Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 22 May 2022

G. Fairhurst
T. Jones
University of Aberdeen

18 November 2021

Datagram PLPMTUD for UDP Options
draft-ietf-tsvwg-udp-options-dplpmtud-01

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 network path.

Table of Contents

1. Introduction 2
2. Terminology 3
3. DPLPMTUD for UDP Options 3
4. Sending UDP-Options Probe Packets 4
4.1. Packet Probes using the Echo Request Option Request Option 4
4.2. DPLPMTUD Procedures for UDP Options 5
4.2.1. Confirmation of Connectivity across a Path 5
4.2.2. Sending Probe Packets to Increase the PLPMTU 5
4.2.3. Validating the Path with UDP Options 6
4.2.4. Sending Packet Probes that include Application Data 6
4.3. PTB Message Handling for this Method 7
5. Acknowledgements 7
6. IANA Considerations 7
7. Security Considerations 7
8. References 7
8.1. Normative References 7
8.2. Informative References 8
Appendix A. Revision Notes 9
Authors' Addresses 10

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. ~~Applications and upper~~ Upper layer protocols (including applications) implement mechanisms for Path MTU discovery above the UDP API.

Packetization Layer Path MTU Discovery ~~PMTUD~~ (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 PL.

Commenté [BMI1]: What about adding a clarification about the positioning of this spec vs. 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.~~ This document specifies how an implementation can use ~~this additional~~UDP options ~~functionality~~ to support DPLPMTUD. 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. 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 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~~ a given 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 content~~s~~ of any probe is determined by the upper PL. Such a design needs to avoid performing discovery at multiple levels, so, ~~when~~ when configurable, this upper PL SHOULD disable DPLPMTUD by UDP Options [RFC8899]).

This section ~~describe~~ 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 ~~the~~ a local interface. The size of a DPLPMTUD probe packet MUST NOT be constrained by the maximum PMTU set by network layer mechanisms (such as PMTUD [RFC1063][RFC8201] or the IP Cache).

Probe packets consume network capacity and incur endpoint processing (~~see~~ Section 4.1 of [RFC8899]). Implementations ought to send a probe with a Request Probe 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 ~~Option~~and ~~-~~Request Options

This section describes a format of probes consisting of an empty UDP datagram, UDP Options area, and Padding. The UDP Options area contains the Echo Request Option (RES), any other required options concluded with an EOL Option followed by any padding needed to inflate to the required probe size. The reception of this option generates an Echo Response Option that confirms reception of a specific received probe.

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

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

* The Echo Response Option (REQ) is generated by the UDP Options receiver in response to reception of ~~a previously received~~an Echo Request Option. Each Echo Response Option echoes a previously received four-byte token.

Commenté [BMI2]: Not sure why this one is cited here.

Commenté [BMI3]: As many interfaces can be supported

Commenté [BMI4]: I don't remember there is a similar requirement in RFC8899.

Commenté [BMI5]: Obsoleted by RFC 1191

Commenté [BMI6]: That is?

Commenté [BMI7]: Do you mean RES/REQ options?

Please add a pointer to Joe' I-D.

Commenté [BMI8]: Some policies may be configured at the remote side to ignore RES.

I would add "absent any local policy".

Mis en forme : Surlignage

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~~ 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~~ four-byte value for the token field provides sufficient space for multiple unique probes to be made within the MSL.

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

4.2. DPLPMTUD Procedures for UDP Options

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

* A probe to confirm the path can support the base PLPMTU.

* 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 ~~ECHO~~ Echo Request/Response Option~~s~~) to confirm that a packet of the size traversed the path.

4.2.2. Sending Probe Packets to Increase the PLPMTU

From time to time, DPLPMTUD searches 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.

Commenté [BMI9]: Should this be restricted per interface?

Commenté [BMI10]: Clarify where this size is defined.

Commenté [BMI11]: Can indication be provided here?

Mis en forme : Surlignage

Commenté [BMI12]: Still the path is locally identified by the interface used to froward the probe. No?

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 MUST elicit a positive confirmation that the path has delivered a ~~Datagram~~ datagram of the specific probed size and, therefore, SHOULD include the Echo Request Option ~~Request 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, i.e., 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~~ must include a UDP Option to elicit a positive confirmation that 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~~Section 3.

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 Options~~ datagram with UDP Options containing control information, padded to the required size. This implements "Probing using padding data", and avoids having to retransmit application data when a probe fails. This type of probes must be used when searching to increase the PLPMTU. These probes do not form a part of the end-to-end transport data and a receiver must ~~does~~ not deliver these to the upper layer protocol. A simple implementation of the method might be designed to only use this format for all probes.

The ~~Probe~~ probe used to confirm the connectivity or to validate support for the current PLPMTU ~~are also permitted to~~may 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

Commenté [BMI13]: Not sure what is meant here.

Commenté [BMI14]: Redundant with « within UDP Options ..”

Commenté [BMI15]: ...but still this is not for granted.

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.

Commenté [BMI16]: That is?

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.

Commenté [BMI17]: Does this assume that a similar validation to 4.6.1 of RFC8899 is supported?

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 and/or timestamp value contained in the UDP Option, before processing the packet using the DPLPMTUD method (~~see~~ Section 4.4.1 of [RFC8899]). 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/RES message is initialised to a randomised value to enhance protection from off-path attacks.

The security considerations for using DPLPMTUD are described in Section 8 of [RFC8899]. The proposed new method does not change the ICMP PTB message validation method described DPLPMTUD: A UDP Options sender that utilises 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>.

[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>.

Commenté [BMI18]: This is not normative.

8.2. 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>.

[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>.

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.

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