Network Working Group
Internet-Draft
Intended status: Informational
Expires: 8 January 2026

P. S. Kim
TU Korea

7 July 2025

Path MTU Algorithms and Issues in QUIC Applications
draft-pskim-pmtu-algorithm-issue-00

Abstract

This draft consider Path MTU (PMTU) alogorithms and issues in QUIC applications. Fistly, a passive probing approach is adopted to discover the PMTU. The process of discovering the PMTU is not performed separately, but is performed simultaneously in the actual application data communication. A probe packet is defined newly using 1-RTT packet which includes actual application data as well as a short packet header and a PING_EXT frame. Until the optimal PMTU is discovered, the size of the probe packet is changed according to the size of the PMTU candidate. Secondly, a PMTU black hole problem in secure and reliable transport protocol is discussed and a possible solution can be suggested from existing researches. Thirdly, PMTU issues for media delivery over UDP, such as WebRTC and media over QUIC (MoQ) are discussed.

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 https://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 8 January 2026.

Copyright Notice

Table of Contents

1. Introduction

A PMTU Discovery (PMTUD) is a standardized technique in computer networking for determining the PMTU size on the network path between two IP hosts, usually with the goal of avoiding IP fragmentation for IPv4[RFC1191] and for IPv6[RFC8201]. When a packet too large for the path was sent, the PMTUD expects to receive a Packet Too Big (PTB) message. However, there are multiple reasons why a PTB message might not arrive at the sender.

Therefore, the PMTUD for the Packetization Layer (PL) that selects the size of IP packets is specified recently in [RFC8899]. RFC8899 works without a signal from the network and covers generic PL protocols such as QUIC of [RFC9000]. Meanwhile, [UDP-PMTUD] complements RFC8899 by specifying how a UDP Options sender implements Datagram PL PMTUD(DPLPMTUD). It allows a datagram application to discover the largest size of datagram that can be sent across a specific network path. However, [RFC8899] does not contain details about how to discovery for the optimal PMTU.

Recently, therefore, [Q-PMTUD] complements RFC8899 by presenting a discovery algorithm with QUIC. Using the discovery algorithm with a set of possible PMTU candidates and their possible probing sequences, the optimal PMTU is obtained. However, to discover the optimal PMTU, some probe packets which have no semantic value might be injecting into network, which is called active probing or active measurement.

The active probing approach can increase a network load and perturb the network.

Based on [Q-PMTUD] and [UDP-PMTUD], this draft consider an alternative PMTUD for QUIC. To discover the optimal PMTU, the passive probing approach is adopted. The process of discovering the optimal PMTU is not carried out separately, but is carried out simultaneously in the actual application data communication. A probe packet is defined newly using 1-RTT packet which includes actual application data as well as a short packet header and a PING_EXT frame. The PING_EXT frame is also defined newly. Until the optimal PMTU is discovered, the size of the probe packet is changed according to the size of the PMTU candidate. A simple discovery algorithm using only the PMTU candidate sequence with linear upward is described in this draft. Other rather complex discovery algorithms that consider various PMTU candidate sequences will be dealt with in the future.

Meanwhile, in classical PMTU discovery, a PMTU black hole problem could arise when the PTB messages are not sent back to the sender for some reason. As an extreme case such as the secure and reliable transport protocol QUIC, a sender that trusts only cryptographically secured information will not use PTB messages. Thus, a possible solution can be suggested from existing researches.

Moreover, recently, Media over QUIC(MoQ) is being designed to handle real-time communication and media streaming over QUIC [MOQT]. PMTUD in the context of MoQ involves determining the MTU of the network path between a client and server to avoid packet fragmentation. This ensures efficient and reliable media delivery over QUIC, which is a fast and efficient transport protocol. Therefore, PMTU issues for media delivery over UDP, such as WebRTC and MoQ are discussed.

1.1. Requirements Language

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

2. Passive Probing for PMTUD with QUIC

The specification of QUIC in RFC9000 recommends to use the PMTUD framework of RFC8899. RFC 8899 DPLPMTUD has the following phases:

- Base: Send the probe in its basic size first. QUIC assumes that the specified 1280 bytes pass, so it starts from the next phase.
- Search: Search for candidate PMTUs while sending probes. Once the optimal PMTU is detected, proceed to the next phase.

- Search Complete: Since PMTU may change due to route changes, check if it is still optimal.

There are three possible ways to create a PMTU probe packet as follows[RFC8899]:

- Probing using padding data
- Probing using application data and padding data
- Probing using application data

[UDP-PMTUD] describes "Probe Packets that include Application Data" to implement "Probing using application data" of [RFC8899].

However, RFC8899 does not contain details about how to discovery for the optimal PMTU.

2.1. Active Probing for PMTUD with QUIC[Q-PMTUD]

[Q-PMTUD] complements the specification, RFC8899, by presenting a discovery algorithm with QUIC. From a practical point of view, it might be a good choice to consider only a set of common PMTU values. However, the PMTU value may usually change over time. Thus,

[Q-PMTUD] considers a set of possible PMTU candidates. PMTU candidates are values every 4 bytes from 1280 bytes to 1500 bytes. Then, a discovery algorithm is proposed, which probes one PMTU candidate after the other. This means, it starts the probe for the next candidate not before the probe for the current candidate either succeeded or failed. Then endpoint uses this discovery algorithm that repeatedly chooses PMTU candidates to probe.

The candidate sequence is required to specify the order in which the discovery algorithm probes PMTU candidates. The endpoint must choose a PMTU candidate larger than the largest successfully probed candidate and smaller than any other probed candidate with a lost probe packet. Seven candidate sequences are considered, evaluated, and compared in [Q-PMTUD].

To probe one PMTU candidate, according to RFC9000, the endpoint builds a probe packet with a short packet header, a PING frame and PADDING frames. The endpoint controls the size of the probe packet by the number of PADDING frames, whose size is one byte each. The PING frame makes the packet ack-eliciting.

However, to discover the optimal PMTU, some probe packets which have no semantic value might be injecting into network, which is thus called active measurement or active probing. This active probing approach can increase a network load and perturb the network.

2.2. A New PMTU Probe Packet (1-RTT packet format)

(1) Probe packet format for active probing [Q-PMTUD]

IP header + UDP header + Short header(QUIC header) + PING frame + PADDING frames

The size of the probe packet is controlled by the number of PADDING frames.

(2) Probe packet format for passive probing

In this drfat, a probe packet is defined newly using 1-RTT packet including actual application data as well as a PING_EXT frame as follows:

IP header + UDP header + Short header(QUIC Header) + PING_EXT frame + Actual application data

- PING_EXT frame (defined newly)
  . Frame Type Name : PING_EXT
  . Type Value : 0x20
  . The PING_EXT frame makes the packet ack-eliciting. In addition, the PING_EXT frame indicates that the current 1-RTT packet is now discovering the optimal PMTU as well as transmitting actual application data.

- Application data
  . Actual application data controls the size of the probe packet by a multiple of four bytes.

The size of probe packet is changed according to PMTU candidates as follows:

. 1280 + incremental where, for example, incremental can be a multiple of four as shown in [Q-PMTUD].

2.3. Passive Probing

Through the new probe packet, it is possible not only to discovery the optimal PMTU, but also to transmit actual application data. That is, to discover the optimal PMTU size and carry actual application data, the endpoint expand the payload of all UDP datagrams.

(1) A simple algorithm for discovering the optimal PMTU

As specified in RFC9000, QUIC must send QUIC packets with the smallest allowed maximum datagram size when validating a path during connection initiation or migration. Thus, the endpoint sets the probe packet initially to the smallest allowed maximum datagram size of

1280 bytes including actual application data as well as a short packet header, a PING_EXT frame.

As mentioned, until the optimal PMTU is discovered, the size of the probe packet is changed successively according to the size of the PMTU candidate. The size of the probe packet is controlled with the size of actual application data. The size of actual application data is a multiple of four.

In the active probing approach [Q-PMTUD], the endpoint uses a simple discovery algorithm that repeatedly chooses PMTU candidates to probe. Thus, seven PMTU candidate sequences are considered and each candidate sequence specifies the order in which the discovery algorithm probes PMTU candidates. In addition, four metrics such as number of probed PMTU candidates, time to discover the optimal PMTU, network load, average PMTU estimation are defined for performance evaluations of seven sequences.

However, because the process of discovering the optimal PMTU is carried out simultaneously in the actual application data communication, only the PMTU candidate sequence with linear upward is adopted first in this draft. The linear upward sequence selects one candidate after the other from a list of candidates in ascending order, starting with the second one (the first one was probed with the smallest allowed maximum datagram size of 1280 bytes). Other rather complex discovery approaches that consider various PMTU candidate sequence will be dealt with in the future.

Until the optimal PMTU is discovered, the endpoint repeats a series of probing steps. In absence of a PTB message, the discovery algorithm considers a probe for a PMTU candidate as failed, only if the probe packet of the size of the candidate were detected as lost. A probe for a PMTU candidate that fails, lets all other probes for larger candidates fail as well. Therefore, the optimal PMTU is the PMTU candidate that succeeded just before the failure.

(2) Discovery completion and PMTU cache

When the algorithm determines that it has discovered the optimal PMTU, the endpoint terminates the probing. Then, the endpoint sets the 1-RTT packet finally to the optimal datagram size using the optimal PMTU discovered. From now on, the 1-RTT packet does not include a PING_EXT frame. QUIC can cache the optimal PMTU discovered and use it for future connections to the same endpoint.

(3) Other rather complex discovery algorithms

Other rather complex discovery algorithms that consider various PMTU candidate sequences will be dealt with in the future.

3. Resolving PMTU Black Hole Problem

Classical PMTU discovery is subject to protocol failures. One failure arises when traffic using a packet size larger than the actual PMTU is black-holed. That is, all datagrams larger than the actual PMTU are discarded, which is known as PMTU black hole problem. This could arise when the PTB messages are not sent back to the sender for some reason. In extreme cases, such as the secure and reliable transport protocol QUIC, a sender who only trusts encrypted security information will not use PTB messages.

The main idea of [RFC8899] is to prevent an endpoint from unintentionally sending packets that are too big by limiting their size using a PMTU estimation that is equal or smaller than the actual PMTU. The discovery begins with a PMTU search that provides successively increasing estimates of the actual PMTU. This process does not require PTB messages. However, a PMTU can change. A decrease in PMTU may cause the endpoint to transmit packets that are too large. [RFC8899] does not describe how to detect this without a PTB message.

Recently, [Q-PMTUBH] presents a new parameterizable PTB detection algorithm for a secure and reliable transport protocol that does not depend on PTB messages. [Q-PMTUBH] chooses QUIC as an example to illustrate how to integrate the new parameterizable PTB detection algorithm into a transport protocol and elaborate it with different parameter values using the QUIC simulation model. Therefore, this new PTB detection algorithm can be a solution to the PMTU black hole problem in QUIC.

Additionally, when applying QUIC in an IPv6 environment, the new IPv6 Hop-by-Hop (HBH) Option, which is used to record or cache the minimum PMTU along the forwarding path between the source and destination hosts, can be a solution. The concept of recording or caching the minimum PMTU was originated by [O-PMTUD] and standardized by [RFC9268].

4. Path MTU Discussion for Media Delivery over UDP

MoQ and WebRTC are both designed to handle real-time communication and media streaming over UDP [MOQT][RFC8835]. However, they cater to different use cases and architectural preferences, with distinct design philosophies and trade-offs.

WebRTC is a monolithic architecture designed for peer-to-peer-video telephony, with a lot complexity under the hood but less flexibility for video stream configuration[RFC8835]. Additional requirements for STUN and TURN servers make deployments challenging.

MoQ is based a modular architecture based on modern standard technologies like WebTransport, HTTP/3, and QUIC[MOQT]. MoQ is a set of protocols designed to provide a low-latency, high-quality media delivery solution, especially for streaming. It leverages QUIC’s efficiency and reliability for media ingest and distribution.

With regard to PMTU, there is a requirement for WebRTC Peer-to-Peer (P2P) data channels between two browsers[RFC 8831]. The WebRTC data channel transport protocol should avoid IP fragmentation. It ust support Path MTU (PMTU) discovery and must not rely on ICMP or ICMPv6 being generated or being passed back, especially for PMTUD. WebRTC typically assumes a smaller MTU of 1200 bytes, and PMTUD helps ensure that packets stay within this size, even when traversing networks with varying MTU settings. WebRTC often uses a lower MTU (around 1200 bytes) than the standard Ethernet MTU (1500 bytes) to ensure compatibility with various network environments, including VPN tunnels and encryption layers. WebRTC leverages PMTUD to dynamically adjust packet sizes, ensuring efficient and reliable communication even when the underlying network has varying MTU settings.

In addition, even in point-to-point RTP sessions, this also allows senders to piggyback audio media in the same UDP packet as video media, for example, and also allows QUIC receivers to piggyback QUIC ACK frames on any QUIC packets being transmitted in the other direction.

PMTUD in the context of MoQ involves determining the MTU of the network path between a client and server to avoid packet fragmentation. This ensures efficient and reliable media delivery over QUIC, which is a fast and efficient transport protocol. MoQ utilizes QUIC’s ability to determine the optimal PMTU, even within the application data stream, for low-latency and high-quality streaming.

By determining the optimal PMTU, MoQ can ensure that packets are sent in the largest possible size without fragmentation, leading to improved performance, especially for high-speed internet connections like 5G.

MoQ can implement PMTUD through various algorithms, including the passive probing where the actual application data is used to discover the PMTU. This means that PMTUD can be performed in parallel with data transmission, reducing overhead.

5. IANA Considerations

This memo includes no request to IANA.

6. Security Considerations

The same security considerations as those described in RFC7880 will apply to this document.

7. 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>.

[RFC8201] McCann, J., S. Deering, J. Mogul, R. Hinden, Ed. "Path MTU Discovery for IP version 6", RFC 8201, DOI 10.17487/RFC8201, July 2017, <https://www.rfc-editor.org/info/rfc8201>.

[RFC8831] Jesup, R., S. Loreto, M. Tuxen, "WebRTC Data Channels", RFC 8831, DOI 10.17487/RFC8831, January 2021, <https://www.rfc-editor.org/info/rfc8831>.

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

[RFC9000] J. Iyengar, Ed., 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>.

[RFC8835] H. Alvestrand, "Transports for WebRTC", RFC 8835, DOI 10.17487/RFC8835, January 2021, <https://www.rfc-editor.org/info/rfc8835>.

[UDP-PMTUD] Work in Progress, Internet-Draft, draft-ietf-tsvwg-udp-options-dplpmtud-15, 20 February 2025, <https://www.ietf.org/archive/id/draft-ietf-tsvwg-udp-options-dplpmtud-15.txt>.

[MOQT] Work in Progress, Internet-Draft, draft-ietf-moq-transport-12, 23 June 2025, <https://www.ietf.org/archive/id/draft-ietf-moq-transport-12.txt>.

[Q-PMTUD] Timo Volker, Michael Tuxen, "The search of the path MTU with QUIC", EPIQ ’21: Proceedings of the 2021 Workshop on Evolution, Performance and Interoperability of QUIC, December 2021

[Q-PMTUBH]
Timo Volker, Michael Tuxen, "Packet Too Big Detection and its Integration into QUIC", 2023 16th International Conference on Signal Processing and Communication System (ICSPCS), September 2023

[O-PMTUD]
Expired, Internet-Draft, draft-lee-optimal-detect-pmtu-00, 8 October 2002, <https://datatracker.ietf.org/doc/draft-lee-optimal-detect-pmtu>.

[RFC9268] B. Hinden, G. Fairhurs, "IPv6 Minimum Path MTU Hop-by-Hop Option", RFC 9268, August 2022, <https://datatracker.ietf.org/doc/rfc9268/>.

Authors' Addresses

Pyung Soo Kim
Tech University of Korea
Siheung, Gyeonggi
South Korea
Email: pskim@tukorea.ac.kr