

文章编号: 1007- 2985(2008)01- 0039- 04

基于 SIP 协议视频广播的流媒体服务器的设计^{*}

李新国

(湖南省第一师范学校信息技术系, 湖南 长沙 410002)

摘要: 探讨了实现基于 SIP 协议视频广播服务所需的关键技术, 设计了流媒体服务器的协议分层模型和流媒体服务器的体系结构, 根据所设计的体系结构实现了 1 支持视频广播的原型系统。

关键词: SIP 协议; 视频广播; 流媒体服务器

中图分类号: TP393

文献标识码: A

会话初始化协议(Session Initiation Protocol, 简称 SIP) 是由 Internet 工程任务组定义的一种多媒体通信控制协议, 这种基于文本的应用层协议, 由于其简洁、开放等特性, 正逐渐成为 Internet 多媒体体系结构的核心控制协议。流媒体将音频、视频等多媒体信息转化成“流”的形式, 信息在传输的同时即可进行播放。流媒体技术的应用越来越广泛, 从视频点播、网络电台到远程教育以及在线的新闻网站等都用到流媒体技术。随着网络与通信技术的发展, 人们对通过计算机网络提供流媒体服务的需求也越来越强烈, 由此也为人们在更高层次上利用网络打开了一扇方便之门, 笔者重点阐述了基于 SIP 视频广播服务的流媒体服务器的体系结构及其实现的关键技术。

1 SIP 协议及广播传送方式

一个具有支持视频广播功能的流媒体服务器包括 SIP 协议、SDP 协议及广播传送方式。

1.1 SIP 协议的功能及层次结构

1.1.1 SIP 协议的功能 (1) 服务定位: 确定通信中流媒体服务器的位置; (2) 服务可用性: 确定流媒体服务器是否愿意参与通信, 是否提供媒体服务; (3) 参数协商: 确定通信中所使用的流媒体参数; (4) 会话建立: 呼叫双方会话参数的建立; (5) 会话管理: 对会话转移、终止及会话参数变更等内容进行管理。

1.1.2 SIP 协议的层次结构 SIP 协议是一种应用层控制协议, 可以运行在几个不同的传输层协议上, 如 UDP、TCP。同时, SIP 协议本身是一个分层协议, 它分成 4 层, SIP 协议的分层结构如图 1 所示。

(1) 语法编码层。语法编码层位于协议栈的最低层, 定义了 SIP 协议所使用的各种参数、消息的结构、语法和格式, 采用递增 BNF(Backus Normal Form) 语法编码方式。

(2) 传输层。传输层定义了客户端和流媒体服务器如何发送请求和接收应答, 所有 SIP 实体必须包含传输层, 并支持 TCP 和 UDP 传输协议。

(3) 事务层。事务层负责保存事务的状态, 处理应用层的重传和超时, 以实现消息的可靠传输、匹配应答与请求。用户代理及保留状态代理服务器均包含事务层, 无状态代理服务器不包含事务层, 而直接使用传输层提供的服务进行消息转发。

(4) 事务用户层。事务用户层位于最顶层, 除无状态代理服务器外, 每个 SIP 实体都是事务用户, 当一个事务用户要发送一个请求消息时, 先要创建一个客户事务实例, 然后把请求消息、目的 IP 地址、端口号和传输方法传给该客户事务, 然后该客户事务负责指导消息可靠的发送到目的地。

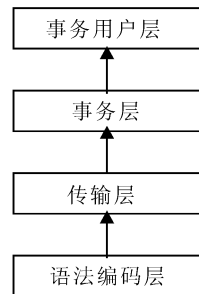


图 1 SIP 协议的层次结构

* 收稿日期: 2007- 11- 13

基金项目: 湖南省教育厅科学研究项目(06C053)

作者简介: 李新国(1965-), 男, 湖南绥宁人, 湖南省第一师范学校信息技术系教授, 主要从事 IP 通信及远程教育技术教学与研究。

1.2 SDP 协议

会话描述协议SDP(Session Description Protocol)用于描述多媒体会话,实现会话的发布、会话邀请和其他形式多媒体会话的启动,使参加会话的各方能够相互了解对方的媒体描述信息,从而能够通过协商达成一致。SDP是一种会话描述的格式,它可以由不同的传送协议使用,包括会话宣布协议SAP、会话初始协议SIP、实时流协议RTSP、使用MIME扩展电子邮件SMTP和超文本传输协议HTTP。

1.3 广播传送方式

在计算机网络中,有很多时候要用到广播技术,对于视频广播服务自然要使用到网络的广播传送方式。广播在数据链路层和网络层都会被用到,前者将广播信息发送到多个物理设备-后者则将广播信息发送到多个逻辑设备。在TCP/IP网络中,有2种广播:全面广播,即将广播数据发送到每一台主机;部分广播,即将信息广播到一个网络或子网。

2 流媒体服务器的设计

2.1 流媒体服务器的工作原理

流媒体服务器提供媒体服务的典型呼叫流程:流媒体服务器解析 Invite 消息中所包含的客户端所支持媒体的信息后,根据其提供媒体服务所需媒体是否与客户端的媒体相匹配,流媒体服务器沿原路回复 Invite 消息,标志建立或结束此次 SIP 会话,当回复标志已建立会话并收到客户端确认后,流媒体服务器便进入了流媒体会话期。在流媒体传送期间必须与客户的播放器保持双向通信——服务器到客户端的信息是通过广播传送的,而客户端到服务器的信息则是点对点传送的(这种通信是必须的,因为服务器需要客户端的反馈信息,同时又利用广播减轻了网络流量)。媒体服务呼叫流程如图2所示,具体工作过程:支持SIP协议的客户端向媒体服务器发出 Invite 消息,当媒体服务器接收到该消息后便建立一次 SIP 会话,同时将此次 SIP 会话的媒体请求与当前的媒体会话内容进行比较,映射到已存在的媒体会话或者建立新的媒体会话,在映射或建立的过程中需进行媒体协商,当新的媒体会话建立后,则调用媒体处理、传送媒体流。

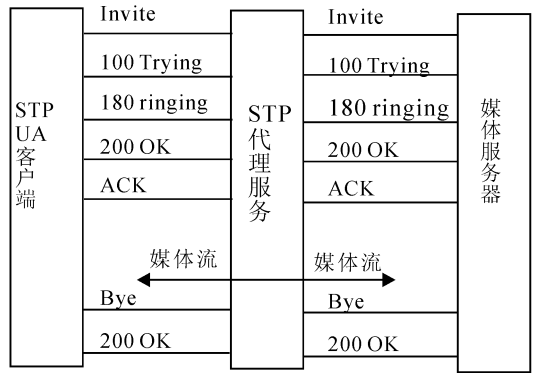


图2 典型的媒体服务呼叫流程

2.2 流媒体服务器的体系结构

流媒体服务器的基本功能:服务器接收客户端的请求,产生新的信令会话或者修改已存在的信令会话的参数,并实现由信令会话向媒体会话的映射,映射方式为媒体会话根据请求播放的媒体文件来产生。如果播放的媒体文件已形成媒体会话,则将该会话的必要参数传送给客户端,并将此次信令会话一一对应到该媒体会话;相反,如果播放的媒体文件未形成媒体会话,则新建媒体会话,并将此次信令会话一一对应到新建的媒体会话,当新的媒体会话形成后,调用媒体处理模块处理相应的媒体文件并向本地子网播放。

综上所述,笔者设计的流媒体服务器系统由各种不同的软件构成,这些软件在各个不同层面上互相通信。基本的流媒体服务器系统包含以下几个组件:信令服务器、媒体解析器、媒体编码器、媒体播放器、媒体服务控制器、会话管理器和播放管理器。整个流媒体服务器的体系结构如图3所示。

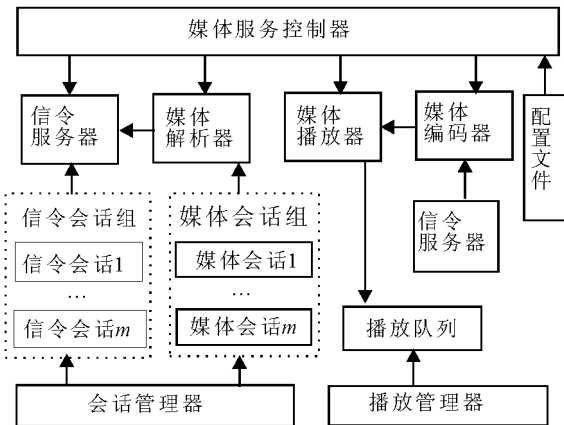


图3 流媒体服务器体系结构图

(1) 信令服务器。信令服务器实现的主要功能是为服务请求者定位和确定请求所需的服务,根据请求者支持的媒体和请求的服务所需媒体的比较来实现媒体协商,在实现媒体协商后建立相应的信令会话,并实现信令会话到媒体会话的一一对应。

(2) 媒体解析器。媒体解析器的主要功能是对信令服务器接收到的 SIP 信令中携带的 SDP 消息进行解析,获得要播放的媒体文件和服务请求者支持的媒体,并根据该媒体文件是否已有相关的媒体会话来确定此次媒体服务所需的参数,再传给信令服务器,以便进行媒体协商。

(3) 媒体编码器。在多媒体领域为了方便网络传输,制订了一些适合以流媒体方式传输的音视频标准,包括 MPEG 系列(MPEG-1/MPEG-2/MPEG-4)和 H 系列(H.261/H.263/H.

264). 不同格式的媒体数据在应用当中会用到不同的编码器, 通过特定算法压缩文件的数据量, 在本系统中采用的是 H. 264 视频编码标准.

(4) 媒体播放器. 媒体播放器要实现的主要功能是根据相应的媒体会话所提供的参数确定媒体编码器的编码方式, 并根据请求的服务来确定媒体播放器应具备的功能(主要是为了便于系统扩展), 最后则是将已编码媒体数据进行 RTP 打包、放入播放队列中.

(5) 会话和播放管理器. 会话管理器的功能是对 2 个会话组进行管理, 主要任务是当媒体数据播放完毕后, 相应媒体会话将标注自己已完成. 会话管理器定期检查, 若发现有已完成的媒体数据播放, 则将删除媒体会话和相应的信令会话信息, 并结束提供对该媒体数据的服务.

(6) 媒体服务控制器. 媒体服务控制器要实现的主要功能是在系统启动阶段根据配置文件初始化各个相关部件, 并将其作为用户接口, 因此, 管理员能通过该接口来实现对系统全局和各个部件的状态查询及控制.

3 流媒体服务器的实现

本设计的流媒体服务器的实现运行环境是基于 LINUX 操作系统, 采用 GNU C/C++ 编码. 由于是基于 SIP 协议和网络广播, 服务器的网络运行环境为支持 SIP/UDP/IPv4 的本地子网. 信令会话和媒体会话的数据结构分别如下:

```

Class AmISession{
    auto. ptr< AmRequest> req;
    AmMSessionKey MSessionKey;
    Void negotiate( AmRequest * request);
};

Class AmMSession: public AmThread{
    AmMSessionKey MSessionKey;
    AmDialogState * dialog. st;
    AmRtpStream rtp. str;
    String MFileName;
    SdpPayload payload;
    Int ISessionNum;
    typedef queue< AmISession * > ISessionQueue;
    /* * AmThread * /
    int getRPort( );
    Void run( ); void on. stop( ); void onIdle( );
};

```

3.1 信令处理

信令处理主要是实现设计的信令服务器的功能. 为了实现客户端与媒体服务器的信令交互和媒体协商, 且由于信令处理通过 SIP 代理服务器与客户端交互, 此处 SIP 代理服务器采用的是 SIP Express Router (简称 SER). 为此需在 SER 的路由表中设置相应的拨号规划, 以便定位媒体服务, 且 SIP 代理服务器与媒体服务器通过 FIFO 来交互信令, 当客户端的 SIP 消息通过代理服务器中转到达媒体服务器时, 媒体服务器接收消息并检查消息合法性后, 在内存中形成了 AmRequestUAS 对象, AmRequestUAS 类是 AmRequest 的子类, 用于通过代理服务器同客户端打交道. 在 AmRequestUAS 对象产生后, 则会根据到来的 SIP 消息的类型进行处理. 若是 Invite 消息则产生新的信令会话 AmISession 对象, 并将新产生的对象放入信令会话队列中, AmISession 对象中主要包括指向客户端请求所产生的对象的指针 req 和对应到媒体会话后的媒体会话关键字 MsessionKey. 若是 CANCEL 消息则找到此消息对应的 AmISession 对象, 并将其从信令会话队列中清除; 若是 BYE 消息则不但要进行 CANCEL 消息相应处理, 而且要对媒体会话作相应处理, 在相应的媒体会话对象中的 ISessionQueue 中的信令对话指针清除, 且 ISessionNum 减 1. 在 AmISession 对象生成后, 再进行媒体协商处理, 调用 negotiate 函数, 当获得正确的 SDP 消息且确认信令会话建立后相应地媒体会话已存在, 此时再将 MSessionKey 赋值为对应的媒体会话的关键字, 实现信令会话到媒体会话的一一对应, 最后将产生的 AmISession 对象放入信令会话队列中以便管理.

3.2 媒体协商过程

媒体协商过程表明了媒体解析器参与工作的过程, 在信令处理过程中需要媒体协商的结果来建立信令会话, 为信令会话和媒体会话的相互关系提供参数, 在 AmRequestUAS 对象生成后, AmRequestUAS 对象中包含有请求播放的文件名, 以此为关键字查询媒体会话组, 若存在相应的媒体会话 AmMSession, 将 AmMSession 中的 SDP 消息 payload 和端口号(通过函数 getRPort 获得) 传回信令处理供媒体协商. 若不存在则将产生新的 SDP 消息和新端口号提供给 negotiate 函数处理, 一个媒体文件对应一个媒体会话, 每个媒体会话采用不同的端口号提供媒体广播服务.

3.3 媒体流建立过程

媒体流的建立则是在媒体会话存在后调用媒体播放器及相应的编码器产生媒体流的过程. 媒体流在产生之前先要有对应的媒体会话存在, 然后调用 run 函数产生相应的线程来产生媒体流, 在产生媒体会话时将用 AmMSession 对象表示该会话, 并将该对象放入媒体会话队列中以便管理. 媒体会话对象 AmMSession 是包括媒体会话关键字 MsessionKey, 播放的媒体文件名

mfileName, 媒体SDP信息 payload, 媒体RTP打包对象 rtp_str, 对应到此次媒体会话的信令会话队列 lSessionQueue 和对应的信令会话数目 lSessionNum 等内容. 媒体会话指向为产生相应媒体编码的对象指针 dialog_st, 则媒体流建立过程为在相应线程启动后先调用 dialog_st 指向的指针获得媒体编码数据, 再由 rtp_str 对其作 RTP 打包放入播放队列准备网络传输, 由播放队列中的 RTP 数据媒体播放管理器中的计时器来触发传输.

3.4 流媒体服务器的管理

在设计中的媒体服务控制器、会话管理器及播放管理器分别由对应的进程来完成其功能, 与上述的各进程组成进程组. 为便于设计的媒体服务器的扩展性, 系统采用了模块化处理, 则媒体编码器及具体的媒体服务功能模块采用了动态链接库(Shared Object)的形式, 所以进程组、动态链接库、信号处理的装载、信令服务器进程都由媒体服务控制器来建立和实现. 当然, 会话管理器和播放管理器也是由媒体服务控制器建立的, 会话管理器主要是清除已完的媒体服务对应的媒体会话和信令会话, 而播放管理器实现了触发播放队列中数据的传输以及监控报文的网络传输状态.

4 结语

以Linux操作系统为平台, 在支持SIP协议的本地子网中实现了一个支持视频广播的SIP媒体服务器的原型系统, 实验测试其基本功能已完成. 运行良好, 随着网络性能的提高, 通过网络向用户提供视频广播也将越来越现实, 相信在未来能突破实验室环境限制, 成为新的网络服务项目, 并对网络教育及远程教育的发展提供新的技术支持.

参考文献:

- [1] 何克抗. 基于多媒体网络的课件脚本设计 [EB/OL]. (2006-02-20) [2007-12-10] http://www.ctc.edu.cn/articledigest4/mukimedia-design_software_via_internet.htm.
- [2] ROSENBERG J, SCHULZRINNE H, CAMARILLO G, et al. SIP: Session Initiation Protocol RFC3261 [S]. [S. l.]: RFC Editor United States, 2002.
- [3] RFC3550 H. SCHULZRINNE, CASNER S, FREDERICK R, JACOBSON V. RTP: A Transport Protocol for Real-Time Applications [S]. [S. l.]: RFC Editor United States, 2003.
- [4] IAIN RICHARDSON, HALSTED PRESS H. 264 And MPEG-4 Video Compression- souxin [S]. [S. l.]: Wiley Editorial Offices, 2004.
- [5] RFC3605 C, Huitema Real Time Control Protocol (RTCP) Attribute in Session Description Protocol(SDP) [S]. [S. l.]: RFC Editor United States, 2003.
- [6] COLIN PERKINS. RTP: Audio and Video for the Internet [M]. [S. l.]: Addison Wesley, 2003.

Design and Implementation of streaming Media Server Based on SIP Protocol Video Broadcasting

LI Xinguo

(Department of Information and Technology, Hunan NO. 1 Normal College, Changsha 410002, China)

Abstract: This article discusses the key technology to provide video broadcasting service based on SIP protocol. It designs the protocol hierarchical model and the system structure of streaming media server. According to the system structure, a prototypical system is realized to support the video broadcasting.

Key words: SIP Protocol; video broadcasting; streaming media server

(责任编辑 陈炳权)