CHANGE REQUEST

DASH-IF IOP	CR		rev	-	Current version:	V4.3

Status:		Draft		Internal Review	X	Community Review		Agreed

Title:	Low-latency Modes for DASH
Source:	Live TF
Supporting Companies:	Akamai, castLabs, Comcast, Elemental Technologies, Ericsson, Harmonic, Qualcomm Incorporated, Sony, TNO, Unified Streaming, Frontier Communications
Category:	**A** ***Date:*** 2019-06-28 *Use one of the following categories:* ***C*** *(correction)* ***A*** *(addition of feature)* ***B*** *(editorial modification)*

Reason for change:	DASH-IF collected information related to Low-Latency Streaming in a Report, together with DVB. The report (available here: https://dash-industry-forum.github.io/docs/Report%20on%20Low%20Latency%20DASH.pdf) provides use cases, service scenarios, deployment experience and existing potential technologies. Also, DASH-IF already generated the DASH profile for ATSC that includes a mode supporting low latency. Based on this information in the report, it is recommended to add low latency to DASH-IF IOP Guidelines.
Summary of change:	This change provides a new clause for live services that addresses specification updates as well as implementation guidelines to support Low-Latency DASH services addressing the requirements above.
Consequences if not approved:	Low-Latency DASH not available

Sections affected:	References in clause 2 New section 4.X
Other comments:	This document contains several notes highlighted in yellow. Feedback during community review is welcome specifically on these topics.

Disclaimer:	This document is not yet final. It is provided for public review until the deadline mentioned below. If you have comments on the document, please submit comments by one of the following means: - at the github repository https://github.com/Dash-Industry-Forum/Live/issues (public at https://gitreports.com/issue/haudiobe/Live), or - dashif+iop@groupspaces.com with a subject tag [LL-DASH] Please add a detailed description of the problem and the comment. Based on the received comments a final document will be published latest by the expected publication date below, integrated in a new version of DASH-IF IOP if the following additional criteria are fulfilled: - All comments from community review are addressed - The relevant aspects for the Conformance Software are provided - Verified IOP test vectors are provided
Commenting Deadline:	July 31st, 2019
Expected Publication:	August 31st, 2019

References

[X1] DASH-IF IMPLEMENTATION GUIDELINES: TOKEN-BASED ACCESS CONTROL FOR DASH (TAC) V1.0: http://dashif.org/wp-content/uploads/2016/07/DASH-TAC-v0.9.pdf

[X2] ISO/IEC 23009-1:2020: Draft of ISO/IEC 23009-1:2014 4th edition (available in w18416)

[X3] DASH-IF/DVB Report on Low-Latency Live Service with DASH, July 2017, available here: https://dash-industry-forum.github.io/docs/Report%20on%20Low%20Latency%20DASH.pdf

[X4] DASH Ingest Protocol, available for community review here: https://dashif-documents.azurewebsites.net/Ingest/master/DASH-IF-Ingest.html

[X5] CTA-5003: "CTA WAVE Device Playback Capabilities Specification (CTA-5003)", available here: https://cta.tech/cta/media/EventImages/TechStandards/CTA-5003-Final.pdf

[X6] Draft of ISO/IEC 23009-1:2014 4th edition, available as N18416 at http://wg11.sc29.org/doc_end_user/documents/126_Geneva/wg11/w18416.zip

[X7] DVB-DASH BlueBook A168, "MPEG-DASH Profile for Transport of ISO BMFF Based DVB Services over IP Based Networks", February 2019, https://www.dvb.org/resources/public/standards/a168_dvb_mpeg-dash_feb_2019.pdf

4.X Low Latency DASH

4.X.0 Definition

Based on a report developed jointly between DVB and DASH-IF on Low-Latency DASH [X3], this clause defines details on how to support consistent latency in DASH for linear TV services. The Low Latency DASH scenario is a variant of the Live Services recommended approach focused on ensuring that the end-to-end latency of the streams will be comparable to the broadcast latency. This broadcast latency is not a universal value, as it is influenced by many factors such as the duration of the broadcast encoding pipeline, the latency of the transport channel which can slightly differ per type (satellite, cable, IPTV, DTT...), or the artificial delays introduced by local content moderation regulations. However, most of the measurements converge on a 3 - 10 seconds latency between the moment where the source signal is acquired for encoding and the moment when it's played back on the TVs. For details refer to [X3].

4.X.1 Scenario and Architecture

Depending on the position of the OTT content preparation workflow in the global media workflow, upstream or downstream, the value of the ideal latency is variable. If the OTT encoders are positioned upstream in the chain, near the broadcast encoders, then a 6 seconds value might work well. If the encoders are positioned downstream, after the broadcast encoders or worst case after a first broadcast delivery layer, then the ideal value should be decreased by the duration of the broadcast workflow latency at the signal acquisition (SDI signal case) or ingest (IP case) time. In order to achieve some of the target service scenarios, the end-to-end latency must be stable and predictive, meaning that the latency introduced by each component in the architecture should be measurable and controllable. On top of the end-to-end latency value, the Low Latency DASH scenario should ensure that the switch latency is comparable to the one of a broadcast service, typically a second, and that the random access delay (result of a seek operation in the media timeline) stays at reasonable level, one second also.

In the typical workflow presented in Figure 1, the source signal is encoded in several resolutions/bitrates by the ABR encoder, which produces fragmented MP4 or MP4/CMAF compliant elementary streams and eventually encrypts them. Those streams are then ingested onto the packager using protocol suitable protocol to deliver the data in streams/chunks. The packager proceeds with the necessary operations such as the transformation of the incoming streams' segments duration, DRM preparation of the streams, and the generation of the DASH manifests. Typically, the resulting set of files is made available on a disk or RAM storage to the origin server in such a way that the origin server can load the chunks while they are packaged by the packager.

In parallel, the packager needs to generate the MPD and the DASH clients are requesting the DASH MPD and leverage it to calculate the live edge time of the stream:

- when a regular DASH Client does the calculation, it defines which segment is supposed to be completely available on the Origin Server and requests it from the CDN. Every media segment is fully fetched by the Client before being added to the playback buffer.
- when a Low Latency DASH Client does the calculation, it should do the same calculation but request the next segment when its first chunk will start to be available on the origin. This time can be calculated by leveraging the `@availabilityTimeOffset` parameter from the DASH manifest. Every media segment can be added to the playback buffer in one or several steps, as the CDN is delivering the Live Edge Segment in HTTP chunked mode.

In both cases, once the playback buffer reaches the minBufferTime amount expected by the client through a single request or series of requests for media segments, the playback can start. The decryption and decoding processes are identical on both Client types, with the difference that their scope can be a full segment or a partial segment.

When the request for a DASH manifest comes to the CDN, it can leverage an intermediate manifest manipulation service instead of requesting the manifest straight from the Origin Server. This covers use cases where the manifest has to be customized for each end-user or group of end-users, with such manipulations as ad insertion, bitrate filtering, A/B watermarking or content replacement for blackout use cases. The manifest manipulation operations should not significantly impact the end-to-end latency, at least not more than the duration of a single media segment.

Requests for media segments are all forwarded to the Origin Server. Ideally, all requests for all stream components (manifests and media segments) should be done in HTTPS on the DASH Client side, and forwarded in HTTPS to all downstream services (Manifest manipulation and Origin services), in order to avoid protocol downgrade from an initial request in HTTPS to downstream requests in plain HTTP. In order to protect DRM services, all requests should be sent by the DASH Client to the CDN, possibly using a tokenization mechanism such as the one defined in the TAC specification [XX], so that no illegitimate do not reach the DRM License server during the playback session.

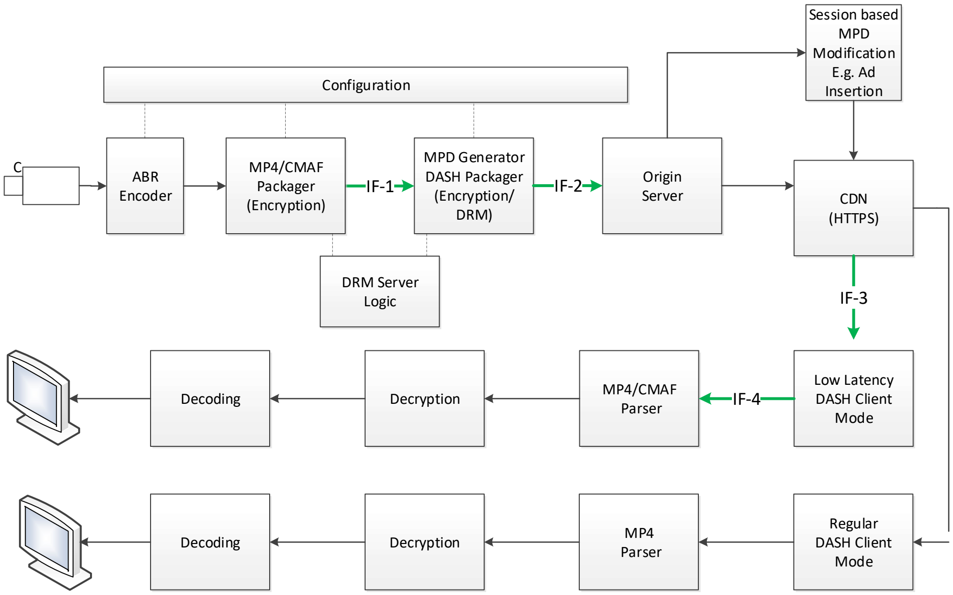


Figure 1: DASH-IF Low-Latency Architecture

When chunked MP4/CMAF is used in the Low Latency DASH scenario to enable such latency levels to be reached, the stream structure shall be backward compatible with DASH players that rely on the reception of a full media segment before the playback of the stream can start. But HTTP chunked transfer encoding shall at least be supported up from the ingest into the packager up to the CDN edge, whereas the last mile delivery shall happen using HTTP chunked transfer encoding or HTTP in regular mode. If HTTP chunked transfer encoding is supported by the DASH player, it basically means that a media segment carrying the latest moment of the program (also known as the "live edge time") could be consumed on the player while it's still being produced by the encoder and the packager.

This specification strongly aligns with the DVB-DASH specification (including the low-latency extensions) [X7] and joint development and design meetings were held between DVB and DASH-IF to ensure industry alignment.

A brief summary of how this extended DASH specification for low-latency addresses some of the issues is provided in the following:

- This specification extension is fully aligned with earlier DASH specifications and basically no changes are done to core formats MPD and Segments. This specification only provides sufficient guidelines for a service provider and client implementor to make use of additional functionalities and signaling to support low-latency. Legacy clients are expected to still be able to use the service, just at some higher latency.
- This specification makes full use of the ubiquitously available HTTP/1.1, and modern CDN architectures are therefore fully usable, possibly with some updates on handling the relay of partially complete Segments through the network. This specification recommends the use of HTTP Chunked Transfer encoding for reduced latency which is a mandatory client part of HTTP/1.1. Secondly, the low-latency content is fully cacheable for consumption in time-shift and network-PVR mode without modification.
- This specification enables CDN-cache-friendly and efficient operation as the Segments that are stored on the CDN for later consumption are of typical Segment duration ranges of several seconds.
- This specification can benefit from improved network protocols, such as HTTP/2 or HTTP/3, but does not require any of those.
- This specification relies on existing MPD compactness based on segment templates using the same approach as for regular DASH live services, i.e. no changes on MPD processing and generation is required.
- This specification relies on the existing DASH timing model and asynchronous client-server time synchronization. By this, a DASH client once synced to the server has full knowledge, when a Segment was published and when the next segment will be published without requiring more frequent MPD updates than done for a regular DASH service. This knowledge can be used be the DASH client to request the proper Segment at startup and at the right time (i.e. when it gets available) and also maintain consistent end-to-end latency.
- This specification also supports fast random access without any MPD changes. Random access in the middle of Segments is indicated in the MPD as well as in Segments. In addition, fast joining may be supported by the combination of knowledge on segment availability and accelerated decoding.

- The specification fully relies on technologies defined in MPEG in updates for ISO/IEC 23009-1 (MPEG-DASH) and ISO/IEC 23000-19 (CMAF).

In the same timeline as this specification was developed and published, another CMAF based streaming specification[1] was published to address low-latency and was analyzed here[2], raising some aspects discussed above.

4.X.2 Assumptions

The following assumptions for the service and the content are applied

1. The distribution latency and random access is following the considerations in of the scenario in clause 4.x.1. The following service configuration parameters are assumed:

    a. Desired end-to-end latency

    b. Desired random-access times

    c. Constraints on the maximum and minimum playout speed of the content

2. Program events may occur for different reasons, for example Program changes, switches from Programs to Advertisements or vice versa, media blackouts or other program changes. Such changes are typically anticipated only on short notice, i.e. within a few seconds. In the following we refer to the time that changes are announced as *change lead time*. The service should also provide a *minimum change lead time*, i.e. the smallest time in media time between the change being announced in the stream and the time between the change occurs. Changes may for example include one or more of the following:

    a. Ad Insertion opportunity

    b. Number of source audio languages or formats can change. For example:

        i. Programming with English and Spanish to other content with only English

        ii. Descriptive audio may disappear / reappear

        iii. Programming with 5.1 E-AC-3 and AAC Stereo content to other content with only Stereo AAC

    c. Resolution or format of source video content can change, e.g. HD to/from SD, HDR to/from SDR, etc.

    d. Codecs may change, or at least the profile or level of the codecs

    e. The number of Representations in an Adaptation Set may change

    f. A distribution network may be changed, added or removed.

    As an example, the media stream from the broadcast origination points if MPEG-2 TS is used, then the Program Map Table (PMT) typically indicates changes such changes. Typically, these changes also result in discontinuities in the media timeline.

    Another option are MP4/CMAF encoded tracks that may for example include metadata tracks or event messages for additional information.

    When program changes happen, it is assumed that the content is properly conditioned.

    Note: Details on conditioning are currently developed as part of the Ad Insertion and Multi-Period content and will be referenced in the final version of this document. Pre-dominantly, this addresses that the content has no gaps and that period continuity connectivity is signalled across period boundaries.

3. The IDR frame placement is determined by the ABR encoders and the IDR frame placement also predominantly determines random access delay and the possible switching frequency. This encoding may result in occasional slight variations in IDR frame placements during a period (as compared to the last segment

[1] https://developer.apple.com/documentation/http_live_streaming/protocol_extension_for_low-latency_hls_preliminary_specification

[2] https://www.linkedin.com/pulse/apples-implementation-low-latency-hls-explained-phil-harrison/

in a period) due to encoding optimizations around scene changes. It is expected that the ABR encoder has knowledge about desired random-access times and desired switching granularity, but the ABR encoder may also make decisions for exact placement of IDR frames based on encoding optimizations. Some guidelines on the operation of the encoder are provided in this document.

4. Unanticipated losses and operational failures or outages, possibly happen just for a single encoding (typically at the input of the encoder, but also possibly also downstream packaging). Examples are

    a. An encoder for one or more Representations or the output of an encoder fails for some time and does not produce content.

    b. An encoder or the input to the encoder or the output of the encoder fails for a media component/Adaptation Set for some time and do not produce content.

    c. All encoding or the input to the encoder fails for some time.

    In all cases an MPD can still be produced and the MPD server is up and running.

    Also in the distribution, single Segments may be lost for different reasons.

5. MPD updates should be minimizeded

6. The content may be encrypted

7. The content may be received by Regular DASH clients in live mode or may be consumed and/or the generated content may be converted to a VoD asset without changing the encoding.

8. The DASH client is aware that content is offered in low-latency mode and can apply appropriate actions. In addition, the DASH client may be provided the ability to measure the experienced end-to-end latency.

4.X.3 DASH Service Configuration for Low-Latency Modes

4.X.3.1 Introduction

This clause provides basic requirements and recommendations on how to signal and configure a low latency mode. Detailed implementation guidelines for service offering and clients are provided in clause 4.X.4 and 4.X.5, respectively.

Figure 1 provides a basic flow of information for operating a low-latency DASH service. The DASH packager gets information on the general description of the service as well as the encoder configuration. The encoder produces CMAF chunks and fragments. The chunks are mapped by the MPD packager onto Segments and provided to the network in incremental fashion. Segments are not delivered as a whole, but progressively as they are generated.

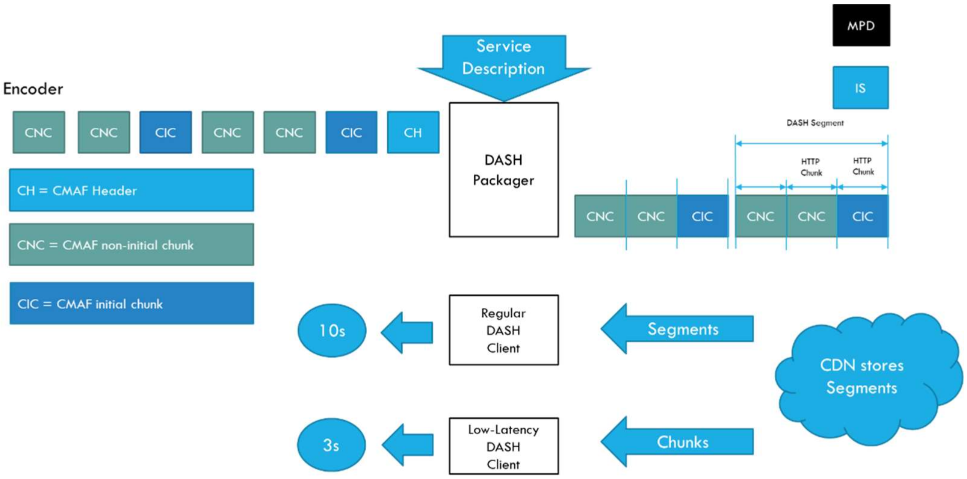


Figure 1 Basic operation flow Low-Latency DASH

The key technologies for LL-DASH are briefly summarized in the following. Detailed requirements for service offering and clients are provided in the remainder of clause 4.X.3.

- The encoding is based on fragmented ISO BMFF files and typically CMAF Fragments and CMAF chunks are assumed.
- Each chunk is individually accessible be the DASH packager and gets mapped to an HTTP chunk that is uploaded to an origin server. This 1-to-1 mapping is a recommendation for low-latency operation, but not a requirement. By no means the client should assume that this 1-to-1 mapping is preserved to the client.
- A low delay protocol, e.g. HTTP Chunked Transfer Encoding, of partially available Segments is used such that clients can access the Segments before they are completed. The availability start time is adjusted for clients being able to make use of this feature.
- Two operation modes are permitted
    - Simple live offering is used by applying `@duration` signaling and `$Number$` based templating
    - Main live offering with the **`SegmentTimeline`** as either `$Number$` or `$Time$` is supported by the proposed updates in DASH fourth edition.
- MPD validity expiration events are encouraged to be used but are not essential for clients to be understood.
- Generally, Inband event messages may be present, but clients are only expected to recover those at the start of Segments, not at arbitrary chunks. DASH packagers however may receive notifications from the encoder at chunk boundaries or completely asynchronously using timed metadata tracks. For details refer to the DASH-IF ingest protocol [X4].
- It is permitted in a single Media Presentation and within one Period of a Media Presentation to have Adaptation Sets that are using the chunked low-latency mode and Adaptation Sets using short segments for different media types.
- A certain amount of playback control of DASH client on the media pipeline may be available and should be used for robustness of DASH clients, for example the playback may be accelerated or decelerated for some period of time, or may do a seek into Segments.
- The system is designed to be workable with standard HTTP/1.1, but also should be applicable to HTTP extensions and other protocols for improved low-latency operation.
- The MPD includes explicit signaling on the service configuration as well as the service properties (for example including the target latency of the service).
- MPDs and possibly also Segments include anchor times that allow DASH clients to measure the current latency compared to live and adjust to meet the the service expectations.
- Operational robustness is addressed, for example in case of encoder failures.
- Existing DRM and encryption modes are compatible with proposed low-latency operation.

Note: Compatibility with the ad insertion is checked as part of the community review for the updates to DASH Ad Insertion as well as this document.

4.X.3.2 Service Offering Requirements and Recommendations

Based on the discussion above, for a DASH client to suitably operate in low-latency or at least in controlled latency mode, this clause provides requirements and recommendations for service offerings:

Either by signalling on application level, or by signalling in the MPD using the **`ServiceDescription`** element, sufficient signalling shall be provided such that the DASH client can measure the observed latency.

The service offering shall conform to one of the DASH-IF IOP Live profile according to clause 4.8. In addition, the following applies for low-latency services in terms of presence and settings of certain parameters.

The following information is present on MPD or Period level:

- At least one **`ServiceDescription`** element shall be present as follows:
    - **`Scope`**: Specifies the scope of the service description element. If present, this service description only targets DASH clients within the scope of this descriptor. DASH clients not in scope, i.e. not recognizing any of the scope descriptor elements, are expected to ignore this service description.
        - **`Note`**: Please provide feedback on the suitability of a DASH-IF defined scope, possible referring to the below client requirements.
    - A **`Latency`** element shall be present including at least one of the following parameters:
        - `TargetLatency`: The service provider's preferred presentation latency in milliseconds compared to the producer reference time.
        - `MaximumLatency`: The service provider's indication about the maximum presentation latency in milliseconds. Indicates a content provider's desire for the content not to be presented if the latency exceeds the maximum latency.

    - `MinimumLatency`: The service provider's indication about minimum presentation latency in milliseconds for example to avoid inconsistencies with second screen applications, overlays, etc.
  - A **`PlaybackSpeed`** element may present to provide a permission on the boundaries for this service. If present, it shall contain at least
    - `maxPlaybackRate`: The maximum playback speed that the content provider indicates is appropriate for the purposes of automatically adjusting playback latency and buffer occupancy during normal playback.
    - `minPlaybackRate`: The minimum playback speed that the content provider indicates is appropriate for the purposes of automatically adjusting playback latency and buffer occupancy during normal playback.
  - Other service description parameters may be present, for details refer DASH Amd.
    - Note: May be moved to generic part
- **`UTCTiming`**
  - At least one UTC timing description shall be present and be restricted to one of the followings `@schemeIdUri` set to one of the following:
    - `urn:mpeg:dash:utc:http-xsdate:2014`
    - `urn:mpeg:dash:utc:http-iso:2014`
    - `urn:mpeg:dash:utc:http-ntp:2014`

If the packager is expected to operate on controlled latency based on the above signaling, it shall follow the service offering requirements and recommendations of the main live service in clause 4.5, with the following additional requirements and recommendations:

1. Adaptation Sets that are expected to be consumed in low latency modes should include at least one **`ProducerReferenceTime`** element as follows
   a. `@id`: provide a unique id for this reference
   b. `@type`: either `encoder` or `captured`
   c. A **`UTCTiming`** element that is identical to one present in the MPD shall be present and is used for deriving the value of the `@wallclockTime`.
   d. `@wallclockTime`: shall be present and provide the value at the `@presentationTime`
   e. `@presentationTime`: shall be the value of the `@presentationTimeOffset`, if present, or 0 otherwise. Note that this means that the producer reference time is provided at the start of the Period.

   If the **`ProducerReferenceTime`** element is not present, then the Period Start time is assumed to be used as the wall-clock time and the value of the `@presentationTimeOffset` is assumed as the presentation time.

2. For each Representation delivered in low-latency mode using CMAF chunking (possibly by defaulting on Adaptation Set level),
   a. the **`SegmentTemplate`**`@duration` or **`SegmentTimeline`** attribute shall be present.
   b. the **`SegmentTemplate`**`@media` shall be present with either the `$Number$` or `$Time$` template included.
   c. the nominal duration of a Segment should at least be 960 ms.
   d. If the Adaptation Set contains more than one Representation for switching bitrates, then the maximum segment duration should be smaller than the anticipated client buffer size. An upper bound to the client buffer size is the target latency. Hence, the maximum segment duration shall be smaller than the signaled target latency.
      i. Note: This requirement may be relaxed, but in this case, guidelines need to be provided for clients how to downswitch in case of longer segment durations.
   e. the **`Representation`**`@availabilityTimeOffset` shall be present with the following constraints
      i. the value is set to at most Segment Duration minus the duration of the first chunk included in the Segment.
      ii. the value shall be greater than zero
   f. the **`Representation`**`@availabilityComplete` shall be present and set to `'FALSE'`.
3. Adaptation Sets with short duration Segments with segment availability times in the same range as the adjusted segment availability times of the low-latency modes may be used together with low-latency Representations, for example video may run with `@availabilityTimeOffset` being set and audio without.

4. Event Messages may be used if in low-latency media presentations. If Adaptation Sets with short duration Segments are present, they may preferably be used to carry Event messages. If chunked Segments are used any 'emsg' box shall be placed as follows:
    a. It may be placed in between any `'mdat'` and `'moof'` box or before the first `'moof'` box
    b. However, if not placed before the first `'moof'` box, an equivalent `'emsg'` with the same id value shall be present before the first `'moof'` box of the next Segment.
5. The service provider should provide an Inband Event Stream carrying MPD validity expiration events as defined in clause 4.5. If used, `@value` shall be set to 1.
6. The Segment durations for one Representation should not vary significantly. In particular, the segment duration shall not vary more than indicated in the MPD applying the rules defined in clause 4.11.3.7.
7. The Segments may contain multiple movie fragments per segments. If CMAF chunks are signaled using the styp, then CMAF chunks shall be generated such that the range of presentation times contained in any CMAF chunk of the CMAF Track should not overlap with the range of presentation times in any other CMAF chunk of the same CMAF Track.
    a. Note: no explicit signaling is defined. Please provide comments if you believe this would be important for players.
8. The Segments may allow to randomly access the Representation not only at the Segment boundardy, but also within the Segment. If such Random Access is provided, this should be signaled in the MPD.
    a. Note: This is currently discussed in MPEG and an updated is expected to be provided in the final version.
9. Operational robustness is expected to be provided by the means documented in clause 4.8.
10. For encrypted content, the rules documented in clause 7 apply. Additional guidelines are provided below.

4.X.3.3 Client Requirements and Recommendations

For a client to consume a low-latency live services, the following applies:

1. DASH clients shall compute the presentation latency within each Period based on a wall clock anchor `WCA` and a presentation time anchor `PTA`. `WCA` and `PTA` are determined as follows:
    a. If the **`ProducerReferenceTime`** element is present as defined in clause 4.X.3.2, then the
        i. `WCA` is the value of the `@wallClockTime`
        ii. `PTA` is the value of the `@presentationTime`
        iii. If the `@inband` attribute is set to `TRUE`, then it should parse the segments to continuously update PTA and WCA accordingly
    b. Else
        i. `WCA` is the value of the `PeriodStart`
        ii. `PTA` is the value of the `@presentationTimeOffset`
    c. Then the presentation latency `PL` of a presentation time `PT` presented at wall clock time WC is determined as `PL = (WC – WCA) - (PT – PTA)`.

    Note: we expect to add a diagram to explain this relation. Input welcome.

2. It shall support service offering according to 4.X.3.2. This includes
    a. support media segments that contain more than one pair of `'moof'` and `'mdat'` boxes, where each moof/mdat pair may contain any number of ISO BMFF samples between 1 and the full segment duration inclusive.
    b. Support playing two or more Adaptation Sets for which the Segments of one Adaptation Set do not align with the segments of another (e.g. due to differing segment durations) or the segments of one Adaptation Set contain multiple moof/mdat pairs and the segments of another only have a single moof/mdat pair.
3. It shall implement means to support the service description functionality (either in the DASH client signaled through MPD or by the means of an API). The parameter interpretation is as follows:
    a. If the target latency is set, the client when consuming in live mode, it should play the content within 500ms tolerance of the described target latency. However, the client should consider meeting the latency target also by taking into acccount knowledge of its own capabilities, the network conditions, and any relevant knowledge of past streaming performance.
    b. If the max latency is set, the client when consuming in live mode, it should not exceed the described max latency even if it observes rebuffering. In this it is preferred to terminate the service.
    c. If the min latency is set, the client shall not fall below the described min latency.

   d. If the playback speed element is present, clients should use these tools to adapt to the target latency but should not exceed the playback speed range.

4. The player should implement an appropriate joining algorithm to support fast enough random access and at the same time meet the latency constraints. The following aspects should be considered:
   a. If Random Access within Segments is signalled, then client should use such random access points in order to join within the joining time as well as to keep the client in the target latency.
   b. If no such Random Access points are present, or if they are too spread to not allow to fulfill joining and latency requiremrents, the client is recommended, depending on the wall-clock time to
      i. Either download the latest available Segment from the beginning, but then do seek or an accelerated decoding to reach the time. Note that this may need a client that supports faster than real-time decoding.
      ii. Or to wait for the next segment to appear if this is expected to happen within the random access bounds and then request and start decoding the new Segment. If the latency is too short and the client supports decelerated decoding, it may apply this to reach the target latency.
   c. Suitable combination of the above.
5. Clients should access Segments at adjusted segment availability times (i.e. compensated by `@availabilityTimeOffset`). In addition, clients should request the next segment after the last HTTP chunk is received.
6. Clients should follows the Content Protection relevant recommendations and guidelines as provided in clause X.
7. Clients should implement an appropriate bandwidth estimation when requesting segments that are not yet complete at the time when requested. For more details, refer to clause 4.X.5.
8. Clients should implement appropriate failover mechanisms to resume to the target latency. For more details, refer to clause 4.X.5.

Note: Appropriate requirements from the DVB LL-DASH specification will be checked and may be added to the final version. Comments welcome.

4.X.4 Guidelines for Low-Latency Service Offering (Informative)

4.X.4.1 Introduction

This clause provides basic considerations on how the operation of a service provider should work. It is not meant to provide a normative implementation but provides a reference implementation as well as a set of guidelines on how to operate a head for low-latency distribution.

Note: This clause is not yet complete and additional information will be added during and after community review.

4.X.4.2 ABR Encoder, Encryption and MP4/CMAF Packaging

4.X.4.2.1 General

This clause provides best practices on how to encode and encrypt content for low-latency DASH streaming. In quite many circumstances, the ABR encoder is provided information by a contribution link that itself has encoded material and metadata included. Despite such contribution encoders and links are not a matter of this document, the following is recommended for contribution encoders and links:

- They should avoid adding unnecessary latency
- Ad insertion markers and other messages requiring changes in the encoding configuration should be provided in advance to the actual change
- The encoding should avoid providing virtual segmentation, this should be the duty of the ABR encoder. If provided and important, this virtual segmentation should be consistent in terms of segment durations.
- Regular encoding patterns such as IDR frames should be signaled and be consistent.

For the ABR encoder generating CMAF chunks and CMAF fragments, the following is recommended:

- The ABR encoder should be aware of the MP4/CMAF logic for chunk generation.
- If there is a desire for running specific latencies and configuration, an ABR encoder encoder for low latency streaming should permit to configure the following parameters:
  - Maximum Fragment duration: i.e. the nominal interval of Switch Points. Note that this duration may be chosen from a set of of selected parameters and may depend on, for example the frame rate of content.

- Maximum Chunk duration: i.e. the nominal maximum duration. Note that this duration may be chosen from a set of of selected parameters and may depend on for example the frame rate of content.
- The maximum bitrate/size of a Fragment for this CMAF Track
- The maximum bitrate/size of a Chunk for this CMAF Track
- The maximum variation of fragment durations
    1. Signalling of constant segment duration using `@duration`, permitting a variation of `+/- max_variation` of the segment duration.
    2. For each media segment in each Representation the MPD start time of the segment should approximately be EPT - PTO.
    3. Specifically, the MPD start time of the segment is preferably be in the range of EPT - PTO – max_variation/100*DUR and EPT - PTO + max_variation*DUR, otherwise the DASH packager needs to add a new Period.

4.X.4.2.2 Encoding and CMAF Chunk Duration Recommendations

4.X.4.2.2.1 General

CMAF chunks is a way to reduce streaming latency without decreasing the IDR frame frequency and the DASH segment sizes. There have been demonstrations using chunk sizes of one video frame, but in the general case if for example typical efficient encoding configurations with B-frames are used, then creating chunks with duration of 1 sample per chunk is not desirable. This not only applies to video, but also audio and possibly subtitles, need to be taken intoconsideration. For practical use, it is important to find an appropriate chunk duration for all media types. This following text is a first set of recommendations on this matter.

One reason for using chunks instead of shorter Segments is that Segments must start with a SAP type 1 or 2, i.e. for video a closed GOP. This decreases the coding efficiency of video. CMAF Chunks do not have this restriction since they are neither generated for bitrate switching nor for randomly accessing the Representation.

The following is recommended

- CMAF chunks are generated such that the range of *decode* times contained in any CMAF chunk of the CMAF Track do not overlap with the range of *decode* times in any other CMAF chunk of the same CMAF Track.
- CMAF chunks are be generated such that the range of presentation times contained in any CMAF chunk of the CMAF Track should not overlap with the range of presentation times in any other CMAF chunk of the same CMAF Track. This has the following consequences.

The content authoring impacts all of the three, (i) achievable latency, (ii) switching granularity and (iii) startup delay. A content author should carefully offer content such that the client can meet the desired parameters provided in the service description.

- Note: In a revised version we need to provide some considerations on how RAP frequency, chunk size and segment size impact the switching, end-to-end latency as well as the start-up latency. Those desired values determine proper chunk size as well as proper fragment duration/RAP distance).

4.X.4.2.2.2 Video

Applying the above rules, to make each chunk displayable, without waiting for more data, it is important that all B-frames which should be displayed before the P-frame are included in the chunk.

As an example, hierarchical B-frames with the display order

I0 B1 B2 B3 P4 | B5 B6 B7 P8 | B9 B10 B11 P12 | B13 B14 B15 P16 |

have a send (decode) order (order in mdat) like

I0 P4 B1 B2 B3 | P8 B6 B5 B7 | P12 B10 B9 B10 | P16 B14 B13 B15 |

Thus, here the segment should be broken after a multiple of 4-5 frames (160-200ms for 25Hz video). The general recommendation is therefore to put chunk boundaries so that all display times (decode time + composition_time_offset) are before the display-time of the earliest display time of the next chunk.

Depending on the encoder, this is either a fixed or variable duration.

This recommendation results that during the process of packaging the encoded video when packed into CMAF chunks does not add further delay beyond that arising from frame re-ordering in the encoder, helping to make chunks available at the earliest moment that they could be.

4.X.4.2.2.3 Audio

For audio, one could make similar chunks as for video, but there is an issue that the audio bitrate is relatively low. For example, the size of an 200ms audio chunk of 64kbps is 1.6kB. That may possibly be too small to propagate through network or receiver buffers. For this reason, it may make sense to have longer chunks (e.g. 0.5s for audio) or even not applying chunking for audio but run at shorter Segment duration.

4.X.4.2.2.4 Subtitles

Subtitles is also low bitrate, but has different characteristics compared to audio. In particular, for IMSC-1 there is typically only one sample in each segment. However, that sample is a TTML XML document with relatively big boilerplate resulting in a size of roughly 2kB even though the actual information is just one short sentence or no text at all. For 2s segments, the bitrate will then be in the order of 10kbps. However, a chunk also needs to have one sample, so a chunk will have approximately the same size as an ordinary segment. If we would split a 2s subtitle segment into 10 chunks the bitrate would therefore increase from 10kbs to roughly 100kbps. Furthermore, it would increase the XML parsing in the client by a factor of 10. Since these effects are both undesirable, it is suggested that subtitles are not being chunked but delivered as separate segments every 1s or similar.

4.X.4.2.3 Producer Reference Time

The Producer Reference Time supplies times corresponding to the production of associated media. This information permits among others to (i) provide media clients with information to enable consumption and production to proceed at equivalent rates, thus avoiding possible buffer overflow or underflow, and (ii) enable measuring and potentially controlling the latency between the production of the media time and the playout.

The Producer Reference Time (`'prft'`) as defined in ISO/IEC 14496-12.

NOTE: Please refer to the latest updates in N18416 [X6]

The information may be provided inband as part of the Segments in the (`'prft'`), in the MPD or both. In the context of the low-latency DASH service offerings, providing information in the MPD is strongly recommended, whereas providing inband information is left to the deployment.

If the CMAF Switching Set contains (`'prft'`) information and the `flags` is set to 0 or the `flags` 8 and 16 are set, then this information may be exposed to the MPD by adding a Producer Reference Time element **`ProducerReferenceTime`** as follows:

- `@id` is set to a unique value in the contexnt of the Media Presentation
- `@inband` is set to `true`
- `@type` to `encoder` for flag set to 0 or to captured for both flag 8 and flag 16 being set.

Regardless, whether the inband information is present or not, it is recommended to provide information in the MPD for the `@wallclockTime` is and the `@presentationTime.`

Assume that a value wall-clock `WC` is known that corresponds to a presentation time `PT`, either by the availability of a pair of for `ntp_timestamp` and `media_time` as contained in a (`'prft'`) as defined in 8.16.5 of ISO/IEC 14496-12 or by other means. Also it is assumed that the value of the `@presentationTimeOffset` PTO is known. The MPD packager should act as follows:

- derive the wall-clock time WCA that corresponds to the PTO, namely `WCA = WC + (PT – PTO)`
- convert the `WCA` into the format of a **`UTCTiming`** element format present in the MPD
- add this **`UTCTiming`** element into the Producer Reference Time element **`ProducerReferenceTime`**
- add the value of PTO to the `@presentationTime`

- add the value of WCA to the `@wallclockTime`

Note that multiple producer reference time elements may be added.

4.X.4.2.4 Content Conditioning at Splice Points

Note: This aspect is currently developed as part of the Ad insertion task force. More details and references will be added in a revised version.

4.X.4.2.5 Robustness

For robustness, the guidelines in clause 4.8 apply. Also, please take into account the deployment scenarios in clause 4.11.

4.X.4.2.6 Encryption

For protected content, please take into account the considerations in clause 7.

Note: Extensions are currently developed for the next revision of the IOP guidelines that also apply for low-latency DASH.

4.X.4.2.7 Service description

Annex K of ISO/IEC 23009-1 defines the DASH Service Description. In the DASH model in ISO/IEC 23009-1, the DASH client has significant control over the algorithms and user perception for a DASH service. The DASH client may for example decide on the applied rate adaptation algorithm, the buffer strategy, the buffer duration and the resulting latency and channel access times. However, by leaving all decisions to the client, this may result inconsistent behavior as different client implementations may for example chose different strategies and therefore, as an example, one may observe significantly different latencies for the same service on different clients.

Hence, the Annex K defines the following a service description reference model for the client as shown below. The semantics of the service description and the associated keys to describe the service are provided in K.3. This is defined as an abstract set of APIs that can be used by the application, regardless on how the application received this information. The usage of the service description information may be left to client implementations, but it may also be the case that application standards formulate stronger requirements on the client in order to fulfil such service parameters. Service Descriptions can also be scoped for specific clients, for example those implementing specific rules, or clients in specific environments.

Figure X.X shows an extended client model that includes the ability to provide explicit service description information to the DASH access client. The information may origin from the service provider and may be delivered by application defined signalling, may be generated in the application or may be delivered by the MPD as defined in Annex K.4 of ISO/IEC 23009-1.

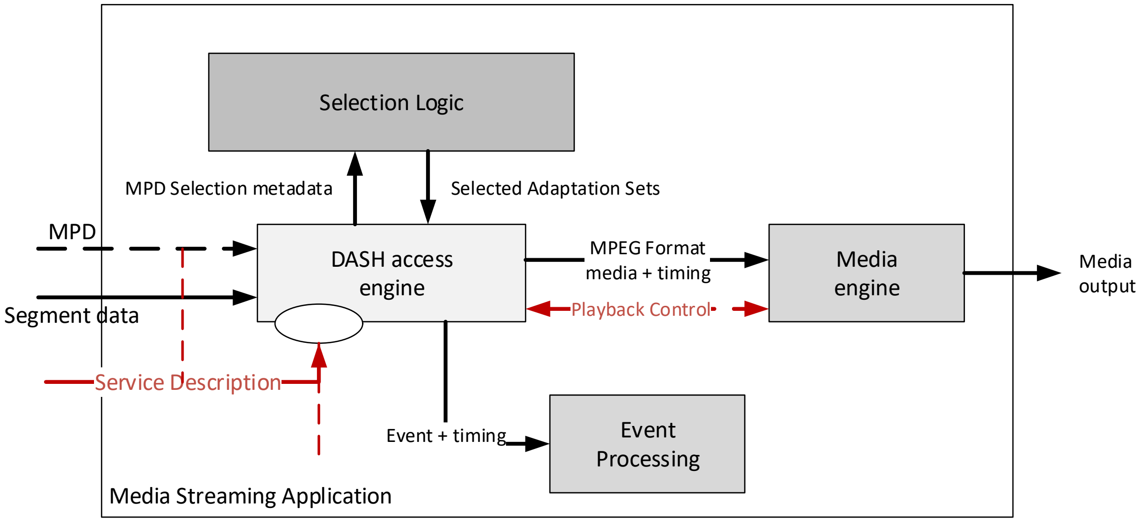


Figure X.X Extended Client Model with service description

In the context of LL-DASH applications, it is recommened that the MPD or the controlling application contains at least the following information with respect to latency targets for the client:

- TargetLatency: The service provider's preferred presentation latency in milliseconds compared to the producer reference time. This information is strongly recommended.
- MaximumLatency: The service provider's indication about the maximum presentation latency in milliseconds. Indicates a content provider's desire for the content not to be presented if the latency exceeds the maximum latency. If this applies for the service, then this information should be added
- MinimumLatency: The service provider's indication about minimum presentation latency in milliseconds for example to avoid inconsistencies with second screen applications, overlays, etc. If this applies for the service, then this information should be added.
- The latency should refer to a producer reference time in the MPD.

With respect to permitting DASH clients to do accelerated or decelerated playback, the **PlaybackSpeed** element may be present to provide a permission on the boundaries for this service. If present, the following nay be defined:

- maxPlaybackRate: The maximum playback speed that the content provider indicates is appropriate for the purposes of automatically adjusting playback latency and buffer occupancy during normal playback.
- minPlaybackRate: The minimum playback speed that the content provider indicates is appropriate for the purposes of automatically adjusting playback latency and buffer occupancy during normal playback.

4.X.4.3 Interface IF-1

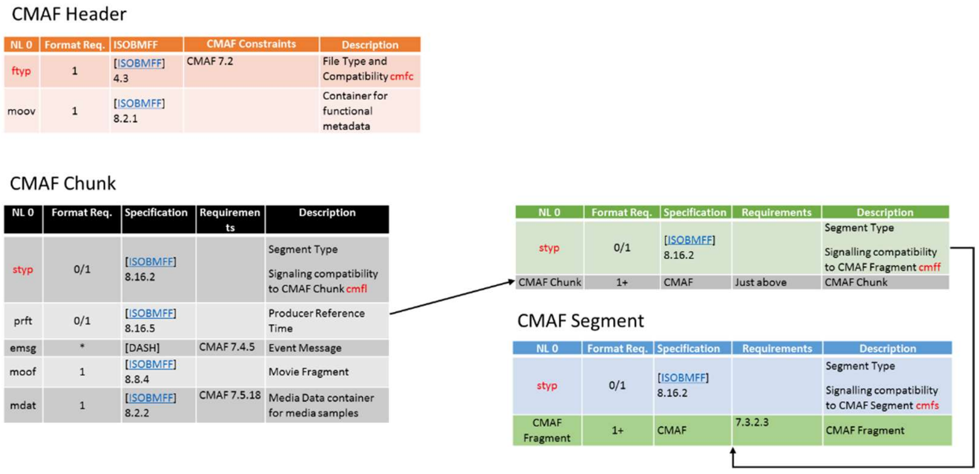
Interface IF-1 describes the formats and timing requirements between the MP4/CMAF Packager and the DASH Packager. Note that the MP4/CMAF packager is used as a prominent example, but other variants may be used as well.

The protocol is not defined and may be any protocol that is able to deliver the configuration parameters and formats defined in this clause, together with the timing requirements.

The generated formats by the MP4/CMAF Packager in low latency mode are:

- CMAF Headers, identified with an ftyp and an appropriate brand indicating the media profile. MP4/CMAF Headers are timeless but apply in causality to the chunks following this MP4/CMAF Header.
- CMAF Chunks:
    - Initial MP4/CMAF chunk that corresponds to the start of a MP4/CMAF Fragment. They are signalled by styp boxes and contain two compatibility brands for chunk 'cmfl' as well as the start of a fragment 'cmff'.
    - Non-Initial MP4/CMAF chunks corresponding to a chunk that does not co-incide with the start of a MP4/CMAF Fragment. The must not contain the 'cmff' brand, but the brands for chunk 'cmfl'.
    - Random Access MP4/CMAF chunks corresponding to a chunk that does not co-incide with the start of a MP4/CMAF Fragment, but permit random access to gether with the CMAF Header.

  Note: MP4/CMAF Fragments only provide a conceptional approach, i.e. they are a sequence of chunks and provide the granularity of switch points and random-access points.
- According to the MP4/CMAF specification, MP4/CMAF chunks may carry in addition to styp also prft and emsg boxes. Their usage and interpretation are further discussed below.
- Availability Times of the MP4/CMAF Header as well as for the first byte of the first MP4/CMAF fragment.
- Additional Metadata for which delivery is not defined, but is expected to be available to the DASH packager as shown in Table 1.



CMAF Header

NL 0	Format Req.	ISOBMFF	CMAF Constraints	Description
ftyp	1	[ISOBMFF] 4.3	CMAF 7.2	File Type and Compatibility cmfc
moov	1	[ISOBMFF] 8.2.1		Container for functional metadata

CMAF Chunk

NL 0	Format Req.	Specification	Requirements	Description
styp	0/1	[ISOBMFF] 8.16.2		Segment Type Signaling compatibility to CMAF Chunk cmfl
prft	0/1	[ISOBMFF] 8.16.5		Producer Reference Time
emsg	*	[DASH]	CMAF 7.4.5	Event Message
moof	1	[ISOBMFF] 8.8.4		Movie Fragment
mdat	1	[ISOBMFF] 8.2.2	CMAF 7.5.18	Media Data container for media samples

NL 0	Format Req.	Specification	Requirements	Description
styp	0/1	[ISOBMFF] 8.16.2		Segment Type Signalling compatibility to CMAF Fragment cmff
CMAF Chunk	1+	CMAF	Just above	CMAF Chunk

CMAF Segment

NL 0	Format Req.	Specification	Requirements	Description
styp	0/1	[ISOBMFF] 8.16.2		Segment Type Signalling compatibility to CMAF Segment cmfs
CMAF Fragment	1+	CMAF	7.3.2.3	CMAF Fragment

Figure 2 CMAF/MP4 structures

In the following, configuration Parameters are provided to the DASH packager in order for the DASH packager to generate the MPD and offer the media presentation. If the MP4/CMAF header and segments are well formed (containing language, `track_id`, bitrate box, cenc boxes for encryption information etc) the packager can have enough information, the packager can detect how many streams are ingested and with the bit-rates it can determine the number of switching sets, also default fragment durations etc can be signalled in the MP4/CMAF stream.

Note: this clause and table is expected to be aligned with interface DASH-IF Ingest Interface

Table 1 Configuration Parameters needed for DASH Packager

Parameter		Symbol	Explanation
Service Description			Provides a Service Description object from the content provider following the MPEG semantics as defined in clause 4.X.4.2.5
Change Lead Time		`CLT`	the minimum time in media time that the DASH Packager gets a pre-warning for a change in the program offering, e.g. SCTE-35 messages lead time.
Number of Switching Set		`NoSS`	The number of Switching Sets that are generated by the encoder
For each Switching Set `i=1, …, NoSS`			
	CMAF Header	`CH[i]`	The describing MP4/CMAF Header, i.e. the one that provides a compatibility point for all MP4/CMAF Tracks in the MP4/CMAF Switching Set. The MP4/CMAF header allows to extract the following parameters at the minimum: • Timescale • Media Type/Handler • Codecs Parameter (Sample entry) • Encryption
	Aligned Switching Set	`ASS[i]`	Specifies all aligned switching sets that this switching set can switch to from an alignment perspective.
	prftIndicator	`PRRTF`	Indicates that the prft is present in the media stream and that the ntp_time provides a production time of the media time associated in the prft.

	Maximum Fragment Duration	`FDmax[i]`	The maximum MP4/CMAF Fragment duration in the Switching Set (note that this is identical in terms of presentation time and in terms of decode time).
	Reference Buffer		Provides buffer information (the movie box may contain such information in the BitRateBox) `bufferSizeDB` gives the size of the decoding buffer for the elementary stream in bytes. `maxBitrate` gives the maximum rate in bits/second over any window of one second. `avgBitrate` gives the average rate in bits/second over the entire presentation.
	Reference Buffer Duration	`RBD[i]`	specifies a common duration used in the definition of the MP4/CMAF track data rate. If not present, then the value of FD[i] is assumed.
	Reference Bandwidth	`RBW[i]`	Consider a hypothetical constant bitrate channel of bandwidth with the value of this attribute in bits per second (bps). Then, if the MP4/CMAF Track is continuously delivered at this bitrate, starting at any Fragment Boundary a client can be assured of having enough data for continuous playout providing playout begins after `RBD[i]` * `RBW[i]` bits have been received (i.e. at time `RBI[i]` after the first bit is received).
	Nominal Fragment Duration	`FD[i]`	The nominal MP4/CMAF Fragment duration in the Switching Set (note that this is identical in terms of presentation time and in terms of decode time). The nominal time is the one that the encoder plans to use unless unforeseen aspects happen. This should typically be an average as well (to avoid drift) as well as the number that is used frequently
	target MP4/CMAF chunk duration	`CD[i]`	The target MP4/CMAF chunk duration in decode time. This is in scale of the timescale
	maximum MP4/CMAF chunk duration	`CDmax[i]`	The maximum MP4/CMAF chunk duration in decode time. This is in scale of the timescale
	Maximum composition offset	`MCmax[i]`	The maximum composition offset (a couple of frames). Generally, information is needed on how the encoding is done and how the chunks are generated and produced.
	InbandEventStreams or sparse metadata tracks or anything else that carries such timed events.		Such events, if provided result in actions in the DASH packager. Different types of messages may exist, for example SCTE-35 messages for ad insertion, black outs or program changes. Other type of messages may indicate the change of a program as well. As an abstraction, we define three different types of triggers from upstream. - An update of the MPD is necessary that needs to be indicated to the client, e.g. due to a program change or similar events. - A emsg message needs to be inserted/removed/carried through at the start in the start of a media segment - The absence of incoming data or data that is known to be not correct (error handling) The message may be included as part of the media or may be provided at external triggers. The important aspect is, on what is

			the maximum delay from the media time that is currently processed at the packager to the media time this requires an information being provided the DASH client. The CLT provides a maximum difference on this, but the event could occur earlier, permitting changes quicker.

4.X.4.4 MPD Generator and Packager for Low-Latency and Regular Client producing Multi-Period Content

<need to check with Ad Insertion/MultiPeriod>

4.X.4.4.1 General

This clause introduces a reference DASH packager for low-latency that also adds a period boundary at indicated times. The content at such program changes is expected to be properly formatted. The implementation only provides an example and different way may be used to accomplish low-latency.

The basic operation of the DASH packager is as follows:

- The DASH packager creates an initial MPD based on the configuration information following Table 1.
- DASH packager acts as a slave to the ABR encoder and incoming data formats on IF-1.
- The DASH packager formats and generates MPDs dynamically as well ingests the Segments into the CDN.
    - The DASH packager ensures the Segment availability times in the MPD are correct at the origin.
    - The DASH packager manages upstream triggers and generates the MPD and/or inserts events into the MPD and/or into the Segments.
- The DASH packager manages upstream errors, for example missing or corrupt data, etc.

The DASH packager communicates with the origin such that it can transfer Segments into the CDN, such that

1) The Segment URL is accessible even if only parts of the Segments are delivered
2) The Segment can be delivered in a progressive manner or chunked manner whereby the packager controls the chunk size. An example is HTTP Transfer Encoding, but also other ingest protocols may fulfil this property.

Note: We need to check to what extent this can be combined with DASH-IF Ingest interface-2

This clause provides more details for different DASH modes.

4.X.4.4.2 Backward-Compatible Simple Live

The clause provides a reference implementation for a DASH packager using the following DASH features and configuration:

- the simple live profile (with `$Number$` or `$Time$` and `@duration` for duration signalling is used
- the DASH client can either use MPD updates following `@minimumUpdatePeriod` or may rely on MPD validity expirations sent inband.
- multi-period offerings are supported, either for ad insertion opportunities, program changes or for operational purposes

The following service offering is assumed:

- Each CMAF fragment generates one DASH segment
- Each CMAF chunk is offered as an HTTP Chunk. Note that this assumption should not be considered by the client implementation.

The following two processes are documented:

- generation of the initial MPD
- dynamic operation of the packager including MPD processing and Segment offering

Assuming that the DASH packager receives configuration parameters and MP4/CMAF or MP4/CMAF equivalent data as as introduced in clause 4.X.4.2. Then the initial MPD parameters may be generated (following Table 11) as shown in Table XX:

Table XX – MPD values for Backward-Compatible Simple Live

MPD Information	Status	Proposed Value
MPD@type	mandatory, set to "dynamic"	the type of the Media Presentation is dynamic, i.e. Segments get available over time.
MPD@availabilityStartTime	mandatory	the start time is the anchor for the MPD in wall-clock time. The value is denoted as *AST*.
MPD@minimumUpdatePeriod	mandatory	this field is mandatory except for the case where the **MPD**@mediaPresentationDuration is present. However, such an MPD falls then in an instance as documented in section **Error! Reference source not found.**. This value needs to be set low enough such that with the CLT you can announce MPD updates fast enough. If unsure on the value, set it to 0.
MPD@minBufferTime	Mandatory	Set it to the RBI[i] of the video
Period@start	mandatory	the start time of the Period relative to the MPD availability start time. The value is denoted as *PS*.
AdaptationSet	Mandatory	Extract Information from MP4/CMAF Header. For details refer to <clause XXX>.
Representation@productionReferenceTime	Optional	If present, indicates that the prft box is present in the MPD
Representation@presentationTimeOffset	Optional default: 0	Sets this value to the presentation time for this Representation at the start of the Period. It is preferred that this value is set on Adaptation Set level.
Representation@bandwidth		Set it to the RBW[i] of the Representation (need to do some conversion if RBI of Adaptation Set is different than minbuffertime.
Representation@availabilityTimeOffset	Recommended. If absent, default is 0.	Setting this value to a value greater than 0 enables to signal that the segment availability time of the segments is earlier than nominally announced, namely offset as provided by the value of this attribute. In the case of low-latency DASH (i.e. the operation documented below), it is recommened to set the value to at most the difference of the

		nominal segment duration minus the nominal fragment duration.
Representation@isComplete	Set to false	Indicates that the segment is not yet available as a whole at the announced availability time. In the case of low-latency DASH (i.e. the operation documented below), it is recommened to set the value to false.
SegmentTemplate@media	Mandatory	the template for the Media Segment
SegmentTemplate@startNumber	optional default	the number of the first segment in the Period. The value is denoted as *SSN*.
SegmentTemplate@duration		the duration of each Segment in units of a time. The value divided by the value of @timescale is denoted as *MD*[*k*] with k=1, 2, ... The segment timeline may contain some gaps. It is recommended to set this to the nominalFragmentDuration as provided by the MP4/CMAF packager.

For each MP4/CMAF Track in a Switching Set, the track is mapped to a Representation in an Adaptation Set. Assuming that the MP4/CMAF track is a MP4/CMAF Track with the properties documented above and this MP4/CMAF Track is received by the DASH packager in sequences of MP4/CMAF Headers and MP4/CMAF Chunks.

The DASH packager may operate as follows:

- The @minimumUpdatePeriod is set to a value that is equal or smaller than the change lead time CLT provided by the segment stream.
- With incoming MP4/CMAF Headers and MP4/CMAF Chunks, the DASH packager operates as follows
    - CMAF Header: A new Period is generated and the Adaptation Set values are set as follows:
        - The @timescale of the Adaptation Set is set to the timescale of the included media (look into the MP4/CMAF Header and extract the time scale from the track header)
        - The @duration attribute is set such that the nominal duration nominalFragmentDuration is documented in the MPD for this Adaptation Set.
        - $Number$ or $Time is used of segment templating.
        - Other Adaptation Set relevant parameters are extracted from the MP4/CMAF Header for the Adaptation Set and for each Representation
    - initial MP4/CMAF chunk:
        - if it is not the first segment in the period, determine the duration of the previous segment (summing the truns of the previous chunks)
        - If an emsg is present that is targeted for the DASH packager processing, processing needs to be done by the DASH packager and the emsg is removed from the fragment before sent to the DASH client.
        - If a prft if present, the media time and the corresponding wall-clock time are stored in the packager such that at a new period, this information can be added to the manifest

    - a new segment is generated in the MPD by the DASH packager and the DASH packager checks the validity of the MPD offering. If still valid, no changes to MPD are done. Only if changes are done that are no longer valid, a new MPD is written. Specifically,
      - The MPD start time of the new segment must be in the range of *EPT* - *PTO* - 0.5**DUR* and *EPT* - *PTO* + 0.5**DUR* with *DUR* the value of `@duration`.
      - If this is not fulfilled a new Period is written that includes the following:
        - The **`Period`**`@start` is set such that the MPD start time is correct.
        - The `@presentationTimeOffset` is set to the EPT of the first segment
        - The **`ProducerReferenceTime`** element is added following the instructions from clause 4.X.4.3.2.
        - The `@startNumber` is set to the first segment in the new Period.
        - The Adaptation Sets are continued by providing Period continuity signalling with each Adaptation Set.
        - The `@availabilityTimeOffset` is set to the (segment duration – largest chunk duration).
    - The initial chunk is offered as a resource that is referenced by the URL announced in the MPD and sent as an HTTP chunk to the origin
  - Non-initial MP4/CMAF chunk
    - Event Message Processing
      - If an emsg is present, the emsg is stored and added to the next segment in a consistent manner. The emsg may be kept in the chunk if it is targeted for the client.
    - The non-initial chunk is appended to the segment that is offered by the corresponding URL announced in the MPD and sent as an HTTP chunk to the origin

- Generally it is assumed that the contribution signal and the encoder are always available. However, when an encoder fails for one or more specific Representations to generate the next chunk, then the DASH content generator
  - terminates the Segment with the last sample in the segment, (which is possibly corrupted)
  - generates a new MPD as follows:
    - The `@minimumUpdatePeriod` is set to `0`.
    - If all or at least many Representations fail, the Period`@duration` is set to the value of the media time in the Period that is still available.
    - If only a subset of the Representations fail, the `@presentationDuration` for the last segment is set to the value of the last presentation time in the Representation that is still available.
    - By doing so, the content provider basically informs the DASH client that for the duration of the Segment as announced, no media is available. The DASH client revalidates this after every Segment duration. The MPD is not changed on the server until either the decoder resumes or the Media Presentation is terminated.
    - If the `@minimumUpdatePeriod` is long, then the client may request non-existent segments, which itself may then trigger that the DASH client revalidates the MPD. If the DASH client has the possibility, it should add the ‘`lmsg'` brand as a compatibility brand to the last generated segment. In addition, when the segment is distributed over HTTP, the HTTP header should signal the content type of the segment including the compatibility

brand 'lmsg'. If the DASH client can identify this, it is expected to refetch the MDP and may by this means observe the early terminated Period or Representations.

- Only after the encoder resumes, a new MPD is written as follows:
  - A new Period is provided with **Period**@start according to the value of the new Period. The @presentationTimeoffset of the Representation of the Period shall match the the earliest presentation time of the newly generated Segment. If appropriate, Period connectivity should be signaled.
  - The @minimumUpdatePeriod is set again to the minimum change lead time.

- when a program change is announced, generates a new MPD as follows:
  - The @minimumUpdatePeriod is set to 0.
- When the program change occurs
  - Write a new MPD with all the parameters
  - Reset the @minimumUpdatePeriod is set to a value that is equal or smaller than the change lead time provided

4.X.4.4.3 Backward-Compatible Main Live

Note: This extension using the SegmentTimeline is provided in a future version.

4.X.4.4.4 Broadcast TV Profile

Note: This extension using the Broadcast TV profile is provided in a future version.

4.X.4.5 Per Session MPD Modification

Note: This aspect is discussed as part of the Ad Insertion work and will updated once completed.

4.X.5 Client Implementation Guidelines (Informative)

4.X.5.1 Low-Latency DASH Client

4.X.5.1.1 General

Note: dash.js v3.0 implements a low-latency mode. Important aspects done for low-latency additions will be added here in a revised version.

Considered issues

- DASH client implementation and configurations (define control mechanisms, segment request times, rate control and ABR, etc.)
- DASH client implementation
  - It needs to be configurable on using the low-latency mode and what configuration to use, e.g. how aggressive it is in the latency.
  - Updates and considerations in the ABR logic and bitrate estimations
  - Considerations start-up operations to maintain latency and start-up delays (joining needs to be done carefully, better to wait for the next RAP than playing out old stuff).
  - The client is configured by
    - Information in the MPD
    - External information, for example a controlling app ➔ needs to have an app
    - Client does its own magic to identify proper configuration
    - Combinations of the above
    - Obeying the service description

4.X.5.1.2 Backward-Compatible Simple Live

Note: Specific issues will be added during a future revision

4.X.5.1.3 Backward-Compatible Main Live

Note: Specific issues will be added during a future revision

4.X.5.1.4 Broadcast TV Profile

Note: Specific issues will be added during a future revision

4.X.5.2 Reference Playback of Low-Latency

The reference playback platform is expected to support the CTA WAVE Device Playback Specification requirements [X5], in particular playback of low-latency and chunked content.

4.X.6 Network Operations Guidelines (Informative)

Note: Specific issues will be added during a future revision

Issues considered

- Origin Server (push or pull)
    - Transfer encoding starts from here
    - TLS encryption will be applied from here
    - New HTTP codes? Such as come back later
- CDN operation
    - Special operations as well
    - Need to keep transfer functions in place
    - And need to cache the object
    - Needs to run with https
    - New HTTP codes? Such as come back later
- (Access Network)
    - Is preferably stable in throughput
- Network protocols and configuration, CDN operation (new status codes, etc.), etc. This may include recommendations to use push-protocols (e.g. HTTP/2, WebRTC) to reduce latencies

4.X.7 Potential Optimizations (Informative)

Note: Specific issues will be added during a future revision

Issues considered

DASH Encoding & Packaging (includes media encoding, MP4/CMAF usage, MPD Signalling)

- Service offerings: how to operate work flow of encoder, dash packager, external triggers (e.g. ads ,etc.) efficiency and CDN robustness
- Is there anything on media decoders, for example start playback audio before video is available
- adding join Representations for faster access to the content?