

PPTV HTTP 流媒体协议

(PPTV HTTP Streaming Protocol)

修订记录

| 日期 | 作者 | 版本 | 主要内容 | 备注 |
|----------|-----|-----|------|----|
| 2012-2-9 | 郭春茂 | 1.0 | 初稿 | |

目录

| | |
|------------------------------|------------------------------|
| 修订记录 | 1 |
| 目录 | 2 |
| 1 前言 | 1 |
| 2 消息格式 | 1 |
| 2.1 HTTP 头域定义 | 1 |
| 3 控制方法定义 | 2 |
| 3.1 正常播放 | 2 |
| 3.2 拖动播放 | 2 |
| 3.2.1 FLV 格式拖动参考 | 4 |
| 3.2.1.1 前提: | Error! Bookmark not defined. |
| 3.2.1.2 Metadata 参数介绍: | Error! Bookmark not defined. |
| 3.2.1.3 拖动位置获取: | Error! Bookmark not defined. |
| 3.2.2 TS 格式拖动参考 | 5 |
| 3.3 暂停播放 | 3 |
| 3.4 恢复播放 | 3 |
| 3.5 停止播放 | 3 |
| 4 响应码 | 3 |
| 4.1 成功响应 2XX | 3 |
| 4.2 错误响应 5XX | 3 |

1 前言

PPTV HTTP 流媒体协议是基于 HTTP 协议（RFC 2616）的流媒体播放控制协议，该协议能够工作在现有的 CDN 体系上，支持基本的播放控制——拖动、暂停。

2 消息格式

PPTV HTTP 流媒体协议的消息格式与 HTTP 协议消息格式相同，每个消息包含一个消息头和消息体。消息有两种类型：请求消息和应答消息。

请求消息必须是 HTTP GET 方法，其 URI 描述具体的音视频流。

应答消息是 HTTP GET 方法的应答。

2.1 HTTP 头域定义

- Accept-Ranges

服务器应答中包含此头域，参数可以是“none”或者“bytes”，如果没有该头域，默认为“bytes”。如果指定了参数值是“none”，则该视频流不支持拖动。

- Content-Length

服务器应答中必须包含此头域，表示响应的体部数据总长，单位为 byte。

- Content-Type

服务器应答中必须包含此头域，表示所传输数据的格式。

- Content-Range

服务器处理带 Range 的请求时，应答中必须包含此头域，表示响应中接收数据的范围。

- Range

客户端请求部分数据时，附带该头域，表示请求数据的范围，单位为 byte。

3 控制方法定义

3.1 正常播放

客户端通过 HTTP 的 GET 方法向媒体服务器获取视频流，如果接收的数据达到 Content-Length 字段的值时，则播放结束，中断与媒体服务器的 TCP 连接。

例如：

```
GET /a.ts HTTP/1.1
Host: 192.168.45.22
Connection: keep-alive

HTTP/1.1 200 OK
Accept-Ranges: bytes
Content-Length: 21460839
Connection: keep-alive
Content-Type: video/ts

{Media Data}
```

3.2 拖动播放

通过新的 HTTP 带 Range 的 GET 方法请求媒体数据，Range 的开始位置如何计算不在本协议描述范围。

服务器在应答消息中必须有 Content-Range 头域。

如果服务器在先前的 Accept-Ranges 头域中返回 none，则服务器不支持拖动播放，客户端应该禁止用户拖动。

例如：

```
GET /a.ts HTTP/1.1
Host: 192.168.45.22
Range: bytes=20709376-
Connection: keep-alive

HTTP/1.1 200 OK
Content-Length: 751463
Accept-Ranges: bytes
Content-Range: bytes=20709376-21460838/21460839
Connection: keep-alive
Content-Type: video/ts

{Media Data}
```

3.3 暂停播放

在不断开 TCP 连接的情况下，暂停接收数据，通过阻塞的方式暂停播放。

3.4 恢复播放

在上述暂停的基础上，恢复媒体流数据的接收。

3.5 停止播放

客户端主动断开与媒体服务器的 TCP 连接，停止接收流数据，实现停止播放。

4 响应码

4.1 成功响应 2XX

200 正确响应

206 Range 请求的正确响应

4.2 错误响应 5XX

500 请求失败

例如：

```
HTTP/1.1 500 Internal Server Error
Content-Type: text/xml
Content-Length: 100

<?xml version="1.0" encoding="utf-8"?>
<root>
  <category>Demuxer</category>
  <vaule>404</vaule>
  <message>not find film</message>
</root>
```

5 附录

5.1 FLV Range 位置算法

FLV 文件要流化处理，也就是在 FLV 文件中加入 metadata tag 记录一个关键帧索引表，一般用 flvtool2, yamdi 等开源工具生成，当然也可以自己开发工具生成索引表

FLV 的 metadata tag 由服务器生成主要记录每个关键帧的时间和对应的文件偏移量，并且保存在文件中的第一个 tag 中，以下是索引表数据：

| KeyFrame Time | KeyFrame offset |
|---------------|-----------------|
| 1571 | 0 |
| 255073 | 4.933 |
| 432405 | 10.933 |
| | |

表中的第一行表示的是影片的第一个视频关键帧在文件中的偏移量和对的时间戳，第二行表示的是影片的第二个视频关键帧在文件中的偏移量和对的时间戳，依次类推记录影片中所有视频关键帧的时间戳和偏移量，不会记录音频的帧的偏移量。

播放器根据时间总长，按进度条比例获取大概的拖动时间点，查找 metadata 记录的关键帧索引表，找到离这个时间点最近的关键帧在影片中的偏移量，该偏移量则是 Range 的位置。

5.2 TS Range 位置算法

该算法假定 TS 文件是固定码率（CBR），并且视频帧流中的同步帧的密度足够大。

客户端通过 HTTP 的 GET 方法获取 TS 流时，可以通过 Content-Length 字段获取文件总长度，通过进程条的比例可以算出需要拖动的偏移位置，该偏移量则是 Range 的位置了。

由于每次按文件 offset 拖动的时候，下载的数据可能不是完整的 TS 包，需要播放器做过滤，从找到的第一个完整的 ts 包开始解封装解码，如果要保证解码的完整性，最好是在找到第一个视频关键帧时开始解码