

基于应用层组播的远程授课直播系统的设计

Design of Remote Schoolteaching System Based on Application Layer Multicast

刘秀花^① 张 卫 (华东师范大学 计算机科学系 网络实验室 上海 200241)

摘要: 远程同步讲授型教学模式对网络带宽、传输时延等指标的要求较高,约束了教学系统的规模。本文结合应用层组播技术的特点,提出了基于应用层组播的远程授课系统的设计,阐述了系统总体框架的设计过程。该系统可以通过教师授课直播室用相对有限的网络资源完成屏幕演示、现场直播、文件管理/下载、消息广播等四个方面的远程教学应用。

关键词: 应用层组播 远程教学 MRTP 学生机端/教师机端 远程授课直播

1 引言

现代远程教学中为方便远程学习者较直观快速的学习,以视频会议为主的实时在线远程教学往往成为学习者的首选。这种同步讲授型教学模式在实现时普遍采用点对点实时传输技术,网络资源的开销开始时较小,但是随着接入主机数目的增多,服务器数据传输负荷明显增大,占用越来越多的服务器资源和网络资源,甚至造成网络拥塞,需要不断增加网络带宽,这些都会影响系统的扩展。而随着计算机网络的发展,应用层组播技术作为组播网络的前沿技术得到了广泛的研究并取得一些成熟的理论,其最显著的特点是“时间上集中,空间上分布”,单源发送一个包,多端接收,充分利用带宽和计算机资源,具有较好的数据传输效率,弥补了已有实时网络在可扩展性和伸缩性等方面的不足。应用层组播技术所具有的这些特点较适合远程同步讲授型教学模式。鉴于此,本文提出了基于应用层组播技术的远程教学系统架构,阐述了系统总体框架的设计方法,通过教师授课直播室用相对有限的网络资源完成屏幕演示、现场直播、文件管理/下载、消息广播四个方面的远程教学应用。

2 应用层组播实时传输协议(MRTP)

应用层组播的基本思想^[1]是把传统的组播功能放

在应用层来实现,保持互联网原有的简单、不可靠、单播的转发模型,由端系统实现组播数据路由、复制、转发功能,在成员主机之间建立一个叠加在 IP 网络之上的、实现组播业务逻辑的功能性网络,不需要附加额外的硬件设备,便于实现和推广,也便于针对特定应用优化。

本远程教学系统使用的协议是应用层组播实时传输协议(MRTP, Multicast Real-time Transport Protocol)^[2,3]。MRTP的基本思想是:系统中包括一个根服务器,一个用户管理服务器,以及若干用户节点组成;组播网络是以根服务器为根的组播树,数据从组播树的根结点发出,沿着树的分支不断转发,到达各叶子结点。在 Internet 中的每一台主机连接到用户管理服务器,发送加入指定组的请求,并附上自己的 IP 地址等信息;经用户管理服务器的认证许可后获得网络信息并加入到网络中,成为组播树上的一个节点。每个子节点担负着不断寻找父节点、传送数据,并接收父节点传过来的祖父节点信息,用于当父节点离开时子节点的切换。用户登录后每隔 10 秒钟必须重新请求服务器一次,用于更新信息;若 30 秒钟没有请求服务器将其删除。对于开始发送等控制信息在 TCP 通道上传输,保证了控制数据的可靠性;而真正的数据在 UDP 通道上传输,保证了数据的高效实时性,两者结合数据

^① 刘秀花的第二个单位:山东省潍坊市机械工业学校(262100)

能被快速可靠的传输。当出现错误时仅重传一次,通过 MRTP 协议由终端进行复制转发,外加自适应缓存管理的使用,能充分利用带宽流畅地传输数据,保证了数据的实时性和同步性,较适合同步远程实时教学的应用需求。

3 系统架构的设计

该远程教学系统从总体上来说是一个基于 C/S 模型、架构在 MRTP 上的应用系统。系统程序可以分割为服务器端和客户端,分别位于教师机端和学生机端,这两部分都需要到一个用户管理服务器进行注册,以便于被管理和认证,从而保证每个组播组的唯一性。下面从硬件结构和软件结构两个方面阐述系统总体设计架构。

3.1 硬件结构

基于 MRTP 的远程授课直播系统所需的硬件结构包括计算机通信网络、web 服务器、摄像系统、话筒和音响等设备。教师机、学生机和用户管理服务器都是 Web 服务器,它们的通信是通过计算机通信网来传输的;另外摄像系统、话筒和音响等设备是视频音频传输所必须的硬件设备。

3.2 软件结构

3.2.1 原理架构

由授课端、听课端及应用层组播授课内容组成了上层远程授课直播系统,即 MRTP 的上层应用。授课端模块就是教师节点模块,主要负责采集源数据、对采集的数据源进行压缩处理,借助 MRTP 协议将数据发送出去,是数据的发送端。听课端模块就是学生节点模块,主要负责从应用层组播平台接收数据,并对接收的数据进行组合、解压、还原处理,是数据的接收端。授课端模块和听课端模块的运行前提是需要优先向用户管理服务器成功注册。

由授课端模块和听课端模块组成教师授课直播室,它和底层的应用层组播平台之间通过控制接口 MRTPControl 进行消息的传递,如图 1。MRTPControl 主要负责 TCP 控制通道的协议实现,维护协议状态,控制协议动作。教师授课直播室的服务器端数据源根据要传输的媒体不同可分为屏幕传输、视频传输、音频传输、文字传输等,它们通过控制接口 MRTPControl 利用自定义的消息机制,借助应用层组播网络进行实时同

步传输。学生端发起对服务器的成功连接后,同步接收这些数据。授课端通过 MRTPControl 控制接口不断向下层应用层组播平台扔消息,由 MRTP 进行数据传输,而听课端通过下层的 MRTP 控制接口不停地接收应用层组播消息。

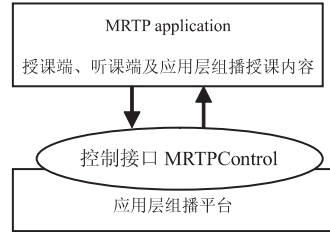


图 1 基于 MRTP 的远程授课直播系统

3.2.2 系统功能模块

随着多媒体技术的发展,远程同步实时教学需要传输的数据量也越来越多,尤其是视频数据,成为远程教学发展的瓶颈。应用层组播数据源只发送一次,由加入组播组的终端进行复制转发,网络中传输的数据量相对较少,能够较大程度上减轻服务器数据传输的负荷,充分利用带宽。教师授课直播室系统就是在应用层组播技术的基础上开发的,根据教师授课直播的媒体不同,传输的数据可以划分为图像、视频、音频、文本等形式,因此教师授课直播室系统主要由屏幕传输模块、视频模块、文件管理模块、消息广播模块组成,如图 2,它们的功能如下:

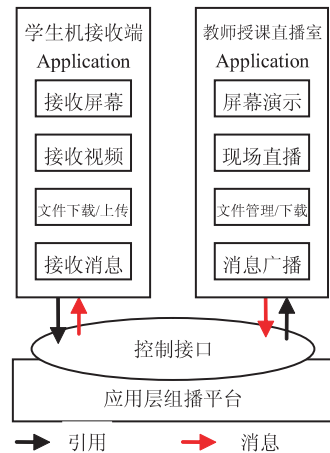


图 2 基于应用层组播的远程授课直播系统节点框图

- 屏幕演示是教师从教学资源中选择授课用的课件进行同步播放,或者教师对授课内容进行屏幕演

示操作的过程;接收屏幕是学生端同步接收教师端屏幕信息,方便学生同步学习。

- 现场直播用来直播优秀教师举行示范课的过程,接收视频是学生可以同步接收优秀教师的授课过程,该模块需要传输的数据量较大,与其它已有的实时视频传输相比,本系统的优势在于能充分利用带宽,减轻服务器的数据负荷,保持良好的同步性。

- 文件管理或下载是教师对教学资料进行管理,指定下载的教学资料或学生上传的优秀作品;文件下载/上传是学生可以接收教师机发来的文件,还可以上传作业,便于师生之间信息传送。

- 消息广播是教师机端下达一些通知信息;接收消息是学生机接收教师机发来的消息,明确本次学习目的,便于同教师机保持同步。

4 系统中用到的关键技术

该远程教学系统是以 MRTP 为基础,在 Windows 环境下以 Microsoft Visual Studio .NET 2003 作为程序设计平台,采用典型的客户/服务器结构来实现的。在系统中采用了 VC.NET 的多线程、TIMER、消息队列和事件处理等程序设计技术,具体在实现每一个模块时还要解决好如下一些技术问题。

4.1 远程屏幕同步传输

远程屏幕图像传输中用到的关键技术有屏幕图像动态感知变化的获取、屏幕图像的压缩和屏幕图像的远程传输。(1)捕获的屏幕图像考虑到用 bmp 位图操作会占用较大的内存空间,引用图像数据结构困难,速度不快,因此本系统中采用 directdraw^[4] 直接对硬件显存进行读写操作,速度较快,较好地保证系统远程传输的实时性。(2)因用 BitBlt() 函数连续捕捉屏幕图像得到的数据量太大,难以直接传输,而相邻的两个屏幕图像在时间和空间上具有很大的相似性,为减少传输的数据量,采用帧内帧间二次压缩算法,提取变化图像的轮廓进行压缩。具体方案如下:

- ①将相邻的两屏图像进行帧间行程压缩:逐行对应比较,记录下发生变化的部分;

- ②对变化部分进行帧内 zlib 压缩,由 int compress (Bytef * dest, uLongf * destLen, const Bytef * source, uLong sourceLen) 来实现;

- ③将得到的压缩数据分块通过 MRTP 传输;

- ④当数据包到达接收端时将收到的数据包进行组合;

- ⑤接着用 zlib 的 int uncompress (Bytef * dest, uLongf * destLen, const Bytef * source, uLong sourceLen) 解压操作;

- ⑥再做一次帧间行程解压缩运算进行还原;

- ⑦最后通过输出设备——屏幕显示出来。

经过上述处理能够完整、无冗余的监测到屏幕图像的变化,对捕获的数据经过两次压缩,减少数据传输量,较大程度地提高屏幕图像传输速率,再加上应用层组播由端系统来实现数据的复制转发,较好地避免抖动时延的发生。另外,为使新加入的节点能较快的接收同屏信息,还要定时的发送捕捉的原始屏幕图像即关键帧,如图 3:

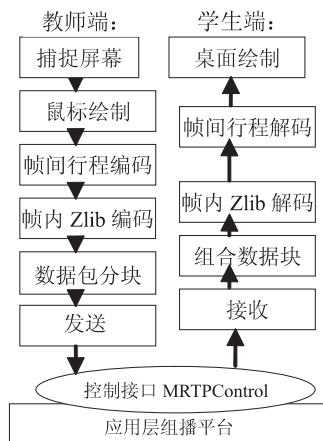


图 3 屏幕图像模块程序流程图

4.2 现场直播

考虑到视频处理与硬件设备无关性,并能较好的支持新式多媒体设备,现场直播主要采用 DirectShow^[5] 技术来完成的。DirectShow 是 Microsoft 为高性能多媒体应用而开发的底层应用程序接口 (API)。它提供的是一个开放性的应用框架,是一套基于 COM 的编程接口。Filter 是 directshow 中的最小 com 模块,每个 filter 上有一个或多个 pin,用 pin 把相邻的两 filter 连成链,由 Filter Graph 对 filter 进行管理。在这个现场直播模块中需要自己开发发送过滤器和接收过滤器。基本设计思路如下:在教师端完成视频的实时采集,需要用到音频视频捕捉过滤器;通过分离过滤器将采集到的视频流分为本地播放数据流、视频数据流、音频数

据流,对视频数据经由 MPEG-4 标准的 Xvid Video Codec 进行编码解码,对音频数据采用了 Mp3 的方式进行编码解码,然后将编码后的数据送到视频转发端,并进行同步合成处理,当捕捉的数据写满缓冲区后,把缓冲区里的数据一次写入管道,由 Sender filter 启动相应的发送线程,将编码后的数据分片通过 MRTP 协议发送到网络上,然后清空缓冲区,等待接收下一批数据。学生端刚好是反过程:由 Receiver filter 从网络上接收数据,组合成同步数据,分别对其解码,同步处理后在学生机上回放视频音频,如图 4 所示:

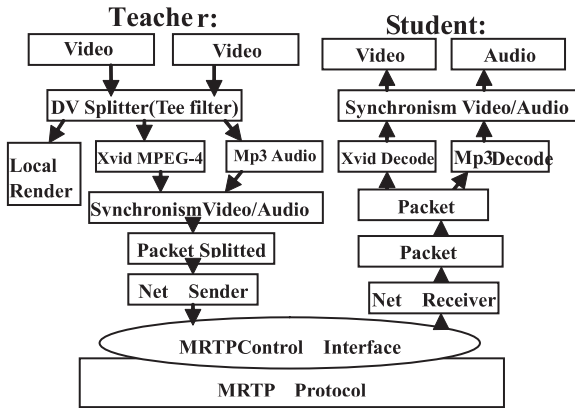


图 4 屏幕图像模块程序流程图

(1) 准备工作

要完成视频的采集,首先要实现采集 Filter Graph 构建,但是创建带有采集设备的 Filter Graph 一般比较复杂,可以先用 GraphEdit 来构建,再以 GraphEdit 中的 Filter 连接图为模板用程序代码来实现。为保证能够在不同机器不同操作系统支持下使用已有的多媒体设备,用系统枚举来获得机器上装有的捕捉设备。系统枚举大致过程如下:

- ①使用 CoCreateInstance 函数创建一个系统枚举组件对象,并获得 ICreateDevEnum 接口。
- ②使用 ICreateDevEnum::CreateClassEnumerator 为指定的类型目录创建一个枚举器,并获得 IEnumMoniker 接口。
- ③使用接口方法 IEnumMoniker::Next 枚举指定类型目录下所有的设备标识(DeviceMoniker)。每个设备标识对象上都实现了 IMoniker 接口。
- ④调用 IMoniker::BindToStorage 之后就可以访问设备标识的属性集。

⑤调用 IMoniker::BindToObject 把设备标识绑定成一个 DirectShow Filter,再调用 IFilterGraph::AddFilter 加入到 Filter Graph 中就可以参与工作了。

(2) 构建捕获的过滤器

DirectShow 专门提供的 Capture Graph Builder 辅助组件(CLSID_CaptureGraphBuilder2)简化了 Filter Graph 的构建,使得 Capture Filter 加入 Filter Graph 后,能够与其它 Filter 相连。同时调用 QueryInterface 函数获取对象指针,实现播放控制或发送消息到应用层。该步的大体思路如下:

- ①用 CoCreateInstance 创建 Filter Graph
- ②用 CoCreateInstance 创建 Capture GraphBuilder
- ③ CaptureFilter 与 Filter graph 连接在一起,组成捕获回路:hr = m_pCapture->SetFiltergraph(m_pGB);……
- ④调用 QueryInterface 取得本地回放接口指针。

音频捕获是应用 DirectSound 接口程序来实现,代表声卡的 Filter 注册在 CLSID_AudioInputDeviceCategory 目录下,利用设备枚举器的枚举,找到要创建的声卡对象。该声卡使用的 Filter 是 DirectSoundCapture Filter, Filter 内部则使用以 wavIn 开头的一套 API 实现,应用 DirectSoundCapture 对象和 DirectSoundCaptureBuffer 采集缓冲区来共同完成声音的捕获。其中捕获缓存是一种循环缓存,用于存放从音频输入设备捕获来的音频数据。通过 IDirectSound Notify 接口,可以设置通知点,在捕获缓存指针到达通知点时立即发出一个事件,应用程序就可以取得声音数据。

(3) 视频音频编码解码处理

视频编码解码采用 MPEG-4 标准的 Xvid Video 编码器来压缩/解压的,XviD 是针对 Windows 用户设计的,对于 Mpeg-4 视频编码内核来说,XviD 被认为是当前最快的 Mpeg-4 编码器。XviD 在实现编码器初始化、编码和销毁功能时都通过 Int xvid_encore(void * handle, int opt, void * param1, void * param2) 接口函数的调用来实现的,只是在实现不同功能时赋的参数不同。

音频的编码解码是采用了 Mp3 编码器的方式进行的,它是一种基于知觉的有损的音频编码方案,全称为 Mpeg—1Layer—3 音频文件。Mp3 提供了在 28kbit/s 码率下接近 CD 的音质,完全能够满足远程学习中语音交流的要求。

(4) 视频音频同步处理

现场直播中除了视频音频的捕捉、编码、解码、播放外,还需要克服视频和音频的同步问题,本系统中是采用时间戳和缓冲器控制来实现的。可通过在单一通道中加盖全局的时间戳 TimeStamp 来实现双通道的功能;需要周期性地检测缓存数据量,若缓存量超过预定的警界线就认为存在不同步的现象,其解决方法有:在接收端,通过加快或放慢接收的时钟频率,也可删去或复制缓存器中的某些数据单元,使缓存器中的数据量逐渐恢复到警界线之内;同时还可通过网络发送端反馈再同步的控制信息来实现。视频和音频的同步具体可分为以下几个阶段:①视频和音频在捕获时分别在数据头部加统一的时间戳来保持同步;②对数据压缩处理在发送前,进行媒体内同步和视频音频间的合成同步处理;③通过 MRTP 协议传输到达学生端时,做一次接收同步处理;④解码后进行视频音频同步播放。

4.3 网络发送/接收处理技术

在网络带宽比较低的情况下(如十几 Kbps),数据丢帧现象比较严重,这对于图像质量有很大的影响,所以这里采用拆帧(拆成 1400 字节,要小于 MTU)以后再发送的方法。具体来说就是用 MRTP 协议传输压缩后的屏幕图像、视频流等数据时,在教师端需要先把每个数据帧分成 1400bytes 的块后用 MRTP 协议传输,学生端收到数据时需要先将这些数据块组合成帧,再进行解压缩处理,最后在学生机屏幕上显示出来,如图 5 所示。拆帧策略有效地避免了低带宽下的数据丢帧现象,提高传送数据的质量。

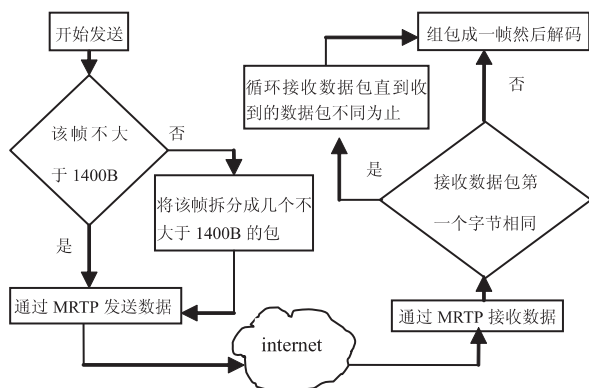


图 5 通过 MRTP 协议发送接收数据程序流程图

4.4 其他模块的处理

文件管理模块较易实现,在设计时,教师端文件管理/下载可通过应用层组播进行传输,而对学生端的文件上传可通过点对点方式进行上传。消息广播/接收消息直接实现较简单,也可以和屏幕图像一起传输,这时就需要在数据包发送时将文字和屏幕图像数据混合打包后再传输。

5 结论

基于应用层组播技术的远程授课直播系统在完成一个传输流量约为 300kbs 的视频直播应用测试中,有 40 多个节点参与,节点间的数据转发延迟在 150 - 200ms 之内,整体时延小于 5 秒,传输的视频画面具有良好的效果。说明该系统利用应用层组播技术保证了数据传输质量,节省了网络带宽,提高了数据传输效率,具有良好的实时性和同步性,较适合于地域分散的远程同步者实时学习的需要。另外,该系统能自适合调整,且不需要额外的硬件设备,便于扩展,将为远程实时教学带来便利。

参考文献

- 1 Zhi Li and Yongjoo Shin . Survey of Overlay Multicast Technology. 2nd June 2002.
- 2 姚玮,张卫. 基于应用层组播的实时视频传送协议关键技术研究. 计算机系统应用, 2007, 16(8): 81 - 84.
- 3 陈旭孟,张卫. 一个面向实时传输的应用层组播协议的设计与实现. 计算机应用与软件, 2008.
- 4 戴博,周杨,龚涛,李仪. Visual C++ 程序员成长攻略. 北京:中国水利水电出版社, 2007: 344 - 350.
- 5 陆其明. DirectShow 实务精选. 北京:科学出版社, 2004: 1 - 2.