Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: May 01, 2018

G. Fairhurst
T. Jones
University of Aberdeen
M. Tuexen
I. Ruengeler
Muenster University of Applied Sciences
October 30, 2017

Packetization Layer Path MTU Discovery for Datagram Transports
draft-fairhurst-tsvwg-datagram-plpmtud-01.txt

Abstract

This document describes a robust method for Path MTU Discovery (PMTUD) for datagram packetization layers. It allows these layers to probe an Internet path with progressively larger packets to determine a maximum packet size This method is described as an extension to RFC 1191 and RFC 8201, which specify ICMP-based Path MTU Discovery for IP versions 4 and 6. The document provides functionally for datagram transports that is equivalent to the packetization layer PMTUD specification for TCP, specified in RFC4821.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 01, 2018.

Copyright Notice

Table of Contents

1. Introduction 2
2. Terminology 4
3. Features required to provide PLPMTUD at the Transport Layer . 6
3.1. PMTU Probe Packets 8
3.2. Validation of the current effective PMTU 9
3.3. Reduction of the effective PMTU 9
4. Datagram PLPMTUD 9
4.1. Probing 10
4.2. Timers 10
4.3. Constants 11
4.4. Variables 11
4.5. State Machine 11
5. Specification of Protocol-Specific Methods 13
5.1. UDP and UDP-Lite 13
5.1.1. UDP Options 14
5.1.2. UDP Options required for PLPMTUD 14
5.1.2.1. Echo Request Option 14
5.1.2.2. Echo Response Option 14
5.1.3. Sending UDP-Option Probe Packets 14
5.1.4. Validating the Path with UDP Options 15
5.1.5. Handling of PTB Messages by UDP 15
5.2. SCTP 15
5.2.1. SCTP/IP4 and SCTP/IPv6 15
5.2.1.1. Sending SCTP Probe Packets 15
5.2.1.2. Validating the Path with SCTP 16
5.2.1.3. PTB Message Handling by SCTP 16
5.2.2. SCTP/UDP 16
5.2.2.1. Sending SCTP/UDP Probe Packets 16
5.2.2.2. Validating the Path with SCTP/UDP 16
5.2.2.3. Handling of PTB Messages by SCTP/UDP 16
5.2.3. SCTP/DTLS 16
5.2.3.1. Sending SCTP/DTLS Probe Packets 17
5.2.3.2. Validating the Path with SCTP/DTLS 17
5.2.3.3. Handling of PTB Messages by SCTP/DTLS 17
5.3. Other IETF Transports 17
6. Acknowledgements 17
7. IANA Considerations 17
8. Security Considerations 17
9. References 17
9.1. Normative References 17
9.2. Informative References 19
Appendix A. Event-driven state changes 19
Appendix B. Revision Notes 22
Authors' Addresses 23

1. Introduction

The IETF has specified datagram transport using UDP, SCTP, SCTP/UDP, DCCP, and DCCP/UDP, as well as protocols layered on top of these transports.

Classical Path Maximum Transmission Unit Discovery (PMTUD) can be used with any transport that is able to process ICMP Packet Too Big (PTB) messages (e.g., [RFC1191] and [RFC8201]). It adjusts the effective Path MTU (PMTU), based on reception of ICMP Path too Big (PTB) messages to decrease the PMTU when a packet is sent with a size larger than the value supported along a path, and a method that from time-to-time increases the packet size in attempt to discover an increase in the supported PMTU.

However, Classical PMTUD is subject to protocol failures. One failure arises when traffic using a packet size larger than the actual supported PMTU is blackholed (silently discarded). This may happen when ICMP PTB messages are not delivered back to the sender for some reason [RFC2923]). For example, ICMP messages are increasingly filtered by middleboxes (including Firewalls) [RFC4890], and may not be correctly processed by tunnel endpoints.

Another failure could result if a system not on the path sends a PTB that attempts to force the sender to change the effective PMTU [RFC8201]. A sender could protect itself by using the quoted packet within the PTB message payload to verify that the received PTB message was generated in response to a packet that had actually been sent. However, there are situations where a sender is unable to provide this verification (e.g., when the PTB message does not include sufficient information, often the case for IPv4; or where the information corresponds to an encrypted packet). At the network layer there also could be insufficient context to perform this verification, which depends on information about the active transport flows (e.g., the socket/address pairs being used, and other protocol header information). This verification is more straight forward at a the Packetization Layer (PL) or a higher layer.

The term Packetization Layer has been introduced to describe the layer that is responsible for placing data blocks into the payload of packets and selecting an appropriate maximum packet size. This function is often performed by a transport protocol, but can also be performed by other encapsulation methods working below the application.

In contrast to PMTUD, Packetization Layer Path MTU Discovery (PLPMTUD) [RFC4821] does not rely upon reception and verification of PTB messages. It is therefore more robust than Classical PMTUD. This has become the recommended approach for implementing PMTU discovery with TCP. It uses a general strategy where the PL searches for an appropriate PMTU by sending probe packets along the network path with a progressively larger packet size. If a probe packet is successfully delivered (as determined by the PL), then the effective Path MTU is raised to the probe size.

PLPMTUD introduces flexibility in the implementation of PMTU discovery. At one extreme, it can be configured to only perform PTB black hole recovery to increase the robustness of Classical PMTUD, or at the other extreme, all PTB processing can be disabled and PLPMTUD can completely replace Classical PMTUD. PLPMTUD can also include additional consistency checks without increasing the risk of blackholing.

The UDP-Guidelines [RFC8085] state "an application SHOULD either use the path MTU information provided by the IP layer or implement Path MTU Discovery (PMTUD)", but does not provide a mechanism for discovering the largest size of unfragmented datagram than can be used on a path. PLPMTUD has not currently been specified for UDP, while Section 10.2 of [RFC4821] recommends a PLPMTUD probing method for SCTP that utilises heartbeat messages as packet probes, but does not provide a complete specification. This document provides the details to complete that specification. Similarly, the method defined in this specification could be used with the Datagram Congestion Control Protocol (DCCP) [RFC4340] requires implementations to support Classical PMTUD and states that a DCCP sender "MUST maintain the maximum packet size (MPS) allowed for each active DCCP session". It also defines the current congestion control maximum packet size (CCMPS) supported by a path. This recommends use of PMTUD, and suggests use of control packets (DCCP-Sync) as path probe packets, because they do not risk application data loss. The document also contains information that enables the implementation of PLPMTUD with other datagram transports

Section Section 4 of this document presents a set of algorithms for datagram protocols to discover a maximum size for the effective PMTU across a path. The methods described rely on features of the PL Section 3 and apply to transport protocols over IPv4 and IPv6. It does not require cooperation from the lower layers (except that they are consistent about which packet sizes are acceptable). It can utilise PTB messages when these are available.

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 [RFC2119].

Other terminology is directly copied from [RFC4821], and the definitions in [RFC1122].

Black-Holed: When the sender is unaware that packets are not delivered to the destination endpoint (e.g., when the sender is unaware of a change in the path to one with a smaller PMTU).

Classical Path MTU Discovery: Classical PMTUD is a process described in [RFC1191] and [RFC8201], in which nodes rely on PTB messages to

learn the largest size of unfragmented datagram than can be used across a path.

Datagram: A datagram is a transport-layer protocol data unit, transmitted in the payload of an IP packet.

Effective PMTU: The current estimated value for PMTU used by a Packetization Layer.

EMTU_S: The Effective MTU for sending (EMTU_S) is defined in [RFC1122] as "the maximum IP datagram size that may be sent, for a particular combination of IP source and destination addresses...".

EMTU_R: The Effective MTU for receiving (EMTU_R) is designated in [RFC1122] as "the largest datagram size that can be reassembled by EMTU_R ("Effective MTU to receive")".

Link: A communication facility or medium over which nodes can communicate at the link layer, i.e., a layer below the IP layer. Examples are Ethernet LANs and Internet (or higher) layer and tunnels.

Link MTU: The Maximum Transmission Unit (MTU) is the size in bytes of the largest IP packet, including the IP header and payload, that can be transmitted over a link. Note that this could more properly be called the IP MTU, to be consistent with how other standards organizations use the acronym MTU. This includes the IP header, but excludes link layer headers and other framing that is not part of IP or the IP payload. Other standards organizations generally define link MTU to include the link layer headers.

MPS: The Maximum Packet Size (MPS), the largest size of application data block that may be sent unfragmented across a path. In PLPMTUD this quantity is derived from Effective PMTU by taking into consideration the size of the application and lower protocol layer headers, and may be limited by the application protocol.

Packet: An IP header plus the IP payload.

Packetization Layer (PL): The layer of the network stack that places data into packets and performs transport protocol functions.

Path: The set of link and routers traversed by a packet between a source node and a destination node.

Path MTU (PMTU): The minimum of the link MTU of all the links forming a path between a source node and a destination node.

PLPMTUD: Packetization Layer Path MTU Discovery, the method described in this document for datagram PLs, which is an extension to Classical PMTU Discovery.

3. Features required to provide PLPMTUD at the Transport Layer

TCP PLPMTUD has been defined using standard TCP protocol mechanisms. All of the requirements in [RFC4821] also apply to use of the technique with a datagram PL. Unlike TCP, some datagram PLs require additional mechanisms to implement PLPMTUD.

There are ten requirements for performing the datagram PLPMTUD method described in this specification:

1. PMTU parameters: A PLPMTUD sender is REQUIRED to provide information about the maximum size of packet that can be transmitted by the sender on the local link (the Link MTU and MAY utilize similar information about the receiver when this is supplied (note this may be less than EMTU_R). Some applications also have a maximum transport protocol data unit (PDU) size, in which case there may be no benefit from probing for a size larger than this (unless a transport allows multiplexing multiple applications PDUs into the same datagram.)

2. Effective PMTU: A datagram application MUST be able to choose the size of datagrams sent to the network, up to the effective PMTU, or a smaller value (such as the MPS) derived from this. This value is managed by the PMTUD method. The effective PMTU (specified in Section 1 of [RFC1191]) is equivalent to the EMTU_S (specified in [RFC1122]).

3. Probe packets: On request, a PLPMTUD sender is REQUIRED to be able to transmit a packet larger than the current effective PMTU (but always with a total size less than the link MTU), which the method can use as a probe packet. In IPv4, a probe packet is always sent with the Don’t Fragment (DF) bit set and without network layer endpoint fragmentation.

4. Processing PTB messages: A PLPMTUD sender MAY optionally utilize PTB messages received from the network layer to help identify when a path does not support the current size of packet probe. Any received PTB message SHOULD/MUST be verified before it is used to update the PMTU discovery information [RFC8201]. This verification confirms that the PTB message was sent in response to a packet originating by the sender, and needs to be performed before the PMTU discovery method reacts to the PTB message. When the router link MTU is indicated in the PTB message this MAY be

used by datagram PLPMTUD to reduce the size of a probe, but MUST NOT be used increase the effective PMTU.

5. Reception feedback: The destination PL endpoint is REQUIRED to provide a feedback method that indicates when a probe packet has been received by the destination endpoint. The local PL endpoint is REQUIRED to pass this feedback to the sender PLPMTUD method.

6. Probing and congestion control: The isolated loss of a probe packet SHOULD NOT be treated as an indication of congestion and its loss not directly trigger a congestion control reaction.

7. Probe loss recovery: If the data block carried by a probe message needs to be sent reliably, the PL (or layers above) MUST arrange retransmission/repair of any resulting loss. This method MUST be robust in the case where paket probes are lost due to other reasons (including link transmission error, congestion). The PLPMTUD method treats isolated loss of a probe packet (with or without an PTB message) as a potential indication of a PMTU limit on the path. The PL is permitted to retransmit any data included in a lost probe packet without adjusting its congestion window.

8. Cached effective PMTU: The sender MUST cache the effective PMTU value between probes and needs also to consider the disruption that could be incurred by an unsuccessful probe - both upon the flow that incurs a probe loss, and other flows that experience the effect of additional probe traffic.

9. Shared effective PMTU state: The specification of PLPMTUD [RFC4821] states: "If PLPMTUD updates the MTU for a particular path, all Packetization Layer sessions that share the path representation (as described in Section 5.2 of [RFC4821]) SHOULD be notified to make use of the new MTU and make the required congestion control adjustments". Such methods need to robust to the wide variety of underlying network forwarding behaviours. Considerations about caching have been noted [RFC8201].

In addition the following design principles are stated:

- Suitable MPS: The PLPMTUD method SHOULD avoid forcing an application to use an arbitrary small MPS (effective PMTU) for transmission while the method is searching for the currently supported PMTU. Datagram PLs do not necessarily support fragmentation of PDUs larger than the PMTU. A reduced MPS can adversely impact the performance of a datagram application.

- Path validation: The PLPMTUD method MUST be robust to path changes that could have occurred since the path characteristics were last confirmed.

- Datagram reordering: A method MUST be robust to the possibility that a flow encounters reordering, or has the traffic (inlcuding probe packets) is divided over more than one network path.

- When to probe: The PLPMTUD method SHOULD determine whether the path capacity has increased since it last measured the path. This determines when the path should again be probed.

3.1. PMTU Probe Packets

PMTU discovery relies upon the sender being able to generate probe messages with a specific size. TCP is able to generate probe packets by choosing to appropriately segment data being sent [RFC4821].

In contrast, datagram PLs either have to request an application to send a data block with a specified size, or to utilise padding functions to extend the datagram beyond the size of the application data block. Protocols that permit exchange of control messages (without an application data block) could alternatively prefer to generate a probe packet by extending a control message with padding data.

When the method fails to validate the PMTU for the path, the required size of probe packet can need to be less than the size of the data block generated by an application. In this case, the PL could provide a wat to fragment a datagram at the PL, or could instead utilise a control packet with padding.

A receiver needs to be able to be able to distinguish in-band data from any added padding, and ensure that any added padding is not passed to an application at the receiver.

This results in three ways that a sender can create a probe packet:

Probing using appication data: A probe packet that contains a data block supplied by an application that matches the size required for the probe. This requires a method to request the application to issue a data block of the desired probe size. If the application/transport needs protection from the loss of this probe packet, the application/transport may perform transport-layer retransmission/repair of the data block (e.g., by retransmission after loss is detected or by duplicating the data block in a datagram without the padding data).

Probing using appication data: A probe packet that contains a data block supplied by an application that is combined with padding to inflate the length of the datagram to the size required for the probe. If the application/transport needs protection from the loss of this probe packet, the application/transport may perform transport-layer retransmission/repair of the data block (e.g., by retransmission after loss is detected or by duplicating the data

block in a datagram without the padding data).

Probing using appication data: A probe packet that contains only control information and padding to inflate the packet to the size required for the probe. Since these probe packets do not carry any application-supplied data block,they do not typically require retransmission, although they do still consume network capacity.

3.2. Validation of the current effective PMTU

The PL needs a method to determine when packet probes have been successfully received end-to-end across a network path.

Transport protocols can include end-to-end methods that detect and report reception of specific datagrams that they send (e.g., DCCP and SCTP provide keep-alive/heartbeat features). This can also be used by PLPMTUD to acknowledge reception of a probe packet.

A PL that does not acknowledge data reception (e.g., UDP and UDP-Lite) is unable to detect when the packets it sends are discarded because their size is greater than the actual PMTUD. These PLs need to either reply on application protocol to detect this, or use of an additional transport method such as UDP-Options [I-D.ietf-tsvwg-udp-options], and then need to send a reachability probe (e.g., periodically solicit a response) to determine if the current effective PMTU is still supported by the network path.

PMTU discovery can also utilise PTB messages to detect when the actual PMTU supported by a network path is less than the current size of datagrams that are being sent.

3.3. Reduction of the effective PMTU

When the current effective PMTU is no longer supported by the network path, the transport needs to detect this and reduce the effective PMTU.

- A PL that sends a datagram larger than the actual PMTU that includes no application data block, or one that does not attempt to provide any retransmission, can send a new probe packet with an updated probe size.

- A PL that wishes to resend the application data block, may need to re-fragment the data block to a smaller datagram size. This could utilise network-layer or PL fragmentation when these are available.

4. Datagram PLPMTUD

This section specifies Datagram PLPMTUD.

The central idea of PLPMTU discovery is probing by a sender. Probe packets of increasing size are sent to find out the maximum size of a user message that is completely transferred across the network path from the sender to the destination. If a PTB message is received from a router or middlebox, this information ought to be verified and SHOULD used. The PTB messages can improve performance compared to one that relies solely on probing.

4.1. Probing

The PLPMTUD method utilises a timer to trigger the generation of probe packets. The probe_timer is started each time a probe packet is sent to the destination and is cancelled when receipt of the probe packet is acknowledged. Each time the probe_timer expires, the probe_error_counter is incremented, and the probe packet is retransmitted. The counter is initialised to zero when a probe packet is first sent with a particular size. The maximum number of retransmissions per probing size is configured (MAX_PROBES). If the value of the PROBE_COUNT exceeds MAX_PROBES, probing will be stopped and the last successfully probed PMTU is set as the effective PMTU.

Once probing is completed, the sender continues to use the effective PMTU until either a PTB message is received or the PMTU_RAISE_TIMER expires. If the PL is unable to verify reachability to the destination endpoint after probing has completed, the method uses a REACHABILITY_TIMER to periodically repeat a probe packet for the current effective PMTU size, while the PMTU_RAISE_TIMER is running. If the resulting probe packet is not acknowledged (i.e. the PROBE_TIMER expires), the method re-starts probing for the PMTU.

4.2. Timers

This method utilises three timers:

PROBE_TIMER: Configured to expire after a period longer than the maximum time to receive an acknowledgment to a probe packet.

PMTU_RAISE_TIMER: Configured to the period a sender ought to continue use the current effective PMTU, after which it re-commences probing for a higher PMTU. This timer has a period of 600 secs, as recommended by [RFC4821].

REACHABILITY_TIMER: Configured to the period a sender ought to wait before confirming the current effective PMTU is still supported. This is less than the PMTU_RAISE_TIMER.

An implementation could implement the various timers using a single

timer process.

4.3. Constants

The following constants are defined:

MAX_PROBES: The maximum value of the PROBE_ERROR_COUNTER.

MIN_PMTU: The smallest allowed probe packet size. This value is 1280 bytes as specified in [RFC2460].

BASE_PMTU: The BASE_PMTU is a considered a size that should work in most cases. The size equal to or larger than the minimum permitted and smaller than the maximum allowed. In the case of IPv6, this value is 1280 bytes as specified in [RFC2460]. When using IPv4, a size of 1200 is RECOMMENDED.

MAX_PMTU: This is the largest size of PMTU that is probed. It must be less than or equal to the minimum of the local MTU of the outgoing interface and the destination effective MTU for receiving.

4.4. Variables

This method utilises a set of variables:

effective PMTU: The effective PMTU is the maximum size of datagram that the method has currently determined can be supported along the entire path.

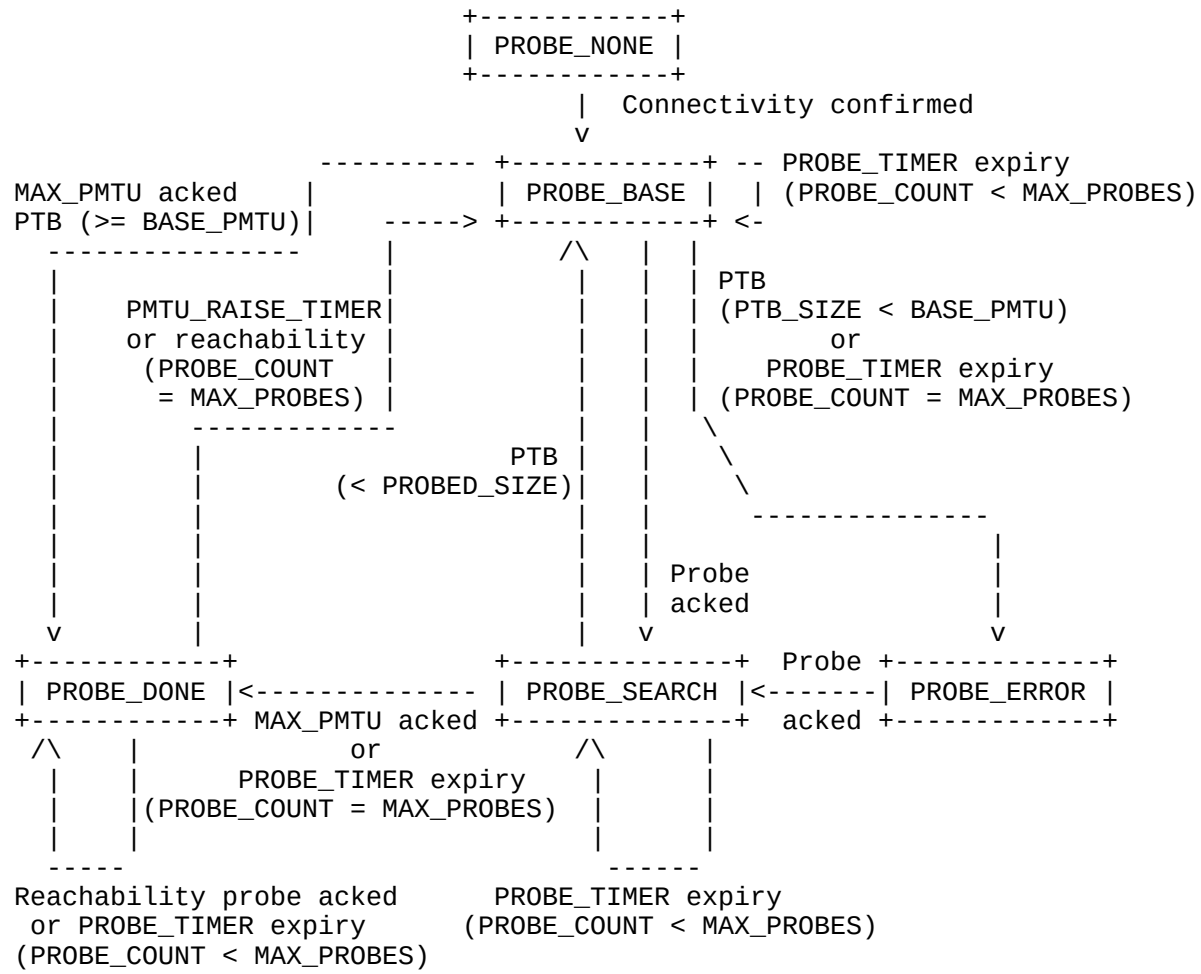
PROBED_SIZE: The PROBED_SIZE is the size of the current probe packet. This is a tentative value for the effective PMTU, which is awaiting confirmation by an acknowledgment.

PROBE_COUNT: This is a count of the number of unsuccessful probe packets that have been sent with size PROBED_SIZE. The value is initialised to zero when a particular size of PROBED_SIZE is first attempted.

PTB_SIZE: The PTB_Sizde is value returned by a verified PTB message indicating the local MTU size of a router along the path.

4.5. State Machine

A state machine for Datagram PLPMTUD is depicted in Figure 1. If multihoming is supported, a state machine is needed for each active path.



The following states are defined to reflect the probing process.

PROBE_NONE: The PROBE_NONE state is the initial state before probing has started. PLPMTUD is not performed in this state. The state transitions to PROBE_BASE, when a path has been confirmed, i.e. when a packet has arrived on this path. The effective PMTU is set to the BASE_PMTU size. Probing ought to start immediately after connection setup to prevent the loss of user data.

PROBE_BASE: The PROBE_BASE state is the starting point for datagram PLPMTUD, and used to confirm whether the BASE_PMTU size is supported by the network path. On entry, the PROBED_SIZE is set

to the BASE_PMTU size and the PROBE_COUNT is set to zero. A probe packet is sent, and the PROBE_TIMER is started. The state is left when the PROBE_COUNT reaches MAX_PROBES; a PTB message is received, or a probe packet is acknowledged.

PROBE_SEARCH: The PROBE_SEARCH state is the main probing state. This state is entered either when probing for the BASE_PMTU was successful or when there is a successful reachability test in the PROBE_ERROR state. On entry, the effective PMTU is set to the last acknowledged PROBED_SIZE.

On the first probe packet for each probed size, the PROBE_COUNT is set to zero. Each time a probe packet is acknowledged, the effective PMTU is set to the PROBED_SIZE, and then the PROBED_SIZE is increased. When a probe packet is not acknowledged within the period of the PROBE_TIMER, the PROBE_COUNT is incremented and the probe packet is retransmitted. The state is exited when the PROBE_COUNT reaches MAX_PROBES; a PTB message is verified; or a probe of size PMTU_MAX is acknowledged.

PROBE_ERROR: The PROBE_ERROR state represents the case where the network path is not known to support an effective PMTU of at least the BASE_PMTU size. It is entered when either a probe of size BASE_PMTU has not been acknowledged or a verified PTB message indicates a smaller link MTU than the BASE_PMTU. On entry, the PROBE_COUNT is set to zero and the PROBED_SIZE is set to the MIN_PMTU size, and the effective PMTU is reset to MIN_PMTU size. In this state, a probe packet is sent, and the PROBE_TIMER is started. The state transitions to the PROBE_SEARCH state when a probe packet is acknowledged.

PROBE_DONE: The PROBE_DONE state indicates a successful end to a probing phase. Datagram PLPMTUD remains in this state until either the PMTU_RAISE_TIMER expires or a PTB message is verified.

When PLPMTUD uses an unacknowledged PL and is in the PROBE_DONE state, a REACHABILITY_TIMER periodically resets the PROBE_COUNT and schedules a probe packet with the size of the effective PMTU. If the probe packet fails to be acknowledged after MAX_PROBES attempts, the method enters the PROBE_BASE state. An acknowledged PL SHOULD NOT continue to probe in this state.

Appendix Appendix A contains an informative description of key events:

5. Specification of Protocol-Specific Methods

This section specifies protocol-specific details for datagram PLPMTUD for IETF-specified transport protocols.

5.1. UDP and UDP-Lite

The current specifications of UDP and UDP-LIte [RFC3828] do not define a method in the RFC-series that supports PLPMTUD. In particular, these transport do not provide the transport layer features needed to implement datagram PLPMTUD.

5.1.1. UDP Options

UDP-Options [I-D.ietf-tsvwg-udp-options] supply the additional functionality required to implement datagram PLPMTUD. This enables padding to be added to UDP datagrams and can be used to provide feedback acknowledgement of received probe packets.

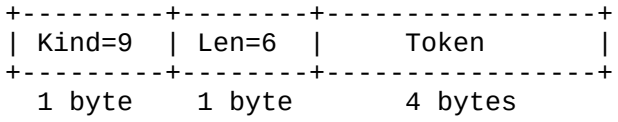
5.1.2. UDP Options required for PLPMTUD

This subsection proposes two new UDP-Options that add support for requesting a datagram response be sent and to mark this datagram as a response to a request.

<< We may define a parameter in an Option to indicate the EMTU_R to the peer.>>

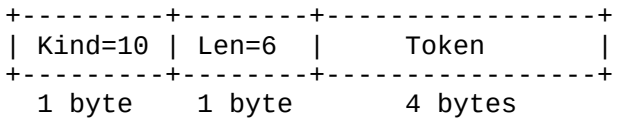
5.1.2.1. Echo Request Option

The Echo Request Option allows a sending endpoint to solicit a response from a destination endpoint. The Echo Request carries a four byte token set by the sender.



5.1.2.2. Echo Response Option

The Echo Response Option is generated by the PL in response to reception of a previously received Echo Request. The Token field is associates the response with the Token value carried in the most recently-received Echo Request. The rate of generation of UDP packets carrying an Echo Response Option MAY be rate-limited.



5.1.3. Sending UDP-Option Probe Packets

This method specifies a probe packet that does not carry an application data block. The probe packet consists of a UDP datagram header followed by a UDP Option containing the ECHOREQ option, which is followed by NOP Options to pad the remainder of the datagram payload. The NOP padding is used to control the length of the probe

packet.

A UDP Option carrying the ECHORES option is used to provide feedback when the probe packet is received at the destination endpoint.

5.1.4. Validating the Path with UDP Options

Since UDP is an unacknowledged PL, a sender that does not have higher-layer information confirming correct delivery of datagrams SHOULD implement the REACHABILITY_TIMER to periodically send probe packets while in the PROBE_DONE state.

5.1.5. Handling of PTB Messages by UDP

Normal ICMP verification MUST be performed as specified in Section 5.2 of [RFC8085]. This requires that the PL verifies each received PTB messages to verify these are received in response to transmitted traffic. A verified PTB message MAY be used as input to the PLPMTUD algorithm.

5.2. SCTP

Section 10.2 of [RFC4821] specifies a recommended PLPMTUD probing method for SCTP. It recommends the use of the PAD chunk, defined in [RFC4820] to be attached to a minimum length HEARTBEAT chunk to build a probe packet. This enables probing without affecting the transfer of user messages and without interfering with congestion control. This is preferred to the use of DATA chunks (with padding as required) to serve as path probes.

<< We might define a parameter contained in the INIT and INIT ACK chunk to indicate the MTU to the peer. However, multihoming makes this a bit complex, so it might not be worth doing.>>

5.2.1. SCTP/IP4 and SCTP/IPv6

The base protocol is specified in [RFC4960].

5.2.1.1. Sending SCTP Probe Packets

Probe packets consist of an SCTP common header followed by a HEARTBEAT chunk and a PAD chunk. The PAD chunk is used to control the length of the probe packet. The HEARTBEAT chunk is used to trigger the sending of a HEARTBEAT ACK chunk. The reception of the HEARTBEAT ACK chunk acknowledges reception of a successful probe.

The HEARTBEAT chunk carries a Heartbeat Information parameter which should include, besides the information suggested in [RFC4960], the probing size, which is the MTU size the complete datagram will add up to. The size of the PAD chunk is therefore computed by reducing the probing size by the IPv4 or IPv6 header size, the SCTP common header, the HEARTBEAT request and the PAD chunk header. The payload of the PAD chunk contains arbitrary data.

To avoid the fragmentation of retransmitted data, probing starts right after the handshake before data is sent. Assuming normal behaviour (i.e., the PMTU is smaller than or equal to the interface MTU), this process will take a few RTTs depending on the number of PMTU sizes probed. The Heartbeat timer can be used to implement the PROBE_TIMER.

5.2.1.2. Validating the Path with SCTP

Since SCTP provides an acknowledged PL, a sender does MUST NOT implement the REACHABILITY_TIMER while in the PROBE_DONE state.

5.2.1.3. PTB Message Handling by SCTP

Normal ICMP verification MUST be performed as specified in Appendix C of [RFC4960]. This requires that the first 8 bytes of the SCTP common header are quoted in the payload of the PTB message , which can be the case for ICMPv4 and is normally the case for ICMPv6. When the verification is completed, the router Link MTU indicated in the PTB message SHOULD be used with the PLPMTUD algorithm.

5.2.2. SCTP/UDP

The UDP encapsulation of SCTP is specified in [RFC6951].

5.2.2.1. Sending SCTP/UDP Probe Packets

Packet probing can be performed as specified in Section 5.2.1.1. The maximum payload is reduced by 8 bytes, which has to be considered when filling the PAD chunk.

5.2.2.2. Validating the Path with SCTP/UDP

Since SCTP provides an acknowledged PL, a sender does MUST NOT implement the REACHABILITY_TIMER while in the PROBE_DONE state.

5.2.2.3. Handling of PTB Messages by SCTP/UDP

Normal ICMP verification MUST be performed for PTB messages as specified in Appendix C of [RFC4960]. This requires that the first 8 bytes of the SCTP common header are contained in the PTB message, which can be the case for ICMPv4 (but note the UDP header also consumes a part of the quoted packet header) and is normally the case for ICMPv6. When the verification is completed, the router Link MTU size indicated in the PTB message SHOULD be used with the PLPMTUD algorithm.

5.2.3. SCTP/DTLS

The DTLS encapsulation of SCTP is specified in [I-D.ietf-tsvwg-sctp-dtls-encaps]. It is used for data channels in WebRTC implementations.

5.2.3.1. Sending SCTP/DTLS Probe Packets

Packet probing can be done as specified in Section 5.2.1.1.

5.2.3.2. Validating the Path with SCTP/DTLS

Since SCTP provides an acknowledged PL, a sender does MUST NOT implement the REACHABILITY_TIMER while in the PROBE_DONE state.

5.2.3.3. Handling of PTB Messages by SCTP/DTLS

It is not possible to perform normal ICMP verification as specified in [RFC4960], since even if the ICMP contains enough information, the reflected SCTP common header would be encrypted. Therefore it is not possible to process PTB messages at the PL.

5.3. Other IETF Transports

QUIC is a UDP-based transport that provides reception feedback [I-D .ietf-quic-transport].

<< This section will be completed in a future revision of this ID >>

6. Acknowledgements

This work was partially funded by the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 644334 (NEAT). The views expressed are solely those of the author(s).

7. IANA Considerations

This memo includes no request to IANA.

If there are no requirements for IANA, the section will be removed during conversion into an RFC by the RFC Editor.

8. Security Considerations

The security considerations for the use of UDP and SCTP are provided in the references RFCs. Security guidance for applications using UDP is provided in the UDP-Guidelines [RFC8085].

PTB messages could potentially be used to cause a node to inappropriately reduce the effective PMTU. A node supporting PLPMTUD SHOULD appropriately verify the payload of PTB messages to ensure these are received in response to transmitted traffic (i.e., a reported error condition that corresponds to a datagram actually sent by the path layer.

9. References

9.1. Normative References

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", Internet-Draft draft-ietf-quic-transport-04, June 2017.

[I-D.ietf-tsvwg-sctp-dtls-encaps]
Tuexen, M., Stewart, R., Jesup, R. and S. Loreto, "DTLS Encapsulation of SCTP Packets", Internet-Draft draft-ietf-tsvwg-sctp-dtls-encaps-09, January 2015.

[I-D.ietf-tsvwg-udp-options]
Touch, J., "Transport Options for UDP", Internet-Draft draft-ietf-tsvwg-udp-options-01, June 2017.

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <http://www.rfc-editor.org/info/rfc1122>.

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

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

[RFC3828] Larzon, L-A., Degermark, M., Pink, S., Jonsson, L-E.Ed., and G. Fairhurst, Ed., "The Lightweight User Datagram Protocol (UDP-Lite)", RFC 3828, DOI 10.17487/RFC3828, July 2004, <http://www.rfc-editor.org/info/rfc3828>.

[RFC4820] Tuexen, M., Stewart, R. and P. Lei, "Padding Chunk and Parameter for the Stream Control Transmission Protocol (SCTP)", RFC 4820, DOI 10.17487/RFC4820, March 2007, <https://www.rfc-editor.org/info/rfc4820>.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <https://www.rfc-editor.org/info/rfc4960>.

[RFC6951] Tuexen, M. and R. Stewart, "UDP Encapsulation of Stream Control Transmission Protocol (SCTP) Packets for End-Host to End-Host Communication", RFC 6951, DOI 10.17487/RFC6951, May 2013, <https://www.rfc-editor.org/info/rfc6951>.

[RFC8085] Eggert, L., Fairhurst, G. and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <http://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>.

9.2. Informative References

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

[RFC2923] Lahey, K., "TCP Problems with Path MTU Discovery", RFC 2923, DOI 10.17487/RFC2923, September 2000, <https://www.rfc-editor.org/info/rfc2923>.

[RFC4340] Kohler, E., Handley, M. and S. Floyd, "Datagram Congestion Control Protocol (DCCP)", RFC 4340, DOI 10.17487/RFC4340, March 2006, <https://www.rfc-editor.org/info/rfc4340>.

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

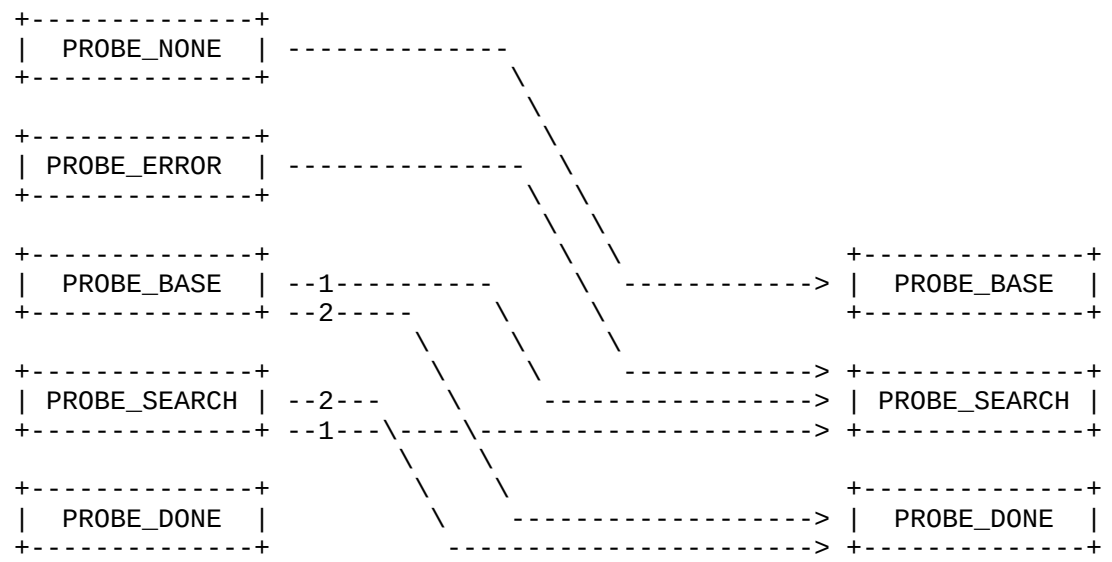
[RFC4890] Davies, E. and J. Mohacsi, "Recommendations for Filtering ICMPv6 Messages in Firewalls", RFC 4890, DOI 10.17487/RFC4890, May 2007, <http://www.rfc-editor.org/info/rfc4890>.

Appendix A. Event-driven state changes

This appendix contains an informative description of key events:

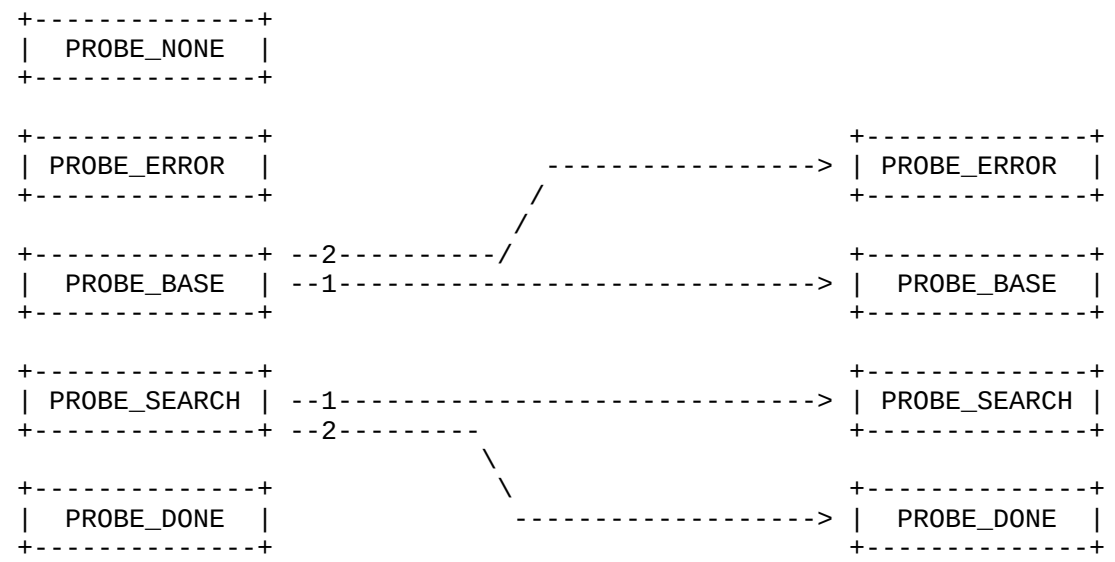
Path Setup: When a new path is initiated, the state is set to PROBE_NONE. As soon as the path is confirmed, the state changes to PROBE_BASE and the probing mechanism for this path is started. A probe packet with the size of the BASE_PMTU is sent.

Arrival of an Acknowledgment: Depending on the probing state, the reaction differs according to Figure 4, which is just a simplification of Figure 1 focusing on this event.



Condition 1: The maximum PMTU size has not yet been reached.
Condition 2: The maximum PMTU size has been reached.

Probing timeout: The PROBE_COUNT is initialised to zero each time the value of PROBED_SIZE is changed. The PROBE_TIMER is started each time a probe packet is sent. It is stopped when an acknowledgment arrives that confirms delivery of a probe packet. If the probe packet is not acknowledged before,the PROBE_TIMER expires, the PROBE_ERROR_COUNTER is incremented. When the PROBE_COUNT equals the value MAX_PROBES, the state is changed, otherwise a new probe packet of the same size (PROBED_SIZE) is resent. The state transitions are illustrated in Figure 5. This shows a simplification of Figure 1 with a focus only on this event.



Condition 1: The maximum number of probe packets has not been reached. Condition 2: The maximum number of probe packets has been reached.

PMTU raise timer timeout: The path through the network can change over time. It impossible to discover whether a path change has increased in the actual PMTU by exchanging packets less than or equal to the effective PMTU. This requires PLPMTUD to periodically send a probe packet to detect whether a larger PMTU is possible. This probe packet is generated by the PMTU_RAISE_TIMER. When the timer expires, probing is restarted with the BASE_PMTU and the state is changed to PROBE_BASE.

Arrival of an ICMP message: The active probing of the path can be supported by the arrival of PTB messages sent by routers or middleboxes with a link MTU that is smaller than the probe packet size. If the PTB message includes the router link MTU, three cases can be distinguished:

1. The indicated link MTU in the PTB message is between the already probed and effective MTU and the probe that triggered the PTB message.

2. The indicated link MTU in the PTB message is smaller than the effective PMTU.

3. The indicated link MTU in the PTB message is equal to the BASE_PMTU.

In first case, the PROBE_BASE state transitions to the PROBE_ERROR state. In the PROBE_SEARCH state, a new probe packet is sent with the sized reported by the PTB message. Its result is handled according to the former events.

The second case could be a result of a network re-configuration. If the reported link MTU in the PTB message is greater than the BASE_MTU, the probing starts again with a value of PROBE_BASE. Otherwise, the method enters the state PROBE_ERROR.

In the third case, the maximum possible PMTU has been reached. This is probed again, because there could be a link further along the path with a still smaller MTU.

Note: Not all routers include the link MTU size when they send a PTB message. If the PTB message does not indicate the link MTU, the probe is handled in the same way as condition 2 of Figure 5.

Appendix B. Revision Notes

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

Individual draft -00:

- Comments and corrections are welcome directly to the authors or via the IETF TSVWG working group mailing list.
- This update is proposed for WG comments.

Individual draft -01:

- Contains the first representation of the algorithm, showing the states and timers
- The text describing when to set the effective PMTU has not yet been verified by the authors
- The text describing how to handle a PTB message indicating a link MTU larger than the probe has yet not been verified by the authors
- No text currently describes how to handle inconsistent results from arbitrary re-routing along different parallel paths
- Some middleboxes lie about the MTU they report in PTB messages.
- Some constants and times do not yet have recommended values
- To determine security to off-path-attacks: We need to decide whether a received PTB message SHOULD be verified or MUST be verified?
- This update is proposed for WG comments.

Authors' Addresses

Godred Fairhurst
University of Aberdeen
School of Engineering
Fraser Noble Building
Aberdeen, AB24 3UE
UK

Email: gorry@erg.abdn.ac.uk

Tom Jones
University of Aberdeen
School of Engineering
Fraser Noble Building
Aberdeen, AB24 3UE
UK

Email: tom@erg.abdn.ac.uk

Michael Tuexen
Muenster University of Applied Sciences
Stegerwaldstrasse 39
Steinfurt, 48565
DE

Email: tuexen@fh-muenster.de

Irene Ruengeler
Muenster University of Applied Sciences
Stegerwaldstrasse 39
Steinfurt, 48565
DE

Email: i.ruengeler@fh-muenster.de