Society of Cable Telecommunications Engineers

ENGINEERING COMMITTEE
Digital Video Subcommittee

AMERICAN NATIONAL STANDARD

ANSI/SCTE 217 2017

MPEG DASH Reference Architecture for IP-based Cable Services

NOTICE

The Society of Cable Telecommunications Engineers (SCTE) Standards and Recommended Practices (hereafter called documents) are intended to serve the public interest by providing specifications, test methods and procedures that promote uniformity of product, interchangeability, best practices and ultimately the long term reliability of broadband communications facilities. These documents shall not in any way preclude any member or non-member of SCTE from manufacturing or selling products not conforming to such documents, nor shall the existence of such standards preclude their voluntary use by those other than SCTE members, whether used domestically or internationally.

SCTE assumes no obligations or liability whatsoever to any party who may adopt the documents. Such adopting party assumes all risks associated with adoption of these documents, and accepts full responsibility for any damage and/or claims arising from the adoption of such Standards.

Attention is called to the possibility that implementation of this document may require the use of subject matter covered by patent rights. By publication of this document, no position is taken with respect to the existence or validity of any patent rights in connection therewith. SCTE shall not be responsible for identifying patents for which a license may be required or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Patent holders who believe that they hold patents which are essential to the implementation of this document have been requested to provide information about those patents and any related licensing terms and conditions. Any such declarations made before or after publication of this document are available on the SCTE web site at http://www.scte.org.

140 Philips Road
Exton, PA 19341

Table of Contents

List of Figures

1. Scope

This MPEG DASH Reference Architecture document is to serve as informational background to a suite of specifications that define the usage of MPEG DASH in cable networks. It introduces adaptive bit rate streaming as a general service and defines reference architecture in which content processing components, flows of process, use cases and scope definition of each part of other relevant documents are described.

2. Normative References

The following documents contain provisions, which, through reference in this text, constitute provisions of the standard. At the time of Subcommittee approval, the editions indicated were valid. All standards are subject to revision; and while parties to any agreement based on this standard are encouraged to investigate the possibility of applying the most recent editions of the documents listed below, they are reminded that newer editions of those documents may not be compatible with the referenced version.

- None are applicable.

3. Informative References

The following documents may provide valuable information to the reader but are not required when complying with this standard.

3.1. SCTE References

[1] SCTE 223 2017, Adaptive Transport Stream Standard
[2] ANSI/SCTE 214-1, MPEG DASH for IP-Based Cable Services Part 1: MPD Constraints and Extensions
[3] ANSI/SCTE 214-2, MPEG DASH for IP-Based Cable Services Part 2: DASH/TS Profile
[4] ANSI/SCTE 214-3, MPEG DASH for IP-Based Cable Services Part 3: DASH ISO BMFF Profile

3.2. Standards from other Organizations

[5] ISO/IEC 23009-1:2014 2nd Ed., Information technology -- Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats.
[6] ISO/IEC 14496-12:2015, Information technology -- Coding of audio-visual objects -- Part 12: ISO base media file format

3.3. Published Materials

[7] R. Pantos, W. May, HTTP Live Streaming, https://tools.ietf.org/html/draft-pantos-http-live-streaming-18

4. Compliance Notation

shall	This word or the adjective "***required***" means that the item is an absolute requirement of this specification.
shall not	This phrase means that the item is an absolute prohibition of this specification.
forbidden	This word means the value specified shall never be used.
should	This word or the adjective "*recommended*" means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighted before choosing a different course.
should not	This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
may	This word or the adjective "*optional*" means that this item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because it enhances the product, for example; another vendor may omit the same item.
deprecated	Use is permissible for legacy purposes only. Deprecated features may be removed from future versions of the standard. Implementations should avoid use of deprecated features.

5. Abbreviations and Definitions

5.1. Abbreviations

ABR	Adaptive Bit Rate
ATS	Adaptive Transport Stream
CDN	Content Delivery Network
cDVR	cloud Digital Video Recording
DASH	Dynamic Adaptive Streaming over HTTP
DASH/TS	DASH in MPEG-2 Transport Stream
EBP	Encoder Boundary Point
GOP	Group Of Pictures
HLS	HTTP Live Streaming
HSS	HTTP Smooth Streaming
HDS	HTTP Dynamic Streaming
HTTP	Hypertext Transfer Protocol
IDR	Instantaneous Decoding Refresh
IP	Internet Protocol
JIT Packager	Just In Time Packager
MPD	Media Presentation Description
SPTS	Single Program Transport Stream
VOD	Video On Demand

5.2. Definitions

Adaptive Sync	ATS streams that are both Time Synced and Chunk Synced.
Chunk	A discrete section of content that can be independently decoded, possibly given additional initialization information.

Chunk Boundary Point	A specialized Encoder Boundary Point that indicates the beginning of a chunk, and is a stream access point.
Chunk Sync	Chunk sync implies the identical AUs across representations at boundaries indicated by EBP.
Chunk Type	Refers to a specific partition that is contained in the ATS stream. Segment and fragment are examples of different types of chunks.
Closed GOP	A Closed GOP means that frames from the current GOP does not reference frames from the previous GOP.
Encapsulator	Processes a conditioned continuous group of elementary streams to create specific ABR-format chunks of mixed or separated elementary streams that are stored in a file or transmitted. Each file is wrapped to be in one or more adaptive streaming formats. An encapsulator does not normally perform any transcoding functions but depends on the conditioned stream to create those independently decodable sections. An encapsulator can also be known as a fragmenter, packager, or segmentor.
Fragment	A chunk that is aligned with boundaries in one component stream in the source multiplex. Fragment boundaries are typically explicit for each component. Smooth Streaming is an example ABR format that uses fragments.
Segment	A chunk with boundaries aligned to include all component streams in the source multiplex across the target presentation time range. Segment boundaries are typically explicit for only one main component (video, for example), and other component boundaries are implicitly derived from this main component. A segment is typically used when packaging content in HLS.
Time Sync	Indicates identical AUs across representations have the same presentation time stamp.

6. ABR Overview

There is a variety of Adaptive Bit Rate streaming wire formats. Some are based on an MPEG-2 Transport Stream container such as HTTP Live Streaming (HLS: Apple) and others on a fragmented MP4 container such as HTTP Smooth Streaming (HSS: Microsoft) and HTTP Dynamic Streaming (HDS: Digital Plus, Adobe), whereas Dynamic Adaptive Streaming over HTTP (DASH: MPEG) supports both containers. While different, they utilize common video and audio compression formats; namely: ISO/IEC 14496-10 (AVC) and ISO/IEC 14496-3 (AAC). Additional video formats, such as ISO/IEC 23008-2 (HEVC), and audio formats, such as AC-3, *may* be used.

In adaptive bit rate streaming, a file or linear source asset is encoded into multiple representations, typically uniquely characterized by resolution and bitrate. For example, the video of an asset *may* be encoded into N different bitrates, some of which *may* be of common picture resolutions. Furthermore, the encoding, while continuous, is broken up into small chunks. Each video chunk starts with a Stream Access Point of type 1-3 [6], typically an IDR or closed GOP. Each new audio chunk starts at an audio access unit boundary. These access points are referred to as Chunk Boundary Points.

In some ABR formats, audio is not multiplexed with video; discrete audio streams per language are produced. In some ABR formats, auxiliary data streams such as Subtitles and/or Captions *may* also be delivered as discrete streams.

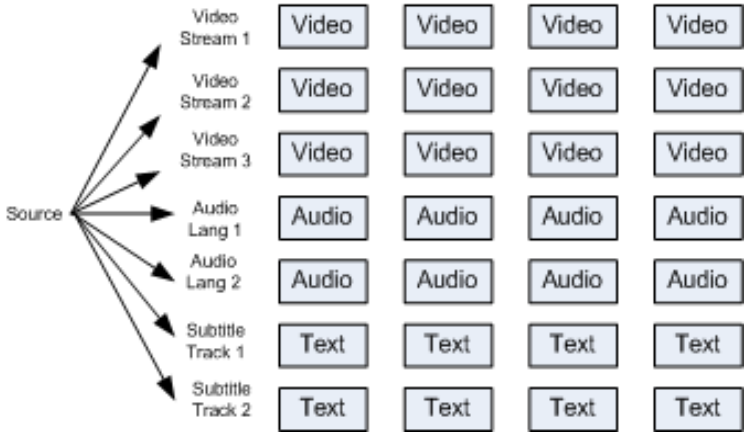

Figure 1 - ABR with discrete audio and two subtitle streams

Some formats do not require audio chunks to be of the same duration as the video chunks (shown here with only a single audio language):

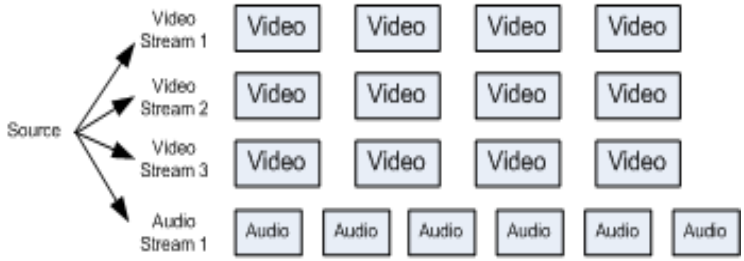

Figure 2 - ABR with audio durations different than video

Other formats (such as HLS prior to iOS-5) have audio multiplexed with video as below, here shown with two audio languages 1 and 2. As shown, all audio languages are to be present in all representations.

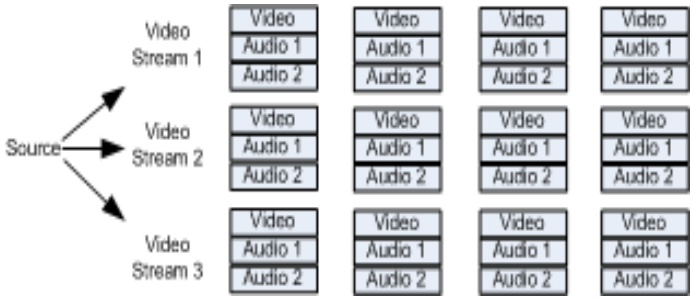

Figure 3 - ABR with interleaved audio and video streams

The adaptive nature is such that a client *may* dynamically switch for various reasons, such as bandwidth fluctuation, from one representation to another. It does so at chunk boundary points. In **Figure 4 - ABR chunk switching**, three streams at different bitrates are represented. At time A no switch is required. At time B, a change to Stream 3 occurs and then it continues until time D, at which time a choice is made to switch up to Stream 2. Finally at time E, it switches back to Stream 3.

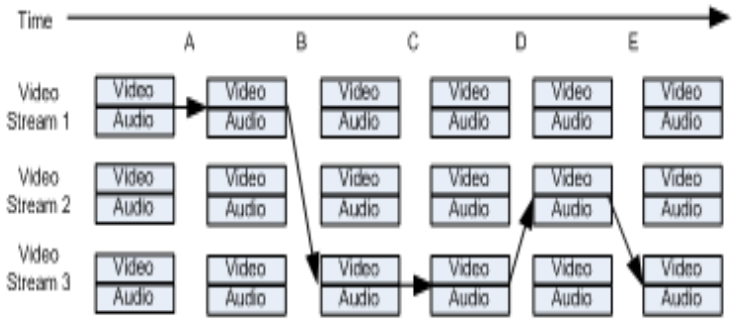

Figure 4 - ABR chunk switching

In order to maintain a seamless experience to the end user, the video and audio data of each of the various streams need to be in time sync with one another. This implies that the same source video frame V, or audio sample A, in all representations has the same rendered timestamp. Furthermore, the various streams need to be in chunk sync with one another. This implies that the same source video frame V, or audio sample A, at the beginning of each chunk in all representations has the same rendered timestamp. If streams are not in time and chunk sync with one another, then the end user would likely experience some form of discontinuity, a video and/or audio forward or backward disruption. The overall quality of experience would be degraded.

For various reasons, the optimal size of a chunk varies amongst the adaptive formats. HSS and HDS typically use a 2-second chunk duration. HLS typically uses anywhere from 6 to 10 second chunks. An encoding/transcoding system producing adaptive streams for use in multiple formats and also utilizing the same encoded data for the formats needs to maintain chunk alignment for all formats. Ideally, chunks of various formats usually align with one another and thus chunks of longer durations contain an integer number of chunks of shorter durations.

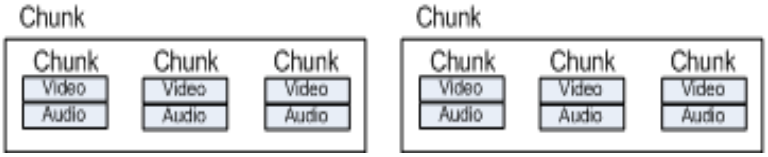

Figure 5 - Chunk alignment of various durations

Chunks in MPEG DASH are referred to as Media Segments. Chunks in the HSS/HDS formats are typically referred to as Fragments. Chunks in the HLS format are typically referred to as Segments. Here is another way to view the previous diagram using these terms:

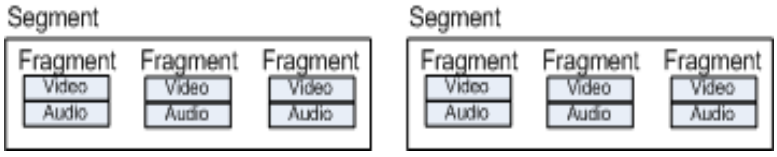

Figure 6 - ABR segments composed of ABR fragments

To further exemplify chunk sync in terms of fragment and segment sync, consider **Figure 7 - Fragment and Segment Sync/Alignment**, which illustrates a number of AVC streams based off of a common source. A switch point in the video domain occurs at the IDR of a chunk boundary. A chunk can contain additional I-pictures, additional GOPs, and the location of these can vary from one stream to another as

seen in streams 1 and 2. However, the start of fragment-chunks will be on the same IDR. Therefore, stream 3 is not in adaptive sync with streams 1 and 2. The same is true for segment-chunks. While stream 5's first segment is fragment-aligned with stream 4, it is not segment-aligned.

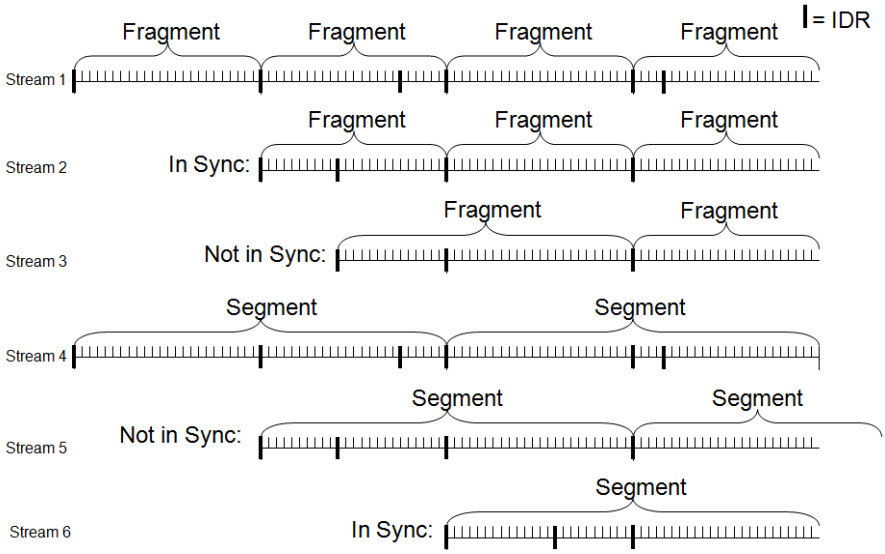

Figure 7 - Fragment and Segment Sync/Alignment

7. Reference Architecture

The MPEG DASH [5] standardizes a content delivery protocol using HTTP over IP network. It uses the adaptive bit rate streaming technology discussed above and is completely different from the traditional content delivery approach used by cable operators, in which MPEG-2 TS is transported over coax cable. It specifies a protocol interface between content server and subscriber client and defines the requirements of content preparation, e.g. encoding/transcoding, encapsulation and segmentation etc., to support adaptive bit rate streaming. However it assumes all of these is done in a unified system and does not consider interfaces with cable network structure for content preparation and delivery.

In a unified ABR encoding/transcoding and segmentation system, video and audio data are encoded and conditioned for adaptive streaming purposes and the resultant elementary compressed access units are fed to one or more ABR encapsulators or packagers to be formatted into ABR-specific wire formats.

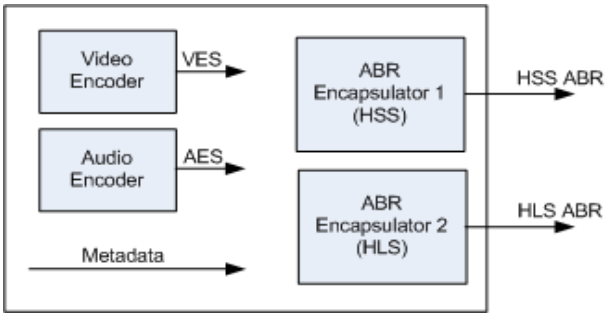

Figure 8 - Unified Transcoder/Encapsulator

In cable networks, content preparation and encapsulation *may* have to be separated to support a centralized encoding/transcoding and a distributed network edge based ABR encapsulation. The Adaptive Transport Stream (ATS) format, which is defined by DVS1196, allows separation between

encoding/transcoding and per ABR format encapsulation. The ATS provides a single set of conditioned streams with EBP marked virtual chunks that can be chunked by the encapsulator into different ABR formats. As **Figure 9 - Separate Transcoder/Encapsulator with ATS in between** illustrates, the ATS is used to carry ABR conditioned streams to edge encapsulators for producing ABR-specific formats.

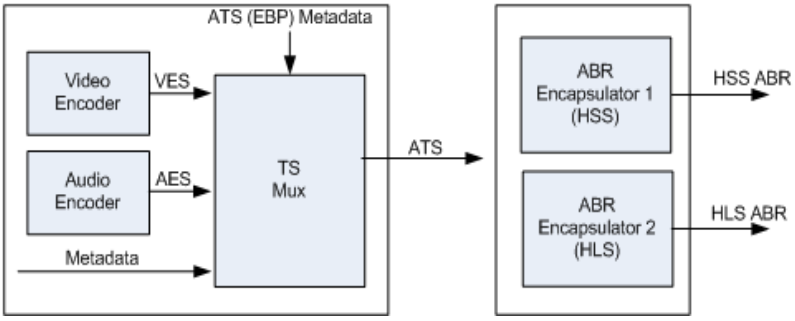

Figure 9 - Separate Transcoder/Encapsulator with ATS in between

In this document, it extends this design and defines an end to end reference architecture in which media source is taken as input, then prepared and conditioned for readiness of adaptive steaming, and then transported using ATS or DASH/TS format as mezzanine format to network edge server, where the subscribe devices is served via different adaptive streaming formats, including MPEG DASH. This reference architecture is used to help understanding of the scope and definition of a group of specifications that specifies the requirements and constrains to support MPEG DASH based content delivery in cable networks.

7.1. Content Processing Components and Flows

Figure 10 – DASH Service Reference Architecture Diagram summarizes all components and content process flows in the reference architecture.

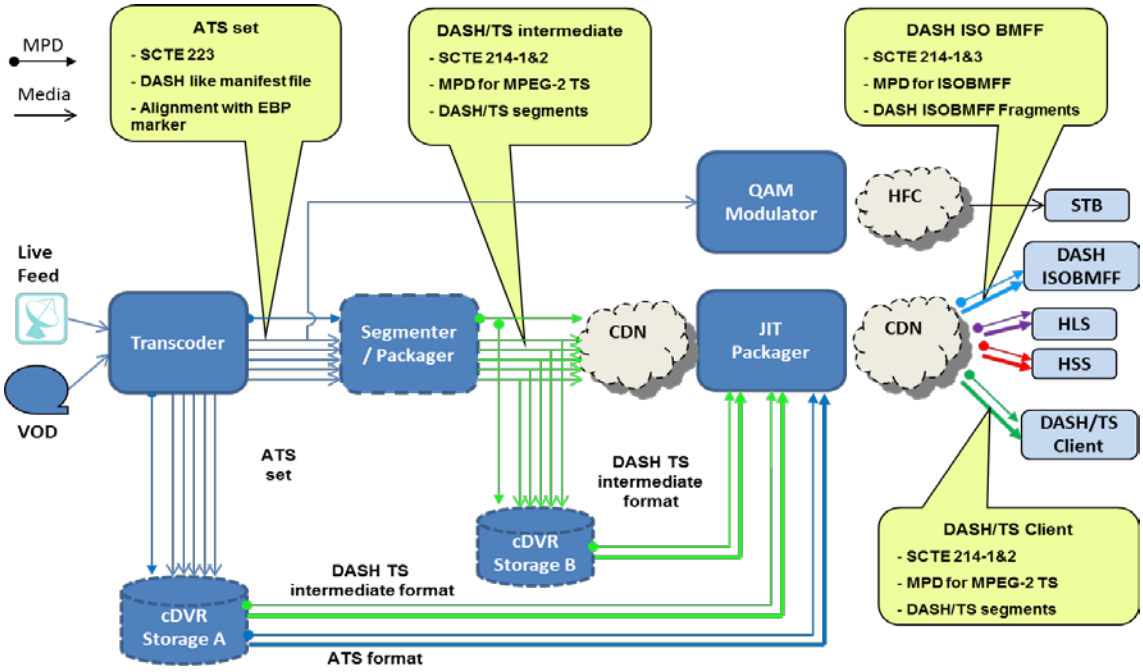

Figure 10 – DASH Service Reference Architecture Diagram

Transcoder: it takes content source either in linear stream format or in VOD file format, then transcodes it to multiple MPEG SPTS output streams in multi-bit rates and maybe also in multi-resolutions. This output stream set is referred as ATS set, which is a group of well-conditioned ATS streams, plus an ATS source description. See the definition and the details of ATS stream, ATS source description and ATS set in the SCTE 223 [1].

Segmenter/Packager: it takes ATS set in linear stream format as input and creates DASH/TS media segments, as well as converts the ATS source description to DASH/TS MPD, which can be transported by CDN or saved by cDVR storage server for DVR service.

cDVR Storage A server: it takes ATS set in linear stream format to support cDVR content storage or ATS set in file format to support VOD service. If DASH/TS output format is required, the saved media content needs to be parsed and prepared per DASH/TS requirements, such as DASH/TS MPD, index file, segmentation etc.

cDVR Storage B server: it takes content in DASH/TS format before or from CDN and saves it per cDVR service request. The saved media content can be published as DASH/TS output with minor modification of DASH/TS MPD. A DASH/TS index file maybe generated as part of the storage process.

JIT Packager: it takes either DASH/TS segment in intermediate format or ATS set in storage format as input upon the request of user client, and converts it to different ABR client formats, including DASH/TS or DASH ISO BMFF client format. The output from the JIT Packager is delivered to CDN origin server and then to the ABR clients as regular ABR service does.

QAM Modulator: it takes one stream of an ATS set in linear stream format as input and converts it into cable QAM format to serve the legacy set top boxes via HFC access network. The ATS is fully compliant with MPEG-2 Transport Stream and can be used directly to support legacy cable TV network.

7.2. Use Cases of ABR Services

The IP network based ABR services offer subscriber similar to or beyond the traditional cable TV services. Here it lists a few use cases.

7.2.1. Linear Service

The linear service is similar to the traditional Cable linear TV program, but it targets to deliver the program to terminal devices that are connected via IP network. The service flow starts from receiving linear content source, which then is encoded or transcoded and conditioned by encoder/transcoder to multi-bit rate multi-resolution MPEG-2 SPTS streams per the ATS specification [1]. The ATS streams are segmented and packaged by segmenter/packager to CDN deliverable chunks per the DASH/TS specification in SCTE 214-1 [2] and SCTE 214-2 [3], and then they are retrieved by JIT Packager upon end user client request, and finally they are converted to different ABR formats per client ABR format specification, such as DASH ISOBMFF format [2][4] or DASH/TS client format [2][3].

7.2.2. Cloud DVR Service

The cDVR service records the linear content in network cloud server for subscriber and supports playback on subscriber's request. There could be two recording options,

A. Recording linear content in ATS format off from encoder/transcoder output
B. Recording linear content in DASH/TS format off from segmenter/packager output

In the option A, the recording server receives a group of ATS streams together with its ATS source description document, and stores them per subscriber DVR schedule. The stored ATS set is prepared and packaged either for ATS file access per the ATS specification [1] or for DASH/TS access per the DASH/TS specification [2][3]. The preparation and package process DASH/TS includes ATS source description to MPD conversion, indexing and encapsulation if necessary.

In the option B, the recording server receives DASH/TS chunks and stores them without re-packaging. The DASH MPD file needs to be modified from DASH/TS linear to DASH/TS file access and an indexing file *may* also be generated.

When subscriber wants to playback the recorded content, the user client sends request to JIT Packager, which translates the user request to DASH/TS request and retrieves the content from cDVR storage server, and then delivers it to client device.

7.2.3. VOD service

The VOD service uses an off-line encoder/transcoder that takes content source, such as VOD content file, creates an ATS set and stores it to VOD storage server. Similar to the option A in the cDVR case, the stored ATS set is prepared and packaged either for ATS file access per the ATS specification [1] or for DASH/TS access per the DASH/TS specification [2][3]. The preparation and package process for DASH/TS includes ATS source description to MPD conversion, indexing and encapsulation if necessary.

When subscriber wants to play the VOD content, the user client sends request to JIT Packager, which translates the user request to DASH/TS request and retrieves the content from VOD storage server, and then delivers it to client device.

7.3. SCTE Specifications for Cable ABR Services

Based on these cable network ABR service requirements, the SCTE DVS DASH working group develops a suite of specifications. Referring to the **Figure 10 – DASH Service Reference Architecture Diagram**, each of the content flow points is addressed as below.

1) ATS set specification: it covers the scope of content preparation and conditioning during encoding/transcoding of content source to multi-bit rates and maybe multi-resolutions ATS streams and creation of ATS source description. The ATS format differs from all existing ABR formats, in which the media content is physically segmented, while an ATS stream is a MPEG-2 TS that carries pre-conditioned chunk boundary markers and is not segmented. The ATS set specification is documented in SCTE 223 [1].
2) DASH/TS intermediate specification: it covers the scope of content format used for transportation via CDN or stored for cDVR. The DASH/TS intermediate format serves as a mezzanine ABR format before it is converted to a specific ABR format. The DASH/TS intermediate specification is documented by SCTE 214-1 [2] and SCTE 214-2 [3].
3) DASH/TS client specification: it covers the scope of content format used specifically for content delivery to ABR client using MPEG-2 TS. The DASH/TS client specification is also documented in SCTE 214-1 [2] and SCTE 214-2 [3].
4) DASH ISO BMFF specification: it leverages the specification from DASH Industry Forum and defines specific constrains or restrictions to cable ABR client using ISO BMFF. The DASH ISO BMFF client specification is documented in SCTE 214-1 [2] and SCTE 214-3 [4].