ATSC 3.0 with MMT Mobility

10.13.2022

jjustman@ngbp.org

Rev I

ATSC 3.0 with MMT Mobility

This presentation provides real-world research and observations in evaluating ATSC 3.0 mobility.

Limitations of the ROUTE/DASH transport and packaging format prevent suitable experiences for live linear content on mobility device reception.

ATSC 3.0 with MMT Mobility

Convergence of broadcast and broadband services by leveraging IP Multicast is an incredibly powerful aspect of the ATSC 3.0 ecosystem.

But why did the TG3 adopt two different media streaming transport protocols – ROUTE/DASH and MMT (MPEG Media Transport) – essentially both designed to deliver real-time content to receiver devices?

A/331: 2022-09 - 4. System Overview

“Two methods of broadcast Service delivery are specified in this Standard.

The method depicted on the left side of Figure 4.1 is based on MPEG Media Transport (MMT), ISO/IEC 23008-1 [38] and uses MMT protocol (MMTP) to deliver Media Processing Units (MPU).

The method shown in the center is based on the DASH-IF [12] profile, which is based on MPEG DASH [58]. It uses Real-time Object delivery over Unidirectional Transport (ROUTE) protocol to deliver DASH Segments…”

https://www.atsc.org/wp-content/uploads/2022/09/A331-2022-09-Signaling-Delivery-Sync-FEC.pdf

Standards and Recommend Practices

TG3 approaches the standards development process from the component level of the transmission airchain up to the point of broadcast emission.

CTA provides a publicly available Recommended Practice for ATSC 3.0 Television Sets, System Issues (CTA-CEB32.1), Dec 2019.

TG3 produces recommended practices for broadcasters and vendors for clarity, and provides technical - not business nor viewer-based best practices - as those extend past the “broadcast emission”

Confusion? ROUTE-DASH vs. MMT?

The A/331 specification could have been clarified as follows (in bold):

"…The method depicted on the left side of Figure 4.1 is based on MPEG Media Transport (MMT), ISO/IEC 23008-1 [38] and uses MMT protocol (MMTP) to deliver Media Processing Units (MPU). ***MMT shall be used for robust live linear content transmission, including hybrid or scalable emission delivery models.***

The method shown in the center is based on the DASH-IF [12] profile, which is based on MPEG DASH [58]. It uses Real-time Object delivery over Unidirectional Transport (ROUTE) protocol to deliver DASH Segments. ***ROUTE-DASH shall be used for pre-positioned (VOD), carousel, or NRT content delivery emissions.***

Quick MMT Overview - ISO23008-1

Based on ISOBMFF - ISO14496-15

Two “data types” adopted in ATSC3.0:

- MPU - “Media aware fragment of the MPU”
  (MPU - Media processing unit)
- Signalling Message - contain media essence information (packet_id), timing, and other stream properties descriptors

Quick MMT Overview - ISO23008-1

MMTP Bitstream Representation

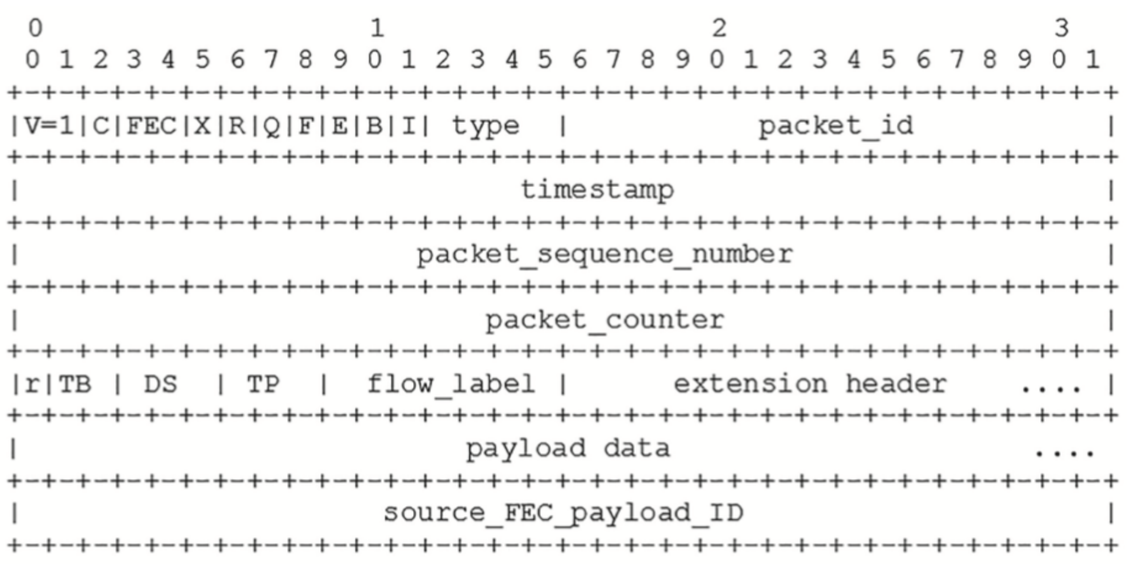
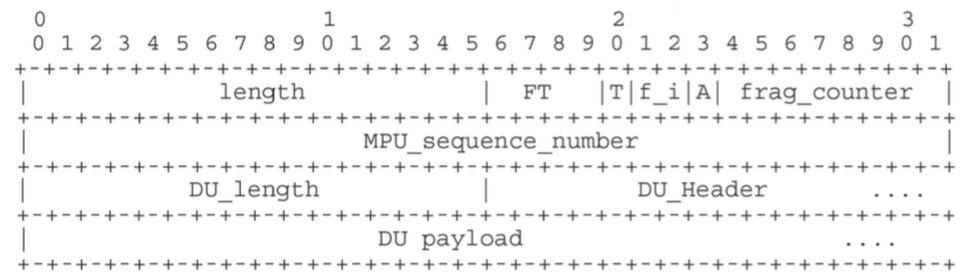


Figure 9 — MMTP packet header, payload and footer (V = 1)



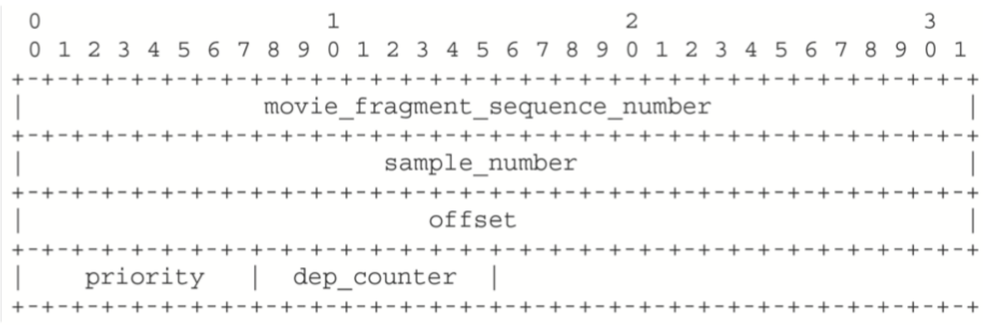


Figure 12 — DU header for timed-media MFU

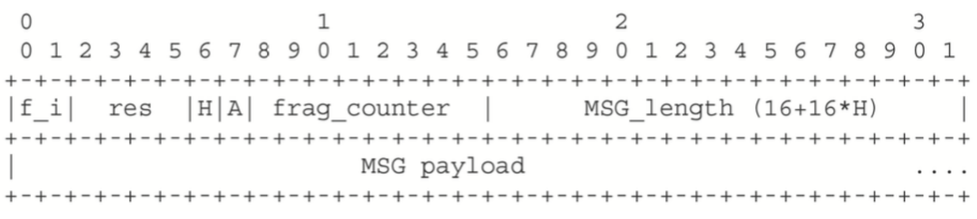


Figure 15 — MMTP payload header for signalling message mode

ATSC 3.0 with MMT Mobility

MMT Features not described in ISO23008-1, A/331, or A/351 RP:

1.) Robust Depacketization:

Missing or incomplete MFUs can be easily identified, and not dependent on the TRUN (track run box)

2.) Media-fragmentation aware transport

MFUs can be easily re-constructed, and any missing samples can be skipped over when pushing to decoder buffer, relying on decoder error concealment (e.g. mpeg2 macro blocking).

3.) Sparse media tracks across varying physical layer pipes

Ultra-robust audio emission, scalable HEVC (540/1080p), leverage LDM, even if individual tracks are missing samples.

MMT - Robust Depacketization - out-of-order mode

Robust Depacketization via out-of-order mode enables processing of each media sample via MFU basis, instead of the MPU level.

Think:

MFU: Media Fragmentation Unit - a single video, audio, or cc sample, e.g. 1 frame of essence.

MPU: Media Processing Unit - a group of pictures (v/a/cc), e.g. 1 or 2 seconds of media essence.

MDAT: Movie data - all MFU's concatenated together. position in box determined via offset (e.g. length) via either **MMTHSample** hint or **ISOBMFF trun** box.

ISOBMFF moof (movie fragment) and trun box

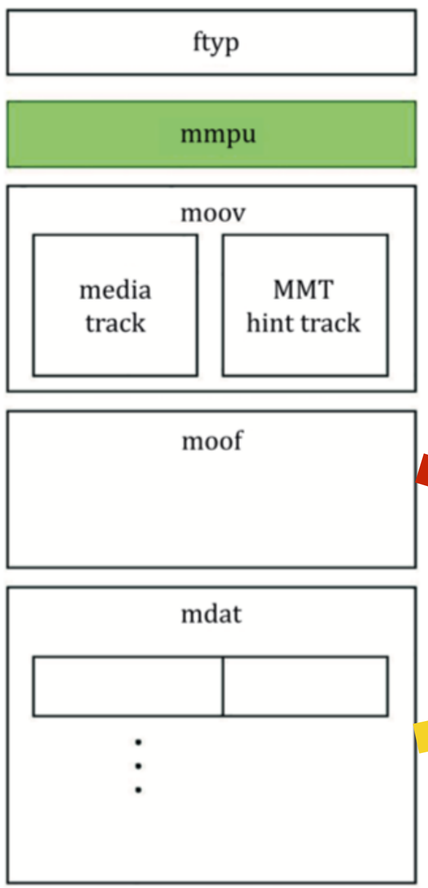
ISO14496-15 trun Box: Contained in the “moof” box

8.8.8.2 Syntax

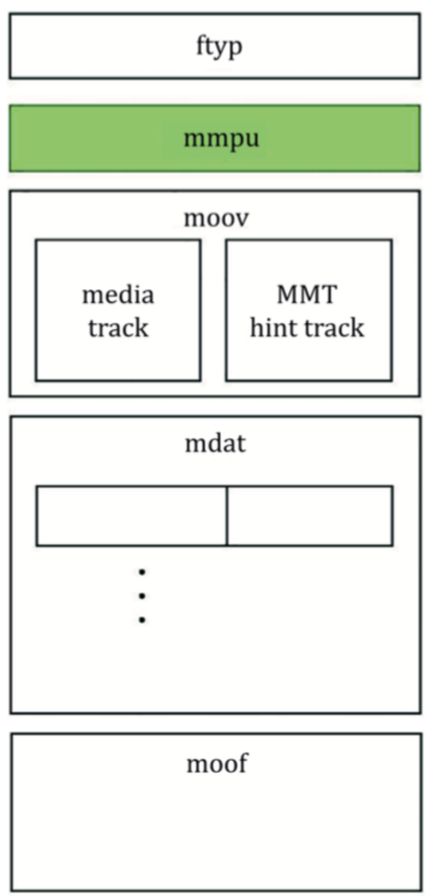
```
aligned(8) class TrackRunBox
          extends FullBox('trun', version, tr_flags) {
   unsigned int(32)  sample_count;
   // the following are optional fields
   signed int(32) data_offset;
   unsigned int(32)  first_sample_flags;
   // all fields in the following array are optional
   {
      unsigned int(32)  sample_duration;
      unsigned int(32)  sample_size;
      unsigned int(32)  sample_flags
      if (version == 0)
         { unsigned int(32)    sample_composition_time_offset; }
      else
         { signed int(32)      sample_composition_time_offset; }
   }[ sample_count ]
}
```

MMT - Robust Depacketization - Out of order mode

ISO23008-1:
“informative” reference
“in-order” mode



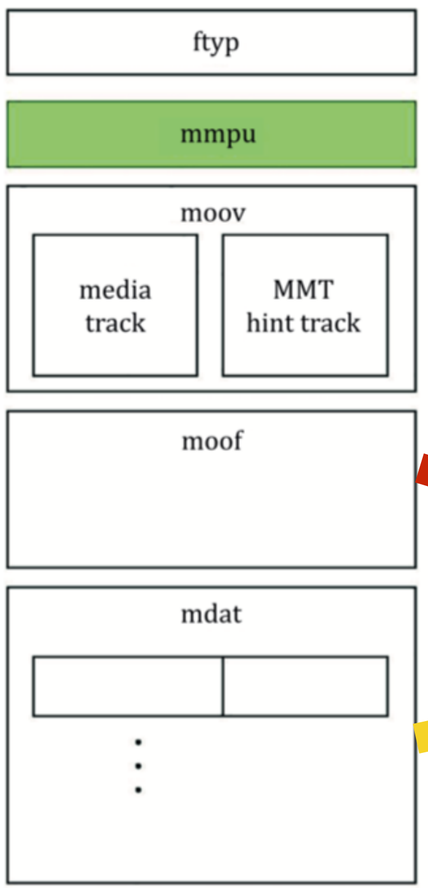
Implied
“normative” reference
“out-of-order” mode



https://github.com/jjustman/libatsc3/raw/master/architecture/2019-10-25%20-%20MMT%20Design%20-%20Enhancement%20-%20MFU%20and%20SCTE35%20Support%20for%20ATSC3.0%20-%20jjustman%20-%20draft%20v1.8%20-%20EXT.pdf

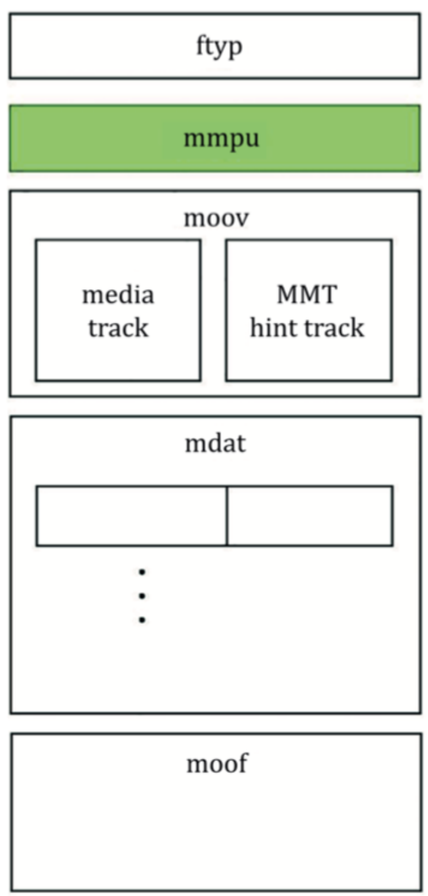
MMT - Robust Depacketization - Out of order mode

“in-order” mode



How can an encoder create a list of all samples and their length before the samples have been encoded?

“out-of-order” mode



https://github.com/jjustman/libatsc3/raw/master/architecture/2019-10-25%20-%20MMT%20Design%20-%20Enhancement%20-%20MFU%20and%20SCTE35%20Support%20for%20ATSC3.0%20-%20jjustman%20-%20draft%20v1.8%20-%20EXT.pdf

MMT - Robust Depacketization - Out of order mode

Receiver Privilege implies Encoder must support MFU emission

9.4.2.2 MMT receiving entity operation

The depacketization procedure is performed at the MMT receiving entity to rebuild the transmitted MPU. The depacketization procedure may operate in one of the following modes, depending on the application needs:

— MPU mode: in the MPU mode, the depacketizer reconstructs the full MPU before forwarding it to the application. This mode is appropriate for non-time critical delivery, i.e. the MPU's presentation time as indicated by the presentation information document is sufficiently behind its delivery time.

— Movie Fragment mode: in the Fragment mode, the depacketizer reconstructs a complete fragment including the fragment metadata and the "`mdat`" box with media samples before forwarding it to the application. This mode does not apply to non-timed media. This mode is suitable for delay-sensitive applications where the delivery time budget is limited but is large enough to recover a complete fragment.

— MFU mode: in the media unit mode, the depacketizer extracts and forwards media units as fast as possible to the application. This mode is applicable for very low delay media applications. In this mode, the recovery of the MPU is not required. The processing of the fragment media data is not required but may be performed to resynchronize. This mode tolerates out of order delivery of the fragment metadata MFUs, which may be generated after the media units are generated. This mode applies to both timed and non-timed media.

Using the MFU sequence numbers, it is relatively easy for the receiver to detect missing packets and apply any error correction procedures such as ARQ to recover the missing packets. The payload type may be used by the MMT sending entity to determine the importance of the payload for the application and to apply appropriate error resilience measures.

https://github.com/jjustman/libatsc3/raw/master/architecture/2019-10-25%20-%20MMT%20Design%20-%20Enhancement%20-%20MFU%20and%20SCTE35%20Support%20for%20ATSC3.0%20-%20jjustman%20-%20draft%20v1.8%20-%20EXT.pdf

ATSC 3.0 with ROUTE-DASH

ROUTE-DASH limitations not described in A/331 nor A/351RP

1.) No robust depacketization of ROUTE objects.

Specification provides the ability to use AL-FEC, but contains a temporal latency (e.g. upwards of 30s), and payload overhead (5% additional utilization for repair symbols).

2.) No Media-fragmentation aware transport.

Live-linear media essences, including signaling information, must receive EVERY PACKET in order to reconstruct the delivered object.

3.) DASH does not support sparse media tracks.

Multi-plp with robust or scalable emission will fail with DASH.

ATSC 3.0 with ROUTE-DASH

ROUTE-DASH limitations implied in A/331, but lacking implication and impact with the “application” being DASH playback:

Upon recovery of both the complete set of packet payloads for the delivery object associated with a given TOI value, and the metadata for that delivery object, the object is handed to the application.

Metadata recovery depends on the applied delivery mode.

A.3.10.3. General Metadata Recovery

Typically, delivery objects are only handed up to the application if they are complete and intact.

However, in certain circumstances a partially received object may be handed up, if the application API permits this and assuming that sufficient metadata is available to enable the application to handle the partial object. Regarding one defined mechanism for this, please refer to ETSI TS 126 247 [49], Clause 5.3.2 and Clause 7.3.9, as well as to ETSI TS 16 346 [14], Clause 7.9.

MMT Packet Analysis - Start of “GOP” / MPU Sequence

MPT Messages: Carry Asset Information, including “PTS” timing per <packet_id, mpu_sequence_number>

```
22019  77.586807  172.16.200.1  239.255.51.1  ATSC3 MMTP  5001   130 MPT_message 2 (0x0012): packet_id:   100, si_message_version:  99
22020  77.586820  172.16.200.1  239.255.51.1  ATSC3 MMTP  5001  1330 packet_id: 100, mpu_seq: 1663358407, moov
22021  77.586830  172.16.200.1  239.255.51.1  ATSC3 MMTP  5001   130 MPT_message 2 (0x0012): packet_id:   100, si_message_version:  99
22034  77.653876  172.16.200.1  239.255.51.1  ATSC3 MMTP  5001  1514 packet_id: 100, mpu_seq: 1663358407, mdat sample: 1, offset: 0
22035  77.653989  172.16.200.1  239.255.51.1  ATSC3 MMTP  5001  1514 packet_id: 100, mpu_seq: 1663358407, mdat sample: 1, offset: 1432
```

```
Identifier Mapping Type: MMT_IDENTIFIER_TYPE_MAPPING_ASSET_ID (0)
Asset ID Scheme: 0
Asset ID Length: 16
Asset ID Bytes: 11111111111111111111111111111111
Asset Type: hev1
1111 11.. = Reserved_6: 63
.... ..0. = Default Asset Flag: 0
.... ...1 = Asset Clock Relation Flag: 1
Asset Clock Relation ID: 0
1111 111. = reserved_7: 127
.... ...1 = Asset Timescale Flag: 1
Asset Timescale: 90000
Asset Location Count: 1
Location Type: MMT_LOCATION_TYPE_SAME_MMTP_PACKET_FLOW (0)
Packet ID: 100
Asset Descriptors Length: 15
Asset Descriptors Bytes: 00010c6324d5c7e6e8b5c568311fff
Descriptor
  Descriptor Tag: MMT_MPU_TIMESTAMP_DESCRIPTOR (1)
  MPU Sequence Number: 1663358407
  MPU Presentation Time: 16638748682781532159
```

```
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ........ ......E·
0010  00 74 58 82 40 00 40 11  4a e4 ac 10 c8 01 ef ff   ·tX·@·@· J·......
0020  33 01 cd 65 13 89 00 60  97 84 62 02 00 64 b5 c9   3··e···` ··b··d··
0030  20 99 68 e3 3c 8e 85 e5  4e ad a4 00 00 00 00 12    ·h·<··· N·......
0040  63 00 3f 12 63 00 3b 04  01 00 00 00 00 00 00 00   c·?·c·;· ........
0050  00 10 11 11 11 11 11 11  11 11 11 11 11 11 11 11   ........ ........
0060  11 11 68 65 76 31 fd 00  ff 00 01 5f 90 01 00 00   ··hev1·· ···_····
0070  64 00 0f 00 01 0c 63 24  d5 c7 e6 e8 b5 c5 68 31   d·····c$ ······h1
0080  1f ff                                             ··
```

MMT Packet Analysis - Start of “GOP” / MPU Sequence

MPU Fragment Type: Init Metadata - similar to the MOOV ISOBMFF box header,

```
22020 77.586820  172.16.200.1   239.255.51.1    ATSC3 MMTP   5001    1330 packet_id: 100, mpu_seq: 1663358407, moov
22021 77.586830  172.16.200.1   239.255.51.1    ATSC3 MMTP   5001     130 MPT_message 2 (0x0012): packet_id:   100, si_message_version:  99
22034 77.653876  172.16.200.1   239.255.51.1    ATSC3 MMTP   5001    1514 packet_id: 100, mpu_seq: 1663358407, mdat sample: 1, offset: 0
22035 77.653989  172.16.200.1   239.255.51.1    ATSC3 MMTP   5001    1514 packet_id: 100, mpu_seq: 1663358407, mdat sample: 1, offset: 1432
22036 77.654004  172.16.200.1   239.255.51.1    ATSC3 MMTP   5001    1514 packet_id: 100, mpu_seq: 1663358407, mdat sample: 1, offset: 2864
```

```
        MPU Payload Length: 1268
        0000 .... = MPU Fragment Type: Init/MPU Metadata (0)
        .... 1... = Timed Flag: 1
        .... .00. = Fragmentation Indicator: Complete DU (0)
        .... ...0 = Aggregation Flag: 0
        Fragmentation Counter: 0
        MPU Sequence Number: 1663358407
        DU
```

```
0050  00 00 00 00 69 73 6f 6d  6d 70 75 66 00 00 00 25    ····isom mpuf···%
0060  6d 6d 70 75 00 00 00 00  80 63 24 d5 c7 00 00 00    mmpu···· ·c$·····
0070  01 00 00 00 0c 76 69 64  65 6f 61 73 73 65 74 30    ·····vid eoasset0
0080  31 00 00 04 b1 6d 6f 6f  76 00 00 00 6c 6d 76 68    1····moo v···lmvh
0090  64 00 00 00 00 00 00 00  00 00 00 00 00 00 03 a9    d··· ···· ·····
00a0  80 00 00 00 00 00 01 00  00 01 00 00 00 00 00 00    ···· ···· ····
00b0  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00    ···· ···· ····
00c0  00 00 00 00 00 00 01 00  00 00 00 00 00 00 00 00    ···· ···· ····
00d0  00 00 00 00 00 40 00 00  00 00 00 00 00 00 00 00    ····· @·· ····
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00    ···· ···· ····
00f0  00 ff ff ff ff 00 00 02  7a 74 72 61 6b 00 00 00    ···· ···· ztrak···
0100  5c 74 6b 68 64 00 00 00  07 00 00 00 00 00 00 00    \tkhd··· ····
0110  00 00 00 00 01 00 00 00  00 00 00 00 00 00 00 00    ···· ···· ····
0120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 01 00    ···· ···· ····
0130  00 00 00 00 00 00 00 00  00 00 00 00 00 00 01 00    ···· ···· ····
```

MMT - Start of each sample - “Hint Track”

MMTHSample provides **media-fragmentation aware transport**

8.3 Sample format

8.3.1 Definition

Each media sample will be assigned to one or more MFUs. Each sample of the MMT hint track will generate one or more MFUs. The hint sample may omit certain bytes of an MFU if deemed redundant, such as the length field of a NAL unit in the case of AVC or HEVC video bitstreams.

8.3.2 Syntax

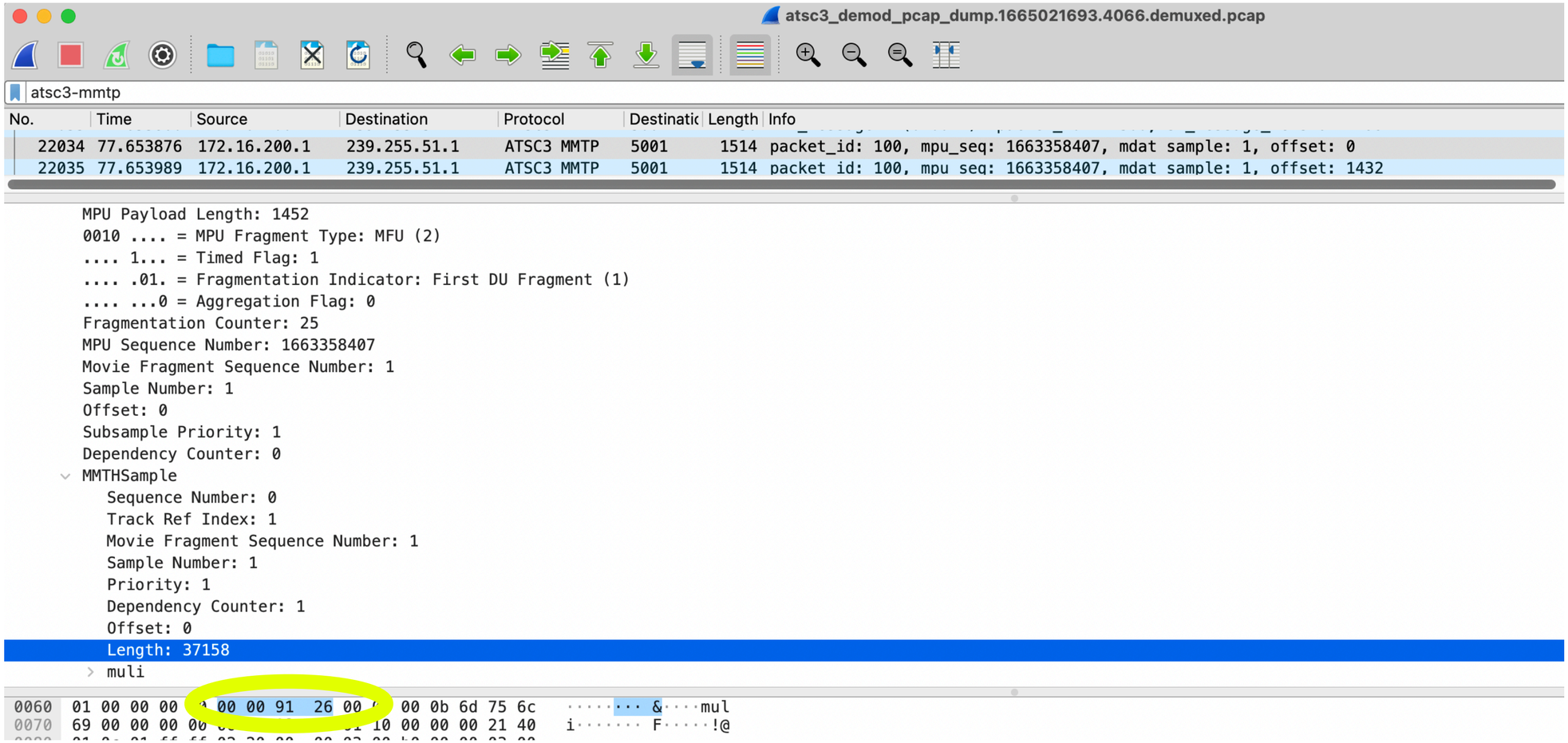
```
aligned(8) class MMTHSample {

   unsigned int(32) sequence_number;
   if (is_timed) {
      signed int(8) trackrefindex;
      unsigned int(32) movie_fragment_sequence_number
      unsigned int(32) samplenumber;
      unsigned int(8)  priority;
      unsigned int(8)  dependency_counter;
      unsigned int(32) offset;
      unsigned int(32) length;
      multiLayerInfo();
   } else {
      unsigned int(16) item_ID;
   }
}
```

…most importantly the “**length**” of this sample

MMT - Start of each sample - “Hint Track”

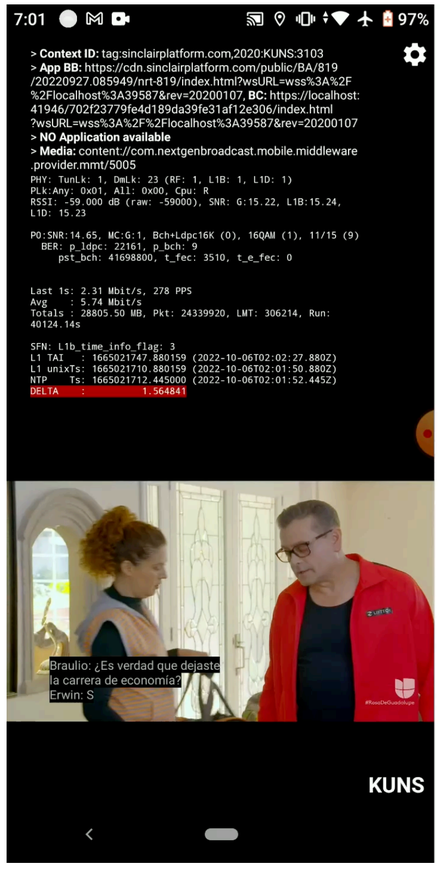
Example Parsing of MMTHSample for Sample #1 @ offset 0



1.) MMTHSample.length: 0x00 0x00 0x91 0x26 -> **37158 bytes long**

2.) Fragmentation Counter: 25 DU’s contained in this sample

MMT Playback Demo - 16QAM

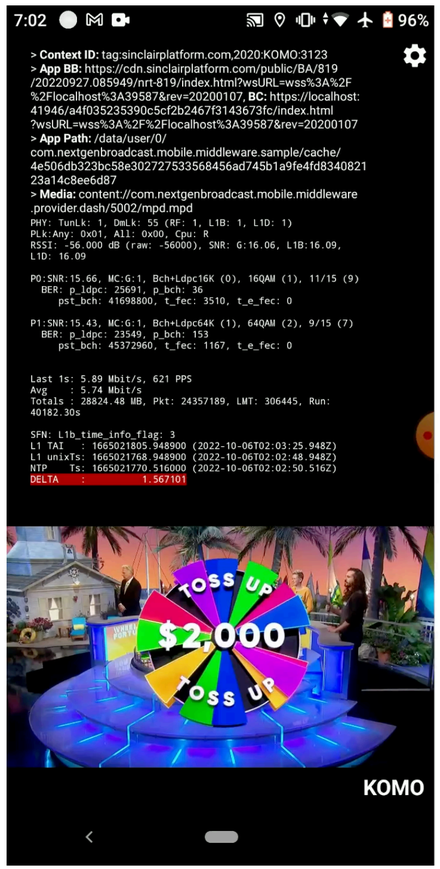


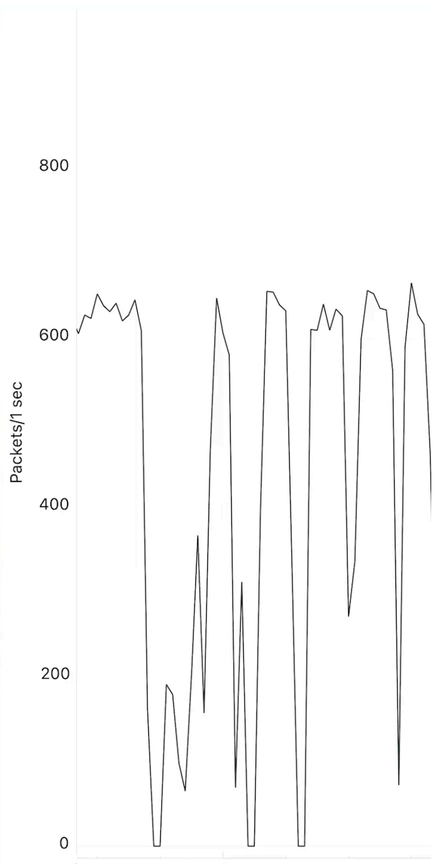
Graph shows demodulated PPS (Packets per second)

Video Link:

https://www.dropbox.com/s/fhm3s9mq7pkpfa6/2022-10-05-wba-screen-recording.mmt.mp4?dl=0

ROUTE Playback Demo - 64QAM



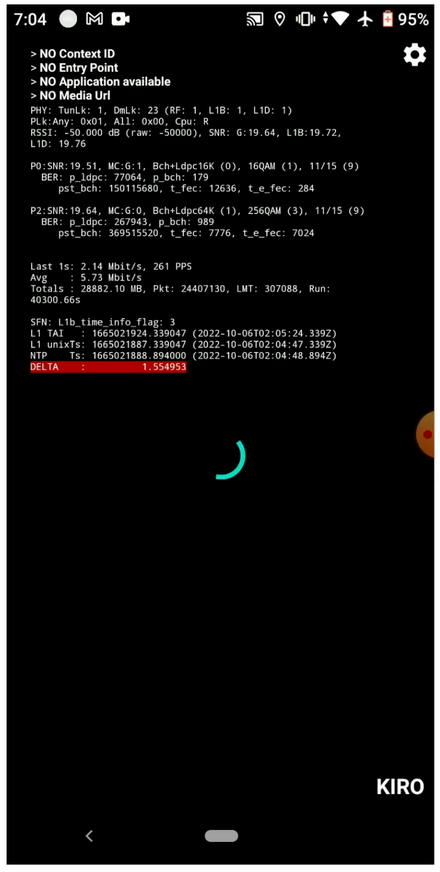


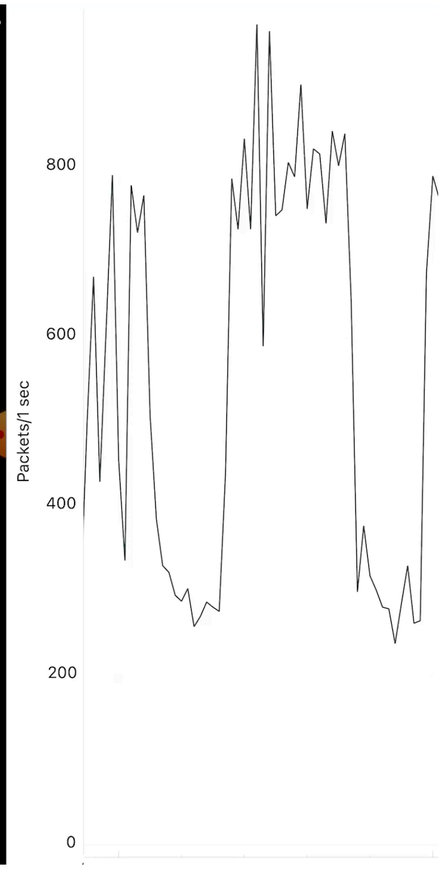
Graph shows demodulated PPS (Packets per second)

Video Link:

https://www.dropbox.com/s/r0jinay6kqzp6lx/2022-10-05-wba-screen-recording-route-64qam.mp4?dl=0

ROUTE Playback Demo - 256QAM





Graph shows demodulated PPS (Packets per second)

Video Link:

https://www.dropbox.com/s/ys51y18vt3xynsn/2022-10-05-wba-screen-recording-route-256qam.mp4?dl=0

Conclusion

MMT enables robust and reliable live linear content via ATSC 3.0 and IP multicast to the next generation of mobile handset devices and mobility receivers, without sacrificing traditional fixed CE device experiences.

Supplemental Links

libatsc3: https://github.com/jjustman/libatsc3

libatsc3 Android Sample App: https://github.com/jjustman/libatsc3-middleware-sample-app

libatsc3 Android TV App: https://github.com/jjustman/libatsc3-android-tv-app

Wireshark with ATSC3.0 Protocol Support: https://github.com/jjustman/wireshark-atsc3

Presentation Video Demos/ATSC3 Demodulated PCAP for Wireshark analysis:

https://www.dropbox.com/sh/wfokixhx0qtldba/AAB92nOboJQbJFeU5ZJ-SY-ya?dl=0

THANK YOU