

移动流媒体播放器的设计与实现^{*}

吕伟梅 郑庆华 黄华 林杰 姜山

(西安交通大学电子与信息工程学院 西安 710049)

摘要 3G 移动网络的高速数据传输能力使得移动流媒体的实现成为可能。本文从移动网络和移动流媒体传输协议的特点入手,结合 DirectShow 开发多媒体的优势,探讨了移动流媒体播放器设计中的流媒体服务器的交互、缓冲区的设置、视音频同步以及 RTSP 源过滤器等关键技术,研制出一个适用于移动网络环境的流媒体播放器,并进行了实际验证和测试。

关键词 DirectShow, RTSP, 移动, 流媒体, 播放器

Design and Implementation of Mobile Streaming Media Player

LU Wei-mei ZHENG Qing-hua HUANG Hua LIN Jie JIANG Shan

(School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract With the bandwidth increasing of mobile network from GPRS, CDMA to 3G, which makes it possible to play streaming media on mobile device. A new type of mobile streaming media player based on DirectShow platform is proposed due to its good performance for multimedia functionalities. The paper gives the solution to the main techniques related to mobile media player, such as streaming media server interaction, buffer setting, video and audio synchronization and RTSP Source Filter, and so on. Based on these key techniques, a mobile streaming media player has been implemented and verified on the mobile network environment.

Keywords Directshow, RTSP, Mobile, Streaming media, Player

随着移动通信技术的飞速发展以及手机等移动终端移动处理能力的提高,人们希望能够随时随地得到视频媒体信息的服务。宽带化的无线网络为流媒体服务提供基础,3G 通信标准 TMT-2000 中的 UMTS 承诺用户的接入速度在最快移动情况下可达到 144kbps,步行时可达 384kbps,而在户内可达到 2Mbps。同时,手机等移动终端设备的运算能力越来越强,存储空间也日益增大,已经能够满足基本的流媒体服务。

目前国外 3G 手机播放器占据了绝大多数的市场,由于其技术的封闭性,因此不利于国内厂家进行定制,更不利于国内手机产业的长远发展。同时,随着移动流媒体业务的不断发展,手机播放器需要支持的流媒体格式会越来越多样化。

因此,本文首先介绍了流媒体播放的支撑技术——DirectShow 的原理,分析了移动流媒体协议的组成,提出并解决了移动流媒体播放器实现的关键技术,最后给出了一个适合 3G 手机的流媒体播放器原型系统,进而研制一套基于我国自主研发开发的昊鹏(Hopen)嵌入式操作系统、支持多种媒体格式且具有自主知识产权的流媒体播放器软件。

1 基本原理的介绍

1.1 DirectShow 技术

DirectShow 支持来自本地或网络的各种视频、音频压缩格式的媒体文件的解码和回放,以处理各种压缩算法处理的流媒体。它广泛地支持各种媒体格式,包括 Asf, Mpeg, Avi, Mp3, Wave 等等,使得多媒体数据的回放变得轻而易举。另

外,DirectShow 是一种基于 COM 组件开发的模式,拥有很好的扩展性,所以我们选择了支持 DirectX 具有 DirectDraw, Direct Sound, DirectShow 等媒体处理功能的 Hopen 手机模拟平台作为我们的实验环境,力求在此基础上开发适合 3G 手机的流媒体播放器原型。

DirectShow 应用架构采用模块化组件结构,该模组结构将多媒体数据的处理看成一条流水线,该流水线包含的每道工序由称为过滤器(Filter)的组件来完成。每个过滤器完成特定的功能,根据实现功能的不同,过滤器大致分为 3 类:源过滤器、转换过滤器和表示过滤器,把不同的过滤器组合在一起来处理多媒体数据。应用程序按照所要实现的功能建立起相应的过滤器图表,然后借助过滤器图表管理器来控制整个数据的处理过程。DirectShow 能在过滤器图表运行的时候接收到各种事件,并通过消息的方式发送到应用程序,从而实现应用程序与 DirectShow 系统间的交互。

1.2 移动流媒体的传输协议

在移动流媒体技术中,网络协议占有十分重要的地位,服务器和客户端间的控制信号通过网络传输,同时媒体流也通过网络传输实现。移动流媒体系统的网络协议结构如图 1 所示,其中视频、音频等流媒体数据通过 TCP/UDP 来承载。实时流传输协议 RTSP 是应用层协议,控制实时数据的发送。SDP(Session Description Protocol, 会话描述协议)只是一种用于会话描述的格式,它的目的是进行多媒体会话通知、邀请和一些媒体会话的初始化工作。

^{*}基金项目:本文受到国家自然科学基金重点项目(60633020)、西安市科技计划项目(信息技术专项 ZX06024)以及国家科技支撑计划课题(2006BAH02A24-2)的资助。吕伟梅 硕士研究生,主要研究方向为移动流媒体;郑庆华 教授,博士生导师;黄华 副教授;林杰 硕士研究生;姜山 博士生。

Application Control Commands+SDP	Audio Data, Video Data Sender/Receiver Reports
RTSP	RTP/RTCP
TCP	UDP
IP	
Radio Link/Data Link	
Physical Layer	

图 1 流媒体系统的网络协议结构图

为了实现流媒体的码速调整,3GPP 的 PSS 规范中对于 RTSP 协议在移动分组流媒体中的应用定义了一些扩展字段,主要包括:1)Range 头字段,在 PLAY 的响应中包含这个字段;2)Bandwidth 头字段;3)3GPP-link-Char 头字段,用于移动流媒体客户端向移动流媒体服务器报告无线链路特性;4)3GPP-Adaptation 头字段,用于移动流媒体客户端设置速率自适应参数;5)QoE 头字段,用于移动流媒体客户端和服务器协商移动流媒体客户端应发送哪些 QoS 矩阵,以什么样的频率发送以及如何取消发送这些矩阵;6)Video Buffering 头字段,用于缓冲区的管理。

2 基于 DirectShow 的移动流媒体播放器设计

2.1 系统结构

该播放器主要具有以下三项基本功能:

- 1)支持 wmv, asf, rm, mp4, mp3, 3gp 等格式的播放;
- 2)支持 rtsp 协议的流媒体视音频播放;
- 3)允许用户快进、后退以及暂停等操作。

根据播放器实现的功能需求,深入分析 DirectShow 的工作原理以及移动流媒体网络的传输特点,给出移动流媒体播放器的系统框图,如图 2 所示。

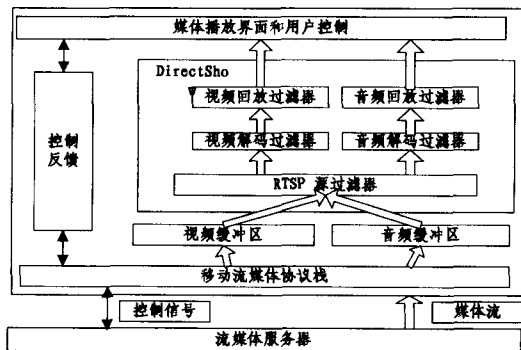


图 2 播放器的系统结构框图

分析播放器的系统结构图,将设计的重点集中在两方面:1)同 Media Sever 进行交互,获取数据,以及合理设计缓冲区和同步机制,确保流媒体的顺畅播放;2)将各模块集成到 DirectShow 的体系中。

2.2 流媒体数据的接收模块设计

为获取网络数据,首先分析播放器同服务器的交互过程,RTSP 中所有的操作都是通过服务器和客户端的消息应答来完成的。移动流媒体播放器获取服务器的媒体数据的交互过程如图 3 所示。

从图 3 交互情况可以看到,流媒体服务器对于客户的要求,会做出如下的反应:1)当客户端通过 RTSP 的 DESCRIBE 方法向服务器发送请求时,服务器会根据用户请求的资源情况,采用 SDP 的信息组织方式返回资源的描述信息,包括媒体的组成、名称、地址、编码方式、时间长度等。2)之后客户端向具体媒体地址发送 SETUP 命令。如果服务器正确建立连

接后,就返回一个独立的会话序列号。根据这个序列号,客户端可以同服务器进一步通信。3)连接成功后,客户端向服务器发送 PLAY 请求,打开 RTP 接收端口。此时需要再开启四个端口,一组为视频接收 RTP/RTCP 端口,另一组为音频接收 RTP/RTCP 端口。此时,客户端就收到媒体的视音频 RTP 包。4)实现 PAUSE 和 TEARDOWN 命令来暂停和结束同流媒体服务器的连接。

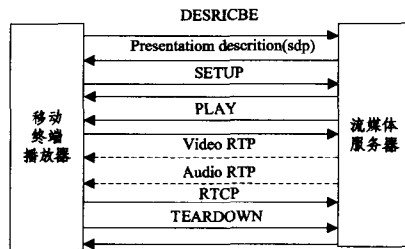


图 3 播放器同服务器的交互过程

在移动网络中传输实时流媒体数据时,网络状况经常变化,带宽不能保证。因此,视音频数据到达终端的网络延迟是不确定的。为避免播放产生空隙,在播放器端设置了一个数据预处理器,对接收到的 RTP 数据进行预处理。预处理主要有以下作用:

- 1)对接收数据包的准入控制;
- 2)将数据包按时间戳存于缓冲队列中。

对于第 2 点,为了便于对缓冲区数据的操作,设计了一个链表结构,用来存储缓冲数据的信息,每个节点对应一个 RTP 包,链表中各节点严格按照时间戳进行排序。实际的 RTP 包按其到达顺序解包后,将数据存放在缓冲队列末尾。当解码器需要从缓冲区读取数据时,从链表头节点开始,根据其指向缓冲区位置,依次读取所需数据。由于视音频数据是分开发送的,因而设计两个双向链表结构来分别存放。链表结构如图 4 所示。

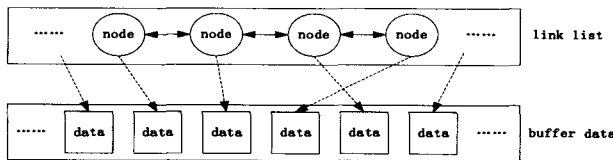


图 4 缓冲区队列及链表结构

对于到达的视音频数据,首先要对其进行流内同步。由于视音频流内同步的处理方法是相似的,因此下面仅讨论视频数据包的处理。当一帧视频数据 f 到达播放器时,首先提取出其时间戳 Vts ,然后在链表中检查第一个节点 f_0 ,取出其时间戳 Vts_0 与 Vts 相比较。若 Vts 小于 Vts_0 ,说明 f 之后的数据帧已经送至解码器进行解码, f 帧到得太晚,已没有解码意义,故将其丢弃;如果 Vts 大于 Vts_0 ,则接纳 f ,将其存入缓冲队列末端,并根据其时间戳 Vts 将其插入链表中的适当位置。

一般情况下,数据以乱序到达播放器的概率是非常小的,即接收数据包的时间戳一般是依次增大的。要保证各节点按时间戳排序,故在节点插入操作时,从时间戳大的一端开始查找其插入位置,确保缓冲区中媒体帧的视频时间戳(Vts)和音频时间戳(Ats)满足下面的关系,确保视音频流内的同步:

$$Vts_0 < Vts_1 < \dots < Vts_i < \dots < Vts_{(n-1)} < Vts_n;$$

$$Ats_0 < Ats_1 < \dots < Ats_i < \dots < Ats_{(n-1)} < Ats_n$$

2.3 流媒体数据的播放模块设计

至此,网络的数据已经成功接收到本地缓冲区中。下面按照 DirectShow 的设计播放器的整体框架,根据需求,设计五个 Filter,包括 Source Filter, Audio Decoder, Video Decoder, Video Render 和 Audio Render。利用 Graphedit 工具设计过滤器连接,如图 5 所示,图中选取了播放 MPEG4 的实例。由于 DirectShow 基于 COM 技术,因此遵循 COM 技术的第三方解码器都可以被 DirectShow 使用,表现过滤器则是 DirectShow 自带的,因此设计的重点就是如何实现接收 RTSP 数据的 Source Filter。

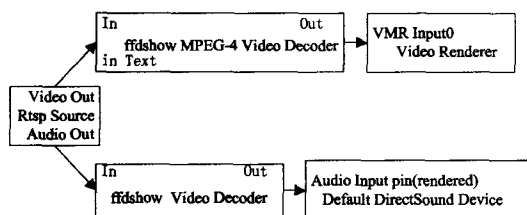


图 5 播放流媒体文件的过滤器连接图

分析源过滤器获取的数据,数据由服务器主动发送,是自动产生的,因此选择符合需求的推模式(Push Model)的数据传输方式。在这种模式下,它的输出 Pin 上有独立的子线程,负责将数据发送出去。源过滤器的任务为设置媒体类型,并为下游解码器提供视音频 Sample,设置正确的时间戳。DirectShow 提供了推模式的源过滤器的基类 CSource 和相应 Pin 的基类 CSourceStream,将网络接收模块和数据处理模块集成到 Source Filter 中,同时继承 CSource 和 CSourceStream 来实现。

在接收模块中,已经处理了媒体流内的同步,此时则运用 DirectShow 的机制来实现流间同步,选一个公共的参考时钟,给每个 Sample 打上时间戳。为了确保单路视频或音频的准确性,选取系统时间为参考时钟,Video Render 或 Audio Render 根据 Sample 的时间戳来控制播放,同时引入反馈机制。

GetCurrentTime()获取系统当前时刻。DirectShow 在播放开始时,调用 GetCurrentTime()获取起始时间,这个值记为 StartTime。经 Run 函数传入各 Filter,以表明播放开始时刻,各 Filter 予以记录。当一帧数据到达时,Render 级以该帧的时戳加上开始时刻作为本帧的播放绝对时刻,向 IReferenceClock 申请播放相对时刻。

由于 RTP 的时间戳长度为 32,而参考时钟为 64 位,因此帧的绝对时间戳(FrameTime)和等待时间(WaitTime)的计算公式如下:

$$\text{FrameTime} = \text{当前帧包头时间戳} - \text{第一帧的时间戳}$$

$$\text{WaitTime} = \text{StartTime} + \text{FrameTime} - \text{GetCurrentTime}() / 1000$$

WaitTime 的单位为 ms。如果 WaitTime < 0,说明该帧到达过晚,理论上应该丢弃。但直接丢弃容易引起视音频的不连贯,所以将 WaitTime 在 -10~5ms 的数据包都进行处理。

3 软件实现及播放效果

根据本文中对播放器的整体设计和关键模块的设计,同

时结合吴鹏手机模拟平台下对界面设计的具体规范,实现了移动流媒体播放器。

同时,在以 802.11b 无线网卡连接的无线局域网内部进行构成慢移动环境进行测试。采用 RealNetWorks 公司的第 10 版本移动流媒体服务器,它支持传送 50 多种的流媒体格式,包括 3g2 和 mp4 格式。图 6 为运行在 Hopen 手机模拟平台上的播放器在线播放 mp4 格式的效果图。测试显示播放效果良好,播放视音频流畅并且基本同步,网络带宽占用 57kB,内存占用 34MB,达到预期的目标。



图 6 播放器运行效果图

结束语 本文讨论了基于 DirectShow 的移动流媒体播放器的设计和程序开发,针对移动网络的不稳定、带宽小的特点,采取 RTCP 的反馈机制、缓冲区管理和视音频同步控制来确保播放器的播放质量,同时结合 DirectShow 的优势,进行播放质量的控制。

但由于时间有限,在本项目的实现过程中,对于视音频解码部分没有做深入研究,只是利用现有的解码器,没有做相应的优化。同时,虽然该软件在 Hopen 手机模拟平台运行良好,但是并没有在真实环境下运行,因此下一步还需要进一步优化软件播放性能,将该播放器真正应用到手机上。

参考文献

- [1] 陆其明. DirectShow 开发指南[M]. 第 1 版. 北京:清华大学出版社,2004
- [2] 3rd Generation Partnership Project. Transparent End-to-End Packet Switched Streaming Service (PSS) General Description (Release 4). Technical Report, 3GPP-TS-26. 233-V4. 0. 0. Technical Specification Group Services and System Aspects, UMTS Forum, 2000
- [3] Song H, Kuo C-C-J. Rate control for Low Bit Rate Video via Variable Encoding Frame Rates. IEEE Transactions on Circuits and Systems for Video Technology, 2001, 11(4): 512-521
- [4] Kyriakidou A, Karelos N, Delis A. Video Streaming for Fast Moving Users in 3G Mobile Networks// ACM MobiDE'05, Baltimore, Maryland, USA, June 2005
- [5] 余江,刘威,王泰,等. 流媒体代理缓存技术研究[J]. 计算机科学, 2006(1)
- [6] 张钊,谢忠诚,鞠九滨. 基于实时传输协议的丢包实时修复[J]. 软件学报, 2001, 7(12)