Media Over QUIC
Internet-Draft
Intended status: Informational
Expires: 9 October 2026

E. Herz
Vivoh, Inc.

7 April 2026

NMSF - Neural Video Codec Packaging for MOQT Streaming Format
draft-herz-moq-nmsf-01

Abstract

This document updates the MOQT Streaming Format (MSF) by defining a new optional feature for the streaming format. It specifies the syntax and semantics for adding Neural Video Codec (NVC) packaged media to MSF. NVC codecs use learned neural network transforms for video compression, and their bitstreams require a distinct packaging model from traditional block-based codecs. NMSF maps neural keyframes (Intra) and delta frames (Inter) onto MoQ Groups and Objects, and introduces a multi-track model that separates hyperprior side information from latent bitstreams for priority-aware delivery. This enables real-time neural video streaming over any standard MoQ relay.

About This Document

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

Source for this draft and an issue tracker can be found at https://github.com/erikherz/nmsf.

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 9 October 2026.

Table of Contents

1. Introduction

Neural Video Codecs (NVCs) represent a new class of video compression that uses learned neural network transforms instead of block-based motion estimation and discrete cosine transforms. NVCs such as DCVC-RT [DCVC-RT], SSF, FVC, and RLVC produce compressed bitstreams that differ fundamentally from traditional codecs:

* No container format. NVC bitstreams consist of entropy-coded latent tensors, not fMP4 boxes, LOC containers, or NAL units.

* Two-layer compression. NVCs using hyperprior-based entropy coding (such as those built on [CompressAI]) produce two distinct bitstreams per frame: a small hyperprior containing statistical side information, and a larger latent tensor containing the compressed frame data. The latent cannot be decoded without the hyperprior.

* Stateful decoding. The decoder maintains a learned context buffer (analogous to a decoded picture buffer) that must be initialized from a full neural keyframe before delta frames can be decoded.

* Variable-rate representations. Compressed frame sizes vary significantly based on scene complexity, as the codec allocates bits adaptively in a learned feature space.

The existing MSF [I-D.ietf-moq-msf] packaging types -- LOC and the timeline types -- do not accommodate these bitstreams. The CMSF [I-D.ietf-moq-cmsf] extension adds CMAF packaging for traditional block-based codecs. Neither is suitable for NVC data, which has no container structure and requires a different model for keyframe semantics and decoder state management.

This document defines NMSF, an MSF extension that adds NVC packaging. NMSF follows the same extension pattern established by CMSF: it registers a new "packaging" value, defines the Object payload format, and specifies Group-level requirements for decoder random access. Additionally, NMSF introduces a multi-track model that maps the hyperprior and latent bitstreams to separate MoQ tracks, enabling priority-based relay delivery under congestion.

```
MSF (base)
 +-- LOC packaging          (native, MSF)
 +-- Media/Event timelines  (native, MSF)
 +-- CMSF extension         (adds "cmaf" packaging, CMSF)
 +-- NMSF extension         (adds "nvc" packaging, this document)
```

A single MoQ Broadcast MAY contain tracks using any combination of packaging types. For example, an NVC video track may coexist with a LOC or CMAF audio track in the same catalog.

2. MSF Extension

All of the specifications, requirements, and terminology defined in [I-D.ietf-moq-msf] apply to implementations of this extension unless explicitly noted otherwise in this document.

3. NVC Packaging

3.1. Track Model

NVC packaging uses two MoQ tracks per video stream to carry the two layers of the neural codec’s compressed output:

Hyperprior track: Carries the entropy-coded hyperprior (side information) for each frame. The hyperprior describes the statistical distribution of the latent tensor and is required by the entropy decoder before the latent can be decompressed. This track is small (typically 5-15% of total bitstream) and SHOULD be assigned higher delivery priority than the latent track.

Latent track: Carries the entropy-coded latent tensor (the primary compressed frame data) for each frame. The latent track is larger and depends on the corresponding hyperprior Object having been received and decoded first.

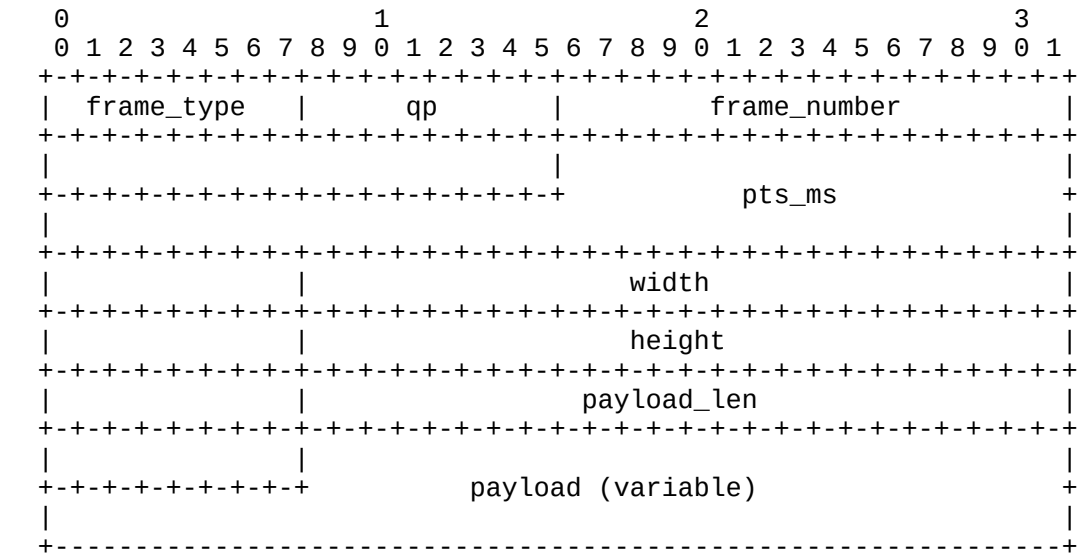
Both tracks MUST use "nvc" packaging and belong to the same MoQ Broadcast. The tracks are linked via a "depends" field in the catalog (Section 3.8). Objects in the two tracks are correlated by Group sequence number and Object index: Object N in Group G of the hyperprior track corresponds to Object N in Group G of the latent track.

This two-track model reflects the inherent two-layer structure of hyperprior-based neural codecs. It enables MoQ relays to prioritize the small hyperprior track under congestion, ensuring that the decoder can begin latent decompression as soon as latent data arrives. The relay requires no awareness of the NVC payload format -- standard MoQ priority mechanisms are sufficient.

Implementations MAY use a single combined track instead of two separate tracks. In this case, the payload sub-format defined in Section 3.6 carries both bitstreams within each Object. The single-track mode is simpler but loses the ability to prioritize hyperprior delivery independently.

3.2. Object Wire Format

Each MoQ Object payload for a track with "nvc" packaging consists of a fixed 26-byte header followed by a variable-length compressed bitstream:



frame_type: 1 byte. The type of neural video frame contained in this Object. See Section 3.3.

qp: 1 byte, unsigned 8-bit integer. The quality parameter index used for encoding this frame. Identifies the rate-distortion operating point from the codec's trained quality levels. The decoder MUST use the same QP index to select matching quantization step sizes. Values 0-63 are defined; values 64-255 are reserved.

frame_number: 4 bytes, unsigned 32-bit integer, big-endian. The absolute sequence number of this frame within the stream, starting from zero.

pts_ms: 8 bytes, unsigned 64-bit integer, big-endian. The capture-side wallclock timestamp in milliseconds since the Unix epoch (1970-01-01T00:00:00Z). This timestamp is set by the publisher at the time of frame capture and is carried through the relay unmodified. Subscribers MAY use the difference between pts_ms and their local wallclock to estimate end-to-end latency. A value of zero indicates that no timestamp is available.

width: 4 bytes, unsigned 32-bit integer, big-endian. The frame

width in pixels.

height: 4 bytes, unsigned 32-bit integer, big-endian. The frame height in pixels.

payload_len: 4 bytes, unsigned 32-bit integer, big-endian. The byte length of the payload field that follows.

payload: Variable length. The NVC compressed bitstream for this frame. In two-track mode, the hyperprior track carries the hyperprior bitstream and the latent track carries the latent bitstream. In single-track mode, this field carries the combined payload defined in Section 3.6.

3.3. Frame Types

The frame_type field identifies the role of the frame in the neural codec's prediction structure:

Value	Type	Description
0x00	Intra	Neural keyframe. Decoder initializes its context buffer from this frame. MUST be the first Object in a Group.
0x01	Inter	Neural delta frame. Decoder uses context buffer from previous reconstructed frame.
0x02-FF	Rsvd	Reserved for future use.

Figure 1

Intra frames are analogous to SAP Type 1 access points in CMAF. They enable random access by fully initializing the decoder's context buffer without dependence on any prior frame.

Inter frames are analogous to non-SAP frames (P-frames). They depend on the decoder's context buffer, which contains the reconstructed output of the immediately preceding frame.

Unlike traditional codecs with multi-frame reference picture buffers, current NVCs maintain a single context buffer containing learned features from the previous reconstructed frame. The reserved range (0x02-0xFF) accommodates future NVCs that may introduce bidirectional prediction, hierarchical quality layers, or multi-reference architectures.

3.4. Object Packaging

The payload of each Object is subject to the following requirements:

* MUST contain exactly one NVC frame (or frame component, in two-track mode) in the wire format defined in Section 3.2.

* MUST NOT span multiple frames. Each frame is carried in a separate Object per track.

* Objects within a Group MUST be sequentially ordered by frame_number. Out-of-order processing causes encoder-decoder context buffer divergence.

* In two-track mode, the hyperprior Object and the corresponding latent Object for the same frame MUST have identical frame_type, qp, frame_number, pts_ms, width, and height header values.

3.5. Group Packaging

Each MOQT Group:

* MUST begin with an Object containing an Intra frame (frame_type = 0x00).

* MUST contain one contiguous Group of Pictures (GOP): one Intra frame followed by zero or more Inter frames.

* The Group boundary aligns with the publisher’s neural GOP boundary. Typical GOP sizes are 30-120 frames (1-4 seconds at 30 fps).

* In two-track mode, both the hyperprior and latent tracks MUST use the same Group sequence numbers and contain the same number of Objects per Group.

This structure ensures that a subscriber joining mid-stream or recovering from loss can begin decoding from the next Group boundary. The Intra frame at the start of each Group fully initializes the decoder context buffer, enabling immediate playback without waiting for a future keyframe.

3.6. Payload Format

In two-track mode, each track's payload contains a single bitstream component:

* Hyperprior track payload: the entropy-coded hyperprior tensor z, preceded by tensor shape metadata.

* Latent track payload: the entropy-coded latent tensor y, preceded by tensor shape metadata.

Each payload component uses this sub-format:

Offset	Size	Field	
0	4 bytes	channels	(uint32, big-endian)
4	4 bytes	height	(uint32, big-endian)
8	4 bytes	width	(uint32, big-endian)
12	4 bytes	data_len	(uint32, big-endian)
16	N bytes	data	(entropy-coded tensor)

The channels, height, and width fields describe the spatial dimensions of the tensor prior to entropy coding. These are required by the decoder to allocate output buffers and configure the entropy decoder.

In single-track mode, the payload carries both components concatenated: the hyperprior component followed immediately by the latent component, using the same sub-format for each.

3.7. Catalog: NVC Packaging Type

This specification extends the allowed packaging values defined in [I-D.ietf-moq-msf] to include one new entry:

Name	Value	Reference
NVC	nvc	This RFC

Figure 2

Every Track entry in an MSF catalog carrying NVC-packaged media data MUST declare a "packaging" type value of "nvc".

3.8. Catalog: NVC Track Fields

This specification adds the following track-level catalog fields for tracks with "nvc" packaging:

Field	JSON Type	Required	Definition
codec	String	Yes	NVC codec identifier. See Section 5.
colorspace	String	Yes	Input colorspace (e.g., "ycbcr-bt709").
gopSize	Number	Yes	Number of frames per Group (GOP size).
nvcRole	String	Yes (2T)	Track role in two-track mode: "hyperprior" or "latent". Omitted in single-track mode.
depends	String	Cond.	Name of the track this track depends on. REQUIRED for latent tracks (nvcRole="latent").
priority	Number	No	Delivery priority hint. Lower values = higher priority. Hyperprior tracks SHOULD use a lower value than latent tracks.
nvc	Object	No	Codec-specific metadata. See Section 3.9.

Figure 3

The standard MSF track fields "name", "packaging", "isLive", "width", "height", and "framerate" retain their definitions from [I-D.ietf-moq-msf] and are REQUIRED for NVC tracks.

3.9. Catalog: NVC Metadata Object

The optional "nvc" object within a track catalog entry carries codec-specific metadata that a subscriber may need to configure its decoder:

Field	JSON Type	Description
modelVersion	String	Model checkpoint version identifier.
entropyFormat	String	Entropy coding format (e.g., "rans64", "arithmetic").
latentChannels	Number	Channel count of the latent tensor.
hyperChannels	Number	Channel count of the hyperprior tensor.
quantParams	Object	Codec-specific quantization parameters.

Figure 4

Subscribers that do not recognize the "codec" value or cannot satisfy the metadata requirements SHOULD NOT subscribe to the track.

4. Decoder Requirements

This section specifies the behavior required of an NMSF decoder. These requirements ensure that encoder and decoder context buffers remain synchronized, preventing visual artifacts caused by state drift.

4.1. Context Buffer Management

The decoder MUST maintain a context buffer containing the reconstructed output of the most recently decoded frame. This buffer is used as input to the synthesis transform when decoding Inter frames.

The context buffer is uninitialized when the decoder starts. The decoder MUST NOT attempt to decode Inter frames until it has successfully decoded an Intra frame.

After decoding each frame (Intra or Inter), the decoder MUST replace its context buffer with the newly reconstructed output. Failure to update the context buffer causes progressive drift between encoder and decoder state, manifesting as visual artifacts commonly described as "ghosting" or "smearing."

4.2. Two-Track Decode Sequence

When operating in two-track mode, the decoder MUST process frames in the following order for each frame N in Group G:

1. Receive Object N from the hyperprior track (Group G).
2. Decode the hyperprior to obtain CDF parameters for the entropy decoder.
3. Receive Object N from the latent track (Group G).
4. Entropy-decode the latent tensor using the CDF parameters from step 2.
5. Run the synthesis transform (Intra or Inter path) to produce the reconstructed frame.
6. Update the context buffer with the reconstructed frame.

The decoder MAY begin receiving the latent Object before the hyperprior decode is complete, but MUST NOT begin entropy decoding the latent until the hyperprior CDF parameters are available.

4.3. Intra Frame Handling

When the decoder receives an Intra frame (frame_type = 0x00):

1. Discard any existing context buffer.
2. Decode the frame using only the payload data (no context).
3. Store the reconstructed output as the new context buffer.
4. Output the reconstructed frame for display.

4.4. Inter Frame Handling

When the decoder receives an Inter frame (frame_type = 0x01):

1. Verify that a context buffer exists (i.e., an Intra frame has been previously decoded). If not, discard the frame.
2. Decode the frame using the payload AND the current context buffer.
3. Replace the context buffer with the newly reconstructed output.

4. Output the reconstructed frame for display.

4.5. Stream Join and Recovery

When a subscriber joins a stream mid-session or recovers from packet loss:

1. Wait for the start of the next MoQ Group.

2. The first Object in the Group is an Intra frame.

3. Decode the Intra frame to initialize the context buffer.

4. Continue decoding subsequent Inter frames normally.

Objects received before the first Intra frame MUST be discarded.

In two-track mode, the subscriber MUST subscribe to both the hyperprior and latent tracks. Subscribing to the latent track alone is insufficient, as the latent cannot be decoded without the hyperprior CDF parameters.

4.6. Encoder Context Synchronization

The encoder MUST maintain its own context buffer that mirrors the decoder’s state. After encoding each frame, the encoder MUST run the synthesis (decoding) transform on the quantized latent representation to produce a reconstructed frame, and use that reconstructed frame as the context for encoding the next frame.

This "encode-decode loop" ensures that the encoder’s context buffer contains exactly the same data that a decoder would produce from the transmitted bitstream, preventing encoder-decoder state drift.

5. Codec Registration

The "codec" field in the catalog identifies which NVC is used:

Value	Full Name	Reference
dcvc-rt	Deep Contextual Video Compression - Real Time	[DCVC-RT]
dcvc-fm	DCVC Feature Modulation	
dcvc-dc	DCVC Data Conditions	
ssf	Scale-Space Flow	
fvc	Feature-space Video Coding	
rlvc	Recurrent Learned Video Compression	
elfvc	Efficient Learned Flexible Video Coding	

Figure 5

New NVC codecs are compatible with NMSF packaging if they produce distinct Intra and Inter frame types and use a single sequential context buffer. Registration requires choosing a unique codec identifier string and documenting the payload sub-format (Section 3.6). No changes to the wire format header (Section 3.2) are required for new codecs.

6. Catalog Examples

The following section provides non-normative JSON examples of catalogs compliant with this draft.

6.1. Two-track NVC video with LOC audio

This example shows a catalog for a live broadcast with DCVC-RT neural video using the two-track model (hyperprior + latent) and one Opus audio track using MSF’s native LOC packaging.

```
{
  "version": 1,
  "streamingFormat": 1,
  "streamingFormatVersion": "0.2",
  "tracks": [
    {
      "name": "video-hyper",
      "packaging": "nvc",
      "isLive": true,
```

```
        "codec": "dcvc-rt",
        "nvcRole": "hyperprior",
        "priority": 1,
        "width": 1280,
        "height": 720,
        "framerate": 30,
        "colorspace": "ycbcr-bt709",
        "gopSize": 60,
        "nvc": {
          "modelVersion": "cvpr2025",
          "entropyFormat": "rans64",
          "hyperChannels": 128
        }
      },
      {
        "name": "video-latent",
        "packaging": "nvc",
        "isLive": true,
        "codec": "dcvc-rt",
        "nvcRole": "latent",
        "depends": "video-hyper",
        "priority": 2,
        "width": 1280,
        "height": 720,
        "framerate": 30,
        "colorspace": "ycbcr-bt709",
        "gopSize": 60,
        "nvc": {
          "modelVersion": "cvpr2025",
          "entropyFormat": "rans64",
          "latentChannels": 128
        }
      },
      {
        "name": "audio",
        "packaging": "loc",
        "isLive": true,
        "codec": "opus",
        "role": "audio",
        "samplerate": 48000,
        "channelConfig": "2",
        "bitrate": 128000
      }
    ]
  }
```

6.2. Single-track NVC video (compact mode)

This example shows a catalog using single-track mode, where the hyperprior and latent are combined in a single payload. This mode is simpler but does not support independent priority control.

```
{
  "version": 1,
  "tracks": [
    {
      "name": "video",
      "packaging": "nvc",
      "isLive": true,
      "codec": "dcvc-rt",
      "width": 1280,
      "height": 720,
      "framerate": 30,
      "colorspace": "ycbcr-bt709",
      "gopSize": 60,
      "nvc": {
        "modelVersion": "cvpr2025",
        "entropyFormat": "rans64",
        "latentChannels": 128,
        "hyperChannels": 128
      }
    },
    {
      "name": "audio",
      "packaging": "cmaf",
      "isLive": true,
      "initData": "AAAAIGZ0eXBpc281AAA...AAAAAAAAAA",
      "codec": "mp4a.40.2",
      "role": "audio",
      "samplerate": 48000,
      "channelConfig": "2",
      "bitrate": 128000
    }
  ]
}
```

7. 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.

8. Security Considerations

NMSF relies on the security properties of MoQ Transport [I-D.ietf-moq-transport], which provides confidentiality and integrity via QUIC's TLS 1.3 encryption. NMSF does not add its own integrity or authentication mechanisms.

The "payload_len" field permits payloads up to 4 GiB. Decoders SHOULD enforce a maximum payload size appropriate for their deployment environment (e.g., 100 MiB for 4K video) and reject Objects exceeding that limit to mitigate resource exhaustion.

A malicious publisher could craft Intra frames that cause the decoder's context buffer to enter a state producing misleading visual output on subsequent Inter frames. This is analogous to reference picture manipulation in traditional codecs and is mitigated by the same trust model: subscribers SHOULD only connect to authenticated and authorized publishers.

Neural video codec model weights are typically large (tens to hundreds of megabytes) and are NOT transmitted via MoQ. Both publisher and subscriber must have compatible model weights pre-installed. The "nvc" catalog metadata (Section 3.9) enables version negotiation, but the secure distribution of model weights is outside the scope of this document.

The pts_ms timestamp reveals the publisher's wallclock time, which may be a privacy concern in some deployments. Publishers MAY set pts_ms to zero to suppress timestamp information.

9. IANA Considerations

This document requests registration of a new packaging type value "nvc" in the MSF packaging registry defined by [I-D.ietf-moq-msf].

This document requests creation of an "NVC Codec Identifiers" registry with the initial values defined in Figure 5. New entries require Specification Required registration policy.

10. References

10.1. Normative References

[I-D.ietf-moq-cmsf]
Law, W., "CMSF- a CMAF compliant implementation of MOQT Streaming Format", Work in Progress, Internet-Draft, draft-ietf-moq-cmsf-00, 1 December 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-moq-cmsf-00>.

[I-D.ietf-moq-msf]
Law, W., "MOQT Streaming Format", Work in Progress, Internet-Draft, draft-ietf-moq-msf-00, 19 January 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-moq-msf-00>.

[I-D.ietf-moq-transport]
Nandakumar, S., Vasiliev, V., Swett, I., and A. Frindell, "Media over QUIC Transport", Work in Progress, Internet-Draft, draft-ietf-moq-transport-17, 2 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-moq-transport-17>.

[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/info/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/info/rfc8174>.

10.2. Informative References

[BT.709] ITU-R, "Parameter values for the HDTV standards for production and international programme exchange", Recommendation BT.709-6, June 2015.

[CompressAI]
InterDigital, "CompressAI: A PyTorch Library and Evaluation Platform for End-to-end Compression Research", <https://github.com/InterDigitalInc/CompressAI>.

[DCVC-RT] Microsoft Research, "Towards Practical Real-Time Neural Video Compression", CVPR 2025, 2025, <https://arxiv.org/abs/2502.20762>.

Acknowledgments

The author would like to thank Will Law for the MSF and CMSF specifications which established the extension pattern that NMSF follows, and the MoQ working group for the transport protocol that makes this work possible.

Changes from draft-herz-moq-nmsf-00

* Added two-track model: separate hyperprior and latent MoQ tracks with dependency and priority signaling (Section 3.1).

* Added per-frame PTS timestamp (pts_ms) to the wire format header for end-to-end latency measurement (Section 3.2).

* Added per-frame quality parameter (qp) to the wire format header for rate-distortion signaling (Section 3.2).

* Wire format header expanded from 17 bytes to 26 bytes.

* Added nvcRole, depends, and priority catalog fields for two-track mode (Section 3.8).

* Added two-track decode sequence to decoder requirements (Section 4.2).

* Retained single-track mode as a simpler alternative.

* Added privacy consideration for pts_ms timestamp (Section 8).

* Normalized payload sub-format with explicit tensor shape metadata preceding each bitstream component (Section 3.6).

Author’s Address

Erik Herz
Vivoh, Inc.
Email: erik@vivoh.com