



创意感动生活
The Creative Life



HLS协议介绍及系统实现

张雷(设计6所)

692463260@qq.com



关于本培训

- 主题：HLS协议介绍及系统实现
- 目标：帮助大家熟悉HLS协议相关知识
- 说明：
 - 闻道有先后，如是而已。
 - 大家共同参与进来吧

培训大纲

1. HLS简介
 - 5W 1H
2. 系统实现与特性
 - 从前端到终端的拓扑
3. HLS协议解析($\leq v3$)
 - 对应IETF Internet-Draft中的内容
4. 新特性(v4, 5, 截至2013-4)
 - 对应IETF Internet-Draft中的内容
5. Q & A
 - 前面每部分均有总结环节，可用于提问
 - 也可在此环节提问

1. HLS简介

- Who: 苹果公司
- When: 2009-05
- What: 推出HLS规范 (IETF草稿、RFC申请) 及相关资源。
- What: 基于http的流媒体传输协议
 - 通过HLS, CP/SP可以使用普通的web server来向iPad/iPod/iPhone/Mac等用户提供优质的视频服务体验
 - 支持直播、点播
 - 支持内容保护、加密节目 (付费)
 - 支持… (后续介绍)

1. HLS简介

- Where can I get help?
 - <http://tools.ietf.org/pdf/draft-pantos-http-live-streaming-11.pdf>
 - <https://developer.apple.com/resources/http-streaming/>
 - 寻找、阅读开源软件
(gstreamer/vlc/android) 相关部分的实现

1. HLS简介-Why is HLS?

- 首先看苹果有哪几个选择
 - HTTP渐进下载流媒体播放
 - Flash
 - RTSP/RTCP/RTP
 - 开发新的协议，基于新的协议建立生态

移动互联网特点

- 位置属性
- 终端多样
- 网络环境复杂多变



1. Why HLS—but not HTTP

- HTTP仅适用于低码率、短视频、点播应用
- CP/SP部署直播业务很难
- server不支持故障备援，而这恰恰对移动互联网很重要
- Client端不支持对网络环境的自适应



1. Why HLS—but not flash

- Client端cpu占用率高，耗电多，不适用于ipad/iphone(后续有对比说明)
- CP/SP部署成本较高
- server没有失效备援技术
- Adobe专利技术，苹果没有话语权



1. Why HLS—but not RTSP

- Server没有失效备援
- 很难与传统的CDN进行集成
- CP/SP部署起来相对困难，
 - 需要额外的服务器，增加了CP/SP成本
 - 需要安装额外的服务软件，配置复杂，增加了部署难度
- 不能穿透防火墙，NAT等，不能为位于其后的用户提供服务

1. Why HLS—What is needed

- 针对移动互联网开发新的协议，建立生态，需要考虑哪些？
 - 可以穿透防火墙、NAT，Router→几乎确定新协议要基于HTTP
 - 方便CP/SP部署，方便与传统的CDN集成→又为使用http增加了砝码
 - 可以支持点播/直播
 - server支持故障备援
 - 尽量小的client端cpu占用率，节约电能
 - Client端支持自动带宽适应
 - 当然，还要支持收费节目，无利不起早

1. How-How to achieve those

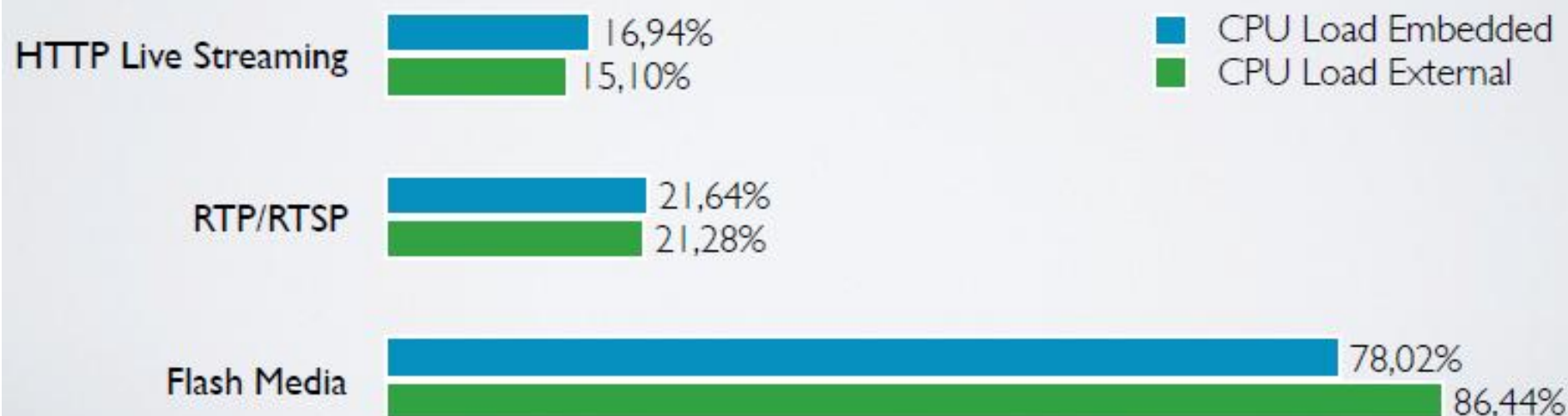
- http本身的特性
 - 穿透防火墙
 - 方便CP/SP部署，方便与传统的CDN集成
- HLS协议本身提供的特性
 - 可以支持点播/直播
 - server支持故障备援
 - 尽量小的client端cpu占用率，节约电能
 - Client端支持自动带宽适应
 - 支持收费节目
- 苹果提供的工具
- 相关的开源软件

1. HLS简介-So, it is called...

- 首先，它基于http；其次，它支持直播
- So it is called **HTTP Live Streaming**
- 优点一：针对移动互联网，增强用户体验
 - 低的client端cpu占用率，节能(见下一页)
 - 可以穿透防火墙，为世界任一角落的用户提供同样的服务
 - 当用户从某个网络环境转换到另一个时，提供无间断的服务
- 优点二：方便CP/SP部署，方便与传统CDN集成，降低了CP/SP转换和部署的成本

1. HLS简介-耗电量对比

- Average CPU Load from 120 samples taken over 40sec
- Ran on a 1.8 Ghz dual core MacBook (MacOS 10.6.2)



1. Every coin has two sides

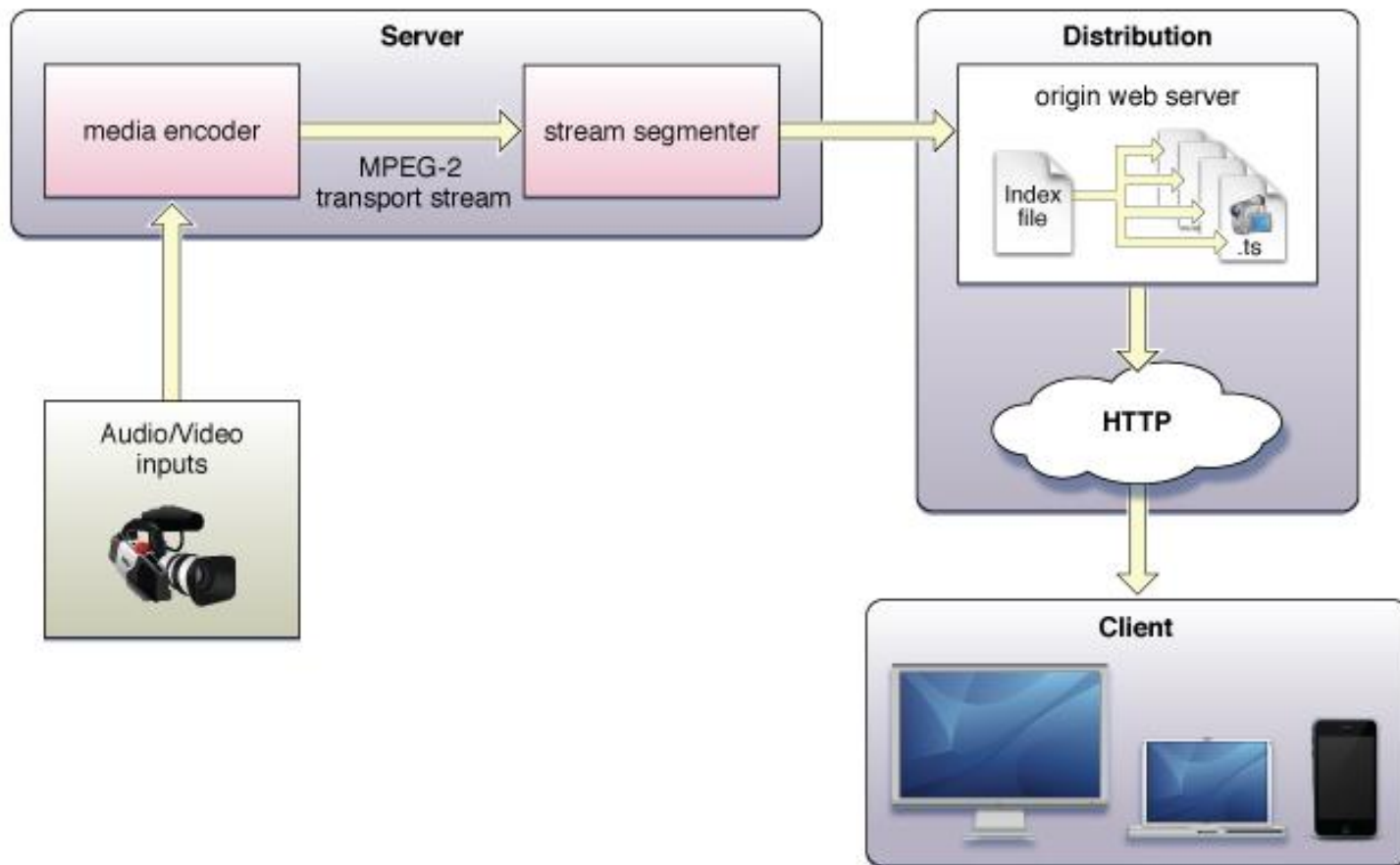
- HLS也有一些不足
- 缺点一：直播相比现场有一定的延时
(后续有阐述)
- 缺点二：貌似目前只有苹果一家提供的部署工具（需确认），与苹果的平台
(Intel-based Mac)绑定太紧
- 缺点三：暂时没有想到

1. 总结

- 5W 1H
- Q & A



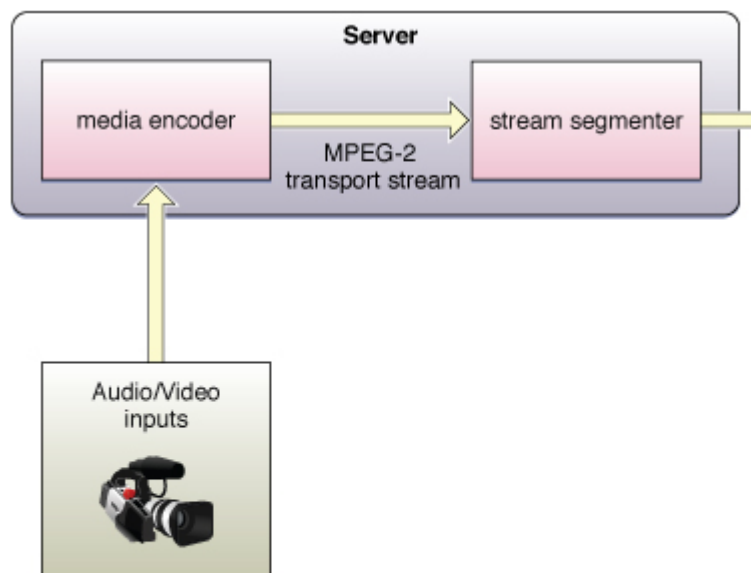
2. 系统架构



2. 系统架构

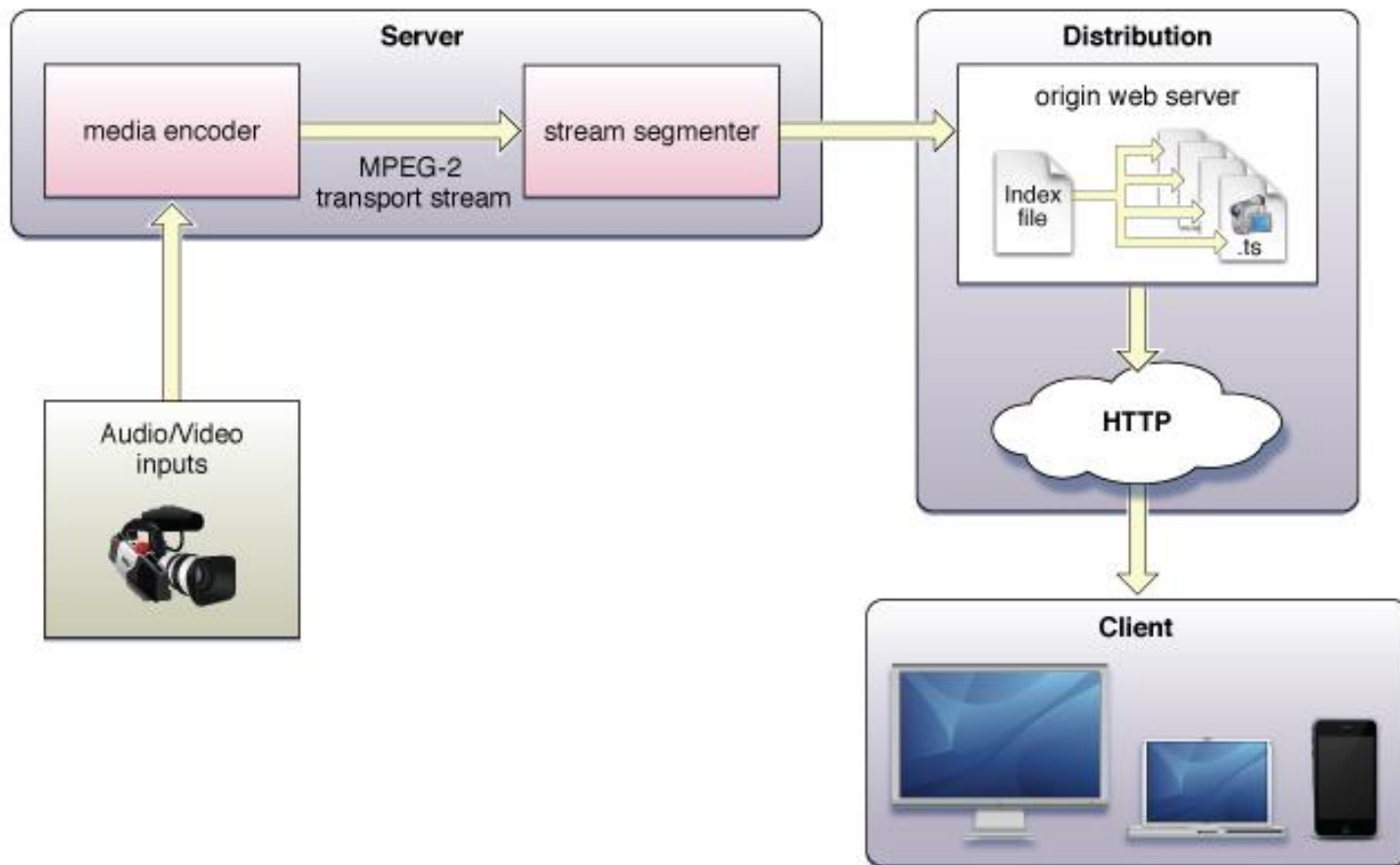
- **Server Component**
 - Encode the source stream, and encapsulate them to ts segments
- **Distribution Component**
 - Store ts segments
 - Response to client' s requests
- **Client Software**
 - fetch the appropriate resource, then finish the play task
 - In iOS system, this part is embeded already
 - Supported by Android Framework/gstreamer/vlc

2. 系统架构

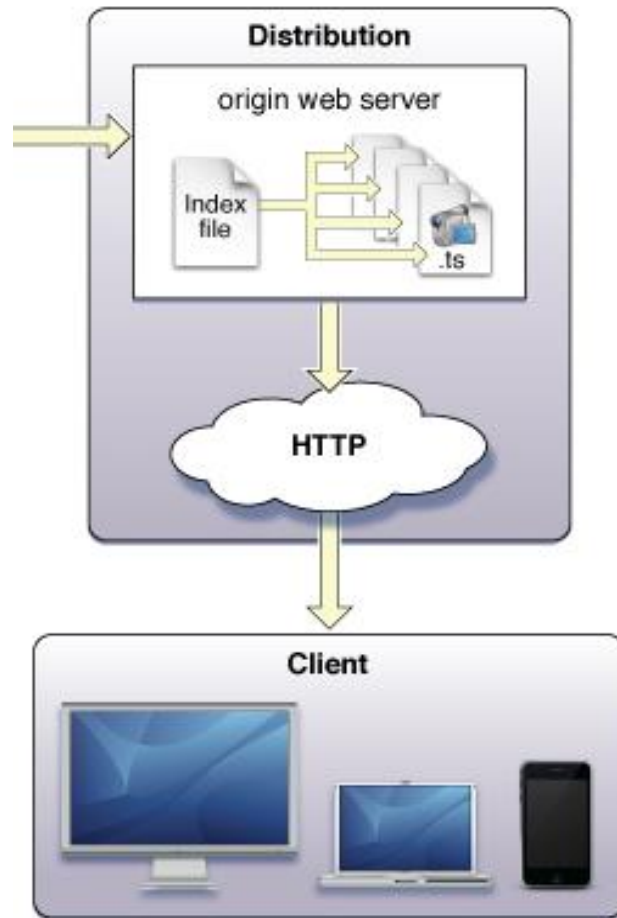


- **A/V inputs**
 - Live or prerecorded source
- **Media encoder**
 - Output the MPEG2 ts stream
- **Stream segmenter**
 - Cut ts stream to small segments
 - Create/update .m3u8 index file
 - .m3u8 is just a plain text file

2. 系统架构



2. 系统架构

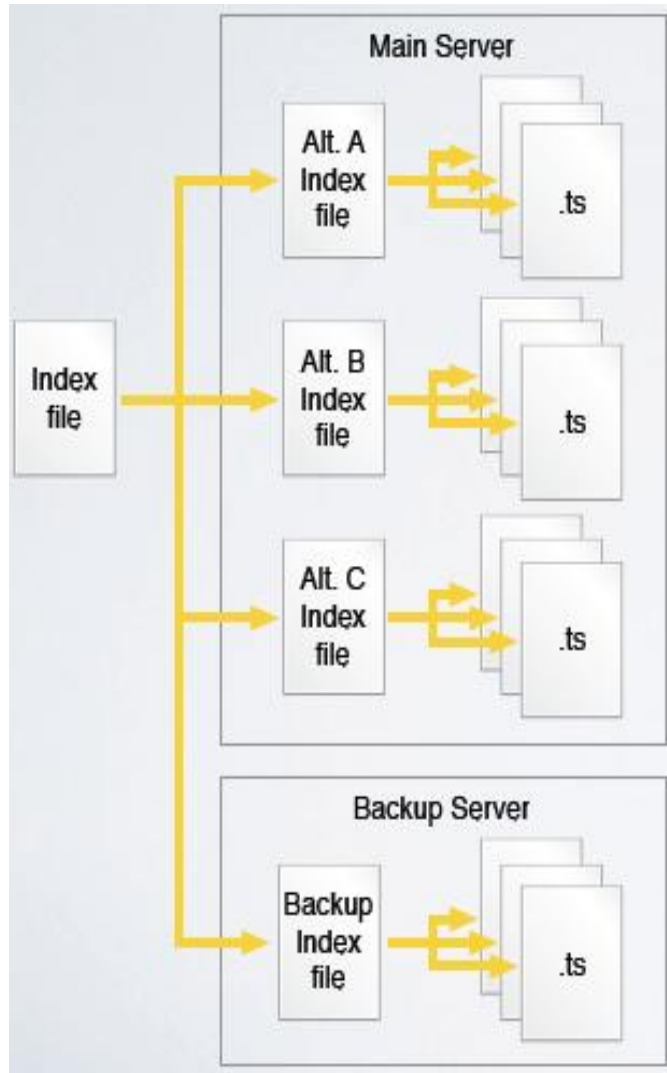


- **Distribution**
 - Ordinary web server or EDGE network based on HTTP
- **Client**
 - First fetch index file
 - Then parse it
 - Fetch ts segments in sequence, store in buffer
 - Play data in the buffer

2. 系统特性

- Support both VoD and Live broadcast
- **Content Protection**
 - Encryption: AES-128
 - Key file list in .m3u8 file
 - Key file can be protected by CP' s session-based authentication scheme
- **Stream Alternatives**
 - Failover protection
 - For different network environment
 - For different devices

2. Stream Alternatives



- Master index file
 - Organize all alternatives
- Alt. A/Alt. B/Alt. C
 - 3Mbps/2Mbps/1Mbps
- Backup server & index file
 - 64kbps
 - Separate physical server, maybe in different city

2. 系统特性

- 支持什么编码格式？
 - HLS本身没有对编码做限制；
 - 苹果提供的工具及iPad/iPhone/Mac生态目前只支持h264的视频, AAC/MP3的音频
- 支持什么码率？
 - HLS本身不限制
 - CP在确定内容的码率时应当考虑到目标client平台及所处的网络环境
 - 低到64kbps 高到3Mbps的流均被iOS支持
 - 建议CP提供64Kbps的流作为client的故障备援

2. 系统特性

- Client如何在Alt. 流间进行切换？
- iOS框架在播放流时会适时获取有效的带宽（下一页介绍思路）
- 如果有更高质量的流且带宽支持它，则切换到高质量的流；
- 如果当前的带宽不能支持当前播放的流且有一个低码率的流，则切换到低码率的流

2. 检测有效带宽思路

- 在下载媒体段数据的过程中进行检测
- 我们知道下载到的数据总量
- 我们也知道下载所花费的时间
- $\text{bps} = \text{data_sum} / \text{system_time_elapsed};$

2. 总结

- 整个系统分为哪些部分？
- 各部分的功能？
- Q & A



3. HLS协议解析(<=v3)

Then I will introduce the .m3u8 file format by using it (according to the IETF drafts)

1. VoD
2. Live broadcast
3. Encrypted program
4. Stream alternatives



3. VoD based on HLS

#EXTM3U //标识这是个m3u8文件
#EXT-X-TARGETDURATION:5220//下面列
出的段的时长不超过此值
#EXTINF:5220, //下方url的info
<http://media.example.com/entire.ts>
//这是一个段
#EXT-X-ENDLIST //列表结束标志，代表
这是个VoD节目

3. Live broadcast on HLS

#EXTM3U

#EXT-X-TARGETDURATION:8

#EXT-X-MEDIA-SEQUENCE:2680 // 第一个段的
sequence num

#EXTINF:8,

<https://priv.example.com/fileSequence2680.ts>

#EXTINF:8,

<https://priv.example.com/fileSequence2681.ts>

#EXTINF:8,

<https://priv.example.com/fileSequence2682.ts>

// 文件结尾没有end_list标记，代表此文件会被定期更新，是直播节目

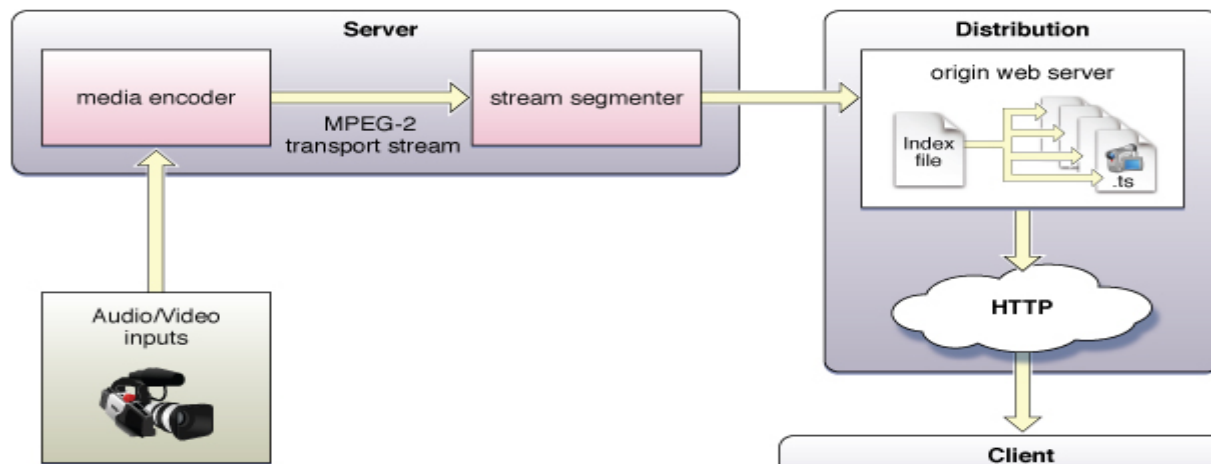
3. 客户端如何处理直播节目

- 周期性的重新获取列表文件
- 协议中对获取的频率做了如下限制
- 如果是第一次获取或者获取之后发现和上次不一样，就按列表中最后一个段的时长
- 否则，首次按0.5倍的target_dur，第二次按1.5倍的target_dur，第三次及以后按3倍的target_dur

3. Live broadcast on HLS

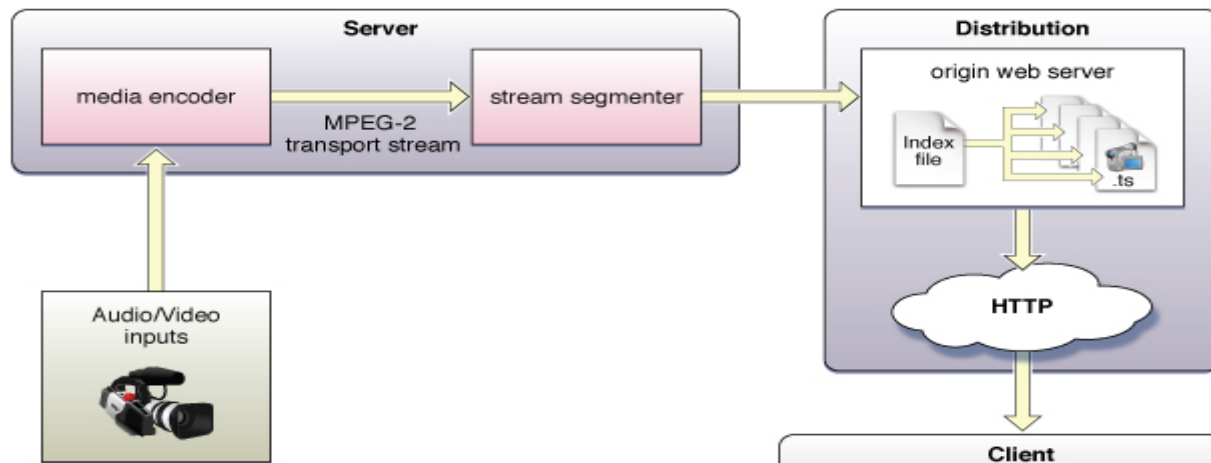
- 段的时长越短，服务器的负载越大，客户端延时越小；时长越长，负载越小，延时越大；
- 在段时长已定时，增加列表中段的个数，客户端的延时相应的会增加
- 因此，服务端应当审慎选择段的时长和列表中段的个数
- 建议10s一个段，列表中段的个数为3；可以在负载和延时之间得到一个理想的平衡，延时大约30S（根据IETF草稿和apple开发者网站）
- 我认为延时可以缩到大约20S（待探讨）

3. 观看直播时客户端的延时



- 从现场开始信号采集，到分发网络中存储有一个m3u8文件/3个ts段，至少需要30秒，故此client看到的已经是30秒前的现场
- 为什么至少需要3个ts段？IETF上的草稿和apple developer上均没有给出原因，仅仅是给出了这么一个武断的结论。

3. 观看直播时客户端的延时



- 我认为可以减少到2个ts段，延时缩短至20秒。
- 为什么不能减少到1个ts段？因为这会造成无任何缓冲数据的情况，会出现播一个段卡一下的情况；
- 为什么是2？因为还要支持client在不同网络环境中进行切换；

#EXT-X-MEDIA-SEQUENCE

- #EXT-X-MEDIA-SEQUENCE: <number>
- 个数: 0...1
- 如果有1个, 则意味着列表中的段可能会在从加载列表开始到列表时长这段时间内变得不可用
- 此标签在判断下一个待播放的段时有用, **如果无此标签**, client 须检查当前加载的段在新列表中的位置和旧列表中是否保持一致; **如果有此标签**, 下一个待播放的段是新列表中第一个比上一个被加载的段的 sequence num 大的段;

3. Encrypted Program

#EXTM3U

#EXT-X-MEDIA-SEQUENCE:7794

#EXT-X-TARGETDURATION:15

#EXT-X-KEY:METHOD=AES-

128, URI="https://priv.example.com/key.php?r=52"

#EXTINF:15, http://media.example.com/fileSequence7794.ts

#EXTINF:15, http://media.example.com/fileSequence7795.ts

#EXTINF:15, http://media.example.com/fileSequence7796.ts

#EXT-X-KEY:METHOD=AES-

128, URI="https://priv.example.com/key.php?r=53"

#EXTINF:15, http://media.example.com/fileSequence7797.ts



#EXT-X-KEY

- #EXT-X-KEY:<attribute-list>
- 个数: $0 \cdots n$, 后一个覆盖前一个
- METHOD: NONE/AES-128
- URI:key的地址
- IV:和key一起使用的Initial Vector (可选)

3. Stream alternatives

#EXTM3U

#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=1280000

<http://example.com/low.m3u8>

#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=2560000

<http://example.com/mid.m3u8>

#EXT-X-STREAM-INF:PROGRAM-ID=1, BANDWIDTH=7680000

<http://example.com/hi.m3u8>

#EXT-X-STREAM-INF:PROGRAM-

ID=1, BANDWIDTH=65000, CODECS="mp4a.40.5"

<http://example.com/audio-only.m3u8>



#EXT-X-STREAM-INF

- #EXT-X-STREAM-INF:<attribute-list>
<URI>
- 个数：0或n；注意不要和 $0\cdots n$ 弄混
- BANDWIDTH：强制，标志client播放此媒体流所需的带宽
- PROGRAM-ID：强制，具有相同ID的不同媒体流代表同一节目的不同展现
- CODECS：建议，标识了媒体数据的编码格式(RFC4281规定了相应的格式)
- RESOLUTION：可选，标识了媒体流的分辨率

3. 其它标签

- #EXT-X-PROGRAM-DATE-TIME:<YYYY-MM-DDThh:mm:ssZ>//仅用于client端的界面展示，不可用于seek/play
- #EXT-X-ALLOW-CACHE:<YES|NO>
- #EXT-X-PLAYLIST-TYPE:<EVENT|VOD>
- #EXT-X-DISCONTINUITY//编码上或者时间上
- #EXT-X-VERSION:<n>//如无即v1

3. 总结

- 常用标签有哪些？
- Q & A



4. New Features since v4

- [Variant Playlist with I-Frames.docx](#)
- [Variant Playlist with Alternative audio.docx](#)
- [Variant Playlist with Alternative video.docx](#)
- #EXT-X-BYTERANGE:<n>[@o]
 - 指定一个媒体片段属于紧接其下的media segment的一部分

4. 总结

- V5相比V3扩充了哪些标签？



Q & A



5. 评估

- 本次讲了哪几部分内容？
- 到哪里获取HLS相关资料？



5. 培训总结

- 本次讲解的内容：
 - HLS (5W & 1H)
 - 系统实现与特性
 - HLS协议解析 (<=version 3)
 - 新特性 (v4, 5, 截至2013-4)
- 进一步深入的学习：
 - iOS开发者网站
 - IETF上的草稿
 - 阅读开源软件的相关实现

谢谢！

