

moq
Internet-Draft
Intended status: Informational
Expires: 16 May 2026

L. Curley
12 November 2025

Media over QUIC - Hang
draft-lcurley-moq-hang-01

Abstract

Hang is a real-time conferencing protocol built on top of moq-lite. A room consists of multiple participants who publish media tracks. All updates are live, such as a change in participants or media tracks.

Discussion Venues

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

Discussion of this document takes place on the Media Over QUIC Working Group mailing list (moq@ietf.org), which is archived at <https://mailarchive.ietf.org/arch/browse/moq/>.

Source for this draft and an issue tracker can be found at <https://github.com/kixelated/moq-drafts>.

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 16 May 2026.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

| | |
|--|---|
| 1. Conventions and Definitions | 2 |
| 2. Terminology | 2 |
| 3. Discovery | 3 |
| 4. Catalog | 4 |
| 4.1. Root | 4 |
| 4.2. Video | 4 |
| 4.3. Audio | 6 |
| 5. Container | 6 |
| 6. Security Considerations | 7 |
| 7. IANA Considerations | 7 |
| 8. Normative References | 7 |
| Acknowledgments | 7 |
| Author's Address | 7 |

1. 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.

2. Terminology

Hang is built on top of moq-lite [moql] and uses much of the same terminology. A quick recap:

- * ***Broadcast***: A collection of Tracks from a single publisher.
- * ***Track***: An series of Groups, each of which can be delivered and decoded *_out-of-order_*.
- * ***Group***: An series of Frames, each of which must be delivered and decoded *_in-order_*.
- * ***Frame***: A sized payload of bytes representing a single moment in time.

Hang introduces additional terminology:

- * ***Room***: A collection of participants, publishing under a common prefix.
- * ***Participant***: A moq-lite broadcaster that may produce any number of media tracks.
- * ***Catalog***: A JSON document that describes each available media track, supporting live updates.
- * ***Container***: A tiny header in front of each media payload containing the timestamp.

3. Discovery

The first requirement for a real-time conferencing application is to discover other participants in the same room. Hang does this using moq-lite's ANNOUNCE capabilities.

A room consists of a path. Any participants within the room MUST publish a broadcast with the room path as a prefix which SHOULD end with the .hang suffix.

For example:

```
/room123/alice.hang  
/room123/bob.hang  
/room456/zoe.hang
```

A participant issues an ANNOUNCE_PLEASE message to discover any other participants in the same room. The server (relay) will then respond with an ANNOUNCE message for any matching broadcasts, including their own.

For example:

```
ANNOUNCE_PLEASE prefix=/room/  
ANNOUNCE suffix=alice.hang active=true  
ANNOUNCE suffix=bob.hang active=true
```

If a publisher no longer wants to participant, or is disconnected somehow, their presence will be unannounced. Publishers and subscribers SHOULD terminate any subscriptions once a participant is unannounced.

```
ANNOUNCE suffix=alice.hang active=false
```

4. Catalog

The catalog describes the available media tracks for a single participant. It's a JSON document that extends the the W3C WebCodecs specification.

The catalog is published as a catalog.json track within the broadcast so it can be updated live as the participant's media tracks change. A participant MAY forgo publishing a catalog if it does not wish to publish any media tracks now and in the future.

The catalog track consists of multiple groups, one for each update. Each group contains a single frame with UTF-8 JSON.

A publisher MUST NOT write multiple frames to a group until a future specification includes a delta-encoding mechanism (via JSON Patch most likely).

4.1. Root

The root of the catalog is a JSON document with the following schema:

```
type Catalog = {  
  "audio": AudioSchema | undefined,  
  "video": VideoSchema | undefined,  
  // ... any custom fields ...  
}
```

Additional fields MAY be added based on the application. The catalog SHOULD be mostly static, delegating any dynamic content to other tracks.

For example, a "chat" section should include the name of a chat track, not individual chat messages. This way catalog updates are rare and a client MAY choose to not subscribe.

This specification currently only defines audio and video tracks.

4.2. Video

A video track contains the necessary information to decode a video stream.

```
type VideoSchema = {
  "renditions": Map<TrackName, VideoDecoderConfig>,
  "priority": u8,
  "display": {
    "width": number,
    "height": number,
  } | undefined,
  "rotation": number | undefined,
  "flip": boolean | undefined,
}
```

The renditions field contains a map of track names to video decoder configurations. See the WebCodecs specification (<https://www.w3.org/TR/webcodecs/#video-decoder-config>) for specifics and registered codecs. Any Uint8Array fields are hex-encoded as a string.

For example:

```
{
  "renditions": {
    "720p": {
      "codec": "avc1.64001f",
      "codedWidth": 1280,
      "codedHeight": 720,
      "bitrate": 6000000,
      "framerate": 30.0
    },
    "480p": {
      "codec": "avc1.64001e",
      "codedWidth": 848,
      "codedHeight": 480,
      "bitrate": 2000000,
      "framerate": 30.0
    }
  },
  "priority": 2,
  "display": {
    "width": 1280,
    "height": 720
  },
  "rotation": 0,
  "flip": false,
}
```

4.3. Audio

An audio track contains the necessary information to decode an audio stream.

```
type AudioSchema = {  
    "renditions": Map<TrackName, AudioDecoderConfig>,  
    "priority": u8,  
}
```

The renditions field contains a map of track names to audio decoder configurations. See the WebCodecs specification (<https://www.w3.org/TR/webcodecs/#audio-decoder-config>) for specifics and registered codecs. Any Uint8Array fields are hex-encoded as a string.

For example:

```
{  
    "renditions": {  
        "stereo": {  
            "codec": "opus",  
            "sampleRate": 48000,  
            "numberOfChannels": 2,  
            "bitrate": 128000  
        },  
        "mono": {  
            "codec": "opus",  
            "sampleRate": 48000,  
            "numberOfChannels": 1,  
            "bitrate": 64000  
        }  
    },  
    "priority": 1,  
}
```

5. Container

Audio and video tracks use a lightweight container to encapsulate the media payload.

Each moq-lite group MUST start with a keyframe. If codec does not support delta frames (ex. audio), then a group MAY consist of multiple keyframes. Otherwise, a group MUST consist of a single keyframe followed by zero or more delta frames.

Each frame starts with a timestamp, a QUIC variable-length integer (62-bit max) encoded in microseconds. The remainder of the payload is codec specific; see the WebCodecs specification for specifics.

For example, h.264 with no description field would be annex.b encoded, while h.264 with a description field would be AVCC encoded.

6. Security Considerations

TODO Security

7. IANA Considerations

This document has no IANA actions.

8. Normative References

- [moql] Curley, L., "Media over QUIC - Lite", Work in Progress, Internet-Draft, draft-lcurley-moq-lite-01, 22 July 2025, <<https://datatracker.ietf.org/doc/html/draft-lcurley-moq-lite-01>>.
- [moqt] Nandakumar, S., Vasiliev, V., Swett, I., and A. Frindell, "Media over QUIC Transport", Work in Progress, Internet-Draft, draft-ietf-moq-transport-15, 20 October 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-moq-transport-15>>.
- [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>>.

Acknowledgments

TODO acknowledge.

Author's Address

Luke Curley
Email: kixelated@gmail.com