

基于代理技术的 HLS 客户端设计与实现

耿立宏^{1,2}, 潘 梁¹, 朱小勇¹, 郭志川¹

(1 中国科学院声学研究所 国家网络新媒体工程技术研究中心, 北京 100190;

2 中国科学院大学, 北京 100190)

摘 要: 为了解决嵌入式终端上内置播放器播放 HTTP Live Streaming (HLS) 流媒体扩展性差的问题以及第三方软件播放器资源消耗高的问题, 设计并实现了一种基于代理技术的 HLS 客户端. 详细分析了该客户端的设计结构、工作流程以及实验平台. 实验结果表明: 提出的 HLS 客户端既能充分利用原系统平台硬件解码能力, 节省系统资源, 又能通过代理动态扩充 HLS 协议, 增强系统扩展性.

关键词: HLS; 代理; 客户端; 系统资源; 扩展性

中图分类号: TP18

文献标识码: A

文章编号: 1000-7180(2016)12-0006-05

Design and Implementation of HLS Client Based on Agent Technique

GENG Li-hong^{1,2}, PAN Liang¹, ZHU Xiao-yong¹, GUO Zhi-chuan¹

(1 National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China; 2 University of Chinese Academy of Sciences, Beijing 100190, China)

Abstract: For HTTP Live Streaming (HLS) on the embedded terminal, the built-in player has a poor scalability and the third-party software player has the problem of high consumption of system resources. To tackle these issues, a HLS client is designed and implemented based on agent technique. The design architecture, working process and experimentation environment of the client are presented in detail. The experimental results show that the proposed client can save system resources by making full use of the hardware decoding capabilities of the native system and enhance system scalability by extending HLS with the agent.

Key words: HLS; agent; client; system resources; scalability

1 引言

随着视频设备的普及以及网络连接的加快, 视频传输已经占据互联网流量^[1]的主要部分.

HLS 协议是苹果公司针对苹果移动设备开发的一种基于 Hypertext Transport Protocol (HTTP) 的自适应流媒体协议. 具有高效、可扩展、兼容性强、交互方便, 低成本等特点^[2]. 目前, 各大视频网站均采用 HLS 协议作为移动流媒体客户端的主流技术^[4-6].

目前在嵌入式终端上实现 HLS 协议主要有两

种方案: 一种是使用嵌入式系统平台内置的播放器, 扩展性差. 另一种方案是使用第三方软件播放器, 这种播放器一般情况下对 HLS 协议的支持比较好, 功能更新也即时, 但由于厂商不对外开放硬件解码接口, 所以只能采用软件解码的方式, 系统资源消耗高, 播放高清视频时容易出现卡顿、马赛克等现象.

为此, 本文设计并实现了一种基于代理技术 HLS 客户端, 通过代理与 HLS 服务器交互获取流媒体数据, 并将解析好的数据交于原生播放器进行播放. 该客户端利用代理技术, 方便动态扩充 HLS 协议, 系统扩展性强; 同时, 利用嵌入式系统平台上

高效的视频硬件解码能力,系统资源占用少。

2 HTTP Live Streaming 介绍

2.1 HLS 系统结构

HLS 系统主要由三部分组成:服务器、分发器、客户端^[4-6],如图 1 所示。

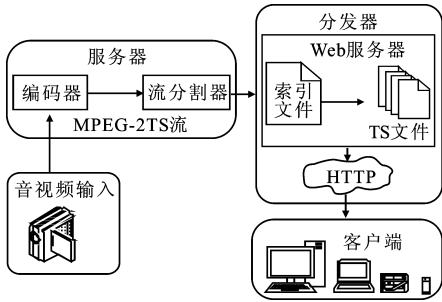


图 1 HLS 系统结构

2.1.1 服务器

服务器部分可理解为内容准备单元,为分发器准备好特定格式的视频内容,分为编码器和流分割器两部分。HLS 协议仅支持 H. 264 视频编码以及 AAC/MP3 音频编码,其视频传输格式为 MPEG-2 TS 流。编码器的作用便是将输入的原始音视频数据按照协议要求格式进行编码,然后将编码后的音视频复用、封装为 MPEG-2 传输流进行输出。采用 MPEG-2 TS 封装音视频媒体流是因为 MPEG-2 TS 格式能够将音视频媒体流严格按照时序进行交织复用,分段后,每一个分片都不依赖与之前的分片,可以独立进行解码和播放^[3]。如果需要实现多码率自适应功能,则需要编码出不同码率的 TS 流。

HLS 流媒体最具特色的地方就是视频分段,然后按时间顺序连续地请求视频段进行播放。流分割器,将编码器输出的 TS 流分割成一系列等间隔(推荐为 10 s)的、连续的媒体片段(后缀为 .ts),并为每个媒体片段生成在分发器(web 服务器)上的存储链接,顺序记录在 M3U8 索引文件(.m3u8)中,该索引文件叫做次级索引文件,其中包含媒体片段序列的链接以及段长、播放方式、目标段长、初始序列号等信息。如果是点播方式,该索引文件是静态的,包含所有媒体片段的链接,且包含结束标志“#EXT-X-ENDLIST”。如果是直播方式,索引文件里不包含结束标志,默认含有 3 个媒体片段链接,每当生成一个新的片段,新的链接将被添加到索引文件末尾,与此同时,索引文件前面信息最旧的链接将被移除,这种类似滑动窗口的更新方式保证索引文件里保持最新

的三个媒体片段的链接。如果含多个码率的 TS 流,流分割器将为每个码率的 TS 流生成次级索引文件,与此同时,将生成一个主级索引文件,其中包含不同码率次级索引文件的链接以及码率值、平均码率值、视频分辨率、音视频编码方式、帧率等信息,结构如图 2 所示。

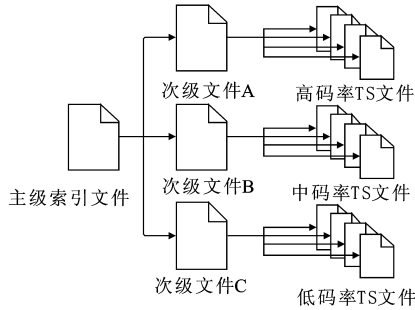


图 2 索引文件结构

此外,流分割器还能实现对每个 TS 片段进行加密,并生成相应的密钥文件,如同媒体片段一样,该密钥文件的链接也会顺序的记录在 M3U8 索引文件中。最后,将所有的媒体片段和 M3U8 索引文件发送到分发器上进行存储。

2.1.2 分发器

分发器存储服务器发送过来的媒体片段序列和 M3U8 索引文件,并响应客户端的请求,与客户端进行交互。它可以是一个简单的 web 服务器或者缓存服务器,几乎不需要做任何特别的配置,就可以直接运用到 HLS 系统中。苹果官方文档推荐配置仅限于对索引文件(.m3u8)和媒体文件(.ts)的 MIME 类型关联,如表 1 所示。

表 1 文件格式和 MIME 类型

文件扩展名	MIME 类型
.m3u8	application/x-mpegURL or vnd.apple.mpegURL
.ts	video/MP2T

2.1.3 客户端

客户端是指支持 HLS 协议解析的终端设备,IOS3.0、Safari4.0 及其以后版本都支持 HLS 协议播放。通常情况下,客户端通过 Uniform Resource Locator (URL)链接向分发器请求主级索引文件,该索引文件进一步指明了不同码率的次级索引文件的链接以及每个码率流信息。客户端可根据当前终端资源情况向分发器请求最合适的码率。对于特定的码率流,客户端按时间顺序依次请求视频段,经解码后完成播放。对于加密的媒体内容,客户端根据索

引文件指引的密钥链接获取解密密钥,对加密媒体流进行解密处理。

对于点播,客户端将持续下载视频段进行播放,直至遇到“#EXT-X-STREAM-INF”结束标志。如果是直播,客户端将周期性的更新索引文件,获取最新的视频段链接。

3 基于代理技术的 HLS 客户端设计

本文提出的基于代理技术的 HLS 客户端主要原理是通过代理与 HLS 服务器交互获取流媒体数据,并将解析好的数据交于原生播放器进行播放。其结构如图 3 所示,包括管理模块和代理模块。

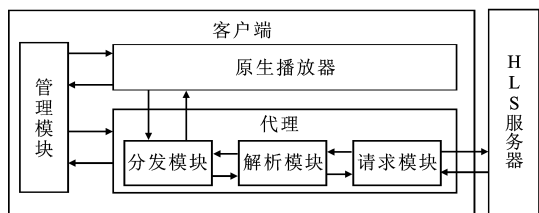


图 3 客户端结构

3.1 管理模块

管理模块负责申请客户端所需资源,对各个模块进行初始化;同时负责控制各模块间启动、同步以及终止;管理模块监听用户发起的视频控制命令,包括视频请求、播放、暂停、拖动、选时、退出等,并将这些命令发送给代理模块。

3.2 代理模块

代理模块是整个客户端的核心所在,与 HLS 服务器进行交互获取流媒体数据,并将解析好的数据发送给原生播放器进行播放。这样,原系统平台不需要支持 HLS 流媒体协议,就可实现 HLS 流媒体的播放。代理模块包括以下子模块:请求模块、解析模块和分发模块。

请求模块接收管理模块发送过来的媒体请求(URL),向 HLS 服务器发出 HTTP 请求,并接收 HLS 服务器响应的媒体数据。本文采用 libcurl 库实现请求模块功能,libcurl 是一个简单易用的基于 URL 传输的客户端开发库,支持多种平台以及三十多种开发语言绑定。

解析模块从请求模块中读取下载的流媒体数据,并负责解析,得到适合原生播放器播放的数据。具体包括 HLS 流媒体协议解析和算法扩展。

流媒体协议解析是指 M3U8 索引文件的解析,包括主级索引文件解析和次级索引文件的解析。通

过“#EXT-X-STREAM-INF”标签加以区别,若含有该标签则是主级索引文件,反之,则为次级索引文件。如上所述,主级索引文件里含有不同码率次级索引文件的链接以及每个码率流信息。主级索引文件解析包括:存储每个码率版本的码率值、平均码率值、分辨率信息以作为码率自适应选取标准;此外,主级索引文件里存储的次级索引文件链接为相对地址,需根据请求模块的请求链接(URL)将此相对地址转换为服务器上的绝对地址。次级索引文件解析包括:(1)根据“#EXT-X-VERSION”标签解析 HLS 协议版本号,以确定可以使用的协议功能;(2)“#EXT-X-ALLOW-CACHE”标签确定是否可以缓存下载的视频段;(3)“#EXT-X-MEDIA-SEQUENCE”标签获取起始视频段序列号,后续视频段的序列号在此序列号上依次加 1;(4)“#EXT-X-TARGETDURATION”标签获取视频段最长段长;(5)“#EXT-X-PLAYLIST-TYPE”标签判断视频播放类型,包括直播和点播;(6)“#EXTINF”标签标示该视频段的段长;(7)其他非标签(不以“#”开始)信息为视频段链接。同样,该链接为相对地址,解析程序需根据请求模块的请求链接将该相对地址转为绝对地址。

算法扩展部分包含终端资源监测算法,如 CPU 占用率、播放器 buffer 大小、网络带宽等,此外还包含码率自适应算法。可很方便地在本文提出的客户端框架上进行算法的改进与扩展。

多数系统平台原生播放器(如 Android)只支持本地文件播放以及网络视频流播放。本地文件播放不同的视频段,将重启播放器,导致视频播放的不连续。为了连续的播放下载的视频段,本文在代理模块集成了一个轻量级服务器(lighthttpd),即分发模块,将下载的视频段连续地发送到原生播放器网络视频流接口进行播放。

4 播放流程设计

时序图(Sequence Diagram)是一种 Unified Modeling Language (UML)交互图。它通过描述对象之间发送消息的时间顺序显示多个对象之间的动态协作^[7]。本文通过时序图来表示客户端播放流程,如图 4 所示。

由图中所示,客户端工作流程大概分为三部分,即启动、正常播放和命令控制。

启动阶段,当用户启动播放器时,管理模块将启动代理模块以及原生播放器,并对客户端所需资源

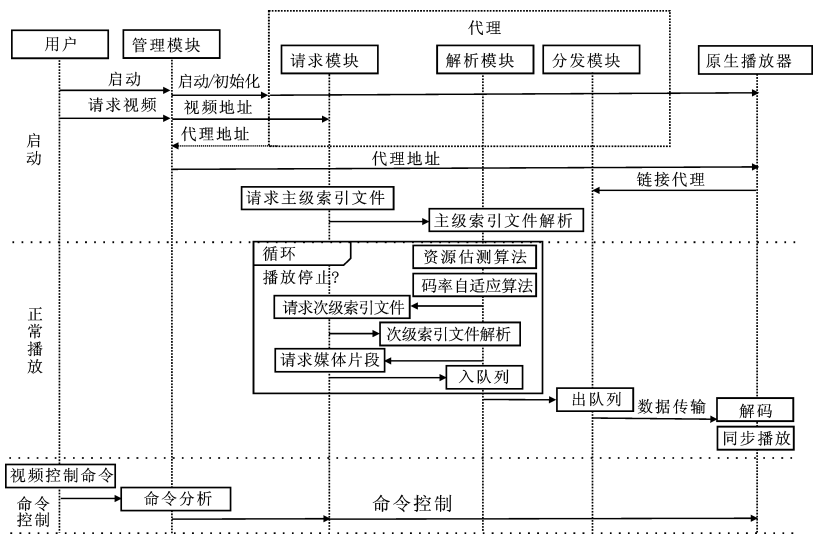


图 4 客户端流程

进行初始化,然后用户输入 HLS 流媒体视频链接,请求视频.管理模块将该请求链接发送给代理模块,由代理模块中请求模块处理该请求.与此同时,代理模块将返回代理的地址链接给管理模块.管理模块将该地址给原生播放器网络视频播放接口,原生播放器通过该地址和代理建立链接.另一方面,请求模块接收到请求链接后向 HLS 服务器请求主级索引文件,并由解析模块对该索引文件进行解析.

正常播放阶段,首先调用客户端的资源估测算法,资源包括 CPU 占用率、内存大小、网络带宽、缓存大小等.然后根据当前资源情况调用码率自适应算法,选取最适合当前客户端的码率.根据选取的码率链接,请求模块向 HLS 服务器请求对应的次级索引文件;解析模块解析次级索引文件,并根据当前记录的序列号决定下一个请求的视频段链接,并将此链接传给请求模块;请求模块以此链接向服务器请求该 TS 视频段;解析模块将下载的 TS 流数据入队列,等待分发模块分发;分发模块不断从队列里取出流媒体数据发送给原生播放器.原生播放器接收到流媒体数据后解码播放.由图 4,可以看到解析模块和分发模块是异步工作,解析模块在完成数据入队列后又回到循环初始,调用资源估测算法.正常播放阶段一直如此循环,直至视频播放结束或者遇到播放停止命令.

命令控制阶段,由客户发起的视频控制命令,包括播放、暂停、拖动、选时、退出等.管理模块接受到命令后,将同时对原生播放器以及代理模块做相应命令的控制.其中播放、暂停命令仅需请求模块做相应处理;拖动、选时、退出命令还需释放分发模块的

缓存队列.

5 实验设计

5.1 实验系统

服务端,通过 CBR 编码模式将原始视频编码为 100、500、1 000、1 500 和 2 000 kb/s 5 个码率版本.此外,采用苹果系统文件分段工具 mediafilesegmenter 将每个码率版本切分为一系列长度为 10 秒的视频段.所有的视频段以及索引文件存在一个 Apache 服务器上,该服务器版本为 2.4.9,为 Mac Mini 10.10.2 系统自身集成,采用“sudo apachectl start”命令启动^[8].此外,为了避免客户端与服务器频繁建立与断开 TCP 连接^[9],本文将 Apache 服务器连接超时参数设置为 60 秒.

Android 系统是智能终端占有量最大的系统,提供了大量的 API 接口,开发简单.本文在小米手机 M2 上实现了 HLS 客户端,该手机配备 1.5 G APQ8064 四核处理器,2 GB RAM.

5.2 实验测试

为验证本文 HLS 客户端性能,本文选取 Android 原生播放器和 VLC 播放器作为对比.其中 Android 原生播放器采用系统平台硬件解码,VLC 播放器采用纯软件解码方式.在相同的条件下播放不同码率的视频,CPU 占用率以及内存消耗统计结果如图 5 和图 6 所示.

如上图所示,本文提出的客户端在播放不同码率视频时,CPU 占用率和内存消耗只略高于 Android 原生播放器.这是因为代理模块的加入增加了一定的 CPU 和内存资源消耗.但庆幸的是,在牺牲这些资源

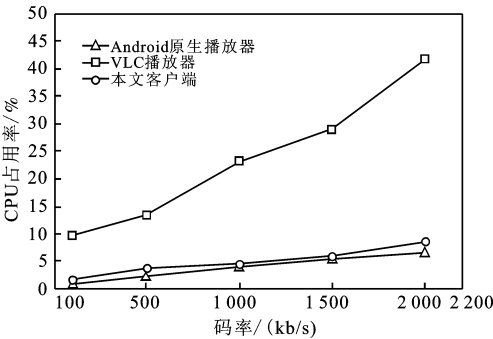


图 5 不同播放器 CPU 占用率对比结果

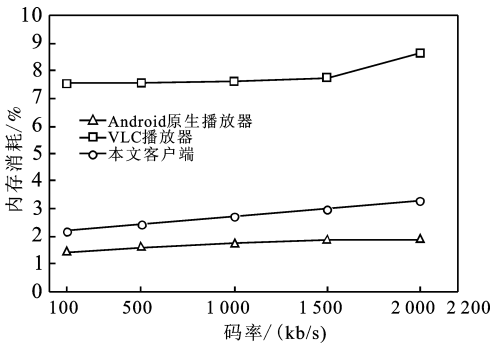


图 6 不同播放器内存消耗对比结果

的同时,本文客户端拥有了 VLC 等第三方软件播放器的良好扩展性.此外,本文提出的客户端的内存消耗和 CPU 占用率都远低于 VLC 播放器,并且随着视频码率的增大,本文客户端优势越明显.这是因为 VLC 播放器采用了软件解码的方式,视频码率越大,消耗系统资源就越多.而本文客户端充分利用了原系统平台硬件解码功能,因此节省了资源.

此外,本文选取同一段视频测试了视频启动延时,平均统计结果表 2.

表 2 不同播放器启动延时对比结果

数据	VLC 播放器	Android 原生播放器	本文客户端
启动延迟/秒	3.217 5	1.67	2.56

由表 2 所示,本文客户端启动延时少于 VLC 播放器,这是因为 VLC 播放器采用了软件解码方式,对视频的软解码带来了启动的延迟.此外,由于代理的加入,使得本文客户端播放视频相对于正常流程多了一次数据分发过程,因此与 Android 原生播放器相比,本文客户端启动延迟稍大.

综上,本文提出的基于代理的 HLS 客户端在牺牲微小的启动延迟下,不仅具备 VLC 播放器良好的扩展性,而且具备 Android 原生播放器资源消耗低的优点.

6 结束语

本文设计并实现了一种基于代理技术的 HLS 客户端,通过代理与 HLS 服务器交互获取流媒体数据,并将解析好的数据交于原生播放器进行播放.本文提出的 HLS 客户端具有以下优点:(1)充分利用嵌入式系统平台上高效的视频硬件解码能力,系统资源(CPU、内存)占用少;(2)利用代理模块,动态扩充 HLS 协议,系统扩展性强.

参考文献:

[1] Cisco System Inc. Cisco visual networking index: forecast and methodology, 2014-2019 white paper [EB/OL]. (2015-05-27). http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html.

[2] 许联合国, 顾民. 基于 HLS 协议的视频点播系统客户端设计[J]. 电子制作, 2015, 22(2): 65-67.

[3] 王文轩, 胡晓晨, 田响, 等. 一种基于 HLS 的移动流封装与切片部署架构[J]. 软件, 2015, 36(2): 31-35.

[4] 金达, 叶庆伟, 狄红卫. 基于 HLS 的流媒体播放系统的设计与实现[J]. 信息技术, 2013, 37(10): 49-52.

[5] Kupka T, Halvorsen P, Griwodz C. An evaluation of live adaptive HTTP segment streaming request strategies[C]//Local Computer Networks (LCN), 2011 IEEE 36th Conference on. Germany, Bonn, IEEE, 2011: 604-612.

[6] Apple Inc. HTTP Live Streaming draft-pantos-http-live-streaming-18 [EB/OL]. (2015-11-09). <https://tools.ietf.org/html/draft-pantos-http-live-streaming-18#section-1>.

[7] 侯丽娟. 移动流媒体舞蹈欣赏视频直播系统设计[J]. 自动化技术与应用, 2015, 33(6): 43-47.

[8] 丁耀. 基于 Http Live Streaming 的直播技术研究[J]. 软件, 2014, 35(3): 129-131.

[9] 林锐, 刘峰. 一种基于 HTTP 长连接的自适应流媒体传输系统[J]. 电视技术, 2012, 36(7): 85-88.

作者简介:

耿立宏 男, (1989-), 硕士研究生. 研究方向为多媒体通信. E-mail: genglh@dsp.ac.cn.

潘 梁 男, (1973-), 硕士, 副研究员. 研究方向为智能终端、数字家庭.

朱小勇 男, (1982-), 博士, 副研究员. 研究方向为智能终端、数字家庭.

郭志川 男, (1975-), 博士, 副研究员. 研究方向为智能终端、数字家庭.