Workgroup: Media Over QUIC
Internet-Draft:
draft-ma-moq-relay-for-deadline-02
Published: 3 March 2024
Intended Status: Standards Track
Expires: 4 September 2024
Authors: Y. Cui
Tsinghua University
C. Ma
Tsinghua University
Y. Liao
Tsinghua University
H. Shi
Huawei

MoQ relay for support of deadline-aware media transport

Abstract

This draft specifies the behavior of MoQ relays for delivering media before the deadline to decrease end-to-end latency and save transport costs in media transmission. To achieve this, the draft introduces deadline-aware actions prioritizing media streams with earlier deadlines, ensuring timely transmission while minimizing costs.

About This Document

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

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ma-moq-relay-for-deadline/.

Source for this draft and an issue tracker can be found at https://github.com/STAR-Tsinghua/draft-moq-for-deadline.

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 4 September 2024.

Table of Contents

1. Introduction

Media over QUIC (MoQ) is a transport system designed to provide efficient media transport. However, some use cases, such as live streaming, online meetings, and gaming, require the client to receive their media before a specific time, referred to as the 'deadline.' Exceeding the deadline results in dropped data, which can increase latency and negatively affect user experience.

To address this issue, a deliver-before-deadline transport service can be provided, which is the goal of the Deadline-aware Transport

Protocol (DTP) proposed in [I-D.draft-shi-quic-dtp]. DTP leverages stream-level scheduling, active stream canceling, and redundancy coding to prioritize urgent data and prevent outdated data from blocking later data.

This document proposes the behavior of deadline-aware actions on MoQ relay nodes, extending the basic MoQ relay to provide deliver-before-deadline transmission. The relay design utilizes data scheduling, data canceling, and redundancy coding to decrease queuing time, prevent unnecessary re-transmission of overdue data, and ultimately reduce end-to-end latency. By providing better data delivery strategies, MoQ relays with deadline-aware actions can significantly enhance overall user experience in media transport.

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.

3. Overview of Deadline-aware MoQ Architecture

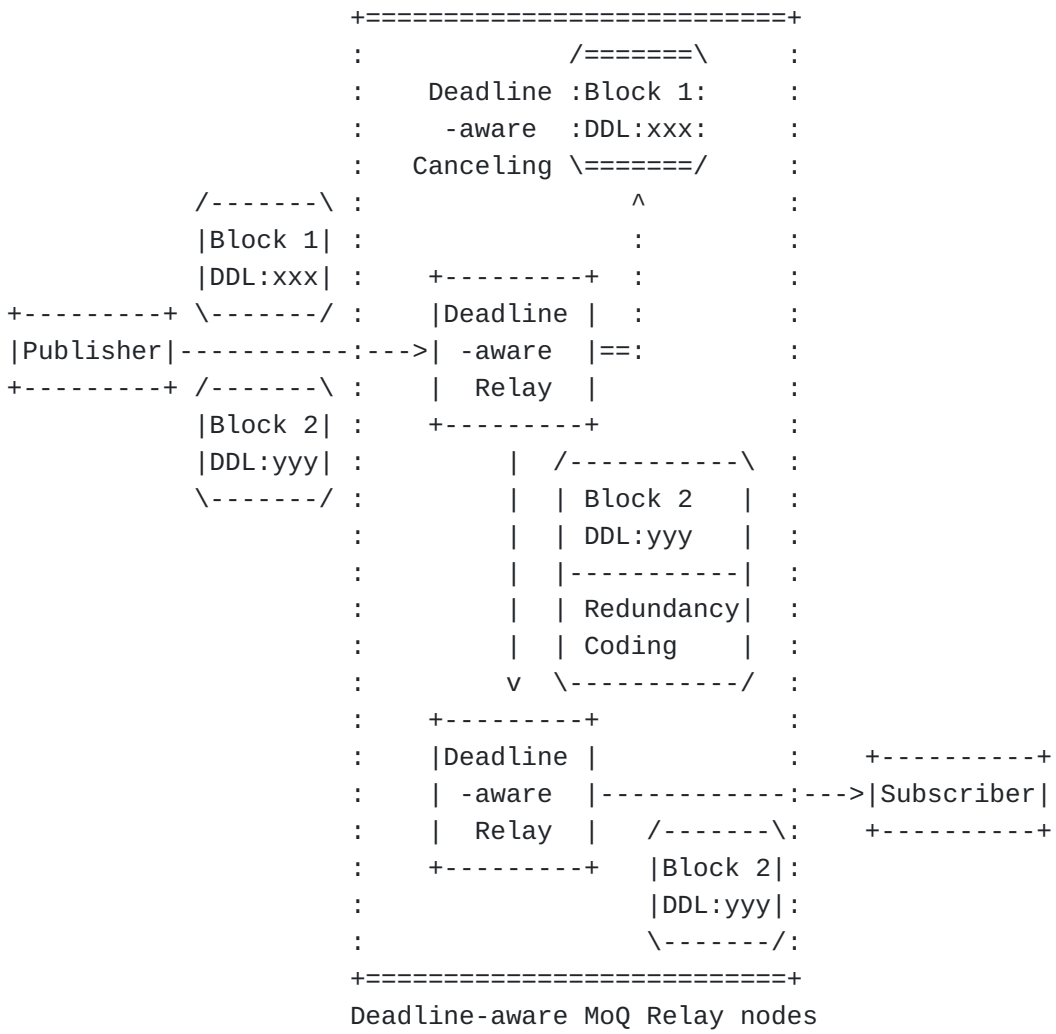

Figure 1: The Architecture of Deadline-aware MoQ

Figure 1 illustrates the fundamental architecture of Deadline-aware MoQ. This architecture involves the extension of MoQ Publishers and Subscribers, which transport block-like data and add 'Deadline' as a component of Metadata within the header. Relay nodes within this system are equipped with deadline-aware actions, including deadline-aware scheduling, canceling, and redundancy coding. The relay may schedule the data blocks, cancel the overdue ones, and add redundancy code to avoid re-transmission. The relays receive block-like data from the publisher, transfer between relays, make deadline-aware actions, and transmit it to the subscriber.

The main focus of this draft is proposing an extension of MoQ relays, the 'Deadline-aware MoQ Relay.' The Deadline-aware MoQ Relay **SHOULD** send data in a block-like style to enable deadline-aware actions. A Block is a basic data unit in the MoQ system, like the Object in [MOQT]. A Block **SHOULD** contain, at a minimum, a Block ID field in the its header to distinguish it from others.

Depending on the implementation of MoQ , the implementation of MoQ relay may map the block transmission to different mechanisms of [QUIC], such as matching a Block to multiple QUIC datagrams or a single QUIC stream. Deadline-aware MoQ Relay **SHOULD** support various MoQ transport implementations. When the relay receives data without any deadline-related information from the endpoint, it **SHOULD** forward it without utilizing any deadline-aware actions.

The Deadline-aware MoQ Relay **SHOULD** support various relay topologies, as discussed in [I-D.draft-shi-moq-design-space-analysis-of-moq]. Each relay topology may require a different MoQ architecture or implementation. Therefore, the deadline-aware actions should act as a plugin that relays can quickly implement regardless of the topology and architecture.

4. Deadline-aware Extension of MoQ

4.1. Object Model: Block

In this draft, we utilize the Block as the fundamental unit for data transmission. A Block comprises two essential components: the metadata and the payload. The payload of a Block is a sequence of data bytes that carries the basic unit in media transport, such as a video frame. Meanwhile, a Block's metadata encompasses deadline-related information, which is necessary for enabling Deadline-aware Actions(see Section 4.2). It's worth noting that the metadata of a Block can remain unencrypted, whereas the payload of a Block **SHOULD** be encrypted.

The Block serves as a model exclusively for data transmission within the MoQ framework. Its purpose is to support the design principles of data units within MoQ, like the Object or the Group in [MOQT]. Importantly, it is crucial that the Block model does not supersede or alter the original data transmission model and should adapt to different designs in MoQ.

4.1.1. Metadata

For Deadline-aware MoQ Relay, data block metadata is required to enable deadline-aware actions. Both the endpoint and the relay **SHOULD** attach the following metadata to each data block when using deadline-aware actions:

*block id: the identifier of each block

*size: the size of the payload in bytes

*priority: the block's relative priority in a single session

*deadline: the expected completion time of the block. The relay can drop overdue data.

The relay **SHOULD** maintain track of the metadata of a block until the block misses its deadline.

Additionally, if the endpoint does not offer metadata in the header of a data block, relays **MAY** implement other mechanisms to acquire and synchronize deadline-related metadata.

e.x. The specific methodology for encapsulating metadata needs to wait until the MoQ specification is standardized.

4.1.1.1. Proirity

TODO: At present, we set the priority as a relative value within a session. However, the priority in MOQT refers to a relative sending order in a group. We are considering postponing this aspect until we devise a solution for implementing MOQT-defined priority ordering on Deadline-aware MoQ Relays.

4.1.1.2. Deadline

Deadlines can be defined in two distinct manners: End-to-End Deadline and Hop-by-Hop Deadline.

The End-to-End Deadline indicates the expected end-to-end delay of the application, beyond which a data block is considered obsolete. Conversely, the Hop-by-Hop Deadline represents the anticipated delay between two nodes within each hop. It defines the tolerance for delay between relay nodes but does not convey the end-to-end latency requirement.

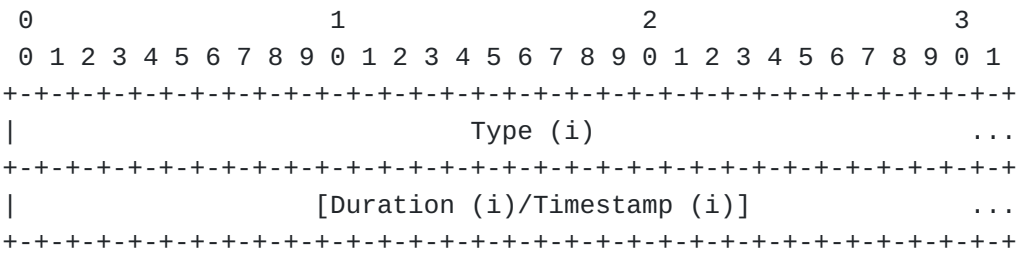

Figure 2: A Prototype Design for the Deadline Field

Figure 2 demonstrates a potential implementation of the Deadline field. The Type field specifies the function of these two fields. The last bit of Type (0x1) indicates whether the Block uses Hop-by-Hop Deadline. The least significant bit of Type (0x1) indicates whether the Block utilizes the Hop-by-Hop Deadline. The second-to-last bit of Type (0x2) indicates whether the Block employs a time duration as the

maximum delay tolerance or a Unix timestamp as the expiration time for the data. A Type value of 0x4 signifies that the Block has no Deadline requirement.

The Deadline-aware MoQ Relay **SHOULD** implement stratergies to manage both End-to-End Deadline and Hop-by-Hop Deadline requirements.

4.2. Deadline-aware Action

4.2.1. Deadline-aware Scheduling and Cancelling

When implementing deadline-aware actions, the Deadline-aware MoQ Relay can utilize the block metadata for scheduling blocks at the block-level. The scheduler **SHOULD** minimize the total time of queuing and try to meet the deadline requirements of as many high-priority blocks as possible.

If a block misses its deadline, Deadline-aware MoQ Relay **MAY** cancel it. In such cases, the endpoints **SHOULD** be able to accept partially received data and not request for data re-transmission when a block is dropped. Additionally, the relays **MAY** inform the endpoints and other relays about the cancellation of these blocks.

4.2.2. Deadline-aware Redundancy Coding

To improve reliability and decrease latency, Deadline-aware MoQ Relay **MAY** introduce redundancy data to blocks close to their deadline or transmitted over a network with a high loss rate. This redundancy can help to prevent the need for re-transmission. If the first relay adds redundancy coding to the data, other relay nodes in the network may benefit from it.

When redundancy coding is enabled, at least two nodes **SHOULD** implement a pair of encoder and decoder that comply with the redundancy coding method. The endpoint **MAY** also implement a redundancy encoder and decoder to utilize the relay's redundancy coding function fully.

The first node that encodes the data with redundancy coding **MUST** add redundancy-related information to the metadata of the data block.

5. Discussions

5.1. Drop Notification

In situations where a data block is dropped by a relay due to a missed deadline or other reasons, sending an explicit dropping message to other relays and both endpoints can be helpful in notifying them of the data loss. The dropping message may include information about the block, such as its ID and metadata. However,

the effective method for notifying other nodes and the decision regarding whether to send drop notifications to other relays are still pending discussion. Broadcasting drop notifications could potentially lead to network flooding and requires further consideration.

5.2. Data Buffer

Further discussion is required to determine if the relay should implement a buffer for data blocks during forwarding and how such a buffer should be implemented. This buffer may be used for re-transmission purposes and may benefit users with larger delay tolerance, among other potential uses.

5.3. Clock Synchronization

To enable accurate deadline-aware actions, it is recommended that all endpoints and relays perform clock synchronization. Nevertheless, achieving high-precision clock synchronization over the Internet can be a formidable task, and it is also impractical to synchronize all devices in a single MoQ session. The challenge then becomes how to carry out deadline-aware actions in the presence of imprecise clock accuracy, which is a critical question that needs to be addressed.

6. Security Considerations

Access to the metadata of the Deadline-aware MoQ Relay **SHOULD** be limited to selected relays. The relay **SHOULD NOT** access the content of the data block.

7. IANA Considerations

This document has no IANA actions.

8. References

8.1. Normative References

[I-D.draft-shi-quic-dtp] Cui, Y., Ma, C., Shi, H., Zheng, K., and W. Wang, "Deadline-aware Transport Protocol", Work in Progress, Internet-Draft, draft-shi-quic-dtp-09, 28 January 2024, <https://datatracker.ietf.org/doc/html/draft-shi-quic-dtp-09>.

[MOQT] Curley, L., Pugin, K., Nandakumar, S., Vasiliev, V., and I. Swett, "Media over QUIC Transport", Work in Progress, Internet-Draft, draft-ietf-moq-transport-02, 24 January 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-moq-transport-02>.

[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>.

[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>.

8.2. Informative References

[I-D.draft-shi-moq-design-space-analysis-of-moq] Shi, H., Cui, Y., and X. Yu, "Design Space Analysis of MoQ", Work in Progress, Internet-Draft, draft-shi-moq-design-space-analysis-of-moq-03, 3 March 2024, <https://datatracker.ietf.org/doc/html/draft-shi-moq-design-space-analysis-of-moq-03>.

Acknowledgments

We sincerely thank Wei Cao for his advice and revisions to this draft.

Authors' Addresses

Yong Cui
Tsinghua University
30 Shuangqing Rd
Beijing
China

Email: cuiyong@tsinghua.edu.cn

Chuan Ma
Tsinghua University
30 Shuangqing Rd
Beijing
China

Email: simonkorl0228@gmail.com

Yixin Liao
Tsinghua University
30 Shuangqing Rd

Beijing
China

Email: lyxceasar@outlook.com

Hang Shi
Huawei

Email: shihang9@huawei.com