Network Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: 20 June 2024

T. Herbert
SiPanda

18 December 2023

Limits on Sending and Processing IPv6 Extension Headers
draft-ietf-6man-eh-limits-12

Abstract

This specification defines various limits that may be applied to receiving, sending, and otherwise processing packets that contain IPv6 extension headers. The need for such limits is pragmatic to facilitate interoperability amongst hosts and routers in the presence of extension headers, thereby increasing the feasibility of deployment of extension headers. The limits described herein establish the minimum baseline of support for use of extension headers on the Internet. If it is known that all communicating parties for a particular communication, including destination hosts and any routers in the path, are capable of supporting more than the baseline then these default limits may be freely exceeded.

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 20 June 2024.

Table of Contents

1. Introduction

Extension headers are a core component of the IPv6 protocol as specified in [RFC8200]. IPv6 extension headers were originally defined with few restrictions. For instance, there is no specified limit on the number of extension headers a packet may have, nor is there a limit on the length in bytes of extension headers in a packet other than being limited by the Path MTU or 1,280 bytes for those hosts that do not discover the Path MTU [RFC7112]. Similarly, variable length extension headers typically do not have prescribed limits such as limits on the number of Hop-by-Hop or Destination options in a packet. The lack of limits essentially requires implementations to handle every conceivable usage of the protocol, including a myriad of use cases those are obviously outside the realm of ever being realistic or useful in real world deployment. Excessive Hop-by-Hop options in a packet has also been raised a security concern [RFC4942].

The lack of limits and the requirements for supporting a virtually open-ended protocol have led to a significant lack of support and deployment of extension headers ([RFC7872], [Cus23b]). Instead of

attempting to satisfy the protocol requirements concerning extension headers, some router and middlebox vendors have opted to invent and apply their own ad hoc limits, relegate packets with extension headers to slow path processing, or have gone so far as to summarily discard all packets with extension headers [RFC9098]. For those hosts and routers that properly attempt to process all extension headers per the specifications, the lack of limits has made them susceptible to Denial of Service attacks. The net effect of this situation is that deployment and use of extension headers is underwhelming to the extent that they are sometimes considered unusable on the Internet, and hence IPv6 extension headers have not lived up to their potential as the extensibility mechanism of IPv6.

As an example, consider that there is no limit on how many Hop-by-Hop or Destination options may be in an extension header in a packet, nor any limits as to how many options a receiver must process. A single 1,280 byte MTU size packet could legally contain a Hop-by-Hop or Destination Options header with over six hundred two byte options. There is no use case for this in the foreseeable future other than a Denial of Service attack where an attacker simply creates packets with hundreds of small unknown Hop-by-Hop or Destination options with the two high order bits in the option type set to 00 meaning to skip the unknown option. Any node in the path that attempts to dutifully process all these options could be easily overwhelmed by the processing needed to parse these options, hence this is an effective DOS attack. Note that this is a problem for both hardware and software implementations, as well as for both hosts and routers.

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

1.1. Related work

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

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

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

Section 5.3 of [RFC8504] defines a number of limits that hosts may apply to processing extension headers. For instance, a limit on the maximum number of non-padding options allowed in a Destination Options header or Hop-by-Hop Options header is defined. This specification expands on the requirements of [RFC8504] to allow limits to be set for routers and intermediate nodes.

[RFC8883] defines a set of ICMP errors that may be sent if a limit concerning extension headers is exceeded and a node discards a packet as a result. [RFC8883] allows both hosts and routers to send such messages (effectively acknowledging that some routers discard packets with extension headers even though such behavior might be non-conformant with [RFC8200]).

[RFC7872] presents real-world data regarding the extent to which packets with IPv6 Extension Headers (EHs) are discarded in the Internet. [RFC9098] summarizes the operational implications of IPv6 extension headers, and attempts to analyze reasons why packets with IPv6 extension headers are often discarded in the public Internet.

Section 2.1.9 of [RFC4942] discusses security concerns with IPv6 extension headers. Excessive Hop-by-Hop options are one concern, and misuse of PAD1 and PADN options are another. [RFC4942] provides some foundation for the limits defined in this specification.

This specification sets the minimal upper bounds on the number of Hop-by-Hop options that a node is expected to process. The lower bound is discussed in [I-D.ietf-6man-hbh-processing].

1.2. Terminology

This section provides definitions for some terms used in this specification.

Node: a device that implements IPv6

Router: a node that forwards IPv6 packets not explicitly addressed to itself

Intermediate node: a node that is addressed by an entry in a Routing Header list where the entry is not the last one in the list

Host: any node that is not a router or intermediate node

IPv6 header chain: the IPv6 header and the set of following IPv6 Extension Headers that precede the upper layer protocol in a packet

2. Overview of extension header limits

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

* Limits on extension header length

* Limits on option length

* Limits on number of extension headers

* Limits on number of options

* Limits on padding for extension headers with options

* Limits on the length of the IPv6 header chain

The limits in this specification are optional.

Limits are defined for both sending hosts and receivers. A receiver limit is set to limit the amount of processing or the amount of data for received extension headers. Sender limits are set to limit the use of extension headers being sent. The purpose of sender limits is to increase the probability of successful delivery.

For receiver limits, a recommended action when a limit is exceeded is specified. The recommended action depends on the type of the node. For a host or intermediate node, the action when a limit is exceeded is to discard the packet. The rationale is that hosts are required to process all of the headers in a packet to process it correctly, and intermediate nodes are required to process all the extension headers through the routing header to process a packet correctly. For a router, the action to take when a limit is exceeded is to stop processing the extension headers and to forward the packet; if a router is processing Hop-by-Hop options and a limited is exceeded then the router skips the option that caused the limit to be exceeded and skips any following Hop-by-Hop options per the procedures for skipping options in Section 5.2 of [I-D.ietf-6man-hbh-processing]. The rationale is that the only extension header a router may process is Hop-by-Hop Options and the packet can be correctly forwarded if none or some of the Hop-by-Hop options are processed.

This specification specifies limits on extension headers and their options both for byte length and number of headers or options. Limits on length are useful to nodes having hardware limitations, such as a fixed size parsing routers, which inherently limits the number of bytes of headers that a node can process. A node with such hardware limitations may choose to set length limits for extension

headers and options accordingly. Limits for the number of options are useful to nodes, such as end hosts, that have no inherent processing limitations. For these nodes, limits on number of headers or options are set to limit the cost of processing which is more a function of the number of items processed than the byte length of the items.

Each receiver limit described in this specification has a recommended default value or minimum value when limit is enabled. The intent of default limits is to establish an expected baseline of support. The default limits for senders correspond to the associated receiver default limits, thereby establishing the same default limits for senders and receivers. The derivation for default number of options is discussed in Appendix A.1. The derivation of default length limits is discussed in Appendix A.2.

Padding options in Hop-by-Hop and Destination options have a particular purpose to align the next option or to pad the length of the extension header to a multiple of eight bytes. Similar to non-padding options, padding options require processing to parse over. Unlike non-padding options, padding options serve no other purpose than padding. To that end, limits on padding can be more restrictive than those on non-padding options. The justification for padding limits is discussed in Appendix A.3.

3. Host limits for sending extension headers

The requirements for limits related to a host sending packets with extension headers are:

* A source host SHOULD NOT send more than 8 non-padding options in a Destination Options header unless it has explicit knowledge that the destination host, and all intermediate nodes in a routing header in the case of a Destination Options header before the routing header, are able to process a greater number of options.

* A source host SHOULD NOT send a packet with a Destination Options header larger than 64 bytes unless unless it has explicit knowledge that the destination host, and all intermediate nodes in a routing header in the case of a Destination Options header before the routing header, are able to process a larger option size.

* A source host SHOULD NOT send a packet with a Destination option larger than 60 bytes unless unless it has explicit knowledge that the destination host, and all intermediate nodes in a routing header in the case of a Destination Options header before the routing header, are able to process options of a larger size.

* A source host SHOULD NOT send more than 8 non-padding options in a Hop-by-Hop Options header unless it has explicit knowledge that all possible routers, intermediate nodes, and the destination host in the path are able to process a greater number of options or will ignore options that exceed their limit in the case of routers.

* A source host SHOULD NOT send a packet with a Hop-by-Hop Options header larger than 64 bytes unless it has explicit knowledge that all possible routers, intermediate nodes, and the destination host in the path are able to process options of a larger header size.

* A source host SHOULD NOT send a packet with a Hop-by-Hop option larger than 60 bytes unless unless it has explicit knowledge that all possible routers, intermediate nodes, and the destination host in the path are able to process options of a larger size.

* A source host SHOULD NOT send a packet with an IPv6 header chain larger than 104 bytes unless it has explicit knowledge that all possible routers, intermediate nodes, and the destination host in the path are able to process a larger IPv6 header chain. If a packet contains an IPsec header then this limit only applies through the end of the IPsec header (the IPsec header obfuscates following headers so that they are unreadable by nodes in the path). This requirement is equivalently stated as a host SHOULD NOT send a packet with more than 64 bytes of aggregate extension headers.

* A source host SHOULD NOT set more than one consecutive pad option, either PAD1 or PADN, in a Destination Options header or Hop-by-Hop Options header.

* A host SHOULD NOT send a packet with more than seven consecutive bytes of padding, using PAD1 or PADN options, in a Destination Options header or Hop-by-Hop Options header.

* A source host SHOULD follow the recommendations in Section 4.1 of [RFC8200] for extension header ordering and number of occurrences of extension headers.

4. Host and intermediate node limits for receiving extension headers

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

As described in [RFC8504] a destination host may establish limits on the processing of extension headers. This specification reiterates those requirements, expands the requirements to be applicable to intermediate nodes, add allows a receiving node to send an ICMP error [RFC8883] if a limit has been exceeded.

The requirements for limits related to a host or intermediate node receiving packets with extension headers are:

* A host or intermediate node MAY set a limit on the maximum number of non-padding options allowed in a Destination Options header or Hop-by-Hop Options header. If this limit is supported then the maximum number SHOULD be configurable, the limit SHOULD be greater than or equal to 8, and the RECOMMENDED default value is 8. The limits for Destination Options headers and Hop-by-Hop Options headers MAY be separately configurable. If a packet is received and the number of Destination or Hop-by-Hop options exceeds the limit, then the receiving node SHOULD discard the packet and MAY send an ICMP Parameter Problem message with code 9 (Too Many Options in Extension Header) [RFC8883] to the packet's source address.

* A host or intermediate node MAY set a limit on the length of a Destination Options header or a Hop-by-Hop Options header. If this limit is supported then the limit SHOULD be configurable and the limit SHOULD be greater than or equal to 64 bytes. The length limits for Destination Options headers and Hop-by-Hop Options headers MAY be separately configurable. If a packet is received and the length of an extension header exceeds the limit, then the receiving node SHOULD discard the packet and MAY send an ICMP Parameter Problem message with code 6 (Extension Header Too Big) [RFC8883] to the packet's source address.

* A host or intermediate node MAY set a limit on the maximum length of the IPv6 header chain, or equivalently a host MAY set a limit on the aggregate length of extension headers in a packet. If the limit is set then it SHOULD be greater than or equal to 104 bytes, or, equivalently, the limit on aggregate header extension length SHOULD be greater than or equal to 64 bytes. If a packet is received and the aggregate length of the IPv6 header chain exceeds the limit, then the receiving node SHOULD discard the packet and MAY send an ICMP Parameter Problem message with code 7 (Extension Header Chain Too Long) [RFC8883] to the packet's source address.

* A host or intermediate node MAY limit the number of consecutive bytes of padding in PAD1 or PADN options in a Destination Options header or Hop-by-Hop Options header to 7. If the limit is enabled and a packet is received and there are more than 7 consecutive

bytes of padding, then the receiving node SHOULD discard the packet and MAY send an ICMP Parameter Problem message with code 9 (Too Many Options in Extension Header) [RFC8883] to the packet's source address.

* A host or intermediate node MAY disallow consecutive padding options, either PAD1 or PADN, to be present in a packet. If a packet is received with consecutive padding options that are disallowed by the receiving node, then the receiving node SHOULD discard the packet and MAY send an ICMP Parameter Problem message with code 9 (Too Many Options in Extension Header) [RFC8883] to the packet's source address.

* A host or intermediate MAY enforce the recommended extension header ordering and number of occurrences of extension headers described in Section 4.1 of [RFC8200]. Per the ordering recommendations, each extension header can occur at most once in a packet with the exception of Destination Options header which can occur twice. The recommended extension header ordering per [RFC8200] is:

  - IPv6 header

  - Hop-by-Hop Options header

  - Destination Options header

  - Routing header

  - Fragment header

  - Authentication header

  - Encapsulating Security Payload header

  - Destination Options header

  - Upper-Layer header

  If a host or intermediate node enforces extension header ordering and a packet is received with extension headers out of order or the number of occurrences of an extension header is greater than one, or two for the Destination Options header, then the receiving node SHOULD discard the packet and MAY send an ICMP Parameter Problem message with code 8 (Too Many Extension Headers) [RFC8883] to the packet's source address.

Note that a host may enforce extension header ordering for all extension headers in a packet, but an intermediate node may only enforce ordering for extension headers up to and including the Routing Header.

5. Router limits for receiving extension headers

A router may establish limits for processing packets with received extension headers. If a limit is exceeded, routers SHOULD still forward the packet and SHOULD NOT drop packets because a limit is exceeded.

The requirements for limits related to a router receiving packets with extension headers are:

* If a router needs to parse the upper layer protocol, for instance to deduce the transport layer port numbers, it MUST be able to correctly forward packets that contain an IPv6 header chain of 104 or fewer bytes, or equivalently, a router MUST be able to process a packet with an aggregate length of extension headers less than or equal to 64 bytes.

* If a router needs to parse the upper layer protocol, for instance to deduce the transport layer port numbers, it MUST be able to correctly forward a packets containing eight or fewer extension headers that precede the transport layer header.

* A router MAY limit the number of non-padding Hop-by-Hop options that it processes. If a packet is received with a Hop-by-Hop Options header having a number of non-padding options than exceeds the limit, then the router SHOULD stop processing the Hop-by-Hop Option header and ignore any Hop-by-Hop options beyond the limit. It is NOT RECOMMENDED that a router discards the packet because the limit is exceeded, however if it does then the router MAY send an ICMP Parameter Problem message with code 9 (Too Many Options in Extension Header) [RFC8883] to the packet's source address.

* A router MAY set a limit on the maximum length of a Hop-by-Hop Options header. If a packet is received with a Hop-by-Hop Options header having a length that exceeds the limit, then the router SHOULD either: 1) ignore the Hop-by-Hop Options extension header and forward the packet normally; or 2) process Hop-by-Hop options that are contained within the extent of length limit, ignore any Hop-by-Hop options beyond the limit, and forward the packet normally. It is NOT RECOMMENDED that the router discards the packet because the limit is exceeded, however if it does then the router MAY send an ICMP Parameter Problem message with code 6 (Extension Header Too Big) [RFC8883] to the packet's source address.

* A router MAY limit the number of consecutive bytes of padding in PAD1 or PADN options in a Hop-by-Hop Options header to 7. If the limit is enabled and a packet is received and there are more than 7 consecutive bytes of padding, then the router SHOULD stop processing the Hop-by-Hop Option header and ignore any Hop-by-Hop options beyond the limit. It is NOT RECOMMENDED that the router discards the packet because the limit is exceeded, however if it does then the router MAY send an ICMP Parameter Problem message with code 9 (Too Many Options in Extension Header) [RFC8883] to the packet's source address.

* A router MAY disallow consecutive padding options, either PAD1 or PADN, to be present in the Hop-by-Hop Options header. If a packet is received with consecutive padding options that are disallowed by the router, then the router SHOULD stop processing the Hop-by-Hop Option header and ignore any Hop-by-Hop options beyond the limit. It is NOT RECOMMENDED that the router discards the packet because the limit is exceeded, however if it does then the router MAY send an ICMP Parameter Problem message with code 7 (Too Many Options in Extension Header) [RFC8883] to the packet's source address.

6. Security Considerations

Security issues with IPv6 extension headers are well known and have been documented in several places including [RFC6398], [RFC6192], [RFC7045], [RFC4942], and [RFC9098].

Of particular concern is a Denial-of-Service attack (DOS) wherein an attacker sends many Hop-by-Hop options or Destination options in a packet for the purposes of forcing receivers to consume inordinate resources processing packets. Since there is no hard limit on the number of options in an extension header, it is conceivable that an attacker could craft MTU sized packets with hundreds of small Hop-by-Hop or Destination options where the option type is chosen to be one

that will be unknown to receivers and the higher order type bits are set to 00 to indicate that an unknown option is ignored. A receiver attempting to process all the options in such packet would require a lot of resources as TLV processing is notoriously difficult to do efficiently. The potential for this DOS attack exists routers, hosts, and intermediate nodes. Routers are susceptible to the attack using Hop-by-Hop options, hosts are susceptible using Hop-by-Hop options or Destination options, and intermediate nodes are susceptible using Hop-by-Hop options or Destination options before the Routing Header. Also note, the threat exists for both software and hardware implementations.

This specification addresses the DOS concern of extension headers and options in extension headers by allowing receivers to configure limits on the length or number of extension headers or options that they process. Such limits cap the amount of processing needed for extension headers and hence mitigate the DOS concerns of extension headers. These limits may be set for hosts, routers, and intermediate nodes.

This specification does not introduce any new security concerns.

7. Acknowledgments

The author would like to thank Brian Carpenter, Bob Hinden, Nick Hilliard, Gorry Fairhurst, Darren Dukes, Jen Linkova, Ole Troan, and Vasilenko Eduard for their comments and suggestions that improved this specification.

8. References

8.1. Normative References

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8504] Chown, T., Loughney, J., and T. Winters, "IPv6 Node Requirements", BCP 220, RFC 8504, DOI 10.17487/RFC8504, January 2019, <https://www.rfc-editor.org/info/rfc8504>.

[RFC8883] Herbert, T., "ICMPv6 Errors for Discarding Packets Due to Processing Limits", RFC 8883, DOI 10.17487/RFC8883, September 2020, <https://www.rfc-editor.org/info/rfc8883>.

8.2. Informative References

[APNIC] Huston, G., "IPv6 Extension headers revisited", October 2022, <https://blog.apnic.net/2022/10/13/ipv6-extension-headers-revisited/>.

[Cus23a] Custura, A. and G. Fairhurst, "Internet Measurements: IPv6 Extension Header Edition", IEPG, IETF-116 , March 2023, <http://www.iepg.org/2023-03-26-ietf116/eh.pdf>.

[Cus23b] Custura, A., Secchi, R., Boswell, E., and G. Fairhurst, "Is it possible to extend IPv6?", Computer Communications X, October 2023, <https://www.sciencedirect.com/science/article/pii/S0140366423003705>.

[I-D.ietf-6man-hbh-processing] Hinden, R. M. and G. Fairhurst, "IPv6 Hop-by-Hop Options Processing Procedures", Work in Progress, Internet-Draft, draft-ietf-6man-hbh-processing-12, 21 November 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-6man-hbh-processing-12>.

[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>.

[RFC4942] Davies, E., Krishnan, S., and P. Savola, "IPv6 Transition/Co-existence Security Considerations", RFC 4942, DOI 10.17487/RFC4942, September 2007, <https://www.rfc-editor.org/info/rfc4942>.

[RFC6192] Dugal, D., Pignataro, C., and R. Dunn, "Protecting the Router Control Plane", RFC 6192, DOI 10.17487/RFC6192, March 2011, <https://www.rfc-editor.org/info/rfc6192>.

[RFC6398] Le Faucheur, F., Ed., "IP Router Alert Considerations and Usage", BCP 168, RFC 6398, DOI 10.17487/RFC6398, October 2011, <https://www.rfc-editor.org/info/rfc6398>.

[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>.

[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>.

[RFC9098] Gont, F., Hilliard, N., Doering, G., Kumari, W., Huston, G., and W. Liu, "Operational Implications of IPv6 Packets with Extension Headers", RFC 9098, DOI 10.17487/RFC9098, September 2021, <https://www.rfc-editor.org/info/rfc9098>.

Appendix A. Deriving default limits

This appendix provides an explanation and justification for the recommended default values for limits defined in this specification. The derived default values are based on current capabilities in deployment, expectations for extensibility, and an extrapolation of needs for future extensibility.

A.1. Limits on number of options

The default limit for the number of non-padding Hop-by-Hop or Destination options is 8. This matches the default value in [RFC8504]. At the time of writing, it is observed that in the almost thirty year history of IPv6 there are only thirteen defined non-deprecated Destination options and Hop-by-Hop options and three temporarily assigned. Extrapolating for increased growth and new options, a default limit of 8 should be adequate for the foreseeable future.

A.2. Limits on length

The default limit for the IPv6 header chain is 104 bytes. From this value the default length limit for Hop-by-Hop and Destination options headers, sixty-four bytes, and the default length limit for a Hop-by-Hop or Destination option, sixty bytes, can be deduced.

The 104 byte limit is derived from an assumed parsing buffer size of 128 bytes. A parsing buffer is a memory buffer in many router implementations that allows header processing in the high performance processing fast path. The typical sizes for parsing buffers are 64, 128, 256, or 384 bytes. When a packet is received by a router, the headers of the packet, up to the size of the parsing buffer, are loaded into the parsing buffer. If all the headers that the router needs to process fit within the parsing buffer then the packet can be processed in the fast path. If the necessary headers don't fit in the parsing buffer then a router may either defer processing to a CPU slow path or may just drop the packet.

A de facto requirement of many routers is that they need to process transport layer headers in packets. In particular, a router may inspect the port numbers of transport layer header, such as TCP or UDP, to perform ECMP or port filtering. Typically, a router would need to read at least the first four bytes of the transport layer header which contains the port numbers, so these bytes would need to be in the parsing buffer for a packet.

The default IPv6 header chain limit is derived from the expected size of parsing buffers assuming that there is space to accommodate the first bytes of the transport layer header in the parsing buffer. The IPv6 header chain limit in this specification assumes a common parsing buffer size of 128 bytes. Recent data [APNIC] and [Cus23a] suggests that 128 byte parsing buffers are common and feasible on the Internet. From [APNIC]:

> The experiment used five [Destination Option] extension header lengths (8, 16, 32, 64 and 128 bytes), and in our case, the 8-, 16- and 32-byte headers had the greatest success rates, while the two larger sizes experienced greater drop rates. There is nothing obvious in the Linux source code that could explain this behaviour, unlike the PadN issues. That tends to indicate that the size-related differential response for DST Extension header handling might be due to network equipment behaviours rather than host platform behaviours.

Per [APNIC], the drop rate for Destination Options with sizes 8, 16, and 32 bytes was about 30%. The drop rates for Destination Options with size 64 was about 40%, and the drop rate with size 128 bytes was about 85%. As [APNIC] mentions, these differences are most likely due to network equipment. We can extrapolate from this data the effects of a parsing buffer. Support for 128 byte extension headers implies at least a 256 byte parsing buffer, support for 64 byte extension headers implies at least a 128 byte parsing buffer, and support for smaller extension headers implies a smaller parsing buffer.

Based on this analysis, assuming common support for a 128 byte parsing buffer seems reasonable. A 128 byte parsing buffer accommodates 104 byte IPv6 header chain length including 64 bytes of extension headers. Note that 32 byte extension headers did have a bit more success than 64 bytes extension headers (30% versus 40% drop rate), however requiring support for just 32 bytes of extension header would significantly limit the utility of extension headers. Therefore, 128 bytes is chosen as the expected minimum parsing buffer size on the Internet.

The 128 byte parsing buffer would be expected to at least contain:

* 16 bytes for a Layer 2 header (for instance an Ethernet header)

* 40 bytes for the IPv6 header

* 64 bytes for the extension headers

* 8 bytes for the transport layer (i.e the first eight bytes of the transport layer header)

This scheme thus establishes a requirement that Internet devices must be capable of correctly processing packets with up to sixty-four bytes of extension headers, and subsequently it establishes a requirement that a host shouldn't send packets with more than sixty-four bytes of extension headers unless it known that all the nodes in the path can process packets with larger extension headers and a larger IPv6 header chain. Note that this establishes a global minimum baseline requirement across the Internet; within a limited domain higher limits could freely be applied.

A.3. Padding limits

[RFC4942] establishes that the number of consecutive bytes in padding options in Hop-by-Hop and Destination Options headers should be limited to no more than seven:

> There is no legitimate reason for padding beyond the next eight octet boundary since the whole option header is aligned on an eight-octet boundary but cannot be guaranteed to be on a 16 (or higher power of two)-octet boundary.

[RFC8504] established that a receiver MAY limit the number of consecutive padding bytes in a received packet to seven. This specification expands on those limits to allow routers and intermediate hosts to set a limit. Additionally, requirements are established for sending hosts that they shouldn't set more than seven consecutive bytes of padding.

Note that limit on seven consecutive bytes of padding is "hardcoded" and enforced in the Linux networking stack.

In addition to a limit on the number of consecutive bytes of padding, this specification allows a receiver disallow consecutive padding options. The rationale is that a single PAD1 or PADN option can be used to provide 1 to 257 bytes of padding which is sufficient for any practical use case. Correspondingly, this specification also recommends that a sender does not send a packet with consecutive padding options.

Author's Address

Tom Herbert
SiPanda
Santa Clara, CA,
United States of America
Email: tom@herbertland.com