Live Source Simulator
Summary of features May 2019

Torbjörn Einarsson, Edgeware AB
Waqar Zia, Nomor Research

00: 01: 09: 02
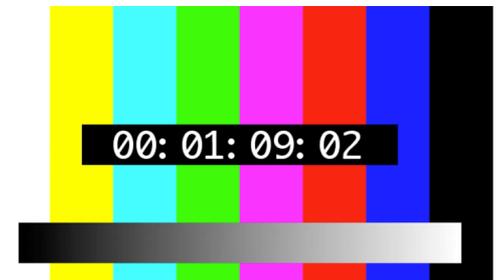
List of available features

- SegmentTemplate
- SegmentTimeline (Time + Number)
- Time-limited content
- Media session control
- Multi-period
- Multi-period Period continuity
- Inband event SCTE35
- UTCTiming
- Subtitles- EBU-TT-D
- Mixed segment length
- Xlink periods
- Early terminated period

- Start position control
- Multiple BaseURL
- MPD callback events
- *Segment chunk support*
- *Thumbnail support*
- *Start-over support*
- *Trick-play support*
- *Upcoming*
  - *More UTCTiming*
  - *Programmable segment loss*

On DASH-IF github incl. wiki info

New since last year

▸ Hosting
- AWS mid-size virtual machine (Oregon)
- New URL https://livesim.dashif.org (vm2.dashif.org will be closed down)
- Github: https://github.com/Dash-Industry-Forum/dash-live-source-simulator
- Wiki docs: https://github.com/Dash-Industry-Forum/dash-live-source-simulator/wiki

▸ Version 1.6 (Dec 2018)
- SegmentTimeline with $Number$
- suggestedPresentationDelay
- propagates default-sample-duration from trex and tfhd
- Can now generate sidx boxes

▸ Upcoming (next few weeks)
- Two more UTCTiming methods
- Programmable missing segments

Feature description 1

▶ SegmentTemplate with $Number

The feature supports segment addressing using SegmentTimeline with $Number$

http://livesim.dashif.org/livesim/testpic_2s/Manifest.mpd

Feature description 2

▸ SegmentTimeline

SegmentTimeline support with S elements consisting t,d and r attributes.

SegmentTimeline with $Time$-based addressing

http://live.dashif.org/livesim/segtimeline_1/testpic_6s/Manifest.mpd

SegementTimelne with $Number$-based addressing

http://live.dashif.org/livesim/segtimelinenr_1/testpic_6s/Manifest.mpd

Feature description 3

▸ Time-limited content

Support for controlling availability start time and end time.

Live service with 30min later availabilityStartTime (30min difference on screen): http://livesim.dashif.org/livesim/start_1800/testpic_2s/Manifest.mpd

In addition, to make the initialization segments available earlier than AST, by specifying init_ot where ot is an offset time in seconds: http://livesim.dashif.org/livesim/start_1800/init_10800/testpic_2s/Manifest.mpd

Here, the initialization segments are set to be available 3 hours in advance.

Note that one can influence the minimumUpdatePeriod by the parameter mup_x in the path and timeShiftBufferDepth by the parameter tsbd_x. Here x is the value in seconds.

Feature description 4

▸ Media-session control

Test content is available in a cyclic fashion (modulo x), the availabilityStartTime and the mediaPresentationDuration vary in an x-minute periodic pattern.

For example- 10-min cycle with 8min session and 5 different MPDs in this interval

http://livesim.dashif.org/livesim/modulo_10/testpic_2s/Manifest.mpd

Feature description 5

▶ Multi-period

New periods are available and added to the MPD periodically.

For ex: A new period every 3 min (20 times/hour)

http://livesim.dashif.org/livesim/periods_20/testpic_2s/Manifest.mpd

Feature description 6

▸ Multi-period Period continuity

The Adaptation Sets are continued after the first period, by providing Period continuity signaling with each Adaptation Set in the subsequent period.

Ex: New period every 1 min (60 times/hour) with period-continuity signaled

http://livesim.dashif.org/livesim/periods_20/continuous_1/testpic_2s/Manifest.mpd

Feature description 7

▸ Inband event SCTE35

Inband event signalled with schemeIdUri-urn:scte:scte35:2013:xml

SCTE35 Ad Splice cues 2 times per minute:

http://livesim.dashif.org/livesim/scte35_2/testpic_2s/Manifest.mpd

Feature description 8

- ▸ UTCTiming

UTCTiming element is provided in the MPD which can be used by the client to synchronize its clock.

Several UTCTiming schemes are supported.

1. UTC Direct

http://livesim.dashif.org/livesim/utc_direct/testpic_2s/Manifest.mpd

2. http-head

http://livesim.dashif.org/livesim/utc_head/testpic_2s/Manifest.mpd

3. UTC NTP

http://livesim.dashif.org/livesim/periods_60/utc_ntp/testpic_2s/Manifest.mpd

4. UTC SNTP

http://livesim.dashif.org/livesim/periods_60/utc_sntp/testpic_2s/Manifest.mpd

Feature description 9

▸ Subtitles EBU-TT-D

Live subtitles support with single/multiple subtitle tracks.

Sample MPDs:

http://livesim.dashif.org/livesim/testpic_2s/Manifest_stpp.mpd

http://livesim.dashif.org/livesim/testpic_2s/multi_subs.mpd

Feature description 10

▸ Mixed segment length

The segment duration for audio and video can be different.

(Supported for audio segment of 6s and video of 2s)

http://livesim.dashif.org/livesim/testpic_6s/mixeddur.mpd

Feature description 11

▸ Xlink period

Some of the periods in the MPD are accessed via xlink

Ex: 60 periods per hours, out of which 30 periods are accessed via xlink

http://livesim.dashif.org/livesim/periods_60/xlink_30/testpic_2s/Manifest.mpd

Ad insertion using xlink is supported.

Case-1: Normal ad insertion, where xlinks refer to a remote period that contains on-demand ad.

http://livesim.dashif.org/livesim/periods_60/xlink_30/insertad_1/testpic_2s/Manifest.mpd

Case-2: Ad insertion, where xlinks refer to a remote entity that contains two periods (of on-demand ad).

http://livesim.dashif.org/livesim/periods_60/xlink_30/insertad_2/testpic_2s/Manifest.mpd

Case-3: Ad insertion, the xlinks will refer to a remote entity that in turn refers to another remote entity.

http://livesim.dashif.org/livesim/periods_60/xlink_30/insertad_3/testpic_2s/Manifest.mpd

Feature description 12

▸ ## Early terminated period

Early terminated periods can be used to signal gaps in the media. For Early Terminated Periods, the value of the **Period**@duration is the presentation duration in Media Presentation time of the media content represented by the Representations in this Period.

For ex: 60 periods per hour, out of which 30 are early terminated periods. The duration of the early terminated period is 10 seconds.

http://livesim.dashif.org/livesim/periods_60/etp_30/etpDuration_10/testpic_2s/Manifest.mpd

Feature description 13

▶ ## Start position control

Support for client to start the playback from a certain position.

The time is specified using the POSIX clock with t parameter as below.

1. To start from live edge

http://livesim.dashif.org/livesim/testpic_2s/Manifest.mpd#t=posix:now

2. Start from a given UTC time

http://livesim.dashif.org/livesim/testpic_2s/Manifest.mpd#t=posix:1465406946

3. Play the stream range from Wed, 21 Jan 2015 20:04:05 GMT to Wed, 21 Jan 2015 23:44:33 GMT

http://livesim.dashif.org/livesim/testpic_2s/Manifest.mpd#t=posix:1421870645, 1421883873

Feature description 14

▸ Multiple BaseURL

Multiple BaseURLs with alternate availability of content in case of outage. Outage signaled with a 404 response in one of the BaseURL.

http://livesim.dashif.org/livesim/baseurl_d40_u20/baseurl_u40_d20/testpic_2s/Manifest.mpd

Feature description 15

▸ ## MPD callback events

DASH Callback events are indications in the content that it is expected by a DASH client to issue an HTTP GET request to a given URL and ignore the HTTP response.

Callback event is signalled in the MPD by adding a EventStream element at the period level.

Ex: This creates 60 periods per hour, out of which 30 periods per hour have mpdcallback eventstreams at the period level.

http://livesim.dashif.org/livesim/periods_60/mpdcallback_30/testpic_2s/Manifest.mpd

Feature description 16

▸ Segment chunk support

The support is available for test vectors which expose low latency chunking. This stream has 8s segments with 1s chunks:

https://livesim.dashif.org/livesim-chunked/chunkdur_1/ato_7/testpic4_8s/Manifest300.mpd

It is possible to set the availabilityTimeOffset and chunkdur to floating point numbers like chunkdur_0.5/ato_7.5/ in the URL. The original content is 30Hz, so setting ato to 7.9 and chunkdur to 0.1 is also possible.

Chunks are always regenerated as independent moof + mdat pairs.

This is a branch of livesim running at a separate URL. It is always returning content using HTTP Chunked Transfer Encoding.

▸ Thumbnail support

DASH-IF has defined a way of providing thumbnails for scrub bars as an AdaptationSet with jpeg images. This is handled as an extra media type and is looked up and mapped to the wrap-around time.

http://livesim.dashif.org/livesim/testpic_2s/Manifest_thumbs.mpd

Feature description 18

‣ Start-over support

The possibility to define a finite time interval where the end may be in the future. For easy testing, there is a the possibility to specify relative times which are translated into absolute times and included in a <Location> element.

This is supported for both Number and SegmentTimeline

https://livesim.dashif.org/livesim/segtimeline_1/
startrel_-20/stoprel_20/timeoffset_0/testpic_2s/Manifest.mpd

https://livesim.dashif.org/livesim/
startrel_-20/stoprel_20/timeoffset_0/testpic_2s/Manifest.mpd

Feature description 19

‣ Trickplay support

Extra AdaptationSet where each segment just have one frame with a duration of the full segment. Signaled by essentialProperty.

http://livesim.dashif.org/livesim/testpic_2s/Manifest_trickmode.mpd

Feature description 20

▸ Addition of signaling for Segment Loss Timeline

- When next segment generation is not possible, add *emsg* to last generated segment. The MPD validity expiration is set to the duration of the current segment or smaller. Only after the Representation(s) under loss resumes, a new S element is written with S@t matching the earliest presentation time of the newly generated Segment (IOP section 4.11 Deployment scenarios)

- Github issue: https://github.com/Dash-Industry-Forum/Test-Vectors/issues/100
- URL sample
  - *livesim-dev/baseurl_u10_d20/segtimeline_1/segtimelineloss_1/testpic_2s/Manifest.mpd*
- Scenario is illustrated with live test vector MPD snapshots at different time intervals (next slides).

Feature description 20

Segment loss scenario - example

▸ Step 1: Segments available, new segments are added in this interval to *SegmentTimeline*.

```xml
<BaseURL>http://localhost:8059/livesim-dev/segtimeline_1/segtimelineloss_1/tsbd_10/baseurl_u10_d20/testpic_2s/</BaseURL>
<Period id="p0" start="PT0S">
- <AdaptationSet contentType="audio" lang="eng" mimeType="audio/mp4" segmentAlignment="true" startWithSAP="1">
    <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" />
  - <SegmentTemplate initialization="$RepresentationID$/init.mp4" media="$RepresentationID$/t$Time$.m4s" timescale="48000">
    - <SegmentTimeline>
        <S d="95232" t="74788703520768" />
        <S d="96256" r="2" />
        <S d="95232" />
      </SegmentTimeline>
    </SegmentTemplate>
```

▸ Step 2 : When segments becomes unavailable, *emsg* is added to segment and *InbandEventStream* signaled in MPD.

```xml
<BaseURL>http://localhost:8059/livesim-dev/segtimeline_1/segtimelineloss_1/tsbd_10/baseurl_u10_d20/testpic_2s/</BaseURL>
<Period id="p0" start="PT0S">
- <AdaptationSet contentType="audio" lang="eng" mimeType="audio/mp4" segmentAlignment="true" startWithSAP="1">
    <InbandEventStream schemeIdUri="urn:mpeg:dash:event:2012" value="1" />
    <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" />
  - <SegmentTemplate initialization="$RepresentationID$/init.mp4" media="$RepresentationID$/t$Time$.m4s" timescale="48000">
    - <SegmentTimeline>
        <S d="96256" r="2" t="74788704000000" />
        <S d="95232" />
      </SegmentTimeline>
    </SegmentTemplate>
```

Segment loss scenario - example

▸ Step 3 : *SegmentTimeline* is kept as is and no new *S* element added

```
<BaseURL>http://localhost:8059/livesim-dev/segtimeline_1/segtimelineloss_1/tsbd_10/baseurl_u10_d20/testpic_2s/</BaseURL>
<Period id="p0" start="PT0S">
- <AdaptationSet contentType="audio" lang="eng" mimeType="audio/mp4" segmentAlignment="true" startWithSAP="1">
    <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" />
  - <SegmentTemplate initialization="$RepresentationID$/init.mp4" media="$RepresentationID$/t$Time$.m4s" timescale="48000">
    - <SegmentTimeline>
        <S d="96256" r="2" t="74788704000000" />
        <S d="95232" />
      </SegmentTimeline>
    </SegmentTemplate>
```

▸ Step 4 : New *S* elements added when segments become available again, with their earliest presentation time.

```
<BaseURL>http://localhost:8059/livesim-dev/segtimeline_1/segtimelineloss_1/tsbd_10/baseurl_u10_d20/testpic_2s/</BaseURL>
<Period id="p0" start="PT0S">
- <AdaptationSet contentType="audio" lang="eng" mimeType="audio/mp4" segmentAlignment="true" startWithSAP="1">
    <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main" />
  - <SegmentTemplate initialization="$RepresentationID$/init.mp4" media="$RepresentationID$/t$Time$.m4s" timescale="48000">
    - <SegmentTimeline>
        <S d="96256" t="74788704960512" />
        <S d="95232" />
        <S d="96256" r="2" />
      </SegmentTimeline>
    </SegmentTemplate>
```

Feature description 21

▸ Test vectors supporting UTC timing schemes

- Test vectors support for UTC timing with schemes ntp, sntp, head, direct were available. Newly http-xsdate, and http-iso are provided.
- Github issue :https://github.com/Dash-Industry-Forum/Test-Vectors/issues/96

```
−<AdaptationSet contentType="video" maxFrameRate="60/2" maxHeight="360" maxWidth="640"
   mimeType="video/mp4" minHeight="360" minWidth="640" par="16:9" segmentAlignment="true"
   startWithSAP="1">
     <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
     <SegmentTemplate duration="2" initialization="$RepresentationID$/init.mp4"
     media="$RepresentationID$/$Number$.m4s" startNumber="0"/>
     <Representation bandwidth="300000" codecs="avc1.64001e" frameRate="60/2" height="360"
     id="V300" sar="1:1" width="640"/>
   </AdaptationSet>
 </Period>
<UTCTiming schemeIdUri="urn:mpeg:dash:utc:http-iso:2014" value="http://time.akamai.com/?iso"/>
</MPD>
```

```
−<AdaptationSet contentType="video" maxFrameRate="60/2" maxHeight="360" maxWidth="640"
   mimeType="video/mp4" minHeight="360" minWidth="640" par="16:9" segmentAlignment="true"
   startWithSAP="1">
     <Role schemeIdUri="urn:mpeg:dash:role:2011" value="main"/>
     <SegmentTemplate duration="2" initialization="$RepresentationID$/init.mp4"
     media="$RepresentationID$/$Number$.m4s" startNumber="0"/>
     <Representation bandwidth="300000" codecs="avc1.64001e" frameRate="60/2" height="360"
     id="V300" sar="1:1" width="640"/>
   </AdaptationSet>
 </Period>
<UTCTiming schemeIdUri="urn:mpeg:dash:utc:http-xsdate:2014" value="http://time.akamai.com/?iso"/>
</MPD>
```

Feature description 21

▸ UTC timing schemes open issue: For the remaining scheme (http-ntp), no time server found supporting this. The 'ntp' format is already supported

Some Shortcomings

▸ publishTime is not only set when the MPD would change
  - Essentially it is the time of the request

▸ Rather hard to use own content
  - Conditions may not be easy to meet
  - There is a preprocessor, but still a little tricky
  - Only segments with average duration whole seconds
  - The audio and video being looped must both have durations which are an integral number of seconds

▸ Not leap-second aware

▸ Probably quite a few more…