

Media over QUIC – MoQ

IETF MoQ

Charter for Working Group

Media over QUIC (moq) will develop a simple low-latency media delivery solution for ingest and distribution of media. This solution addresses use cases including live streaming, gaming, and media conferencing and will scale efficiently. The solution will be implementable in both browser and non-browser endpoints.

The work focuses on building protocol mechanisms for publication of media and means to identify and receive the media.

The media publication protocol will enable sending media including audio, video, and timed metadata, such as closed captions and cue points. The common protocol for publishing media for ingest and distribution will support:

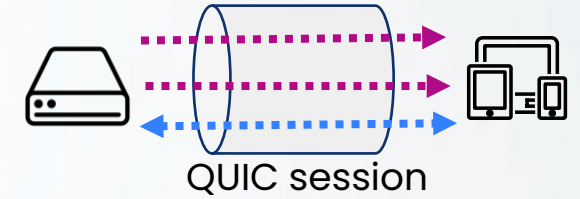
- one or more media formats
- an interoperable way to indicate the media and the media format being sent
- rate adaptation strategies based on changing codec rates, changing chosen media encoding/qualities, or other mechanisms
- cache friendly media mechanisms

Media will be mapped onto underlying QUIC mechanisms (QUIC streams and/or QUIC datagrams) and can be used over raw QUIC or WebTransport.

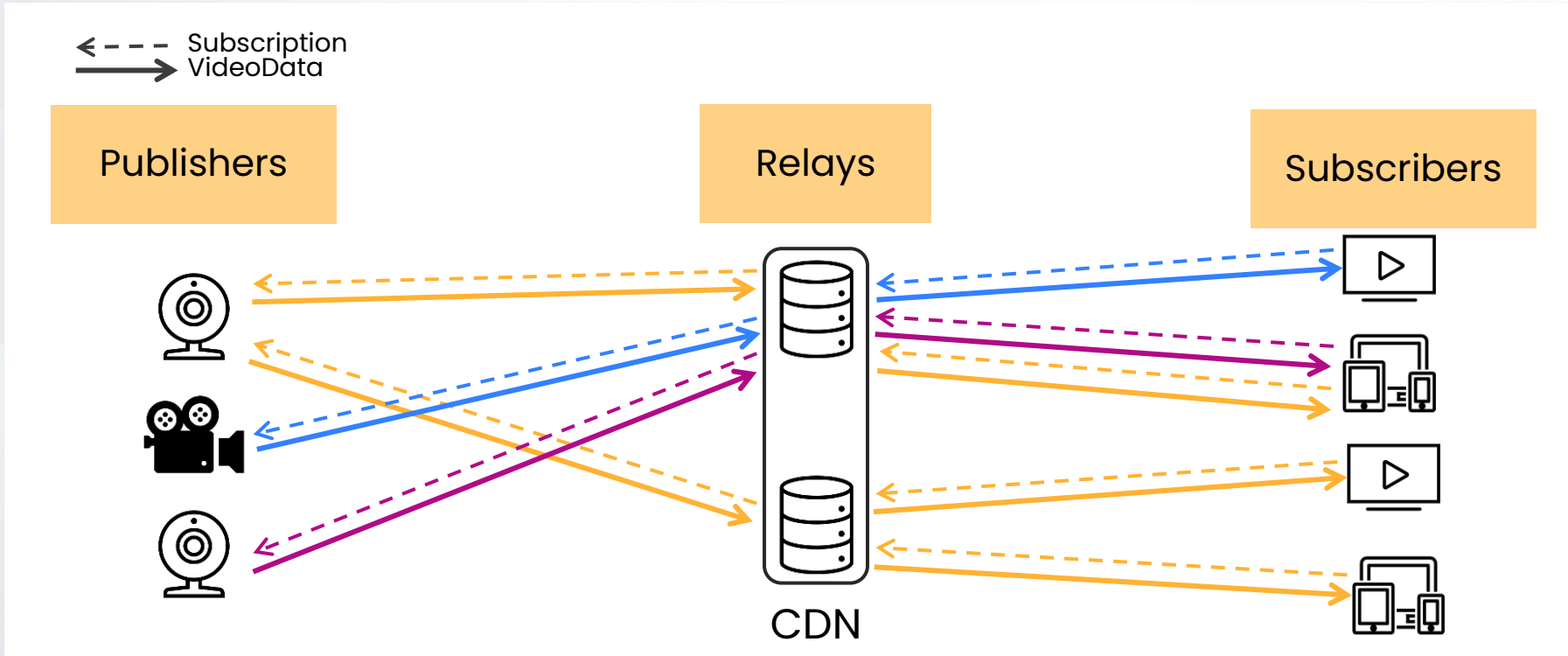
- Chartered in **Sept 2022**
 - Promoters: Cisco, Google, Meta
 - Active members: Akamai, Norsk, Synamedia
- Draft status
 - **Version 07** out in February 2025
 - 709 GitHub issues to date

Cacheable Publish - Subscribe

Unidirectional streams
 • Only **Object** messages

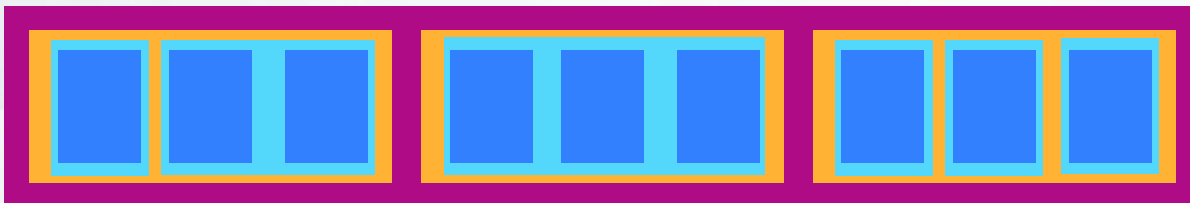
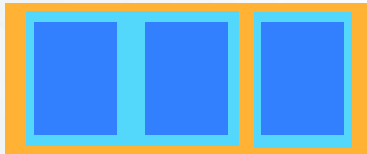


Bidirectional control stream
 • Highest priority
 • All setup messages



ID	Messages
0x2	SUBSCRIBE_UPDATE (Section 7.5)
0x3	SUBSCRIBE (Section 7.4)
0x4	SUBSCRIBE_OK (Section 7.15)
0x5	SUBSCRIBE_ERROR (Section 7.16)
0x6	ANNOUNCE (Section 7.22)
0x7	ANNOUNCE_OK (Section 7.9)
0x8	ANNOUNCE_ERROR (Section 7.10)
0x9	UNANNOUNCE (Section 7.23)
0xA	UNSUBSCRIBE (Section 7.6)
0xB	SUBSCRIBE_DONE (Section 7.19)
0xC	ANNOUNCE_CANCEL (Section 7.11)
0xD	TRACK_STATUS_REQUEST (Section 7.12)
0xE	TRACK_STATUS (Section 7.24)
0x10	GOAWAY (Section 7.3)
0x11	SUBSCRIBE_ANNOUNCES (Section 7.13)
0x12	SUBSCRIBE_ANNOUNCES_OK (Section 7.25)
0x13	SUBSCRIBE_ANNOUNCES_ERROR (Section 7.26)
0x14	UNSUBSCRIBE_ANNOUNCES (Section 7.14)
0x15	MAX_SUBSCRIBE_ID (Section 7.20)
0x1A	SUBSCRIBES_BLOCKED (Section 7.21)
0x16	FETCH (Section 7.7)
0x17	FETCH_CANCEL (Section 7.8)
0x18	FETCH_OK (Section 7.17)
0x19	FETCH_ERROR (Section 7.18)
0x40	CLIENT_SETUP (Section 7.2)
0x41	SERVER_SETUP (Section 7.2)

Principles



Object: The basic data element. An addressable unit.

Subgroup: A collection of objects to transmit in a single QUIC stream

Group: A temporal sequence of **objects**. A join point for subscriptions

Track: A temporal sequence of **groups**. The stream to subscribe to.

Full Track name = Track namespace + Track name

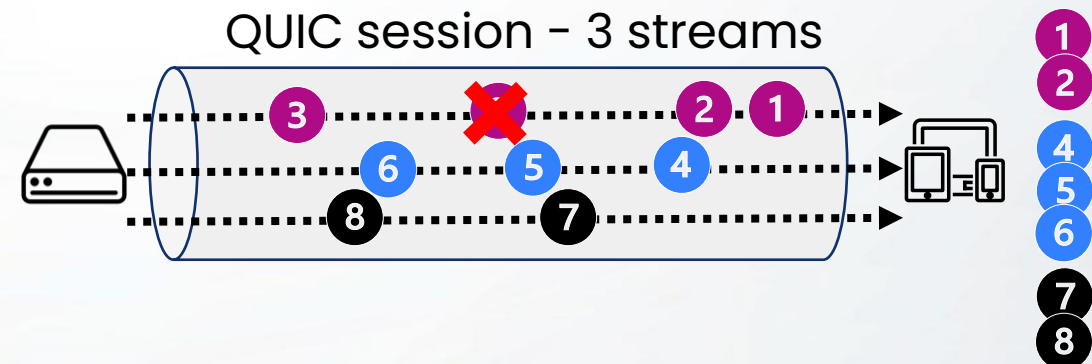
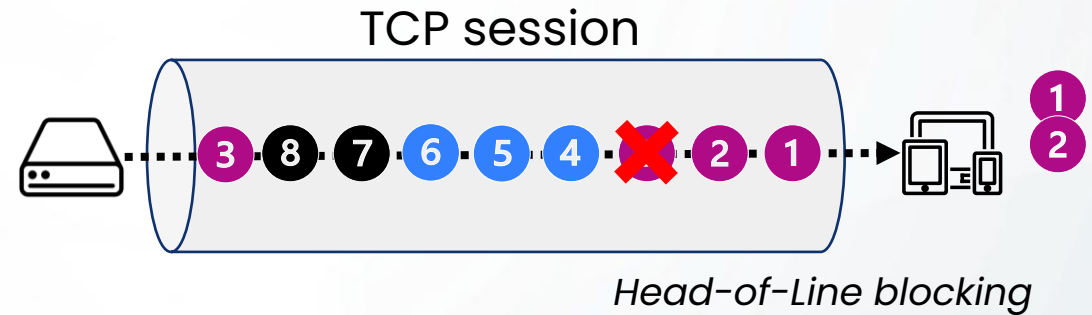
MoQ Subgroups Lever QUIC Streams

QUIC Streams

- Independent
- Priority
- Controllable (RST/FIN)

MoQ Subgroups

- Contains Objects
- Priority (in 0..255 lower better)
- Set by Publisher or Subscriber



Examples

- **Videoconference:** 1- Speaker audio track; 2- Speaker video track; 3- Others
- **Social network:** 1- Current video; 2- Next video; 3- Future videos
- **Video streaming:** 1- I-frame; 2- Independent set of frames in GoP; 3- Another independent frame

Publish-Subscribe Flavors

Fetch

An end user issues a **FETCH** to request a range of **already published objects**.

```
FETCH Message {
  Type (i) = 0x16,
  Length (i),
  Subscribe ID (i),
  Subscriber Priority (8),
  Group Order (8),
  Fetch Type (i),
  [Track Namespace (tuple),
  Track Name Length (i),
  Track Name (..),
  StartGroup (i),
  StartObject (i),
  EndGroup (i),
  EndObject (i),]
  [Joining Subscribe ID (i),
  Preceding Group Offset (i),]
  Number of Parameters (i),
  Parameters (..) ...
}
```

Standalone FETCH

- Range of Objects

Joining FETCH

- Past Objects from Live

Subscribe

An end user issues a **SUBSCRIBE** to get **data** the publisher will publish **in the future**.

```
SUBSCRIBE Message {
  Type (i) = 0x3,
  Length (i),
  Subscribe ID (i),
  Track Alias (i),
  Track Namespace (tuple),
  Track Name Length (i),
  Track Name (..),
  Subscriber Priority (8),
  Group Order (8),
  Filter Type (i),
  [StartGroup (i),
  StartObject (i)],
  [EndGroup (i)],
  Number of Parameters (i),
  Subscribe Parameters (..) ...
}
```

Start

- Latest Object
- Latest Group
- Future Group

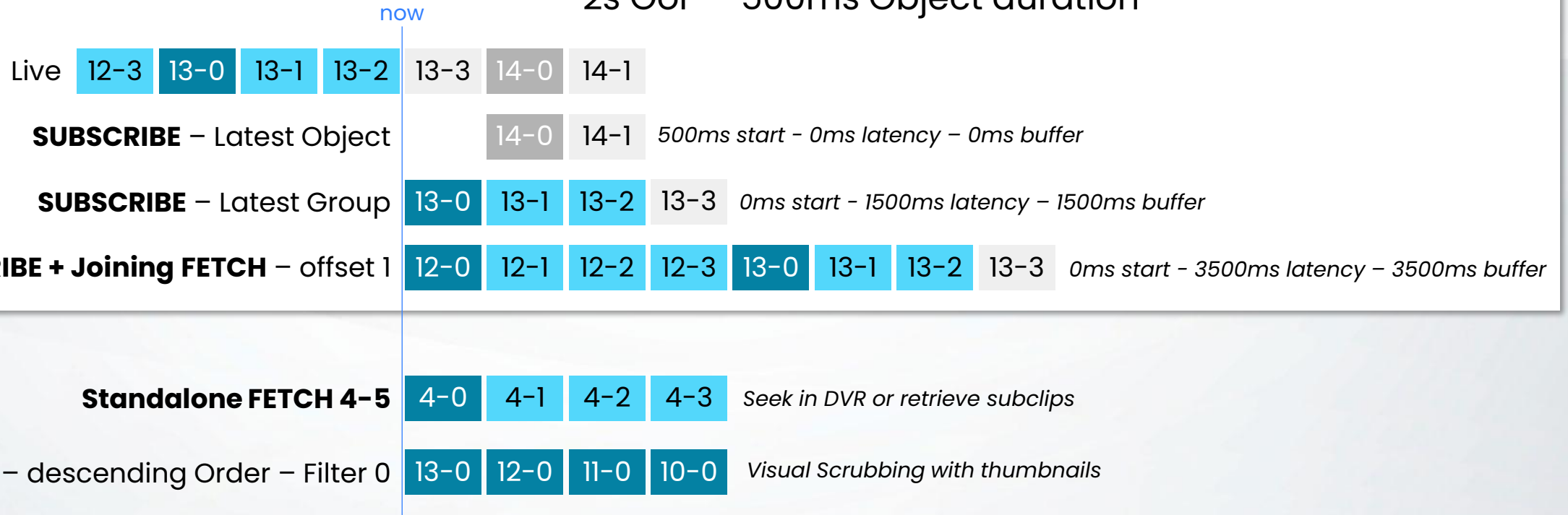
End

- Open range
- End Group

FETCH Use-Cases

2s GoP – 500ms Object duration

Courtesy of Will Law (Akamai)



VoD **interstitials** can be implemented by combining **SUBSCRIBE** and **FETCH**

Last Week's Interim Meeting

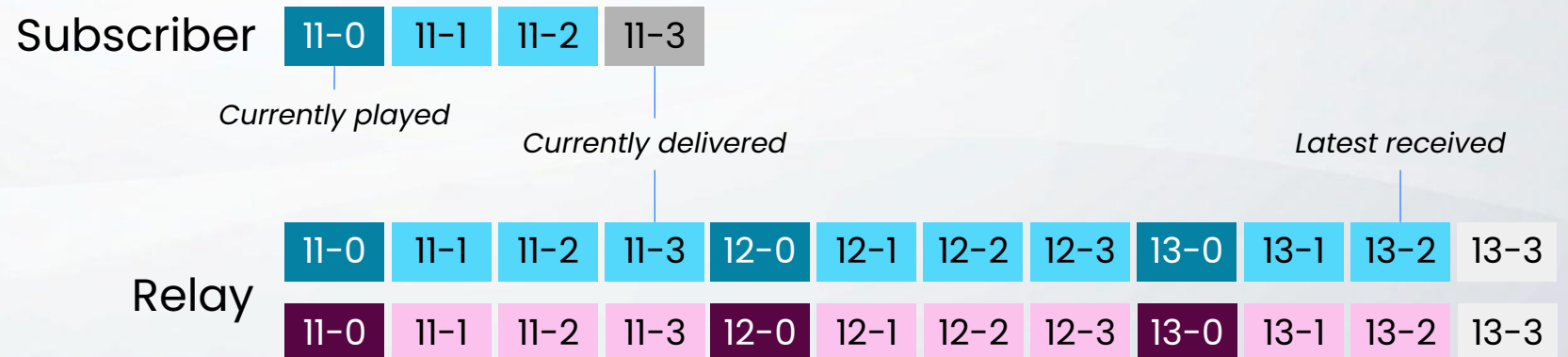
Extension headers

Authentication with Common Access Token (CAT)

- Token at Web Transport establishment
- Validation for SUBSCRIBE, FETCH, and ANNOUNCE

Switch: Client-side ABR with Relay assistance

1. SUBSCRIBE B low-priority
2. **New** Absolute Joining FETCH B 12
3. SUBSCRIBE_UPDATE A end 12
4. SUBSCRIBE_UPDATE B high-priority



Using MOQ with non-real-time media

Benefits of MOQ over HLS/DASH

- Scalable latency
 - from sub-second to 20+ seconds.
- Synchronize-able clients
 - A timing track can be added to any broadcast
- Unlimited parallel metadata flows
 - 3D data, scores, live chats/commentary
- Transport of media decoupled from its decoding.
 - Package I/P frames separately from B frames.
- Reduced thundering herd for live events
 - Far fewer requests than HAS.
- Well-suited to interstitial advertising
 - Can inherit interstitial ad inventory
- Well-suited to Multiview and XR applications
 - Head mounted, smart glasses, next-gen gaming
- Can work with non-frame-based codecs
 - Security cameras
- Bi-directional player communications
 - Collect player logs and analytics data over same network that distributes media

Headwinds

- HLS/DASH have a huge installed player base
- Resistance to change
 - New engineering required, new support, new players
- Economics
 - HAS operates at commodity prices
- HLS/DASH are good enough
 - MOQ needs to offer a compelling benefit to induce a switch

Prediction: we'll likely see MOQ distribution deployments in gaming/betting in about 2026 followed by live sports in 2027 and VOD in the years after that.

MoQ is 2025

Real-Time Streaming

- **Cisco** conferencing

Large-Scale Ingest

- **YouTube** Live

Contribution

- **Vindral** Rallye

Distribution

- Undisclosed projects

Delivery

- More likely 2026

SVTA Network & Transport MoQ Projects

MoQ Observability platform (analytics)

- Replace HTTP Transactions logs at Relay
- Generate CMCD reports at Subscriber
- Leverage MoQ to transport logs (Cisco proposal)
- Include *qlog* in logs (mixing layers?)

WARP

- Standardizing URL formats for WebTransport connections
- Carriage and use of CAT
- Support for WARP as a production media format
- Support for WARP-compliant media players



Synamedia
Connect a bigger audience