
Stream:	Internet Engineering Task Force (IETF)	
RFC:	9869	
Category:	Standards Track	
Published:	September 2025	
ISSN:	2070-1721	
Authors:	G. Fairhurst <i>University of Aberdeen</i>	T. Jones <i>University of Aberdeen</i>

RFC 9869

Datagram Packetization Layer Path MTU Discovery (DPLPMTUD) for UDP Options

Abstract

This document specifies how a UDP Options sender implements Datagram Packetization Layer Path MTU Discovery (DPLPMTUD) as a robust method for Path MTU Discovery (PMTUD). This method uses the UDP Options packetization layer. It allows an application to discover the largest size of datagram that can be sent across a network path. It also provides a way to allow the application to periodically verify the current Maximum Packet Size (MPS) supported by a path and to update this when required.

Status of This Memo

This is an Internet Standards Track document.

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). Further information on Internet Standards is available in 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/rfc9869>.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions

with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Terminology	4
3. DPLPMTUD for UDP Options	4
3.1. Packet Formats	6
3.2. Sending Probe Packets with the Request Option	6
3.3. Receiving UDP Options Probe Packets and Sending the RES Option	7
4. DPLPMTUD Sender Procedures for UDP Options	7
4.1. Confirmation of Connectivity Across a Path	7
4.2. Sending Probe Packets to Increase the PLPMTU	8
4.3. Validating the Path with UDP Options	8
4.4. Probe Packets That Do Not Include Application Data	9
4.5. Probe Packets That Include Application Data	9
5. Receiving Events from the Network	9
5.1. Changes in the Path	10
5.2. Validation of PTB Messages	10
6. Examples with Different Receiver Behaviors	10
7. IANA Considerations	11
8. Security Considerations	11
9. References	12
9.1. Normative References	12
9.2. Informative References	12
Acknowledgements	13
Authors' Addresses	13

1. Introduction

The User Datagram Protocol (UDP) [RFC0768] offers a minimal transport service on top of IP and is frequently used as a substrate for other protocols. Section 3.2 of UDP Guidelines [RFC8085] recommends that applications implement some form of Path MTU Discovery (PMTUD) to avoid the generation of IP fragments:

Consequently, an application **SHOULD** either use the path MTU information provided by the IP layer or implement Path MTU Discovery (PMTUD) itself [RFC1191] [RFC1981] [RFC4821] to determine whether the path to a destination will support its desired message size without fragmentation.

The UDP API [RFC8304] offers calls for applications to receive ICMP Packet Too Big (PTB) messages and to control the maximum size of datagrams that are sent, but it does not offer any automated mechanisms for an application to discover the MPS supported by a path. Upper Layer protocols, which include applications, can implement mechanisms for PMTUD above the UDP API.

Packetization Layer Path MTU Discovery (PLPMTUD) [RFC4821] describes a method for a bidirectional Packetization Layer (PL) to search for the largest Packetization Layer PMTU (PLPMTU) supported on a path. DPLPMTUD [RFC8899] specifies this support for datagram transports. PLPMTUD and DPLPMTUD gain robustness by using a probing mechanism that does not solely rely on ICMP PTB messages and works on paths that drop ICMP PTB messages.

UDP Options [RFC9868] supplies functionality that can be used to implement DPLPMTUD within the transport service or in an Upper Layer protocol (including an application) that uses UDP Options. This document specifies how DPLPMTUD using UDP Options is implemented (Section 6.1 of [RFC8899]) and requires support to be enabled at both the sender and receiver.

Implementing DPLPMTUD within the transport service above UDP Options avoids the need for each Upper Layer protocol to implement the DPLPMTUD method. It provides a standard method for applications to discover the current MPS for a path and to detect when this changes. It can be used with Equal-Cost Multipath (ECMP) routing and/or multihoming. If multipath or multihoming is supported, a state machine is needed for each path.

DPLPMTUD is not specified for multicast. The method requires explicit acknowledgement of probe packets provided by UDP Options, which is primarily intended for unicast use (see Section 23 of [RFC9868]).

2. Terminology

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.

This document uses the terms defined for DPLPMTUD (Sections 2 and 5 of [RFC8899]).

3. DPLPMTUD for UDP Options

A UDP Options sender implementing DPLPMTUD uses the method specified in [RFC8899]. In this specification, DPLPMTUD is realized using a pair of UDP Options: the Request (REQ) Option and the Response (RES) Option (Section 11.7 of [RFC9868]). The method also uses the End of Options List (EOL) Option (Section 11.1 of [RFC9868]) to introduce padding to set the size of a probe packet.

Use of DPLPMTUD **MUST** be explicitly enabled by the application, for instance, once an application has established connectivity and is ready to exchange data with the remote Upper Layer protocol. Similarly, a DPLPMTUD receiver **MUST NOT** respond to a UDP REQ Option until DPLPMTUD has been enabled. This is to help protect from misuse of the mechanism for other forms of probing.

Probe packets consume network capacity and incur endpoint processing (Section 4.1 of [RFC8899]). Implementations ought to send a probe packet with a UDP REQ Option only when required by their local DPLPMTUD state machine, i.e., when confirming the base PMTU for the path, probing to increase the PLPMTU, or confirming the current PLPMTU.

DPLPMTUD can be implemented over UDP Options in two ways:

- Implementation within the UDP transport service.
- Implementation in an Upper Layer protocol (or application) that uses UDP Options.

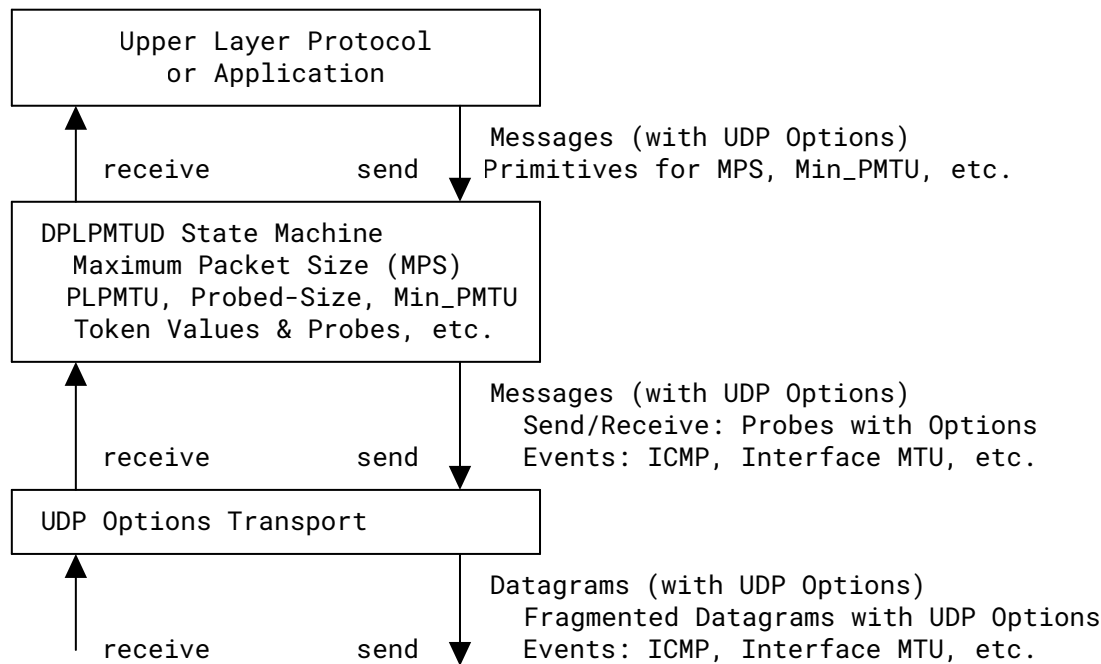
When DPLPMTUD is implemented within the UDP transport service, the DPLPMTUD state machine is responsible for sending probe packets to determine a PLPMTU, as described in this document. This determines an MPS, the largest size of application data block that can be sent across a network path using a single datagram. The Upper Layer protocol is responsible for deciding when a session enables DPLPMTUD.

The discovered PLPMTU can be used to either:

- set the maximum datagram size for the current path or

- set the maximum fragment size when a sender uses the UDP Fragmentation Option to divide a datagram into multiple UDP fragments for transmission. The size of each UDP fragment is then less than or equal to the size of the discovered largest IP packet that can be received across the current path.

The figure below shows an implementation of DPLPMTUD within the UDP transport service. It illustrates key interactions between the layers. This design requires an API primitive to allow the application to control whether the DPLPMTUD state machine is enabled for a specific UDP port. By default, this API **MUST** disable DPLPMTUD processing.



Note: UDP allows an Upper Layer protocol to send datagrams with or without payload data (with or without UDP Options). These are delivered across the network to the remote Upper Layer. When DPLPMTUD is implemented within the UDP transport service, probe packets that include a REQ or RES UDP Option can be sent with no UDP payload. In this case, these probe packets were not generated by a sending application; therefore, the corresponding received datagrams are not delivered to the remote application.

When DPLPMTUD is instead implemented by an Upper Layer protocol, the format and content of probe packets are determined by the Upper Layer protocol. This design is also permitted to use the REQ and RES Options provided by UDP Options.

If DPLPMTUD is active at more than one layer, then the values of the tokens used in REQ Options need to be coordinated with any values used for other DPLPMTUD probe packets to ensure that each probe packet can be identified by a unique token. When configurable, a configuration ought to avoid performing such discovery both within UDP Options and also by an Upper Layer

protocol that sends and receives probe packets via UDP Options. [Section 6.1](#) of [\[RFC8899\]](#) recommends that: "An application **SHOULD** avoid using DPLPMTUD when the underlying transport system provides this capability."

3.1. Packet Formats

The UDP Options used in this document are described in [\[RFC9868\]](#) and are used in the following ways:

- The REQ Option is set by a sending PL to solicit a response from a remote receiver. A four-byte (four-octet) token identifies each request.
- A sending PL can use the EOL Option together with a minimum datagram length to pad probe packets.
- The RES Option is sent by a UDP Options receiver in response to a previously received REQ Option. Each RES Option echoes the last received four-byte token.
- If a UDP Options endpoint creates and sends a datagram with a RES Option solely as response to a received REQ Option, the responder **MUST** limit the rate of these responses (e.g., limiting each pair of ports to send 1 per measured RTT or 1 per second). This rate limit is to mitigate the DoS vector without significantly impacting the operation of DPLPMTUD. An example in [Section 6](#) describes a case where this might be used.
- Reception of a RES Option by the REQ sender confirms that a specific probe packet has been received by the remote UDP Options receiver.

The token allows a UDP Options sender to distinguish between acknowledgements for initial probe packets and acknowledgements confirming receipt of subsequent probe packets (e.g., travelling along alternate paths with a larger RTT). Each probe packet **MUST** be uniquely identifiable by the UDP Options sender within the Maximum Segment Lifetime (MSL) [\[RFC8085\]](#). The UDP Options sender **MUST NOT** reuse a token value within the MSL. A four-byte value for the token field provides sufficient space for multiple unique probe packets to be made within the MSL. Since UDP Options operates over UDP, the token values only need to be unique for the specific 5-tuple over which it is operating.

The value of the four-byte token field **SHOULD** be initialized to a randomized value to enhance protection from off-path attacks, as described in [Section 5.1](#) of [\[RFC8085\]](#).

3.2. Sending Probe Packets with the Request Option

DPLPMTUD relies upon sending a probe packet with a specific size. Each probe packet includes the UDP Options area containing a REQ Option and any other required options concluded with an EOL Option ([Section 11.1](#) of [\[RFC9868\]](#)), followed by any padding needed to inflate to the required probe size.

A probe packet can therefore be up to the maximum size supported by the local interface (i.e., the Interface MTU). Item 2 in [Section 3](#) of [\[RFC8899\]](#) requires the network interface below DPLPMTUD to provide a way to transmit a probe packet that is larger than the current PLPMTU.

The size of this probe packet **MUST NOT** be constrained by the maximum PMTU set by network layer mechanisms (such as discovered by PMTUD [RFC1191][RFC8201] or the PMTU size held in the IP-layer cache), as noted in item 2 in [Section 3](#) of [RFC8899]).

UDP datagrams used as DPLPMTUD probe packets, as described in this document, **MUST NOT** be fragmented at the UDP or IP layer. Therefore, [Section 3](#) of [RFC8899] requires: "In IPv4, a probe packet **MUST** be sent with the Don't Fragment (DF) bit set in the IP header and without network layer endpoint fragmentation."

3.3. Receiving UDP Options Probe Packets and Sending the RES Option

When DPLPMTUD is enabled, a UDP Options receiver responds by sending a UDP datagram with the RES Option when it receives a UDP Options datagram with the REQ Option.

The operation of DPLPMTUD can depend on the support at the remote UDP Options endpoint, the way in which DPLPMTUD is implemented, and in some cases, the application data that is exchanged over the UDP transport service. When UDP Options is not supported by the remote receiver, DPLPMTUD will be unable to confirm the path or to discover the PLPMTU. This will result in the minimum configured PLPMTU (MIN_PLPMTU). More explanation of usage is provided in [Section 6](#).

Note: A receiver that only responds when there is a datagram queued for transmission by the Upper Layer could potentially receive multiple datagrams with a REQ Option before it can respond. When sent, the RES Option will only acknowledge the latest received token value. A sender would then conclude that any earlier REQ Options were not successfully received. However, DPLPMTUD does not usually result in sending more than one probe packet per timeout interval, and a delay in responding will already have been treated as a failed probe attempt. Therefore, this does not significantly impact performance, although a more prompt response would have resulted in DPLPMTUD recording reception of all probe packets.

4. DPLPMTUD Sender Procedures for UDP Options

DPLPMTUD utilizes three types of probe. These are described in the following sections:

- Probes to confirm the path can support the BASE_PLPMTU ([Section 5.1.4](#) of [RFC8899]).
- Probes to detect whether the path can support a larger PLPMTU.
- Probes to validate that the path supports the current PLPMTU.

4.1. Confirmation of Connectivity Across a Path

The DPLPMTUD method requires a PL to confirm connectivity over the path ([Section 5.1.4](#) of [RFC8899]), but UDP itself does not offer a mechanism for this.

UDP Options can provide this required functionality. A UDP Options sender implementing this specification **MUST** elicit a positive confirmation of connectivity for the path by sending a probe packet padded to size `BASE_PLPMTU`. This confirmation probe **MUST** include the REQ UDP Option to elicit a response from the remote DPLPMTUD. Reception of a datagram with the corresponding RES Option confirms the reception of a packet of the probed size that has successfully traversed the path to the receiver. This also confirms that the remote endpoint supports the RES Option.

4.2. Sending Probe Packets to Increase the PLPMTU

From time to time, DPLPMTUD enters the SEARCHING state, described in [Section 5.2 of \[RFC8899\]](#), (e.g., after expiry of the `PMTU_RAISE_TIMER`) to detect whether the current path can support a larger PLPMTU. When the remote endpoint advertises a UDP Maximum Datagram Size (MDS) Option (see [Section 11.5 of \[RFC9868\]](#)), this value **MAY** be used as a hint to initialize this search to increase the PLPMTU.

Probe packets seeking to increase the PLPMTU **SHOULD NOT** carry application data (see "Probing using padding data" in [Section 4.1 of \[RFC8899\]](#)), since they will be lost whenever their size exceeds the actual PMTU. [\[RFC8899\]](#) requires a probe packet to elicit a positive acknowledgement that the path has delivered a datagram of the specific probed size; therefore, a probe packet **MUST** include the REQ Option when using transport options for UDP [\[RFC9868\]](#).

At the receiver, a received probe packet that does not carry application data does not form a part of the end-to-end transport data and is not delivered to the Upper Layer protocol (i.e., application or protocol layered above UDP). A zero-length payload notification could still be delivered to the application (see [Section 5 of \[RFC8085\]](#)), although [Section 18 of \[RFC9868\]](#) discusses the implications when using UDP Options.

4.3. Validating the Path with UDP Options

A PL using DPLPMTUD **MUST** validate that a path continues to support the PLPMTU discovered in a previous search for a suitable PLPMTU value, as defined in [Section 6.1.4 of \[RFC8899\]](#). This validation sends probe packets in the DPLPMTUD SEARCH_COMPLETE state ([Section 5.2 of \[RFC8899\]](#)) to detect black-holing of data ([Section 4.3 of \[RFC8899\]](#) defines a DPLPMTUD black hole).

Path validation can be implemented within UDP Options by generating a probe packet of size PLPMTU, which **MUST** include a REQ Option to elicit a positive confirmation that the path has delivered this probe packet. A probe packet used to validate the path **MAY** use either "Probing using padding data" to construct a probe packet that does not carry any application data or "Probing using application data and padding data"; see [Section 4.1 of \[RFC8899\]](#). When using "Probing using padding data", the UDP Options API does not indicate receipt of the zero-length probe packet (see [Section 6 of \[RFC9868\]](#)).

4.4. Probe Packets That Do Not Include Application Data

A simple implementation of the method might be designed to only use probe packets in a UDP datagram that includes no application data. The size of each probe packet is padded to the required probe size including the REQ Option. This implements "Probing using padding data" ([Section 4.1](#) of [RFC8899]) and avoids having to retransmit application data when a probe fails. This could be achieved by setting a minimum datagram length, such that the options list ends in EOL ([Section 11.1](#) of [RFC9868]) with any additional space zero-filled as needed (see [Section 15](#) of [RFC9868]). In this use, the probe packets do not form a part of the end-to-end transport data and a receiver does not deliver them to the Upper Layer protocol.

4.5. Probe Packets That Include Application Data

An implementation always uses the format in [Section 4.4](#) when DPLPMTUD searches to increase the PLPMTU.

An alternative format is permitted for a probe packet that is used to confirm the connectivity or to validate the path. These probe packets **MAY** carry application data. (UDP payload data is permitted because these probe packets perform black-hole detection and will therefore usually have a higher probability of successful transmission, similar to other packets sent by the Upper Layer protocol.) [Section 4.1](#) of [RFC8899] provides a discussion of the merits and demerits of including application data. For example, this reduces the need to send additional datagrams.

This type of probe packet **MAY** use a control message format defined by the Upper Layer protocol, provided that the message does not need to be delivered reliably. The REQ Option **MUST** be included when the sending Upper Layer protocol performs DPLPMTUD. The DPLPMTUD method tracks the transmission of probe packets (using the REQ Option token).

A receiver that responds to DPLPMTUD **MUST** process the REQ Option and include the corresponding RES Option with an Upper Layer protocol message that it returns to the requester (examples of receiver processing are provided in [Section 6](#)).

Probe packets that use this format form a part of the end-to-end transport data and can be used to manage the PLPMTU in just one direction or can be used for both directions.

5. Receiving Events from the Network

This specification does not rely upon reception of events from the network, but an implementation can utilize these events when they are provided.

5.1. Changes in the Path

A change in the path or the loss of a probe packet can result in DPLPMTUD updating the PLPMTU. DPLPMTUD [RFC8899] recommends that methods are robust to path changes that could have occurred since the path characteristics were last confirmed and to the possibility of inconsistent path information being received. For example, a notification that a path has changed could trigger path validation to provide black-hole protection (Section 4.3 of [RFC8899]).

An Upper Layer protocol could trigger DPLPMTUD to validate the path when it observes a high packet loss rate (or a repeated protocol timeout) [RFC8899].

Section 3 of [RFC8899] requires any methods designed to share the PLPMTU between PLs (such as updating the IP cache PMTU for an interface/destination) to be robust to the wide variety of underlying network forwarding behaviors. For example, an implementation could avoid sharing PMTU information that could potentially relate to packets sent with the same address over a different interface.

5.2. Validation of PTB Messages

Support for receiving ICMP PTB messages is **OPTIONAL** for DPLPMTUD. A UDP Options sender can therefore ignore received ICMP PTB messages.

Before processing an ICMP PTB message, the DPLPMTUD method needs to perform two checks to ensure that the message was received in response to a sent probe packet:

- DPLPMTUD first utilizes the quoted information in each PTB message. The receiver **MUST** validate the protocol information in the quoted packet carried in an ICMP PTB message payload to validate the message originated from the sending node (see Section 4.6.1 of [RFC8899]).
- The receiver **SHOULD** utilize information that is not simple for an off-path attacker to determine (see Section 4.6.1 of [RFC8899]). Specifically, a UDP Options receiver **SHOULD** confirm that the token contained in the UDP REQ Option of the quoted packet has a value that corresponds to a probe packet that was recently sent by the current endpoint.

An implementation unable to support this validation **MUST** ignore received ICMP PTB messages.

6. Examples with Different Receiver Behaviors

When enabled, a DPLPMTUD endpoint that implements UDP Options normally responds with a UDP datagram with a RES Option when requested by a sender.

The following examples describe various possible receiver behaviors:

- No DPLPMTUD receiver support: One case is when a sender supports this specification, but no RES Option is received from the remote endpoint. In this example, the method is unable to discover the PLPMTU. This will result in using the MIN_PLPMTU. Such a remote endpoint

might be not configured to process UDP Options or might not return a datagram with a RES Option for some other reason (e.g., packet loss, insufficient space to include the option, filtering on the path, etc.).

- DPLPMTUD receiver uses application datagrams: In a second case, both the sender and receiver support DPLPMTUD using the specification, and the receiver only returns a RES Option with the next UDP datagram that is sent to the requester. Therefore, reception of a REQ Option does not systematically trigger a response. This allows DPLPMTUD to operate when there is a flow of datagrams in both directions, provided there is periodic feedback (e.g., one acknowledgement packet per RTT). It requires the PLPMTU at the receiver to be sufficiently large enough that the RES Option can be included in the feedback packets that are sent in the return direction. This method avoids opportunities to misuse the method as a DoS attack. However, when there is a low rate of transmission (or no datagrams are sent) in the return direction, this will prevent prompt delivery of the RES Option. At the DPLPMTUD sender, this results in probe packets failing to be acknowledged in time and could result in a smaller PLPMTU than is actually supported by the path or in using the MIN_PLPMTU.
- Unidirectional transfer: Another case is where an application only transfers data in one direction (or predominantly in one direction). In this case, the wait at the receiver for a datagram to be queued before returning a RES Option could easily result in a probe timeout at the DPLPMTUD sender. In this case, DPLPMTUD could allow exchanging datagrams without a payload (as discussed in earlier sections) to return the RES Option.
- DPLPMTUD receiver permitted to send responses in UDP datagrams with no payload: A DPLPMTUD receiver can generate a datagram (e.g., with zero payload data) solely to return a RES Option (e.g., sent when no other datagrams are queued for transmission). This would allow an endpoint to probe the set of UDP ports that have been configured with support for this specification using a DPLPMTUD probe packet. Although this results in some additional traffic overhead, it has the advantage that it can ensure timely progress of DPLPMTUD. [Section 3.1](#) specifies: "If a UDP Options endpoint creates and sends a datagram with a RES Option solely as response to a received REQ Option, the responder **MUST** limit the rate of these responses (e.g., limiting each pair of ports to send 1 per measured RTT or 1 per second)". This rate limit is to mitigate the DoS vector, without significantly impacting the operation of DPLPMTUD.

7. IANA Considerations

This document has no IANA actions.

8. Security Considerations

The security considerations for using UDP Options are described in [\[RFC9868\]](#). The method does not change the integrity protection offered by the UDP Options method.

The security considerations for using DPLPMTUD are described in [\[RFC8899\]](#). On-path attackers could maliciously drop or modify probe packets to seek to decrease the PMTU or to maliciously modify probe packets in an attempt to black-hole traffic.

The specification recommends that the token value in the REQ Option is initialized to a randomized value. This is to enhance protection from off-path attacks. If a subsequent probe packet uses a token value that is easily derived from the initial value (e.g., incrementing the value), a misbehaving on-path observer could then determine the token values used for subsequent probe packets from that sender, even if these probe packets are not transiting via the observer. This would allow probe packets to be forged, with an impact similar to other on-path attacks against probe packets. This attack could be mitigated by using an unpredictable token value for each probe packet.

The method does not change the ICMP PTB message validation method described by DPLPMTUD: A UDP Options sender that utilizes ICMP PTB messages received to a probe packet **MUST** use the quoted packet to validate the UDP port information in combination with the token contained in the UDP Option before processing the packet using the DPLPMTUD method.

Upper Layer protocols or applications that employ encryption ought to perform DPLPMTUD at a layer above UDP Options and not enable UDP Options support for DPLPMTUD. This allows the application to control when DPLPMTUD is used to control the additional traffic that this generates. This also ensures that DPLPMTUD probe packets are encrypted.

9. References

9.1. Normative References

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <<https://www.rfc-editor.org/info/rfc768>>.
- [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>>.
- [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>>.
- [RFC9868] Touch, J. and C. Heard, Ed., "Transport Options for UDP", RFC 9868, DOI 10.17487/RFC9868, September 2025, <<https://www.rfc-editor.org/info/rfc9868>>.

9.2. Informative References

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

- [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>>.
- [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>>.
- [RFC8304] Fairhurst, G. and T. Jones, "Transport Features of the User Datagram Protocol (UDP) and Lightweight UDP (UDP-Lite)", RFC 8304, DOI 10.17487/RFC8304, February 2018, <<https://www.rfc-editor.org/info/rfc8304>>.

Acknowledgements

Gorry Fairhurst and Tom Jones are supported by funding provided by the University of Aberdeen. The authors would like to thank Magnus Westerlund and Mohamed Boucadair for their detailed comments and also other people who contributed to completing this document.

Authors' Addresses

Godred Fairhurst

University of Aberdeen
School of Engineering
Fraser Noble Building
Aberdeen
AB24 3UE
United Kingdom
Email: gorry@erg.abdn.ac.uk

Tom Jones

University of Aberdeen
School of Engineering
Fraser Noble Building
Aberdeen
AB24 3UE
United Kingdom
Email: tom@erg.abdn.ac.uk