Workgroup: Network Working Group
Internet-Draft: draft-ietf-6man-eh-limits-04
Updates: 8200, 8504 (if approved)
Published: 6 May 2023
Intended Status: Standards Track
Expires: 7 November 2023
Authors: T. Herbert
SiPanda

Limits on Sending and Processing IPv6 Extension Headers

Abstract

This specification defines various limits that may be applied to receiving, sending, and otherwise processing packets that contain IPv6 extension headers. The need for such limits is pragmatic to facilitate interoperability amongst hosts and routers in the presence of extension headers and thereby increasing the feasibility of deployment of extension headers. The limits described herein establish the minimum baseline of support for use of extension headers in the Internet. If it is known that all communicating parties for a particular communication, including end hosts and any intermediate nodes in the path, are capable of supporting more than the baseline then these default limits may be freely exceeded. When published, this document updates [RFC8200] and [RFC8504].

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 MTU). Similarly, variable length extension headers typically do not have prescribed limits such as limits on the number of Hop-by-Hop or Destination options in a packet. The lack of limits essentially requires implementations to handle every conceivable usage of the protocol, including a myriad

of use cases those are obviously outside the realm of ever being realistic or useful in real world deployment.

The lack of limits and the requirements for supporting a virtually open-ended protocol have led to a significant lack of support and deployment of extension headers [RFC7872]. Instead of attempting to satisfy the protocol requirements concerning extension headers, some router and middlebox vendors have opted to either invent and apply their own ad hoc limits, relegate packets with extension headers to slow path processing, or have gone so far as to summarily discard all packets with extension headers [RFC9098]. The net result 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 1500 byte MTU sized packet could legally contain a Hop-by-Hop Options header with over seven hundred two byte options. There is no use case for this other than a Denial of Service attack where an attacker simply creates packets with hundreds of small unknown Hop-by-Hop options with the two high order bits in the option type set to 00 meaning to skip the unknown option. Any node in the path that attempts to dutifully process all these options per the requirements of [RFC2460] would be easily overwhelmed by the processing needed to parse these options (this is true for both hardware or software implementations).

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

When published, this document updates requirements pertaining to extension header processing in [RFC8200] and [RFC8504].

1.1. Related work

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

[RFC8200] relaxed the requirement that all nodes in the path must process 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 host 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 feature is supported, the maximum number SHOULD be configurable, and the default value SHOULD be set to 8.

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

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

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

2. Overview and motivation of extension header limits

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

2.1. Types of nodes

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

* Hosts: The source of an IPv6 packet, as addressed by the source address; or the final destination node of a packet as addressed by the destination address in a packet with no Routing header or as addressed by last segment in a Routing header

* Intermediate destination: An intermediate destination node in a Routing header as addressed by the destination address of a packet with a Routing header where the address is not the address of the last segment in the Routing header

* Intermediate nodes: A router on the path that is not addressed by a packet's destination address

2.2. Types of limits

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

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

2.2.1. Limits on extension header length

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

2.2.2. Limits on option length

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

2.2.3. Limits on number of extension headers

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

2.2.4. Limits on number of options

Limits may be established for the number of options sent or received (specifically applicable to Hop-by-Hop Options headers and Destination Options headers). The need for this limit arises from

the fact that [RFC8200] does not specify a limit. Requiring nodes to process packets with tens or hundreds of options has no foreseeable use cases in deployment except as a denial of service attack. [RFC8504] has proposed such a limit for host processing of a Hop-by-Hop Options header or Destination Options header with a default of eight options. This specification extends that limit to be applicable to intermediate nodes. Specific limits may be established for the number of non-padding options or the number of all options including padding.

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

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

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

2.2.5. Limits on padding options

[RFC8200] defines PAD1 and PADN options that respectively provide one byte or N bytes of padding in a Hop-by-Hop Options or Destination Options header. The purpose of padding is to properly align the following non-padding option to its expected alignment, or to add padding after the last Destination or Hop-by-Hop option so that the length of the extension header is a multiple of eight bytes as required by [RFC8200]. [RFC8504] defines limits on number of bytes used for consecutive padding where the amount of padding between options or at the end of the extension header is no more than seven bytes; this limit is sufficient to align any following data after the padding to eight bytes. These limits are extended to be applicable to intermediate nodes.

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

2.2.6. Limit on IPv6 header chain length

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

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

To derive a size limit for the IPv6 header chain, we need to take into account headers in a packet that might be subject to DPI which include the link layer header through at least the pertinent fields

of the transport layer header. The most common required information is the transport layer port numbers which typically occupy the first four bytes of the transport headers (in TCP, UDP, SCTP, DCCP, etc.). Inspection of port numbers may be needed for stateless load balancing as well as port filtering. There are middleboxes that may need to inspect more of transport layer headers or the transport payload, however those can be considered specialized devices that perform work beyond simple packet forwarding and filtering and hence should have more capabilities for DPI.

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

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

* 16 bytes for a Layer 2 header (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 all Internet devices are capable of correctly processing packets with up to sixty-four bytes of extension headers, and subsequently it establishes a requirement that a host shouldn't send packets with more than sixty-four bytes of extension headers. Note that this establishes a global baseline requirement across the Internet; within a limited domain higher limits could be applied.

128 bytes is likely the minimal useful parsing buffer size in deployment today. Devices performing a very narrow DPI could conceptually use a smaller parsing buffer, for instance that could be as small as sixty-four bytes which accommodates an L2 header, IPv6 header, and eight bytes of transport header; however, such a device would be extremely limited in capabilities and if they do exist they are likely legacy devices that will eventually be

decommissioned. Many routers now have the capability to perform DPI into encapsulation headers which implies they already have a larger parsing buffer than this baseline minimum.

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

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

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

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

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

2.3. Action when limit is exceeded

For each limit that is defined, an action is specified for when the limit is exceeded. The appropriate action depends on whether the processing node is the destination host, an intermediate

destination, or an intermediate node. For a destination host, the typical action to take when a limit is exceeded is to discard the packet. This is appropriate since the destination host is required to process all of the headers in a packet, and if a limit is exceeded then it cannot process the packet so there is no other alternative but to discard.

For intermediate nodes, the typical action to take when a limit is exceeded is to stop processing headers at the point the limit is reached and to forward the packet on. If an intermediate node needs to access transport layer information it may continue inspecting extension headers, but not processing them, after a limit has been reached for the purposes of locating the transport layer header. [RFC8200] allows that an intermediate node may not process the Hop-by-Hop Options headers, therefore an intermediate node may ignore all of the Hop-by-Hop options in a packet. This specification expands on that requirement to allow an intermediate node to process some arbitrary subset of consecutive Hop-by-Hop options in the TLV list and to ignore the following ones. In the case of an egregious violation of a limit, for instance an attacker sends three hundred options in a packet, the destination host can decide if the appropriate response is to drop (the destination host must process all options). Note that this provision motivates the sender to place Hop-by-Hop options in the extension header so that those considered more important are placed first. It should also be noted that [RFC8504] sets a default limit of eight; this specification adds a counterpart for sending hosts that they shouldn't send more than eight Hop-by-Hop options by default.

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

2.4. Design Philosophy

The limits defined in this document are applicable to both senders and receivers. With a few exceptions as described below, the limits described herein are optional to configure and enforce. If a limit is configurable there is a suggested default value.

A sender of extension headers should generally be conservative in its use of extension headers to maximize the chances of packets being delivered to their destination. Default values for sending limits are assumed to be useful in arbitrary environment such as the public Internet, that is they can be considered "baseline limits".

These limits may be relaxed if a sender has a priori information that all possible nodes in path will properly handle packets that exceed the baseline limits. In particular, if a sender is sending in a limited domain, it might be known that all nodes in the limited domain have sufficient capabilities to handle packets exceeding the baseline limits.

Specific mechanisms for a host to determine that baseline limits for extension headers may be exceeded are out of scope for this document. Conceivably, this determination could be done by configuration, capabilities probing, or applying historical knowledge that all intermediate nodes in the path and the destination node are capable of handling packets that exceed the baseline limits.

Receivers of extension headers should be liberal in accepting packets with extension headers, however per this document they may ignore extension headers or options within extension headers (in accordance with [RFC8200]). In particular, the philosophy of this specification is that intermediate nodes should not drop packets with extension headers solely on the basis that they don't have sufficient capabilities to process all the headers in a packet. As such, intermediate nodes may define arbitrarily restrictive limits on what they process with regards to extension headers as long as the action taken when those limits are exceeded is to ignore items beyond the limit. Hosts are more constrained in this regard since they generally can't correctly process a packet without processing all the headers, so when limits are exceeded on a host, packets should be discarded. It should be noted that hosts stacks inherently have more processing capabilities than intermediate nodes, so it is expected that they should be able to support higher limits for processing extension headers.

This specification does specify one hard requirement for receiving nodes, namely nodes must be able to properly handle packets having an IPv6 header chain length up to 104 bytes. This requirement acknowledges that some intermediate nodes perform deep packet inspection to extract information from transport layer headers [RFC9098]. Often a node that requires parsing transport layer information will have a fixed sized "parsing buffer" to contain packet headers. If the transport layer headers within a packet are beyond the extent of the parsing buffer then an implementation might take some detrimental action such as arbitrarily dropping packets. To this end, this specification requires that any intermediate node that requires access to to transport layer header must minimally be able to parse at least 128 bytes of headers, from which the 104 byte limit for the IP header chain is derived.

3. Requirements

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

The requirements in this section update the processing requirements specified in Section 4 of [RFC8200]; in particular, requirements for how many Hop-by-Hop options an intermediate node must process are updated.

The requirements in this section update section 5.3 of [RFC8504] by extending the limits applicable to end host nodes to be applicable to intermediate nodes as well.

3.1. List of limits

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

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

3.2. Host requirements

3.2.1. Sending extension headers

The requirements are:

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

final destination host is able to process a greater number of options.

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

* A host MUST NOT send a packet with an extension header larger than 64 bytes unless it has explicit knowledge that all nodes that might process the extension header are capable of processing a larger header.

* A host MUST NOT send a packet with a Destination option or Hop-by-Hop option with Data Length greater than 60 bytes unless it has explicit knowledge that all nodes that might process the option are capable of processing ones with a larger Data Length.

* A host node MUST NOT send a packet with an IPv6 header chain larger than 104 bytes unless it has explicit knowledge that all nodes in the path are capable of properly handling packets with larger header chains. This requirement is equivalently stated as a host MUST NOT send a packet with more than 64 bytes of aggregate extension headers.

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

* A host MUST NOT send a PadN option in a Hop-by-Hop Options header or Destination Options header with total length of more than seven bytes.

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

* A host node MUST NOT send more than 16 options (padding or non-padding) in a Hop-by-Hop Options header unless it has explicit knowledge that the final destination host is able to process a greater number of options. Note that if the above requirements on a host sending non-padding Hop-by-Hop options and requirements on padding are met, then this requirement is implicitly satisfied.

3.2.2. Receiving extension headers

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

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

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

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

* A host node MAY set a limit on the length of an extension header. If this limit is supported then the limit SHOULD be configurable and the limit MUST be greater than or equal to 64 bytes. The length limits for different extension headers MAY be separately configurable.

* A host node MAY set a limit on the Data Length of a Hop-by-Hop or Destination option. If this limit is supported then the limit SHOULD be configurable, and the limit MUST be greater than or equal to 60 bytes. The limits for Destination options and Hop-by-Hop options MAY be separately configurable. If a packet is received and a Hop-by-Hop or Destination option has a length that exceeds the limit, then the packet SHOULD be discarded and an ICMP Parameter Problem with code 10 MAY be sent to the packet's source address.

* A host MAY limit the number of consecutive PAD1 options in a Destination Options header or Hop-by-Hop Options header to 7. In this case, if there are more than 7 consecutive PAD1 options present, the packet SHOULD be discarded and an ICMP Parameter Problem with code 10 MAY be sent to the packet's source address

* A host MAY limit the number of bytes in a PADN option to be less than 8. In such a case, if a PADN option is present that has a length greater than 7, the packet SHOULD be discarded and an ICMP Parameter Problem with code 10 MAY be sent to the packet's source address.

* A host MAY set a limit on the maximum length of a Destination Options header or Hop-by-Hop Options header. This value SHOULD be configurable, and if the limit is used then the limit MUST be greater than or equal to 64 bytes. If a packet is received and the length of the Destination Options header or Hop-by-Hop Options header exceeds the length limit, then the packet SHOULD be discarded and an ICMP Parameter Problem with code 6 MAY be sent to the packet's source address.

* A host node MAY set a limit on the maximum length of the IPv6 header chain, or equivalently a host MAY set a limit on the aggregate length of extension headers in a packet. If the limit is used then it MUST be greater than or equal to 104 bytes, or, equivalently, the limit on aggregate header extension length MUST be greater than or equal to 64 bytes. If a packet is received and the aggregate length of the IPv6 header chain exceeds the limit then the packet SHOULD be discarded and an ICMP Parameter Problem with code 7 MAY be sent to the packet's source address.

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

3.3. Intermediate node and intermediate destination requirements

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

* An intermediate node MUST be able to correctly forward packets that contain an IPv6 header chain of 104 or fewer bytes, or equivalently an intermediate node MUST be able to process a packet with an aggregate length of extension headers less than or equal to 64 bytes.

* Per [RFC8200] an intermediate node MAY be configured not to process Hop-by-Hop Options headers. If a node is configured as such and a packet with a Hop-by-Hop Options header is received, the extension header MUST be be skipped and the packet MUST otherwise be properly processed and forwarded.

* An intermediate node MAY limit the number of non-padding Hop-by-Hop options that it processes. If a limit is exceeded, that is a Hop-by-Hop Options header contains more non-padding options than are configured to process, the intermediate node SHOULD stop processing the Hop-by-Hop Option header and ignore any Hop-by-Hop options beyond the limit. It is NOT RECOMMENDED that an intermediate node discards the packet because the limit is exceeded, however if it does so then the intermediate node MAY send an ICMP Parameter Problem with code 10 to the packet's source address.

* An intermediate node MAY limit the number of Hop-by-Hop options (padding or non-padding) that it processes. If a limit is exceeded, that is a Hop-by-Hop Options header contains more non-padding options than are configured to process, the intermediate node SHOULD stop processing the Hop-by-Hop Options header and ignore any Hop-by-Hop options beyond the limit. It is NOT RECOMMENDED that the intermediate node discards the packet because the limit is exceeded, however if it does so then the intermediate node MAY send an ICMP Parameter Problem with code 10 to the packet's source address.

* If an intermediate node encounters an unknown Hop-by-Hop option and the two high order bits are not 00 then the node SHOULD immediately stop processing the Hop-by-Hop Options header and ignore any Hop-by-Hop options beyond the unknown option. An intermediate node MAY either elect to discard the packet and MAY send an ICMP Parameter Problem per the requirements of [RFC8200] and [I-D.ietf-6man-hbh-processing]; or the intermediate node MAY forward the packet and effectively disregard the high order two bits in the option type. The motivation for this requirement is to simplify processing at intermediate nodes. Note, that if the high order two bits are non-zero for an option that is unknown to the destination host then the packet will be discarded since the destination host is required to process all Hop-by-Hop options in a packet or to discard a packet if its limit for maximum number of options to process is exceeded.

* An intermediate node MAY set a limit on the maximum length of a Hop-by-Hop Options header. This value SHOULD be configurable. If this limit is exceeded, that is a packet has an extension header larger then the limit, then the intermediate node 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 intermediate node discards the packet because the limit is exceeded, however if it does so then the intermediate node MAY send an ICMP Parameter Problem with code 10 to the packet's source address.

3.4. Intermediate destination requirements

The following are requirements specific to intermediate destinations pertaining to the processing of a Destination Options header before the Routing header. For processing a Hop-by-Hop Options header at an intermediate destination, the requirements for processing them at an intermediate node are assumed.

* An intermediate destination MAY limit the maximum length of a Destination Options header before the Routing header. This value SHOULD be configurable, and the default is to accept options of any length. If a limit is defined is MUST be at least 64 bytes. If the limit is exceeded then the intermediate destination SHOULD discard the packet and MAY send an ICMP Parameter Problem with code 6 to the packet's source address.

* An intermediate destination node MAY limit the number of non-padding options in a Destination Options header before the Routing header. If this limit is supported then the maximum number SHOULD be configurable and the limit MUST be greater than or equal to 8. If a limit is exceeded, that is a packet contains more non-padding options than are configured to process, the intermediate destination node SHOULD discard the packet and MAY send an ICMP Parameter Problem with code 10 to the packet's source address.

* An intermediate destination node MAY limit the number of options (padding or non-padding) in a Destination Options header before the Routing header. If this limit is supported then the maximum number SHOULD be configurable and the limit MUST be greater than or equal to 16. If a limit is exceeded, that is a packet contains more options than are configured to process, the intermediate destination node SHOULD discard the packet and MAY send an ICMP Parameter Problem with code 10 to the packet's source address.

* An intermediate destination MAY limit the total number bytes in consecutive PAD1 options in a Destination Options header before the Routing header to 7. If the limit is exceeded, that is there are more than seven bytes in consecutive PAD1 or PADN options present, the intermediate destination node SHOULD discard the packet and MAY send an ICMP Parameter Problem with code 10 to the packet's source address.

* An intermediate destination MAY limit the number of bytes in a PADN option in a Destination Option header before the Routing header to be less than 8. In such a case, if a PADN option is present that has a length greater than 7, the packet SHOULD be discarded and the intermediate destination node SHOULD discard the packet and MAY send an ICMP Parameter Problem with code 10 to the packet's source address.

* An intermediate MAY limit the maximum length of a Destination Options header before the Routing header. If this limit is supported then the limit SHOULD be configurable and the limit MUST be greater than or equal to 64 bytes. If a packet is received and the length of the a Destination Options header before the Routing header exceeds the length limit, the intermediate destination node SHOULD discard the packet and MAY send an ICMP Parameter Problem with code 10 to the packet's source address.

4. Security Considerations

Security issues with IPv6 Hop-by-Hop options are well known and have been documented in several places, including [RFC6398], [RFC6192], [RFC7045] and [RFC9098].

Of particular concern is a Distributed Denial-of-Service attack (DDOS) 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 the receiver 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 hard to do efficiently (in either hardware or software).

This document addresses the DDOS concern of extension headers and options in extension headers by allowing receivers to configure limits the number of extension headers or options that they process. Such limits cap the amount of processing needed for extension headers and hence mitigate the DDOS concerns of extension headers.

This document does not otherwise introduce any new security concerns.

5. Acknowledgments

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

6. References

6.1. 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>.

6.2. Informative References

[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-08, 30 April 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-6man-hbh-processing-08>.

[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>.

[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>.

[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>.

Author's Address

Tom Herbert
SiPanda
Santa Clara, CA,
United States of America

Email: tom@herbertland.com