Media over QUIC (MoQ): Streaming Format vs Media Packaging, and Where LOC Fits

Prepared for: Personal Study Notes / Research Documentation

February 24, 2026

Abstract

This document clarifies a common source of confusion in modern real-time streaming systems: the difference between *streaming format* and *media packaging*, specifically in the context of *Media over QUIC (MoQ)*. It presents a clean layered model that separates codec, packaging/container, streaming format, and transport. It then positions *Low Overhead Container (LOC)* in this stack, explains its relationship to *CMAF*, and illustrates how object-based delivery in MoQ interacts with packaging choices. Concrete examples and research implications (especially for client-side adaptation) are included.

Contents

1 Motivation and High-Level View **3**
1.1 The Simplified Pipeline . . . 3

2 Core Definitions **3**
2.1 Media Packaging / Container . . . 3
2.2 Streaming Format . . . 4

3 MoQ Context: What MoQ Is (and Is Not) **5**
3.1 MoQ as a Streaming Format . . . 5
3.2 What MoQ Does Not Define . . . 5

4 Where CMAF and WebTransport Fit (as Context) **5**
4.1 CMAF (Packaging/Container Layer) . . . 5
4.2 WebTransport (Transport API / Application Interface Layer) . . . 5

5 The Canonical Layered Stack (MoQ View) **6**

6 Packaging vs Streaming Format (The Main Difference) **6**
6.1 One-Sentence Distinction . . . 6
6.2 A Concrete Example . . . 6

7 Where LOC Fits **7**
7.1 LOC is a Packaging/Container . . . 7
7.2 Why LOC Exists in QUIC-Native Streaming . . . 7
7.3 LOC vs CMAF (Conceptual Comparison) . . . 7

8 MoQ + LOC Synergy: Object Mapping Intuition **7**
8.1 Direct Mapping Idea 7

9 Practical Implications for Client-Side Adaptation **8**
9.1 Packaging-Level Decisions (LOC/CMAF Layer) 8
9.2 Streaming-Format Decisions (MoQ Layer) 8
9.3 Cross-Layer Reality 8

10 Glossary (Quick Reference) **8**

11 References (Optional, Add/Adjust as Needed) **9**

Recommended Resources **9**

1 Motivation and High-Level View

Modern low-latency streaming increasingly uses QUIC-based transports and object-oriented delivery models (e.g., publish–subscribe). In this ecosystem, terms like *streaming format*, *media packaging*, *container*, and *transport* are sometimes used interchangeably, even though they describe different layers.

This note provides a precise conceptual separation to support protocol-level research and implementation decisions (e.g., building with MoQ libraries and systems such as moq-rs/moqtail, or designing client-side adaptation strategies).

1.1 The Simplified Pipeline

A helpful first mental model is:

```
Raw media -> Encoding (codec) -> Packaging/Container -> Streaming Format ->
   Transport (Media Over QUIC)
```

Each stage answers a different question:

- **Codec:** How is media compressed?
- **Packaging/Container:** How is compressed media structured into deliverable units?
- **Streaming Format:** How are those units delivered, requested, prioritized, and adapted?
- **Transport:** How are bytes reliably/unreliably carried with multiplexing and congestion control?

2 Core Definitions

This section defines the two most confusing concepts: *media packaging* and *streaming format*.

2.1 Media Packaging / Container

Media packaging (often implemented via a **container format**) defines how encoded audio/video samples are organized into chunks/fragments/segments along with timing and track metadata so that it can be delivered efficiently over a network.

Packaging answers:

- How do we split compressed media into *network-friendly units*?
- How are *timing*, *tracks*, and *codec configuration* signaled?
- What is the *granularity* of deliverable units (e.g., 50ms chunks vs 1s segments)?

Packaging is <u>not</u> transport logic. Packaging does not define subscription models, retransmission policy, relay semantics, or congestion-responsive scheduling. Those belong to a streaming format and transport.

Example: Common Media Application Format (CMAF):

- Uses fragmented MP4 (fMP4)

- Defines: CMAF segments, CMAF chunks, Track fragments
- Enables low-latency chunked delivery

In MoQ context:

- CMAF chunks can map directly to MoQ objects
- Each object becomes independently deliverable over QUIC streams

2.2 Streaming Format

A **streaming format** defines how packaged media units are delivered over a transport protocol, including:

- Session model (publisher/subscriber vs client/server fetch)
- Delivery model (push/pull; live vs VOD)
- Unit selection and prioritization (e.g., audio first, newest frames first)
- Adaptation logic and buffer behavior
- Scalability semantics (e.g., relay distribution, fan-out)

Streaming format is transport-aware. It is the layer that decides *how to use* QUIC streams/datagrams (or other transport primitives) to meet latency/reliability goals.

Example 1: DASH (as a Streaming Format)

- Uses CMAF or TS packaging
- Delivers over HTTP
- Pull-based model
- Client fetches segments
- Manifest-driven adaptation

DASH is both **a packaging ecosystem and a streaming format**. But conceptually:

- CMAF = packaging
- DASH = streaming logic

Example 2: WebTransport

- Runs over HTTP/3
- Uses QUIC streams & datagrams
- Provides bidirectional streams
- Not media-specific
- General-purpose low-latency streaming

It is **a transport/streaming API but Not a media packaging format.**

3 MoQ Context: What MoQ Is (and Is Not)

Media over QUIC (MoQ) is best understood as a **streaming format** designed for QUIC.

3.1 MoQ as a Streaming Format

MoQ is commonly characterized by:

- **Object-based delivery** (media is carried as objects rather than monolithic byte streams)
- **Publish–subscribe model** (publishers produce tracks; subscribers subscribe to tracks)
- **Relay-friendly distribution** (designed for scalable fan-out)
- **QUIC-native design** (leveraging multiplexing and congestion control characteristics)

3.2 What MoQ Does Not Define

MoQ typically does **not** define:

- Codec choice (e.g., AV1, H.266, Opus)
- Exact container structure (e.g., MP4 box layout)
- Detailed sample formatting rules inside the container

Instead, MoQ transports *packaged media units* as **objects**.

4 Where CMAF and WebTransport Fit (as Context)

This section places two commonly referenced terms in the same layered framework.

4.1 CMAF (Packaging/Container Layer)

CMAF is a packaging/container approach built on fragmented MP4 (ISO BMFF). It defines:

- CMAF segments and chunks
- Track fragmentation and timing metadata conventions
- Interoperable structures used widely in DASH/HLS ecosystems

4.2 WebTransport (Transport API / Application Interface Layer)

WebTransport is a web-facing transport API (often over HTTP/3/QUIC) providing streams and datagrams to applications. It is:

- QUIC/HTTP3-oriented transport interface
- Not media-specific by itself
- Often used as a building block for real-time apps and custom streaming systems

Key distinction:

- CMAF: **packaging/container**
- WebTransport: **transport API / transport usage interface**

5 The Canonical Layered Stack (MoQ View)

A clean stack that avoids ambiguity:

Layer	Example Responsibilities and Examples
Codec	Compression (AV1, H.266/VVC, Opus)
Packaging / Container	Structure + timing metadata (CMAF, LOC)
Streaming Format	Delivery model, subscribe/publish, prioritization (MoQ)
Transport	Multiplexing, congestion control, reliability primitives (QUIC)

6 Packaging vs Streaming Format (The Main Difference)

6.1 One-Sentence Distinction

- **Media Packaging:** how compressed media is *structured* into chunks/fragments/segments with metadata.
- **Streaming Format:** how those units are *delivered, requested, prioritized, adapted* over a transport.

6.2 A Concrete Example

Suppose an encoder outputs AV1 video, and a packager produces deliverable units in 200ms chunks.

Packaging stage (unit creation):

```
Segment 1:
  Chunk 1: 0-200ms
  Chunk 2: 200-400ms
  Chunk 3: 400-600ms
```

Streaming format stage (delivery behavior): With MoQ, each chunk may map to an object, and the system decides:

- which objects to prioritize (audio first, newest video first)
- whether to drop late/old objects under congestion
- how subscribers request objects/tracks (latest-only vs full history)

Same packaging, different streaming behavior. The exact same 200ms packaged chunks can be delivered via different strategies depending on the streaming format logic.

7 Where LOC Fits

7.1 LOC is a Packaging/Container

Low Overhead Container (LOC) fits at the **packaging/container layer**, alongside CMAF.

```
Application Layer
|  Streaming Format     -> MoQ
|  Packaging/Container  -> LOC / CMAF
|  Codec                -> AV1 / H.266 / Opus
|Transport Layer        -> QUIC
```

7.2 Why LOC Exists in QUIC-Native Streaming

CMAF was designed in a world strongly shaped by HTTP-based streaming ecosystems. MoQ, by contrast, is QUIC-native and object-based. LOC aims to provide a container that is:

- lightweight (lower overhead)
- easy to parse (faster processing)
- friendly to object-based delivery models
- better aligned with ultra-low-latency streaming goals

7.3 LOC vs CMAF (Conceptual Comparison)

Aspect	CMAF	LOC
Category	Packaging/container	Packaging/container
Historical fit	HTTP streaming ecosystems	QUIC-native/object streaming
Overhead	Typically higher	Designed to be lower
Parsing complexity	Often heavier (BMFF boxes)	Designed for fast/low overhead parsing
MoQ alignment	Works, but can be indirect	Designed to align well with object mapping

8 MoQ + LOC Synergy: Object Mapping Intuition

MoQ delivers **objects**. A major practical consideration is how a packaging choice maps cleanly into those objects.

8.1 Direct Mapping Idea

A common conceptual mapping is:

```
LOC unit (frame/chunk)  ->  MoQ object  ->  QUIC stream/datagram usage
```

With some packaging approaches, the system may require extra steps:

```
Packaged (fMP4) fragment -> Parse -> Extract samples -> Re-wrap -> MoQ object
```

The more transformations required, the more potential overhead/latency (especially at relays or clients) and the more complicated scheduling becomes.

9 Practical Implications for Client-Side Adaptation

Because MoQ adaptation decisions frequently operate at the **object level**, packaging granularity matters.

9.1 Packaging-Level Decisions (LOC/CMAF Layer)

Examples:

- Chunk duration (e.g., 20ms vs 200ms vs 1s)
- Keyframe alignment and IDR placement
- Track structure (multiple qualities, layers, tiles)
- Metadata frequency and format

9.2 Streaming-Format Decisions (MoQ Layer)

Examples:

- Which track(s) to subscribe to (quality switching)
- Object prioritization strategy (audio, base layer first, newest-first)
- Drop policy under congestion (skip old objects, skip enhancement layers)
- Relay-aware behaviors (fan-out and cache policy for objects)

9.3 Cross-Layer Reality

Even though packaging and streaming format are different layers, they interact:

- Smaller packaging units can improve adaptation responsiveness but increase overhead.
- Larger units reduce overhead but can worsen latency and responsiveness.
- A container that aligns with MoQ objects can simplify implementation and reduce processing delay.

10 Glossary (Quick Reference)

Codec Compression algorithm that turns raw media into encoded samples (e.g., AV1, H.266/VVC).

Packaging/Container Structures encoded samples with timing/track metadata into units suitable for delivery (e.g., CMAF, LOC).

Streaming Format Defines how packaged units are delivered and adapted over a transport (e.g., MoQ).

Transport Moves bytes with multiplexing and congestion control (e.g., QUIC).

Object (MoQ context) A deliverable unit carried through MoQ, often mapped from packaging units (chunks/frames).

11 References (Optional, Add/Adjust as Needed)

- MoQ (Media over QUIC): IETF drafts and related working group materials.
- CMAF (Common Media Application Format): ISO BMFF / CMAF specifications and industry notes.
- WebTransport: W3C specifications and browser platform documentation.
- LOC (Low Overhead Container): IETF drafts and related container-format discussions for low-latency delivery.

Recommended Resources

- IETF MoQ Working Group: https://datatracker.ietf.org/wg/moq/
- MoQ Transport Draft: https://datatracker.ietf.org/doc/draft-ietf-moq-transport/
- CMAF (ISO/IEC 23000-19:2024): https://www.iso.org/standard/85623.html
- LOC (Low Overhead Container) IETF drafts: https://www.ietf.org/archive/id/draft-ietf-moq-loc-01.html
- WebTransport|W3C specifications: https://www.w3.org/TR/media-timed-events/
- https://techdocs.akamai.com/msl/docs/cmaf