Internet Engineering Task Force (IETF)
Request for Comments: 9268
Category: Experimental
ISSN: 2070-1721

R. Hinden
Check Point Software
G. Fairhurst
University of Aberdeen

August 2022

IPv6 Minimum Path MTU Hop-by-Hop Option

Abstract

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

Status of This Memo

This document is not an Internet Standards Track specification; it is published for examination, experimental implementation, and evaluation.

This document defines an Experimental Protocol for the Internet community. This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are candidates for any level of Internet Standard; see Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9268.

Table of Contents

1. Introduction

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

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

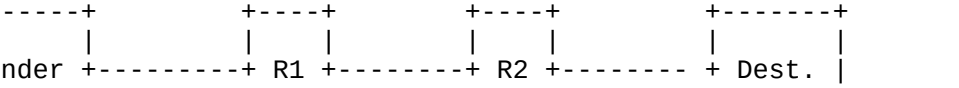
The IPv6 Minimum Path MTU Hop-by-Hop (MinPMTU HBH) Option is designed to work with packet sizes that can be specified in the IPv6 header. The maximum packet size that can be specified in an IPv6 header is 65,535 octets (2^16).

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

The mechanism defined in this document is focused on unicast; it does not describe multicast. That is left for future work.

1.1. Example Operation

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



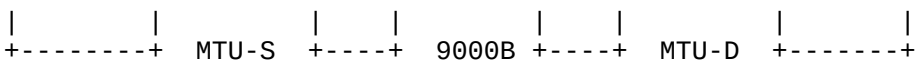


Figure 1: An Example Path between the Source Host and the Destination Host

Three scenarios are described:

* Scenario 1 considers all links to have a 9000 byte MTU, and the method is supported by both routers. The initial Min-PMTU is not modified along the path. Therefore, the PMTU is 9000 bytes.

* Scenario 2 considers the link between R2 and 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 Too Big (PTB) message had not been delivered to the sender [RFC8201].

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

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

Table 1: Three Scenarios That Arise from Using the Path Shown in Figure 1

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

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

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

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

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

2. Motivation and Problem Solved

The current state of Path MTU Discovery on the Internet is problematic. The mechanisms defined in [RFC8201] are known to not work well in all environments. It fails to work in various cases, including when nodes in the middle of the network do not send ICMPv6 PTB messages or rate-limited ICMPv6 messages or do not have a return path to the source host. This results in many transport-layer connections being configured to use smaller packets (e.g., 1280 bytes) by default and makes it difficult to take advantage of paths with a larger PMTU where they do exist. Applications that send large packets are forced to use IPv6 fragmentation [RFC8200], which can reduce the reliability of Internet communication [RFC8900].

Encapsulations and network-layer tunnels further reduce the payload size available for a transport protocol to use. Also, some use cases increase packet overhead, for example, Network Virtualization Using Generic Routing Encapsulation (NVGRE) [RFC7637] encapsulates Layer 2 (L2) packets in an outer IP header and does not allow IP fragmentation.

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

The purpose of this document is to improve the situation by defining a mechanism that does not rely on reception of ICMPv6 PTB 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 mechanisms based on [RFC8201].

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

3. Requirements Language

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

4. Applicability Statements

The Path MTU Option is designed for environments where there is

control over the hosts and nodes that connect them and where there is more than one MTU size in use, for example, in data centers and on paths between data centers to allow hosts to better take advantage of a path that is able to support a large PMTU.

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

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

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

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

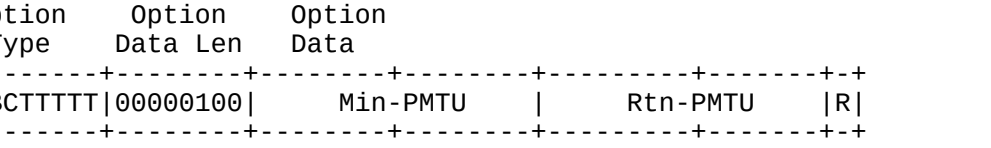


Figure 2: Format of the Minimum Path MTU Hop-by-Hop Option

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

BB 00 Skip over this Option and continue processing.

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

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

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

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

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

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

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

logical ’OR’ with the R-Flag value of the sender. This encoding fits in the minimum-sized Hop-by-Hop Option header.

6. Router, Host, and Transport Layer Behaviors

6.1. Router Behavior

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

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

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

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

6.2. Host Operating System Behavior

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

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

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

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

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

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

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

The normal sequence of operation of the R-Flag using the terminology

from the diagram in Figure 1 is:

1. The source sends a probe to the destination. The sender sets the R-Flag.

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

6.3. Transport Layer Behavior

This Hop-by-Hop Option is intended to be used with a Path MTU Discovery method.

PLPMTUD [RFC8899] uses probe packets for two distinct functions:

* Probe packets are used to confirm connectivity. Such probes can be of any size up to the Packetization Layer Path MTU (PLPMTU). These probe packets are sent to solicit a response using the path to the remote node. These probe packets can carry the Hop-by-Hop PMTU Option, providing the final size of the packet does not exceed the current PLPMTU. After validating that the packet originates from the path (Section 4.6.1 of [RFC8899]), the PLPMTUD method can use the reported size from the Hop-by-Hop Option as the next search point when it resumes the search algorithm. (This use resembles the use of the PTB_SIZE information in Section 4.6.2 of [RFC8899].)

* A second use of probe packets is to explore if a path supports a packet size greater than the current PLPMTU. If this probe packet is successfully delivered (as determined by the source host), then the PLPMTU is raised to the size of the successful probe. These probe packets do not usually set the Path MTU Hop-by-Hop Option. See Section 1.2 of [RFC8899]. Section 4.1 of [RFC8899] also describes ways that a probe packet can be constructed, depending on whether the probe packets carry application data.

The PMTU Hop-by-Hop Option probe can be sent on packets that include application data but needs to be robust to potential loss of the packet (i.e., with the possibility that retransmission might be needed if the packet is lost).

Using a PMTU probe on packets that do not carry application data will avoid the need for loss recovery if a router on the path drops packets that set this Option. (This avoids the transport needing to retransmit a lost packet that includes this Option.) This is the normal default format for both uses of probes.

6.3.1. Including the Option in an Outgoing Packet

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

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

Discussion:

* In the case of TCP, the Option could be included in a packet that carries a TCP segment sent after the connection is established. A

segment without data could be used to avoid the need to retransmit this data if the probe packet is lost. The discovered value can be used to inform PLPMTUD [RFC4821].

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

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

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

* The MinPMTU HBH Option can be used with ICMPv6 [RFC4443]. This requires a response from the remote node and therefore is restricted to use with ICMPv6 echo messages. The MinPMTU HBH Option could provide additional information about the PMTU that might be supported by a path. This could be used as a diagnostic tool to measure the PMTU of a path. As with other uses, the actual supported PMTU is only confirmed after receiving a response to a subsequent probe of the PMTU size.

* A datagram transport can utilize DPLPMTUD [RFC8899]. For example, QUIC (see Section 14.3 of [RFC9000]) can use DPLPMTUD to determine whether the path to a destination will support a desired maximum datagram size. When using the IPv6 MinPMTU HBH Option, the Option could be added to an additional QUIC PMTU probe that is of minimal size (or one no larger than the currently supported PMTU size). Once the return Path MTU value in the MinPMTU HBH Option has been learned, DPLPMTUD can be triggered to test for a larger PLPMTU using an appropriately sized PLPMTU probe packet (see Section 5.3.1 of [RFC8899]).

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

* This method also can be used with anycast to discover the PMTU of the path, but the use needs to be aware that the anycast binding might change.

6.3.2. Validation of the Packet that Includes the Option

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

to protect from off-path data injection; see Section 5.1 of [RFC8085]. IPsec [RFC4301] and TLS [RFC8446] provide greater assurance.

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

6.3.3. Receiving the Option

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

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

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

Discussion:

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

6.3.4. Using the Rtn-PMTU Field

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

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

* The actual PMTU may be lower than the Rtn-PMTU value because there is a Layer 2 device with a lower MTU.

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

* The path might have changed between the time when the probe was sent and when the Rtn-PMTU value received.

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

To avoid unintentional dropping of packets that exceed the actual PMTU (e.g., Scenario 3 in Section 1.1), the source host can delay increasing the PMTU until a probe packet with the size of the Rtn-PMTU value has been successfully acknowledged by the upper layer, confirming that the path supports the larger PMTU. This probing

increases robustness but adds one additional path round-trip time before the PMTU is updated. This use resembles that of PTB messages in Section 4.6 of DPLPMTUD [RFC8899] (with the important difference being that a PTB message can only seek to lower the PMTU, whereas this Option could trigger a probe packet to seek to increase the PMTU).

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

6.3.5. Detecting Path Changes

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

6.3.6. Detection of Dropping Packets that Include the Option

There is evidence that some middleboxes drop packets that include Hop-by-Hop Options. For example, a firewall might drop a packet that carries an unknown extension header or Option. This practice is expected to decrease as an Option becomes more widely used. It could result in the generation of an ICMPv6 message that indicates 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 the dropping of any packet using the Option. This dropping can be avoided by appropriate configuration in a controlled environment, such as within a data center, but it needs to be considered for Internet usage. Section 6.2 recommends that this Option is not used on packets where loss might adversely impact performance.

7. IANA Considerations

IANA has registered an IPv6 Hop-by-Hop Option type in the "Destination Options and Hop-by-Hop Options" registry within the "Internet Protocol Version 6 (IPv6) Parameters" registry group [IANA-HBH]. This assignment is shown in Section 5.

8. Security Considerations

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

8.1. Router Option Processing

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

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

8.2. Network-Layer Host Processing

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

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

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

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

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

8.3. Validating Use of the Option Data

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

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

8.4. Direct Use of the Rtn-PMTU Value

The simplest way to utilize the Rtn-PMTU value is to directly use this to update the PMTU. This approach results in a set of security

issues when the Option carries malicious data:

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

* The method can be used to solicit a response from the destination host. A malicious attacker could forge a packet that causes the destination to add the Option to a packet sent to the source host. A forged value of Rtn-PMTU in the Option Data might also impact the remote endpoint, as described in the previous bullet. This persists until a valid MinPMTU HBH Option is received. This attack could be mitigated by limiting the sending of the MinPMTU HBH Option in reply to incoming packets that carry the Option.

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

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

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

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

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

8.6. Impact of Middleboxes

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

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

9. Experiment Goals

This section describes the experimental goals of this specification.

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

* Support in the sending node (see Section 6.2). This also requires corresponding support in upper-layer protocols (see Section 6.3).

* Router support in nodes (see Section 6.1). The IETF continues to provide recommendations on the use of IPv6 Hop-by-Hop Options, for example, see Section 2.2.2 of [RFC9099]. This document does not update the way router implementations configure support for Hop-by-Hop Options.

* Support in the receiving node (see Section 6.3.3).

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

* reports of implementation experience,

* measurements of the number paths where the method can be used, or

* measurements showing the benefit realized or the implications of using specific methods over specific paths.

10. Implementation Status

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

* Wireshark dissector. This is shipping in production in Wireshark version 3.2 [WIRESHARK].

* A prototype in the open source version of the FD.io Vector Packet Processing (VPP) technology [VPP]. At the time this document was published, the source code can be found [VPP_SRC].

11. References

11.1. Normative References

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

[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>.

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

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <https://www.rfc-editor.org/info/rfc4821>.

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

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

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

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

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

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

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

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based

Multiplexed and Secure Transport", RFC 9000,
DOI 10.17487/RFC9000, May 2021,
<https://www.rfc-editor.org/info/rfc9000>.

[RFC9099] Vyncke, ., Chittimaneni, K., Kaeo, M., and E. Rey,
"Operational Security Considerations for IPv6 Networks",
RFC 9099, DOI 10.17487/RFC9099, August 2021,
<https://www.rfc-editor.org/info/rfc9099>.

[TAPS-ARCH]
Pauly, T., Ed., Trammell, B., Ed., Brunstrom, A.,
Fairhurst, G., and C. Perkins, "An Architecture for
Transport Services", Work in Progress, Internet-Draft,
draft-ietf-taps-arch-12, June 2022,
<https://datatracker.ietf.org/doc/bibxml3/draft-ietf-taps-
arch.xml>.

[VPP] FD.io, "VPP/What is VPP?",
<https://wiki.fd.io/view/VPP/What_is_VPP%3F>.

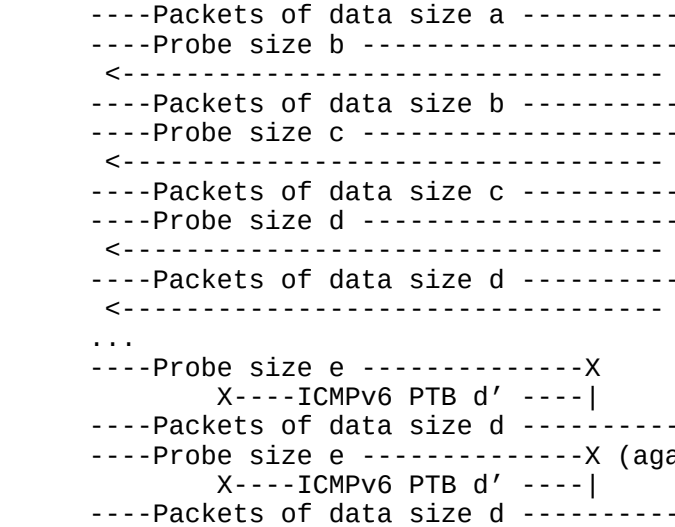
[VPP_SRC] "vpp", commit 21948, ip: HBH MTU recording for IPv6,
<https://gerrit.fd.io/r/c/vpp/+/21948>.

[WIRESHARK]
"Wireshark Network Protocol Analyzer",
<https://www.wireshark.org>.

Appendix A. Examples of Usage

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

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



```
  ...
  etc. until MaxProbes are unsuccessful and search phase completes.
  ----Packets of data size d ----------------------------->
```

Figure 3

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

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

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

Figure 4

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

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

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

Figure 5

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

Acknowledgments

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

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