

Doc. Number: CTA-2066

Date: 02/21/2019

Streaming Quality of Experience Events, Properties and Metrics

For further information:

R4 WG20 Chair: Neel Jadia, Email: njadia@conviva.com

CTA Staff: Mark Levine, Phone: 571-236-7921, Email: MLevine@cta.tech

Development History:

02/21/19: Drafts v1 distributed.

Copyright Notice:

This draft **standard or bulletin** is copyrighted by the Consumer Technology Association™ (CTA). No distribution outside of the responsible formulating group, or its company/organization members, is permitted without the prior written permission of CTA staff. In addition, prior written permission of CTA staff is required to incorporate this draft, in whole or in part, into another document. Distribution includes posting this draft to a website, email, paper or other means. Unauthorized distribution or incorporation will be treated as an infringement of CTA's copyright, and will not be permitted.

NOTE—Distribution of this draft for review and comment within a company/organization that is a member of the responsible formulating group is encouraged and does not constitute a copyright infringement.

Patent Disclosure Request: **[Include for standards only. Do NOT include for bulletins.]**

If your company has a patent(s) or pending patents(s) that is/are required to implement this draft standard, CTA requires a statement from the patent applicant or holder indicating compliance with the CTA intellectual property rights policy by stating one of the following:

1. A license shall be made available without charge to applicants desiring to use the patent for the purpose of implementing the standard, or
2. A license shall be made available to applicants under reasonable terms and conditions that are demonstrably free of any unfair discrimination.

If your company knows of existing patent(s) held by another company that is/are required to implement this draft standard, notify CTA staff. CTA staff will contact the company to

request the above statement from the patent applicant or holder indicating compliance with the CTA intellectual property rights policy as indicated above.

For further information concerning this statement, please contact responsible CTA staff person at 703-907-7640 or mlevine@ce.org.

REMOVE THIS PAGE PRIOR TO PUBLISHING

https://docs.google.com/document/d/1XFuMf4oHmkG7_nzGEq5lqbzF9XssorOZQyH426rKPNs/edit

FOREWORD

This standard was developed by the Consumer Technology Association R4 Video Systems Committee

(This page intentionally left blank.)

CONTENTS

Scope	7
Revision History	7
References	7
Informative References	7
Compliance Notation	8
Definitions	8
Symbols and Abbreviations	10
Introduction	11
Expectations for Primary Audiences of this Standard	13
Media Player API Developer	13
Metrics Aggregator/Reporter, and Analytics Vendor	13
Standardized Aggregate Metrics	14
A note on video & audio quality metrics	14
The Danger of Averages	15
Standardized Playback Session Metrics	18
Optional video quality metrics	19
Standardized Events	21
Standardized Player Properties	23

FIGURES

(This page intentionally left blank.)

Streaming Quality of Experience Events, Properties and Metrics

Scope

CTA-2066 specifies a set of player events, properties, Quality of Experience (QoE) metrics and associated terminology for representing streaming media quality of experience across systems, players and analytics vendors.

This working group aims to define this standard.

Revision History

- 2017-04-12 CTA-2066 r1
- 2017-12-12 CTA-2066 r2

References

Informative References

The following references contain provisions that, through reference in this text, constitute informative provisions of this standard. At the time of publication, the edition indicated was valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standard indicated below.

1. QoE WG Events Document, Streaming Video Alliance, Quality of Experience (QoE) Working Group – Recommendations for Standardized Streaming Event Reports, Chair: Edwards, Thomas, 10/24/2016, sva2016.053.01.docx
2. QoE Metrics Document, Streaming Video Alliance, Quality of Experience (QoE) Working Group – Recommendations for Standardized Streaming Metrics, Chair: Edwards, Thomas, 10/24/2016, sva2016.064.00.docx
3. DASH - IF position Paper: Proposed QoE Media Metrics standardization for segmented media playback, version 1.0 published October 7, 2016, DASH Industry Forum, <http://dashif.org/wp-content/uploads/2016/10/ProposedMediaMetricsforSegmentedMediaDelivery-r12.pdf>
4. ITU-T P.1203 (10/2017), Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport
5. VMAF

Compliance Notation

As used in this document “shall” and “must” denote mandatory provisions of the standard. “Should” denotes a provision that is recommended but not mandatory. “May” denotes a feature whose presence does not preclude compliance, and implementation of which is optional. “Optional” denotes items that may or may not be present in a compliant device.

Definitions

<i>Adaptive Manifest</i>	A data structure, often represented in a text document that describes the available media with various qualitative differences, such as their bitrate. (In HLS this is known as the master playlist, in DASH it is the MPD manifest).
<i>Event</i>	In the context of a streaming media player, an event is a notification of an important change in the operational state of the audio/video player. For example, the player state change from playing to paused constitutes an event.
<i>Playback Session</i>	A <i>Playback Session</i> begins when a user requests content to play, and ends when the intended content is finished or the user abandons the content. The intended content may be a single video, a playlist, or a channel. See the section <i>Standardized Playback Session Metrics</i> for a broader definition.
<i>Primary Content</i>	The content that the viewer is intending to watch or is mostly likely to understand that they are watching. The Primary Content can be an individual video title or a channel that plays multiple video titles.
<i>Ad Content</i>	Any secondary piece of media content that is delaying or interrupting the playback of the Primary Content.
<i>QoE (Quality of Experience)</i>	The viewer’s degree of delight or annoyance with an application or service. It is a measure of [viewer perceivable improvement and degradation of the audio and video] and the [viewer’s satisfaction with the media experience]. For instance, initial video startup time is a metric of user perceivable latency with multiple contributing factors (Time-to-first-byte, Player load time,

rendered video artifacts, download time, CDN congestion...)

QoS (Quality of Service)

A measure of the performance of the different components in the ecosystem that may or may not contribute to the Quality of Experience. For instance, “time-to-first-byte” would be an example of quality of service where a CDN being the component and a measure of latency. QoS may not always have an impact on the user experience.

Aggregation Interval

The wallclock interval for calculating for session level metrics for windowing periods less than the session level of aggregation.

Aggregation Set, Set

The population of *Playback Sessions* considered for a given metric. A ‘Set’ can vary with dimensions of interest (ex. time, geography, device, content, service type etc).

Streaming Media

Streaming media is video and/or audio content sent in a compressed format over the Internet and played on a user’s device.

Media Player

The Media Player is a component that “plays” audio and video files or streams on a computer, phone, tablet, set-top-box, television or any devices capable of rendering A/V files or streams.

Wall-clock Time

The elapsed real time observed by the user.

Play Time

The amount of wall-clock time the video is actively playing for the viewer. This is *exclusive* of any time the player is paused, buffering, seeking, or starting up.

Media Time

The accumulated length of any media that has been played (i.e. does not include buffering time or paused time). Note that speed changes are not accounted for, so a 60-sec media clip accumulates 60-sec of media time even if played at half speed.

Symbols and Abbreviations

<i>QoE</i>	Quality of Experience
<i>QoS</i>	Quality of Service

Introduction

Today, devices from different manufacturers, media players, and media delivery companies expose media playback events in different ways, often with missing or incomplete data. The industry hasn't yet agreed to a minimum baseline of *QoE* events. This situation makes it difficult to compute *QoE* metrics and compare performance across Players, Analytics Solutions and distributors.

If every Over-The-Top (OTT) media player consistently surfaced a minimum set of *QoE* events and properties and the analytics ecosystem used these to compute *QoE* metrics in a standardized manner, then all stakeholders would gain a consistent view of the performance. Such consistent, accurate and rich data would in turn lead to an improvement of the quality experienced by end-users.

The goal of this specification is to define a set of player events, properties, *Quality of Experience (QoE)* metrics, associated terminology, and attribute access for representing streaming media quality of experience across systems, players and analytics vendors. This specification guides media player vendors in implementing a minimal set of standardized client-side properties and events, as well as analytics vendors/metric implementers in measuring the quality of experience consistently.

Figure 1 shows a generic over-the-top (OTT) media player application. Inside the physical device, the operating system supports a player execution environment, within which runs the media player and a (possibly 3rd party) analytics client. The job of the analytics client is to collect information from the player by accessing standardized properties and events.

The analytics client will typically use the collected information to calculate and send standardized metrics to an analytics server. The client may also send just the raw data to the server, which in turn derives and computes the required metrics.

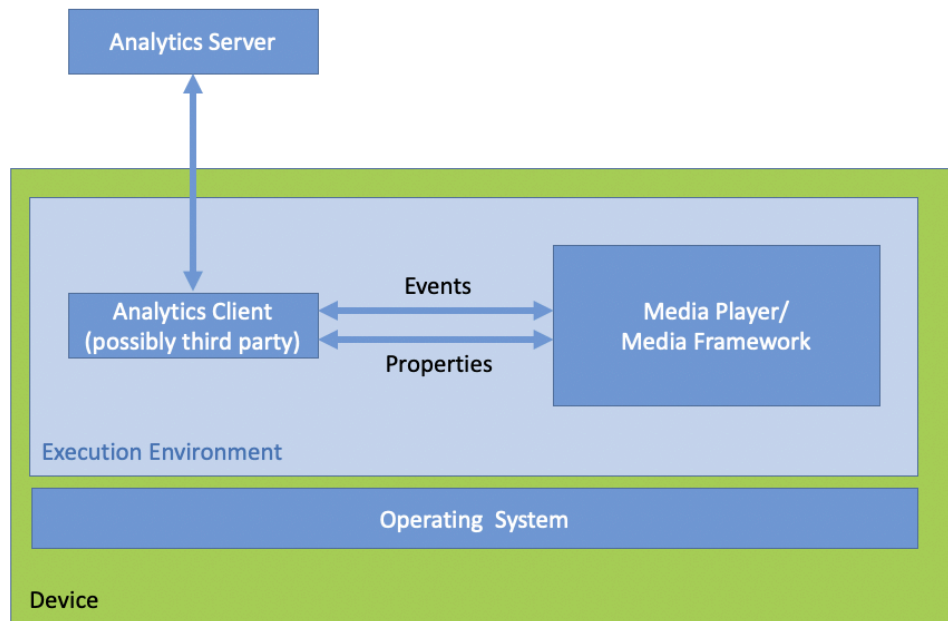


Figure 1: Generic player and analytics agent

- **Media Player** – The Media Player is a component that "plays" audio and video files or streams on a computer, phone, tablet, set-top-box, television or any devices capable of rendering A/V files or streams.
- **Analytics Client** – The client-side analytics component queries a media player for playback properties, listens for playback events, and reports this information to an analytics server for further processing, aggregation, analysis, and display.
- **Analytics Server** – An Analytics Server receives data from the Analytics Client and then logs, aggregates, and computes this information for analysis and displaying the data collected for understanding and optimizing video usage.

Expectations for Primary Audiences of this Standard

Media Player APIs

- Must make available all the specified events and properties in complete conformance with their definition in this standard.
- Should use the names of events and properties as defined in this standard. However, changing the name of events/properties of an existing player is not necessary, but a clear and reasonably findable mapping between existing events/properties and the matching events/properties defined in this document must be provided by the media player vendor. Do not map an existing player event/property that does not match the document definition exactly, instead create a new player event/property that does.
- Additional events and properties may be made available in addition to the minimum specified here.

Metrics Aggregator/Reporter, and Analytics Vendor

- Shall calculate reported session metrics and aggregated metrics in complete conformance with their definition in this standard.
- Should name reported session metrics and aggregated metrics exactly as defined in this document, or provide a mapping table.
- Additional session metrics and aggregated metrics may be calculated, but at a minimum the metrics defined here shall be made available.

Standardized Aggregate Metrics

A goal of this standard is to provide metrics for comparing the viewer experience across players, devices, and other dimensions. For this use case one will want to look at aggregate metrics—calculated from many *Playback Sessions*. Note - this specification attempts to define a core set of aggregate metrics. There are many other useful aggregate metrics that are useful in analysis but which are not defined here. We attempt to ensure that the player events and properties that we define are sufficient to enable the collection and calculation of these additional metrics.

Common problems with the playback experience can be bucketed into four conceptual groups, the latter three of which have much interplay between them (while one improves, the other two may suffer). This standard includes at least one metric in each conceptual group to provide the best chance of capturing major playback problems:

Availability

The video can start and play-through without a playback failure.

Startup Time

The time from when the viewer intends for playback to start to when playback has perceptibly begun.

Continuity

Playback continues without an unintended pause.

Video & Audio Quality¹¹[pThe perceived quality that is presented to the viewer.

Definition of an Aggregation Set (or, Set in short)

Following sections enlist key metrics proposed as part of the standard towards tracking and improving quality of experience. The list is minimal by target, without intending to be exhaustive on all possible metrics, with the goal to signal major problems in each of the aforementioned playback experience metric groups. Future versions of the standard may add additional metrics as deemed relevant.

For any metric, the context around it's measurement is of paramount importance. The context should be captured and presented while interpreting, and/or inferencing based on the metrics. In the scheme of metrics proposed under this standard, the context is defined by the population of Playback sessions that fall within the scope of measurement, hereafter referred to as the 'Set'.

What constitutes a 'Set' is closely tied to semantics of the measurement. For example,

- The 'Set' could limit scope to dimensions of interest; like,

- Geography, Device, Content, Service type, Codec etc.
- The ‘Set’ could span scope to specific time; like,
 - All new sessions that started in a given time window
 - All active sessions in a given time window, whether new, or continuing from the previous window or interval such as started or ended.
- The ‘Set’ could span scope to only completed Playback sessions, independent of time
- The ‘Set’ may have inherent scope limit to sessions matching certain pre-conditions; for example:
 - Average Initial Startup Time should only include playback sessions that included a startup time measurement, i.e. the playback session was able to play the first frame of video. Playback sessions that errored or exited before startup was achieved should not be included because the startup time measurement would be *null*, which if incorporated as "zero" would make the resulting metric misleading.

The exact constructs of a ‘Set’ are not defined as part of the standard, and are left to Vendors/Service Providers. It is a goal of this standard to provide a measurement framework for relatable comparisons across vendor implementations, while leaving room for independent innovations that may drive vendor specific differentiation. For example, a vendor may offer real time metric aggregations to minute granularity on a time continuum, while another may offer same metrics measured only on complete Playback Sessions. A vendor may be looking to provide live monitoring of these metrics, while another may be looking to collect statistics for long term strategic trending/inferences. Such differences in targeted use cases may manifest as different ‘Set’ definitions.

Normative Aggregate Metrics

Metric Name	Type	Units	Calculation Definition	Notes
Playback Failure Percentage	percent, to at least one decimal place		The percent of <i>Playback Session</i> within a given <i>Aggregation Set</i> where <i>playbackFailed</i> is true.	<p>“My platform’s Playback Failure Percentage is 5% with Player A and 3% with Player B”</p> <p><i>Playback Failure Percentage</i> captures how often viewers have a video that fails to play</p>

				or fails to continue to play due to an error.
Average Initial Startup Time		seconds, up to three decimal places	The average <i>initialStartupTime</i> within a given Aggregation Set.	<p>A <i>Playback Session</i> which failed to generate a <i>startupTime</i> value should not be included in the aggregation.</p> <p>Note that calculating the average directly results in the loss of information. It may be desirable to allocate <i>Playback Session</i> counts into buckets to facilitate later analysis by histogram, or even to retain all <i>Playback Session</i> data for deeper analysis and display.</p> <p>Captures the user experience when they are first attempting to watch a video. It measures the delay in wall-clock time from when they initiate an action to launch the video (such as pressing a play button, poster image or thumbnail) to when the video begins smooth playback. If a video autostarts on page or application load without an explicit user-action, then it is counted from when the onload event is</p>

				dispatched by the page or application.
Exits Before Video Start Percentage	percent		The sum of Attempts that ended before video start without a reported fatal error, within the selected intervals.	
Average Playback Stalled Count	integer		The average of <i>playbackStallCount</i> across all <i>playbackSessions</i> in a given <i>Aggregation Set</i> .	
Playback Stalled Rate	float	stalls/min	The sum of <i>playbackStallCount</i> divided by the total stalled duration plus the total <i>Play Time</i> .	$\text{total_stall_count} = \text{sum}(\text{playbackStallCount from all sessions})$ $\text{total_stalled_duration} = \text{sum}(\text{playbackStallDuration from all sessions})$ $\text{total_play_time} = \text{sum}(\text{playTime from all sessions})$ $\text{playback_stalled_rate} = \frac{\text{total_stall_count}}{(\text{total_stalled_duration} + \text{total_play_time})}$
Playback Stalled Percentage	percent		The total stalled duration across sessions divided by the total stalled duration plus the total <i>Play Time</i> .	$\text{total_stalled_duration} = \text{sum}(\text{playbackStallDuration from all sessions})$ $\text{total_play_time} = \text{sum}(\text{playTime from all sessions})$ $\text{playback_stalled_percentage} = \frac{\text{total_stalled_duration}}{(\text{total_stalled_duration} + \text{total_play_time})}$

<i>Average Playback Bitrate</i>		kbps	The sum of <i>bitsPlayed</i> divided by the sum of session <i>Media Time</i> across a given <i>Aggregation Set</i> .	This metric uses <i>Media Time</i> instead of <i>Play Time</i> to track the actual media played.
---------------------------------	--	------	--	--

Standardized Playback Session Metrics

The previous aggregate metrics are calculated by aggregating *Playback Session*-level measurements.

The metrics in this clause define a consistent set that can be used to understand the experience for the end user for a single *Playback Session*.

A *Playback Session* encapsulates the experience of a single viewer while consuming the intended content, represented by the *contentId*. The intended content is determined by user-initiated actions. A *Playback Session* begins when a user requests content to play, and ends when the intended content is finished or the user abandons the content (following sections describe this more thoroughly).

The intended content may be a single video, a playlist, or a channel, and the *contentId* should reflect the user's intent when selecting the content.

An ad (secondary content not requested by the user) does not create a new *Playback Session* and is included in the *Playback Session* of the intended content.

A *Playback Session* starts if:

- The user clicks play for a content resource (event *playbackRequest*)
- The user opens a content resource that has autoplay enabled (event *playbackRequest*)

A *Playback Session* continues if:

- The playout intermittently stops due to rebuffering
- Ad content is shown before or during the primary content playout
- The user seeks within the same content
- The user pauses and restarts the content playout (although very long pauses might end the view due to practical measurement reasons)
- The next chained (e.g. playlist) content is automatically started, in the case where this matches with the user's initial intent.

A *Playback Session* ends if:

- The end of the current content has been played out (event *playbackFinish*)
- The user selects new content e.g. by clicking on an embedded video link (event *playbackRequest*)
- The user ends the session e.g. by closing the player, application window, or web page (e.g. window. beforeunload)
- The playback is terminated by the service, by the client (or e.g. as a result of an error condition)
- A measurement timeout expires because the analytics client did not receive or could not send an event indicating that the session had ended before the application shut down.

There is no standardized limit on the time a *Playback Session* can continue for. However, there is an open discussion on if a limit would be valuable and should be included in a future version.

There is also no standardized limit on the length of a specific metric measurement (e.g. a single stall duration) though analytics vendors may set their own limits. There is an open discussion on if limits would be valuable and should be included in a future version.

Metric Name	Type	Units	Calculation Definition
<i>playbackFailed</i>	boolean		The value is 1 true if a <i>playbackFail</i> event occurred at any point during the <i>Playback Session</i> . The value is 0 false otherwise.
<i>initialStartupTime</i>		seconds, up to three decimal places	The time in seconds between the first <i>playbackRequest</i> event and the first <i>playbackStart</i> event within a <i>playback session</i> . Note that the first <i>playbackStart</i> event might refer to an ad in case of pre-rolls.
<i>exitedBeforeVideoStart</i>	boolean		Completed <i>Playback Session</i> includes a <i>playbackRequest</i> event but did not include a <i>playbackStart</i> event or <i>playbackFail</i> event. If a <i>playbackFail</i> is reported when the video terminates before starting, the video termination is counted as a <i>playbackFailed</i> instead of <i>exitedBeforeVideoStart</i> .
<i>playbackStallCount</i>	integer		The number of <i>playbackStall</i> events that occurred during a <i>Playback Session</i> .
<i>playbackStallDuration</i>		seconds, up to three decimal places	The sum of the time deltas between a <i>playbackStall</i> event and the next <i>playbackStart</i> event, <i>playbackPause</i> event or session termination.
<i>bitsPlayed</i>	integer		$(\text{videoReportedBitrate} + \text{audioReportedBitrate}) * \text{playbackRate}$ multiplied by the elapsed <i>Media Time</i> , summed between <i>renditionUpdate</i> events.

Standardized Events

Playback Session-level measurements count the occurrence of or time between events.

Events are sent by the player and should be the preferred way for the analytics client to collect information, as this is more efficient than polling separate player properties.

In multi-threaded environments, there is the possibility of events being received in a different order to that in which they were dispatched. This difference may only be a millisecond or two, but it would be sufficient to confuse an analytics application following a rigid interpretation of the event sequence. It is suggested that analytics implementations maintain a play state model in order to correctly interpret event sequences.

Event Name	Dispatch Definition	Notes
<i>adBreakStart</i>	Dispatched at the initiation of an ad break, immediately prior to the <i>playbackRequest</i> event for the first ad	An ad break is composed of one or more ad spots being played
<i>adBreakEnd</i>	Dispatched at the completion of an ad break, immediately prior to the <i>playbackRequest</i> event for the <i>Primary Content</i> .	
<i>playbackRequest</i>	User expectation that playback will start. Dispatched when the player receives an instruction to play the media or transitions to another media asset.	This can happen when the user presses play, or when the media, either primary or ad content, is requested to start automatically and sequentially. Also dispatched when a user wants to resume after a pause.
<i>playbackStart</i>	Dispatched when the first frame of media is rendered after the player previously was not playing.	In practice, this occurs after a the <i>playbackRequest</i> event, or the player was paused, or the playback stalled.
<i>playbackPause</i>	Dispatched when the user pauses playback, or when the media player is programmatically instructed to pause.	

<i>playbackFinish</i>	Dispatched when playback reaches the natural end of media content, whether it is the <i>Primary Content</i> or a <i>Ad Content</i> . This event should not be dispatched if <i>playbackFail</i> has been dispatched.	Skipping an ad content should also fire the <i>playbackFinish</i> event.
<i>playbackStall</i>	Dispatched when the media unintentionally stops playing, usually due to lack of data in the media buffer.	This does not include start up time or subsequent seeking, which is considered starting up at a new time.
<i>playbackFail</i>	Dispatched when a fatal error occurs that prevents further playback.	This event is dispatched when a player experiences unrecoverable playback error. It does not include sessions which terminate due to timeout or user abandonment.
<i>playerResize</i>	Dispatched when the player width, height, or both have changed.	This event needs to be dispatched only when the visible display resolution of the player changes, not when the encoded content resolution changes.
<i>renditionUpdate</i>	Dispatched when the <i>encodedVideoWidth</i> , <i>encodedVideoHeight</i> , <i>videoReportedBitrate</i> , <i>audioReportedBitrate</i> or <i>videoFramerate</i> has been updated.	<i>encodedVideoWidth/Height</i> refers to the width & height reported by the decoder, which may not match what was reported in the manifest. The decoder is the source of truth. In the event that the decoder information is not available to the player, then the reported video dimensions from the manifest may be used.
<i>seekStart</i>	Dispatched by media player when instructed to move playhead time to indicate media player began seeking	
<i>seekEnd</i>	Dispatched by media player to indicate media player stopped seeking and has accumulated sufficient audio and video content in the buffer for playback	

Standardized Player Properties

A player property represents the actual value of a certain characteristic at the time of the request. Most events (see xx) already include related properties as metadata, but an analytics client might use specific property requests to gather specific information for other purposes.

When time is specified, it is in seconds with precision of up to three decimal places.

Property Name	Type	Units	Definition
<i>encodedVideoWidth</i>	integer	pixel	The encoded width in pixels of the currently playing video rendition
<i>encodedVideoHeight</i>	integer	pixel	The encoded height in pixels of the currently playing video rendition.
<i>playerWidth</i>	integer	pixel	The width of the player viewport.
<i>playerHeight</i>	integer	pixel	The height of the player viewport.
<i>videoReportedBitrate</i>	integer	kbps	The bitrate of the currently playing video as reported by the <i>Adaptive Manifest</i> .
<i>audioReportedBitrate</i>	integer	kbps	The bitrate of the currently playing audio as reported by the <i>Adaptive Manifest</i> .
<i>playbackRate</i>	float, two decimal places	-	The current rate of playback. For a video that is playing twice as fast as the default playback, the <i>playbackRate</i> value should be 2.00
<i>contentId</i>	string		A unique identifier for the content that is playing or is intended to be played. This may be a specific video title, or a channel that may play multiple titles, or even ad content.
<i>currentVideoCodec</i>	string		The currently used video codec as specified in the <i>Adaptive Manifest</i> .
<i>currentAudioCodec</i>			The currently used audio codec as specified in the <i>Adaptive Manifest</i> .
<i>videoFrameRate</i>	float, two		The current frame rate of the video as specified in the adopted manifest.

	decimal places		
--	----------------	--	--

Player Event & Properties Mapping

Each player should either emit QoE events & properties as named and defined in this spec, or otherwise include a script-accessible object/dictionary/enum that maps the standardized naming to the matching names for their player implementations. Media player vendors shall publish their mapping tables as part of their public API documentation. The mapping table shall also provide the CTA-2066 specification version number against which the media player is compliant.

Video & Audio Quality Metrics

Video and audio quality metrics attempt to give insight into the perceptual quality, or more specifically, differences in perceptual quality.

Playback bitrate is a common metric used for this purpose, however bitrate alone is not enough to understand the perceptual quality as the content type (low vs. high complexity), codec choice, frame rate and resolution can have a dramatic impact on the bitrate. It also does not communicate how well media is addressing a viewer's specific device.

For this reason we have included additional properties (e.g. video codec, frame rate, resolution etc.) that can allow for better understanding of the quality. We also recommend expanding on quality metrics in later releases of this standard, for example with a metric to measure scaling of the media content within the player.

For a more advanced measurement of the streaming quality experienced by the user, the optional P.1203 [4] metric can also be calculated. P.1203 is an ITU-T standard which estimates the video, audio and total user experience, based on different measurable entities. The resulting P.1203 metrics represents a quality value between 1 and 5, with 1 being very bad quality, and 5 being the best quality.

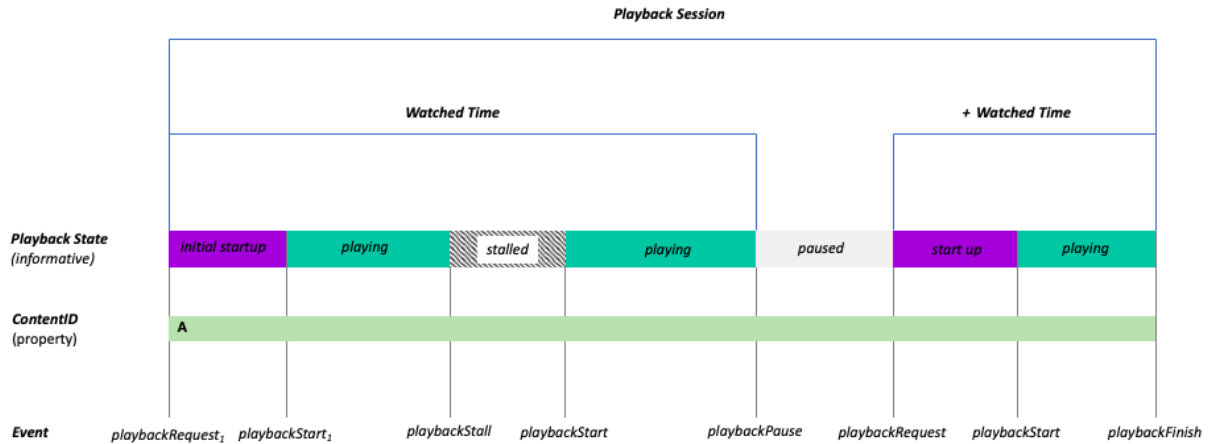
The least complex implementation variant of P.1203 is Mode 0, which uses the following input data:

- The video codec (property `currentVideoCodec`)
- The audio codec (property `currentAudioCodec`)
- Stalling information (events `playbackRequest`, `playbackStart`, `playbackStall`)
- Source information (event `renditionUpdate`, properties `encodedVideoWidth`, `encodedVideoHeight`, `videoReportedBitrate`, `audioReportedBitrate`, `videoFramerate`)
- Device information (event `playerResize`, properties `playerWidth`, `playerHeight`)

Note that only the standardized properties and events are needed to calculate the ITU-T P.1203 metric, so the metrics and aggregated metrics also defined in CTA-2066 are not needed for calculating ITU-T P.1203 metrics.

Appendix A: Architectural Workflow Models

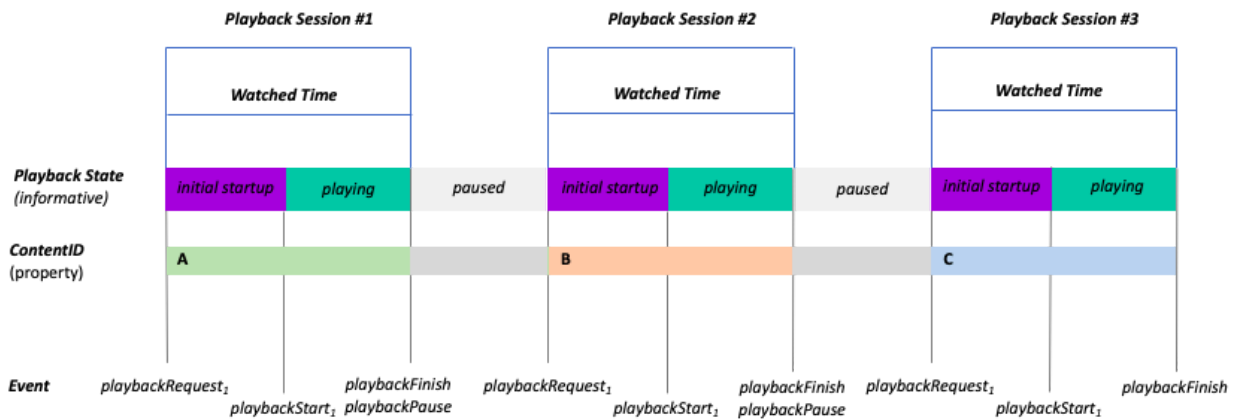
Playback of one asset with a stall and a pause, watched until the content completed.



$$\text{initialStartupTime} = \text{playbackStart}_1 - \text{playbackRequest}_1$$

Note: initialStartupTime metric only includes the duration marked as "initial startup". The subsequent "startup" duration in the diagram above can be used to create additional metrics beyond the fundamental set of metrics targeted for this version of the spec.

Playback of three independent assets

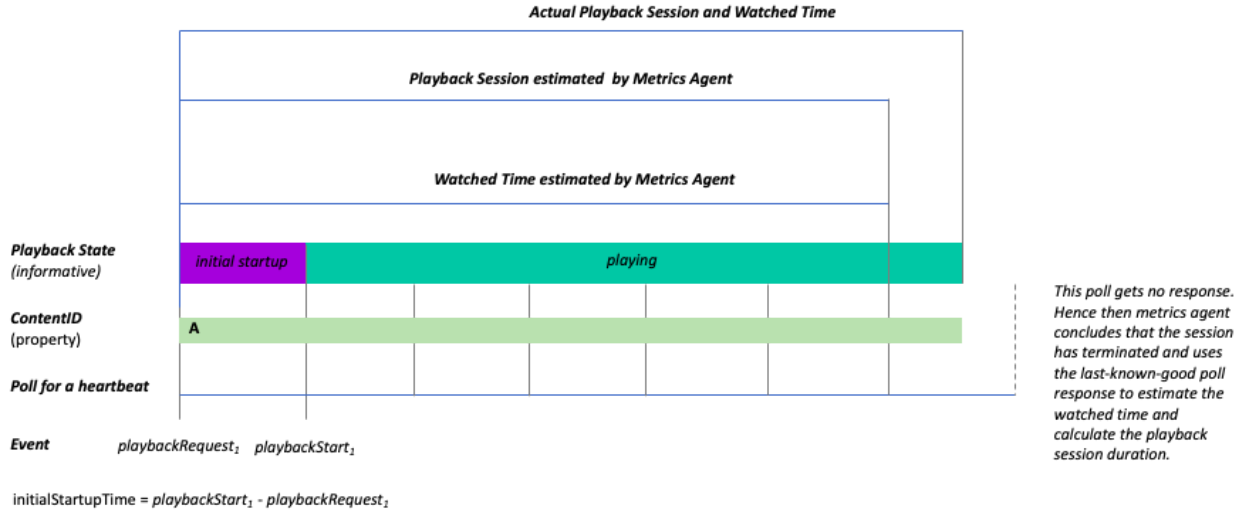


$$\text{initialStartupTime} = \text{playbackStart}_1 - \text{playbackRequest}_1$$

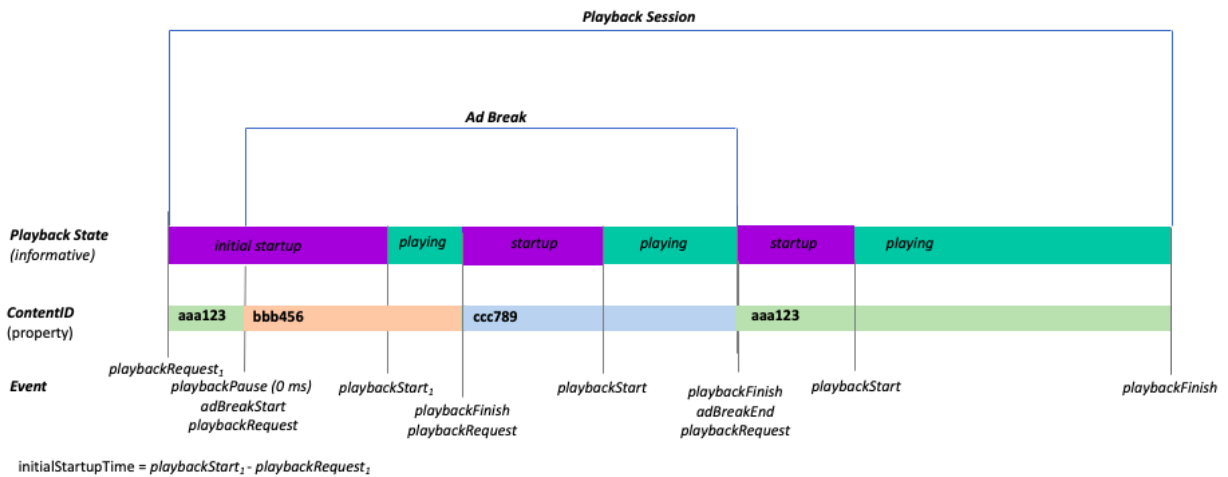
Note: The time between playbackFinish and playbackRequest, or playbackRequest and playbackStart of later assets may be 0ms, but playback still transitions between those states.

Playback of one asset with incomplete session termination

This solution shows the analytics client polling the player at some interval. Since the playbackFinish event is never sent, content watched outside of that last interval is missed.



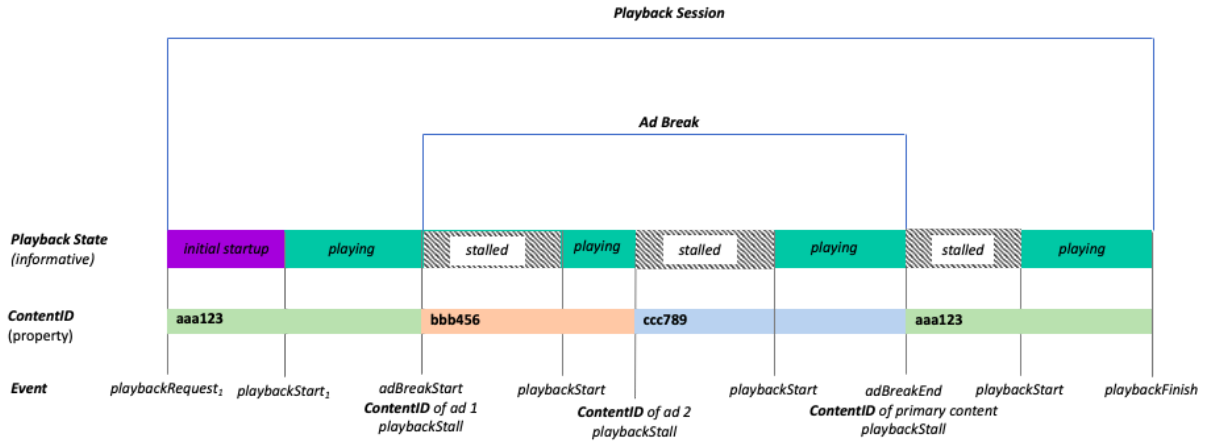
Pre-roll playback of 2 ads in succession; using non-stitched ads



Note: initialStartupTime metric only includes the duration marked as "initial startup". The subsequent "startup" duration in the diagram above can be used to create additional metrics beyond the fundamental set of metrics targeted for this version of the spec.

Note: When in an ad break, all playback events are still possible and valid (e.g. playbackPause, playbackStall) and should be treated the same when calculating Playback Session metrics, with the exception of playbackFinish, which designates the end of an individual unit of content. The contentID is used to differentiate between different ad content and primary content.

Mid-roll playback of 2 ads in succession; using non-stitched ads

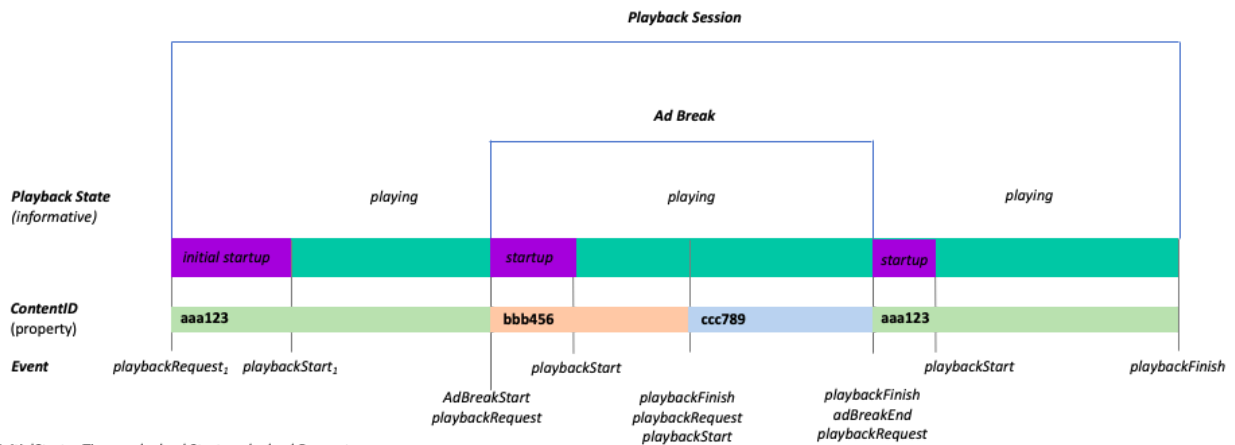


$initialStartupTime = playbackStart_1 - playbackRequest_1$

Note: when in an ad break, all playback events are still possible and valid (e.g. *playbackPause*, *playbackStalled*) and should be treated the same when calculating Playback Session metrics, with the exception of *playbackFinish*, which designates the end of an individual unit of content. The contentID is used to differentiate between different ad content and primary content.

Playback of one asset with a mid roll ad break containing 2 ads; using server-side stitched ads

The player dispatches *playbackRequest* event when it encounters a discontinuity and/or ad-break in the otherwise continuous stream.

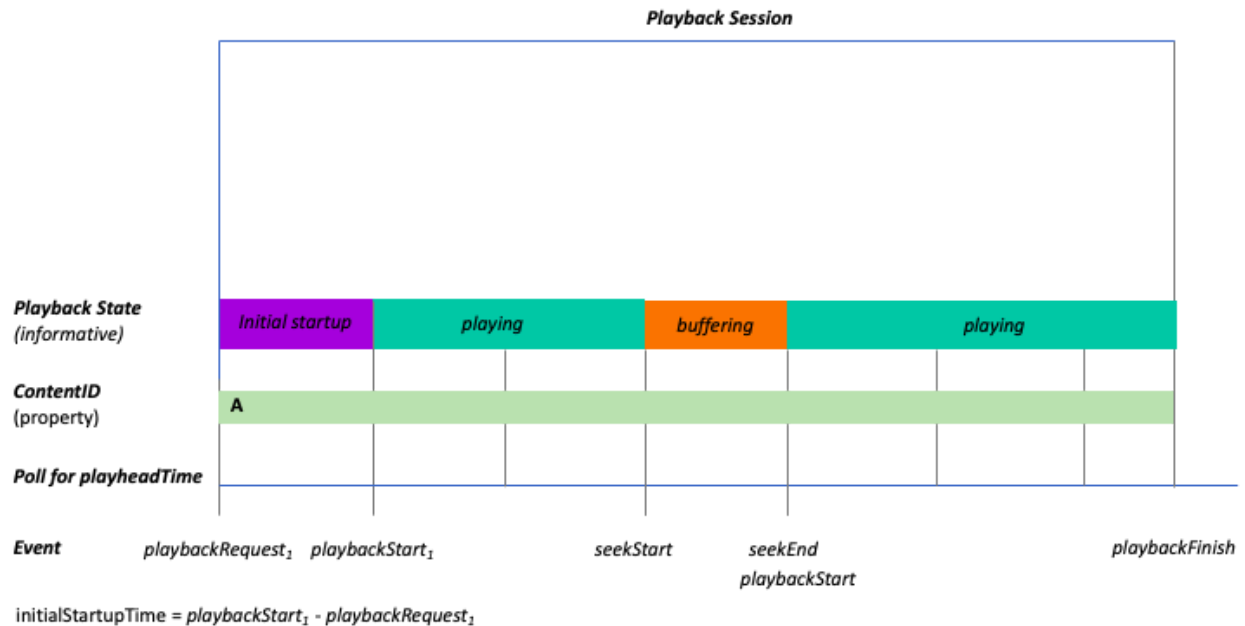


$initialStartupTime = playbackStart_1 - playbackRequest_1$

Note: initialStartupTime metric only includes the duration marked as "initial startup". The subsequent "startup" duration in the diagram above can be used to create additional metrics beyond the fundamental set of metrics targeted for this version of the spec.

Note: when in an ad break, all playback events are still possible and valid (e.g. *playbackPause*, *playbackStall*) and should be treated the same when calculating Playback Session metrics, with the exception of *playbackFinish*, which designates the end of an individual unit of content. The contentID is used to differentiate between different ad content and primary content.

Playback of one asset with single seek scenario



Appendix B: The Danger of Averages

Startup time can be a difficult metric to measure and interpret. The average is defined as the core metric since this provides a single number with which to compare this QoE dimension across analytics providers. However, average alone is a poor measurement when there isn't a normal (bell curve) distribution, and startup delays typically have a power law distribution. Single percentiles (Median, 95th) can hide important problems within the curve and are not easily aggregated over time. The best option appears to be a histogram, measuring the percent of *Playback Sessions* within buckets of measured startup time. This can give insight across all viewer experiences at time measurements with specific importance. For this reason we suggest that the *Playback Session* data is aggregated in to buckets while keeping a running average. This allows the core metric to trigger a deeper analysis, which can immediately display the bucketed data in a histogram format.

Selecting meaningful bucket boundaries will differ for content types and applications. For example a viewer may be willing to wait 5 seconds for a 2-hour movie to begin playing, but may not wait more than a second for a video in a scrolling news feed. A histogram is a useful

graphing construct in this case as it allows platforms to choose buckets which are meaningful to their particular application.