

文章编号:1001-9081(2009)04-1076-03

基于 RTSP 控制的流媒体代理服务器集成算法研究

朱世交, 杨 珺, 唐 忠

(上海电力学院 电子与信息工程系, 上海 200090)

(mediate@163.com)

摘 要:随着网络流媒体技术得到广泛应用,为了更好地利用网络资源并对客户端数据链接进行控制,提出一种建立流媒体代理访问控制的集成方法,通过调整网络数据流,控制实时流协议(RTSP)、实时传输控制协议(RTCP)、实时传输协议(RTP)在服务器端与客户端数据的传输,实现流媒体数据服务器的代理功能,满足不同网络情况下客户端链接的需求,并通过集成实验验证了方法的有效性。

关键词:流媒体;代理;实时流协议;实时传输控制协议;实时传输协议

中图分类号: TP37 **文献标志码:** A

Algorithm study of proxy integration method based on RTSP controlling of streaming media server

ZHU Shi-jiao, YANG Jun, TANG Zhong

(Department of Electronic and Information Engineering, Shanghai University of Electric Power, Shanghai 200090, China)

Abstract: Streaming media technology has been widely used. In order to better use network resource and control the user's connections, a proxy method was proposed. It adjusted the data streaming in network using Real Time Streaming Protocol (RTSP), Real-time Control Protocol (RTCP) and Real-time Transport Protocol (RTP) between server and clients to implement a streaming media proxy to meet the requirement of different client connections. From the results of integration experiments, it shows that the proposed method is valid.

Key words: streaming media; proxy; Real Time Streaming Protocol (RTSP); Real-time Control Protocol (RTCP); Real-time Transport Protocol (RTP)

0 引言

当前,随着网络及 3G 移动通信技术的发展,通过移动终端观看音视频成为应用的热点,与此相关的流媒体技术也有了长足的发展^[1-2]。

流媒体技术是将音视频文件和音视频流通过压缩处理后,通过网络服务器进行分段传输,客户端计算机不用将整个音视频文件下载到本地,便可以即时收听和收看。对于典型的流媒体服务应用,一般要实现用户认证、重定向、控制器、会议发布、计费等多个模块^[3],但由于各个流媒体服务器实现技术并不相同,因而需要研究如何在这些服务器之间建立相应的集群结构,及其扩展功能,优化服务资源,例如:健全控制用户资源访问、调整流媒体服务器的均衡负载等。

为了分析流媒体服务对应的客户端和服务端之间的数据交互,研究中使用的流媒体服务器是 Darwin Streaming Server^[4],通过控制实时流协议(Real Time Streaming Protocol, RTSP)^[5],实时传输控制协议(Real-time Control Protocol, RTCP)及实时传输协议(Real-time Transport Protocol, RTP)^[6]的数据流向,实现代理服务分组交换功能^[7-8]。

1 流媒体代理交互的可行性分析

图 1 给出了通信协议之间的相互结构关系,可以看出 RTSP 通过 TCP 控制数据的双向传递,RTP,RTCP 报文一般通

过用户数据报协议(User Datagram Protocol, UDP)数据包进行通信。

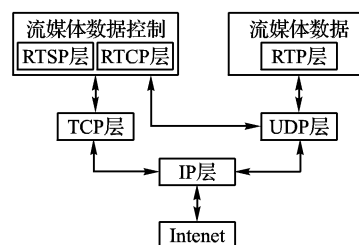


图 1 流媒体通信结构相互关系组成

因此,RTSP 通过 TCP 层控制 RTP 和 RTCP 的服务器和客户端的连接,其中包括一些交互方法,这些方法类似于 HTTP 的传输模式方式,通过 UTF8 编码的文本方式编码。表 1 给出了其中的主要方法,特定消息请求(request)类型都有相应的反馈(response)数据包,具体参见文献[5]。

表 1 RTSP 连接时的主要方法

| 方法 | 传送方向 | 附加消息对象类型 |
|----------|----------|----------|
| DESCRIBE | C→S | P, S |
| OPTIONS | C→S, S→C | P, S |
| SETUP | C→S, S→C | S |
| PLAY | C→S | P, S |
| PAUSE | C→S | P, S |
| TEARDOWN | C→S | P, S |

收稿日期:2008-10-10;修回日期:2008-12-17。 基金项目:上海市重点科技攻关计划项目(071605125)。

作者简介:朱世交(1980-)男,安徽五河人,讲师,博士,主要研究方向:通信网络、多媒体技术、神经网络; 杨珺(1978-)女,江西上饶人,讲师,博士,主要研究方向:信息检索、知识表示; 唐忠(1964-)男,湖南武冈人,副教授,博士,主要研究方向:电力、通信系统。

表1中,C表示Client流媒体客户端;S表示Server流媒体服务器端;P表示方法可用于表示模式;S表示方法可用于流模式。

客户端在连接流媒体服务器播放视频时,其建立连接过程的时序描述如图2所示。

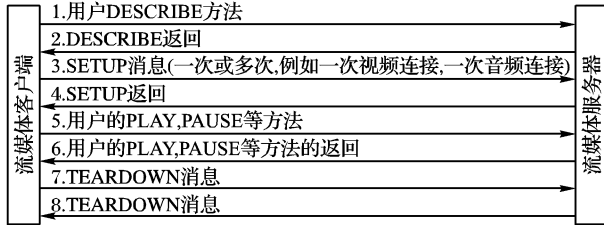


图2 流媒体播放时主要消息时序

流媒体客户端首先通过 DESCRIBE 方法与流媒体服务器进行信息交互,主要起协商双方播放参数的作用;其后通过 SETUP 方法进行双方 RTP/RTCP 端口确定,为后续数据的传输提供方便;双方端口映射后客户端可通过 PLAY 或 PAUSE 方法来操纵流媒体的播放和暂停等服务;要终止此次服务,正常情况下客户端通过 TEARDOWN 方法通知服务器断开连接。上述 RTSP 协议方法通过 TCP 连接进行数据发送,播放中的音频和视频信息通过 RTP、RTCP(UDP 方式)端口进行包传递。

2 流媒体代理服务器的实现

通过服务器和客户端之间交互数据的分析,可以看出,利用服务器和客户端双方对 RTSP(即 TCP 长连接)的连接内容,可实现控制客户端和服务器的访问,这是一种简单而有效的方式,无须对 RTSP、RTP、RTCP 内容进行全部信息解码,可以减少代理系统在操作时的额外时间开销,为了达到上述描述的目的,此时要考虑的内容包括:

1) 访问控制对客户端和服务端 RTSP(TCP 连接)数据流的支持。

2) 访问控制对客户端和服务端 RTP 和 RTCP 的支持。

2.1 客户端与服务端的 RTSP 连接控制

因为 RTSP 为 TCP 长连接,因此在客户端进行服务器连接时,访问代理控制模块就可分配两个 TCP 链接分别为客户端和服务端提供链接。

对于 RTP 和 RTCP 的端口确立,在 SETUP 方法之后方能确定双方端口的建立,而 SETUP 方法通过 RTSP 的 UTF8 明文内容传递。

例如在客户端发起 SETUP 消息:

SETUP RTSP://211. 136. 109. 148; 554/sample_50kbit. 3gp/trackID=3 RTSP/1.0;

Transport: RTP/AVP;unicast;client_port=9700-9701。

其中 client_port=9700-9701 为用户端告诉服务端自身对应的 RTP 和 RTCP 端口号。

通过访问代理控制模块之后发向服务器端的对应的客户端数据:

SETUP RTSP://XXX. XXX. XXX. XXX; 554/sample_50kbit. 3gp/trackID=3 RTSP/1.0;

Transport: RTP/AVP;unicast;client_port=7000-7001。

其中的 XXX. XXX. XXX. XXX 为流媒体服务器真正的地址,

client_port 被修改为访问代理控制使用的连接端口号,用于充当虚拟的客户端。

对上述客户端发起的 SETUP 命令,此时服务器返回的数据包含以下主要数据片段:

Server: DSS/5.5 (Build/489.7; Platform/Win32; Release/Darwin);

Transport: RTP/AVP; unicast; source = XXX. XXX. XXX. XXX; client_port = 7000 - 7001; server_port = 6970 - 6971; ssrc = 0000018。

通过代理控制模块之后的服务器端数据:

Server: DSS/5.5 (Build/489.7; Platform/Win32; Release/Darwin);

Transport: RTP/AVP; unicast; source = 211. 136. 109. 148; client_port = 9700 - 9701; server_port = 6000 - 6001; ssrc = 0000018。

利用修改 RTSP 明文的地址端口,来改变数据发送的流向。图3描述了原有无代理控制时的端口连接。图4是使用代理控制时端口连接图例。

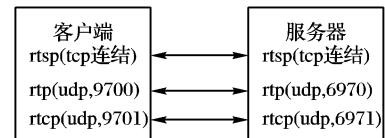


图3 无代理情况下的服务器与客户端端口

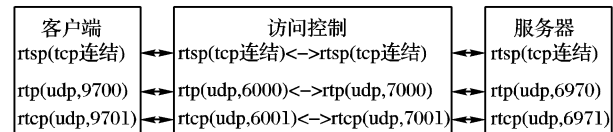


图4 代理控制条件的端口新映射状况

可以看出访问控制在真正发起的客户端和服务端之间起到了桥梁嫁接的作用,通过“欺骗”客户端和服务端,一方面模拟为虚拟客户端,另一方面模拟为虚拟服务器端。

2.2 针对数据流 RTP、RTCP 的代理算法

使用代理服务器进行实体端口的建立和数据端口 UDP(RTP,RTCP)的映射,需要建立地址和端口映射来路由 UDP 报文,五元组描述如下:

$G = \{P, S_A, S_P, D_A, D_P\}$;

P:控制代理的侦听端口;

S_A:表示接收到的 UDP 包的源地址;

S_P:表示接收到的 UDP 包的源端口;

D_A:表示 UDP 数据要发送的目的地址;

D_P:表示 UDP 数据要发送的目的端口;

访问代理控制通过检测端口(在 SETUP 方法时建立)每接收到一个 UDP(RTP、RTCP)数据报文,就用哈希搜索 UDP 报文源地址和源端口;而后通过五元组信息中的 P、S_A、S_P 找到 D_A、D_P,决定数据发送的目的地址和端口,进而把数据投递出去,代理服务器框架伪代码算法如下:

Module of proxy

Parse method of RTSP describe utf-8 text

case (method in [RTSP method])

{

setup:

to find setup ports of server and

client and register them.

```

to setup TCP and UDP mapping router.
play[ pause, stop]:
to play[ pause, stop] proxy UDP streaming
to set stat to playing[ pause, stop]
}
if (stat is playing) {
to find the target server and client ports inspectively.
to send the UDP package to target port
}

```

End of module

2.3 系统整体功能模块实现策略

通过上述数据分析,设计模块如图 5 所示。

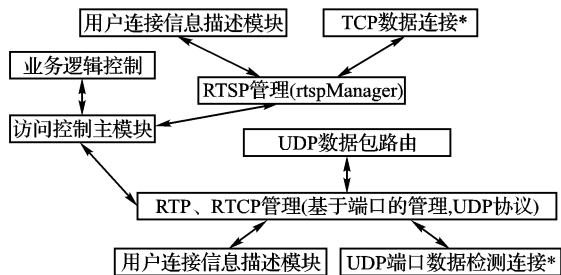


图 5 代理服务器内部连接模块结构

模块主要包括访问控制主模块、业务逻辑控制、RTP、RTCP 控制、RTSP 管理。

访问控制主模块 主要起协调主系统的运行,包括对 RTSP、RTCP 及 RTP 状态控制和协调。

业务逻辑控制 对用户访问,系统控制具体业务逻辑,进行信息的变更与记录等操作。

RTP、RTCP 控制 控制 UDP 数据传送和路由表。

RTSP 管理 控制客户端和服务端数据传递。

3 实验及数据分析

在系统设计过程中,采用了三层结构模型,代理服务对客户端和服务器的连接起到中间“透明”衔接的作用,其可操作性在 JDK5.0 平台,流媒体服务器 Darwin 5.5 及客户端 pvplayer 3.0.1 上验证通过,以下分析代理中连接数及效率。

服务器的连接数 使用 UDP 传输的 RTP、RTCP 端口,通过两对端口进行传送控制视频和音频,TCP 连接的 RTSP 需要一个,因此当 N 个客户端发起请求时,最坏情况下需要 N 个 TCP 和 $(N+2)$ 个 UDP 连接才能提供服务。

客户端 它包括一个 RTSP 方式的 TCP 连接,两对 UDP 连接,当 N 个客户端发起请求时,最坏情况下其连接数为 N 个 TCP 及 $4 \times N$ 个 UDP 连接。

监控代理的服务器 对单个客户端服务而言,它包括 2 个分别服务于客户端和服务端的 TCP 连接,4 个用以虚拟客户端的 UDP 连接,2 个用以提供虚拟服务器的 UDP 连接。当客户端个数为 N 时,最坏情况下需要 $2 \times N$ 个 TCP 连接和 $4 \times N + 2$ 个 UDP 代理连接才能完成代理服务。

同时,通过网络实验验证,正常情况下具体流媒体控制数据包 RTSP 解析内容大约有 10%,RTCP 带宽消耗大约为 5%^[5],其他数据以透明传送方式传送到目的地,由此可看出控制系统对网络带宽及时间消耗占用很小,系统连接资源服务利用率平均测试比较如图 6 所示。

通过图 6 可以看出访问控制效率受客户端发起的并发连

接数限制,由于充当虚拟客户端和服务端,因此其效率随客户端连接数的增加而增加,同时兼顾不同的单独流媒体服务,做到整体服务资源分配,均衡负载的作用,因此整体系统效率要略高于单独服务器;另外,采用缓冲机制进行连接 RTSP 控制,降低连接数也值得研究。

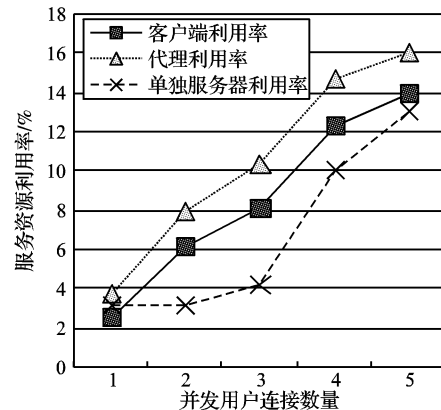


图 6 网络流媒体服务资源利用率

4 结语

通过对流媒体数据传送协议的分析,利用代理方式在流媒体客户端和服务端建立了一个具体的访问控制层,此控制层独立于具体流媒体客户端和服务端的实现,通过异构数据控制的方法实现对用户流媒体服务请求的控制,从而起到控制用户访问和流媒体服务负载均衡等作用;同时,实现了逻辑控制的基本功能,并通过 Java 代码进行了构建和验证;当然,对用户访问控制模型及效率的研究有待于进一步深入,提高其泛化能力。

参考文献:

- [1] 尹浩,林闯,文浩,等.大规模流媒体应用中关键技术的研究[J].计算机学报,2008,31(5): 755-774.
- [2] AHUJA S, KRUNZ M. Algorithms for server placement in multiple-description-based media streaming[J]. IEEE Transactions on Multimedia, 2008, 10(7): 1382-1392.
- [3] PAPADIMITRIOU P, TSAOUSSIDIS V, MAMATAS L. A receiver-centric rate control scheme for layered video streams in the Internet[J]. Journal of Systems and Software, 2008, 81(12): 2396-2412.
- [4] Apple Inc. Darwin Server[EB/OL]. [2008-09-02]. <http://developer.apple.com/darwin/projects/streaming/>.
- [5] SCHULZRINNE H, RAO A, LANPHIER R. Real Time Streaming Protocol (RTSP)[EB/OL]. [2008-08-10]. <http://rfc.sunsite.dk/rfc/rfc2326.html>.
- [6] SCHULZRINNE H, CASNER S, FREDERICK R, et al. RTP: A transport protocol for real-time applications[EB/OL]. [2008-08-10]. <http://www.ietf.org/rfc/rfc3550.txt>.
- [7] SHEN LI-QUAN, LIU ZHI, ZHANG ZHAO-YANG, et al. An adaptive and fast multiframe selection algorithm for H.264 video coding[J]. IEEE Signal Processing Letters, 2007, 14(11): 836-839.
- [8] KIM B G. Novel inter-mode decision algorithm based on macroblock (MB) tracking for the P-slice in H.264/AVC video coding[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2008, 18(2): 273-279.