HYBRID CONTENT GENERATION WORKFLOWS: FROM MPEG-2 TS TO ADAPTIVE STREAMING AND ISO-BMFF

Yasser Syed, PhD and Alex Giladi
Comcast/Technology & Product: VIDEO Architecture Group
InterDigital Communications, Inc.

Abstract

A transformation is happening in the content network delivery systems of the cable / IPTV world, which is increasingly moving us from a traditional MPEG-2 TS [Transport Stream] push-based ecosystem, toward a pull-based unicast ecosystem. As this transformation is occurring, we still need to maintain a consistent service experience, including aspects such as linear programming, DVR, accessibility features, etc. An established set of mechanisms and operational procedures already exists, via the traditional MPEG-2 TS model. Rather than rebuilding these mechanisms and operational procedures again, a hybrid content generation workflow model can be utilized to convert these services. This paper examines how this conversion can occur, such that the source content & service data may originate, be conditioned, and processed in a traditional MPEG-2 TS environment – but ultimately delivered to the customer in one or more adaptive streaming formats.

INTRODUCTION

There are two concurrently-happening paradigm shifts marking the industry's transition from traditional QAM-based distribution to multi-screen adaptive streaming. The first shift is the underlying distribution model. While the industry's original digital video distribution model offered push-based multicast delivery, the emergence of multi-screen environments that rely on HTTP downloads, which inherently are pull-based.

The second paradigm shift the container format. The industry relied on MPEG-2 transport streams for nearly two decades, but use of fragmented ISO-BMFF [ISO - Base Media File Format] as a container format is on the rise. The two formats had completely different design goals – while MPEG TS were built for efficient low-latency real-time transmission, the ISO-BMFF is instead a media *storage* format. Segment-based delivery makes the use of ISO-BMFF feasible, and the departure from tightly synchronized MPEG-2 TS multiplexes makes advanced features possible – such as late binding, where audio and video components can be separately selected by the client and downloaded, then synchronized at the receiver.

Those two shifts are buttressed by one over-riding reality: The functionality and customer experience in a multi-screen environment needs to be a superset of the one to which they are accustomed, and not a radical departure. Moreover, many regulatory requirements apply – or are expected to apply – to multi-screen world same way as they apply to current cable systems. This requires a precise translation of existing functionalities into the new ecosystem.

Despite the tectonic shifts at the transport and container level, the basics have not changed as much: The bitstreams being decoded are still the same bitstream formats, regardless of whether they're packaged into ISO-BMFF boxes or MPEG-2 TS packets.

This paper examines how a hybrid system can be designed using MPEG-2 TS and DASH (Dynamic Adaptive Streaming over HTTP) constructs to create a resource-efficient and robust system. The system supports scalable and cost-efficient content delivery using one or more popular of adaptive streaming technologies. It also discusses constructs and processes that are being defined in standards bodies and consortia, including MPEG, SCTE (WG7) and the DASH Industry Forum (DASH-IF), all of which assist in the definition of and conversion between network distribution and storage, and end-client distribution.

THE MPEG-2 TS MULTIPLEX

The MPEG-2 Transport Stream is the backbone of all current cable, IPTV and satellite distribution networks. It entered the digital video marketplace two decades ago. MPEG-2 TS also accounts for a sizeable part of the adaptive streaming landscape, especially considering that the TS-based Apple's HLS (HTTP Live Streaming) is still considered a dominant adaptive streaming system . Beyond linear and on demand distribution systems, MPEG-2 TS is also used in physical media (Blu-Ray).

The MPEG-2 transport stream is a ubiquitous media carriage mechanism that can define parameters and interaction between multiple packetized elementary streams, to create a viewable media experience. Some of its features include:

- Tight encoder/decoder synchronization: The transport stream format provides enough information to assist the decoder in reconstructing the encoder clock
- Single clock use for all media delivery, decoding and presentation
- A single buffer model to allow reliable receiver implementation
- Multiplexing: A single MPEG-2 TS stream may contain multiple components (video, audio, and data) for a single program, as well as multiple unrelated programs
- Codec independence: While originally designed in early '90s together with MPEG-2 audio and video, MPEG-2 TS successfully carries newer formats, such as H.264, H.265, AAC and (E-)AC-3; moreover it can be easily extended to support newer generations of codecs.
- Fast channel acquisition and recovery in the stream or file, through repetition of the system information (SI) located in tables such as the Program Management Tables (PMTs) and Program Association Tables (PAT.) The underlying codecs are expected to repeat all sequence-level information – such as sequence and picture parameter sets (SPS / PPS) in H.264 and H.265 video codecs.

Over the past two decades, a significant amount of infrastructure was built that supports various broadcast, cable and IPTV use cases related to transport streams (see `Figure 1`.) In the case of North America, an example is SCTE 35, an in-band cueing and messaging mechanism for ad insertion, blackouts, and alternate content. SCTE 35 allows easy communication between content providers and service providers [2.] Traditional TS-based systems also provide in-band functionality for purposes such as CEA 608/708 closed captioning and active format description (AFD).

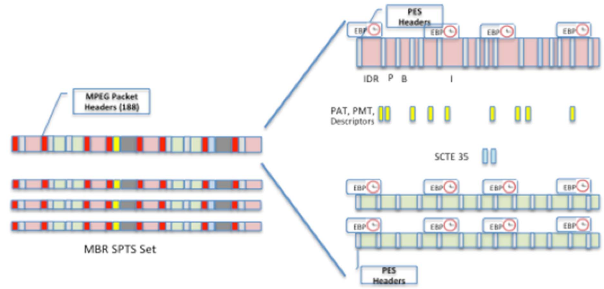

Figure 1: MPEG-2 TS STRUCTURES

Furthermore, the maturity of the technology enabled the development of a proven set of testing methodologies and equipment. Those developments validated media, enabled service monitoring, and shored up the troubleshooting mechanisms that can both isolate faults and ensure service quality .

By contrast, the shift to adaptive streaming shares more commonality with progressive HTTP download of video files, than with the traditional push-based delivery.

As opposed to the traditional and continuous stream, adaptive streaming methods segment each stream, and an end-client is offered a description of available segments – a *manifest*. The majority of the logic resides at the client device, and it is the client device that decides which of the available alternatives it can sustainably present, as described in the manifests. The alternatives – *representations* are listed in the manifest.

For the rest of this discussion, we will use terminology based on DASH, as it provides a very rich and precise vocabulary for manifest description (see `Figure 2`). In the DASH taxonomy, *representation* conceptually means single media (one or more elementary streams), comprised of multiple short *segments* (otherwise known as chunks.) An *adaptation set* is a set of seamlessly switchable representations, and a *period* is a time box within which representations exist. Each period can be viewed as an independent playlist (similar to a master m3u8 in HLS) or a completely different program, and a period border can be seen as a splice point. [Note that the fact that we are using DASH terminology here does not mean that we are limiting ourselves to DASH outputs.]

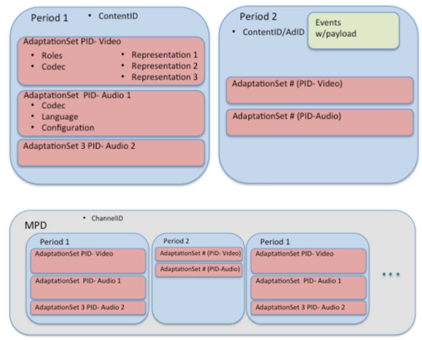

Figure 2: DASH MANIFEST STRUCTURE

The adaptive streaming market today is fragmented into several technologies, which are at their cores are very similar, usually varying only by container type. The dominant system is Apple's venerable HTTP Live Streaming (HLS), which is deployed in several different variations (protocol version 3 is very different from 6, for instance.) Next in popularity are Adobe's HDS (HTTP Dynamic Streaming), Microsoft's SmoothStreaming and MPEG DASH – the latter being a relatively new international standard for adaptive streaming. In practice if support for a reasonable crop of devices is needed this means a requirement for several "profiles," targeted to different devices classes. For context, a Netflix presentation from May 2014 mentions 6 profiles in use within its content generation workflow [8]. Each profile will invariably target a different crop of devices and use different adaptive streaming systems.
As far as segment formats are concerned, several options are used simultaneously. HLS uses MPEG-2 TS segments, as well as MPEG-1 audio elementary streams. HDS and SmoothStreaming use fragmented ISO-BMFF. Lastly, DASH allows both ISO-BMFF and MPEG-2 TS. The DASH-IF IOP (DASH Industry Forum / Interoperability Points) and DVB (Digital Video Broadcast) DASH groups use ISO-BMFF, while the SCTE (Society of Cable Telecommunications Engineers) recently standardized both MPEG-2 TS-based profile (SCTE DASH/TS) and ISO-BMFF-based profile (SCTE DASH/FF) as DVS 1202, DVS 1203 and DVS 1208.

Moreover, if the work of providing multi-rate transcoding is already happening, a convenient optimization would focus on generating content for traditional QAM-based distribution, using the same infrastructure.

Complicating things even further is the matter of multiple encryption schemes, which may also need to be supported. HLS alone comes with two incompatible segment encryption modes, and neither are suitable for MPEG-2 conditional access systems, which can be trivially retargeted to traditional QAM-based distribution.

This demand to produce a multitude of output formats is not necessarily an insurmountable problem – notice that the underlying audio, video and closed captioning bitstreams are the same. The change is only at container and encryption layers. Because at the core there are more things in common than are different, this immediately led us to establish the concept of a "common intermediate format" or CIF. With it, the generation of segments in specific container format (*packaging* or *multiplexing*) can be decoupled from the transcoder's processing of content.

MULTI-TIER CONTENT GENERATION WORKFLOWS

A simple approach to generating multi-rate content for ABR distribution, from a real-time MPEG-2 TS feed, can be summarized as (1) transcoding the source into the appropriate set of bitrates, (2) writing the generated segments to the origin server, and (3) letting CDNs [Content Delivery Networks] take care of the rest. Had this solution worked satisfactorily and scaled well, this paper would have been much shorter! As often is the case, the devil is in the details – details of the service being deployed, the types of coverage needed, and the technologies supported.

Having considered a multitude of profiles – and the resulting outputs – we arrived at a seemingly more complex solution, but one that is ultimately more scalable. It's a multi-tiered system that consists of (1) a transcoding tier, which outputs in an intermediate multi-rate format, and (2) a packager tier that generates the final content from this intermediate format. The third tier of this system is the content protection tier, which integrates with DRM systems used for the deployment. This tier encrypts the packager output into the required format, such that the key, provided later by the licensing server, can be used to authorize secure playback in the user device. This solution reduces the amount of variations required to support today's video marketplace.

The computational complexity of packaging is orders of magnitude smaller than that of a transcoder. This means that a high-density solution – a single physical server simultaneously handling a large amount of streams – is feasible, and can run on off-the-shelf, inexpensive hardware. This produces order-of-magnitude hardware cost reductions – a single, inexpensive, off-the-shelf server, rather than a rack of real-time encoders running on specialized hardware. Operational benefits abound, too, in the form of lower energy and space overhead. Moreover, the horizontal scaling characteristics of a multi-tier system are significantly better than these of a traditional "all-in-one" transcoder.

Efficient multi-packager deployment benefits from the use of multicast distribution within an MSO-operated internal network. The easiest way to integrate the existing infrastructure with the new hybrid multi-rate workflows involves the reuse of the (currently ubiquitous) MPEG-2 TS over UDP (User Datagram Protocol) multicast. An alternative approach exists -- ISO-BMFF in conjunction with a reliable file multicast protocol (such as File Delivery over Unidirectional Transport, FLUTE). It provides a good solution for the wireless multicast case (e.g., LTE eMBMS, or Long Term Evolution / Evolved Multimedia Broadcast Multicast Service), but it, in our opinion, is an unjustifiably complex solution to the simpler problem of multicasting in a provisioned core network that is practically error-free. Hence, MPEG-2 TS became our container format of choice for the common intermediate format (CIF), and UDP multicast, with packets carrying a number of complete transport stream packets, is our multicast format of choice.

ADAPTIVE TRANSPORT STREAMS AS AN INTERMEDIATE FORMAT

The Adaptive Transport Stream (ATS) specification, SCTE DVS 1196, was created to satisfy the need for the CIF format. It provides backwards-compatible virtual segmentation, and adds some constraints to the MPEG-2 TS stream to allow for an easy conversion to adaptive streaming technologies (see `Figure 5`). The resulting single-program MPEG-2 TS stream is playable in its regular context, but the modifications allow the stream to be easily adapted into an adaptive streaming workflow.

ATS is built upon three main concepts: An *EBP structure*, which is a marker inside a transport stream packet; an *EBP descriptor,* which describes the multiplex as the part of in-band system information, and a *source description,* which is a DASH-inspired manifest that completely defines an *ATS set* – a set of multiplexes or/and files associated with the same content (see `Figure 3`).

The EBP structure is carried in adaptation field of a transport stream packet and is inserted at the transcoding stage. It adds the following:

- GOP [Group of Picture] conditioning of video and audio elementary streams, to make them segmentable, with the option to incorporate more than one segment size
- Segment adjustment for splice points
- Stream mark-up indicators to indicate segmentation points (e.g. virtual segments)
- A wall clock time indicator that accompanies each segment
- An adaptation field structure for stream mark-up, labeling, and time information (EBP structure)
- The creation and ability to segment-align multiple outputs from a single content source, so as to provide a multi-bit-rate set of streams as a source for different representation segments.
- Segmentation in encrypted domain is possible. Adaptation fields are never encrypted when a MPEG-2 CAS is used, hence segmentation is possible even without decrypting the incoming stream.

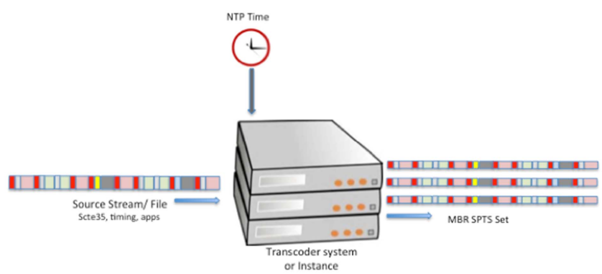

Figure 3: TRANSCODER ATS COMPLIANT STREAMS

The EBP descriptor in the PMT describes the element(s) of the ATS set carried in the multiplex. This description contains information such location of EBP structures for each elementary streams, its random access characteristics, and expected segment size. A more detailed description of the ATS set is provided within using source description document, which is presented in more detail later.

Adaptive transport streams can be transmitted via IP multicast to various parts of the network, to bring them closer to CDNs and edge caches (see `Figure 4`). To use these types of streams for adaptive streaming purposes, manifests and indexes need to be created.

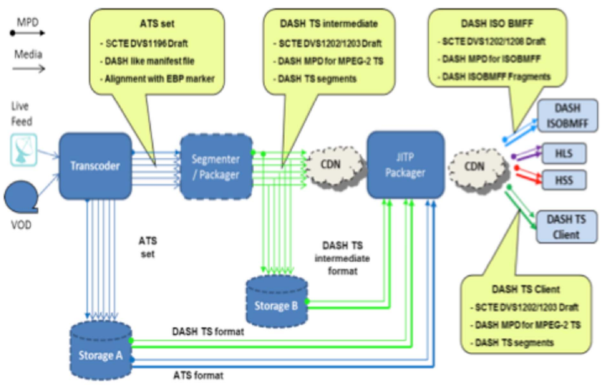

Figure 4: REFERENCE ARCHITECTURE DIAGRAM [7]

ATS SOURCE DESCRIPTION AS AN INTERMEDIATE MANIFEST FORMAT

An ATS set needs a detailed description format that brings together differerent files and multiplexes within the set and provides sufficient amount of information for generating content in the end format. Feature-richness of DASH makes it a perfect candidate for describing the multitude of representations contained in our intermediate format. It's a suitable base for the generation of client manifests used by different adaptive streaming systems. The purpose of source description is somewhat different from the one in adaptive streaming – a packager is told by the client explicitly which representation to chose. The packager needs to be able to select and transform intermediate format into the requested format; frequently the packager will also double as a manifest generator – providing the client with the manifest it will use for selecting the representation and requesting the segments.

Source description cannot really be called "DASH," as we have to allow for non-HTTP sources, such as UDP multicast and locally stored files – but it is as close as it can be. The selection of DASH Media Presentation Description (MPD) as a basis for the manifest format carries an interesting consequence: It makes it possible to mix UDP and HTTP into a single manifest, which means that an extended DASH/TS client can bring non-multicast resources over HTTP.

An ATS manifest needs to be designed that can be converted to an adaptive streaming manifest format – e.g. an HLS m3u8 or DASH MPD. DVS 1196 contains a definition of such a format, which is based upon DASH/TS – a format defined in SCTE DVS 1202 and 1203. The purpose of DVS 1202 is to establish container-independent constraints at the manifest and codec level that apply to both MPEG-2 TS (1203) and ISO-BMFF (1208) formats. DVS 1203 defines additional constraints, DASH MPD attributes and elements specific to an MPEG-2 TS format, while DVS 1208 defines additional constraints, DASH attributes and formats that are specific to ISO-BMFF profile and implemented through DASH-IF specification. This combination allows for the creation of a manifest for the intermediate format (using DVS 1202/1203) and the ability to convert this to a manifest for an ISO-BMFF file to an end client player (aka cloud-to-ground).

Intermediate format accommodates a superset of bitrates and formats needed to support traditional QAM distribution – thus MPEG-2 TS streams can be generated from it and be transmitted to existing set-top boxes, or used in legacy pay-per-view and VOD services delivered to QAM customers. Since our intermediate format is based on MPEG-2 TS, recreating a compatible stream is as simple as remultiplexing two-three representations, and applying the appropriate conditional access system.

Using this hybrid structure offers several advantages for handling video services. It allows the adaptive streaming systems – packagers and clients – to use information acquired through known signaling mechanisms in MPEG-2 TS (e.g. SCTE 104 / SCTE 35), without creating new mechanisms to gather this information. Further, video and audio quality can be monitored and analyzed in the network using existing testing equipment, rather than building new equipment in the adaptive streaming environment to do the same thing. It can save on storage by storing one common intermediate format, and it can save on transcoding costs by transcoding once for both services.

PACKAGING TIER – CREATING FLEXIBILITY, ROBUSTNESS, AND SCALE

The packager described in `Figure 4` fetches and converts CIF segments to end-client ISO-BMFF segments, and converts the CIF manifest into a manifest for ISO-BMFF-based adaptive streaming system, such as the DASH-IF IOP profile of MPEG DASH.

A simple packager implementation would rely on random access point signaling in MPEG-2 transport stream packets. Inserting random access points at fixed time intervals, and segmenting at every IDR frame, would suffice in this case. Despite our desire to see a single, universally-adopted adaptive streaming system and serve only it to all clients, this is far from becoming current marketplace realities. Thus dictates the need to support at least 2-3 of these systems. Different systems often require different segment lengths – e.g., HLS typically uses 8 second segments, while ISO-BMFF systems use 2 second chunks. The A-V delay is also an issue – for same video, the corresponding audio segment border would be in a different place in multiplexed MPEG-2 TS and in ISO-BMFF segments.

In a two-tier workflow implementation design, multiple packagers perform simultaneous segment generation at different locations. In order to allow a truly robust, scalable and distributed packager system, the packagers would be designed to be completely independent of each other, only implicitly synchronized by the PCR clock of their respective multicast inputs. It follows that in a fault-tolerant system, a failure of a packager instance should never affect the rest of the system.

Load balancing is another important consideration in package implementation design. It is typically handled internally at a CDN or cloud level, hence it is inadvisable, to perform crude DIY load balancing, using packager-specific URLs, and having the same session always bound to a single packager – which renders the overall system far more fragile and less scalable. As a result, we arrived at a requirement to have requests to same segment URLs return bit-identical HTTP response bodies, no matter which packager generated the segment. A client should be unaware which packager was serving it – this should be a load balancer decision and the load balancer should be able to change its decision at any arbitrary point in time.

The ATS format is the intermediate format of choice for a packager design – especially when mullticasts are needed or virtual segmentation is advantageous. An alternative and physically segmented format is the TS-based DASH/TS.

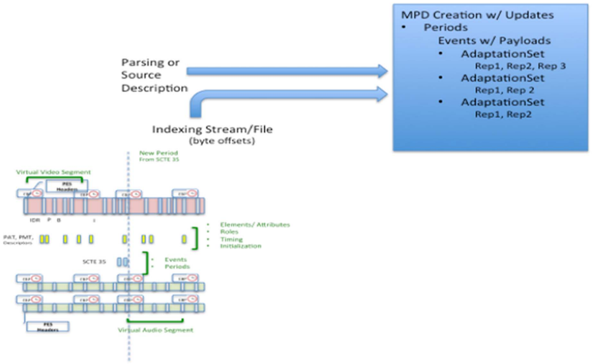

Figure 5: MANIFEST CREATION THROUGH MPEG-2 TS

In addition to segment generation, a just-in-time packager is expected to provide an up-to-date manifest upon client request. For this to happen, the just-in-time packager can use the intermediate format paradigm – using the DASH-like ATS source description.

Initial "bootstrapping" information needs to also make it to the new packager at its start-up time, by unspecified means. Once a packager has a sufficient knowledge of where the virtually segmented streams' location and characteristics, it then will maintain the manifest, reacting to the changes reflected in the incoming stream. These streams may be virtually segmented, adaptive transport streams or physically segmented DASH/TS.

In case of DASH/TS input, generation of an MPD is trivial – all that's required is a tight synchronization between the packager MPD and the origin MPD.

In some cases, an MPD needs to be generated and maintained by the expected changes in the stream, which can come directly from encoder. In the case of information propagated from the transcoder, a DASH MPD Update message can be used. This message contains a complete MPD within the stream, on PID 4, which was recently reserved by MPEG for precisely that purpose. MPD Update messages can be sent every time there is a change – e.g., switch to a different asset.

In the case of adaptive transport streams, maintenance of an MPD can be done either at the transcoder, or locally, at each packager. In both cases, new periods can be started at each splice point, or program changes can be indicated by the incoming SCTE 35 cue message. Out-of-band metadata or/and UPIDs in SCTE 35 segmentation descriptors can be used to derive period asset identifiers.

In case MPD is created automatically from the incoming MPEG-2 transport stream, translation of PMT and elementary stream information into MPD elements and attributes is relatively straightforward. Codec information can be derived from the PMT and – if needed – from parsing elementary streams.

As an example, the maximum segment duration and segment access point type can be learned from the EBP descriptor, and bandwidth values are also present in the adaptive stream PMTs.

Population of many of the adaptation sets and representation attributes also requires elementary stream parsing. Although some bits and pieces of information are available from (often optional) PMT descriptors, the expected level of detail in the MPD is higher than that. Profile/level information, if absent in the PMT, needs to be derived from the elementary streams.

ATSC caption service descriptors are used to signal carriage of CEA 608 / 708 closed captioning in MPEG-2 TS. These also need to be mapped into the MPD (for SCTE DASH and DASH-IF IOP 3.0) or into the m3u8 manifest for HLS.

Beyond language, audio descriptors can be used to derive signaling for associated services and broadcast / receiver mix. In case of MPEG DASH, a recent amendment to the specification created an interoperable way of signaling them

CONCLUSIONS

In this paper, a hybrid workflow system that allows reuse of existing infrastructure and creates a cost-effective backwards-compatible architecture for serving linear and on demand programming to large over-the-top audiences. It uses adaptive streaming technologies for segments using multiple container formats, without causing a large storage and transmission burden in order to support multiple formats. The system adds optimization, stability, scalability, and synchronization between transcoder systems and multiple packagers, using multicasting of MPEG-2 TS structures. Using the MPEG-2 TS structure and the manifest derived from this as a common intermediate format provides an elegant way to support today's fragmented adaptive streaming market, without being an undue burden. In the end, the hybrid delivery system takes advantage of the similarities in player decoding behaviors across adaptive streaming containers, and across the MPEG-2 TS and File format structures. Lastly, the concept of common intermediate format is extended to the player platform, to enable consistent browser playability. The paper describes an efficient way to convert MPEG-2 TS into ISO-BMFF-based adaptive streaming formats using the constraints defined in DVS 1196, DVS 1202, DVS 1203, and DVS 1208.

REFERENCES

[1] Jan van der Meer, Fundamental and Evolution of MPEG-2 Systems: Paving the MPEG Road, John Wiley & Sons, West Sussex, United Kingdom, 2014.

[2] ANSI/SCTE 35 2014, Digital Program Insertion Cueing Message for Cable.

[3] ANSI/DVS 1196 R1a 2015, Adaptive Transport Stream.

[4] ANSI/DVS 1202 R1a 2015, MPEG DASH for IP-Based Cable Services Part 1: MPD Constraints and Extensions.

[5] ANSI/DVS 1203 R1a 2015, MPEG DASH for IP-Based Cable Services Part 2: DASH/TS Profile.

[6] ANSI/DVS 1208 R1a 2015, MPEG DASH for IP-Based Cable Services Part 3: DASH ISO-BMFF Profile.

[7] ANSI/DVS 1206 R1a 2015, Reference Architecture for Cable ABR Services.

[8] Encoding At Netflix Media Meetup, May 2014- Netflix Digital Supply Chain, http://files.meetup.com/9842252/BAMM Meetup_NetflixEncoding_140521.pdf