Workgroup: Network Working Group
Internet-Draft: draft-ietf-6man-mtu-option-02
Published: 14 September 2020
Intended Status: Experimental
Expires: 18 March 2021
Authors: R. Hinden
Check Point Software
G. Fairhurst
University of Aberdeen

IPv6 Minimum Path MTU Hop-by-Hop Option

Abstract

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

This Hop-by-Hop option is intended to be used in environments like Data Centers and on paths between Data Centers, to allow them to better take advantage of paths able to support a large Path MTU. The method could also be useful in other environments, including the general Internet.

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 18 March 2021.

Table of Contents

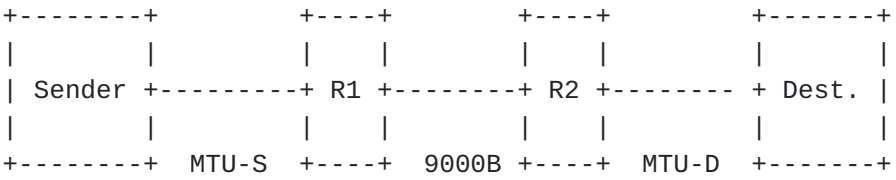
1. Introduction

This draft proposes a new IPv6 Hop-by-Hop Option to be used to record the minimum of the Maxmim Transmission Unit (MTU) along the forward path between the source and destination hosts. The source host creates a packet with this Hop-by-Hop Option and fills the Min-PMTU field in the option with the value of the MTU for the outbound link that will be used to forward the packet towards the destination host. This option is carried in the IPv6 Hop-by-Hop Options Header.

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

1.1. Example Operation

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



Three scenarios are described:

- Scenario 1, considers all links to have an 9000 byte MTU and the method is supported by both routers. The PMTU is therefore 9000 bytes.

- Scenario 2, considers the link to the destination host (MTU-D) to have an MTU of 1500 bytes. This is the smallest MTU, router R2 updates the Min-PMTU to 1500 bytes and the method correctly updates the PMTU to 1500 bytes. Had there been another smaller MTU at a link further along the path that also supports the method, the lower MTU would also have been detected.

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

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

These scenarios are summarized in the table below.

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

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

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

*The Hop-by-Hop Options header is not inserted or deleted, but may be examined or processed by any node along a packet's delivery path, until the packet reaches the node (or each of the set of nodes, in the case of multicast) identified in the Destination Address field of the IPv6 header. The Hop-by-Hop Options header, when present, must immediately follow the IPv6 header. Its presence is indicated by the value zero in the Next Header field of the IPv6 header.

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

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

2. Motivation and Problem Solved

The current state of Path MTU Discovery on the Internet is problematic. The mechanisms defined in [RFC8201] are known to not

work well in all environments. This fails to work in various cases, including when nodes in the middle of the network do not send ICMP PTB messages, or rate-limited messages to the point of not making them a useful mechanism, or do not have a return path to the source host.

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

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

Sending small packets can limit performance, e.g., when packet processing is limited by the packet rate. The potential of multi-gigabit Ethernet will not be realized if the packet size is limited to 1280 bytes, because this exceeds the packet per second rate that most nodes can process. For example, the packet per second rate required to reach wire speed on a 10G Ethernet link with 1280 byte packets is about 977K packets per second (pps), vs. 139K pps for 9000 byte packets. A significant difference.

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

3. Requirements Language

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

4. Applicability Statements

This Hop-by-Hop Option header is intended to be used in environments such as Data Centers and on paths between Data Centers, to allow a

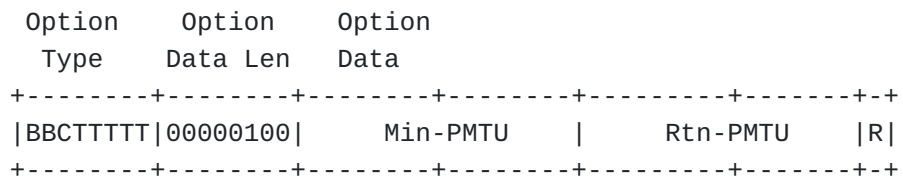
host to better take advantage of a path that is able to support a large PMTU.

The design of the option is sufficiently simple that it could be executed on a router's fast path. A strong pull from router vendors customers will be required to create critical mass for this to happen. This could initially be the case for connections within and between Data Centers.

The method could also be useful in other environments, including the general Internet, if and when this Hop-by-Hop Option is supported on these paths.

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

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



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

BB 00 Skip over this option and continue processing.

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

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

Length: 4 The size of the each value field in Option Data field supports PMTU values from 0 to 65,535 octets.

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

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

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

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

6. Router, Host, and Transport Behaviors

6.1. Router Behaviour

Routers that are not configured to support Hop-by-Hop Options SHOULD ignore this option and SHOULD forward the packet.

Routers that support Hop-by-Hop Options, but that are not configured to support this option SHOULD ignore the option and SHOULD forward the packet.

Routers that recognize this option SHOULD compare the value of the Min-PMTU field with the MTU configured for the outgoing link. If the MTU of the outgoing link is less than the Min-PMTU, the router rewrites the Min-PMTU in the Option to use the smaller value.

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

Discussion:

* The design of this option makes it feasible to be implemented within the fast path of a router, because the processing requirements are minimal.

6.2. Host Behavior

When requested to send an IPv6 packet with the Minimum Path MTU option, the source host includes the option in an outgoing packet. The source host SHOULD fill the Min-PMTU field with the MTU configured for the link over which it will send the packet on the next hop towards the destination host. If this value is not updated, the field MUST be set to zero.

The source host SHOULD set the Rtn-PMTU field to the cached value of the reported Min-PMTU value for the flow (see Section 6.3.3). If this value is not set, for example, because there is no cached reported Min-PMTU value, the field MUST be set to zero.

The source host MAY request the destination host to return the reported Min-PMTU value by setting the R-Flag in the option of an outgoing packet.

6.3. Transport Behavior

6.3.1. Including the Option in an Outgoing Packet

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

When it includes the option, the host supplies the previously cached value of the received Minimum Path MTU for the flow to set the Rtn-PMTU field (see Section 6.3.3). If a valid cached received Minimum Path MTU is not available, the Rtn-PMTU field value MUST be set to zero.

The source host MAY request the destination host to send a packet carrying the option by setting the R-Flag. The R-Flag SHOULD NOT be

set when the Minimum Path MTU Option was sent solely to feedback the return Path MTU.

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

Discussion:

- In the case of TCP, the option could be included in packets carrying a SYN segment as part of the connection set up, or can periodically be sent in packets carrying other segments. Including this packet in a SYN could increase the probability that the SYN segment is lost when routers on the path drop packets with this option (see Section 6.3.5). NOTE: A TCP connection can also negotiate the Maximum Segment Size (MSS), which acts as an upper limit to the packet size that can be sent by a TCP sender.

- The use with datagram transport protocols (e.g., UDP) is harder to characterize because applications using datagram transports range from very short-lived (low data-volume applications) exchanges, to longer (bulk) exchanges of packets between the source and destination hosts [RFC8085].

- Simple-exchange protocols (i.e., low data-volume applications [RFC8085] that only send one or a few packets per transaction, might assume that the PMTU is symmetrical. That is, the PMTU is the same in both directions, or at least not smaller for the return path. This optimisation does not hold when the paths are not symmetric.

- The use of this option with DNS and DNSSEC over UDP ought to work, for symmetric paths. The DNS server will learn the PMTU from the DNS query messages. If the Rtn-PMTU value is smaller, then a large DNSSEC response might be dropped and the known problems with PMTUD will occur. DNS and DNSSEC over transport protocols that can carry the PMTU ought to work.

- Applications that use Anycast should include this option in all packets, because the actual destination host will vary due to the nature of Anycast.

6.3.2. Validation by the Upper Layer Protocol

An upper layer protocol (e.g., transport endpoint) using this option needs to provide protection from data injection attacks by off-path devices [RFC8085]. This requires a method to assure that the

information in the Option Data is provided by a node on the path. For example, a TCP connection or UDP application that maintains the related state and uses a randomised ephemeral port would provide this basic validation to protect from off-path data injection. IPsec [RFC4301] and TLS [RFC8446] provide greater assurance.

The Upper Layer discards any received packet when the packet validation fails. When this packet validation fails, the Upper Layer MUST also discard the associated Option Data from the minimum Path MTU option without further processing.

6.3.3. Receiving the Option

An upper layer protocol that receives a Minimum Path MTU Option iccnluded with a valid packet caches the value of the last received Min-PMTU. This value is specific to the instance of the upper layer protocol (i.e., matching the IPv6 flow ID, port-fields in UDP or the SPI in IPsec [RFC4301], etc), not to the pair of source and destination addresses, because network devices can make forwarding decisions that impact the PMTU of a flow based on the presence and value of the packet's upper layer fields.

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

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

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

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

can be advantage in using this to detect a change in the path characteristics.

Discussion:

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

6.3.4. Using the Rtn-PMTU Field

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

* The actual PMTU can be lower than the Rtn-PMTU value because Min-PMTU field was not updated by a router on the path that did not process the option.

* The actual PMTU may be lower than the Rtn-PMTU value because the there is a layer 2 device with a lower MTU that does not perform IPv6 forwarding.

* The actual PMTU may be larger than the Rtn-PMTU value because of a corrupted, delayed or mis-ordered response. A source host SHOULD ignore a Rtn-PMTU value larger than the MTU configured for the outgoing link.

Using the method has the potential to complete discovery of the correct value in a single round trip time, even over paths that have successive links each configured with a lower MTU.

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

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

6.3.5. Detection of Droping Packets that include the Option

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

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

7. IANA Considerations

No IANA actions are requested in this document.

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

8. Security Considerations

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

8.1. Network Layer Host Processing

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

Network layer option processing is normally done before any upper layer protocol delivery checks are performed. Reception of this packet will incur receive processing as the network stack parses the packet before the packet is delivered to the upper layer protocol.

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

reception of the option apply to all flows between a pair of endpoints.

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

8.2. Validating use of the Option Data

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

A network node on the path has visibility of all packets it forwards. By observing the network packet payload, the node might be able to construct a packet might be validated by the destination host. Such a node would also be able to drop or limit the flow in other ways that could be potentially more disruptive. Authenticating the packet, for example, using IPsec [RFC4301] or TLS [RFC8446] mitigates this attack.

8.3. Direct use of the Rtn-PMTU Value

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

* A direct update of the PMTU using the Rtn-PMTU value could result in an attacker inflating or reducing the size of the host PMTU for the destination. Forcing a reduction in the PMTU can decrease the efficiency of network use, might increase the number of packets/fragments required to send the same volume of payload data, and prevents sending an unfragmented datagram larger than the PMTU. Increasing the PMTU can result in black-holing (see Section 1.1 of [RFC8899]) when the source sends packets larger than the actual PMTU. This persists until the PMTU is next updated.

* The method can be used to solicit a response from the destination host. A malicious attacker could forge a packet that cause the sender to add the option to a packet sent to the source. A forged value of Rtn-PMTU in the Option Data might also impact the remote

endpoint, as described in the previous bullet. This persists until a valid minimum Path MTU option is received. This attack could be mitigated by limiting the sending of the minimum Path MTU option in reply to incoming packets that carry the option.

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

Another way to utilise the Rtn-PMTU value is to indirectly trigger a probe to determine if the path supports a PMTU of size Rtn-PMTU. This approach needs context for the flow, and hence assumes an upper layer protocol that validates the packet that carries the option Section 8.2. This is the case when used in combination with DPLPMTUD [RFC8899]. A set of security considerations result when an option carries malicious data:

* If the forged packet carries a validated option with a non-zero Rtn-PMTU field, the upper layer protocol can utilise the information in the Rtn-PMTU field. A Rtn-PMTU larger than the current PMTU can trigger a probe for a new size.

* If the forged packet carries a non-zero Min-PMTU field, the upper layer protocol would change the cached information about the path from the source. The cached information at the destination host will be overwritten when the host receives another packet that includes a minimum Path MTU option corresponding to the flow.

* Processing of the option could cause a destination host to add the minimum Path MTU option to a packet sent to the source host. This option will carry a Rtn-PMTU value that could have been updated by the forged packet. The impact of the source host receiving this resembles that discussed previously.

8.5. Impact of Middleboxes

There is evidence that some middleboxes drop packets that include Hop-by-Hop options. For example, a firewall might drop a packet that carries an unknown extension header or option. This practice is expected to decrease as the option becomes more widely used. Methods to address this are discussed in Section 6.3.5.

When a forged packet cause a packet to be sent including the minimum Path MTU option, and the return path does not forward packets with this option, the packet will be dropped Section 6.3.5. This attack is mitigated by validating the option data before use and by limiting the rate of responses generated. An upper layer could further mitigate the impact by responding to a R-Flag by including the option in a packet that does not carry application data.

9. Acknowledgments

A somewhat similar mechanism was proposed for IPv4 in 1988 in [RFC1063] by Jeff Mogul, C. Kent, Craig Partridge, and Keith McCloghire. It was later obsoleted in 1990 by [RFC1191] the current deployed approach to Path MTU Discovery.

Helpful comments were received from Tom Herbert, Tom Jones, Fred Templin, Ole Troan, [Your name here], and other members of the 6MAN working group.

10. Change log [RFC Editor: Please remove]

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

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

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

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

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

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

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

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

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

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

*Editorial changes.

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

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

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

*Initial draft.

11. References

11.1. Normative References

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

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

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

[RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <https://www.rfc-editor.org/info/rfc8201>.

11.2. Informative References

[RFC1063] Mogul, J., Kent, C., Partridge, C., and K. McCloghrie, "IP MTU discovery options", RFC 1063, DOI 10.17487/RFC1063, July 1988, <https://www.rfc-editor.org/info/rfc1063>.

[RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <https://www.rfc-editor.org/info/rfc1191>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <https://www.rfc-editor.org/info/rfc2460>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <https://www.rfc-editor.org/info/rfc6438>.

[RFC7637] Garg, P., Ed. and Y. Wang, Ed., "NVGRE: Network Virtualization Using Generic Routing Encapsulation", RFC 7637, DOI 10.17487/RFC7637, September 2015, <https://www.rfc-editor.org/info/rfc7637>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8899] Fairhurst, G., Jones, T., Tüxen, M., Rüngeler, I., and T. Völker, "Packetization Layer Path MTU Discovery for Datagram Transports", RFC 8899, DOI 10.17487/RFC8899, September 2020, <https://www.rfc-editor.org/info/rfc8899>.

[RFC8900] Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", BCP 230, RFC 8900, DOI 10.17487/RFC8900, September 2020, <https://www.rfc-editor.org/info/rfc8900>.

Authors' Addresses

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

Email: bob.hinden@gmail.com

Godred Fairhurst
University of Aberdeen
School of Engineering
Fraser Noble Building
Aberdeen
AB24 3UE
United Kingdom

Email: gorry@erg.abdn.ac.uk