Workgroup: MOQ Mailing List
Internet-Draft:
draft-gruessing-moq-requirements-02
Published: 11 July 2022
Intended Status: Informational
Expires: 12 January 2023
Authors: J. Gruessing · Nederlandse Publieke Omroep
S. Dawkins · Tencent America LLC

Media Over QUIC - Use Cases and Requirements for Media Transport Protocol Design

Abstract

This document describes use cases that have been discussed in the IETF community under the banner of "Media Over QUIC", provides analysis about those use cases, recommends a subset of use cases that cover live media ingest, syndication, and streaming for further exploration, and describes requirements that should guide the design of protocols to satisfy these use cases.

Note to Readers

RFC Editor: please remove this section before publication

Source code and issues for this draft can be found at https://github.com/fiestajetsam/draft-gruessing-moq-requirements.

Discussion of this draft should take place on the IETF Media Over QUIC (MoQ) mailing list, at https://www.ietf.org/mailman/listinfo/moq.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes use cases that have been discussed in the IETF community under the banner of "Media Over QUIC", provides analysis about those use cases, recommends a subset of use cases that cover live media ingest, syndication, and streaming for further exploration, and describes requirements that should guide the design of protocols to satisfy these use cases.

1.1. For The Impatient Reader

- Our proposal is to focus on live media use cases, as described in Section 5, rather than on interactive media use cases or on-demand use cases.
- The reasoning behind this proposal can be found in Section 5.1.
- The requirements for protocol work to satisfy the proposed use cases can be found in Section 6.

Most of the rest of this document provides background for these sections.

1.2. Why QUIC For Media?

It is not the purpose of this document to argue against proposals for work on media applications that do not involve QUIC. Such proposals are simply out of scope for this document.

When work on the QUIC protocol ([RFC9000]) was chartered ([QUIC-goals]), the key goals for QUIC were:

- Minimizing connection establishment and overall transport latency for applications, starting with HTTP,
- Providing multiplexing without head-of-line blocking,
- Requiring only changes to path endpoints to enable deployment,
- Enabling multipath and forward error correction extensions, and
- Providing always-secure transport, using TLS 1.3 by default.

These goals were chosen with HTTP ([I-D.draft-ietf-quic-http]) in mind.

While work on "QUIC version 1" (version codepoint 0x00000001) was underway, protocol designers considered potential advantages of the QUIC protocol for other applications. In addition to the key goals for HTTP applications, these advantages were immediately apparent for at least some media applications:

- QUIC endpoints can create bidirectional or unidirectional ordered byte streams.
- QUIC will automatically handle congestion control, packet loss, and reordering for stream data.

- *QUIC streams allow multiple media streams to share congestion and flow control without otherwise blocking each other.
- *QUIC streams also allow partial reliability, since either the sender or receiver can terminate the stream early without affecting the overall connection.
- *With the DATAGRAM extension ([I-D.draft-ietf-quic-datagram]), further partially reliable models are possible, and applications can send congestion controlled datagrams below the MTU size.
- *QUIC connections are established using an ALPN.
- *QUIC endpoints can choose and change their connection ID.
- *QUIC endpoints can migrate IP address without breaking the connection.
- *Because QUIC is encapsulated in UDP, QUIC implementations can run in user space, rather than in kernel space, as TCP typically does. This allows more room for extensible APIs between application and transport, allowing more rapid implementation and deployment of new congestion control, retransmission, and prioritization mechanisms.
- *QUIC is supported in browsers via HTTP/3 or WebTransport.
- *With WebTransport, it is possible to write libraries or applications in JavaScript.

The specific advantages of interest may vary from use case to use case, but these advantages justify further investigation of "Media Over QUIC".

2. Terminology

2.1. The Many Meanings of "Media Over QUIC"

Protocol developers have been considering the implications of the QUIC protocol ([RFC9000]) for media transport for several years, resulting in a large number of possible meanings of the term "Media Over QUIC", or "MOQ". As of this writing, "Media Over QUIC" has had at least these meanings:

- *any kind of media carried directly over the QUIC protocol, as a QUIC payload
- *any kind of media carried indirectly over the QUIC protocol, as an RTP payload ([RFC3550])
- *any kind of media carried indirectly over the QUIC protocol, as an HTTP/3 payload
- *any kind of media carried indirectly over the QUIC protocol, as a WebTransport payload
- *the encapsulation of any Media Transport Protocol in a QUIC payload
- *an IETF mailing list ([MOQ-ml]), which was requested "... for discussion of video ingest and distribution protocols that use

QUIC as the underlying transport", although discussion of other Media Over QUIC proposals have also been discussed there.

There may be IETF participants using other meanings as well.

As of this writing, the second bullet ("any kind of media carried indirectly over the QUIC protocol, as an RTP payload"), seems to be in scope for the IETF AVTCORE working group, and was discussed at some length at the February 2022 AVTCORE working group meeting [AVTCORE-2022-02], although no drafts in this space have yet been adopted by the AVTCORE working group.

2.2. Latency Requirement Categories

}The "Operational Considerations for Streaming Media" document ([I-D.draft-ietf-mops-streaming-opcons]) described a range of latencies of interest to streaming media providers, as

*ultra low-latency (less than 1 second)
*low-latency live (less than 10 seconds)
*non-low-latency live (10 seconds to a few minutes)
*on-demand (hours or more)

Because the IETF Media Over QUIC community now expresses interest in interactive media Section 4.1 and live media Section 4.3} use cases will have requirements that are significantly less than the "streaming media"-defined "ultra-low latency".

Within this document, we are using

*near real-time (less than 50 ms)
*Ull-200 (less than 200 ms)

Perhaps obviously, these last two latency bands are the shortened form of "ultra-low latency - 50 ms" and "ultra-low-latency - 200 ms".

Perhaps less obviously, bikeshedding on better names and more useful values is welcomed.

3. Prior and Existing Specifications

Several draft specifications have been proposed which either encapsulate existing Media Transport Protocols in QUIC ([I-D.draft-sharabayko-srt-over-quic]), make use of RTP, RTCP, and SDP ([I-D.draft-engelbart-rtp-over-quic]) or define their own new Media Transport Protocol on top of QUIC. Some have already seen deployment into the wild (e.g. [I-D.draft-kpugin-rush], [I-D.draft-lcurley-warp]) where as others are unconfirmed. Whilst most just focus on defining wire format, [I-D.draft-jennings-moq-quicr-arch] defines an architecture using a pub/sub model for both producers and consumers.

3.1. Comparison of Existing Specifications

- *Some use QUIC Datagram frames, while others use QUIC streams.
- *All drafts take differing approaches to flow/stream identification and management. Some address congestion control and others just defer this to QUIC to handle.
- *Some drafts specify ALPN identification, while others do not.

4. Use Cases Informing This Proposal

Our goal in this section is to understand the range of use cases that have been proposed for "Media Over QUIC".

Although some of the use cases described in this section came out of "RTP over QUIC" proposals, they are worth considering in the broader "Media Over QUIC" context, and may be especially relevant to MOQ, depending on whether "RTP over QUIC" requires major changes to RTP and RTCP, in order to meet the requirements arising out of the corresponding use cases.

An early draft in the "media over QUIC" space, [I-D.draft-rtpfolks-quic-rtp-over-quic], defined several key use cases. Some of the following use cases have been inspired by that document, and others have come from discussions with the wider MOQ community (among other places, a side meeting at IETF 112).

For each use case in this section, we also define

- *the number of senders or receiver in a given session transmitting distinct streams,
- *whether a session has bi-direction flows of media from senders and receivers, and
- *the expected lowest latency requirements using the definitions specified in Section 2.

It is likely that we should add other characteristics, as we come to understand them.

4.1. Interactive Media

The use cases described in this section have one particular attribute in common - the target latency for these cases are on the order of one or two RTTs. In order to meet those targets, it is not possible to rely on protocol mechanisms that require multiple RTTs to function effectively. For example,

- When the target latency is on the order of one RTT, it makes sense to use FEC [RFC6363] and codec-level packet loss concealment [RFC6716], rather than selectively retransmitting only lost packets. These mechanisms use more bytes, but do not require multiple RTTs in order to recover from packet loss.
- When the target latency is on the order of one RTT, it is impossible to use congestion control schemes like BBR [I-D.draft-cardwell-iccrg-bbr-congestion-control], since BBR has probing mechanisms that rely on temporarily inducing delay and amortizing the consequences of that over multiple RTTs.

This may help to explain why these use cases often rely on protocols such as RTP [RFC3550], which provide low-level control of packetization and transmission.

4.1.1. Gaming

Attribute	Value
Senders/Receivers	One to One
Bi-directional	Yes
Latency	Ull-50

Table 1

Where media is received, and user inputs are sent by the client. This may also include the client receiving other types of signalling, such as triggers for haptic feedback. This may also carry media from the client such as microphone audio for in-game chat with other players.

4.1.2. Remote Desktop

Attribute	Value
Senders/Receivers	One to One
Bi-directional	Yes
Latency	Ull-50

Table 2

Where media is received, and user inputs are sent by the client. Latency requirements with this usecase are marginally different than the gaming use case. This may also include signalling and/or transmitting of files or devices connected to the user's computer.

4.1.3. Video Conferencing/Telephony

Attribute	Value
Senders/Receivers	Many to Many

Attribute	Value
Bi-directional	Yes
Latency	Ull-50 to Ull-200

Table 3

Where media is both sent and received; This may include audio from both microphone(s) or other inputs, or may include "screen sharing" or inclusion of other content such as slide, document, or video presentation. This may be done as client/server, or peer to peer with a many to many relationship of both senders and receivers. The target for latency may be as large as Ull-200 for some media types such as audio, but other media types in this use case have much more stringent latency targets.

4.2. Hybrid Interactive and Live Media

For the video conferencing/telephony use case, there can be additional scenarios where the audience greatly outnumbers the concurrent active participants, but any member of the audience could participate. As this has a much larger total number of participants - as many as Live Media Streaming Section 4.3.3, but with the bi-directionality of confercing, this should be considered a "hybrid".

4.3. Live Media

The use cases in this section, unlike the use cases described in Section 4.1, still have "humans in the loop", but these humans expect media to be "responsive", where the responsiveness is more on the order of 5 to 10 RTTs. This allows the use of protocol mechanisms that require more than one or two RTTs - as noted in Section 4.1, end-to-end recovery from packet loss and congestion avoidance are two such protocol mechanisms that can be used with Live Media.

To illustrate the difference, the responsiveness expected with videoconferencing is much greater than watching a video, even if the video is being produced "live" and sent to a platform for syndication and distribution.

4.3.1. Live Media Ingest

Attribute	Value
Senders/Receivers	One to One
Bi-directional	No
Latency	Ull-200 to Ultra-Low

Table 4

Where media is received from a source for onwards handling into a distribution platform. The media may comprise of multiple audio and/

or video sources. Bitrates may either be static or set dynamically by signalling of connection inforation (bandwidth, latency) based on data sent by the receiver.

4.3.2. Live Media Syndication

Attribute	Value
Senders/Receivers	One to One
Bi-directional	No
Latency	Ull-200 to Ultra-Low

Table 5

Where media is sent onwards to another platform for further distribution. The media may be compressed down to a bitrate lower than source, but larger than final distribution output. Streams may be redundant with failover mechanisms in place.

4.3.3. Live Media Streaming

Attribute	Value
Senders/Receivers	One to Many
Bi-directional	No
Latency	Ull-200 to Ultra-Low

Table 6

Where media is received from a live broadcast or stream. This may comprise of multiple audio or video outputs with different codecs or bitrates. This may also include other types of media essence such as subtitles or timing signalling information (e.g. markers to indicate change of behaviour in client such as advertisement breaks). The use of "live rewind" where a window of media behind the live edge can be made available for clients to playback, either because the local player falls behind edge or because the viewer wishes to play back from a point in the past.

4.4. On-Demand Media

Finally, the "On-Demand" use cases described in this section do not have a tight linkage between ingest and streaming, allowing significant transcoding, processing, insertion of video clips in a news article, etc. The latency constraints for the use cases in this section may be dominated by the time required for whatever actions are required before media are available for streaming.

4.4.1. On-Demand Ingest

Attribute	Value
Senders/Receivers	One to Many
Bi-directional	No

Attribute	Value
Latency	On Demand

Table 7

Where media is ingested and processed for a system to later serve it to clients as on-demand media. This media provided from a pre-recorded source, or captured from live output, but in either case, this media is not immediately passed to viewers, but is stored for "on-demand" retrieval, and may be transcoded upon ingest.

4.4.2. On-Demand Media Streaming

Attribute	Value
Senders/Receivers	One to Many
Bi-directional	No
Latency	On Demand

Table 8

Where media is received from a non-live, typically pre-recorded source. This may feature additional outputs, bitrates, codecs, and media types described in the live media streaming use case.

5. Proposed Scope for "Media Over QUIC"

Our proposal is that "Media Over QUIC" discussions focus first on the use cases described in Section 4.3, which are Live Media Ingest (Section 4.3.1), Syndication (Section 4.3.2), and Streaming (Section 4.3.3). Our reasoning for this suggestion follows.

Each of the above use cases in Section 4 fit into one of three classifications of solutions.

5.1. Analysis for Interactive Use Cases

The first group, Interactive Media, as described in Section 4.1, and covering gaming (Section 4.1.1), screen sharing (Section 4.1.2), and general video conferencing (Section 4.1.3), are largely covered by RTP, often in conjunction with WebRTC [WebRTC], and related protocols today.

Whilst there may be benefit in these use cases having a QUIC based protocol it may be more appropriate given the size of existing deployments to extend the RTP protocols and specifications.

5.2. Analysis for Live Media Use Cases

The second group of classifications, in Section 4.3, covering Live Media Ingest (Section 4.3.1), Live Media Syndication (Section

4.3.2), and Live Media Streaming (Section 4.3.3) are likely the use cases that will benefit most from this work.

Existing ingest and streaming protocols such as HLS [RFC8216] and DASH [DASH] are reaching limits towards how low they can reduce latency in live streaming and for scenarios where low-bitrate audio streams are used, these protocols add a significant amount of overhead compared to the media bitstream itself.

For this reason, we suggest that work on "Media Over QUIC" protocols target these use cases at this time.

5.3. Analysis for On-Demand Use Cases

The third group, Section 4.4, covering On-Demand Media Ingest (Section 4.4.1) and On-Demand Media streaming (Section 4.4.2) is unlikely to benefit from work in this space. Without the same "Live Media" latency requirements that would motivate deployment of new protocols, existing protocols such as HLS and DASH are probably "good enough" to meet the needs of these use cases.

This does not mean that existing protocols in this space are perfect. Segmented protocols such as HLS and DASH were developed to overcome the deficiencies of TCP, as used in HTTP/1.1 [RFC7230] and HTTP/2 [RFC7540], and do not make full use of the possible congestion window along the path from sender to receiver. Other protocols in this space have their own deficiencies. For example, RTSP [RFC7826] does not have easy ways to add support for new media codecs.

Our expectation is that these use cases will not drive work in the "Media Over QUIC" space, but as new protocols come into being, they may very well be taken up for these use cases as well.

6. Requirements for Protocol Work

TODO: Quite a lot, really ...

7. IANA Considerations

This document makes no requests of IANA.

8. Security Considerations

As this document is intended to guide discussion and consensus, it introduces no security considerations of its own.

9. Informative References

[AVTCORE-2022-02]

"AVTCORE 2022-02 interim meeting materials", February 2022, <https://datatracker.ietf.org/meeting/interim-2022-avtcore-01/session/avtcore>.

[DASH] "ISO/IEC 23009-1:2019: Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats (2nd edition)", n.d., <https://www.iso.org/standard/79329.html>.

[I-D.draft-cardwell-iccrg-bbr-congestion-control] Cardwell, N., Cheng, Y., Yeganeh, S. H., Swett, I., and V. Jacobson, "BBR Congestion Control", Work in Progress, Internet-Draft, draft-cardwell-iccrg-bbr-congestion-control-02, 7 March 2022, <https://datatracker.ietf.org/doc/html/draft-cardwell-iccrg-bbr-congestion-control-02>.

[I-D.draft-engelbart-rtp-over-quic] Ott, J. and M. Engelbart, "RTP over QUIC", Work in Progress, Internet-Draft, draft-engelbart-rtp-over-quic-04, 24 June 2022, <https://datatracker.ietf.org/doc/html/draft-engelbart-rtp-over-quic-04>.

[I-D.draft-ietf-mops-streaming-opcons] Holland, J., Begen, A., and S. Dawkins, "Operational Considerations for Streaming Media", Work in Progress, Internet-Draft, draft-ietf-mops-streaming-opcons-10, 21 April 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-mops-streaming-opcons-10>.

[I-D.draft-ietf-quic-datagram] Pauly, T., Kinnear, E., and D. Schinazi, "An Unreliable Datagram Extension to QUIC", Work in Progress, Internet-Draft, draft-ietf-quic-datagram-10, 4 February 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-quic-datagram-10>.

[I-D.draft-ietf-quic-http] Bishop, M., "HTTP/3", Work in Progress, Internet-Draft, draft-ietf-quic-http-34, 2 February 2021, <https://datatracker.ietf.org/doc/html/draft-ietf-quic-http-34>.

[I-D.draft-ietf-webtrans-overview] Vasiliev, V., "The WebTransport Protocol Framework", Work in Progress, Internet-Draft, draft-ietf-webtrans-overview-03, 7 March 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-webtrans-overview-03>.

[I-D.draft-jennings-moq-quicr-arch] Jennings, C. and S. Nandakumar, "QuicR - Media Delivery Protocol over QUIC", Work in Progress, Internet-Draft, draft-jennings-moq-quicr-arch-01, 11 July 2022, <https://datatracker.ietf.org/doc/html/draft-jennings-moq-quicr-arch-01>.

[I-D.draft-kpugin-rush] Pugin, K., Frindell, A., Cenzano, J., and J. Weissman, "RUSH - Reliable (unreliable) streaming protocol", Work in Progress, Internet-Draft, draft-kpugin-rush-01, 7 March 2022, <https://datatracker.ietf.org/doc/html/draft-kpugin-rush-01>.

[I-D.draft-lcurley-warp] Curley, L., "Warp - Segmented Live Media Transport", Work in Progress, Internet-Draft, draft-lcurley-warp-01, 9 July 2022, <https://datatracker.ietf.org/doc/html/draft-lcurley-warp-01>.

[I-D.draft-rtpfolks-quic-rtp-over-quic] Ott, J., Even, R., Perkins, C., and V. Singh, "RTP over QUIC", Work in Progress, Internet-Draft, draft-rtpfolks-quic-rtp-over-quic-01, 1 September 2017, <https://datatracker.ietf.org/doc/html/draft-rtpfolks-quic-rtp-over-quic-01>.

[I-D.draft-sharabayko-srt-over-quic] Sharabayko, M. and M. Sharabayko, "Tunnelling SRT over QUIC", Work in Progress, Internet-Draft, draft-sharabayko-srt-over-quic-00, 28 July 2021, <https://datatracker.ietf.org/doc/html/draft-sharabayko-srt-over-quic-00>.

[MOQ-ml] "Moq -- Media over QUIC", n.d., <https://www.ietf.org/mailman/listinfo/moq>.

[QUIC-goals] "Initial Charter for QUIC Working Group", October 2016, <https://datatracker.ietf.org/doc/charter-ietf-quic/01/>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <https://www.rfc-editor.org/rfc/rfc3550>.

[RFC6363] Watson, M., Begen, A., and V. Roca, "Forward Error Correction (FEC) Framework", RFC 6363, DOI 10.17487/RFC6363, October 2011, <https://www.rfc-editor.org/rfc/rfc6363>.

[RFC6716] Valin, JM., Vos, K., and T. Terriberry, "Definition of the Opus Audio Codec", RFC 6716, DOI 10.17487/RFC6716, September 2012, <https://www.rfc-editor.org/rfc/rfc6716>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/rfc/rfc7230>.

[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>.

[RFC7826] Schulzrinne, H., Rao, A., Lanphier, R., Westerlund, M., and M. Stiemerling, Ed., "Real-Time Streaming Protocol Version 2.0", RFC 7826, DOI 10.17487/RFC7826, December 2016, <https://www.rfc-editor.org/rfc/rfc7826>.

[RFC8216] Pantos, R., Ed. and W. May, "HTTP Live Streaming", RFC 8216, DOI 10.17487/RFC8216, August 2017, <https://www.rfc-editor.org/rfc/rfc8216>.

[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/rfc/rfc9000>.

[WebRTC] "Web Real-Time Communications Working Group", n.d., <https://www.w3.org/groups/wg/webrtc>.

Appendix A. Acknowledgements

The authors would like to thank the many authors of the specifications referenced in Section 3 for their work. The authors would also like to thank Alan Frindell, Luke Curley, and Maxim Sharabayko for text contributions to this draft.

James Gruessing would also like to thank Francesco Illy and Nicholas Book for their part in providing the needed motivation.

Authors' Addresses

James Gruessing
Nederlandse Publieke Omroep
Netherlands

Email: james.ietf@gmail.com

Spencer Dawkins
Tencent America LLC
United States of America

Email: spencerdawkins.ietf@gmail.com