Workgroup: Internet Engineering Task Force
Internet-Draft:
draft-ietf-tsvwg-udp-options-dplpmtud-02
Published: 25 November 2021
Intended Status: Standards Track
Expires: 29 May 2022
Authors: G. Fairhurst T. Jones
University of Aberdeen University of Aberdeen

Datagram PLPMTUD for UDP Options

Abstract

This document specifies how a UDP Options sender implements Datagram Packetization Layer Path Maximum Transmission Unit Discovery (DPLPMTUD) as a robust method for Path Maximum Transmission Unit discovery. This method uses the UDP Options packetization layer. It allows a datagram application to discover the largest size of datagram that can be sent across a specific network path.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The User Datagram Protocol [RFC0768] offers a minimal transport service on top of IP and is frequently used as a substrate for other protocols. Section 3.5 of UDP Guidelines [RFC8085] recommends that applications implement some form of Path MTU discovery 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)".

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 does not offer any automated mechanisms for an application to discover the maximum packet size supported by a path. Upper layer protocols (which can include applications) implement mechanisms for Path MTU discovery above the UDP API.

Packetization Layer Path MTU Discovery (PLPMTUD) [RFC4821] describes a method for a Packetization Layer (PL) (such as UDP Options) to search for the largest Packetization Layer PMTU (PLPMTU) supported on a path. Datagram PLPMTUD (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.

This document specifies how UDP Options [I-D.ietf-tsvwg-udp-options] can be used as a PL to implement DPLPMTUD (see Section 6.1 of [RFC8899]). In summary, UDP Options [I-D.ietf-tsvwg-udp-options] supplies functionality that can be used to implement DPLPMTUD within the UDP transport service. Implementing DPLPMTUD using UDP Options avoids the need for each upper layer protocol or application to implement the DPLPMTUD method. This provides a standard method for applications to discover the current maximum packet size for a path and to detect when this changes.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD 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.

3. DPLPMTUD for UDP Options

There are two ways an upper PL can perform DPLPMTUD:

* The UDP Options sender implementing DPLPMTUD uses the method specified in [RFC8899] and the upper PL (or application) does not perform PMTU discovery. In this case, UDP Options processing is responsible for sending probes to determine a PLPMTU, as described in this document. "An application SHOULD avoid using DPLPMTUD when the underlying transport system provides this capability" (Section 6.1 of [RFC8899]). This discovered PLPMTU can be used by UDP Options to either:

  - set the maximum datagram size for the current path (based on the discovered largest IP packet that can be received across the current path).

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

* An upper PL (or application performs DPLPMTUD (e.g., QUIC [RFC9000]). This upper PL then uses probes to determine a safe PLPMTU for the datagrams that it sends. The format and content of any probe is determined by the upper PL. Such a design should avoid performing discovery at multiple levels, so, when when

configurable, this upper PL SHOULD disable DPLPMTUD by UDP Options.

This section describes packet formats and procedures for DPLPMTUD using UDP Options.

4. Sending UDP-Options Probe Packets

DPLPMTUD relies upon the ability of a UDP Options sender to generate a probe with a specific size, up to the maximum for the size supported by a local interface. This MUST NOT be constrained by the maximum PMTU set by network layer mechanisms (such as PMTUD [RFC1191][RFC8201] or the PMTU size held in the IP- layer cache), as noted in bullet 2 of Section 2 in [RFC8899]).

Probe packets consume network capacity and incur endpoint processing (see Section 4.1 of [RFC8899]). Implementations ought to send a probe with an 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 to confirm the current PLPMTU.

4.1. Packet Probes using the Echo Request and Response Options

A UDP Options node that supports DPLPMTUD MUST support sending and receiving of the REQ Option and the RES Option. When not supported, DPLPMTUD will be unable to confirm the Path or to discover the PMTU.

This section describes a format of probe consisting of an empty UDP datagram, UDP Options area and Padding.

A Probe Packet includes the UDP Options area containing a RES Option and any other required options concluded with an EOL Option followed by any padding needed to inflate to the required probe size.

The UDP Options used in this document are described in Section 6 of [I-D.ietf-tsvwg-udp-options]:

* The REQ Option is set by a sending PL to solicit a response from a remote UDP Options receiver. A four-byte token identifies each request.

* The RES Option is generated by the UDP Options receiver in response to reception of a previously received REQ Option. Each RES Option echoes a previously received four-byte token.

* Reception of a RES Option confirms reception of a specific received probe by the remote UDP Options receiver.

The token value allows a sender to distinguish between acknowledgements for initial probes and acknowledgements confirming receipt of subsequent probes (e.g., travelling along alternate paths with a larger round trip time). This needs each probe to be uniquely identifiable by the UDP Options sender within the Maximum Segment Lifetime (MSL). The UDP Options sender therefore MUST NOT recycle token values until they have expired or have been acknowledged. A four byte value for the token field provides sufficient space for multiple unique probes 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 DPLPMTUD is operating.

The initial value of the four byte token field SHOULD be assigned to a randomised value to enhance protection from off-path attacks, as described in Section 5.1 of [RFC8085]).

4.2. DPLPMTUD Procedures for UDP Options

DPLPMTUD utilises three types of probes. These are described in the following sections:

* A probe to confirm the path can support the BASE_PLPMTU see Section 5.1.4 of [RFC8899]).

* A probe to detect whether the path can support a larger PLPMTU.

* A probe to validate the path supports the current PLPMTU.

4.2.1. Confirmation of Connectivity across a Path

The DPLPMTUD method requires a PL to confirm connectivity over the path using the BASE_PLPMTU (see Section 5.1.4 of [RFC8899]), but UDP 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, padded to size BASE_PLPMTU. This confirmation probe MUST include a UDP Option that elicits a response from the remote endpoint (e.g., by including the RES and REQ Options) to confirm that a packet of the size traversed the path. This also confirms that the remote receiver supports use of the RES and REQ Options.

4.2.2. Sending Probe Packets to Increase the PLPMTU

From time to time, DPLPMTUD enters the SEARCHING state [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 Segment Size (MSS) option, this value can be used as a hint to initialise 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.

A probe seeking to increase the PLPMTU needs to elicit a positive acknowledgment that the path has delivered a datagram of the specific probed size and, therefore, MUST include the REQ Option.

Received probes that do not carry application data do not form a part of the end-to-end transport data and are not delivered to the upper layer protocol.

4.2.3. Validating the Path with UDP Options

A PL using DPLPMTUD needs to validate that a path continues to support the PLPMTU discovered in a previous search for a suitable PLPMTU value (see Section 6.1.4 of [RFC8899]). This validation sends probes in the DPLPMTUD SEARCH_COMPLETE state to detect black-holing of data (see Section 4.2 of [RFC8899]).

This function can be implemented within UDP Options, by generating a probe of size PLPMTU, which MUST include a RES Option to elicit a positive confirmation whether the path has delivered the probe. This confirmation probe MAY use "Probing using padding data" or "Probing using application data and padding data" (see Section 4.1 of [RFC8899]) or can construct a probe packet that does not carry any application data, as described in a previous section.

4.2.4. Sending Packet Probes that include Application Data

The method can be designed to only use probes that are formed of a UDP datagram that includes application data (which could be applications control information), padded to the required size and include a RES Option. This implements "Probing using padding data", and avoids having to retransmit application data when a probe fails. This type of probe must be used when searching to increase the PLPMTU. In this use, the RES and REQ Options do not form a part of the end-to-end transport data and a receiver does not deliver them to the upper layer protocol. A simple implementation of the method might be designed to only use this format for all probes.

The probe used to confirm the connectivity or to validate support for the current PLPMTU are also permitted to carry application data, since this type of probe is expected to be successful. 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 an additional datagram when confirming that the current path supports datagrams of size PLPMTU and could be designed to utilise a control message format defined by the PL that does not need to be delivered reliably. In this use, the RES and REQ Options need to be included by the sending upper layer and the values of tokens need to be coordinated with values used for other DPLPMTUD probe packets. These probes do form a part of the end-to-end transport data and a receiver does deliver the RES and REQ Options to the upper layer protocol.

4.2.5. Changes in the Path

A change in the path or the loss of probe packets can result in a change of 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 could have changed could trigger path validation to provide black hole protection Section 4.3 of [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.

4.3. PTB Message Handling for this Method

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

A UDP Options sender that utilises ICMP PTB messages received in response to a probe packet MUST use the quoted packet to validate the UDP port information in combination with the token value contained in the UDP Option, before processing the packet using the DPLPMTUD method. Section 4.6.1 of [RFC8899] specifies this validation procedure. An implementation unable to support this validation needs to ignore received ICMP PTB messages.

5. Acknowledgements

Gorry Fairhurst and Tom Jones are supported by funding provided by the University of Aberdeen.

6. IANA Considerations

This memo includes no requests to IANA.

7. Security Considerations

The security considerations for using UDP Options are described in [I-D.ietf-tsvwg-udp-options]. The proposed new method does not change the integrity protection offered by the UDP options method.

The specification recommends that the token in the REQ Option is initialised to a randomised value to enhance protection from off-path attacks.

The security considerations for using DPLPMTUD are described in [RFC8899]. The proposed new method does not change the ICMP PTB message validation method described DPLPMTUD: A UDP Options sender that utilies ICMP PTB messages received to a probe packet MUST use the quoted packet to validate the UDP port information in combination with the token and/or timestamp value contained in the UDP Option, before processing the packet using the DPLPMTUD method.

8. References

8.1. Normative References

[I-D.ietf-tsvwg-udp-options]
Touch, J. D., "Transport Options for UDP", Work in Progress, Internet-Draft, draft-ietf-tsvwg-udp-options-13, 19 June 2021, <https://www.ietf.org/archive/id/draft-ietf-tsvwg-udp-options-13.txt>.

[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>.

8.2. 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>.

[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>.

Appendix A. Revision Notes

XXX Note to RFC-Editor: please remove this entire section prior to publication. XXX

Individual draft-00.

* This version contains a description for consideration and comment by the TSVWG.

Individual draft-01.

* Address Nits

* Change Probe Request and Probe Reponse options to Echo to align names with draft-ietf-tsvwg-udp-options

* Remove Appendix B, Informative Description of new UDP Options

* Add additional sections around Probe Packet generation

Individual draft-02.

*Address Nits

Individual draft-03.

*Referenced DPLPMTUD RFC.

*Tidied language to clarify the method.

Individual draft-04

*Reworded text on probing with data a little

*Removed paragraph on suspending ICMP PTB suspension.

Working group draft-00

*-00 First Working Group Version

*RFC8899 call search_done SEARCH_COMPLETE, fix

Working group draft -01

*Update to reflect new fragmentation design in UDP Options.

*Add a description of uses of DPLPMTUD with UDP Options.

*Add a description on how to form probe packets with padding.

*Say that MSS options can be used to initialise the search algorithm.

*Say that the recommended approach is to not use user data for probes.

*Attempts to clarify and improve wording throughout.

*Remove text saying you can respond to multiple probes in a single packet.

*Simplified text by removing options that don't yield benefit.

Working group draft -02

*Update to reflect comments from MED.

*More consistent description of DPLPMTUD with UDP Options.

*Clarify token is intended per 5-tuple, not interface.

*BASE_PLPMTU related to RFC8899.

*Probes with user data can carry application control data.

*Added that application data uses RES and REQ tokens from the app.

*QUIC was intended as an informational reference to an example of RFC8899.

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