC7872]) suggest that there is widespread dropping of IPv6 packets that contain IPv6 Extension Headers (EHs). In some cases, such packet drops occur at transit routers. While some operators "officially" drop packets that contain IPv6 EHs, it is possible that some of the measured packet drops be the result of improper configuration defaults, or inappropriate advice in this area.

This document analyzes both the general security implications of IPv6 EHs and the specific security implications of each EH and Option type, and provides advice on the filtering of IPv6 packets based on the IPv6 EHs and the IPv6 options they contain. Since various protocols may use IPv6 EHs (possibly with IPv6 options), discarding packets based on the IPv6 EHs or IPv6 options they contain may have implications on the proper functioning of such protocols. Thus, this document also attempts to discuss the operational and interoperability implications of such filtering policies.

The filtering policy typically depends on where in the network such policy is enforced: when the policy is enforced in a transit network, the policy typically follows a "black-list" approach, where only packets with clear negative implications are dropped. On the other hand, when the policy is enforced closer to the destination systems, the policy typically follows a "white-list" approach, where only traffic that is expected to be received is allowed. The advice in this document is aimed only at transit routers that may need to enforce a filtering policy based on the EHs and IPv6 options a packet may contain, following a "black-list" approach, and hence is likely to be much more permissive that a filtering policy to be employed e.g. at the edge of an enterprise network. The advice in this document is meant to improve the current situation of the dropping of packets with IPv6 EHs in the Internet [RFC7872].

This document is similar in nature to [RFC7126], which addresses the same problem for the IPv4 case. However, in IPv6, the problem space is compounded by the fact that IPv6 specifies a number of IPv6 EHs, and a number of IPv6 options which may be valid only when included in specific EH types.

This document completes and complements the considerations for protecting the control plane from packets containing IP options that can be found in [RFC6192].

Section 2 of this document specifies the terminology and conventions employed throughout this document. Section 3 of this document discusses IPv6 EHs and provides advice in the area of filtering IPv6 packets that contain such IPv6 EHs. Section 4 of this document discusses IPv6 options and provides advice in the area of filtering IPv6 packets that contain such options.

2. Terminology and Conventions Used in This Document

2.1. Terminology

The terms "fast path", "slow path", and associated relative terms ("faster path" and "slower path") are loosely defined as in Section 2 of [RFC6398].

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

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

2.2. Applicability Statement

This document provides advice on the filtering of IPv6 packets with EHs at transit routers for traffic *not* explicitly destined to such transit routers, for those cases in which such filtering is deemed as necessary.

2.3. Conventions

This document assumes that nodes comply with the requirements in [RFC7045]. Namely (from [RFC7045]),

- If a forwarding node discards a packet containing a standard IPv6 EH, it MUST be the result of a configurable policy and not just the result of a failure to recognise such a header.

- The discard policy for each standard type of EH MUST be individually configurable.

- The default configuration SHOULD allow all standard IPv6 EHs.

The advice provided in this document is only meant to guide an operator in configuring forwarding devices, and is *not* to be interpreted as advice regarding default configuration settings for network devices. That is, this document provides advice with respect to operational configurations, but does not change the implementation defaults required by [RFC7045].

We recommend that configuration options are made available to govern the processing of each IPv6 EH type and each IPv6 option type. Such configuration options may include the following possible settings:

- Permit this IPv6 EH or IPv6 Option type
- Discard (and log) packets containing this IPv6 EH or option type
- Reject (and log) packets containing this IPv6 EH or option type (where the packet drop is signaled with an ICMPv6 error message)
- Rate-limit traffic containing this IPv6 EH or option type
- Ignore this IPv6 EH or option type (as if it was not present) and forward the packet. We note that if a packet carries forwarding information (e.g., in an IPv6 Routing Header) this might be an inappropriate or undesirable action.

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

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

3. IPv6 Extension Headers

3.1. General Discussion

IPv6 [RFC8200] EHs allow for the extension of the IPv6 protocol. Since both IPv6 EHs and upper-layer protocols share the same namespace ("Next Header" registry/namespace), [RFC7045] identifies which of the currently assigned Internet Protocol numbers identify IPv6 EHs vs. upper-layer protocols. This document discusses the filtering of packets based on the IPv6 EHs (as specified by [RFC7045]) they contain.

> NOTE: [RFC7112] specifies that non-fragmented IPv6 datagrams and IPv6 First-Fragments MUST contain the entire IPv6 header chain [RFC7112]. Therefore, intermediate systems can enforce the filtering policies discussed in this document, or resort to simply discarding the offending packets when they fail to comply with the requirements in [RFC7112]. We note that, in order to implement

filtering rules on the fast path, it may be necessary for the filtering device to limit the depth into the packet that can be inspected before giving up. In circumstances where there is such a limitation, it is recommended that implementations discard packets if, when trying to determine whether to discard or permit a packet, the aforementioned limit is encountered.

3.2. General Security Implications

In some specific device architectures, IPv6 packets that contain IPv6 EHs may cause the corresponding packets to be processed on the slow path, and hence may be leveraged for the purpose of Denial of Service (DoS) attacks [I-D.gont-v6ops-ipv6-ehs-packet-drops] [Cisco-EH] [FW-Benchmark].

Operators are urged to consider IPv6 EH filtering and IPv6 options handling capabilities of different devices as they make deployment decisions in future.

3.3. Summary of Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers

This section summarizes the advice provided in Section 3.4, providing references to the specific sections in which a detailed analysis can be found.

EH type	Filtering policy	Reference
IPv6 Hop-by-Hop Options (Proto=0)	Drop or Ignore	Section 3.4.1
Routing Header for IPv6 (Proto=43)	Drop only RTH0 and RTH1. Permit other RH Types	Section 3.4.2
Fragment Header for IPv6 (Proto=44)	Permit	Section 3.4.3
Encapsulating Security Payload (Proto=50)	Permit	Section 3.4.4
Authentication Header (Proto=51)	Permit	Section 3.4.5
Destination Options for IPv6 (Proto=60)	Permit	Section 3.4.6
Mobility Header (Proto=135)	Permit	Section 3.4.7
Host Identity Protocol (Proto=139)	Permit	Section 3.4.8
Shim6 Protocol (Proto=140)	Permit	Section 3.4.9
Use for experimentation and testing (Proto=253 and 254)	Drop	Section 3.4.10

Table 1: Summary of Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers

3.4. Advice on the Handling of IPv6 Packets with Specific IPv6 Extension Headers

3.4.1. IPv6 Hop-by-Hop Options (Protocol Number=0)

3.4.1.1. Uses

The Hop-by-Hop Options header is used to carry optional information that may be examined by every node along a packet's delivery path. It is expected that nodes will examine the Hop-by-Hop Options header if explicitly configured to do so.

NOTE: [RFC2460] required that all nodes examined and processed the Hop-by-Hop Options header. However, even before the publication of [RFC8200] a number of implementations already provided the option of ignoring this header unless explicitly configured to examine it.

3.4.1.2. Specification

This EH is specified in [RFC8200]. At the time of this writing, the following options have been specified for the Hop-by-Hop Options EH:

- Type 0x00: Pad1 [RFC8200]
- Type 0x01: PadN [RFC8200]
- Type 0x05: Router Alert [RFC2711]
- Type 0x07: CALIPSO [RFC5570]
- Type 0x08: SMF_DPD [RFC6621]
- Type 0x23: RPL Option [I-D.ietf-roll-useofrplinfo]
- Type 0x26: Quick-Start [RFC4782]
- Type 0x4D: (Deprecated)
- Type 0x63: RPL Option [RFC6553]
- Type 0x6D: MPL Option [RFC7731]
- Type 0x8A: Endpoint Identification (Deprecated) [draft-ietf-nimrod-eid]
- Type 0xC2: Jumbo Payload [RFC2675]
- Type 0xEE: IPv6 DFF Header [RFC6971]
- Type 0x1E: RFC3692-style Experiment [RFC4727]
- Type 0x3E: RFC3692-style Experiment [RFC4727]

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

3.4.1.3. Specific Security Implications

Legacy nodes that may process this extencion header could be subject to Denial of Service attacks.

NOTE: While [RFC8200] has removed this requirement, the deployed base may still reflect the traditional behavior for a while, and hence the potential security problems of this EH are still of concern.

3.4.1.4. Operational and Interoperability Impact if Blocked

Discarding packets containing a Hop-by-Hop Options EH would break any of the protocols that rely on it for proper functioning. For example, it would break RSVP [RFC2205] and multicast deployments, and would cause IPv6 jumbograms to be discarded.

3.4.1.5. Advice

Nodes implementing [RFC8200] would already ignore this extension header unless explicitly required to process it. For legacy ([RFC2460] nodes, the recommended configuration for the processing of these packets depends on the features and capabilities of the underlying platform. On platforms that allow forwarding of packets with HBH Options on the fast path, we recommend that packets with a HBH Options EH be forwarded as normal. Otherwise, on platforms in which processing of packets with a IPv6 HBH Options EH is carried out in the slow path, and an option is provided to rate-limit these packets, we recommend that this option be selected. Finally, when packets containing a HBH Options EH are processed in the slow-path, and the underlying platform does not have any mitigation options available for attacks based on these packets, we recommend that such platforms discard packets containing IPv6 HBH Options EHs.

Finally, we note that, for obvious reasons, RPL (Routing Protocol for Low-Power and Lossy Networks) [RFC6550] routers must not discard packets based on the presence of an IPv6 Hop-by-Hop Options EH.

3.4.2. Routing Header for IPv6 (Protocol Number=43)

3.4.2.1. Uses

The Routing header is used by an IPv6 source to list one or more intermediate nodes to be "visited" on the way to a packet's destination.

3.4.2.2. Specification

This EH is specified in [RFC8200]. [RFC2460] had originally specified the Routing Header Type 0, which was later obsoleted by [RFC5095], and thus removed from [RFC8200].

At the time of this writing, the following Routing Types have been specified:

- Type 0: Source Route (DEPRECATED) [RFC2460] [RFC5095]
- Type 1: Nimrod (DEPRECATED)
- Type 2: Type 2 Routing Header [RFC6275]
- Type 3: RPL Source Route Header [RFC6554]
- Types 4-252: Unassigned
- Type 253: RFC3692-style Experiment 1 [RFC4727]
- Type 254: RFC3692-style Experiment 2 [RFC4727]
- Type 255: Reserved

3.4.2.3. Specific Security Implications

The security implications of RHT0 have been discussed in detail in [Biondi2007] and [RFC5095].

3.4.2.4. Operational and Interoperability Impact if Blocked

Blocking packets containing a RHT0 or RTH1 has no operational implications, since both have been deprecated. However, blocking packets employing other routing header types will break the protocols that rely on them.

3.4.2.5. Advice

Intermediate systems should discard packets containing a RHT0 or RHT1. Other routing header types should be permitted, as required by [RFC7045].

3.4.3. Fragment Header for IPv6 (Protocol Number=44)

3.4.3.1. Uses

This EH provides the fragmentation functionality for IPv6.

3.4.3.2. Specification

This EH is specified in [RFC8200].

3.4.3.3. Specific Security Implications

The security implications of the Fragment Header range from Denial of Service attacks (e.g. based on flooding a target with IPv6 fragments) to information leakage attacks [RFC7739].

3.4.3.4. Operational and Interoperability Impact if Blocked

Blocking packets that contain a Fragment Header will break any protocol that may rely on fragmentation (e.g., the DNS [RFC1034]).

3.4.3.5. Advice

Intermediate systems should permit packets that contain a Fragment Header.

3.4.4. Encapsulating Security Payload (Protocol Number=50)

3.4.4.1. Uses

This EH is employed for the IPsec suite [RFC4303].

3.4.4.2. Specification

This EH is specified in [RFC4303].

3.4.4.3. Specific Security Implications

Besides the general implications of IPv6 EHs, this EH could be employed to potentially perform a DoS attack at the destination system by wasting CPU resources in validating the contents of the packet.

3.4.4.4. Operational and Interoperability Impact if Blocked

Discarding packets that employ this EH would break IPsec deployments.

3.4.4.5. Advice

Intermediate systems should permit packets containing the Encapsulating Security Payload EH.

3.4.5. Authentication Header (Protocol Number=51)

3.4.5.1. Uses

The Authentication Header can be employed for provide authentication services in IPv4 and IPv6.

3.4.5.2. Specification

This EH is specified in [RFC4302].

3.4.5.3. Specific Security Implications

Besides the general implications of IPv6 EHs, this EH could be employed to potentially perform a DoS attack at the destination system by wasting CPU resources in validating the contents of the packet.

3.4.5.4. Operational and Interoperability Impact if Blocked

Discarding packets that employ this EH would break IPsec deployments.

3.4.5.5. Advice

Intermediate systems should permit packets containing an Authentication Header.

3.4.6. Destination Options for IPv6 (Protocol Number=60)

3.4.6.1. Uses

The Destination Options header is used to carry optional information that needs be examined only by a packet’s destination node(s).

3.4.6.2. Specification

This EH is specified in [RFC8200]. At the time of this writing, the following options have been specified for this EH:

- Type 0x00: Pad1 [RFC8200]
- Type 0x01: PadN [RFC8200]
- Type 0x04: Tunnel Encapsulation Limit [RFC2473]
- Type 0x4D: (Deprecated)
- Type 0xC9: Home Address [RFC6275]
- Type 0x8A: Endpoint Identification (Deprecated) [draft-ietf-nimrod-eid]
- Type 0x8B: ILNP Nonce [RFC6744]
- Type 0x8C: Line-Identification Option [RFC6788]
- Type 0x1E: RFC3692-style Experiment [RFC4727]
- Type 0x3E: RFC3692-style Experiment [RFC4727]
- Type 0x5E: RFC3692-style Experiment [RFC4727]
- Type 0x7E: RFC3692-style Experiment [RFC4727]
- Type 0x9E: RFC3692-style Experiment [RFC4727]
- Type 0xBE: RFC3692-style Experiment [RFC4727]
- Type 0xDE: RFC3692-style Experiment [RFC4727]
- Type 0xFE: RFC3692-style Experiment [RFC4727]

3.4.6.3. Specific Security Implications

No security implications are known, other than the general implications of IPv6 EHs. For a discussion of possible security implications of specific options specified for the DO header, please see the Section 4.3.

3.4.6.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain a Destination Options header would break protocols that rely on this EH type for conveying information, including protocols such as ILNP [RFC6740] and Mobile IPv6 [RFC6275], and IPv6 tunnels that employ the Tunnel Encapsulation Limit option.

3.4.6.5. Advice

Intermediate systems should permit packets that contain a Destination Options Header.

3.4.7. Mobility Header (Protocol Number=135)

3.4.7.1. Uses

The Mobility Header is an EH used by mobile nodes, correspondent nodes, and home agents in all messaging related to the creation and management of bindings in Mobile IPv6.

3.4.7.2. Specification

This EH is specified in [RFC6275].

3.4.7.3. Specific Security Implications

A thorough security assessment of the security implications of the Mobility Header and related mechanisms can be found in Section 15 of [RFC6275].

3.4.7.4. Operational and Interoperability Impact if Blocked

Discarding packets containing this EH would break Mobile IPv6.

3.4.7.5. Advice

Intermediate systems should permit packets containing this EH.

3.4.8. Host Identity Protocol (Protocol Number=139)

3.4.8.1. Uses

This EH is employed with the Host Identity Protocol (HIP), an experimental protocol that allows consenting hosts to securely establish and maintain shared IP-layer state, allowing separation of the identifier and locator roles of IP addresses, thereby enabling continuity of communications across IP address changes.

3.4.8.2. Specification

This EH is specified in [RFC5201].

3.4.8.3. Specific Security Implications

The security implications of the HIP header are discussed in detail in Section 8 of [RFC6275].

3.4.8.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain the Host Identity Protocol would break HIP deployments.

3.4.8.5. Advice

Intermediate systems should permit packets that contain a Host Identity Protocol EH.

3.4.9. Shim6 Protocol (Protocol Number=140)

3.4.9.1. Uses

This EH is employed by the Shim6 [RFC5533] Protocol.

3.4.9.2. Specification

This EH is specified in [RFC5533].

3.4.9.3. Specific Security Implications

The specific security implications are discussed in detail in Section 16 of [RFC5533].

3.4.9.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain this EH will break Shim6.

3.4.9.5. Advice

Intermediate systems should permit packets containing this EH.

3.4.10. Use for experimentation and testing (Protocol Numbers=253 and 254)

3.4.10.1. Uses

These IPv6 EHs are employed for performing RFC3692-Style experiments (see [RFC3692] for details).

3.4.10.2. Specification

These EHs are specified in [RFC3692] and [RFC4727].

3.4.10.3. Specific Security Implications

The security implications of these EHs will depend on their specific use.

3.4.10.4. Operational and Interoperability Impact if Blocked

For obvious reasons, discarding packets that contain these EHs limits the ability to perform legitimate experiments across IPv6 routers.

3.4.10.5. Advice

Intermediate systems should discard packets containing these EHs. Only in specific scenarios in which RFC3692-Style experiments are to be performed should these EHs be permitted.

3.5. Advice on the Handling of Packets with Unknown IPv6 Extension Headers

We refer to IPv6 EHs that have not been assigned an Internet Protocol Number by IANA (and marked as such) in [IANA-PROTOCOLS] as "unknown IPv6 extension headers" ("unknown IPv6 EHs").

3.5.1. Uses

New IPv6 EHs may be specified as part of future extensions to the IPv6 protocol.

Since IPv6 EHs and Upper-layer protocols employ the same namespace, it is impossible to tell whether an unknown "Internet Protocol Number" is being employed for an IPv6 EH or an Upper-Layer protocol.

3.5.2. Specification

The processing of unknown IPv6 EHs is specified in [RFC8200] and [RFC7045].

3.5.3. Specific Security Implications

For obvious reasons, it is impossible to determine specific security implications of unknown IPv6 EHs. However, from security standpoint, a device should discard IPv6 extension headers for which the security implications cannot be determined. We note that this policy is allowed by [RFC7045].

3.5.4. Operational and Interoperability Impact if Blocked

As noted in [RFC7045], discarding unknown IPv6 EHs may slow down the deployment of new IPv6 EHs and transport protocols. The corresponding IANA registry ([IANA-PROTOCOLS]) should be monitored such that filtering rules are updated as new IPv6 EHs are standardized.

We note that since IPv6 EHs and upper-layer protocols share the same numbering space, discarding unknown IPv6 EHs may result in packets encapsulating unknown upper-layer protocols being discarded.

3.5.5. Advice

Intermediate systems should discard packets containing unknown IPv6 EHs.

4. IPv6 Options

4.1. General Discussion

The following subsections describe specific security implications of different IPv6 options, and provide advice regarding filtering packets that contain such options.

4.2. General Security Implications of IPv6 Options

The general security implications of IPv6 options are closely related to those discussed in Section 3.2 for IPv6 EHs. Essentially, packets that contain IPv6 options might need to be processed by an IPv6 router's general-purpose CPU,and hence could present a DDoS risk to that router's general-purpose CPU (and thus to the router itself). For some architectures, a possible mitigation would be to rate-limit the packets that are to be processed by the general-purpose CPU (see e.g. [Cisco-EH]).

4.3. Advice on the Handling of Packets with Specific IPv6 Options

The following subsections contain a description of each of the IPv6 options that have so far been specified, a summary of the security implications of each of such options, a discussion of possible interoperability implications if packets containing such options are discarded, and specific advice regarding whether packets containing these options should be permitted.

4.3.1. Pad1 (Type=0x00)

4.3.1.1. Uses

This option is used when necessary to align subsequent options and to pad out the containing header to a multiple of 8 octets in length.

4.3.1.2. Specification

This option is specified in [RFC8200].

4.3.1.3. Specific Security Implications

None.

4.3.1.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain this option would potentially break any protocol that relies on IPv6 EHs.

4.3.1.5. Advice

Intermediate systems should not discard packets based on the presence of this option.

4.3.2. PadN (Type=0x01)

4.3.2.1. Uses

This option is used when necessary to align subsequent options and to pad out the containing header to a multiple of 8 octets in length.

4.3.2.2. Specification

This option is specified in [RFC8200].

4.3.2.3. Specific Security Implications

Because of the possible size of this option, it could be leveraged as a large-bandwidth covert channel.

4.3.2.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain this option would potentially break any protocol that relies on IPv6 EHs.

4.3.2.5. Advice

Intermediate systems should not discard IPv6 packets based on the presence of this option.

4.3.3. Jumbo Payload (Type=0XC2)

4.3.3.1. Uses

The Jumbo payload option provides the means of specifying payloads larger than 65535 bytes.

4.3.3.2. Specification

This option is specified in [RFC2675].

4.3.3.3. Specific Security Implications

There are no specific issues arising from this option, except for improper validity checks of the option and associated packet lengths.

4.3.3.4. Operational and Interoperability Impact if Blocked

Discarding packets based on the presence of this option will cause IPv6 jumbograms to be discarded.

4.3.3.5. Advice

Intermediate systems should discard packets that contain this option. An operator should permit this option only in specific scenarios in which support for IPv6 jumbograms is desired.

4.3.4. RPL Option (Type=0x63)

4.3.4.1. Uses

The RPL Option provides a mechanism to include routing information with each datagram that an RPL router forwards.

4.3.4.2. Specification

This option was originally specified in [RFC6553]. It has been deprecated by [I-D.ietf-roll-useofrplinfo].

4.3.4.3. Specific Security Implications

Those described in [RFC6553].

4.3.4.4. Operational and Interoperability Impact if Blocked

This option is meant to be employed within an RPL instance. As a result, discarding packets based on the presence of this option (e.g. at an ISP) will not result in interoperability implications.

4.3.4.5. Advice

Non-RPL routers should discard packets that contain an RPL option.

4.3.5. RPL Option (Type=0x23)

4.3.5.1. Uses

The RPL Option provides a mechanism to include routing information with each datagram that an RPL router forwards.

4.3.5.2. Specification

This option is specified in [I-D.ietf-roll-useofrplinfo].

4.3.5.3. Specific Security Implications

Those described in [I-D.ietf-roll-useofrplinfo].

4.3.5.4. Operational and Interoperability Impact if Blocked

This option is meant to survive outside of an RPL instance. As a result, discarding packets based on the presence of this option would break some use cases for RPL (see [I-D.ietf-roll-useofrplinfo]).

4.3.5.5. Advice

Intermediate systems should not discard IPv6 packets based on the presence of this option.

4.3.6. Tunnel Encapsulation Limit (Type=0x04)

4.3.6.1. Uses

The Tunnel Encapsulation Limit option can be employed to specify how many further levels of nesting the packet is permitted to undergo.

4.3.6.2. Specification

This option is specified in [RFC2473].

4.3.6.3. Specific Security Implications

Those described in [RFC2473].

4.3.6.4. Operational and Interoperability Impact if Blocked

Discarding packets based on the presence of this option could result in tunnel traffic being discarded.

4.3.6.5. Advice

Intermediate systems should not discard packets based on the presence of this option.

4.3.7. Router Alert (Type=0x05)

4.3.7.1. Uses

The Router Alert option [RFC2711] is typically employed for the RSVP protocol [RFC2205] and the MLD protocol [RFC2710].

4.3.7.2. Specification

This option is specified in [RFC2711].

4.3.7.3. Specific Security Implications

Since this option causes the contents of the packet to be inspected by the handling device, this option could be leveraged for performing DoS attacks.

4.3.7.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain this option would break RSVP and multicast deployments.

4.3.7.5. Advice

Intermediate systems should discard packets that contain this option. Only in specific environments where support for RSVP, multicast routing, or similar protocols is desired, should this option be permitted.

4.3.8. Quick-Start (Type=0x26)

4.3.8.1. Uses

This IP Option is used in the specification of Quick-Start for TCP and IP, which is an experimental mechanism that allows transport protocols, in cooperation with routers, to determine an allowed sending rate at the start and, at times, in the middle of a data transfer (e.g., after an idle period) [RFC4782].

4.3.8.2. Specification

This option is specified in [RFC4782], on the "Experimental" track.

4.3.8.3. Specific Security Implications

Section 9.6 of [RFC4782] notes that Quick-Start is vulnerable to two kinds of attacks:

- attacks to increase the routers' processing and state load, and,
- attacks with bogus Quick-Start Requests to temporarily tie up available Quick-Start bandwidth, preventing routers from approving Quick-Start Requests from other connections.

We note that if routers in a given environment do not implement and enable the Quick-Start mechanism, only the general security implications of IP options (discussed in Section 4.2) would apply.

4.3.8.4. Operational and Interoperability Impact if Blocked

The Quick-Start functionality would be disabled, and additional delays in TCP's connection establishment (for example) could be introduced. (Please see Section 4.7.2 of [RFC4782].) We note, however, that Quick-Start has been proposed as a mechanism that could be of use in controlled environments, and not as a mechanism that would be intended or appropriate for ubiquitous deployment in the global Internet [RFC4782].

4.3.8.5. Advice

Intermediate systems should not discard IPv6 packets based on the presence of this option.

4.3.9. CALIPSO (Type=0x07)

4.3.9.1. Uses

This option is used for encoding explicit packet Sensitivity Labels on IPv6 packets. It is intended for use only within Multi-Level Secure (MLS) networking environments that are both trusted and trustworthy.

4.3.9.2. Specification

This option is specified in [RFC5570].

4.3.9.3. Specific Security Implications

Presence of this option in a packet does not by itself create any specific new threat. Packets with this option ought not normally be seen on the global public Internet.

4.3.9.4. Operational and Interoperability Impact if Blocked

If packets with this option are discarded or if the option is stripped from the packet during transmission from source to destination, then the packet itself is likely to be discarded by the receiver because it is not properly labeled. In some cases, the receiver might receive the packet but associate an incorrect sensitivity label with the received data from the packet whose CALIPSO was stripped by an intermediate router or firewall. Associating an incorrect sensitivity label can cause the received information either to be handled as more sensitive than it really is ("upgrading") or as less sensitive than it really is ("downgrading"), either of which is problematic.

4.3.9.5. Advice

Intermediate systems that do not operate in Multi-Level Secure (MLS) networking environments should discard packets that contain this option.

4.3.10. SMF_DPD (Type=0x08)

4.3.10.1. Uses

This option is employed in the (experimental) Simplified Multicast Forwarding (SMF) for unique packet identification for IPv6 I-DPD, and as a mechanism to guarantee non-collision of hash values for different packets when H-DPD is used.

4.3.10.2. Specification

This option is specified in [RFC6621].

4.3.10.3. Specific Security Implications

None. The use of identifiers is subject to the security and privacy considerations discussed in [I-D.gont-predictable-numeric-ids].

4.3.10.4. Operational and Interoperability Impact if Blocked

Dropping packets containing this option within a MANET domain would break SMF. However, dropping such packets at the border of such domain would have no negative impact.

4.3.10.5. Advice

Intermediate system should discard packets that contain this option.

4.3.11. Home Address (Type=0xC9)

4.3.11.1. Uses

The Home Address option is used by a Mobile IPv6 node while away from home, to inform the recipient of the mobile node’s home address.

4.3.11.2. Specification

This option is specified in [RFC6275].

4.3.11.3. Specific Security Implications

No (known) additional security implications than those described in [RFC6275].

4.3.11.4. Operational and Interoperability Impact if Blocked

Discarding IPv6 packets based on the presence of this option will break Mobile IPv6.

4.3.11.5. Advice

Intermediate systems should not discard IPv6 packets based on the presence of this option.

4.3.12. Endpoint Identification (Type=0x8A)

4.3.12.1. Uses

The Endpoint Identification option was meant to be used with the Nimrod routing architecture [NIMROD-DOC], but has never seen widespread deployment.

4.3.12.2. Specification

This option is specified in [NIMROD-DOC].

4.3.12.3. Specific Security Implications

Undetermined.

4.3.12.4. Operational and Interoperability Impact if Blocked

None.

4.3.12.5. Advice

Intermediate systems should discard packets that contain this option.

4.3.13. ILNP Nonce (Type=0x8B)

4.3.13.1. Uses

This option is employed by Identifier-Locator Network Protocol for IPv6 (ILNPv6) for providing protection against off-path attacks for packets when ILNPv6 is in use, and as a signal during initial network-layer session creation that ILNPv6 is proposed for use with this network-layer session, rather than classic IPv6.

4.3.13.2. Specification

This option is specified in [RFC6744].

4.3.13.3. Specific Security Implications

Those described in [RFC6744].

4.3.13.4. Operational and Interoperability Impact if Blocked

Discarding packets that contain this option will break INLPv6 deployments.

4.3.13.5. Advice

Intermediate systems should not discard packets based on the presence of this option.

4.3.14. Line-Identification Option (Type=0x8C)

4.3.14.1. Uses

This option is used by an Edge Router to identify the subscriber premises in scenarios where several subscriber premises may be logically connected to the same interface of an Edge Router.

4.3.14.2. Specification

This option is specified in [RFC6788].

4.3.14.3. Specific Security Implications

Those described in [RFC6788].

4.3.14.4. Operational and Interoperability Impact if Blocked

Since this option is meant to be employed in Router Solicitation messages, discarding packets based on the presence of this option at intermediate systems will result in no interoperability implications.

4.3.14.5. Advice

Intermediate devices should discard packets that contain this option.

4.3.15. Deprecated (Type=0x4D)

4.3.15.1. Uses

No information has been found about this option type.

4.3.15.2. Specification

No information has been found about this option type.

4.3.15.3. Specific Security Implications

No information has been found about this option type, and hence it has been impossible to perform the corresponding security assessment.

4.3.15.4. Operational and Interoperability Impact if Blocked

Unknown.

4.3.15.5. Advice

Intermediate systems should discard packets that contain this option.

4.3.16. MPL Option (Type=0x6D)

4.3.16.1. Uses

This option is used with the Multicast Protocol for Low power and Lossy Networks (MPL), that provides IPv6 multicast forwarding in constrained networks.

4.3.16.2. Specification

This option is specified in [RFC7731], and is meant to be included only in Hop-by-Hop Option headers.

4.3.16.3. Specific Security Implications

Those described in [RFC7731].

4.3.16.4. Operational and Interoperability Impact if Blocked

Dropping packets that contain an MPL option within an MPL network would break the Multicast Protocol for Low power and Lossy Networks (MPL). However, dropping such packets at the border of such networks will have no negative impact.

4.3.16.5. Advice

Intermediate systems should not discard packets based on the presence of this option. However, since this option has been specified for the Hop-by-Hop Options, such systems should consider the discussion in Section 3.4.1.

4.3.17. IP_DFF (Type=0xEE)

4.3.17.1. Uses

This option is employed with the (Experimental) Depth-First Forwarding (DFF) in Unreliable Networks.

4.3.17.2. Specification

This option is specified in [RFC6971].

4.3.17.3. Specific Security Implications

Those specified in [RFC6971].

4.3.17.4. Operational and Interoperability Impact if Blocked

Dropping packets containing this option within a routing domain that is running DFF would break DFF. However, droping such packets at the border of such domains will have no security implications.

4.3.17.5. Advice

Intermediate systems that do not operate within a routing domain that is running DFF should discard packets containing this option.

4.3.18. RFC3692-style Experiment (Types = 0x1E, 0x3E, 0x5E, 0x7E, 0x9E, 0xBE, 0xDE, 0xFE)

4.3.18.1. Uses

These options can be employed for performing RFC3692-style experiments. It is only appropriate to use these values in explicitly configured experiments; they must not be shipped as defaults in implementations.

4.3.18.2. Specification

Specified in RFC 4727 [RFC4727] in the context of RFC3692-style experiments.

4.3.18.3. Specific Security Implications

The specific security implications will depend on the specific use of these options.

4.3.18.4. Operational and Interoperability Impact if Blocked

For obvious reasons, discarding packets that contain these options limits the ability to perform legitimate experiments across IPv6 routers.

4.3.18.5. Advice

Intermediate systems should discard packets that contain these options. Only in specific environments where RFC3692-style experiments are meant to be performed should these options be permitted.

4.4. Advice on the handling of Packets with Unknown IPv6 Options

We refer to IPv6 options that have not been assigned an IPv6 option type in the corresponding registry ([IANA-IPV6-PARAM]) as "unknown IPv6 options".

4.4.1. Uses

New IPv6 options may be specified as part of future protocol work.

4.4.2. Specification

The processing of unknown IPv6 options is specified in [RFC8200].

4.4.3. Specific Security Implications

For obvious reasons, it is impossible to determine specific security implications of unknown IPv6 options.

4.4.4. Operational and Interoperability Impact if Blocked

Discarding unknown IPv6 options may slow down the deployment of new IPv6 options. As noted in [draft-gont-6man-ipv6-opt-transmit], the corresponding IANA registry ([IANA-IPV6-PARAM] should be monitored such that IPv6 option filtering rules are updated as new IPv6 options are standardized.

4.4.5. Advice

Enterprise intermediate systems that process the contents of IPv6 EHs should discard packets that contain unknown options. Other intermediate systems that process the contents of IPv6 EHs should permit packets that contain unknown options.

5. IANA Considerations

This document has no actions for IANA.

6. Security Considerations

This document provides advice on the filtering of IPv6 packets that contain IPv6 EHs (and possibly IPv6 options) at IPv6 transit routers. It is meant to improve the current situation of widespread dropping of such IPv6 packets in those cases where the drops result from improper configuration defaults, or inappropriate advice in this area.

7. Acknowledgements

The authors would like to thank Ron Bonica for his work on earlier versions of this document.

The authors of this document would like to thank (in alphabetical order) Mikael Abrahamsson, Brian Carpenter, Darren Dukes, Mike Heard, Bob Hinden, Jen Linkova, Carlos Pignataro, Maria Ines Robles, Donald Smith, Pascal Thubert, Ole Troan, Gunter Van De Velde, and Eric Vyncke, for providing valuable comments on earlier versions of this document.

This document borrows some text and analysis from [RFC7126], authored by Fernando Gont, Randall Atkinson, and Carlos Pignataro.

Fernando Gont would like to thank Eric Vyncke for his guidance.

8. References

8.1. Normative References

[draft-gont-6man-ipv6-opt-transmit]
Gont, F., Liu, W., and R. Bonica, "Transmission and Processing of IPv6 Options", IETF Internet Draft, work in progress, August 2014.

[I-D.ietf-roll-useofrplinfo]
Robles, I., Richardson, M., and P. Thubert, "When to use RFC 6553, 6554 and IPv6-in-IPv6", draft-ietf-roll-useofrplinfo-23 (work in progress), May 2018.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

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

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <https://www.rfc-editor.org/info/rfc2205>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <https://www.rfc-editor.org/info/rfc2460>.

[RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <https://www.rfc-editor.org/info/rfc2473>.

[RFC2675] Borman, D., Deering, S., and R. Hinden, "IPv6 Jumbograms", RFC 2675, DOI 10.17487/RFC2675, August 1999, <https://www.rfc-editor.org/info/rfc2675>.

[RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, DOI 10.17487/RFC2710, October 1999, <https://www.rfc-editor.org/info/rfc2710>.

[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>.

[RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, DOI 10.17487/RFC3692, January 2004, <https://www.rfc-editor.org/info/rfc3692>.

[RFC4302] Kent, S., "IP Authentication Header", RFC 4302, DOI 10.17487/RFC4302, December 2005, <https://www.rfc-editor.org/info/rfc4302>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

[RFC4304] Kent, S., "Extended Sequence Number (ESN) Addendum to IPsec Domain of Interpretation (DOI) for Internet Security Association and Key Management Protocol (ISAKMP)", RFC 4304, DOI 10.17487/RFC4304, December 2005, <https://www.rfc-editor.org/info/rfc4304>.

[RFC4727] Fenner, B., "Experimental Values In IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers", RFC 4727, DOI 10.17487/RFC4727, November 2006, <https://www.rfc-editor.org/info/rfc4727>.

[RFC4782] Floyd, S., Allman, M., Jain, A., and P. Sarolahti, "Quick-Start for TCP and IP", RFC 4782, DOI 10.17487/RFC4782, January 2007, <https://www.rfc-editor.org/info/rfc4782>.

[RFC5095] Abley, J., Savola, P., and G. Neville-Neil, "Deprecation of Type 0 Routing Headers in IPv6", RFC 5095, DOI 10.17487/RFC5095, December 2007, <https://www.rfc-editor.org/info/rfc5095>.

[RFC5201] Moskowitz, R., Nikander, P., Jokela, P., Ed., and T. Henderson, "Host Identity Protocol", RFC 5201, DOI 10.17487/RFC5201, April 2008, <https://www.rfc-editor.org/info/rfc5201>.

[RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, DOI 10.17487/RFC5533, June 2009, <https://www.rfc-editor.org/info/rfc5533>.

[RFC5570] StJohns, M., Atkinson, R., and G. Thomas, "Common Architecture Label IPv6 Security Option (CALIPSO)", RFC 5570, DOI 10.17487/RFC5570, July 2009, <https://www.rfc-editor.org/info/rfc5570>.

[RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <https://www.rfc-editor.org/info/rfc6275>.

[RFC6398] Le Faucheur, F., Ed., "IP Router Alert Considerations and Usage", BCP 168, RFC 6398, DOI 10.17487/RFC6398, October 2011, <https://www.rfc-editor.org/info/rfc6398>.

[RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <https://www.rfc-editor.org/info/rfc6550>.

[RFC6553] Hui, J. and JP. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams", RFC 6553, DOI 10.17487/RFC6553, March 2012, <https://www.rfc-editor.org/info/rfc6553>.

[RFC6554] Hui, J., Vasseur, JP., Culler, D., and V. Manral, "An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL)", RFC 6554, DOI 10.17487/RFC6554, March 2012, <https://www.rfc-editor.org/info/rfc6554>.

[RFC6621] Macker, J., Ed., "Simplified Multicast Forwarding", RFC 6621, DOI 10.17487/RFC6621, May 2012, <https://www.rfc-editor.org/info/rfc6621>.

[RFC6740] Atkinson, RJ. and SN. Bhatti, "Identifier-Locator Network Protocol (ILNP) Architectural Description", RFC 6740, DOI 10.17487/RFC6740, November 2012, <https://www.rfc-editor.org/info/rfc6740>.

[RFC6744] Atkinson, RJ. and SN. Bhatti, "IPv6 Nonce Destination Option for the Identifier-Locator Network Protocol for IPv6 (ILNPv6)", RFC 6744, DOI 10.17487/RFC6744, November 2012, <https://www.rfc-editor.org/info/rfc6744>.

[RFC6788] Krishnan, S., Kavanagh, A., Varga, B., Ooghe, S., and E. Nordmark, "The Line-Identification Option", RFC 6788, DOI 10.17487/RFC6788, November 2012, <https://www.rfc-editor.org/info/rfc6788>.

[RFC6971] Herberg, U., Ed., Cardenas, A., Iwao, T., Dow, M., and S. Cespedes, "Depth-First Forwarding (DFF) in Unreliable Networks", RFC 6971, DOI 10.17487/RFC6971, June 2013, <https://www.rfc-editor.org/info/rfc6971>.

[RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <https://www.rfc-editor.org/info/rfc7045>.

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

[RFC7731] Hui, J. and R. Kelsey, "Multicast Protocol for Low-Power and Lossy Networks (MPL)", RFC 7731, DOI 10.17487/RFC7731, February 2016, <https://www.rfc-editor.org/info/rfc7731>.

[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>.

8.2. Informative References

[Biondi2007]
Biondi, P. and A. Ebalard, "IPv6 Routing Header Security", CanSecWest 2007 Security Conference, 2007, <http://www.secdev.org/conf/IPv6_RH_security-csw07.pdf>.

[Cisco-EH]
Cisco Systems, "IPv6 Extension Headers Review and Considerations", Whitepaper. October 2006, <http://www.cisco.com/en/US/technologies/tk648/tk872/technologies_white_paper0900aecd8054d37d.pdf>.

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

[FW-Benchmark]
Zack, E., "Firewall Security Assessment and Benchmarking IPv6 Firewall Load Tests", IPv6 Hackers Meeting #1, Berlin, Germany. June 30, 2013, <http://www.ipv6hackers.org/meetings/ipv6-hackers-1/zack-ipv6hackers1-firewall-security-assessment-and-benchmarking.pdf>.

[I-D.gont-predictable-numeric-ids]
Gont, F. and I. Arce, "Security and Privacy Implications of Numeric Identifiers Employed in Network Protocols", draft-gont-predictable-numeric-ids-02 (work in progress), February 2018.

[I-D.gont-v6ops-ipv6-ehs-packet-drops]
Gont, F., Hilliard, N., Doering, G., (Will), S., and W. Kumari, "Operational Implications of IPv6 Packets with Extension Headers", draft-gont-v6ops-ipv6-ehs-packet-drops-03 (work in progress), March 2016.

[I-D.ietf-6man-hbh-header-handling]
Baker, F. and R. Bonica, "IPv6 Hop-by-Hop Options Extension Header", draft-ietf-6man-hbh-header-handling-03 (work in progress), March 2016.

[IANA-IPV6-PARAM]
Internet Assigned Numbers Authority, "Internet Protocol Version 6 (IPv6) Parameters", December 2013, <http://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml>.

[IANA-PROTOCOLS]
Internet Assigned Numbers Authority, "Protocol Numbers", 2014, <http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>.

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

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

[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>.

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

[RFC7739] Gont, F., "Security Implications of Predictable Fragment Identification Values", RFC 7739, DOI 10.17487/RFC7739, February 2016, <https://www.rfc-editor.org/info/rfc7739>.

[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>.

Authors' Addresses

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

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

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

Email: liushucheng@huawei.com

Internet Engineering Task Force
Internet-Draft
Updates: 4861,6275 (if approved)
Intended status: Experimental
Expires: September 28, 2017

E. Kline
Google Japan KK
M. Abrahamsson
T-Systems Nordic

March 27, 2017

IPv6 Router Advertisement Prefix Information Option eXclusive Flag
draft-pioxfolks-6man-pio-exclusive-bit-02

Abstract

This document defines a new control bit in the IPv6 RA PIO flags octet that indicates that the node receiving this RA is the exclusive receiver of all traffic destined to any address within that prefix.

Termed the eXclusive flag (or "X flag"), nodes that recognize this can perform some optimizations to save time and traffic (e.g. disable ND and DAD for addresses within this prefix) and more immediately pursue the benefits of being provided multiple addresses (vis. [RFC7934] section 3). Additionally, network infrastructure nodes (routers, switches) can benefit by minimizing the number of {link layer, IP} address pairs required to offer network connectivity (vis. [RFC7934] section 9.3).

Use of the X flag is backward compatible with existing IPv6 standards compliant implementations.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 28, 2017.

Copyright Notice

Table of Contents

1. Introduction 3
2. Motivation 3
2.1. Efficiency improvements 4
2.2. New architectural possibilities 4
3. Applicability statement 5
4. Terminology 6
4.1. Requirements Language 6
4.2. Abbreviations 6
4.2.1. PIO-X 6
4.2.2. PIO-X RA 6
4.2.3. Host 6
5. Updated Prefix Information Option 6
5.1. Updated format description 6
5.2. Receiver processing 7
5.2.1. PIO R flag 7
5.2.2. (Re)Interpretation of other flags 7
5.2.2.1. PIO L flag 8
5.2.2.2. PIO A flag 8
5.3. Sender requirements 8
5.4. Comparison with DHCPv6 PD 9
6. Host behavior 9
6.1. PIO-X processing 9
6.2. Neighbor Discovery implications 9
6.2.1. Duplicate Address Detection (DAD) 9
6.2.2. Router Solicitations (RSes) 10
6.3. Link-local address behavior 10
6.4. Source address selection 10
6.5. Next hop router selection 11
6.6. Implications for Detecting Network Attachment 11
6.7. Additional guidance 11
7. Router behavior 11
7.1. PIO-X RA destination address 11
7.2. Detecting hosts to send PIO-X RAs to 12

7.3. Binding table requirements 12
7.4. Preparations before sending a PIO-X RA 13
7.5. Implementation considerations 13
8. Acknowledgements 13
9. IANA Considerations 13
10. Security Considerations 13
11. References 14
11.1. Normative References 14
11.2. Informative References 14
Authors' Addresses 15

1. Introduction

This document defines a new control flag in the Internet Protocol version 6 (IPv6) Router Advertisement (RA) Prefix Information Option (PIO) flags octet that indicates that the node receiving this RA is the exclusive receiver of all traffic destined to any address with that prefix. Subject to the lifetime constraints within the PIO, the receiving node effectively has exclusive use of the prefix, and will be the next hop destination for the sending router, and possibly other routers, for all traffic destined toward the prefix.

Termed the eXclusive flag (or "X flag"), nodes that recognize this can perform some optimizations to save time and traffic (e.g. disable Neighbor Discovery (ND) and Duplicate Address Detection (DAD) for addresses within this prefix) and more immediately pursue the benefits of being provided multiple addresses (vis. [RFC7934] section 3).

Additionally, network infrastructure nodes (routers, switches) can benefit by minimizing the number of {link layer, IP} address pairs required to offer network connectivity (vis. [RFC7934] section 9.3). A router, for example, need not create any {link layer, IP} address pair entries for IP address within a proffered exclusive-use prefix--it can reliably forward all traffic to the network node to which it advertised the prefix. This solves one potential link layer state exhaustion problem, i.e excessive number of {link layer, IP address pairs}, using IP layer forwarding.

Use of the X flag is backward compatible with existing IPv6 standards compliant implementations. [RFC4861]-compliant nodes that do not understand the X flag are not negatively impacted. They must ignore it, and can process the PIO under existing standards, making use of the information exactly as if the X flag were not set.

2. Motivation

This work is motivated by the pursuit of two categories of benefits: modest host and network side improvements in efficiency, and support for new deployment architectures and address space use models.

2.1. Efficiency improvements

If a host knows it has exclusive use of a prefix it can perform some optimizations to save time and traffic. It can avoid ND on the receiving interface for addresses within these prefixes. Network interfaces can even drop Neighbor Solicitations for these addresses on the receiving interface to save power by not waking up more power-hungry CPUs.

Additionally, a host can save time by not performing DAD for addresses within an exclusive-use prefix on the receiving interface. A host that wanted, for example, to use 2**64 unique IPv6 source addresses for DNS queries in order to improve resilience against forged answers (as recommended in section 9.2 of [1]), could do so without delaying each query from a newly formed address. A node could in theory implement the same strategy using Optimistic Duplicate Address Detection [2], but it could be very unfriendly to the network infrastructure (in terms of {link-layer, IP address} pair state) to do so without this kind of explicit signal.

A host that recognizes the X flag might perform other traffic-saving optimizations, like not attempt Multicast DNS in some cases, or avoid trying to register addresses with sleep proxies. Being the only host on this link these may be of little benefit.

2.2. New architectural possibilities

There are several initiatives that propose network side practices that provide customer isolation, enhanced operational scalability, power efficiency, security and other benefits in IPv6 network deployments. Some of these involve isolating a host (or RA accepting client node) so that the host is the only node to receive a specific prefix, including

- DHCPv6 Prefix Delegation to hosts ([3]), and
- advertising a unique prefix per host via unique RAs. ([4]).

Some architectures further isolate the host layers below IPv6, for improved client node security.

Regardless of the specific level of isolation, the host can best make choices about its use of a prefix exclusively forwarded to itself if the host can be informed of the exclusivity. (In the case of a

DHCPv6 Prefix Delegation the prefix can be assumed to be of exclusive use by the requesting node, in accordance with the model in [RFC3633].) An implementation can, for example, safely "bind to an IPv6 subnet" in the style of [5], or start 64sharing [6] (given a prefix of sufficient size).

This memo documents an additional flag in the IPv6 RA PIO that makes this information explicit to receiving node.

3. Applicability statement

Use of the X flag in PIOs is only applicable to networks where the architecture (i.e. serving infrastructure like routers, link-layer equipment, et cetera) can collectively guarantee the following criteria are met:

1. an RA containing a PIO with the X flag set MUST be delivered to one and only target node (host) such that no two nodes can reasonably expect exclusive access to the same prefix at the same time

2. any router advertising an RA containing a PIO with the X flag set SHOULD be notified quickly when a node leaves the network

The first criterion ensures that the same exclusive use prefix is not advertised to more than one host at a time (and hence no longer "exclusive"). This implies that an allocated exclusive-use prefix must be tracked by the issuing router for at least the minimum of (a) the lifetime of the recipient node’s continuous attachment to the network and (b) the lifetime of the prefix itself in the PIO, if not longer.

The second criterion aims to help the prefix allocation infrastructure reclaim unused prefixes quickly while also helping routers drop (possibly with appropriate ICMPv6 errors) traffic that can no longer be delivered.

It is expected that in practice this primarily describes networks where the IPv6 infrastructure and the link-layer have a tight integration. All point-to-point links meet these criteria (e.g. PPPoE and VPNs), as does the 3GPP architecture [RFC7066] and some IEEE 802.11 deployment architectures ([7]).

4. Terminology

4.1. Requirements Language

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

4.2. Abbreviations

Throughout this document the following terminology is used purely for the sake of brevity.

4.2.1. PIO-X

The term "PIO-X" is used to refer to a Prefix Information Option (PIO) that has the X flag set.

4.2.2. PIO-X RA

The phrase "PIO-X RA" is used to refer to an IPv6 Router Advertisement (RA) that contains one or more PIO-X entries (the same RA may also contain one or more PIOs without the X flag set).

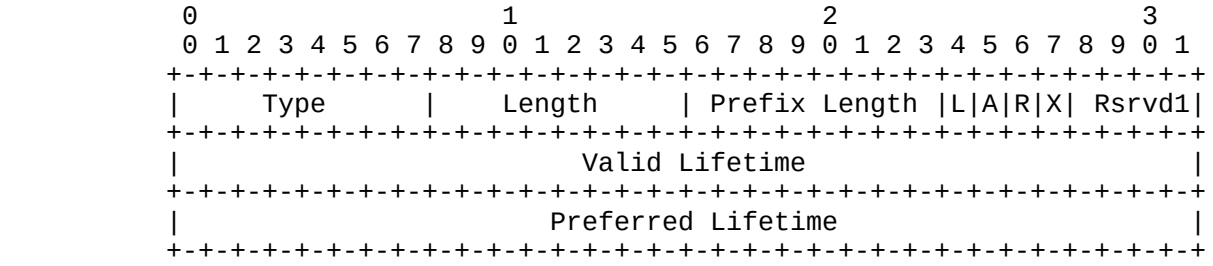
4.2.3. Host

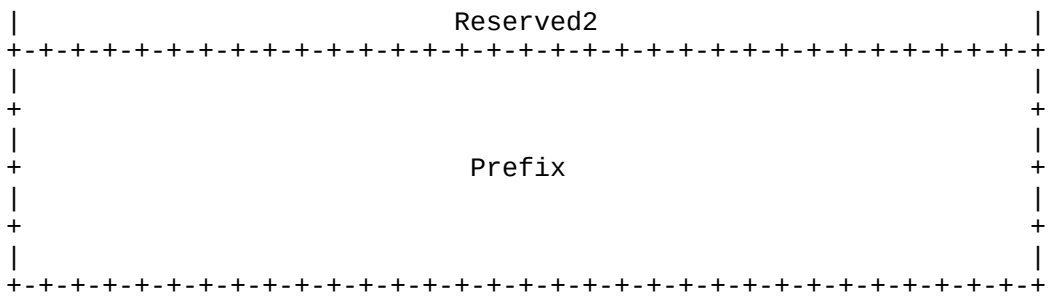
The term "host" may be used interchangeably throughout this document to mean a network node receiving and processing an RA. The receiving node may itself be a router, or may temporarily become one by routing all or a portion of an exclusive use prefix.

5. Updated Prefix Information Option

This document updates the Prefix Information Option specification in RFC 4861 [8] section 4.6.2 and RFC 6275 [9] section 7.2 with the definition of a flag from the former Reserved1 field as follows.

5.1. Updated format description





Fields:

X The eXclusive use indicator flag, defined by this document. When set, the receiving node can be assured that all traffic destined to any address within the specified Prefix will be forwarded to itself by, at a minimum, the router from which the encapsulating RA was received, but possibly other routers as well.

When not set, the receiving node MUST NOT make any assumptions of exclusive use of the specified Prefix, i.e. processing is unchanged from previous standards behavior.

Rsrvd1 Retains the same meaning as Reserved1 from [10] section 4.6.2.

All other fields Retain their same meaning from [11] section 4.6.2.

5.2. Receiver processing

Nodes compliant with this specification perform the following additional processing of RAs and PIO-X options when a PIO-X option is present.

5.2.1. PIO R flag

If the R flag is set then the X flag MUST be ignored. The R flag indicates that the PIO includes an address the router has selected for itself from the prefix. Logically, the prefix cannot exclusively be used by the receiving node if the router has allocated any addresses for itself from the prefix.

5.2.2. (Re)Interpretation of other flags

Nodes compliant with this specification, i.e. those that understand the X-flag, MUST, when the X-flag is set, ignore the actual values of the L and A flags and instead interpret them as follows:

- interpret the L flag as if it were 0 (L=0)
- interpret the A flag as it it were 1 (A=1)

The rationale for this is as follows.

5.2.2.1. PIO L flag

Because a PIO-X aware node will know that it has exclusive use of a prefix with non-zero valid lifetime, the prefix itself cannot be considered to be on-link with respect to the link on which the PIO-X RA was received.

Note that a given address from within the prefix may be considered on-link according to the definition in [12] section 4, item 1, should the receiving node chose to configure that address on said link, but this is in no way synonymous with the entire prefix being considered on-link.

5.2.2.2. PIO A flag

Because a PIO-X aware node will know that it has exclusive use of a prefix with non-zero valid lifetime, autoconfiguration of addresses according to any desired scheme, e.g. [13], [14], et cetera, is implicit in the setting of the X flag.

Accordingly, the A flag can be interpreted as having been set, should the host choose to apply standard address generation schemes that require the flag to be set. It is free to assign any address formed from an exclusive prefix to any available interface; it is not required to configure the address on the link over which the PIO-X RA was received (i.e. it is under no obligation to form addresses such that they would be classified as on-link (according to the definition in [15] section 4, item 1).

5.3. Sender requirements

When a router transmits an RA containing one or more PIO-X options it SHOULD unicast the PIO-X RA to its intended recipient at the IPv6 layer and, if applicable, at the link-layer.

It is RECOMMENDED that a PIO with the X-flag set also have the PIO flags L=0 and A=1 explicitly configured, for backward compatibility (i.e. use by non X-flag aware nodes).

A router transmitting a PIO-X RA MUST NOT configure for itself any address from with the PIO-X prefix. (If it did, the prefix would logically no longer be of exclusive use for the receiving node.)

5.4. Comparison with DHCPv6 PD

There exists a key difference in semantics between PIO-X and DHCPv6 PD: with PIO-X the network keeps the client refreshed with its prefix whereas with DHCPv6 PD the client is responsible for refreshing its prefix from the server. This is one reason it is important for the data link layer to be able to quickly inform routers of client detachment.

Another difference is that [16] section 12.1 states:

> ... the requesting router MUST NOT assign any delegated prefixes or subnets from the delegated prefix(es) to the link through which it received the DHCP message from the delegating router.

In contrast, a node receiving a PIO-X RA is explicitly free to treat the entire prefix as on-link with respect to the interface via which it was received.

6. Host behavior

TODO: This section needs some work.

6.1. PIO-X processing

A receiving node compliant with this document processes an RA with a PIO entry with the X flag set according the requirements in previous standards documents (chiefly [17] section 6.3.4) subject to the additional requirements documented in Section 5.2.

6.2. Neighbor Discovery implications

6.2.1. Duplicate Address Detection (DAD)

Whatever use the host makes of the exclusive prefix during its valid lifetime, it SHOULD NOT perform Duplicate Address Detection ("DAD", [18] section 5.4) on any address it configures from within the prefix if that address is configured on either the interface over which the PIO-X RA was received or on a loopback interface. Note that this does not absolve the host from performing DAD in all scenarios; if, for example, the host uses the prefix for 64sharing [19] it MUST at a minimum defend via DAD any addresses it has configured for itself as documented in Requirement 2 of [20] section 3.

6.2.2. Router Solicitations (RSes)

Routers announcing PIO-X RAs do so via IPv6 unicast to the intended receiving node and may note the IPv6 unicast destination address of an RS as the next hop for the exclusive prefix. As such, hosts compliant with this SHOULD NOT use the unspecified address (::) when sending RSes; they SHOULD prefer issuing Router Solicitations from a link-local address.

It is possible for a node to receive multiple RAs with a mix of exclusive and non-exclusive PIOs and even non-zero and zero default router lifetimes. While it is not possible for a host (receiving node) to be sure it has received all the RA information available to it, hosts compliant with this specification SHOULD implement Packet-Loss Resiliency for Router Solicitations [RFC7559] so that the host continues to transmit Router Solicitations at least until an RA with a non-zero default router lifetime has been seen.

6.3. Link-local address behavior

Routers announcing PIO-X RAs may record the source (link-local) address of an RS as the next hop for the exclusive prefix. A node compliant with this specification MUST continue to respond to Neighbor Solicitations for the source address used to send RSes (alternatively: the destination address of unicast PIO-X RAs received). Hosts that deprecate or even remove this address may experience a loss of connectivity.

6.4. Source address selection

No change to existing source address selection behavior is required or specified by this document.

6.5. Next hop router selection

No change to existing next hop router selection behavior is required or specified by this document.

6.6. Implications for Detecting Network Attachment

TODO: Describe implications for Detecting Network Attachment in IPv6 [21] (DNAv6). Probably the best that can be done is (a) no change to RFC6059 coupled with (b) a host MAY send a test packet (e.g. ICMPv6 Echo Request) with a source and destination address from within the PIO-X prefix to the PIO-X RA issuing router and verify the packet is delivered back to itself. Consistent failure to receive such traffic MAY be considered a signal that the exclusive prefix should no longer be used by the host.

6.7. Additional guidance

The intent of networks that use PIO-X RAs is not to enable sophisticated routing architectures that could be far better handled by an actual routing protocol but rather to propagate a prefix’s exclusive use information to enable the receiving node to make better use of the available addresses. As such:

> A PIO-X receiving node SHOULD NOT issue ICMPv6 Redirects ([RFC4861] section 4.5) for any address within an exclusive use prefix via the link over which the PIO-X RA was received. Redirecting portions of exclusive prefixes to other "upstream" on-link nodes is not a supported configuration.
>
> A PIO-X receiving node SHOULD NOT transmit RAs with any subset of its exclusive prefixes via the same interface through which the exclusive prefix was learned.

7. Router behavior

TODO: This section needs some work.

7.1. PIO-X RA destination address

Since the host will not perform DAD for addresses within prefix announced via PIO-X, it’s very important that only a single host receives the PIO-X RA. Therefore, the router MUST only include PIO-X in RAs that are sent using unicast RAs to destination unicast link-layer address and IPv6 link-local unicast address for a specific host. For point-to-point media without link-layer addresses or where there is guaranteed to only be single host that will receive the PIO-X RA (e.g. as enforced by link layer mechanisms), the router MAY

send PIO-X RA with multicast destination IPv6 address. Under all circumstances the router MUST maintain a binding table of state information as discussed in Section 7.3.

7.2. Detecting hosts to send PIO-X RAs to

When the host starts using a network connection it normally sends out an RS (Router Solicitation) packet. This is one way for the router to detect that a new host is connected to the network and detects its link-local address. If the router is configured to use PIO-X, it can now perform necessary processing/configuration and then send the PIO-X RA.

For some networks, the host information regarding link-layer and link-local address might be available through other mechanism(s). Examples of this are PPP, 802.1x and 3GPP mobile networks. In that case this information MAY be used instead of relying on the host to send RS. It is however RECOMMENDED that these networks also provide indication whether the host is no longer connected to the network so that the router can invalidate the prefix binding prior to binding expiration (timeout).

7.3. Binding table requirements

Routers transmitting PIO-X RAs have state maintenance and operational requirements similar to delegating routers in networks where DHCPv6 Prefix Delegation [RFC3633] is used. The state maintained is describe here in terms of a conceptual binding table.

R1 The router SHOULD keep track of which PIO-X prefix has been issued to each node.

R2 The router SHOULD keep the binding between prefix and link-local address for the advertised valid lifetime, plus some operationally determined delay prior to reissuing a prefix ("grace period"), of the prefix.

R3 The router MUST monitor the reachability of each node in the binding table via Neighbor Unreachability Detection ("NUD", [22] section 7.3) or an equivalent link-layer mechanism.

R4 The binding SHOULD be considered refreshed every time a periodic PIO-X RA is sent to a node.

R5 If the router is informed by some other mechanism (link-layer indication for instance) that a node is no longer connected to the link, it MAY immediately invalidate the prefix binding. (DISCUSS: Is this the correct approach? Do we want to point to some definition somewhere else?)

7.4. Preparations before sending a PIO-X RA

When the router intends to send a PIO-X RA, it SHOULD before sending the PIO-X RA, complete any and all necessary processing for the host to start using the PIO-X prefix to communicate through the router to other networks. This is so that the host can start using PIO-X based addresses without delay or error after receipt of the PIO-X RA.

7.5. Implementation considerations

TODO: Out of scope things that are worth careful consideration include...

Routers SHOULD NOT announce the same prefix to two different nodes within the valid lifetime of the earlier of the two PIO-X announcements.

A link may operate in a mode where routers announce RAs to all nodes, possibly with non-exclusive PIO data, and non-zero default router lifetimes. Separately, one or more other nodes on the link may announce exclusive PIO information to nodes along with zero default router lifetimes. Except in the presence of a non-expired more specific route, e.g. learning from an [23] Route Information Option (RIO), the receiving node should send exclusive use prefix originated or forwarded traffic destined off-link through routers with non-zero default router lifetimes.

8. Acknowledgements

9. IANA Considerations

This memo contains no requests of IANA.

10. Security Considerations

This document fundamentally introduces no new protocol or behavior substantively different from existing behavior on a link which guarantees a unique /64 prefix to every attached host. It only describes a mechanism to convey that topological reality, allowing the host to make certain optimizations as well as share the exclusive prefix as it sees fit with other nodes according to its capabilities and policies.

11. References

11.1. Normative References

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

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <http://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <http://www.rfc-editor.org/info/rfc4862>.

[RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <http://www.rfc-editor.org/info/rfc6275>.

[RFC7559] Krishnan, S., Anipko, D., and D. Thaler, "Packet-Loss Resiliency for Router Solicitations", RFC 7559, DOI 10.17487/RFC7559, May 2015, <http://www.rfc-editor.org/info/rfc7559>.

11.2. Informative References

[RFC3633] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, DOI 10.17487/RFC3633, December 2003, <http://www.rfc-editor.org/info/rfc3633>.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <http://www.rfc-editor.org/info/rfc4191>.

[RFC4429] Moore, N., "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, DOI 10.17487/RFC4429, April 2006, <http://www.rfc-editor.org/info/rfc4429>.

[RFC5452] Hubert, A. and R. van Mook, "Measures for Making DNS More Resilient against Forged Answers", RFC 5452, DOI 10.17487/RFC5452, January 2009, <http://www.rfc-editor.org/info/rfc5452>.

[RFC7066] Korhonen, J., Ed., Arkko, J., Ed., Savolainen, T., and S. Krishnan, "IPv6 for Third Generation Partnership Project

(3GPP) Cellular Hosts", RFC 7066, DOI 10.17487/RFC7066, November 2013, <http://www.rfc-editor.org/info/rfc7066>.

[RFC7934] Colitti, L., Cerf, V., Cheshire, S., and D. Schinazi, "Host Address Availability Recommendations", BCP 204, RFC 7934, DOI 10.17487/RFC7934, July 2016, <http://www.rfc-editor.org/info/rfc7934>.

Authors' Addresses

Erik Kline
Google Japan KK
6-10-1 Roppongi
Mori Tower, 44th floor
Minato, Tokyo 106-6126
JP

Email: ek@google.com

Mikael Abrahamsson
T-Systems Nordic
Kistagangen 26
Stockholm
SE

Email: Mikael.Abrahamsson@t-systems.se

Network Working Group
Internet-Draft
Updates: rfc4191 (if approved)
Intended status: Standards Track
Expires: December 26, 2019

F. Templin, Ed.
Boeing Research & Technology
J. Woodyatt
Google

June 24, 2019

Route Information Options in IPv6 Neighbor Discovery
draft-templin-6man-rio-redirect-08.txt

Abstract

The IPv6 Neighbor Discovery (ND) protocol allows nodes to discover neighbors on the same link. Router Advertisement (RA) messages can also convey routing information by including a non-zero (default) Router Lifetime, and/or Route Information Options (RIOs). This document specifies backward-compatible extensions that permit nodes to include RIOs in other IPv6 ND messages to support the discovery of more-specific routes among neighbors on the link.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at 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 26, 2019.

Copyright Notice

Table of Contents

1. Introduction 2
2. Terminology 3
3. Motivation 3
4. Route Information Options (RIOs) in IPv6 Neighbor Discovery Messages 5
4.1. RIO Update 5
4.2. RIO Requirements 7
4.3. Classic Redirection Scenario 7
4.4. RIO Redirection Scenario 9
4.4.1. Router Specification 9
4.4.2. Source Specification 10
4.4.3. Target Specification 11
4.5. Operation Without Redirects 11
4.6. Multiple RIOs 12
4.7. Multicast 12
4.8. Why NS/NA? 12
5. Implementation Status 13
6. IANA Considerations 13
7. Security Considerations 14
8. Acknowledgements 15
9. References 15
9.1. Normative References 15
9.2. Informative References 16
Appendix A. Link-layer Address Changes 17
Appendix B. Interfaces with Multiple Link-Layer Addresses . . . 17
Appendix C. Change Log 17
Authors' Addresses 18

1. Introduction

"Neighbor Discovery for IP version 6 (IPv6)" [RFC4861] (IPv6 ND) provides a Router Solicitation (RS) function allowing nodes to solicit a Router Advertisement (RA) response from an on-link router, a Neighbor Solicitation (NS) function allowing nodes to solicit a Neighbor Advertisement (NA) response from an on-link neighbor, and a Redirect function allowing routers to inform nodes of a better next hop neighbor on the link toward the destination. Further guidance for processing Redirect messages is given in "First-Hop Router Selection by Hosts in a Multi-Prefix Network" [RFC8028].

"Default Router Preferences and More-Specific Routes" [RFC4191] specifies a Route Information Option (RIO) that routers can include

in RA messages to inform recipients of more-specific routes (section 1 of that document provides rationale for the use of RA messages instead of an adjunct routing protocol). This document specifies a backward-compatible and incrementally-deployable extension to allow nodes to include RIOs in other IPv6 ND messages to support the dynamic discovery of more-specific routes. This allows nodes to discover a better neighbor for more-specific routes to both increase performance and reduce the workload on default routers.

This approach applies to any link type on which there may be many nodes that provision delegated prefixes on their downstream interfaces and do not provide transit services between upstream networks. These nodes can either be routers that forward packets on behalf of their downstream networks, or hosts that use a delegated prefix for their own multi-addressing purposes [I-D.templin-v6ops-pdhost][RFC7934].

This work benefits from the experience of [RFC6706] - an experimental protocol that uses UDP-based "pseudo-ND" messages instead of actual ICMPv6 message codes. That experience has shown that using synthesized UDP messages in addition to the IPv6 ND messaging already present on the link is inefficient. Furthermore, the UDP approach is neither backward-compatible nor incrementally-deployable, since sending UDP messages blindly to a node that does not have the port open could be misinterpreted as a port scan attack. This specification avoids these issues by using the already-present and natural IPv6 ND messaging available on the link, as specified in this document.

2. Terminology

The terminology in the normative references applies.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. Lower case uses of these words are not to be interpreted as carrying RFC2119 significance.

3. Motivation

An example of a good application for RIO is the local-area subnets served by the routers described in "Basic Requirements for IPv6 Customer Edge Routers" [RFC7084]. While many customer edge routers are capable of operating in a mode with a dynamic routing protocol operating in the local-area network, the default mode of operation is typically designed for unmanaged operation without any dynamic routing protocol. On these networks, the only means for any node to

learn about routers on the link is by using the Router Discovery protocol described in [RFC4861].

Nevertheless, hosts on unmanaged home subnets may use prefix delegation services such as DHCPv6 [RFC8415] to receive IPv6 routing prefixes for additional subnets allocated from the space provided by the service provider, and operate as routers for other links where hosts in delegated subnets are attached. Hosts may even learn about more specific routes than the default route by processing RIOs in RA messages as described in [RFC4191].

However, due to perceptions of the security considerations for hosts in processing RIOs on unmanaged networks, the default configuration for common host IPv6 implementations is to ignore RIOs. Accordingly, on typical home networks the forwarding path from hosts on one subnet to destinations on every off-link local subnet always passes through the customer edge router, even when a shorter path would otherwise be available through an on-link router. This adds costs for retransmission on shared LAN media, often adding latency and jitter with queuing delay and delay variability. This is not materially different under the scenarios described in "IPv6 Home Networking Architecture Principles" [RFC7368] except that routers may use an interior dynamic routing protocol to coordinate sending of RIOs in RA messages, which as explained above, are not processed by typical hosts.

In increasingly common practice, a node that receives a prefix delegation can use the prefix for its own multi-addressing purposes or can connect an entourage of "Internet of Things (IoT)" back end devices (an approach sometimes known as "tethering" [RFC7934]). On many link types, the number of such nodes may be quite large which would make running a dynamic routing protocol between the nodes impractical. Example use cases include:

- IETF conference, airport, and hotel WiFi networks, where large numbers of nodes on the link could receive IPv6 prefix delegations. Using the extensions described in this document, the nodes could dynamically discover more-specific routes to enable direct neighbor-to-neighbor communications.

- Mobile enterprise devices that connect into a corporate network via VPN links. Using the extensions described in this document, mobile devices could dynamically establish pair-wise VPN links between themselves without having to use the enterprise network as transit.

- Civil aviation networks where an aircraft holds an IPv6 prefix derived from the identification value assigned to it by the

International Civil Aviation Organization (ICAO). Using the extensions described in this document, direct paths between the aircraft and Air Traffic Control (ATC) can be established to provide a more direct route for communications.

- Unmanned Air System (UAS) networks where each UAS receives an IPv6 prefix delegation for operation within the Unmanned Air Traffic Management (UTM) service under development within NASA and the FAA. Using the extensions described in this document, very large numbers of UAS can be accommodated by the UTM service for both vehicle-to-infrastructure and vehicle-to-vehicle communications.

By using RIOs in IPv6 ND messages, the forwarding path between subnets can be shortened while accepting a much narrower opening of attack surfaces on general purpose hosts related to the Router Discovery protocol. The basic idea is simple: hosts normally send packets for off-link destinations to their default router unless they receive ND Redirect messages designating another on-link node as the target. This document allows ND Redirects additionally to suggest another on-link node as the target for one or more routing prefixes, including one with the destination. Hosts that receive RIOs in ND Redirect messages then send NS messages to the target containing those RIOs, and process the NA messages the target sends in reply. If hosts only process RIOs in NA messages when they have previously sent them in NS messages to the targets of received ND Redirect messages, then hosts only process the RIOs at the initiative of routers they already accept as authoritative.

4. Route Information Options (RIOs) in IPv6 Neighbor Discovery Messages

The RIO is specified for inclusion in RA messages in Section 2.3 of [RFC4191], while the neighbor discovery functions are specified in [RFC4861]. This specification permits routers to include RIOs in other IPv6 ND messages so that recipients can discover a better next hop for a destination *prefix* instead of just a specific destination address. This specification therefore updates [RFC4191] as discussed in the following sections.

4.1. RIO Update

The RIO format given in Section 2.3 of [RFC4191] is updated by this specification as shown in Figure 1:

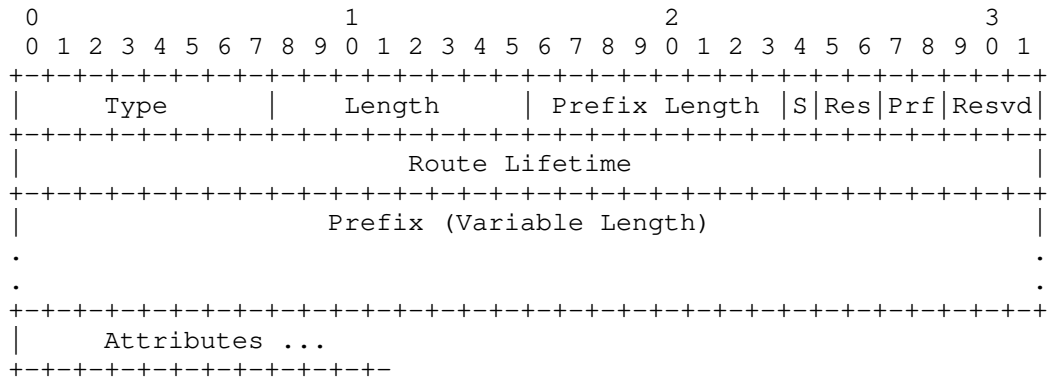


Figure 1: Updated RIO Format

This format introduces a new S flag and variable-length Attributes. The fields of the main body of the RIO are set as follows:

- Type, Prefix Length, Prf, Route Lifetime and Prefix are set exactly as specified in Section 2.3 of [RFC4191].

- For RA messages, Length is set exactly as specified in Section 2.3 of [RFC4191] and no Attributes are included. For all other IPv6 ND messages, Length MUST be initialized to exactly 1 when Prefix Length is 0, to exactly 2 when Prefix Length is between 1 and 64, and to exactly 3 when Prefix Length is greater than 64. Length is then incremented by the length of all included Attributes in units of 8-octets (see below).

- S is set to '1' to "Solicit" route information or to '0' (i.e., the default value) to "Assert" route information.

- Res and Resvd are reserved and MUST be set to '0'.

Attributes MAY be included as ancillary route information. Each Attribute is formatted in the same manner as specified for IPv6 ND options in Section 4.6 of [RFC4861] and as shown in Figure 2:

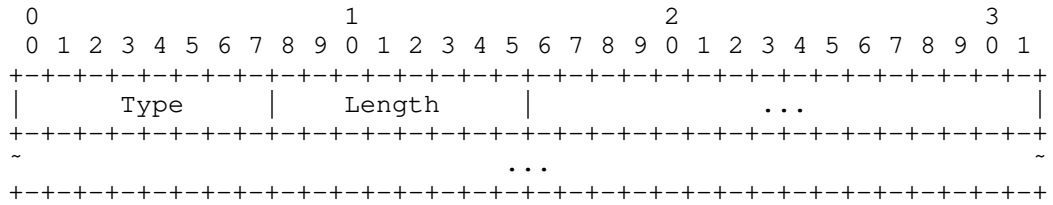


Figure 2: RIO Attribute Format

This document defines the NULL Attribute with Type '0'. Other Attribute Types are assigned through IANA action.

When Type is '0', Length MUST be set to the total number of 8-octet blocks in the Attribute, and the Attribute body MUST include a corresponding number of '0' octets. For example, for Lengths of 1, 2, 3, etc., the Attribute body includes 6, 14, 22, etc. '0' octets, respectively.

Receivers ignore any NULL, unknown or malformed Attributes and continue to process any other Attributes in the RIO that follow.

4.2. RIO Requirements

This specification updates [RFC4191] by allowing RIOs to appear in any IPv6 ND messages with the following requirements:

- Redirect, NA and RA messages MUST NOT include RIOs with the S flag set to '1'; any RIOs received in Redirect, NA and RA messages with S set to '1' MUST be silently ignored.

- NS and RS messages MAY include some RIOs with S set to '1' and others with S set to '0'.

- NA/RA responses to RIOs in NS/RS messages with S set to '1' MUST include RIOs with the solicited route information and with S set to '0'. (If the route information solicited by the NS/RS message is incorrect or unrecognized, however, the RIO MUST be silently ignored.)

- Asserted route information in any RIOs received with S set to '0' SHOULD be considered as "unconfirmed" until the assertion can be verified. Assertion verification can be through a trust anchor such as a trusted on-link router, through a static routing table, or through some other means outside the scope of this document. Any route information that cannot be verified SHOULD be ignored.

The following sections present the classic redirection scenario illustrating an exchange where a trusted on-link router is used to verify RIO assertions. Other IPv6 ND messaging scenarios that can employ some other means of verifying RIO assertions are also acceptable.

4.3. Classic Redirection Scenario

In the classical redirection scenario there are three actors, namely the Source, Router and Target as shown in Figure 3:

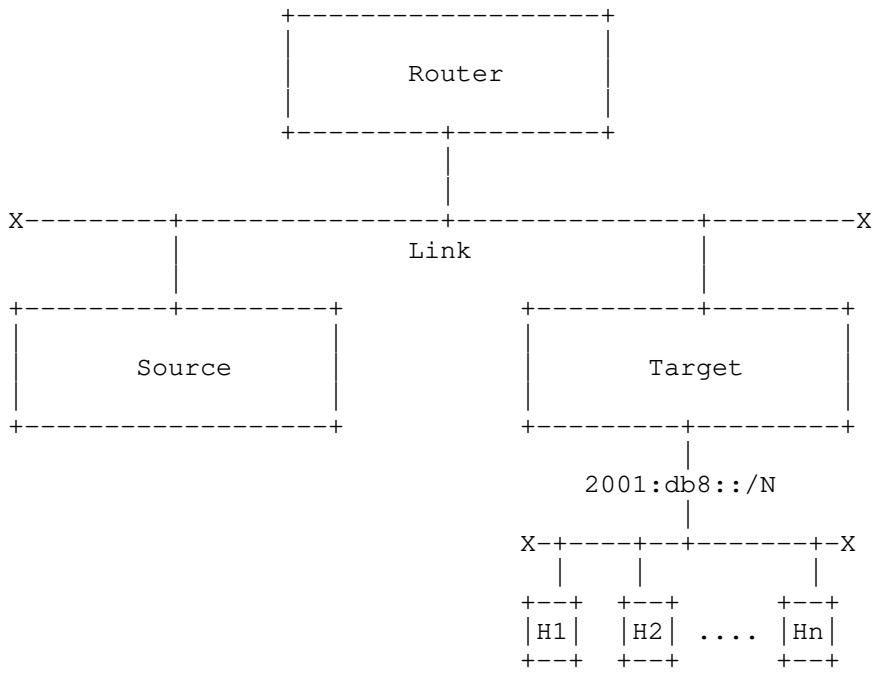


Figure 3: Classical Redirection Scenario

In addition, the Target may be a node that connects an arbitrarily-complex set of IPv6 networks (e.g., as depicted by 2001:db8::/N in the figure) with hosts H(i).

In this scenario, the Source initially has no route for 2001:db8::/N and must send initial packets destined to correspondents H(i) via a first-hop Router. Upon receiving the packets, the Router forwards the packets to the Target and may also send a Redirect message back to the Source with the Destination Address field set to the destination of the packet that triggered the Redirect, the Target Address field set to the target link-local address and with a Target Link Layer Address Option (TLLAO) that includes the target link-layer address. After receiving the message, the Source may begin sending packets destined to H(i) directly to the Target, which will then forward them to addresses within its internal and/or external IPv6 network prefixes.

This specification augments the classical Redirection scenario by allowing the Router to include entire prefixes (e.g., 2001:db8::/N) in RIOs in the Redirect message, and thereafter allowing the Source to include RIOs in an NS message and the Target to include RIOs in

its NA response. The following sections present this "augmented" RIO redirection scenario.

4.4. RIO Redirection Scenario

In the RIO redirection scenario, the Source sends initial packets via the Router the same as in the classical scenario. When the Router receives the packets, it searches its routing tables for a route that is assigned to the Target and that covers the destination address of the packet. The Router then includes the route in an RIO in a Redirect message to send back to the Source. The Router sets the S flag in the RIO to '0' to indicate that a prefix is being asserted.

When the Source receives the Redirect message, it prepares an NS message that includes the route information received in the RIO from the Redirect message and with S set to '1 to indicate that route information is being solicited. At the same time, if the Source needs to assert any route information to the Target, it includes the information in RIOs with S set to '0'. The Source then sends the NS message to the Target.

When the Target receives the NS message, it records any route information in RIOs with S set to '0' as unconfirmed route information for the Source pending verification. At the same time, it determines whether the route information included in any RIOs with S set to '1' matches one of its own routes. If so, the Target includes the route information in an RIO with S set to '0' to return in an NA message reply to the Source.

When the Source receives the NA message it can install any RIO information that matches the Redirect RIOs in its routing table. The following sections present more detailed specifications for the Router, Source and Target.

4.4.1. Router Specification

When the Router receives a packet from the Source it searches its routing table for a prefix that covers the destination address (e.g., 2001:db8::/N as depicted in Figure 1), where prefix could be populated in the routing table during prefix delegation, via manual configuration, etc. If the next hop for the prefix is on-link (i.e., a "Target" in the terms of [RFC4861]), the Router then prepares a Redirect message with the Destination Address field set to the packet's IPv6 destination address, with the Target Address field set to the link-local address of the Target, with a TLLAO set to the link-layer address of the Target, and with an RIO that includes route information for the prefix with Route Lifetime, Prf, and S set to 0.

The Router then sends the Redirect message to the Source (subject to rate limiting).

4.4.2. Source Specification

According to [RFC4861], a Source that receives a valid Redirect message updates its destination cache per the Destination Address and its neighbor cache per the Target Address. According to [RFC4191], Sources can be classified as Type "A", "B" or "C" based on how they process RIOs, where a Type "C" Source updates its routing table per any RIO elements included in an RA message. Finally, according to [RFC8028], a Type "C" Source operating on a Multi-Prefix Network with multiple default routes can make source address selection decisions based on information in its routing table decorated with information derived from the source of the RIO element.

In light of these considerations, this document introduces a new Type "D" behavior for Sources with the same behavior as a Type "C" Source, but which also process RIO elements in other IPv6 ND messages. Type "D" Sources process Redirect messages with RIO elements by first verifying that the Prefix in the first RIO matches the Destination Address. If the Destination Address does not match the Prefix, the Source discards the Redirect message. Otherwise, the Source updates its neighbor cache per the Target Address and its destination cache per the Destination Address the same as for classical redirection. Next, the Source MAY send an NS message to the Target containing an RIO with the Prefix and Prefix Length and with S set to ’1’ to elicit an NA response (at the same time, the Source MAY include RIOs with S set to ’0’ if it needs to assert any route information to the Target).

When the Type ’D’ Source receives the solicited NA message from the Target, if the NA includes an RIO with S set to ’0’ and with a Prefix corresponding to the one received in the Redirect message, the Source installs the route information in its routing table with the Target’s address as the next hop. (Note that the Prefix Length received in the NA message MAY be different than the Prefix Length received in the Redirect message. If the Prefix Length in the NA is the same or longer, the Source accepts the Prefix as verified by the Router; if the Prefix Length is shorter, the Source considers the Prefix as unconfirmed.)

After the Source installs the route information in its routing table, it MAY begin sending packets with destination addresses that match the Prefix directly to the Target Instead of sending them to the Router. The Source SHOULD decrement the Route Lifetime and MAY send new NS messages to receive a fresh Route Lifetime (if the Route Lifetime decrements to 0, the Source instead deletes the route

information from its routing table). The Source MAY furthermore delete the route information at any time and again allow subsequent packets to flow through the Router which may send a fresh Redirect. The Source SHOULD then again test the route by performing an NS/NA exchange with the Target the same as described above.

After updating its routing table, the Source may receive an unsolicited NA message from the Target with an RIO with new route information. If the RIO Prefix is in its routing table, and if the RIO Route Lifetime value is 0, the Source deletes the corresponding route.

After updating its routing table, the Source may subsequently receive a Destination Unreachable message from the Target with Code '0' ("No route to destination"). If so, the Source SHOULD delete the corresponding route information from its routing table and again allow subsequent packets to flow through the Router.

4.4.3. Target Specification

When the Target receives an NS message from the Source containing an RIO with S set to '1', it examines the Prefix and Prefix Length to see if it matches one of the prefixes in its routing table. If so, the Target prepares an NA message with an RIO including a Prefix and Prefix Length, any necessary route information, and with S set to '0'. The Target then sends the NA message back to the Source.

If the NS included any RIO options with S set to '0', the Target SHOULD employ a suitable means to verify the asserted route information, and SHOULD reject any route information that cannot be verified.

At some later time, the Target may either alter or deprecate one of its routes. If the Target has asserted route information in RIOs to one or more Sources, the Target SHOULD send unsolicited NA messages with RIOs that assert new route information to alter the route, where a new Route Lifetime value of '0' deprecates the route. If the Target receives a packet with a destination addresses for which there is no matching route for one of its downstream networks, the Target sends a Destination Unreachable message to the Source with Code '0' ("No route to destination"), subject to rate limiting.

4.5. Operation Without Redirects

If the Source has some way to determine the Target's link-local address without receiving a Redirect message from the Router, the Source MAY send an NS message with an RIO directly to the Target with

S set to 1, Prefix set to the destination address of an IPv6 packet, Prefix Length set to 128 and all other route information is set to 0.

When the Target receives the NS message, it prepares an NA response with an RIO that includes route information for the shortest one of its prefixes that covers the destination address. The Target then sends the NA message to the Source.

When the Source receives the NA message, it SHOULD consider the route information asserted in the RIO as unconfirmed until it can verify the Target's claim (i.e., as described in Section 4.2).

Any node may also assert route information at any time by sending IPv6 ND messages with RIOs with S set to 0. Recipients of such messages SHOULD consider the route information as unconfirmed until the information can be verified.

4.6. Multiple RIOs

If a Redirect includes multiple RIOs, the Source only checks the destination address for a match against the Prefix in the first RIO.

If an NS/RS message includes multiple RIOs with S set to '1', the neighbor responds to those RIOs which match entries in its routing table.

If an NS/NA/RS/RA message includes multiple RIOs with S set to '0', the neighbor considers all of the route information as unconfirmed until the information can be verified.

4.7. Multicast

Nodes MAY send IPv6 ND messages with RIOs to link-scoped multicast destination addresses including All Nodes, All Routers, and Solicited-Node multicast (see: [RFC4291]. As an example, a node could send unsolicited NA messages to the All Nodes multicast address to alter or deprecate a route it had previously asserted to one or more neighbors.

Nodes MUST be conservative in their use of multicast IPv6 ND messaging to avoid unnecessarily disturbing other nodes on the link.

4.8. Why NS/NA?

Since [RFC4191] already specifies the inclusion of RIOs in RA messages, a natural question is why use NS/NA instead of RS/RA?

First, RA messages are only sent over advertising interfaces [RFC4861]. Source and Target nodes typically connect only downstream networks; hence, they configure their upstream interfaces as non-advertising interfaces.

Second, NS/NA exchanges used by the IPv6 Neighbor Unreachability Detection (NUD) procedure are unicast-based whereas RA responses to RS messages are typically sent as multicast. Since this mechanism must support unicast operation, the use of unicast NS/NA exchanges is preferred.

Third, the IPv6 ND specification places restrictions on minimum delays between RA messages. Since this mechanism expects an immediate advertisement from the Target in response to the Source's solicitation, only the NS/NA exchange can satisfy this property.

Fourth, the RA message is the "swiss army knife" of the IPv6 ND protocol. RA messages carry numerous configuration parameters for the link, including Cur Hop Limit, M/O flags, Router Lifetime, Reachable Time, Retrans Time, Prefix Information Options, MTU options, etc. The Target must not advertise any of this information to the soliciting Source.

Fifth, RIOs in legacy RA messages cannot encode attributes and therefore may be limited in the route information they can carry.

Finally, operators are deeply concerned about the security of RA messages - so much so that they deploy link-layer security mechanisms that drop RA messages originating from nodes claiming to be an authoritative router for the link [RFC6105].

5. Implementation Status

The IPv6 ND functions and RIOs are widely deployed in IPv6 implementations, however these implementations do not currently include RIOs in IPv6 ND messages other than RAs.

An experimental implementation of [RFC6706] exists, and demonstrates how the Redirect function can be used to carry route information.

6. IANA Considerations

IANA is instructed to create a registry for "RIO Attributes" as discussed in Section 4.1. The registry includes the following initial entry:

0 - the NULL Attribute [draft-templin-6man-rio-redirect]

Other Attribute types are defined through standards action or expert review.

7. Security Considerations

The Redirect message validation rules in Section 8.1 of [RFC4861] require recipients to verify that the IP source address of the Redirect is the same as the current first-hop router for the specified ICMP Destination Address. Recipients therefore naturally reject any Redirect message with an incorrect source address.

Other security considerations for IPv6 ND messages that include RIOs are the same as specified in Section 11 of [RFC4861]. Namely, the protocol must take measures to secure IPv6 ND messages on links where spoofing attacks are possible.

A spoofed Redirect message containing no RIOs could cause corruption in the recipient's destination cache, while a spoofed Redirect message containing RIOs could corrupt the host's routing tables. While the latter would seem to be a more onerous result, the possibility for corruption is unacceptable in either case.

"IPv6 ND Trust Models and Threats" [RFC3756] discusses spoofing attacks, and states that: "This attack is not a concern if access to the link is restricted to trusted nodes". "SEcure Neighbor Discovery (SEND)" [RFC3971] provides one possible mitigation for other cases. In some scenarios, it may be sufficient to include only the Timestamp and Nonce options defined for SEND without implementing other aspects of the protocol.

"IPv6 Router Advertisement Guard" [RFC6105] ("RA Guard") describes a layer-2 filtering technique intended for network operators to use in protecting hosts from receiving RA messages sent by nodes that are not among the set of routers regarded as legitimate by the network operator.

Nodes must have some form of trust basis for knowing that the sender of an ND message is authoritative for the prefixes it asserts in RIOs. For example, when an NS/NA exchange is triggered by the receipt of a Redirect, the soliciting node can verify that the RIOs in the NA message match the ones it received in the Redirect message (which originally came from a trusted router).

Nodes that do not wish to provide transit services for upstream networks may also receive IPv6 packets via an upstream interface that do not match any of their delegated prefixes. In that case, the node drops the packets and observes the "Destination Unreachable - No route to destination" procedures discussed in [RFC4443]. Dropping

the packets is necessary to avoid a reflection attack that would cause the node to forward packets received from an upstream interface via the same or a different upstream interface.

8. Acknowledgements

Joe Touch suggested a standalone draft to document this approach in discussions on the intarea list. The work was subsequently transferred to the 6man list, where the following individuals provided valuable feedback: Mikael Abrahamsson, Zied Bouziri, Brian Carpenter, Steinar Haug, Christian Huitema, Tatuya Jinmei, Tomoyuki Sahara.

Discussion with colleagues during the "bits-and-bites" session at IETF98 helped shape this document. Those colleagues are gratefully acknowledged for their contributions.

This work is aligned with the NASA Safe Autonomous Systems Operation (SASO) program under NASA contract number NNA16BD84C.

This work is aligned with the FAA as per the SE2025 contract number DTFAWA-15-D-00030.

This work is aligned with the Boeing Information Technology (BIT) MobileNet program and the Boeing Research & Technology (BR&T) enterprise autonomy program.

9. References

9.1. Normative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

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

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <https://www.rfc-editor.org/info/rfc2460>.

[RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <https://www.rfc-editor.org/info/rfc4191>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

9.2. Informative References

[I-D.templin-v6ops-pdhost] Templin, F., "IPv6 Prefix Delegation and Multi-Addressing Models", draft-templin-v6ops-pdhost-23 (work in progress), December 2018.

[RFC3756] Nikander, P., Ed., Kempf, J., and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, DOI 10.17487/RFC3756, May 2004, <https://www.rfc-editor.org/info/rfc3756>.

[RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <https://www.rfc-editor.org/info/rfc3971>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

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

[RFC6706] Templin, F., Ed., "Asymmetric Extended Route Optimization (AERO)", RFC 6706, DOI 10.17487/RFC6706, August 2012, <https://www.rfc-editor.org/info/rfc6706>.

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

[RFC7368] Chown, T., Ed., Arkko, J., Brandt, A., Troan, O., and J. Weil, "IPv6 Home Networking Architecture Principles", RFC 7368, DOI 10.17487/RFC7368, October 2014, <https://www.rfc-editor.org/info/rfc7368>.

[RFC7934] Colitti, L., Cerf, V., Cheshire, S., and D. Schinazi, "Host Address Availability Recommendations", BCP 204, RFC 7934, DOI 10.17487/RFC7934, July 2016, <https://www.rfc-editor.org/info/rfc7934>.

[RFC8028] Baker, F. and 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>.

[RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <https://www.rfc-editor.org/info/rfc8415>.

Appendix A. Link-layer Address Changes

Type "D" hosts send unsolicited NAs to announce link-layer address changes per standard neighbor discovery [RFC4861]. Link-layer address changes may be due to localized factors such as hot-swap of an interface card, but could also occur during movement to a new point of attachment on the same link.

Appendix B. Interfaces with Multiple Link-Layer Addresses

Type "D" host interfaces may have multiple connections to the link; each with its own link-layer address. Type "D" nodes can therefore include multiple link-layer address options in IPv6 ND messages. Neighbors that receive these messages can cache and select link-layer addresses in a manner outside the scope of this specification.

Appendix C. Change Log

<< RFC Editor - remove prior to publication >>

Changes from -07 to -08:

- Changed DHCPv6 Prefix Delegation reference to RFC8415.

Changes from -06 to -07:

- Fixed spelling and grammar.

Changes from -05 to -06:

- Minor adjustments to text and requirements.

Changes from -04 to -05:

- Removed "Ver" field and version numbers.
- Included reference to 'draft-templin-v6ops-pdhost'
- Changed "MAY" to "may" in two places
- Added text on advertising interfaces
- Added UAS use case

Authors' Addresses

Fred L. Templin (editor)
Boeing Research & Technology
P.O. Box 3707
Seattle, WA 98124
USA

Email: fltemplin@acm.org

James Woodyatt
Google
3400 Hillview Ave
Palo Alto, CA 94304
USA

Email: jhw@google.com

Network Working Group
Internet-Draft
Updates: rfc4861 (if approved)
Intended status: Standards Track
Expires: March 23, 2018

O. Troean
Cisco Systems

September 19, 2017

IPv6 ND PIO Flags IANA considerations
draft-troan-6man-ndpioiana-01

Abstract

The Prefix Information Option in the IPv6 Neighbor Discovery Router Advertisement defines an 8-bit flag field with two flags defined and the remaining 6 bits reservered (Reserved1). RFC 6275 has defined a new flag from this field without creating a IANA registry or updating RFC 4861. The purpose of this document is to request IANA to create a new registry for the PIO flags to avoid potential conflict in the use of these flags.

Table of Contents

1. Introduction 2
2. Current Prefix Information Option flags 2
3. Updates to RFC4861 3
4. IANA Considerations 3
5. References 3
5.1. Normative References 3
5.2. Informative References 4
Author's Address 4

1. Introduction

The Prefix Information Option in the IPv6 Neighbor Discovery Router Advertisement defines an 8-bit flag field with two flags defined and the remaining 6 bits reservered (Reserved1). RFC 6275 has defined a new flag from this field without creating a IANA registry or updating RFC 4861. The purpose of this document is to request IANA to create a new registry for the PIO flags to avoid potential conflict in the use of these flags.

2. Current Prefix Information Option flags

Currently, the NDP Prefix Information Option contains the following one-bit flags defined in published RFCs:

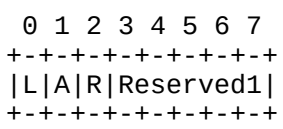


Figure 1

L - On-link Flag [RFC4861]

A - Autonomous Address Configuration Flag [RFC4861]

R - Router Address Agent Flag [RFC6275]

Reserved1 - Reserved

3. Updates to RFC4861

This document updates RFC4861 with the new IANA Considerations section specified below.

4. IANA Considerations

The IANA is requested to create a new registry for IPv6 ND Prefix Information Option flags. This should include the current flags in the PIO option. The format for the registry is:

RA Option Bit	Description	Reference
0	L - On-link Flag	[RFC4861]
1	A - Autonomous Address Configuration Flag	[RFC4861]
2	R - Router Address Flag	[RFC6275]

Figure 2

The assignment of new flags in the PIO option header require standards action or IESG approval.

The registry for these flags should be added to: http://www.iana.org/assignments/icmpv6-parameters

5. References

5.1. 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>.

[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>.

5.2. Informative References

[RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <https://www.rfc-editor.org/info/rfc6275>.

Author's Address

Ole Troean
Cisco Systems
Philip Pedersens vei 1
Lysaker 1366
Norway

Email: ot@cisco.com