LAS: Live Adaptive Streaming

LAS1.0

June 21 , 2020

Weekly Super

Fast Hand
Guo Liang Quick
Yu Bing Kuaishou

directory

1. Preface..... 1

2. Architecture..... 2

3. Media presentation descriptions..... 3

4. LAS Request Description..... 6

5. LAS Service Description..... 8

6. LAS Client Description..... 13

7. Appendix..... 19

1.Preface

LAS (Live Adaptive Streaming) is a streaming-based live multi-bitrate adaptive standard that describes the multi-bitrate adaptive framework and implementation specifications for live streaming-based live broadcasts.

Reference code:

https://github.com/KwaiVideoTeam/las Official website: https://las-tech.org.cn

1.1. Advantages

LAS offers the following advantages:

I.Low latency: LAS implements multi-bitrate adaptation based on streaming live broadcasting, realizes frame-level transmission, and reduces end-to-end latency

II.Easy to scale: LAS live multi-bitrate adaptive framework, and traditional streaming non-multi-bitrate frameworks (such as HTTP-FLV). Fully compatible; LAS supports multiple protocols such as HTTP, QUIC, etc

WebRTC, etc

III.Easy to deploy: All major cloud vendors support it, have been verified on a large scale, and have a large number of excellent cases, such as Kuaishou

IV.Efficiency: Fewer requests and lower overhead. LAS only needs to send requests when it starts or when a bitrate switch occurs

1.2. Core content

The core content of the LAS standard includes

I.Media Rendering Description: Describes the basic semantic elements of the streaming live multi-bitrate adaptive standard

II.LAS Request Description: Describes the request generation method in different scenarios in the multi-bitrate adaptive standard for live streaming based on streaming

III.LAS Service Description: Describes the/ processing logic of the multi-bitrate adaptive standard for live streaming on the server/cloud to support streaming

M.LAS Client Description: The specific implementation of the LAS client, not as LAS ry.

LAS only gives recommended implementation architectures and adaptive algorithm strategies

2. Architecture

This section defines the architecture diagram for laS, as shown in Figure 2.1. The LAS standard mainly stipulates the description of multi-bitrate adaptive media presentation, LAS request description, LAS service description and recommendation based on streaming

LAS client architecture and live multi-bitrate seamless adaptive logic.

Figure 2.1 LAS Architecture

2.1. Media Presentation Description

Media Presentation Description Is a formatted description of Media Presentation (MP) for the purpose of providing streaming media services. Specifically, this section defines the format of the resource identifier for the media stream and the context in which the identified resource is in media rendering.

2.2. LAS Request Description

A LAS request is a request specification for a LAS client to request a media stream from the server. Specifically, this section defines in detail the generation specification and request specification for the request.

2.3. LAS Service Description

The LAS service description specifies the specification of the server's features such as media streaming codes, caching, and laS

Response specifications and exception handling specifications for request logic.

2.4. LAS Client Description

The LAS client is the carrier of LAS request initiation and media processing, and the specific implementation does not belong to the scope of the LAS standard. /b15>The LAS standard only gives recommended client architecture, live multi-bitrate adaptive strategy, seamless switching scheme, etc.

3.Media presentation description

Media rendering (MP) is a set of data that is accessed by a client to provide streaming services to users

gather. This includes the media streams that have been encoded and can be transmitted and their appropriate descriptions. Media renders descriptions

(MPD) is a JSON document that contains metadata. LaS clients use this metadata to build LAS requests that get media streams and provide streaming services to users. An example of MPD is in the appendix

A。

3.1. Hierarchical data model

In JSON format, it defines the elements and semantics of the media presentation description.

3.1.1.Semantics of MPD

The semantics of the MPD element are shown in Table 3.1

Table 3.1 MPD Semantic Tables

The name of the element or attribute	usage	description
MPD		The root element of the media rendering description
@version	Required	The version number
@adaptationSet	Must, contain at least one @representation	Adaptive collections Each MPD consists of one or more Composed of adaptive collections

3.1.2.Semantics of adaptationSet

The semantics of the adaptationSet element are shown in Table 3.2

Table 3.2 The semantics of the adaptationSet element

The name of the	usage	description

element or attribute		
@duration	Required	The length of the media stream GOP, in milliseconds
@id	Required	The unique identification number of the adaptationSet
@representation	Required	A collection of media representations that contains one or more media representations

Table 3.3 The semantics of media representation elements

The name of the element or attribute	usage	description
@id	Required	A unique identification number represented by each medium
@codec	Required	The encoding of the audio and video stream
@url	Required	The URL address represented by the media
@backupUrl	Required	The alternate URL address represented by the media
@host	Optional	The domain name represented by the media
@maxBitrate	Required	The encoding bitrate represented by the media
@avgBitrate	Optional	The average bitrate represented by the media
@width	Optional	The width of the media representation
@height	Optional	The height of the media representation
@frameRate	Optional	The frame rate represented by the media
@qualityType	Optional	The type of quality that the media represents
@qualityTypeName	Optional	The Quality Type Presentation field for media representation

@hidden	Optional	Media represents a hidden option Set to true, the corresponding media representation is not explicit, the user can not select, can only be selected through the adaptive function; Set to false, the corresponding media represents the appearance, the user can manually select
@disabledFromAdaptive	Optional	Media represents adaptive options Set to false, the corresponding media indicates that the adaptive function is visible and can be selected by the adaptive function;

		Set to true, the corresponding media indicates that it is not visible for the adaptive function and cannot be selected by the adaptive function;
@defaultSelected	Optional	Default file function options @defaultSelected media for true indicates that playback is initiated by default concentrate: 1.Of all @representation, only one media representation can appear @defaultSelected is true 2.When no @representation is set to true , the media is automatically selected to start playback

4. LAS Request Description

A LAS request describes the specific format and meaning of a request that a client sends to the server, and aLASrequest is used to request a media stream from a media server.

The LAS basic request format is defined as the form of a media streaming URL address plus an extended field, namely: url&extParam。 Detailed examples of LAS requests are shown in Appendix B.

Table 4.1 Semantics of LAS Request Elements

attribute	usage	description
@url	Required	The address of the media representation The address that points to the media representation, obtained from the media rendering profile
@extParam	Optional	Extend the parameters. Specify different requests from And implement different functions

Table 4.2 Semantics of exeParam elements

attribute	usage	description
@audioOnly	Optional, the default value is false	Audio parameters When set to true, only audio-only streams are pulled, otherwise, audio and video streams are pulled
@startPts	Optional, the default value server is available disposition. int64_t	Pull flow position parameter When set @startPts=0:

	type	Non Audio-only mode when up-to-date Video I frames begin to pull streams; pure tone Frequency mode when from the latest audio The frame starts pulling When set @startPts> 0: From pts equals @startPts medium Body frames begin to pull streams When set @startPts< 0: Pull Take the slow memory length of as

		|@startPts| Milliseconds of media data

5. LAS Service Description

The LAS service description specifies the specifications for the media streaming code, caching, and other features on the server side (CDN or self-built media server), as well as the response specifications and exception handling specifications for the LAS request logic.

Note: In this specification, the server and CDN are used as alternatives to differenceless swapping. At present, mainstream cloud vendors in China support this specification.

5.1. Transcoding specifications

The transcoding specification requires that the transcoding service must not make any modifications to the pts of the I frames in the original stream. Specifically, after entering any source video, after transcoding service, multiple video streams of different resolutions and bitrates are output, and the I frames of each video are strictly aligned, as shown in the following figure:

Figure 5.1 Transcoding specification schematic

5.2. Cache specification

5.2.1. Cache Duration Configuration

Describes the minimum length of data that needs to be cached on the server side, in milliseconds.

Table 5.1 The Meaning of the Cache Duration Element

attribute	usage	description
@maxCachedDuration	Optional, the default value is by Server-side configuration	Specifies the maximum cached data length on the server side, in ms.

		If @maxCachedDuration >0, the media data for @maxCachedDuration milliseconds is cached, otherwise the media data with the default duration is cached

5.2.2. Cache Duration Calculation

Supports calculating the cache duration separately based on the audio and video streams.

By default, the cache length is calculated using a video stream, and the cache length is calculated when there is no video stream.

5.2.3. Timestamp Fallback Processing

5.2.3.1. Cache Timestamp Fallback Definitions

Either of the following scenarios is referred to as cache timestamp fallback

I. When there is video in the cache, the I-frame pts is used as the calculation point (only the pts of the I-frame are considered) and the cache is taken

There is a non-monotonic increment in the I-frame pts sequence

II. When there is no video in the cache, all audio pts are used as the calculation point (only the audio frame is considered.)

pts) , the pts sequence of audio frames in the cache has a non-monotonic increment

5.2.3.2. Timestamp Fallback Processing

Definition 1: latestVideoPts refers to the pts of the latest video frames in the cache

Definition two: latestAudioPts, which refers to the pts of the latest audio frames in the cache

Definition three: A valid buffer, including two cases: a) The handling of case I in 5.2.3.1: when there is video in persist, the video frames pts As a calculation point, the I frame from the last monotonic increment point to

latestVideoPts all media frames as valid buffers; b) Processing of case II in 5.2.3.1: When there is no video in the cache, the audio pts are

used as the calculation point, from the last monotonically increasing audio frame to the latest AudioPts All audio frames are used as valid caches.

5.3. LAS Request Logic Processing Specification

The LAS request logic processing specification describes the specific response logic after the server receives the LAS request.

5.3.1 In LAS requests, the optional @audioOnly Default or @audioOnly=false at the same time

@startPts Default

Use the default value of @startPts (@startPts = defaultStartPts) to send the laS client the video and audio streams specified in the @url of the LAS request, as follows:

I. When there is video in the cache: pts is closest to latestVideoPts from the currently valid buffer

- |defaultStartPts| The I frame starts sending the media stream

II. When there is no video in the cache: pts closest from the current valid cache

latestAudioPts - |defaultStartPts| The audio frame starts sending a media stream

52 In LAS requests, the optional @audioOnly=true while @startPts default Use the default value for @startPts (@startPts = defaultStartPts) to send the @url specified audio stream in the LAS request to the LAS client as follows:

I. pts is closest to latestAudioPts from the currently valid buffer - |faultStartPts |

The audio frame starts sending a media stream

53 In a LAS request, the optional @audioOnly Default or @audioOnly=false, at the same time

@startPts=0

Use @startPts=0 to send the laS client the video and audio streams specified in the @url of the LAS request, as follows:

I. When there is video in the cache: pts is closest to latestVideoPts from the currently valid buffer

The I frame starts sending the media stream

II. When there is no video in the cache: pts is closest to latestAudioPts from the current valid buffer

The audio frame starts sending a media stream

54 In a LAS request, the optional @audioOnly=true and the @startPts=0

Use @startPts=0 to send the @url specified audio stream in the LAS request to the LAS client with the following rules:

I. Sends a media stream starting from the audio frame in the current valid buffer where pts is closest to latestAudioPts

55 In a LAS request, the optional @audioOnly Missing or @audioOnly=false, at the same time

@startPts<0

Using @startPts<0, send the IaS client the video and audio streams specified in the @url in the LAS request, as follows:

I. When there is video in the cache: pts is closest to latestVideoPts from the currently valid buffer

   - |@startPts| The I frame starts sending the media stream

II. When there is no video in the cache: pts closest from the current active hold

latestAudioPts - |@startPts| The audio frame starts sending a media stream

5.6 In a LAS request, the optional @audioOnly=true and @startPts<0

Use @startPts<0 to send the @url specified audio stream in the LAS request to the LAS client with the following rules:

I. pts is closest to latestAudioPts - from the currently valid buffer - |@startPts| Audio

The frame starts sending a media stream

5.7 In a LAS request, the optional @audioOnly Default or @audioOnly=false at the same time

@startPts>0

Using @startPts>0, send the LAS client the video and audio streams specified in the @url request, as follows:

I. When there is a video in the cache and there is no timestamp fallback: Starts with the smallest I frame of pts , along pts increases in the direction in which the I frame that finds the maximum pts and satisfies the pts < = @startPts starts sending media streams; if the pts are not found < = @startPts I frame, the first one is found

pts meets the > @startPts I frames to start sending media streams; if none are found, wait

Wait for the first I frame of pts greater than or equal to @startPts to arrive before starting to send media streams

II. When there is video in the cache and there is a timestamp fallback, the media stream is sent starting with the latest I frame

III. When there is no video in the cache and no timestamp fallback: starting with the smallest audio frame of pts , in the direction of pts increment, Find the first pts > = @startPts audio frame to start sending media streams, if you can't find an audio frame that meets the pts >= @startPts , Then wait for the first one pts greater than or equal to @startPts audio frames or I frames arrive before starting to send media streams

IV. When there is no video in the cache and there is a timestamp

fallback, the media stream is sent starting from the latest audio frame

5. In a LAS request, the optional @audioOnly=true and @startPts>0

Using @startPts>0, send the LAS client the audio stream specified in the @url in the LAS request with the following rules:

I. When there is no timestamp fallback: Starting with the smallest audio frame of pts, along the direction where pts increases, find the first one /b20> The audio frame of pts >= @startPts starts sending a media stream, if you can't find an audio frame that satisfies the pts >= @startPts, then wait for the first pts to be greater than or equal to @startPts the audio frame arrives before starting to send the media stream

II. When there is a timestamp fallback, the media stream is sent starting from the most recent audio frame

5.3.9 Server-side internal back-to-source

When the edge node does not have the media stream specified by the LAS request, it needs to pull the media stream back to the source to the parent node:

I. Third-party origins must carry the @startPts field back to the source

II. The internal node returns to the source, and if it carries @startPts field, press LAS request, based on

The actual value of the @startPts is returned to the source, otherwise it is tacitly recognized

@startPts=defaultStartPts

5.3.10 Exception Handling

When @startPts>0, if a media frame with pts or less than or equal to @startPts does not exist in the valid buffer, there are two ways to handle it:

I. Wait modes such as 5.3.7 and 5.3.8

II. Error Handling Mode: When @startPts exceeds a certain threshold of the maximum pts in the valid buffer

(timeout threshold timeoutPts), the decision is that the LAS request is an error. Threshold timeoutPts

Supported configuration

6. LAS client description

The specific implementation of the LAS client is not in the scope of the LAS standard. LAS only gives recommended implementation architectures and adaptive algorithm strategies.

6.1. LAS Client Architecture

As shown in the figure, the main logic of the client includes:

MPD parsing: Responsible for parsing MPD and obtaining corresponding media information, such as each media representation URL, bitrate, id, and other information

Downloader: Responsible for downloading the media stream and passing the media data to the decoded rendering module, collecting network state information and passing it to the Adaptive Policy module

Decoding rendering: Receives media data from the downloader and decodes the rendering, while passing information such as playback-related states such as buffer size, stuttering, dropped frames, etc., to the adaptive policy module

Adaptive strategy: Combined with the alternate media representation set obtained by MPD parsing, the network status transmitted by the downloader, and the playback status of the decoding rendering module feedback, the best media representation is comprehensively determined and passed to the downloader for download switching of the media representation.

Figure 6.1 LAS client diagram

6.2. MPD Parsing

It is mainly responsible for parsing the media rendering description file and passing each element and attribute to the corresponding module.

6.3. Downloader

Based on the relevant information obtained from the adaptive policy module on the media representation, a LAS request is generated and sent to the server. In LAS, media representation is streamed, and requests need to be resend only when a media representation switch occurs.

6.3.1. LAS Request Generation

6.3.1.1. Initiating LAS Requests

When starting, according to the initial media representation specified by the business or the default initial media representation in MPD, obtain the corresponding @url of the media representation to be requested, record: startUrl.

Set the @startPts to a negative number, such as -8000, which means that it is expected to pull 8 seconds of cached data, that is, the client buffer has up to 8 seconds of buffered data, achieving a compromise between latency and network jitter buffering The specific value is specified by the client business party, for example, it is specified as initStartPts. As shown in Figure 7.2, where blue represents the keyframe, when set @startPts = -8000, the data is actually pulled to 10000ms（pts = 12000 ms ~ pts =). 22000ms）。

Therefore, the request for a started LAS is: startUrl & startPts=initStartPts.

Figure 6.2 Schematic diagram of the start @startPts

6.3.1.2. Media represents a LAS request for switching

During media playback, if the media representation of the adaptive policy output matches the media representation that is currently being downloaded, the output of the current adaptive policy is ignored without generating a new LAS request or sending the request

During media playback, if the media representation of the adaptive policy output is inconsistent with the media representation currently being downloaded, a) obtains the corresponding id according to the media representation of the adaptive policy output @url, denoted as switchUrl; b) the pts of the keyframe represented by the media output according to the adaptive policy, are obtained rtPts, denoted as switchedStartPts. Therefore, the request for a started LAS is: switchUrl & startPts=switchedStartPts.

6.3.2.Network Status Collection

While the media is downloaded, it is responsible for collecting the network status. Unlike the traditional shard-based request/download model, in LAS, streaming is used. At the network state collection level, the mode of fixed time collection point is adopted, that is, every fixed time T(ms) is used to calculate the actual amount of data downloaded during this time period S(Bytes), resulting in a bandwidth sample point$B(kbps) = S * 8/T$。Typically, T=500ms. nt.

Note: The reference implementation logic for bandwidth estimation will be given in the reference code

6.4. Decode the rendering

6.4.1.Decoding Rendering

When decoding rendering, the principle of high quality is the principle, that is, when the bitrate switch occurs, if the high and low bitrates overlap, the media representation of the high bitrate is played first. In addition, when switching between high and low bitrate rendering, it is aligned according to pts for seamless switching.

6.4.2.Playback Status Collection

When the player decodes and renders, it passes the player's state information to the adaptive policy module at regular intervals. It is recommended here to be consistent with the time interval T in 7.3.2, that is, to trigger with the same timer. The corresponding

status information includes the current video cache size, the current audio cache size, the last lag time, the last caton time, the drop rate in T(ms), and so on.

6.5. Adaptive policies

The adaptive strategy combines network status information and playback status information to dynamically select the best media representation, achieving a compromise between stutter rate, clarity, and smoothness. Two switching schemes are recommended here: GOP boundary decisions and arbitrary point decisions.

6.5.1. GOP Boundary Decisions

In LAS, media is streaming-based, and LAS clients pass through header information (for example flv_tag_header) to get the type of frame being transmitted. If it is an I frame, it means that the previous GOP download is over, and the adaptive policy is triggered to make a bitrate adaptive decision to determine the media representation of the next GOP.

Specifically, when the LAS client obtains the first frame of the i + 1 GOP through the header information as the I frame, it determines the ith frame GOP has been downloaded, and the adaptive decision strategy is triggered to make a bitrate adaptive decision to determine the id of the best media representation of the i + 1st GOP /b118>and pass its corresponding URL and the pts of the first frame of the i + 1 GOP to the download module and generate it LAS request, as shown in the following illustration, where blue represents keyframes (I frames).

Figure 6.3 Schematic diagram of @startPts based on GOP boundary switching

6.5.2.Arbitrary Point Decision Making

Arbitrary decision-making means that the adaptive policy can be triggered at any time, without waiting for the current GOP End of load.

For ease of description, defines the set of bitrates for media descriptions as $< r_<, r_>, \ldots, r_@ > (kbps)$, the bitrate represented by the currently downloaded media is r_B, the length of the GOP $D(ms)$, through the sampling point, the estimated final reliable bandwidth

is $B_{DEF}(kbps)$, the current GOP has been downloaded $d(ms)$。 Based on the dual threshold model, the LAS client presets two cache thresholds q_I and q_J, represents when the amount of media cache is greater than q_I When playback is in a safe state, the probability of stuttering is very small, and you can consider increasing the bitrate of the media representation to improve the clarity; on the contrary, when the media cache amount is less than q_J When playback is in a dangerous state, a stutter occurs

The rate is very large, and it is necessary to consider reducing the bit rate of media representation to avoid stuttering.

Considering the reference relationship of decoding, when a media representation switch occurs, it must be opened from the first frame of the GOP

Download first. Suppose the bitrate of the media representation that actually needs to be downloaded is r:

When $r = r_B$, the download continues from the current GOP, and the amount of cached data (duration) at the end of the GOP download is

$$q = q_B + D - d - (D - d) * r_B * 8/B_{DEF} \quad (1)$$

When $r != r_B$, the download needs to start with the first frame of the GOP and end at the end of the GOP download

The amount of cached data (duration) is

$$q = q_B + D - d - D * r * 8/B_{DEF} \quad (2)$$

6.5.2.1. The current amount of media cache is greater than q_I

Try increasing the bitrate of the media representation to improve clarity. The basic principle is that selecting a media indicates a download, and at the end of the current GOP download, the media cache is not less than q_IUnder the premise, select the media representation with the largest bitrate, specifically:

I.In Equation (2), if for any $r > r_B$, no $q > q$ exist$_I$Of the media said,

Then the bitrate represented by the media is kept unchanged, that is, the bitrate of the continued download is $r = r_B$The media said

II.In Equation (2), if for any $r > r_B$, if $q > q_I$Of the media said,

Then select $Q > Q_I$And the media with the largest bitrate indicates that a request for download is made. and set @startPts is the pts for the first frame of the GOP, that is, the first frame of the GOP is requested for download

6.5.2.2. Media cache is less than q_J

It is necessary to reduce the bit rate of media representations and avoid stuttering. The basic principle is that selecting a media representation for download and at the end of the current GOP download, the media cache is not less than q_JUnder the premise, select the media representation with the largest bitrate, specifically:

I. In Equation (2), if for any r, there is no $q \geq q_J$The media said that it was the most

The final requested media indicates the bit rate:

$$r_* = argmax\{(1), (2)\}$$

If $r_* = r_B$, the download of the current media continues without sending a request; otherwise, the download bitrate is equal to

r_*The media representation and settings @startPts pts for the first frame of the GOP , that is, the request for download starting from the first frame of the GOP

In Equation (2), if there is q for any r, if $q \geq q_J$The media representation, then select $q \geq q_J$And the media with the largest bitrate indicates that a request for download is made. and set

@startPts is the pts for the first frame of the GOP, that is, the first frame of the GOP is requested for download.

Figure 6.4 depicts the relationship between the value of the @startPts and whether the bitrate switches in any point switching.

Figure 6.4 Schematic diagram of switching @startPts based on any point

Appendix A. MPD Examples

Appendix B. Examples of LAS Requests

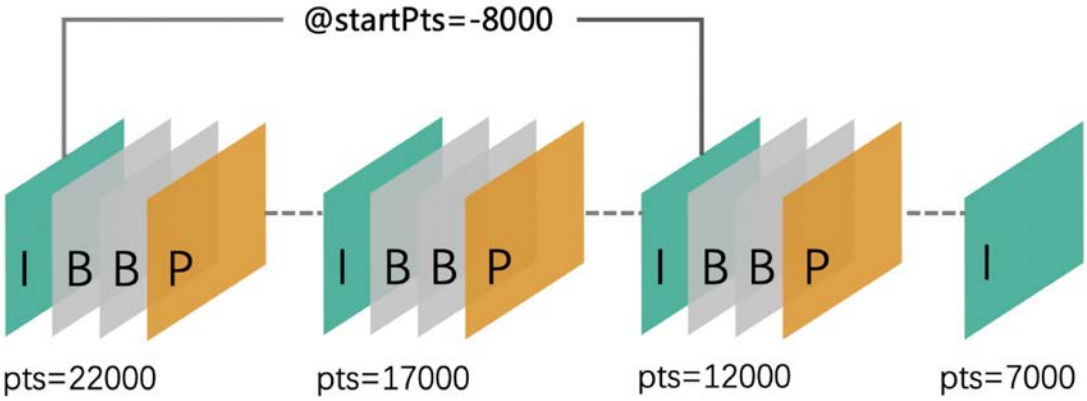

图 6.2 启播@startPts 示意图

6.3.1.1. 媒体表示切换的 LAS 请求

- 在媒体播放过程中，如果自适应策略输出的媒体表示与当前正在下载的媒体表示一致，则忽略当前自适应策略的输出，不用生成新的 LAS 请求，也不用发送请求

- 在媒体播放过程中，如果自适应策略输出的媒体表示与当前正在下载的媒体表示不一致，则 a）依据自适应策略输出的媒体表示的 id，获取对应的@url，记为 switchUrl；b）依据自适应策略输出的媒体表示的关键帧的 pts，获取@startPts，记为 switchedStartPts。
  因此，启播的 LAS 的请求为：switchUrl&startPts=switchedStartPts。

6.3.2. 网络状态收集

在媒体下载的同时，负责收集网络状态。与传统的基于分片请求/下载的模式不同，在 LAS 中，采用流式传输。在网络状态收集层面，采用固定时间采点的模式，即每隔一个固定时间$T(ms)$，统计该时间段实际下载的数据量$S(Bytes)$，从而得到一个带宽采样点$B(kbps) = S * 8/T$。典型的，$T = 500ms$。基于这些带宽的采样点，通过滤波和预测算法，估计网络的真实带宽，作为码率调整的依据。

注：带宽估计的参考实现逻辑，将在参考代码给出

6.4. 解码渲染

6.4.1. 解码渲染

解码渲染时，以高质量优先为原则，即当发生码率切换时，如果高低码率存在重叠的情况，则优先播放高码率的媒体表示。此外，在高低码率渲染切换时，依据 pts 进行对齐，从而达到无缝切换。

6.4.2. 播放状态收集

播放器在解码渲染时，每隔一定时间间隔，将播放器的状态信息传递给自适应策略模块。这里推荐与 7.3.2 中的时间间隔 T 一致，即采用同一个定时器触发。相应的状态信息包括当前视频缓存大小、当前音频缓存大小、最近一次卡顿时间、最近一次卡顿时长、$T(ms)$内的丢帧率等。

6.5. 自适应策略

自适应策略结合网络状态信息和播放状态信息，动态选择最佳的媒体表示，在卡顿率、清晰度、平滑性之间取得折中。这里推荐两种切换方案： GOP 边界决策和任意点决策。

6.5.1. GOP 边界决策

在 LAS 中，媒体是基于流式传输的， LAS 客户端通过头信息（例如 flv_tag_header）来获取当前正在传输的帧的类型。如果为 I 帧，则说明上一个GOP下载结束，此时触发自适应策略做码率自适应决策，决定下一个GOP的媒体表示。

具体而言，当 LAS 客户端通过头信息获取到第$i+1$个 GOP 的第一帧为 I 帧时，就判定第i个 GOP 已经下载完成，此时便触发自适应决策略做码率自适应决策，确定第$i+1$个 GOP 的最佳媒体表示的 id，并将其对应的 url 以及第$i+1$个 GOP 的第一帧的 pts 传递给下载模块，并生成 LAS 请求，如下图所示，其中蓝色代表关键帧（I 帧）。

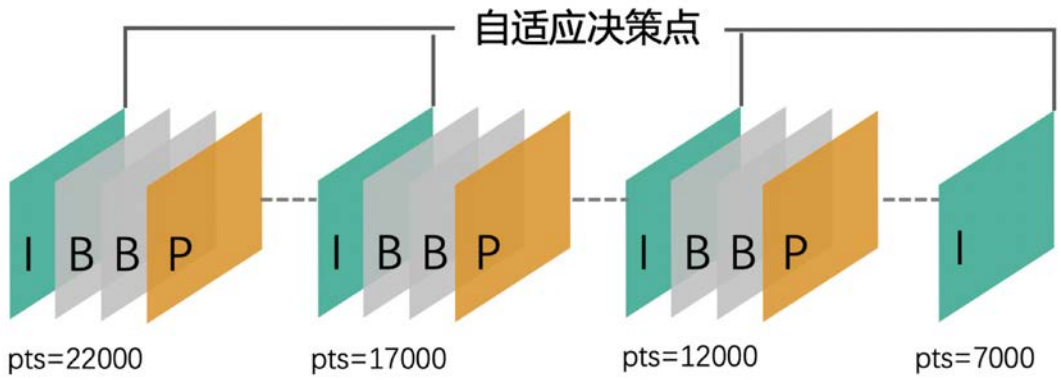

图 6.3 基于 GOP 边界切换@startPts 示意图

6.5.2. 任意点决策

任意点决策是指自适应策略可以在任意时刻触发，而不用等到当前 GOP 下载结束。

方便描述，定义媒体描述的码率集合为$< r_<, r_>, \dots, r_@ > (kbps)$，当前下载的媒体表示的码率为$r_{\mathrm{B}}$，GOP 的长度$D(ms)$，通过采样点，估计出的最终可靠的带宽为$B^{\mathrm{DEF}}(kbps)$，当前 GOP 已经下载完$d(ms)$。基于双阈值模型，LAS 客户端预设定两个缓存阈值$q_{\mathbf{I}}$和q_{J}，分别代表当媒体缓存量大于$q_{\mathbf{I}}$时，播放处于安全状态，发生卡顿的概率很小，可以考虑增大媒体表示的码率，提升清晰度；相反，当媒体缓存量小于q_{J}时，播放处于危险状态，发生卡顿的概率很大，需要考虑降低媒体表示的码率，避免卡顿。

考虑到解码的参考关系，当发生媒体表示切换时，必须从 GOP 的第一帧开始下载。假设实际需要进行下载的媒体表示的码率为r，则:

- 当$r = r_{\mathrm{B}}$，从当前 GOP 继续下载，并且 GOP 下载结束时的缓存数据量（时长）为

$$q = q_{\mathrm{B}} + D - d - (D - d) * r_{\mathrm{B}} * 8/B^{\mathrm{DEF}} \quad (1)$$

- 当$r! = r_{\mathrm{B}}$，需要从 GOP 的第一帧开始下载，并且 GOP 下载结束时的缓存数据量（时长）为

$$q = q_{\mathrm{B}} + D - d - D * r * 8/B^{\mathrm{DEF}} \quad (2)$$

6.5.2.1. 当前媒体缓存量大于$q_{\mathbf{I}}$

尝试增大媒体表示的码率，提升清晰度。基本原则为，选择一个媒体表示进行下载，并且在当前 GOP 下载结束时，在媒体缓存不低于$q_{\mathbf{I}}$的前提下，选择码率最大的媒体表示，具体而言：

- 在公式（2）中，如果对于任意的$r > r_{\mathrm{B}}$，不存在$q > q_{\mathbf{I}}$的媒体表示，则保持媒体表示的码率不变，即继续下载码率为$r = r_{\mathrm{B}}$的媒体表示
- 在公式（2）中，如果对于任意的$r > r_{\mathrm{B}}$，若存在$q > q_{\mathbf{I}}$的媒体表示，则选择满足$q > q_{\mathrm{I}}$且码率最大的媒体表示进行请求下载。并且设置@startPts 为该 GOP 第一帧的 pts，即从该 GOP 的第一帧开始请求下载

6.5.2.2. 媒体缓存量小于q_{J}

需要降低媒体表示的码率，避免卡顿。基本原则为，选择一个媒体表示进行下载，并且在当前 GOP 下载结束时，在媒体缓存不低于q_{J}的前提下，选择码率最大的媒体表示，具体而言：

I. 在公式（2）中，如果对于任意的r，不存在$q \geq q_{\mathrm{J}}$的媒体表示，则最终请求的媒体表示的码率为：

$$r^{*} = argmax\{(1), (2)\}$$

若$r^{*} = r_{\mathrm{B}}$，则继续下载当前媒体表示，不用发送请求；否则，下载码率等于r^{*}的媒体表示，并且设置@startPts 为该 GOP 第一帧的 pts，即从该 GOP 的第一帧开始请求下载

在公式（2）中，如果对于任意的r，若存在$q \geq q_{\mathrm{J}}$的媒体表示，则选择满足$q \geq q_{\mathrm{J}}$且码率最大的媒体表示进行请求下载。并且设置@startPts 为该 GOP 第一帧的 pts，即从该 GOP 的第一帧开始请求下载。

图 6.4 描述了在任意点切换中，@startPts 的取值与码率是否切换的关系。

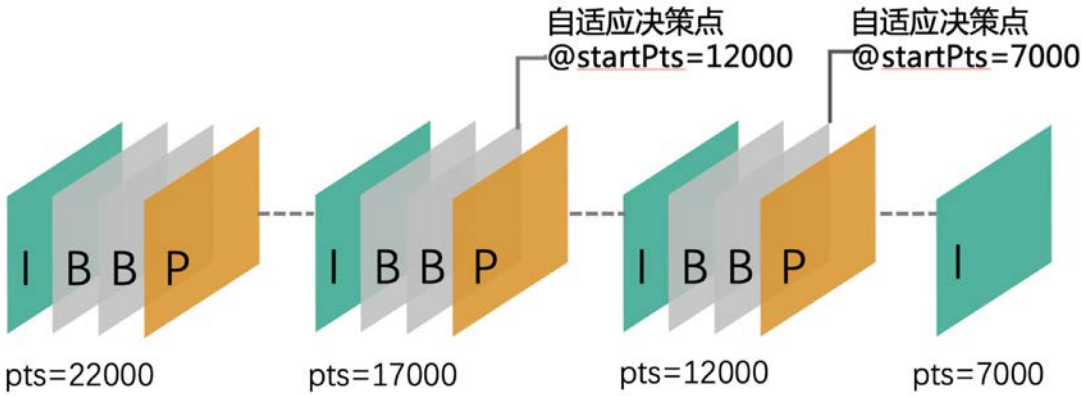

图 6.4 基于任意点切换@startPts 示意图

附录 A. MPD 示例


```
{
    "version": "1.0.0",
    "adaptationSet": [
        {
            "duration": 1000,
            "id": 1,
            "representation": [
                {
                    "id": 1,
                    "codec": "avc1.64001e,mp4a.40.5",
                    "url": "https://las-tech.org.cn/kwai/las-
test_ld500d.flv",
                    "backupUrl": [],
                    "host": "las-tech.org.cn",
                    "maxBitrate": 700,
                    "width": 640,
                    "height": 360,
                    "frameRate": 25,
                    "qualityType": "SMOOTH",
                    "qualityTypeName": "流畅",
                    "hidden": false,
                    "disabledFromAdaptive": false,
                    "defaultSelected": true
                },
                {
                    "id": 2,
                    "codec": "avc1.64001f,mp4a.40.5",
                    "url": "https://las-tech.org.cn/kwai/las-
test_sd1000d.flv",
                    "backupUrl": [],
                    "host": "las-tech.org.cn",
```

```
                    "maxBitrate": 1300,
                    "width": 960,
                    "height": 540,
                    "frameRate": 25,
                    "qualityType": "STANDARD",
                    "qualityTypeName": " 标 清 ",
                    "hidden": false,
                    "disabledFromAdaptive": false,
                    "defaultSelected": false
                },
                {
                    "id": 3,
                    "codec": "avc1.64001f,mp4a.40.5",
                    "url": "https://las-tech.org.cn/kwai/las-test.flv",
                    "backupUrl": [],
                    "host": "las-tech.org.cn",
                    "maxBitrate": 2300,
                    "width": 1280,
                    "height": 720,
                    "frameRate": 30,
                    "qualityType": "HIGH",
                    "qualityTypeName": " 高 清 ",
                    "hidden": false,
                    "disabledFromAdaptive": false,
                    "defaultSelected": false
                }
            ]
        }
    ]
}
```

附录 B. LAS 请求示例

https://las-tech.org.cn/kwai/las-test.flv

https://las-tech.org.cn/kwai/las-test.flv?audioOnly=false

https://las-tech.org.cn/kwai/las-test.flv?audioOnly=true

https://las-tech.org.cn/kwai/las-test.flv?startPts=0

https://las-tech.org.cn/kwai/las-test.flv?startPts=5678536

https://las-tech.org.cn/kwai/las-test.flv?startPts=-7000

https://las-tech.org.cn/kwai/las-test.flv?audioOnly=false&startPts=0

https://las-tech.org.cn/kwai/las-test.flv?audioOnly=false&startPts=5678536

https://las-tech.org.cn/kwai/las-test.flv?audioOnly=false&startPts=-7000

https://las-tech.org.cn/kwai/las-test.flv?audioOnly=true&startPts=0

https://las-tech.org.cn/kwai/las-test.flv?audioOnly=true&startPts=5678536

https://las-tech.org.cn/kwai/las-test.flv?audioOnly=true&startPts=-7000