



FAS: FLV Adaptive Streaming

FAS1.0 draft



周超 快手

郭亮 快手

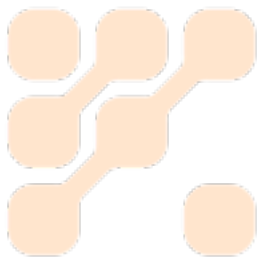
于冰 快手

本文档为 **FAS1.0** 草案，在征集业界意见进一步完善后，将推出 **FAS1.0 FCD** 版本。在此期间，欢迎业界提出宝贵的意见和建议，也欢迎加入 **FAS**，共同完善与推进 **FAS**。

联系方式: Email zhouchao@kuaishou.com WeChat 13426068306

目录

1. 前言	1
2. 架构	2
3. 媒体呈现描述	3
4. FAS 请求	5
5. 服务端规范	6
6. 媒体流	10
7. FAS 客户端	10
8. 附录	16



FAS
FLV ADAPTIVE STREAMING

1. 前言

FAS 标准以实用为基本原则，以落地为核心思想，规范了基于流式的多码率自适应框架。

1.1. 目的

FAS 标准的目的在于将快手在流式多码率自适应上的成功经验回馈予业界，同时协同业界一起打造更加强大、更加健康、更加健全的 **FAS** 生态。

1.2. 优势

FAS 与基于分片的多码率自适应框架（例如 **DASH**、**HLS**）相比，具有以下优势：

- I. 低延迟：基于流式的多码率自适应框架，不同于基于分片模式的多码率自适应框架，**FAS** 实现帧级传输，降低端到端延迟
- II. 通用性强：**FAS** 多码率自适应框架，与传统的基于流式非多码率框架（例如 **HTTP-FLV**）完全兼容
- III. 易部署：服务端（国内第三方商用 **CDN**）已全面支持，客户端（业务方）只需结合业务对传统请求（例如 **HTTP** 请求）进行扩展，生成 **FAS** 请求即可完成部署
- IV. 高效性：不同于基于分片模式的多码率自适应框架，**FAS** 基于流式，只有发生码率切换时才需要发送新请求，极大减小请求数量，降低开销
- V. 大规模实践验证：**FAS** 目前已经大规模部署且全面用于快手的直播业务（直播日活 1 亿+），其稳定性、可靠性、高效性均得到充分验证

1.3. 核心内容

FAS 标准的核心内容包括

- I. 媒体呈现描述：规范基本的语义元素，支持结合具体业务动态扩展
- II. **FAS** 请求：规范了流式多码率自适应策略中，不同场景下请求的生成方式
- III. 服务端规范：规范了 **CDN** 支持流式多码率自适应策略的处理逻辑，目前国内主流 **CDN** 厂商均已支持且经过大规模验证
- IV. **FAS** 客户端：不作为 **FAS** 标准的范畴，但 **FAS** 给出推荐架构与自适应策略，为持续打造 **FAS** 生态及开源奠定基础

2. 架构

本部分定义的 FAS 架构图见图 2.1。FAS 主要规定了基于 FLV 的动态码率自适应的媒体呈现描述、FAS 请求规范、服务端规范与推荐的 FAS 客户端架构及多码率无缝自适应逻辑。

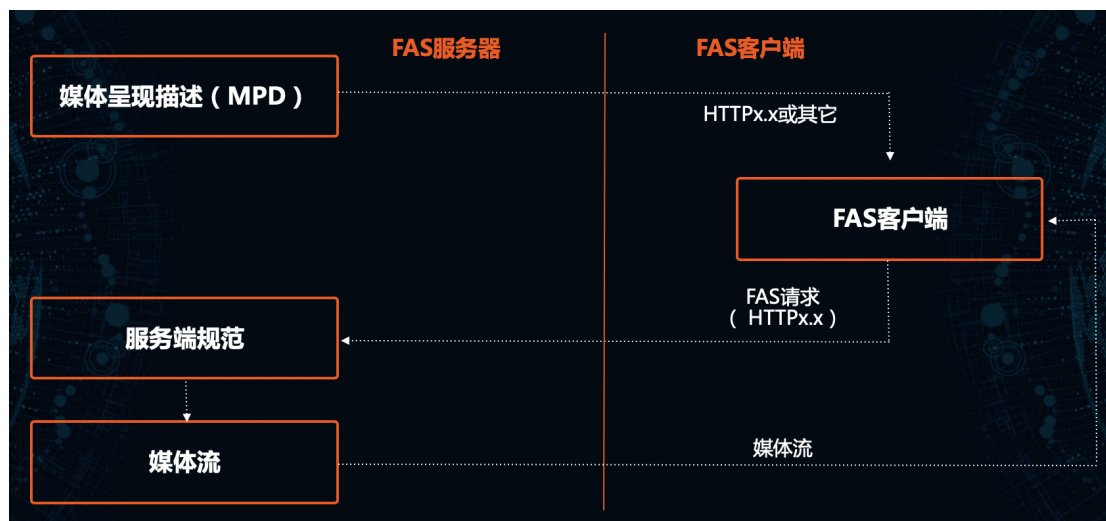


图 2.1 FAS 架构

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。资源标识符为 HTTP-URL。

2.2. FAS 请求

FAS 请求是 FAS 客户端向服务端（CDN）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. 服务端规范

服务端规范描述了服务端（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 FAS 请求逻辑的响应规范、异常处理规范。

2.4. 媒体流

媒体流是标准的 FLV 流，遵循 FLV 标准格式

2.5.FAS 客户端

FAS 客户端是媒体请求与处理的主体，具体实现不属于 FAS 标准的范畴。
FAS 标准给出推荐的客户端架构、码率自适应方案、媒体无缝切换方案。

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以及业务相关的描述。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。FAS 客户端通过这些元数据来构建获取媒体流的 FAS 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
@type	可选，默认为“dynamic”	指定服务类型。 设定为“dynamic”代表直播；设定为“static”代表 VoD
@hideAuto	可选，默认为 false	自适应功能选项。 设定为 true 时，代表关闭自适应功能，且不显示自适应选项； 设定为 false 时，打开自适应功能，且显示自适应选项

@autoDefaultSelect	可选，默认为 false	自适应功能选项。 设定为 true 时，启播默认为自适应； 设定为 false 时，启播默认为非自适应，即默认不启动自适应功能
@adaptationSet	必须，包含至少一个 @representation	描述媒体表示集合的相关元素

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
@gopDuration	必选	媒体流 GOP 的长度，单位为毫秒
@representation	必选	媒体表示的集合。 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	每个媒体表示独一无二的标识符
@codec	必选	音视频流的编码方式
@url	必选	媒体表示的 URL 地址
@bitrate	必选	媒体表示的码率
@qualityType	可选	媒体表示的质量类型。
@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；

		<p>设定为 false，则对应的媒体表示外显，用户可手动选择</p>
@enableAdaptive	可选	<p>自适应功能选项。</p> <p>设定为 true，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> <p>设定为 false，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>
	可选	<p>默认档功能选项。</p> <p>只有自适应功能选项 @autoDefaultSelect 设置为 false 时，即默认关闭自适应功能，@defaultSelect 才有效，且启播默认播放 @defaultSelect 为 true 的媒体表示</p> <p>注：所有 representation 中，最多只能出现一个媒体表示的 @defaultSelect 为 true</p>

4. FAS 请求

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

FAS 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 FAS 请求示例见[附录 B](#)。

表 4.1 FAS 请求元素的含义

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 元素的含义

属性	用法	描述
@onlyAudio	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
@fasSpts	可选，默认值 服务器可配置。 int64_t 类型	拉流位置参数。 当设定 @fasSpts=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 @fasSpts>0 ：从 pts 等于 @fasSpts 的媒体帧开始拉流 当设定 @fasSpts<0 ：拉取缓存长度为 @fasSpts 毫秒的媒体数据

5. 服务端规范

服务端规范描述了服务端（CDN 或自建媒体服务器）对媒体流转码、缓存等特性规范，以及对 FAS 请求逻辑的响应规范、异常处理规范。

在本规范中，服务端和 CDN 作为可无差交换使用的替换说法。

目前阿里、腾讯、百度、网宿、金山等均已完全支持该规范。

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：

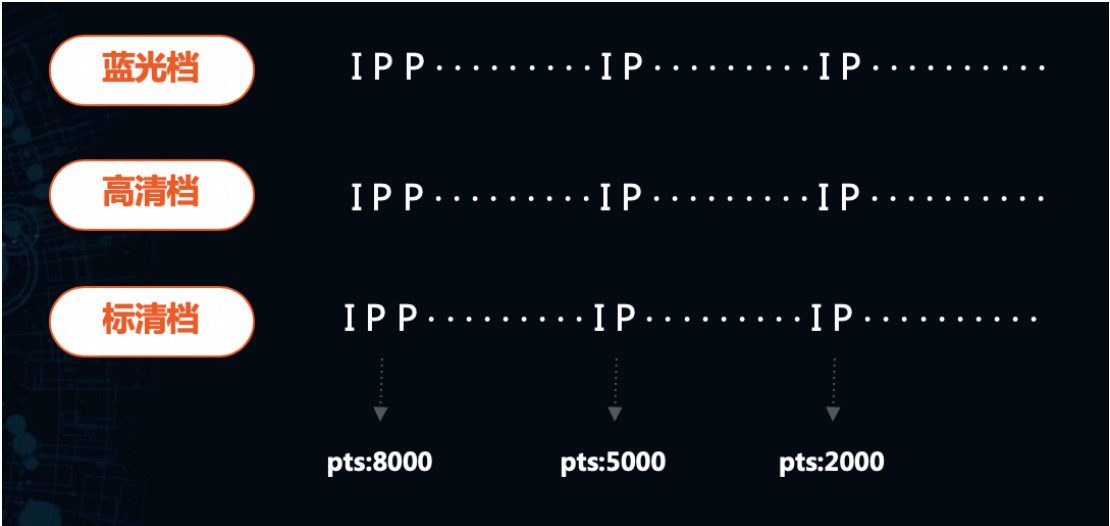


图 5.1 转码规范示意图

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
@cacheLen	可选，默认值由服务端配置	指定服务端（CDN）最多缓存的数据长度，单位 ms。 若@cacheLen>0，则缓存@cacheLen 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，例如纯音频模式。

5.2.3. 时间戳回退处理

5.2.3.1. 缓存时间戳回退定

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，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. FAS 请求逻辑处理规范

FAS 请求逻辑处理规范描述了服务端收到 FAS 请求后，具体的响应逻辑。

5.3.1. FAS 请求中，可选项 @onlyaudio 缺省或 @onlyaudio=false，同时 @fasSpts 缺省

使用 @fasSpts 的默认值（@fasSpts= defaultSpts），向 FAS 客户端发送 FAS 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 - |defaultSpts| 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 - |defaultSpts| 的音频帧开始发送媒体流

5.3.2. FAS 请求中，可选项 @onlyAudio=true，同时 @fasSpts 缺省

使用 @fasSpts 的默认值（@fasSpts= defaultSpts），向 FAS 客户端发送 FAS 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 - |defaultSpts| 的音频帧开始发送媒体流

5.3.3. FAS 请求中，可选项@onlyaudio 缺省或@onlyaudio=false，同时@fasSpts=0

使用@fasSpts=0，向 FAS 客户端发送 FAS 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. FAS 请求中，可选项@onlyAudio=true，同时@fasSpts=0

使用@fasSpts=0，向 FAS 客户端发送 FAS 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. FAS 请求中，可选项@onlyaudio 缺失或@onlyaudio=false，同时@fasSpts<0

使用@fasSpts<0，向 FAS 客户端发送 FAS 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 - |@fasSpts|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 - |@fasSpts|的音频帧开始发送媒体流

5.3.6. FAS 请求中，可选项@onlyAudio=true，同时@fasSpts<0

使用@fasSpts<0，向 FAS 客户端发送 FAS 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 - |@fasSpts|的音频帧开始发送媒体流

5.3.7. FAS 请求中，可选项@onlyaudio 缺省或@onlyaudio=false，同时@fasSpts>0

使用@fasSpts>0，向 FAS 客户端发送 FAS 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 >= @fasSpts 的 I 帧开始发送媒体流，若找不到满足 pts >= @fasSpts 的 I 帧，则等待第一个 pts 大于等于@fasSpts 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 >= @fasSpts 的音频帧开始发送媒体流，若找不到满足 pts >= @fasSpts 的音频帧，则等待第一个 pts 大于等于@fasSpts 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. FAS 请求中，可选项@onlyAudio=true，同时@fasSpts>0

使用@fasSpts>0，向 FAS 客户端发送 FAS 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 >= @fasSpts 的音频帧开始发送媒体流，若找不到满足 pts >= @fasSpts 的音频帧，则等待第一个 pts 大于等于@fasSpts 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（CDN）内部回源

当边缘节点没有 FAS 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@fasSpts 字段
- II. CDN 内部节点回源，如果携带@fasSpts 字段时则按 FAS 请求，依据 @fasSpts 的实际值进行回源，否则依据默认值 @fasSpts=defaultSpts

5.3.10. 异常处理

当@fasSpts>0 时，如果有效缓存区不存在 pts 大于等于@fasSpts 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@fasSpts 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 FAS 请求错误。阈值 timeoutPts 支持配置。

6. 媒体流

媒体流是标准的 FLV 流，遵循 FLV 标准

7. FAS 客户端

FAS 客户端的具体实现不属于 FAS 标准的范畴，该规范给出一种推荐架构与码率自适应策略。

7.1. FAS 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：**从下载器接受媒体数据，并解码渲染；同时将播放相关状态，例如缓冲大小，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。

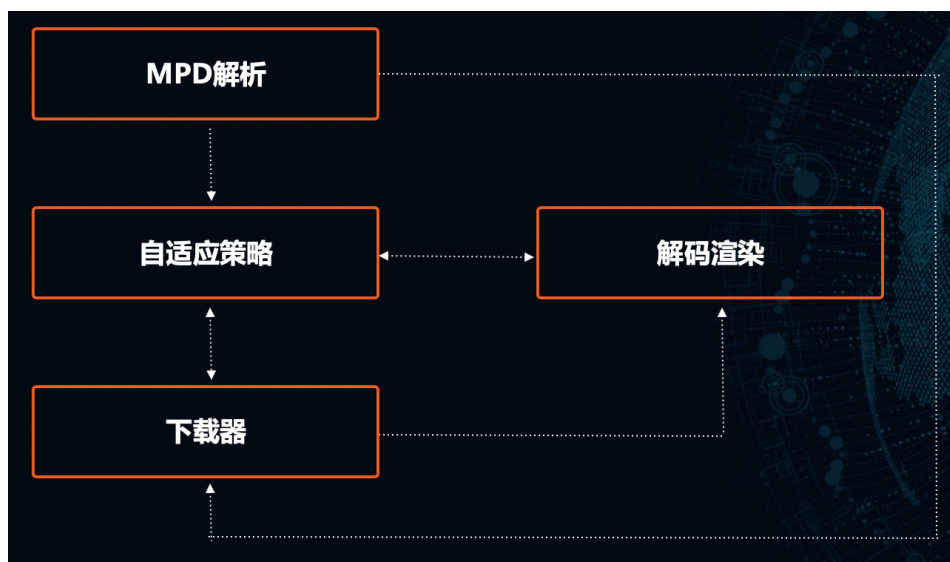


图 7.1 FA 客户端逻辑示意图

7.2.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。

7.3.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 FAS 请求，并发送到服务端（CDN）。与传统的基于分片请求/下载的模式不同，在 FAS 中，媒体表示采用流式传输，且只有发生媒体表示切换时才需要重新发送请求。

7.3.1. FAS 请求生成

7.3.1.1. 启播 FAS 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：urlStart。

设定@fasSpts 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 fasSptsInit。如图 7.2 所示，其中蓝色代表关键帧，当设定@fasSpts=-8000 时，实际拉到 9000ms 的数据（pts=3000~pts=12000）。

因此，启播的 FAS 的请求为：urlStart&fasSpts=fasSptsInit。

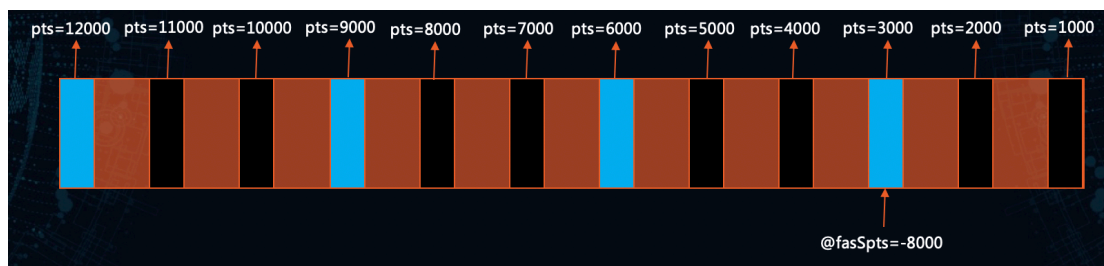


图 7.2 启播@fasSpts 示意图

7.3.1.2. 媒体表示切换的 FAS 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 FAS 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 urlSwitch；b) 依据自适应策略输出的媒体表示的关键帧的 pts，获取@fasSpts，记为 fasSptsSwitch。
因此，启播的 FAS 的请求为：urlSwitch&fasSpts=fasSptsSwitch。

7.3.2. 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 FAS 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 $T(\text{ms})$ ，统计该时间段实机下载的数据量 S (Bytes)，从而得到一个带宽采样点 $B(\text{kbps})=S*8/T$ 。典型的， $T=500\text{ms}$ 。基于这些带宽的采样点，通过滤波和预测算法，从而估计网络的真实带宽，作为码率调整的依据。

7.4. 解码渲染

7.4.1. 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。

7.4.2. 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。

7.5. 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。

7.5.1. GOP 边界决策

在 FAS 中，媒体是流式传输的，FAS 客户端通过 `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。

当 FAS 客户端通过 `flv_tag_header` 获取到 `GOPi+1` 的第一帧为 I 帧时，就判定 `GOPi` 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 `GOPi+1` 的最佳媒体表示的 id，并将其对应的 url 以及 `GOPi+1` 的第一帧的 pts（从 `GOPi+1` 的第一帧的 `flv_tag_header` 获取），传递给下载模块，生成 FAS 请求，如下图所示，其中蓝色代表关键帧（I 帧）。

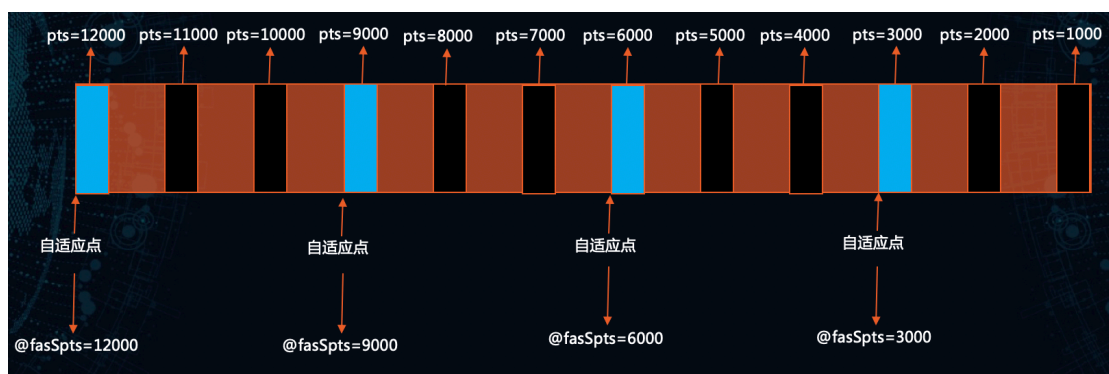


图 7.3 基于 GOP 边界切换@fasSpts 示意图

7.5.2. 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](#)，FAS 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

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

7.5.2.1. 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 $@fasSpts$ 为该 GOP 第一帧的 pts，即从该 GOP 的第一帧开始请求下载

7.5.2.2. 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@fasSpts 为该 GOP 第一帧的 pts，即从该 GOP 的第一帧开始请求下载

- II. 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@fasSpts 为该 GOP 第一帧的 pts，即从该 GOP 的第一帧开始请求下载。

图 7.4 描述了在任意点切换中，@fasSpts 的取值与码率是否切换的关系。

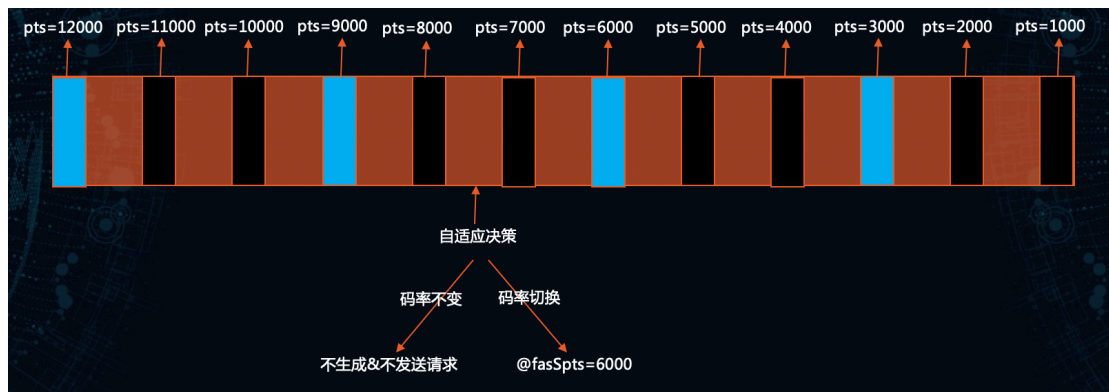


图 7.4 基于任意点切换@fasSpts 示意图

附录 A. MPD 示例

```
{
  "version": "2.0",
  "type": "dynamic",
  "hideAuto": false,
  "adaptationSet": {
    "gopDuration": 2000,
    "representation": [
      {
        "id": 0,
        "codec": "avc1.64001e,mp4a.40.2",
        "url": "http://fas/media/url_500kbps",
        "bitrate": 500,
        "averageBitrate": 500,
        "qualityType": "SMOOTH",
        "hidden": false,
        "enableAdaptive": true,
        "defaultSelect": false
      },
      {
        "id": 1,
        "codec": "avc1.64001e,mp4a.40.2",
        "url": "http://fas/media/url_900kbps",
        "bitrate": 900,
        "averageBitrate": 900,
        "qualityType": "STANDARD",
        "hidden": false,
        "enableAdaptive": true,
        "defaultSelect": false
      },
      {
        "id": 2,
```

```
    "codec": "avc1.640028,mp4a.40.2",
    "url": "http://fas/media/url_1500kbps",
    "bitrate": 1500,
    "averageBitrate": 1500,
    "qualityType": "HIGH",
    "hidden": false,
    "enableAdaptive": true,
    "defaultSelect": false
  }
]
}
}
```



FAS

附录 B. FAS 请求示例

```
http://fas/media/urlink
http://fas/media/urlink&onlyAudio=false
http://fas/media/urlink&onlyAudio=true
http://fas/media/urlink&fasSpts=0
http://fas/media/urlink&fasSpts=5678536
http://fas/media/urlink&fasSpts=-7000
http://fas/media/urlink&onlyAudio=false&fasSpts=0
http://fas/media/urlink&onlyAudio=false&fasSpts=5678536
http://fas/media/urlink&onlyAudio=false&fasSpts=-7000
http://fas/media/urlink&onlyAudio=true&fasSpts=0
http://fas/media/urlink&onlyAudio=true&fasSpts=5678536
http://fas/media/urlink&onlyAudio=true&fasSpts=-7000
```