Workgroup: MASQUE
Internet-Draft:
draft-yang-masque-dgram-retrans-01
Published: 13 March 2023
Intended Status: Experimental
Expires: 14 September 2023
Authors: F. Yang Alibaba Inc.
Y. Liu Alibaba Inc.
Y. Ma Alibaba Inc.

A Configurable Retransmission Extension for HTTP/3 Datagrams

Abstract

When using HTTP/3 Datagrams for traffic tunneling, it is desirable to retransmit HTTP/3 Datagrams in some scenarios where the retransmission is beneficial for the tunneled end-to-end connection. This document defines an extension to the HTTP Datagrams and the Capsule Protocol, which allows HTTP/3 Datagrams to be retransmitted according to the configuration of the HTTP/3 Datagram flow.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Multiplexed Application Substrate over QUIC Encryption Working Group mailing list (masque@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/masque/.

Source for this draft and an issue tracker can be found at https://github.com/yangfurong/draft-yang-masque-retx-dgrams.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

HTTP Datagrams and the Capsule Protocol [HTTP-DATAGRAM] defines how HTTP Datagrams can be sent either unreliably using the QUIC DATAGRAM extension [QUIC-DATAGRAM] or reliably using the Capsule Protocol that encapsulates HTTP Datagrams into HTTP/2 [RFC7540] streams, HTTP/3 [RFC9114] streams or HTTP/1.x connections. The two modes, "reliable mode" and "unreliable mode", all have their pros and cons.

This document takes the scenario where HTTP Datagrams are leveraged to tunnel QUIC [QUIC] connections from a QUIC client and a target QUIC server via an HTTP UDP proxy [CONNECT-UDP] as a reference. However, the problems discussed below are not restricted to the reference scenario. Instead, the problems are general in other scenarios using HTTP Datagrams for traffic tunneling, e.g. [CONNECT-IP].

In the reference scenario, the reliable mode is usually worse than the unreliable mode in terms of the transport performance of the end-to-end QUIC connection (i.e. the connection tunneled by the proxy). The culprit is that the stream-based Capsule Protocol can stall the end-to-end QUIC connection due to head-of-line blocking, which can inflate the RTT estimation of the end-to-end connection, make the connection perceive bursty losses, and hinder different streams of the connection from independent delivery. However, the reliable mode also has advantages sometimes. If the network path between the client and the UDP proxy is lossy and the end-to-end delay is a few times higher than the delay of the tunnel, the reliable mode can quickly recover the lost packets in the tunnel, hide the losses from the end-to-end connection, and avoid the reduction of the connection's congestion window. Some of the above behaviors were observed by a study [MASQUE-EVALUATION].

This document defines an extension to the Capsule Protocol [HTTP-DATAGRAM], which allows HTTP/3 Datagrams to be retransmitted according to the configuration of the HTTP/3 Datagram flow. In Section 4, a new Capsule Type is added to configure peers' retransmission limit of HTTP/3 Datagrams. Having such a signaling mechanism instead of just locally configuring the retransmission capability at endpoints (i.e. the client and the proxy) is necessary for enforcing retransmission policies in both upstream and downstream directions. As the proxy does not know the end-to-end connection's preference for retransmission, the client needs to inform the proxy what is the retransmission preference. Depending on the retransmission limit of HTTP/3 Datagrams, the handling of lost HTTP/3 Datagrams is discussed in Section 6.

This extension brings the benefits of the reliable mode to the unreliable mode. It is beneficial for traffic tunneling scenarios where the last-mile link could be very lossy (e.g. Apple's iCloud Private Relay scenario [PR] where the last-mile link is usually wireless).

2. Conventions and Definitions

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

This document uses the notation from [QUIC] for the format of the new capsule definition. Where fields are encoded using the variable-length integer, they need not be encoded on the minimum number of bytes.

In this document, the term "UDP proxy" aligns with the definition in [CONNECT-UDP], and the term "intermediary" refers to an HTTP intermediary as defined in Section 3.7 of [RFC9110].

The term "HTTP/3 Datagram flow" describes the HTTP/3 Datagrams associated with the same HTTP request, .e.g a Connect-UDP request [CONNECT-UDP] or a Connect-IP request [CONNECT-IP].

3. Negotiating The Extension Between Peers

Peers indicate support for this extension by including the boolean-valued Item Structured Field "DG-Retrans: ?1" in the HTTP Request and Response headers (See Section 3.3.6 of [RFC8941] for information about the boolean format.). Peers **MUST NOT** use any following mechanisms described by this extension unless the support is explicitly expressed.

4. Signaling HTTP/3 Datagram Retransmission Limit

This document defines a new Capsule Type SET_H3_DGRAM_RETX_LIMIT to communicate how many times an HTTP/3 Datagram can be retransmitted at most between peers. Note, the retransmission limit takes effect within the scope of an HTTP/3 Datagram flow.

The format of the SET_H3_DGRAM_RETX_LIMIT capsule is shown in Figure 1. It has the following fields:

Context ID: It is the Context ID defined in [CONNECT-UDP] or [CONNECT-IP]. It describes the effect scope of the capsule. It is optional. If the Capsule Type is 0xbb (tentative), the capsule has no Context ID field, and the retransmission limit applies to all contexts.

Retransmission Limit: It is the maximum retransmission number of an HTTP/3 Datagram.

```
SET_H3_DGRAM_RETX_LIMIT {
    Capsule Type (i) = 0xba..0xbb,
    Capsule Length (i),
    [Context ID (i)],
    Retransmission Limit (i),
}
```

Figure 1: SET_H3_DGRAM_RETX_LIMIT Format

When a peer that recognizes SET_H3_DGRAM_RETX_LIMIT capsules receives a SET_H3_DGRAM_RETX_LIMIT capsule, if it is using HTTP/3 Datagrams, it **MUST** start to retransmit lost HTTP/3 Datagrams until they are acknowledged or their retransmission limit specified in the

capsule is reached. If the peer is an intermediary, it **SHOULD NOT** forward the capsule to the next hop, as the aim of retransmissions is to recover the lost packets at the probably lossy last-mile link between the client and the first hop proxy. If an intermediary does not recognize SET_H3_DGRAM_RETX_LIMIT capsules, it **SHOULD** forward the capsules without any modification for the future extensibility as suggested by [HTTP-DATAGRAM].

Finding the best way to set the limit of retransmission is out of this document's scope. Nonetheless, a possible way to calculate the retransmission limit is as follows. Considering the reference scenario of this document (shown in Figure 2), the client can set its local retransmission limit to floor(RTT2 / RTT1) and use the SET_H3_DGRAM_RETX_LIMIT capsule to set the proxy's retransmission limit to floor(RTT2 / RTT1). As the loss detection algorithm takes at least one RTT to detect a packet loss, this setting intends to only allow a lost packet to be retransmitted by the tunnel before it is retransmitted by the end-to-end QUIC connection. Note, the client can subtract RTT1 from the RTT of the end-to-end QUIC connection to get RTT2.

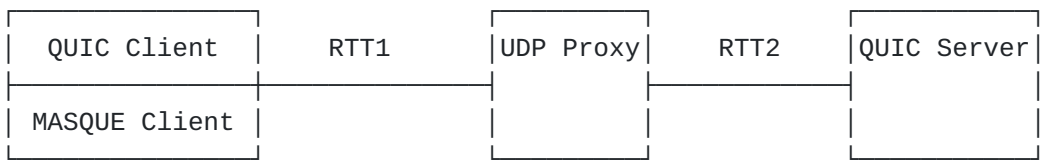

Figure 2: The reference scenario

5. Updating HTTP/3 Datagram Retransmission Limit

A peer can just send a new SET_H3_DGRAM_RETX_LIMIT capsule to update the retransmission limit of its peer if necessary. Note, the new limit will overwrite the old limit specified by a previous SET_H3_DGRAM_RETX_LIMIT capsule.

6. Handling Lost HTTP/3 Datagrams

HTTP/3 Datagrams are encoded in QUIC DATAGRAM frames. As described in [QUIC-DATAGRAM], QUIC **MAY** notify the sender upon a QUIC DATAGRAM frame is acknowledged or declared lost by the loss detection algorithm. This extension relies on the notifications of the acknowledgement and loss of QUIC DATAGRAM frames to handle the retransmission of lost HTTP/3 Datagrams.

A reference way of implementation is as follows. First, when the HTTP/3 Datagram layer calls the unreliable sending API of QUIC to send an HTTP/3 Datagram, it gets a connection-level unique ID (DATAGRAM_ID) from QUIC that corresponds to the underlying QUIC

DATAGRAM frame. Then, if the retransmission limit is larger than zero, the HTTP/3 Datagram layer generates a record {id = DATAGRAM_ID, retx_times = 0} for the HTTP/3 Datagram. Afterwards, whether the HTTP/3 Datagram is acknowledged or declared lost, the HTTP/3 Datagram layer will get a corresponding notification. For the acknowledgement notification, the HTTP/3 Datagram layer just deletes the record. For the loss notification, the HTTP/3 Datagram layer retransmits the HTTP/3 Datagram and updates the id and retx_times of the record if the retransmission limit permits, otherwise, the record is deleted. Note, as QUIC holds the HTTP/3 Datagram as the payload of the QUIC DATAGRAM frame, the payload can be returned to the HTTP/3 Datagram layer for retransmission, which saves the HTTP/3 Datagram layer from buffering HTTP/3 Datagrams for retransmission.

7. Security Considerations

This extension adds no additional considerations to those presented in [HTTP-DATAGRAM].

8. IANA Considerations

This document adds following entry to the "Hypertext Transfer Protocol (HTTP) Field Name Registry":

Header Field	Status	Reference
DG-Retrans	Exp	This document

Table 1: New HTTP Header Field

This document adds following entries to the "HTTP Capsule Types" registry:

Capsule Type	Value	Specification
SET_H3_DGRAM_RETX_LIMIT	0xba, 0xbb	This document

Table 2: New Capsule Type

9. References

9.1. Normative References

[CONNECT-IP] Pauly, T., Schinazi, D., Chernyakhovsky, A., Kühlewind, M., and M. Westerlund, "IP Proxying Support for HTTP", Work in Progress, Internet-Draft, draft-ietf-masque-connect-ip-03, 27 September 2022, <https://

datatracker.ietf.org/doc/html/draft-ietf-masque-connect-ip-03>.

[CONNECT-UDP] Schinazi, D., "Proxying UDP in HTTP", RFC 9298, DOI 10.17487/RFC9298, August 2022, <https://www.rfc-editor.org/rfc/rfc9298>.

[HTTP-DATAGRAM] Schinazi, D. and L. Pardue, "HTTP Datagrams and the Capsule Protocol", RFC 9297, DOI 10.17487/RFC9297, August 2022, <https://www.rfc-editor.org/rfc/rfc9297>.

[QUIC] 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/rfc/rfc9000>.

[QUIC-DATAGRAM] Pauly, T., Kinnear, E., and D. Schinazi, "An Unreliable Datagram Extension to QUIC", RFC 9221, DOI 10.17487/RFC9221, March 2022, <https://www.rfc-editor.org/rfc/rfc9221>.

[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/rfc/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/rfc/rfc8174>.

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/rfc/rfc9110>.

9.2. Informative References

[MASQUE-EVALUATION] Kühlewind, M., Carlander-Reuterfelt, M., Ihlar, M., and M. Westerlund, "Evaluation of QUIC-based MASQUE proxying", Proceedings of the 2021 Workshop on Evolution, Performance and Interoperability of QUIC, DOI 10.1145/3488660.3493806, December 2021, <https://doi.org/10.1145/3488660.3493806>.

[PR] Apple Inc., "iCloud Private Relay Overview", 2021.

[RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <https://www.rfc-editor.org/rfc/rfc7540>.

[RFC8941] Nottingham, M. and P-H. Kamp, "Structured Field Values for HTTP", RFC 8941, DOI 10.17487/RFC8941, February 2021, <https://www.rfc-editor.org/rfc/rfc8941>.

[RFC9114] Bishop, M., Ed., "HTTP/3", RFC 9114, DOI 10.17487/RFC9114, June 2022, <https://www.rfc-editor.org/rfc/rfc9114>.

Contributors

TBD.

Acknowledgments

The authors would like to thank Qinghua Wu, Jiaxing Zhang, and Zhenyu Li for discussions and comments on the design of this draft.

Authors' Addresses

Furong Yang
Alibaba Inc.

Email: yfr256538@alibaba-inc.com

Yanmei Liu
Alibaba Inc.

Email: miaoji.lym@alibaba-inc.com

Yunfei Ma
Alibaba Inc.

Email: yunfei.ma@alibaba-inc.com