



LAS: Live Adaptive Streaming

LAS1.0

2020 年 6 月 10 日

周超 快手
郭亮 快手
于冰 快手

目录

1. 前言	1
2. 架构	2
3. 媒体呈现描述	3
4. LAS 请求描述	6
5. LAS 服务描述	8
6. LAS 客户端描述	13
7. 附录	19

1. 前言

LAS (Live Adaptive Streaming) 是基于流式的直播多码率自适应标准，描述了基于流式的直播多码率自适应框架与实现规范。

参考代码: <https://github.com/KwaiVideoTeam/las>

官方网站: <https://las-tech.org.cn>

1.1. 优势

LAS 具有以下优势:

- I. 低延迟: **LAS** 实现基于流式的直播多码率自适应, 实现帧级传输, 降低端到端延迟
- II. 易扩展: **LAS** 直播多码率自适应框架, 与传统的流式非多码率框架(例如 HTTP-FLV) 完全兼容; **LAS** 支持多种协议, 例如 HTTP、QUIC、WebRTC 等
- III. 易部署: 各大云厂商均支持, 经过大规模验证, 有大量优秀案例, 例如快手等
- IV. 高效性: 请求数量少, 降低开销。**LAS** 只有在起播或发生码率切换时才需要发送请求

1.2. 核心内容

LAS 标准的核心内容包括

- I. 媒体呈现描述: 描述了基于流式的直播多码率自适应标准的基本语义元素
- II. **LAS** 请求描述: 描述了基于流式的直播多码率自适应标准中, 不同场景下的请求生成方式
- III. **LAS** 服务描述: 描述了服务端/云端支持基于流式的直播多码率自适应标准的处理逻辑
- IV. **LAS** 客户端描述: **LAS** 客户端的具体实现, 不作为 **LAS** 标准的范畴。**LAS** 仅给出推荐实现架构与自适应算法策略

2. 架构

本部分定义了 LAS 的架构图，见图 2.1。LAS 标准主要规定了基于流式的直播多码率自适应的媒体呈现描述、LAS 请求描述、LAS 服务描述与推荐的 LAS 客户端架构及直播多码率无缝自适应逻辑。

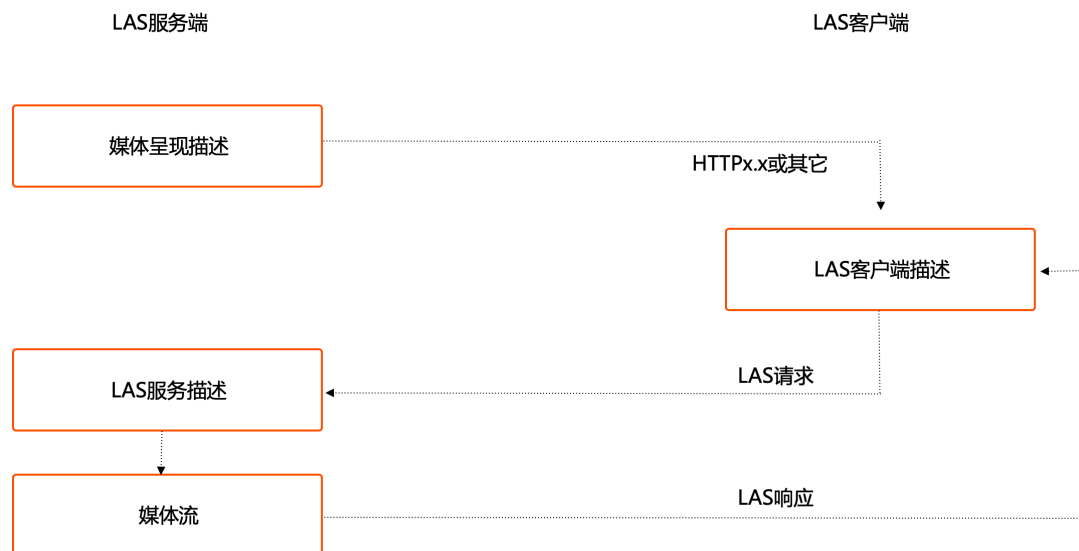


图 2.1 LAS 架构

2.1. 媒体呈现描述

媒体呈现描述是为了提供流媒体业务而对媒体呈现（Media Presentation，简称 MP）进行的格式化描述。具体而言，本部分定义了媒体流的资源标识符格式及被标识的资源在媒体呈现中的上下文。

2.2. LAS 请求描述

LAS 请求是 LAS 客户端向服务端请求媒体流的请求规范。具体而言，本部分详细定义了请求的生成规范和请求规范。

2.3. LAS 服务描述

LAS 服务描述规定了服务端对媒体流转码、缓存等特性的规范，以及对 LAS 请求逻辑的响应规范、异常处理规范。

2.4. LAS 客户端描述

LAS 客户端是 LAS 请求发起与媒体处理的载体，具体实现不属于 LAS 标准的范畴。LAS 标准仅给出推荐的客户端架构、直播多码率自适应策略、无缝切换方案等。

3. 媒体呈现描述

媒体呈现（MP）是由客户端访问、用于向用户提供流媒体服务的一组数据的集合。其中包括了已编码并可传输的媒体流及其恰当描述。媒体呈现描述（MPD）是一个包含元数据的 JSON 文档。LAS 客户端通过这些元数据来构建获取媒体流的 LAS 请求，并向用户提供流媒体服务。MPD 示例见[附录 A](#)。

3.1. 层次化数据模型

采用 JSON 格式，定义了媒体呈现描述的元素与语义。

3.1.1. MPD 的语义

MPD 元素的语义见表 3.1

表 3.1 MPD 语义表

元素或属性名称	用法	描述
MPD		媒体呈现描述的根元素
@version	必选	版本号
@adaptationSet	必须，包含至少一个 @representation	自适应集合 每个 MPD 由一个或多个 自适应集合组成

3.1.2. adaptationSet 的语义

adaptationSet 元素的语义见表 3.2

表 3.2 adaptationSet 元素的语义

元素或属性名称	用法	描述
@duration	必选	媒体流 GOP 的长度，单位为毫秒
@id	必选	adaptationSet 的唯一标识号
@representation	必选	媒体表示的集合，包含一个或多个媒体表示

表 3.3 媒体表示元素的语义

元素或属性名称	用法	描述
@id	必选	每个媒体表示的唯一标识号
@codec	必选	音视频流的编码方式
@url	必选	媒体表示的 URL 地址
@backupUrl	必选	媒体表示的备用 URL 地址
@host	可选	媒体表示的域名
@maxBitrate	必选	媒体表示的编码码率
@avgBitrate	可选	媒体表示的平均码率
@width	可选	媒体表示的宽度
@height	可选	媒体表示的高度
@frameRate	可选	媒体表示的帧率
@qualityType	可选	媒体表示的质量类型
@qualityTypeName	可选	媒体表示的质量类型展示字段
@hidden	可选	媒体表示隐藏选项 设定为 true ，则对应的媒体表示不外显，用户无法选择，只能通过自适应功能选中； 设定为 false ，则对应的媒体表示外显，用户可手动选择
@disabledFromAdaptive	可选	媒体表示自适应选项 设定为 false ，则对应的媒体表示对于自适应功能可见，能被自适应功能选中；

		<p>设定为 true，则对应的媒体表示对于自适应功能不可见，不能被自适应功能选中；</p>
@defaultSelected	可选	<p>默认档功能选项</p> <p>@defaultSelected 为 true 的媒体表示启播默认播放</p> <p>注：</p> <ol style="list-style-type: none"> 1. 所有 @representation 中，最多只能出现一个媒体表示的 @defaultSelected 为 true 2. 没有 @representation 设置 defaultSelected 为 true 时，自动选择媒体表示启播播放

4. LAS 请求描述

LAS 请求描述了客户端向服务端发送请求的具体格式与含义，LAS 请求用于向媒体服务器请求媒体流。

LAS 基本请求格式定义为媒体流 URL 地址加扩展字段的形式，即：
url&extParam。详细的 LAS 请求示例见[附录 B](#)。

表 4.1 LAS 请求元素的语义

属性	用法	描述
@url	必选	媒体表示的地址 指向媒体表示的地址，从媒体呈现描述文件中获取
@extParam	可选	扩展参数。 指定不同的请求方式，从而实现不同的功能

表 4.2 exeParam 元素的语义

属性	用法	描述
@audioOnly	可选，默认值 false	音频参数 当设定为 true 时，只拉取纯音频流，否则，拉取音频流和视频流
@startPts	可选，默认值服务器可配置。int64_t 类型	拉流位置参数 当设定 @startPts=0：非纯音频模式时，从最新的视频 I 帧开始拉流；纯音频模式时，从最新的音频帧开始拉流 当设定 @startPts>0：从 pts 等于 @startPts 的媒体帧开始拉流 当设定 @startPts<0：拉取缓存长度为

		@startPts 毫秒的媒体数据
--	--	--------------------

5. LAS 服务描述

LAS 服务描述规定了服务端（CDN 或自建媒体服务器）对媒体流转码、缓存等特性的规范，以及对 LAS 请求逻辑的响应规范、异常处理规范。

注：在本规范中，服务端和 CDN 作为可无差交换使用的替换说法。目前国内主流云厂商均支持该规范。

5.1.转码规范

转码规范要求转码服务不得对原始流中 I 帧的 pts 做任何修改。具体而言，输入任意一路原始视频，经过转码服务后，输出多路不同分辨率、码率的视频流，各视频的 I 帧严格对齐，如下图所示：

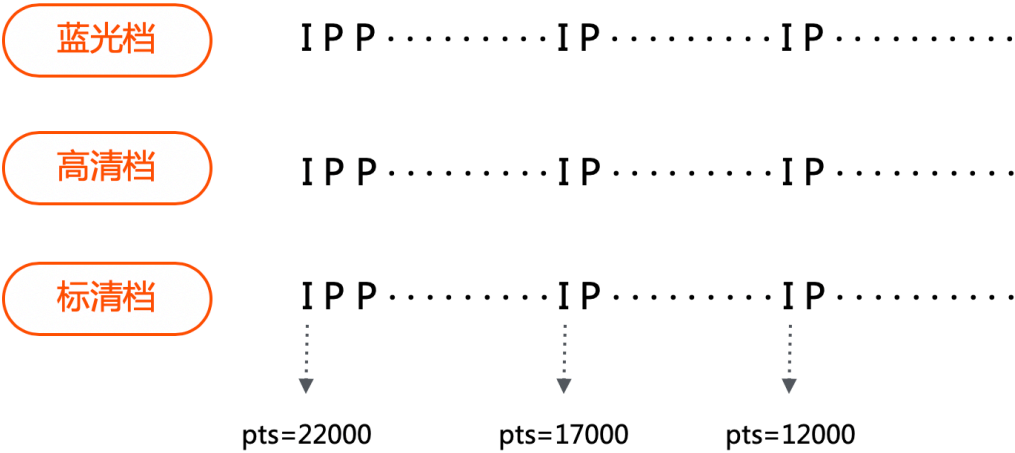


图 5.1 转码规范示意图

5.2.缓存规范

5.2.1. 缓存时长配置

描述了在服务端最少需要缓存的数据长度，单位为毫秒。

表 5.1 缓存时长元素的含义

属性	用法	描述
@maxCachedDuration	可选，默认值由服务端配置	指定服务端最多缓存的数据长度，单位 ms。

		若@ maxCachedDuration >0， 则缓存@ maxCachedDuration 毫秒的媒体数据，否则缓存时长 为默认值的媒体数据
--	--	---

5.2.2. 缓存时长计算

支持依据音视频流分开计算缓存时长。

默认使用视频流计算缓存时长，没有视频流时使用音频流计算缓存时长。

5.2.3. 时间戳回退处理

5.2.3.1. 缓存时间戳回退定义

以下任意一种情况均称为发生缓存时间戳回退

- I. 当缓存中有视频时，则以 I 帧 pts 为计算点(只考虑 I 帧的 pts)，缓存中 I 帧 pts 序列有非单调递增的情况
- II. 当缓存中没有视频时，则以所有音频 pts 为计算点(只考虑音频帧的 pts)，缓存中音频帧的 pts 序列有非单调递增的情况

5.2.3.2. 时间戳回退处理

定义一：latestVideoPts，指缓存中最新的视频帧的 pts

定义二：latestAudioPts，指缓存中最新的音频帧的 pts

定义三：有效缓存区，包括如下两种情况：a) 5.2.3.1 中情况 I 的处理：当缓存中有视频时，以视频 I 帧 pts 作为计算点，从最后一个单调递增点的 I 帧到 latestVideoPts 的所有媒体帧作为有效缓存区；b) 5.2.3.1 中情况 II 的处理：当缓存中没有视频时，以音频 pts 作为计算点，从最后一个单调递增的音频帧到 latestAudioPts 的所有音频帧作为有效缓存区。

5.3. LAS 请求逻辑处理规范

LAS 请求逻辑处理规范描述了服务端收到 LAS 请求后，具体的响应逻辑。

5.3.1. LAS 请求中，可选项 @audioOnly 缺省或 @audioOnly=false，同时 @startPts 缺省

使用 @startPts 的默认值（@startPts= defaultStartPts），向 LAS 客户端发送 LAS 请求中 @url 指定的视频流和音频流，规则如下：

-
- I. 当缓存中有视频时：从当前有效缓存区中 pts 最接近 latestVideoPts - |defaultStartPts|的 I 帧开始发送媒体流
 - II. 当缓存中没有视频时：从当前有效缓存区中 pts 最接近 latestAudioPts - |defaultStartPts| 的音频帧开始发送媒体流

5.3.2. LAS 请求中，可选项@audioOnly=true，同时@startPts 缺省

使用@startPts 的默认值（@startPts= defaultStartPts），向 LAS 客户端发送 LAS 请求中@url 指定的音频流，规则如下：

- I. 从当前有效缓存区中 pts 最接近 latestAudioPts - |defaultStartPts| 的音频帧开始发送媒体流

5.3.3. LAS 请求中，可选项@audioOnly 缺省或@audioOnly=false，同时@startPts=0

使用@startPts=0，向 LAS 客户端发送 LAS 请求中@url 指定的视频流和音频流，规则如下：

- I. 当缓存中有视频时：从当前有效缓存区中 pts 最接近 latestVideoPts 的 I 帧开始发送媒体流
- II. 当缓存中没有视频时：从当前有效缓存区中 pts 最接近 latestAudioPts 的音频帧开始发送媒体流

5.3.4. LAS 请求中，可选项@audioOnly=true，同时@startPts=0

使用@startPts=0，向 LAS 客户端发送 LAS 请求中@url 指定的音频流，规则如下：

- I. 从当前有效缓存区中 pts 最接近 latestAudioPts 的音频帧开始发送媒体流

5.3.5. LAS 请求中，可选项@audioOnly 缺失或@audioOnly=false，同时@startPts<0

使用@startPts<0，向 LAS 客户端发送 LAS 请求中@url 指定的视频流和音频流，规则如下：

- I. 当缓存中有视频时：从当前有效缓存区中 pts 最接近 latestVideoPts - |@startPts|的 I 帧开始发送媒体流

-
- II. 当缓存中没有视频时：从当前有效缓存区中 pts 最接近 latestAudioPts - |@startPts| 的音频帧开始发送媒体流

5.3.6. LAS 请求中，可选项@audioOnly=true，同时@startPts<0

使用@startPts<0，向 LAS 客户端发送 LAS 请求中@url 指定的音频流，规则如下：

- I. 从当前有效缓存区中 pts 最接近 latestAudioPts - |@startPts| 的音频帧开始发送媒体流

5.3.7. LAS 请求中，可选项@audioOnly 缺省或@audioOnly=false，同时@startPts>0

使用@startPts>0，向 LAS 客户端发送 LAS 请求中@url 指定的视频流和音频流，规则如下：

- I. 当缓存中有视频且没有时间戳回退时：从 pts 最小的 I 帧开始，沿着 pts 增大的方向，找到第一个 pts >= @startPts 的 I 帧开始发送媒体流，若找不到满足 pts >= @startPts 的 I 帧，则等待第一个 pts 大于等于@startPts 的 I 帧到来再开始发送媒体流
- II. 当缓存中有视频且有时间戳回退时，则从最新的 I 帧开始发送媒体流
- III. 当缓存中没有视频且没有时间戳回退时：从 pts 最小的音频帧开始，沿着 pts 增大的方向，找到第一个 pts >= @startPts 的音频帧开始发送媒体流，若找不到满足 pts >= @startPts 的音频帧，则等待第一个 pts 大于等于@startPts 的音频帧或 I 帧到来再开始发送媒体流
- IV. 当缓存中没有视频且有时间戳回退时，则从最新的音频帧开始发送媒体流

5.3.8. LAS 请求中，可选项@audioOnly=true，同时@startPts>0

使用@startPts>0，向 LAS 客户端发送 LAS 请求中@url 指定的音频流，规则如下：

- I. 当没有时间戳回退时：从 pts 最小的音频帧开始，沿着 pts 增大的方向，找到第一个 pts >= @startPts 的音频帧开始发送媒体流，若找不到满足 pts >= @startPts 的音频帧，则等待第一个 pts 大于等于@startPts 的音频帧到来再开始发送媒体流
- II. 当有时间戳回退时，则从最新的音频帧开始发送媒体流

5.3.9. 服务端内部回源

当边缘节点没有 LAS 请求所指定媒体流时，需要向上级节点回源拉取媒体流：

- I. 第三方源站回源，必须携带@startPts 字段
- II. 内部节点回源，如果携带@startPts 字段时则按 LAS 请求，依据@startPts 的实际值进行回源，否则依据默认值@startPts=defaultStartPts

5.3.10. 异常处理

当@startPts>0 时，如果有效缓存区不存在 pts 大于等于@startPts 的媒体帧，有两种处理方式：

- I. 等待模式：如 5.3.7 和 5.3.8
- II. 错误处理模式：当@startPts 超过有效缓存区中最大的 pts 一定阈值（超时阈值 timeoutPts）时，判定为 LAS 请求错误。阈值 timeoutPts 支持配置

6. LAS 客户端描述

LAS 客户端的具体实现，不作为 LAS 标准的范畴。LAS 仅给出推荐实现架构与自适应算法策略。

6.1.LAS 客户端架构

如图所示，客户端的主要逻辑包括：

- I. **MPD 解析**：负责解析 MPD，获取相应的媒体信息，例如各媒体表示的 URL、码率、id 等信息
- II. **下载器**：负责媒体流的下载，并将媒体数据传递给解码渲染模块；同时收集网络状态信息，并传递给自适应策略模块
- III. **解码渲染**：从下载器接收媒体数据，并解码渲染；同时将播放相关状态，例如缓冲区大、卡顿情况、丢帧情况等信息，传递给自适应策略模块
- IV. **自适应策略**：结合 MPD 解析获取的备选媒体表示集、下载器传递的网络状态、解码渲染模块反馈的播放状态等信息，综合决策最佳的媒体表示，并传递给下载器，进行媒体表示的下载切换。

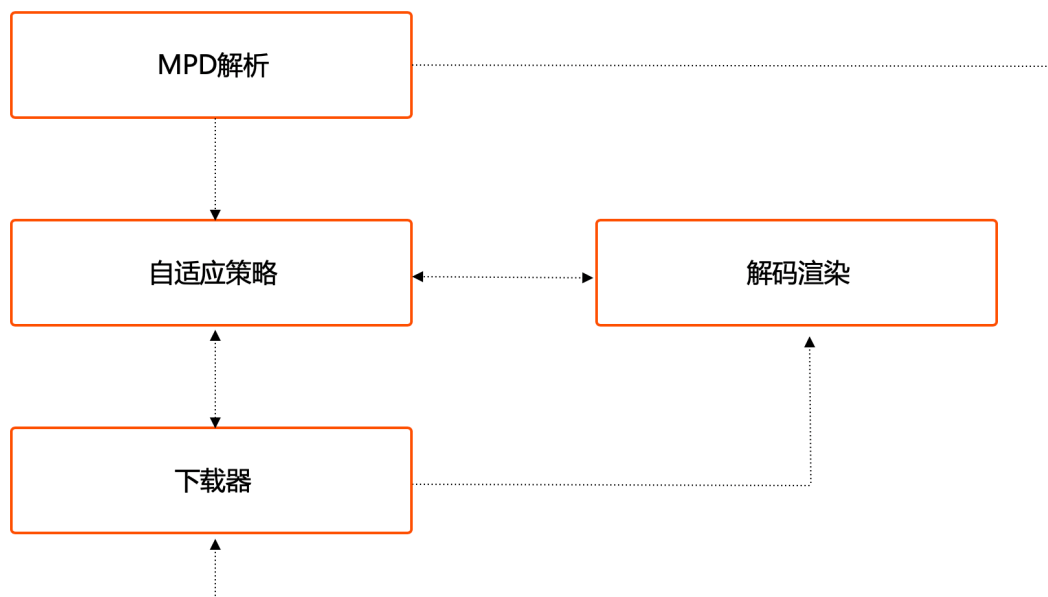


图 6.1 LAS 客户端逻辑示意图

6.2.MPD 解析

主要负责解析媒体呈现描述文件，并将各元素及属性传递给相应的模块。

6.3. 下载器

基于从自适应策略模块获取的媒体表示的相关信息，生成 LAS 请求，并发送到服务端。在 LAS 中，媒体表示采用流式传输，只有发生媒体表示切换时才需要重新发送请求。

6.3.1. LAS 请求生成

6.3.1.1. 启播 LAS 请求

启播时，依据业务指定的初始媒体表示或 MPD 中的默认初始媒体表示，获取待请求的媒体表示对应的@url，记录：startUrl。

设定@startPts 为一个负数，例如-8000，代表期望拉取 8 秒的缓存数据，也即客户端缓冲区最多有 8 秒的缓冲数据，实现延迟与网络抖动缓冲的折中，具体的取值由客户端业务方指定，例如指定为 initStartPts。如图 7.2 所示，其中蓝色代表关键帧，当设定@startPts=-8000 时，实际拉到 10000ms 的数据（ $pts = 12000\text{ ms} \sim pts = 22000\text{ ms}$ ）。

因此，启播的 LAS 的请求为：startUrl&startPts=initStartPts。

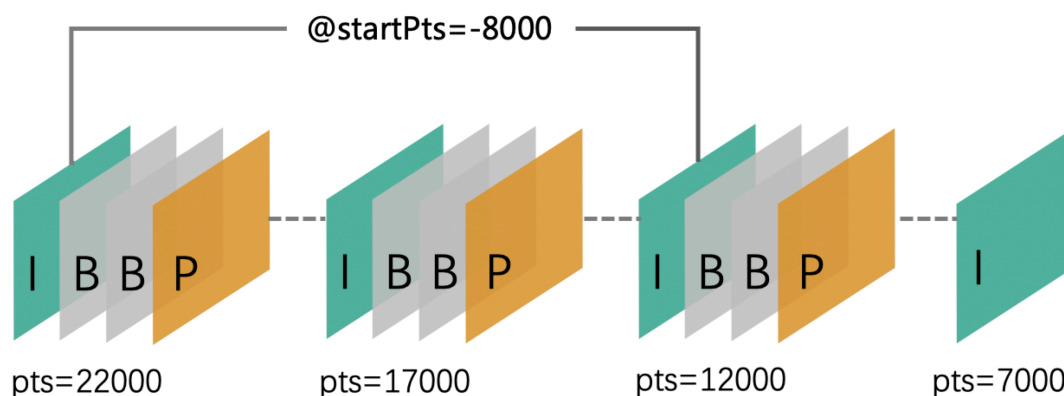


图 6.2 启播@startPts 示意图

6.3.1.2. 媒体表示切换的 LAS 请求

- I. 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

-
- II. 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）。

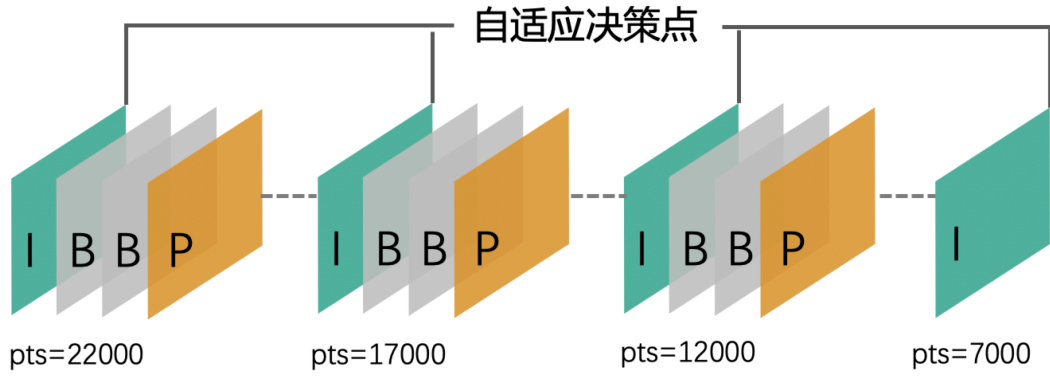


图 6.3 基于 GOP 边界切换@startPts 示意图

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 $\langle r_1, r_2, \dots, r_n \rangle (kbps)$ ，当前下载的媒体表示的码率为 r_c ，GOP 的长度 $D(ms)$ ，通过采样点，估计出的最终可靠的带宽为 $B^{est}(kbps)$ ，当前 GOP 已经下载完 $d(ms)$ 。基于 [双阈值模型](#)，LAS 客户端预设两个缓存阈值 q_h 和 q_l ，分别代表当媒体缓存量大于 q_h 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_l 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：

- I. 当 $r = r_c$ ，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

$$q = q_c + D - d - (D - d) * r_c * 8 / B^{est} \quad (1)$$

- II. 当 $r \neq r_c$ ，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

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

6.5.2.1. 当前媒体缓存量大于 q_h

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_h 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 $r > r_c$ ，不存在 $q > q_h$ 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 $r = r_c$ 的媒体表示
- II. 在公式（2）中，如果对于任意的 $r > r_c$ ，若存在 $q > q_h$ 的媒体表示，则选择满足 $q > q_h$ 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_l

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_l 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 $q \geq q_l$ 的媒体表示，则最终请求的媒体表示的码率为：

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

若 $r^* = r_c$ ，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 $q \geq q_l$ 的媒体表示，则选择满足 $q \geq q_l$ 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。

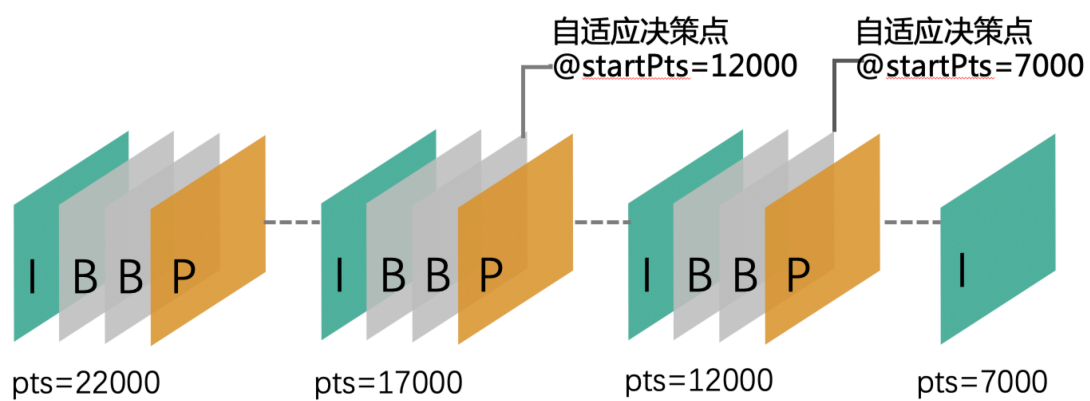


图 6.4 基于任意点切换@startPts 示意图

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>