



一种视频会议系统的研究与实现

黎杨梅

(武汉软件工程职业学院 电子与电气工程系,湖北 武汉 430205)

摘要:介绍了使用局域网组播技术和 Directshow 多媒体开发框架实现多方视频会议系统的方法。系统引入了会议管理服务器以控制多媒体数据在网络中的接入和访问,使网络不会由于多媒体数据的激增而发生拥塞,从而保证服务质量。系统使用了面向对象的方法和 UML 语言,分析和设计了各个模块。系统的视频部分采用了 MPEG-4 编解码技术,音频数据留采用了 G.729 编解码方法。根据该设计方案,完成了该系统的设计和开发,实现了多人同时通讯的模拟视频会议功能。

关键词:组播;DirectShow;RTP;视频会议

中图分类号: F507.454

文献标识码: A

文章编号: 1671-931X (2012) 05-0068-05

68

武汉职业技术学院学报二〇一二年第十一卷第五期(总第六十一期)

在各种通信业务中,多媒体视频会议是近年来发展较快的一项业务。视频会议(Video conference)是指多个不同地方的个人或群体,通过现有的各种通信传输网络,将人物的图像、语音、数据等多种资料分送到各个用户的计算机上,实现实时互动的沟通,达到会议目的的系统设备。多媒体视频会议系统是集现代数据通信与网络技术、信息处理技术、微电子技术、光电技术、计算机软硬件技术于一体的一种远程异地通信方式。当前,视频会议系统的产品主要分为硬件视频会议系统和软件视频会议系统。传统的视频会议系统以硬件为主,其技术主要源于电路交换系统,核心是多点控制单元(MCU)。硬件视频会议系统速度快捷、传输效率高、保密性强;其缺点在于费用较高,而且只能在专线网络环境下运行,安装复杂、使用不够灵活。软件视频会议系统可以利用现有的网络环境和计算机设备,并能够提供不逊色于硬件系统的音视频质量和更为丰富的数据协作功能,而且其系统成本较低,所以前景非常广阔。

一、视频会议系统的分析与设计

与会者参与视频会议采用多用户间点对点的方式。当某台终端有与会者登录后,服务器就会收到相应的与会者信息和终端信息,然后服务器选取一个编号和组播地址标识新加入终端,并且把当前所有的终端编号及其组播地址列表发给所有在线终端;所有在线终端接到该列表后加入到其它的在线的终端的组播组。新加入的终端启动视音频采集和组播发送,并同时接受其他终端的组播多媒体数据,如图 1 所示。

视频会议管理在视频会议服务器上实现,可以启动、停止视频会议,以文本的方式显示信息,查看会议进行时间等功能,可以监视终端联网情况、修改终端编号和重启终端的功能,如图 2 所示。

二、系统的实现

本次开发全部在 Win32 系统下完成,使用的开发软件为 VC6.0,网络部分使用 Socket API 和

收稿日期:2012-09-01

作者简介:黎杨梅(1984-),女,湖北荆门人,武汉软件工程职业学院电子与电气工程系,助教,研究方向:应用电子。

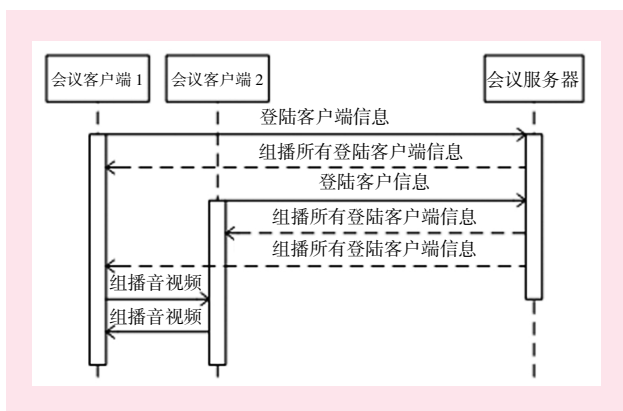


图1 用户加入视频会议

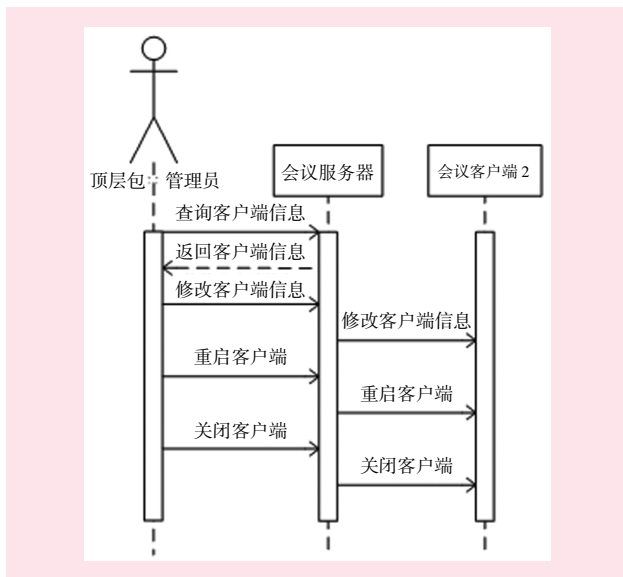


图2 终端管理时序图

JRTPLIB 开发包实现，视频采用 DirectShow 系统框架。

(一)建立系统 Filter Graph

网络连接方面，视频服务器和各个终端之间都单独的采用 TCP 方式的 Socket 通讯,这些面向连接的通信方式用来保证与各个终端之间控制指令的可靠传输;每个终端都加入一个组播组,通过不同的端口进行多媒体数据流的多终端实时传输。网络连接方式如图 3 所示。视频服务器实现整个网络系统的调配管理。当网络客户端有图像访问请求时,首先向

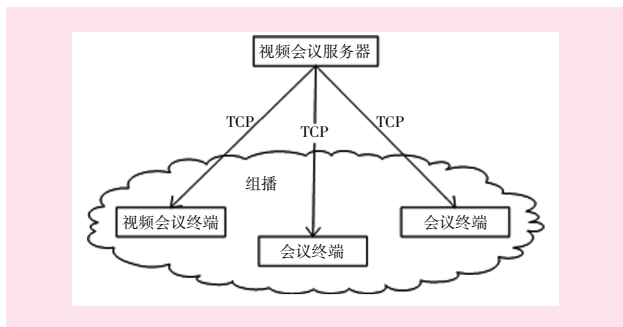


图3 网络结构示意图

视频调配服务器约定的 TCP 端口发送图像访问请求信息,视频调配服务器收到该请求后,将该监控点视频的组播号传送给请求者,后者只要加入该组播组即可实现对视频的访问。原始设计采用了较为简单的循环式服务器,让应用程序循环的跟踪每个连接的不同信息,循环的等待客户请求,在同一个时刻只可以响应一个客户端的请求,正在处理当前客户任务时,对其他客户请求不作任何响应。在视频终端数量较少的条件下,这种服务器方式可以满足需求,而且具备设计简单的优点;但当终端机数量较多、控制信息发送频繁的情况下,由于对终端信息处理的独占性,系统延时会很大,延时过久甚至会导致网络阻塞。所以最终废弃了串行方案而改为并发式服务器。并发式服务器为每个终端连接创建一个单独的服务线程来处理其请求是更为有效地实现方法。每一个客户机的请求并不由服务器直接处理,而是服务器创建一个子线程来处理,每个独立线程只需要管理与其对应的连接状态信息即可。

根据需求每个与会者都能同时接受到其他所有与会者的音视频以及自己的视频。为了实现这一目的,需要组建四个 Filter Graph,分别负责音视频的采集发送和音视频的接收回放,每一路 Graph 采用一个线程来实现和管理。DirectShow 框架定义了两种工作模式:推模式(Push Mode)和拉模式(Pull Mode)。本系统采用推模式。在该模式下,数据被 Source Filter 主动推向与其相连的 Transform Filter,后者再将处理过的数据推向下行 Filter。推模式通常使用在实时数据情况下。对于实时数据源,数据传输速率可能不是恒定的(例如网络媒体传输、视频采集卡等),因此采用推模式,Source Filter 可以根据数据源的实际状况决定如何将数据传给下行 Filter。滤波器图形链路构建是最终实现的关键部分,通过构建链路把各个功能的滤波器模块组合起来,为滤波器图形的各种控制做好准备。智能连接使用十分方便,但引入的额外滤波器是不确定的,所以使用它构成的滤波器图形具有不稳定的特性。组建了四路 Filter Graph,以下分别说明。图中每一个方框表示一个 Filter。

1.视频流采集、智能劈分、预览、MPEG-4 压缩编码和网络发送部分的 Filter Graph,如图 4 所示

该 Filter Graph 主要由 USB 视频采集 Filter、智能劈分(Smart Tee)Filter、帧率控制 Filter、DivX 压缩编码 Filter、视频回放(Video Renderer)Filter 和 RTP 组播(RTP Sender)Filter 构成。除了 RTP Sender Filter 外,基本上都是常用 Filter;RTP Sender Filter 是新建的一个内联 Filter,用于完成网络发送和质量控制,在后面会有进一步详细解释。DivX Codec Filter 提供 MPEG-4 的视频编码,达到压缩视频数据的目的。Smart Tee Filter 用于将一路数据流分为两

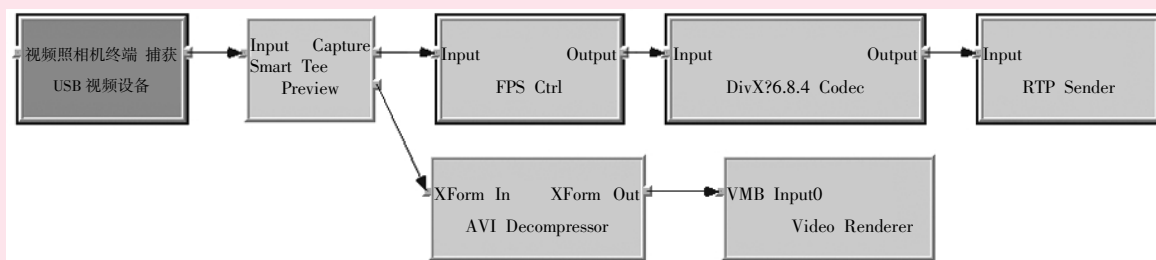


图4 视频采集、编码和发送 Graph

路。第一路 (Capture Filter→DivX Codec Filter→RTP Sender Filter), 用于将采集到的本地视频 MPEG-4 压缩后进行组播发送。由于是需要经过网络发送并需要在接收端和音频数据同步, 这路数据流需要较为精确的同步时间戳 (这里的时间戳要与前文所说的 RTP 时间戳区分开来, 这里的时间戳是用来进行音视频同步的, 而前文所说的 RTP 时间戳是为了保证网络数据到达的次序不发生混乱)。Capture Filter 就将数据写入到 sample 的缓冲区中去, 选一个公共的参考时钟, 并在 sample 上设置适当的时间戳, 然后将它递交到下一个 Filter 去; 第二路 (Capture Filter→Video Renderer Filter), 用于本地显示视频预览。由于本地预览这组数据不需要音频和远程发送, 所以不需要在数据流中加入同步时间戳。

2. 视频流网络接收、MPEG-4 解码和回放部分的 Filter Graph, 如图 5 所示

这是每个终端接收视频的 Filter Graph, RTP Receiver Filter 将网络接收到的数据处理后推至下一个 Filter, DivX Decoder Filter 将这些数据进行解码, Video Renderer Filter 将最终的视频数据回放出来。在接收流媒体数据时, 要注意同步音视频的处理。简单地说, DirectShow 的解决方案就是为 Filter Graph 选择一个公共的参考时钟 (Reference Clock),

并且要求传送到 Renderer Filter 的每个 Sample 都打上时间戳 (Time Stamp); Video Renderer 或 Audio Renderer 根据 Sample 的时间戳以及参考时钟当前的参考时间, 来正确安排 Sample 的播放时机。如果到达 Renderer Filter 的 sample 晚了, 则马上播放这个 Sample, 并发送消息要求加快 Sample 的传送; 如果 sample 到达的时间早了, 则 Renderer Filter 等待, 直到 Sample 时间戳的开始时间再开始播放, 并发送消息要求减慢 sample 的发送。

3. 音频流采集、G.729 压缩编码和网络发送部分的 Filter Graph, 如图 6 所示

通视频采集发送的图表相类似, Audio Capture Filter 负责采集音频数据, G.729 Codec Filter 是采用 ITU-T G.729 源码库文件编写的压缩编码 Filter, RTP Sender Filter 是网络组播发送部分。同样为了使接收端达到同步, 需要加盖时间戳。默认情况下, Audio Capture Filter 使用 0.5s 的缓冲。但对于一些特殊的应用, 这么大的缓冲是没有必要的, 它带来的延迟也是不能容忍的。一般缓冲设置成能够容纳 30ms~80ms 的数据已经很可靠了; 但是也不能太小, 否则会影响音频采集的效率, 严重时损害音质。

4. 音频流网络接收、G.729 解码和回放部分的 Filter Graph, 如图 7 所示

类似于视频的接收图表, 由 RTP Receiver Filter

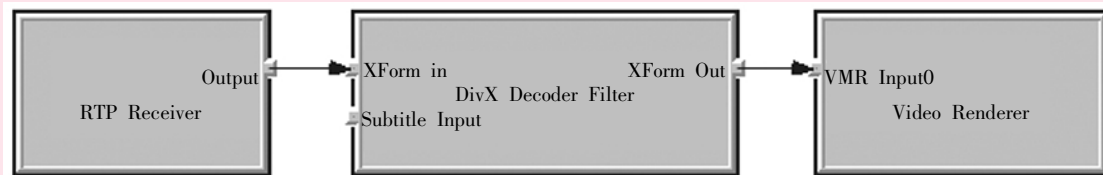


图5 视频接收、解码和回放 Graph

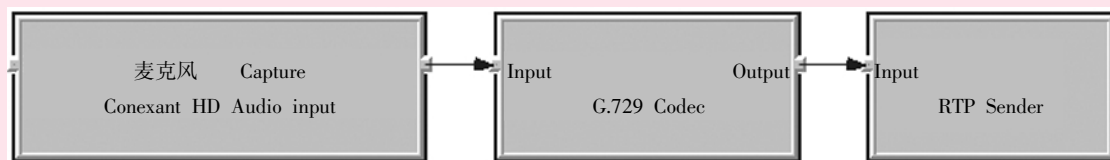


图6 音频采集、编码和发送 Graph

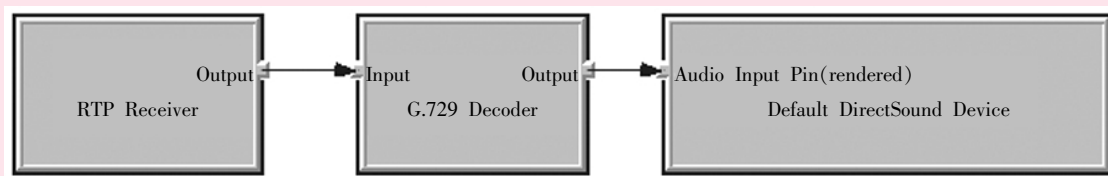


图7 音频接收、解码和回放 Graph

将网络接收到的数据处理后推至 G.729 解码 Filter, 解码后调用音频设备将其播放。在这个过程中要注意与视频数据同步的问题。

需要说明的是, 每一个会议终端都具有组建上述 Filter Graph 的能力; 而视频会议服务器在使用视听模块时, 会组建相应的音视频接收 Filter Graph (Graph2 和 Graph4), 负责接收音视频数据。

(二) 音视频编解码模块的实现

音频编解码是作为内联 Filter 实现的(即不需要发布和注册的 Filter, 只在本程序中实现), 继承于 Transform Filter 类(CTransformFilter)。实现思路是将输入 pin 的数据流经过压缩编解码后推向下一级 Filter。音频编解码的实现是采用 ITU-T G.729A 源码库文件, G.729A 是 ITU 最新推出的语音编码标准 G.729 的简化版本。包含 G729AENC、G729ADEC 两个库, 提供了关于 G.729 的相关编解码接口。本系统采用的是 40ms 的缓冲, 即采用 $40\text{ms} \times 44.1\text{kbps} = 1764\text{bit}$ 的缓存。每个 sample 分四路进行 G.729A 编码, 接收端采用类似的方法来实现解码。相对于自己编写的音频编解码 Filter, 视频编解码的实现较为简单, 采用 DivX 提供的 MPEG-4 编解码 Filter。安装 DivX 后会自动安装 DivX 的编码器 Filter 和解码器 Filter。加入 Divx 的编码器 Filter 的实现, 是在 Video Compressors 目录下枚举到友好名(Friendly Name)为“DivX 6.8.4 Codec”的 Filter 项后, 把它加入到 Filter Graph 中即可。在接收视频的过程中, 采用与之前相似的方法, 通过 CLSID 来查找并调用。

(三) 网络传输与控制模块的实现

网络客户端是一台普通的 PC 机, 它通过软件的方法, 可以实现多路音视频流的解码输出。网络客户端首先需连接视频调配服务器指定的 TCP 端口并提出对某个监控点实施远程监控的请求。视频调配服务器将之转发给目标监控主机的约定的端口, 由目标监控主机来具体实施该请求。系统以 PC 机为终端, 用话筒和摄像头实现音视频信号的采集, 以 RTP 协议方式实现音视频数据的 IP 组播方式传送。系统采用点对组工作方式, 即每一个终端都将 MPEG-4 视频流封装在 RTP 包中, 并定时向一个组播地址发送数据并接受 RTCP 包进行反馈和控制; 其它终端机均要加入到该组播组以实现音视频数据的接收, 然后通过 DirectShow 技术在各自的终端上进行回

放。

(四) 质量保证模块的实现

由于视频数据量很大, 如果要在较小的带宽条件下达到更高的清晰度和实时性, 就必须降低帧率(FPS, Frame Per Second); 但降低帧率后, 图像的连贯性就无法得到保证。

开发帧率控制过滤器(FPS Ctrl Filter)就是为了协调这一矛盾而开发的。它的基本思想就是根据网络负载状况, 选择性的主动丢弃一些非关键帧。这样, 系统就可以根据网络状况动态调节视频数据量, 从而保证网络传输质量和避免拥塞。

FPS Ctrl Filter 继承于 CTransformFilter, 上述功能在该 Filter 的接收函数中实现。该 Filter 相当于一个视频帧数计数器, 每接收到一帧, 并不立即把该帧发给下游的编码 Filter, 而要经过判断比对。

(五) 死锁问题的解决

系统按原设计实现后, 出现了死锁问题, 即接收端有多路音视频进行回放时经常发生视频画面停止的现象。在进行调试时发现, 如果将 MPEG-4 的编解码器都去掉, 只是进行简单的采集、发送、接收和回放时, 不会出现死锁的问题; 另一方面, 发现如果只有一路视频数据进行编解码时, 也不会出现死锁问题。

经过上述实验, 初步认为死锁是由 DivX 编解码器造成的。在更换了多个版本的 DivX 编解码器后, 死锁现象仍然存在。针对上述问题, 采取了以下两个办法: (1) 采用经过改进的 6.8.4 版本解码器, 该解码器被广泛应用于视频会议系统, 包括在国内知名的 V2Conference 视频会议软件。6.8.4 版本解码器已被证明改进了共享内存的算法, 使的界面刷新消息能够分别得到响应。(2) 采用每组 Graph 一个线程的方法, 使用 COutputQueue 类来为每个 output pin 创建传送数据的线程; 同时增大 output pin 的 sample 数量, 增大缓存。采用上述改进方法后, 界面刷新死锁的问题终于得到解决。

三、结束语

经过测试, 该软件视频会议系统可以利用现有的网络环境和计算机设备, 达到视频会议的要求, 功能完善, 性能稳定。编解码性能良好, 视频图像清晰, 无雪花和马赛克等现象, 音频延迟在 1 秒以内, 而且

其系统成本较低,所以前景非常广阔。

参考文献:

- [1] 周长发.精通 Visual C++图像处理编程(第3版)[M].北京:电子工业出版社,2003.
- [2] Douglas C.Schmid,Stephen D.Huston.C++网络编程(1)[M].北京:电子工业出版社,2007.
- [3] 陆其明.DirectShow 实务精选[M].北京:北京科学出版社,2003.
- [4] 石竣.Windows 环境下的实时视频捕获技术[J].计算机工程,1999,(8).
- [5] 李树前.基于 MPEG-4 的网络视频监控技术研究[实现][D].南京:南京航空航天大学,2007.
- [6] 马兰,沈笑云,方棣.精通 VC 视频/音频编解码技术[M].北京:人民邮电出版社,2008.

[责任编辑:刘 骋]

The Research and Implementation of Video Conference System Based on Multimedia Communication Technology

LI Yang-mei

(Department of Electronic and Electrical Engineering, Wuhan Vocational College of Software and Engineering, Wuhan430205, China)

Abstract: A video conference system based on the multicast technology and Directshow framework is proposed in this paper. The conference management server is introduced to handle the access of the multimedia data and avoid network congestion, which can also enhance the quality of the service of network. By using the object orient modeling and UML (unified modeling language), the paper analyzes and designs modules for real-time operation. Video streams are encoded in MPEG-4 format while audio streams are compressed in G.729 format. In this way, the system is designed and developed based on the scheme, fulfilling the anticipated requirement.

Key words: multicast; Directshow; RTP; video conference system