

## 摘 要

现代远程教学系统是现代网络教育的重要组成部分，在教育领域中占据了非常重要的地位。现代远程教学系统多采用实时交互方式，教师与学生在不同的空间自由的进行交流。两者间的交流包括语音、图像及各种教学资源，例如，Word, PowerPoint, Html 等。我国目前对远程实时教学系统已有一些系统实现，国内远程实时教学领域中很少借鉴国际上通用的多媒体通信协议，系统不能随通信标准的发展而扩展，也难以与国际兼容。而且这些系统多使用虚拟打印或捕捉桌面流等技术将各种资源转化为视频数据传输，对网络带宽消耗过大，传输效果也不理想。

基于以上问题，本文对 H.323 协议进行一定程度的探讨，研究了 H.323 协议与远程实时教学系统结合的可行性，并给出了 H.323 协议在远程实时教学系统中的应用实例。本文亦针对远程实时教学系统中语音、图像及教学资源不同的时延特性，创造性地提出了一种降低传输带宽要求的两层框架结构，并进一步阐述了终端及 H.323 服务器(MCU)的详细设计方案。

通过对 H.323 协议在远程实时教学系统中的应用研究，本文给出了两者融合的实现实例，提出了远程实时教学系统与通信标准结合的新思路，为远程教学系统与国际接轨做出了尝试和贡献。本文提出的两层框架结构对目前的远程教学系统框架进行了改进，对远程实时教学系统的设计提供了参考和借鉴。

关键词：网络教育；远程教学；H.323；多点控制单元

## **Abstract**

Modern E-Learning system is an important component of modern network education, which also occupies a very important position in the field of education. Teachers and students in different areas could implement the free exchanges of the information by the method of Real-time interactivity of the modern E-Learning system. Exchanges between the two include audio, video and other variety of teaching resources, such as Word, PowerPoint and HTML etc. Most of the domestic E-Learning systems seldom draw on the common multimedia communication protocols, which could not be extended with the development of standards for telecommunication and are difficult to achieve the international compatibility. Moreover, these systems adopt virtual-print or desktop flow capture technology in order to transform various resources into video data for transmission. The transmission effect is not satisfied due to the excessive consumption of network bandwidth.

Based on the above problems and the research on the H.323 protocol, this thesis focuses on the feasibility of the combination of E-Learning system and H.323 protocol. It also gives an instance of H.323 protocol using in real-time E-Learning system. A creative schema of a two-tier framework to reduce the bandwidth requirement is introduced, as well as the detailed design of the terminal and H.323 server (MCU), on the basis of the different delay characteristics of audio, video and other teaching resources of the E-Learning system.

Through the study of the H.323 protocol using in real-time E-Learning system, this thesis gives the instance of the realization of fusion of the H.323 protocol and the real-time E-Learning system as well as the new ideas for the combination of telecommunication standards and E-Learning systems, which makes a contribution for the E-Learning system in line with international implementation. This thesis also presents the framework two-tier structure which improves the current framework of the E-Learning systems, providing reference for the design of E-Learning system.

**Key Words:** E-Learning; Distance Education; H.323; Multipoint Control Unit

## 缩略语

<b>B-ISDN</b> – Broadband Integrated Service Digital Network	窄带综合业务数字网
<b>COM</b> – Component Object Model	组件对象模型
<b>GK</b> – Gatekeepers	网守
<b>GSTN</b> – General Switched Telephone Network	通用交换电话网
<b>GW</b> – Gateways	网关
<b>IO</b> – Input/Output	输入/输出
<b>IP</b> – Internet Protocol	国际互联网络通讯协定
<b>LAN</b> – Local Area Network	局域网
<b>MC</b> – Multipoint Controller	多点控制器
<b>MCU</b> – Multipoint Control Unit	多点控制单元
<b>MP</b> – Multipoint Processor	多点处理器
<b>N-ISDN</b> – Narrow Integrated Service Digital Network	宽窄带综合业务数字网
<b>PBN</b> – Packet Based Networks	分组网络
<b>RAS</b> – Registration, Admission and Status	注册管理状态
<b>RTP</b> – Real Time Transport Protocol	实时传输协议
<b>RTCP</b> – Real Time Transport Control Protocol	实时传输控制协议
<b>SCN</b> – Switched Circuit Network	电路交换网
<b>TCP</b> – Transmission Control Protocol	传输控制协定
<b>UDP</b> – User Datagram Protocol	用户数据报协议
<b>WDM</b> – Win32 Driver Model	Win32 驱动程序模型

## 独创性声明

本人声明所呈交的学位论文是我个人在导师的指导下进行的研究工作及取得的研究成果。尽我所知，除文中已标明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

## 学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密，在\_\_\_\_\_年解密后适用本授权书。

本论文属于

不保密。

（请在以上方框内打“ ”）

学位论文作者签名：

指导教师签名：

日期： 年 月 日

日期： 年 月 日

# 1 绪 论

## 1.1 课题研究的背景和目的

据 2005 年，中国互联网络信息中心（CNNIC）<sup>[i]</sup>统计，截止到 2005 年 6 月 30 日，我国的上网用户总人数为 10300 万人，同半年前的调查相比，我国上网用户总人数半年增加了 900 万人，增长率为 9.6%，和上年同期相比增长 18.4%。同 1997 年 10 月第一次调查结果 62 万网民人数相比，现在的网民人数已是当初的 166.0 倍。我国的上网计算机总数已达 4560 万台，同半年前的调查结果相比，我国的上网计算机总数半年增加了 400 万台，增长率为 9.6%，和上年同期相比增长 25.6%。

互联网良好的发展势头及网络技术的迅速成熟为教育模式的革新带来了一次新的机遇，现代网络教育结合现代网络技术、多媒体技术与传统教育学的优势，改变传统的面对面的教学模式，集中各种教学资源，提供更丰富更直观的教学内容，为学习者提供更多的学习机会和更好的学习环境，也方便教师根据教学目的及学生的具体情况选择适当的教育资源。网络教育的最大优势就是实现了教育资源的共享利用。网络教育（E-Learning）为人们提供了一种全新的学习方式。许多国家都高度重视网络教育，努力扶持和发展这一全新的教育方式，为推进网络教育的普及与发展投入大量资金，不断扩大网络教育的普及化程度。

随着信息技术的发展，网络教育呈现了越来越多的形式<sup>[ii]</sup>，现代远程教学系统不仅实现了教育资源的共享利用，同时打破了时空及地域限制，延伸至全国乃至世界，实现教育资源的共享和优化。本课题正是在国际网络教育飞速发展的情况下提出的，本课题的来源是湖北省发展与改革委员会的“标准化实时教学应用平台”项目。课题研究的背景是在跨地域的环境下，利用网络实现远程教学过程，使更多的人能够通过本系统方便的进行学习。本系统致力于在获得良好教学质量的前提下大量节省教学资源，不仅在教学过程中节省了大量人力物力，还实现了教学资源的再利用<sup>[iii]</sup>。

由于国际上的高度重视，远程教学应用渐渐广泛起来，而大量的应用对远程教学系统提出了更高的要求。近年来远程教学系统逐渐由广播、点播式教学向实时教学方式转变。实时教学方式以其身临其境的教学感受深受用户欢迎。在这种教学方式下，教师和学生可以通过网络实时进行交流，教师可询问学生的学习情况，随时控制教学进度；学生可向教师提出疑问，表述观点。这种教学方式大大提高了教学质量，具有广播、点播式教学无法比拟的优势。但这种教学方式对远程教学系统提

出了极高的要求，如何在有限带宽下实现教师和学生的视音频互通，如何满足用户对多种教学资源的需求，如何减轻某些配置较低客户端的负载...这都是需要攻克的难点<sup>[iv]</sup>。

本课题是基于实时教学方式设计的，采用 H.323 视频会议通信体系对网络实时教学进行实现。H.323 体系基于通信网分层的特点，发展比较成熟。未来第三代移动通信网中的网关都将支持 H.323 体系多媒体的通信，随着通信技术的发展本课题将会有更广阔的应用空间<sup>[v]</sup>。

## 1.2 国内外远程实时教学系统的发展现状

由于网络教育的飞速发展，近年来远程教学系统的发展相当迅速。在国外，实时远程教学平台作为现代远程教育的一种应用手段，已经发展到比较成熟的阶段。而且，实时远程教学平台正在向着海量数据存储、随时随地接入、复合媒体、高智能化、高自动化和虚拟现实等方向发展，此外，对于实时教学中产生的网络课件的版权保护以及共享需求越来越迫切。

在美国，以网络为基础的实时远程教学平台，得到了很大的发展。到 2004 年，有大约三分之一的州建立了实时远程教学平台，如西弗吉尼实时远程教学平台(WVVS)、阿拉斯加的实时远程教学平台(SeeUonline)、得克萨斯的奥斯丁中学计划(AHSP)、休斯顿独立学区(HISD)等<sup>[vi]</sup>。

国外众多知名公司也看到了网络远程教育市场的发展前景，相关产品公司发展迅速，形成了具有较大规模的产业，如 WebCT 公司的教育信息化平台 WebCT 和相关产品，年销售额逾 100 亿美元。

由于国内网络教育的起步较国外晚，远程教学系统的发展也相对滞后，但随着近年来网络教育的飞速发展，国内已有部分 IT 企业涉足网络教育产品的研发，如台湾讯联公司的串流大师(Streamsuthor)、以太智和公司的课程制作系统、江苏科健的课程制作工具、北京傲姿的课程制作工具、网梯的课程开发工具等<sup>[vii]</sup>。笔者对目前国内远程教学系统的发展状况进行了调研。调研结果表明，近年来国内远程教学系统虽与国外远程教学系统相比仍有较大的差距，但较于自身的发展已经取得了长足的进步，不到五年的时间，远程教学系统已经从广播式教学系统发展为实时交互式教学系统，系统的功能也越来越完善。现代远程实时教学系统整合了教师教学现场的视频、音频、结合同步教学课件命令以及教师板书的实时直播，形成完善的教学资源流，利用计算机网络传送到远端学生的电脑终端，在学生电脑上实时播放。在实时直播的同时，学生可实时与教师进行视频、音频等多种媒体的交互，本地也可

记录教师所有的教学指令，形成教学资源。实时交互过程不仅包括了基本的教师图像、语音，还包括了与教师动作同步展示的教学资源，如 PowerPoint 文档、Word 文档、HTML 文档及图片等资料<sup>[viii]</sup>。今年某系统甚至推出了课堂作业、课后学习情况统计等功能，使远程教学系统越来越贴近真实的课堂，功能越来越丰富多彩。它为了解决我国目前教育资源匮乏、师资严重不足等问题提供了新的有效途径，对于促进我国教育事业的发展，带动产业发展具有重要的意义。

### 1.3 目前远程教学实时系统存在的问题

尽管远程教学系统质量已取得大幅度的提高，但仍然存在比较多的问题有待进一步的研究解决<sup>[ix]</sup>，本文也尝试解决了其中某些问题。

现代远程教学系统越来越多的侧重于功能性的发展，但在性能方面的研究投入相对较少。在功能完善的同时，始终存在着教师图像、语音不流利，教学资源不清晰等问题<sup>[x]</sup>，其原因有两点：

1. 现代远程实时教学系统多采用虚拟打印、捕获桌面流等方式将教学资源的文档数据转换成视频数据进行传输，若网络带宽、传输速度不理想或终端电脑处理能力不佳，即会造成以上问题。但在现实的应用中，教师的教学资源往往长时间不切换，在展示资源不变的情况下仍然传输大量的资源视频数据往往是不必要的，不仅影响了用户使用的效果，还占用了大量的网络资源。本文根据教学资源在实际中的应用特性，创造性的提出了两层框架结构的设计方案，尝试解决了以上的问题。

2. 现代远程实时教学系统在开发时对于音视频编解码的研究投入较少。音视频是远程实时教学系统的核心，因此音视频质量的提高应该是远程实时教学系统的根本追求。而提高音视频质量的根本途径是提高音视频编解码的质量。由于本文内容不涉及音视频编解码的问题，因此不对此做更进一步的探讨。

另外由于现代远程教学系统采用的上述实现方法，导致课件包必须包含教学资源部分的视频，以致课件包包含了太多不必要的信息，占用了大量的空间。对此问题本文提出的两层框架结构也可以比较巧妙的解决。

### 1.4 本文的内容与组织

本文在对远程实时教学系统的论述中，详细描述了 H.323 协议的系统架构、基本通信框架及协议通信流程，着重叙述两层框架结构的整体设计，并详细讨论了基于 H.323 协议的服务器和终端的流程设计及详细设计。本文的内容组织从对远程实

时教学系统的研究出发，从基于 H.323 协议的通信技术与实时教学系统结合的角度进行叙述，给出用 H.323 架构实现实时教学系统的系统框架，并详述了其中的自定义信令及音视频处理模块的设计与实现。

本文共分七章，内容安排如下：

第一章，是本文的总体综述。这一章首先描述了课题研究的背景及课题研究的目的，然后介绍国内外远程教学系统的发展状况，最后总结目前国内远程实时教学系统中存在的问题，为系统未来的开发方向提供了参考。

第二章，分析了 H.323 协议，是整个系统的实现基础，也是整篇论文的技术基础。这一章首先对 H.323 协议做了概括性的介绍，然后介绍了 H.323 协议的系统架构，接着分析了 H.323 协议的通信协议框架和通信流程，最后探讨了 H.323 协议在远程实时教学系统中应用的可行性。本章为后面的内容做出了技术铺垫。

第三章，叙述了远程实时教学系统的总体框架，从总体上介绍了系统的构架及各部分的实现技术。本章首先给出了系统的需求分析和系统两层结构框架，然后分别从资源层及 H.323 层两个层次上介绍了系统的设计方案，并给出了简要的实现技术介绍。

第四章，对整个系统中 H.323 服务器部分进行了详细介绍。这一章首先介绍了 H.323 服务器系统的处理流程，然后着重介绍了信令处理功能及音视频处理功能这两块功能的设计及技术实现。

第五章，对整个系统中终端系统部分进行了详细介绍。这一章首先介绍了终端系统的处理流程，然后着重介绍了信令处理功能及音视频处理功能这两块功能的设计及技术实现。最后介绍了终端资源层与 H.323 层的层间接口。

第六章，对全文设计与实现部分作了总结。这一章首先介绍给出了整个系统的界面效果图，并根据界面效果图对系统已实现的功能作了介绍，最后介绍了系统的测试情况和测试结论分析，给出了系统在现实中的应用实例。

第七章，总结了本文的研究成果与贡献，并简要论述了远程实时教学系统需要尚待完善的问题及未来的研究方向。



## 2 H.323 协议体系结构分析

H.323 协议是 ITU-T (国际电信联盟) 第 16 研究组于 1996 年通过的, 该小组于 2001 年 2 月又通过了 H.323 协议的第 4 版 H.323v4。它是 ITU-T 为了在无服务质量保证的 Packet Based Networks (PBN) 上多媒体通信系统和设备进行通信的协议集, 包括点到点通信和多点通信。H.323 协议对呼叫控制、多媒体管理、带宽管理以及 LAN 和其他网络的接口都进行了详细的规范说明<sup>[xi]</sup>。

H.323 协议为通过基于 IP 的网络进行音频、视频、和数据通信的应用提供了一个可遵循的国际标准。采用 H.323 协议, 各个不同厂商的多媒体产品和应用可以进行相互操作。由于遵循 H.323 协议的终端提供的是点对点会议或多点会议中音频(以及可选视频及数据)的通信能力, 故 H.323 协议俗称视频会议协议。

H.323 现阶段在业界最重要的应用, 是作为 IP 电话系统的技术规范之一, 是各厂商 IP 电话系统设备互通的技术依据, 主要应用于电信领域内。目前在网上部署的 IP 电话系统只涉及 H.323 标准关于话音通信基本功能的一个子集。所以 H.323 协议在集音频、视频和数据通信与一体的视频会议领域的应用研究方面还有很大的发展空间。

### 2.1 H.323 协议简介

视频会议系统是一种支持人们远距离进行实时信息交流、开展协同工作的应用系统, 不仅能实时传输视频与音频信息, 使协作成员可以远距离进行直观、真实的视音频交流, 还可利用多媒体技术的支持, 帮助使用者对工作中各种信息进行处理, 如共享数据、共享应用。会议系统有很多类型, 虽然基于 H.320 技术体系下的 IP 可视电话在 1992 年就已经面世, 但是, 受网络环境特别是带宽影响, 这些设备仅仅在商用市场上有了应用。近年来, 互联网的飞速发展, 许多国外企业重新开始关注家用 IP 可视电话市场<sup>[xii]</sup>。

H.323 是国际电信联盟电信标准部(ITU-T)制订的在各种网络上提供多媒体通信的系列协议 H.32x 的一部分。H.32x 包括 H.320、H.321、H.322、H.323、H.324。其中, H.320 是在窄带综合数字业务网(Narrow-band Integrated Service Digital Network, N-ISDN)上进行多媒体通信的标准; H.321 是在宽带综合数字业务网(Broad-band Integrated Service Digital Network, B-ISDN)上进行多媒体通信的标准; H.322 是在有服务质量保证的局域网(Local Area Network, LAN)上进行多媒体通信

的标准；H.324 是在通用交换电话网（General Switched Telephone Network, GSTN）和无线网络上进行多媒体通信的标准<sup>[xiii]</sup>。H.323 是在原 H.320 协议的基础上发展起来的，完全兼容原 H.320 系统，用来在基于包交换的网络上传输音频、视频和数据。目前 H.323 协议被普遍认为是在分组网上支持语音、图像和数据业务最成熟的协议。随着 IP 网的迅速发展，H.323 视频会议系统主要应用于 IP 网中，和它协作应用的基础是 IP 通信平台，主要由 ISO/OSI 模型下四层（物理层、数据链路层、网络层和传输层）构成，对上层形成一致的接口。采用 H.323 协议，如图 2-1 所示，各个不同厂商的多媒体产品和应用可以互操作，而不必考虑兼容性问题<sup>[xiv]</sup>。

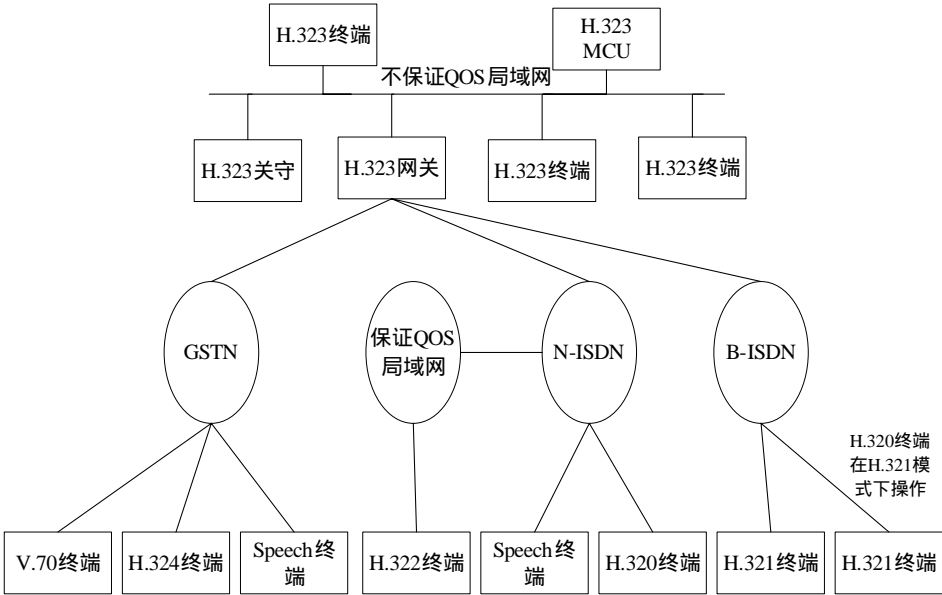


图 2-1 H.323 终端互通性示意图<sup>[xv]</sup>

H.323 协议描述了 H.323 系统的组成部分，H.323 系统包括终端(Terminal)，网关(Gateways, GW)，关守(Gatekeepers, GK)，多点控制器(Multipoint Controller, MC)，多点处理器(Multipoint Processors, MP)以及多点控制单元(Multipoint Control Units, MCU)。

H.323 系统由 H.323 端点和 H.323 实体组成。H.323 端点包括 H.323 终端、网关和 MCU，端点可以呼叫和被呼叫，它发起和终止信息流。H.323 实体没有呼叫能力，例如关守。

### 2.1.1 H.323 系统中的终端

终端是 H.323 网络中可以与其他终端、网关或 MCU 进行实时双向通信的端点。

通信内容包括控制、指示、音频、视频和数据。终端可以仅提供语音、语音和数据、语音和视频、或语音、数据和视频。可见，在 H.323 协议中语音是默认必须提供的，而视频和数据是可选传输的。终端在 H.323 网络中是用户直接面对的部分，有着重要的地位。

H.323 终端的例子如图 2-2 所示，图中显示了用户设备接口、音频编解码器、视频编解码器、系统控制部分、H.225 层以及本地网络接口。所有终端必须具有用户设备接口、音频编解码器、系统控制部分、H.225 层以及本地网络接口，而视频编解码器是可选设备。

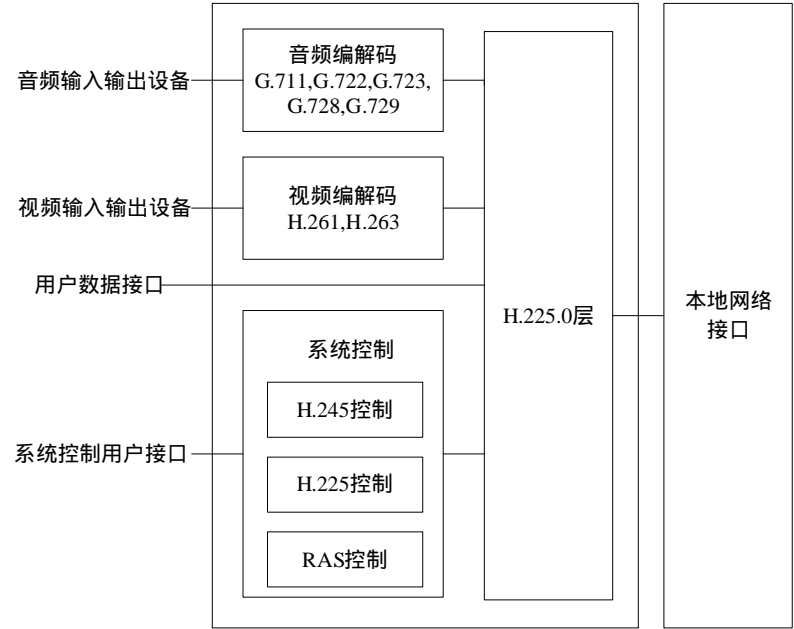


图 2-2 系统终端结构框图

用户设备接口包括音频输入输出设备、视频输入输出设备、用户数据接口及系统控制用户接口。这些设备接口不在 H.323 协议定义范围内。

音频编解码器对麦克风等音频输入设备的音频信号进行编码发送，对接受到的音频码进行解码。由于音频通信能力是 H.323 协议默认具备的，故所有 H.323 终端都必须具有一个音频编解码器。H.323 协议定义了一个默认音频格式，所有 H.323 终端都必须能根据 G.711 协议<sup>[xvi]</sup>进行语音的编码和解码。H.323 协议也定义了对诸如 G.722<sup>[xvii]</sup>、G.723、G.728<sup>[xviii]</sup>、G.729<sup>[xix]</sup>等协议的支持，但要通过 H.245 协议进行能力集协商来确定。

视频编解码器对来自视频源的视频数据进行编码发送，对接受到的视频码进行解码并输出到视频显示器。视频编解码器对于 H.323 系统是可选的。如果提供了视

频能力，则所有终端必须能够根据 H.261 协议进行编解码。H.323 协议还定义了对 H.263<sup>[xx]</sup>协议的支持，支持 H.263 的终端必须支持 H.263QCIF。通过 H.245 的协商，也可以使用其它视频编解码格式。

数据信道传输用户数据，如电子白板、静态图像传输、文件交互等，一般使用 T.120 协议实现。本系统不涉及到 T.120 的内容，因此不做详细介绍。

系统控制单元为 H.323 终端提供呼叫控制、能力集交换、命令和指示的信令以及消息。它由 H.245 协议和 H.225 协议共同完成，2.2.3 节中将对此作详细的介绍。

H.225.0 层对发送的视频、音频、数据和控制流进行格式化，形成特定格式的消息输出到本地网络；从网络接口接收的特定格式消息中提取视频、音频、数据和控制流。同时，它对每一种媒体类型完成逻辑成帧、顺序编号、差错监测和纠正等功能。H.323 系统中的每种数据流都要经过 H.225.0 层处理，它是 H.323 终端系统中的基础。

### 2.1.2 H.323 系统中的 MC、MP 及 MCU

MC（多点控制器）是 H.323 网络中的一个实体，为三个或多个终端参加一个多点会议提供控制功能。它可以在一个点对点会议中连接两个终端，随后扩展为多点会议。MC 与所有终端进行能力协商以达到共同的通信能力。它也可以控制会议资源。

MP（多点处理单元）是 H.323 网络中的一个实体，它提供在一个多点会议中对音频、视频和数据流的集中处理。MP 在 MC 的控制下提供对媒体流的混合、交换或其他处理。MP 可以处理单个或多个媒体流，取决于会议类型。

MCU（多点控制单元）是 H.323 网络中的一个端点，它为三个或更多终端及网关参加一个多点会议服务。它可以连接两个终端构成点对点的会议，随后再扩展成多点会议。MCU 由两部分构成：必须的多点控制器 MC 和可选的多点处理器 MP。本系统 MCU 服务器由 MC 和 MP 两大模块配合完成 MCU 的集中控制和处理功能。

如上文所述，MC 和 MP 分别具有不同的功能，它们在 H.323 系统中的分布位置也很灵活。如图 2-3 所示，终端可以具有 MC，也可以没有 MC。因为终端可以具有控制功能，也可以没有。终端中的 MC 是不可呼叫的，在终端中 MC 与终端的 H.245 控制功能没有什么差别。终端始终不具备多点处理功能，故终端不具有 MP。由于 MCU 是多点控制单元，它具有控制功能，因此必须具有 MC，MCU 是可呼叫的，其中的 MC 处理来自所有端点的 H.245 控制信道。若 MCU 也负责资源的混合、

交换或其他处理，则也可具有 MP。一般的 MCU 同时具备 MC 和 MP。关守和网关可以具有 MC、MC 和 MP、也可以不具有这两部分，取决于关守和网关的设计功能。关守内的 MC 同样是不可呼叫的，但关守内的 MCU 可具有独立的 MCU 功能，故关守内存在的 MCU 是可呼叫的。网关可作为终端或 MCU 运行，作为终端时网关可包括 MC。网关和关守都可以拥有 MCU，但它们是功能独立的，并且是共址的。

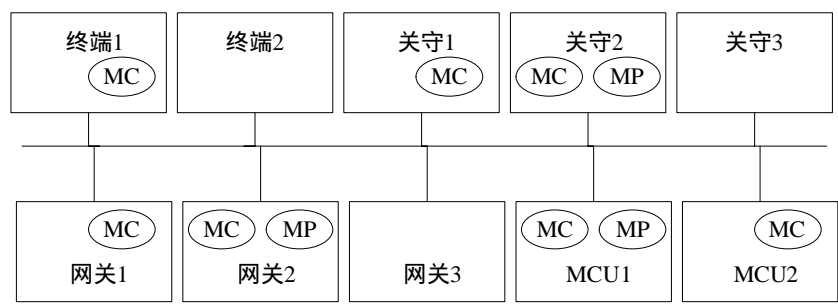


图 2-3 H.323 系统中 MC 和 MP 的可能位置

### 2.1.3 H.323 中的网关和关守

H.323 网关主要提供分组网 H.323 终端及电路交换网(Switched Circuit Network, SCN)SCN 终端间的通信能力，如图 2-1 所示。网关必须提供传输格式（如 H.225.0 与 H.221）和通信过程（如 H.245 与 H.242）间的适当转换；必须能够在分组网络一侧和 SCN 一侧执行呼叫的建立和清除。网关还可以通过 SCN 连接到其它网关，以提供在不同网络上的 H.323 终端间的通信<sup>[xxi]</sup>。一般而言，网关的作用是透明的将分组网络端点反映为 SCN 端点，因此若 H.323 端点可直接与另一 H.323 端点通信则不需要网关。当然，如图 2-3 所示，H.323 网关也可以提供视频、音频和数据格式的转换，它既可以作为终端使用，也可以作为 MCU 使用。由于本系统不涉及网关的设计和实现，因此不再详述。

在 H.323 系统中，关守是可选择的，它为 H.323 端点提供呼叫控制服务。多个关守可以共存，且在逻辑上它独立于端点<sup>[xxii]</sup>。在一个区域任何时刻有且仅有一个关守。关守一旦存在，则必须提供以下地址转换、接入控制、带宽控制、区域管理等功能<sup>[xxiii]</sup>。唯一注册到关守的终端、MCU 和网关等设备提供上述功能。关守还可以提供呼叫管理、目录服务等其他附加功能。关守不需要处理 H.245 信令，本质上他只在终端之间或终端与 MC 之间传输信令<sup>[xxiv]</sup>。由于网关地址转换的传输需要，包含网关的网络也应该包含一个关守。现阶段本系统尚未涉及到关守，因此对关守不作详细介绍，但这仍是本系统未来的改进方向。

## 2.2 H.323 协议框架流程

### 2.2.1 H.323 多点会议架构

现阶段 H.323 协议的应用主要集中在多点会议。多点会议分为集中多点会议、分散多点会议及混合多点会议三种。它们均需要 MCU 进行多点控制。

#### 1. 集中式多点会议

集中多点会议是比较基础的会议模式，所有 H.323 端点都必须具有集中多点能力。在这种工作模式下，MCU 担负比较重的责任。如图 2-4 所示，集中式会议模式下，终端之间不直接进行通信，所有的控制信令和媒体流都必须通过 MCU 进行处理和转发。

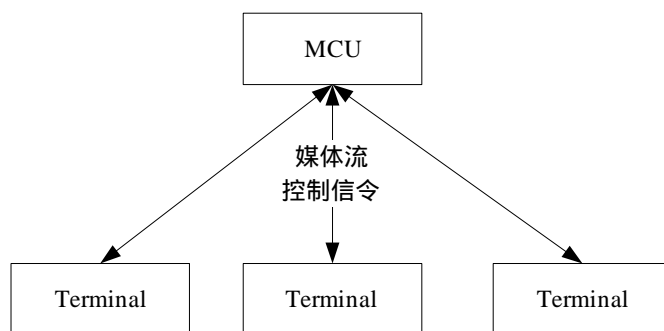


图 2-4 H.323 集中式多点会议示意图

集中式多点会议模式下，MCU 必须同时具备 MC 和 MP。终端以点对点方式在控制信道上与 MCU 中的 MC 通信，以这种方式实现 H.245 多点控制功能；在音频、视频和数据信道上与 MCU 中的 MP 通信。MP 负责视频的混合或切换、音频的混合及 T.120 等数据的分发。MP 将从各终端接收到的视频、音频和数据流处理后分发回各端点。此模式下，MCU 可以具有不同的音频、视频及数据格式间的转换能力，允许各端点以不同的通信模式加入会议。

集中式多点会议模式下，终端不需要分散多点能力，各终端均处在点对点传输模式下，处理相对简单，但由于所有的音频、视频及数据流均需要 MCU 处理及中转，故 MCU 负载较重。这种模式在实际中的应用很广。

#### 2. 分散式多点会议

分散式会议中各终端必须具备分散多点能力，如图 2-5 所示，终端仍以点对点方式在控制信道上与 MCU 中的 MC 通信，实现 H.245 的多点控制能力，但与 MCU 中的 MP 间的通信成为可选项。

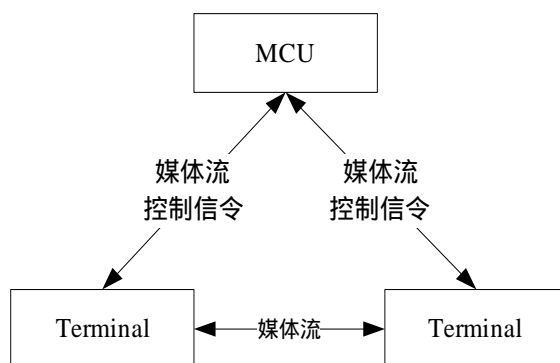


图 2-5 H.323 分散式会议示意图

分散式多点会议模式下 MCU 必须具备 MC，可选的具备 MP。终端与终端之间必须能够多播媒体流。MCU 中的 MC 可以控制哪一个或哪几个端点正在多播音频和视频。终端接收多播的视频信道，选择一个或多个可用的信道提供给用户。终端接收多播音频信道，执行音频混合功能，并向用户提供一个合成的音频信号。此模式下，终端应具备 MP 以混合音频、视频。

分散式多点会议模式下，终端必须具有混合音频、视频的能力，比集中多点式会议负载重。对于 MCU 而言，可以不再集中处理音频和视频，负载相对较轻。分散式多点会议模式在终端较多的大型会议系统中应用较广，但网络带宽消耗过大。

### 3. 混合式多点会议

混合式多点会议同时具备了集中式多点会议和分散式多点会议的特点。如图 2-6 在某些终端中使用了集中式多点会议模式，在另一些终端间使用了分散式多点会议模式。混合式多点会议还可以针对音频、视频分别采用集中和分散模式。设计者可以根据各终端不同的特点和需求选择会议模式，设计非常灵活。

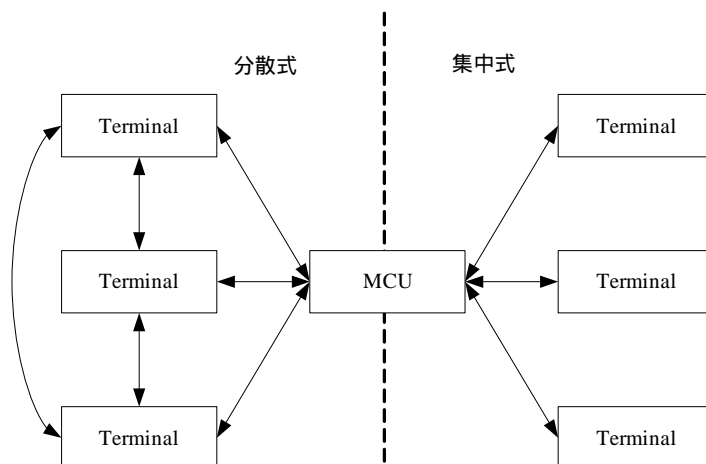


图 2-6 H.323 混合会议示意图

混合式多点会议模式下，设计者可以灵活配置多点会议网络，也可以根据网络情况分别为音频和视频设置不同的会议模式。这种模式在设计和实现上比较复杂，但应用和组网灵活多变，是比较好的组网方式。

2.2.2 H.323 通信协议框架

H.323 协议中包括 H.225, H.245, T.120 等多种通信协议，尤其是 H.225 及 H.245 协议在会议初始化过程中扮演着极其重要的角色。

H.323 协议构架在 IP 协议之上，如图 2-7 所示，分别使用 TCP 和 UCP 协议传输不同性质的协议数据。H.225、H.245 协议负责会议控制的部分，T.120 协议负责会议数据的传输，这些都需要可靠的传输保证，因此 H.245、T.120 及部分 H.225 协议都是基于可靠的 TCP 协议。对于音频和视频及 H.225 的 RAS 部分更倾向于追求传输速度，因此不需要可靠的传输保证，故基于不可靠但速度更快的 UDP 协议。

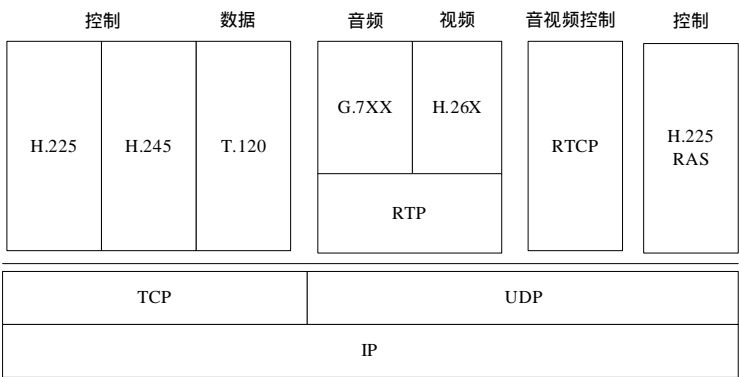


图 2-7 H.323 协议栈示意图

H.323 系统中的媒体流主要是音频流和视频流，H.323 支持 G.711、G.722、G.723、G.728、G.729 等音频协议，支持 H.261 和 H.263 两种视频协议。它们都是基于 RTP 协议，并使用 RTCP 协议控制媒体流。H.225 协议是 H.323 协议簇中重要的组成部分，它的主要功能是在呼叫开始之前，在端点之间建立呼叫联系，同时建立 H.245 控制信道。H.225 协议分为 RAS 信令和 Q.931 协议两部分<sup>[xxv]</sup>。

H.245 协议使用 H.245 控制信道来传输端到端的管理 H.323 实体的控制消息，包括能力交换、打开和关闭逻辑信道、模式优选请求、流控制消息以及通用命令和指示。

H.323 协议描述了三种不同消息类型的使用：RAS、Q.931、H.245。

RAS 信令在终端和关守之间完成注册(Registration)、接入认证(Admission)和状



态(Status)等通信的信令，故合称 RAS。RAS 信令由 8 种信令组成：关守搜索消息；接入认证、地址解析和修改密码消息；关守注册消息；呼叫终止消息；状态查询消息；资源报告消息；带宽管理信令；注销消息。

Q.931 信令的控制对象是呼叫和连接，H.225.0 借鉴了 Q.931 的消息结构和控制过程设计其呼叫信令协议，它由以下消息组成：

1. Setup 呼叫建立消息。该消息由一个打算呼叫的 H.323 实体发送，用来指示它要建立到被叫的连接。
2. Setup Acknowledge 建立证实消息。该消息由一个 H.323 实体发送，表示连接的建立。
3. Alerting 呼叫提示。该消息由被叫用户来发送，用以指示被叫已在振铃。
4. Call Proceeding 呼叫进行。该消息由被叫用户发送，表示被请求的呼叫建立已被发起并且不再有呼叫建立信息要被接收。
5. Connect 连接消息。该消息由被叫实体发送给主叫实体，以指示被叫实体接收该呼叫。
6. Connect Acknowledge 接收证实消息。该消息可以不发送。
7. Information 通知消息。发送此消息用来提供附加的信息，它可被用来为呼叫建立提供消息或与呼叫有关的其他各类消息。
8. Process 进程消息。该消息可由端点在连接消息之前发送。
9. Release Complete 释放完成消息。该消息由一个终端来发送，表示当可靠的呼叫信道被打开时便释放该呼叫。

H.245 消息分为四类：请求、响应、命令和指示。请求和响应由协议项使用。请求消息需要接收端做特殊动作，包括一个即时响应。响应消息响应相应的请求。命令消息需要一个特别的动作，但不要求响应。指示消息只是通知，并不要求任何动作或响应<sup>[xxvi]</sup>。H.323 协议中强制要求的 H.245 消息主要有以下几种<sup>[xxvii]</sup>：主从确定消息、终端能力消息、逻辑信道信令消息、命令消息。

以上三种信令配合完成一个会议的建立和释放过程。

### 2.2.3 H.323 协议通信流程

H.323 系统分无关守模式和有关守模式两种。以下将简单介绍无关守模式下终端间的会议建立、释放流程。

#### 1. 无关守模式终端连接建立流程

在无关守模式下终端间建立连接可不必通过其它端点中转消息。图 2-8 是两端点直接建立连接的示意图。如图 2-8 所示，终端 1 向终端 2 发送 Setup 呼叫建立消息，终端 2 接受连接后向终端 1 发回 Call Proceeding 呼叫进行消息，表示终端 2 接受呼叫，随后终端 2 向终端 1 发回 Alerting 呼叫提示消息，该消息表示终端 2 已在振铃。终端 2 向终端 1 发出 Connect 连接消息，当终端 1 接收到终端 2 发出的 Connect 消息后，连接建立，终端间完成 Q.931 呼叫建立过程，打开 H.245 通道，开始 H.245 的协商过程。协商完成后便可以开始媒体流的交换，当会议结束时结束会议的一方 向另一方发出 Release Complete 消息释放所有会议资源。

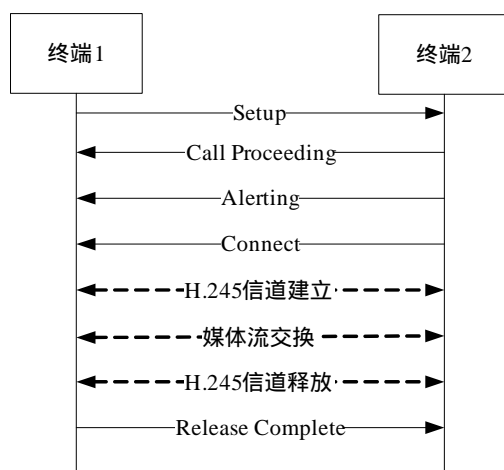


图 2-8 无关守模式终端连接示意图

终端间直接建立连接是最简单的点对点会议建立模式，在点对点会议中应用很广。

## 2. MCU 发起会议流程

在多点会议中一般都需要 MCU 的控制，会议可以由终端发起，也可以由 MCU 发起。终端加入会议的时刻不同，因此需要分别与 MCU 建立连接。MCU 发起会议的流程如图 2-9 所示，终端发起会议的流程与图 2-9 相似，可以由此图推知。

MCU 首先向终端 1 发起会议，流程如下：MCU 向终端 1 发出 Setup 消息，该消息携带会议 CID 标识该会议，终端 1 收到后发回 Connect 消息，该消息携带同样的 CID，标识该会议。会议连接建立后，终端 1 与 MCU 开始 H.245 协商过程。H.245 协商过程先进行能力集的交流，再进行主从确定，一般 MCU 的优先级比终端高，因此一般以 MCU 为主，终端为从<sup>[xxviii]</sup>。主从确定后建立好逻辑信道 H.245 的协商过程就完成了。会议初始化完成，终端 1 与 MCU 就可以开始媒体流的交换。

当 MCU 需要终端 2 加入该点对点会议，MCU 以同样的流程与终端 2 建立连接，则将该会议扩展为多点会议。建立多点会议后，终端 1 和终端 2 发向 MCU 的媒体

流由 MCU 中的 MP 混合处理后发回给终端 1 和终端 2。多个终端加入会议时以同样的流程与 MCU 建立连接，多点会议建立完成。

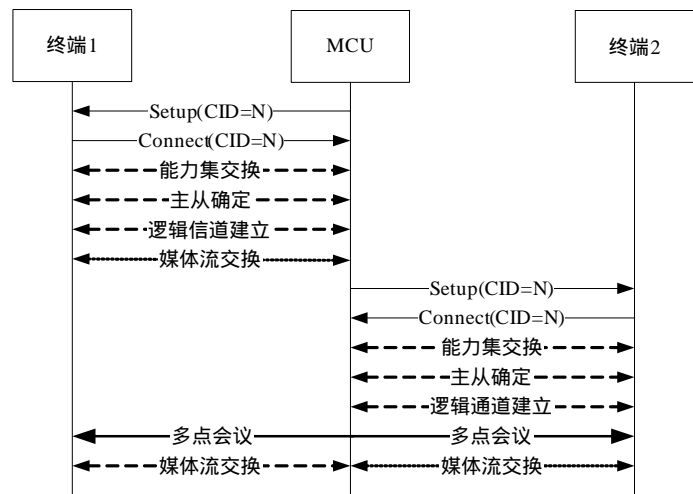


图 2-9 MCU 发起会议示意图

该流程是最简单的多点会议建立模式。本系统即使用此模式，不同的是本系统由终端发起连接，其它通信过程与图 2-9 相同。

### 2.3 H.323 协议在远程实时教学中的应用

如上文所述，H.323 协议规定了视频会议系统从会议初始化到会议结束的大部分通信细节。视频会议系统实现的是多个终端间音频、视频及会议数据的传输，这与远程实时教学系统的功能需求很相似。

远程实时教学系统需求如下：

1. 实时授课。远程实时教学系统属于实时授课系统，通过网络实时传送教师授课的图像和声音，学生在本机上看到的图像，听到的声音，都是通过网络实时获取，使听课者即使不在课堂的环境里，一样能够身临其境地跟随着教师的教学思路同步学习。在远程实时教学系统中一个课堂可以看作一个会议，这一点与视频会议很相似。但教师可以选择发言人，可以要求学生发言和停止发言，也可以切换发言；同时学生也可以申请发言，教师批准后即可发言。教师音视频必须始终呈现给学生，而学生音视频则取决于教师对学生发言的控制情况，正在发言学生音视频需呈现给所有参与者，而未发言的学生则不需发送自己的音视频。这一需求虽然与会议系统不尽相同，但可以通过服务器端的控制实现。
2. 文本交流。通过该功能，使教师和学生之间，学生与学生之间可进行文本方式的交流。在不占用宝贵的网络带宽的前提下，实现课堂交互的实时性，进一步

保证了教学的质量。文本交流可以通过 H.323 协议信令交互的方式传输，同样适用 H.323 协议。

3. 课件浏览。远程实时教学系统需可与各种常用办公软件协同工作，包括 Word、Power Point 等。教师可将 Word 文档、Power Point 文档、html 网页、Flash、媒体文件及各种格式的图片处理成课件，通过系统在授课讲述的同时进行播放，给学生最直接的知识输入和认识。

4. 网页共享。通过使用该功能，教师可向学生展示网页，丰富了教学内容；同时，学生也可使用该功能浏览网页，查找与课程有关的资料，使学习更加多样化。

5. 文件共享。通过使用该功能，教师上传教学资料，学生直接下载即可得到，学生即使身在远方，也可以收到老师的教学资料；同时，学生亦可以上传学习资料（作业等）供教师或其他学生下载，即使没有身在教室，也可以提交作业。该功能为教师与学生，学生与学生之间架起了沟通的桥梁。

以上五点即构成一个完备的远程实时教学系统<sup>[xxix]</sup>。如上文所述，实时授课与文本交互这两大基本功能都可使用 H.323 协议实现。而课件浏览、网页共享及文件共享这三个功能可以使用下文将介绍的两层框架结构，通过其他方式处理后使用 H.323 协议实现。

综上所述，H.323 协议与远程实时教学系统的需求结合非常紧密，因此 H.323 协议非常适合在远程实时教学系统中应用<sup>[xxx]</sup>，对于 H.323 协议在远程教学系统中的研究是非常有价值的。

## 2.4 本章小结

本章简要介绍了 H.323 协议，本章主要分析了无关守情况下 H.323 协议系统的整体构架，并阐述了协议的协议框架流程，为系统基于 H.323 设计方案的讨论做出了铺垫。同时本章中对 H.323 协议在实时远程教学系统中的应用进行了分析。

H.323 协议即是本课题的技术规范之一，对本课题与其它产品之间的互操作性提供了坚实的协议保证。

### 3 系统总体设计

为了解决现代远程实时教学系统中存在的问题，同时为了兼容国际通用的 H.323 通信协议，本文提出了两层框架结构。

两层框架结构分为资源层和 H.323 层这两个层次。资源层主要负责教学资源及用户信息等的传输；H.323 层完全遵循 H.323 通信协议，采用 Openh323<sup>[xxxi]</sup>协议栈实现教学课程的初始化、教学过程信令及音视频数据的传输。根据两层不同的实现特性，分别采用了 C#及 C++两种不同特点的编程语言进行开发。

本章以下内容将详细介绍系统的总体框架及两层的设计方案。

#### 3.1 远程实时教学系统需求分析

本系统整合教师教学现场的视频、音频、结合同步教学课件命令实时直播，形成完善的教学资源流，利用计算机网络传送到远端学生的电脑终端，在学生电脑上实时播放。在实时直播的同时，学生可实时与教师进行视频、音频等多种媒体的交互，本地也可记录教师所有的教学指令，形成教学资源。实时交互过程，如图 3-1 所示，不仅包括了基本的教师图像、语音，还包括了与教师动作同步展示教学资源，PowerPoint 文档、Word 文档、HTML 文档及图片等资料。

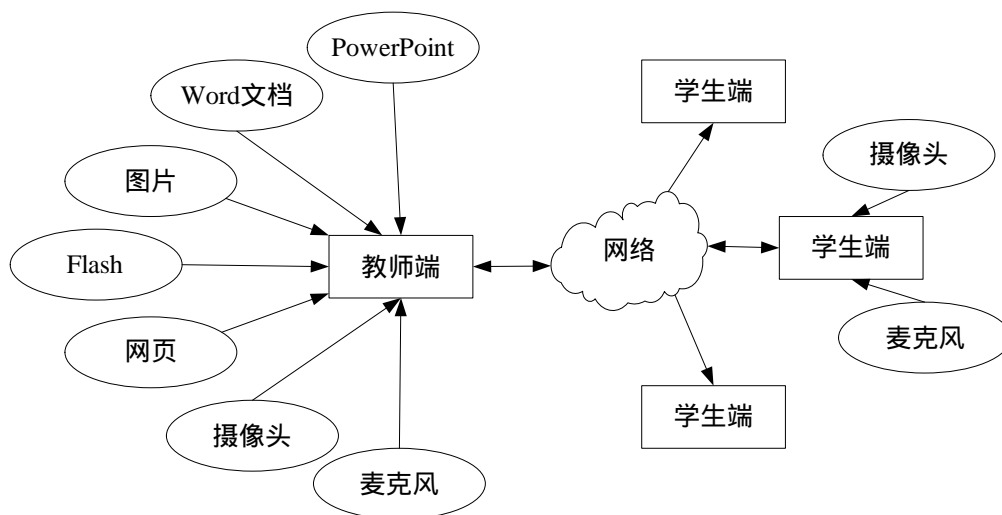


图 3-1 系统功能图

### 3.1.1 系统功能需求

本系统用户分为：教师和学生

#### 1. 课件教学功能

课件浏览功能：教师可将上课所需的图片、各种 office 文档、网页课件等多媒体资源装入课件列表，在上课时实现主控式的同步浏览。

音视频实时交互功能：教师能够把自己的视频广播到教室中，或者广播某个学生端的视频；在教师广播自己视频时，也可以监视某个学生的视频。教师能把自己的声音广播出去，同时可以允许某个学生说话，或者同时允许三个学生广播声音。教师端有学生举手状态的显示框，教师可以选择举手的学生，并允许发言。学生在被允许发言时，系统会弹出对话框，提醒学生发言。教师与学生之间能实现正确实时的交互效果。

文字交互系统：在文字讨论区，教师、学生可以与全体人员或某个人进行文字交谈，当选择对一个人发送信息时，课堂中其它成员则看不到。文字交互系统中还有关于老师操作信息的系统提示和课堂公告，如教师允许某人发言、教师停止某人发言等。

#### 2. 辅助教学功能

网页共享：教师可将某个网页展示给学生浏览。教师输入该网页的 URL 地址请求该网页，所有学生端将同时浏览到此网页。

文件共享：可以共享发送一些文档资料。所有用户都可以上传或下载文件。

#### 3. 管理功能

用户管理：对师生的帐号、密码进行权限管理和分组管理。

选择课程：支持多个课程同时上课，具有多课程、帐号等管理功能。

系统设置：教师端可以根据网络环境情况，选择相应的视频压缩格式，从而调整教学系统所需的带宽。

### 3.1.2 系统设计目标

#### 1. 系统的正确性

任何一个系统都应当保证其运行效果是正确无误的，否则即使它做得再好，也不会为人称赞，更不会有任意的使用价值。待开发的系统应首先满足它的正确性，在此基础上还应强调它的执行效率，即系统对资源的占用率，系统的健壮性等多方

面因素。

## 2. 系统的智能性

系统应借鉴国外优秀教学系统的经验，在实现远程教学系统主要环节的技术上大胆探索，提出智能化教学的概念。智能代表着学生能够感知自身和环境的状态，通过采用一系列的动作实现系统预先设定的目标。

## 3. 系统的创新性

在待开发的系统中，应充分借鉴其它的教学系统的优点，力争在其基础上有所创新，有所提高。同时应发挥当今计算机应用领域中的诸多成果，将其融入本系统。考虑到计算机应用领域日新月异，计算机发展速度之快，在设计期应能充分考虑到本系统的扩展功能，最大限度的增大系统的可扩展性。

# 3.2 系统总体设计框架

远程实时教学系统是用于远距离教学的系统，如图 3-1 所示，它的根本是声音和图像的实时交互，如前文所述，这一点已完全可以由 H.323 协议规定。但除此之外，教学过程中还存在其它教学资源，如 PowerPoint、Word 文档等等，这些教学资源贯穿于整个教学过程始终，现代的远程教学系统多将这些教学资源转化成视频数据，与用户图像一同传输。

本系统也可使用 H.323 协议规范传输教学资源视频流，但教学资源的尺寸一般比较大，其上的文字等细节又必须要求一定的清晰度，作为视频传输将消耗过大的网络带宽，因此本文提出一种两层框架结构以解决此问题。

对于远程教学系统而言，音视频与教学资源有着不同的传输特性。

音视频每一秒都在变化，但对于清晰度要求并不是很高，声音有短暂的间隔并不影响学生听懂教师的发言，同样，看不清教师的脸部细节也并不影响学生学习的效果。且在远程教学系统中，对音频的依赖度大大高于视频，学生不是从教师的脸部细节中获取信息，而是从教师的声音中获得更多的信息。

对于教学资源则完全相反。教学资源往往长时间不切换，但对清晰度的要求很高。学生从 PPT 等资源中的文字、图像等细节获取教学信息，因此对于文字、图像等细节必须以非常高的清晰度传输。

由于以上两者具有完全相反的传输特性，因此使用视频流传输教学资源是非常不科学的。由于教学资源长时间不切换，因此可以将此资源一次性传输至客户端，并在客户端保持不变呈现给用户，当需要切换到另一份资源时再一次性传输该资源，在客户端切换。这样去除了视频流传输所导致的大量数据冗余。对于教学资源需要

较高的清晰度，因此本系统采用 HTML 网页的方式传输教学资源，数据量小，且失真度小。

本远程教学系统构架图如图 3-2 所示。根据上述两种资源不同的传输特性，将整个系统分为两层：资源层和 H.323 层。资源层以 HTML 请求的方式传输 PPT、Word 等教学资源；H.323 层则在 H.323 协议的规范下传输媒体流和控制信令。

本系统共有三个服务器：MCU 服务器、资源服务器和数据库服务器。MCU 服务器完全遵循 H.323 协议，在 H.323 协议的规范下传输媒体流和控制信令；资源服务器负责响应终端对教学资源的请求，将教学资源传输给各个终端；数据库服务器存放所有系统数据，包括会议数据和用户数据，MCU 服务器从数据库服务器获取会议数据以维护所有会议，终端从数据库服务器获取用户信息以完成访问控制、用户权限控制等功能。三个服务器各司其职，互不干扰，即分散了系统的负载，又明确了各自的权责，方便开发和维护。

终端分为教师端和学生端。教师端拥有比学生端更大的权限，可以切换教学资源，可以选择正在发言的学生；学生端则被动的受控于教师端。所有终端不直接通信，仅通过服务器进行交互。根据交互数据性质的区别，本文把交互数据分为两类，一类是信令及音视频数据，一类是资源数据，这两类资源分别在不同的层次进行传输，互不干扰。

如图 3-2 所示，所有终端在教学开始时使用 H.323 协议与 MCU 服务器建立连接，在教学过程中一直保持连接，MCU 服务器将维护整堂课程。教师在教师端将各种教学资源处理成 HTML 网页格式上传到资源服务器供所有终端获取。当教师上传完教学资源便可以开始教学过程。

当教师需要切换某份教学资源时，将该资源在资源服务器上的 URL 地址发送给 MCU 服务器，MCU 服务器向所有终端转发，终端收到该信令后向资源服务器索取该资源，各终端即可浏览到该资源。同样当教师需要向学生展示某网页时，将该网页的 URL 地址（不一定指向资源服务器）发向 MCU 服务器，各终端将以同样的方式浏览到该网页。

当教师需要要求某位学生发言时，教师端向 MCU 发送特定的信令，MCU 将该信令转发到这位学生的终端，此终端接收到该信令后自动向 MCU 发送本地音视频数据，音视频经 MCU 混合处理后分发到各个终端，所有人将接收该学生的音视频，同时所有终端改变该学生的状态，该学生正在发言。同样，当教师需要停止某位学生发言时，教师端向 MCU 发送停止某人发言的信令，MCU 将该信令转发到这位学生的终端，此终端接收到该信令后自动停止向 MCU 发送本地音视频数据，



所有人将不再接收到该学生的音视频，同时所有终端改变该学生的状态，该学生为未发言状态。对于教师的音视频默认传输，所有终端将始终能看到教师的声音和图像。

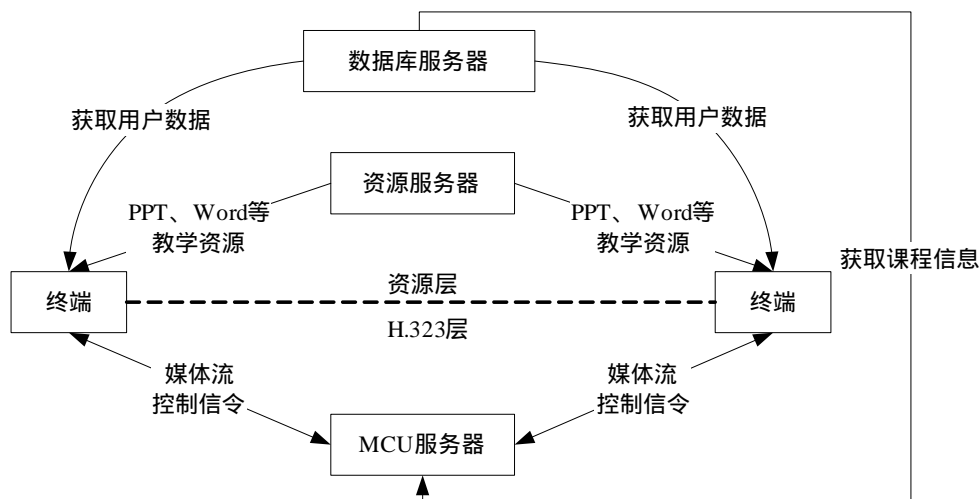


图 3-2 系统总体构架图

使用这种信令和数据分离的两层体系结构，可将资源数据的传输转化成获取资源的命令的传输，减少了传输的数据量，不仅将负载从终端转移到服务器，也将数据量比较大、传输特性完全不同的两种数据：资源数据和音视频数据，分别由两个服务器负载，实现了数据量的分散、数据管理的分类集中。

系统共分为 MCU 服务器、终端、资源服务器和数据库服务器四部分。系统模块设计图如所示。本文主要 MCU 服务器和终端两部分的设计和实现。

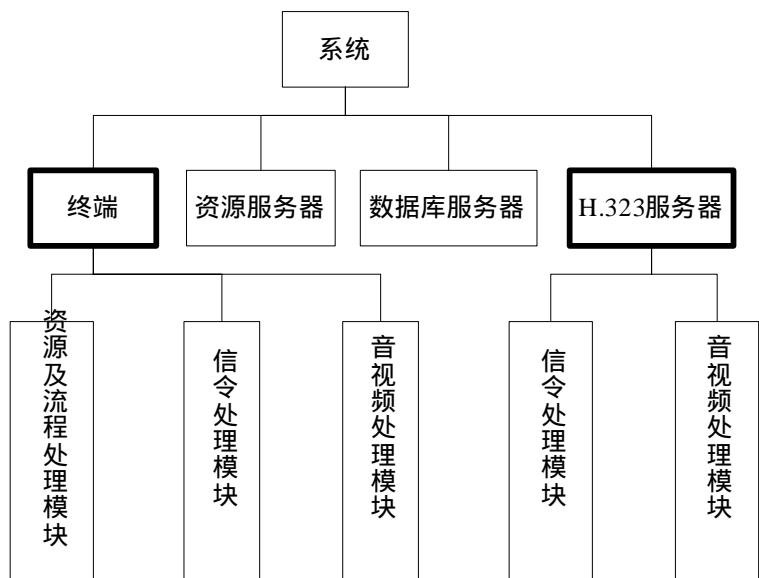


图 3-3 系统总体模块结构设计图

H.323 服务器是只具有 H.323 层的服务器，该服务器主要包括信令处理模块和音视频处理模块两大模块。信令模块处理所有终端的连接，并中转某些连接的自定义信令；音视频处理模块主要处理终端传来的音频和视频数据，并对这些数据进行混合并向所有终端发送。

终端部分分为资源层和 H.323 层两层。资源层主要是资源处理模块，主要负责资源处理、资源上传和流程控制功能。H.323 层包括信令处理模块和音视频处理模块两大模块。终端的信令处理模块主要处理会议的控制信息和用户命令的自定义信息；音视频模块主要负责终端音视频的采集、处理、显示等等。

以上所述模块均在第四章、第五章详细讨论。

### 3.3 资源层设计方案

资源层负责教学资源的传输和数据库数据的获取。由于本层不涉及的比较底层的操作，因此在实际开发中，本层使用 C#语言进行开发。

C#是一种新的编程语言，它是为生成广泛的在 .NET Framework 上运行的企业级应用程序而设计的。C#从 C 和 C++演变而来，是一种简单、现代、类型安全和面向对象的语言。C#代码被编译为托管代码，这意味着它可以从公共语言运行库的服务中获益。这些服务包括语言互操作性、垃圾回收、增强的安全性和改进的版本支持。由于 C#对网页服务和数据库服务都作出了标准的封装，省去了垃圾回收等维护开销，且它具有跨语言集成、开发效率高等方面的优势，故在资源层使用托管代码 C#实现。

#### 3.3.1 资源处理

资源传输是资源层最主要的功能，资源传输贯穿于教学过程始终。资源传输部分的核心是教师端将各种格式资源统一转换为 HTML 格式，通过网页请求的方式传输。

资源格式转换是资源传输的基础。各种资源格式不同，其转换的方式也不同。如图 3-4 所示，Microsoft PowerPoint 和 Microsoft Word 这两种软件均提供了相应的格式转换接口，因此本系统使用其 C#接口将这两种资源分别分页，并转换成 HTML 格式。对于图片和 Flash 可以使用添加 HTML 标签的方式直接转换成 HTML 格式。而网页格式不需要转换。

格式转换不仅将各种资源格式统一处理成数据量小，不损失细节的 HTML，便

于传输和统一管理，且即使学生端未安装 PowerPoint、Word 等软件，仍然可以浏览教学资源，进一步降低了对学生端的需求。

格式转换完成后，如图 3-4 所示，将所有 HTML 网页以特定文件组织结构组合成课件包，然后将课件包以 FTP 方式上传到资源服务器的虚拟目录下，供学生获取以网页方式浏览。

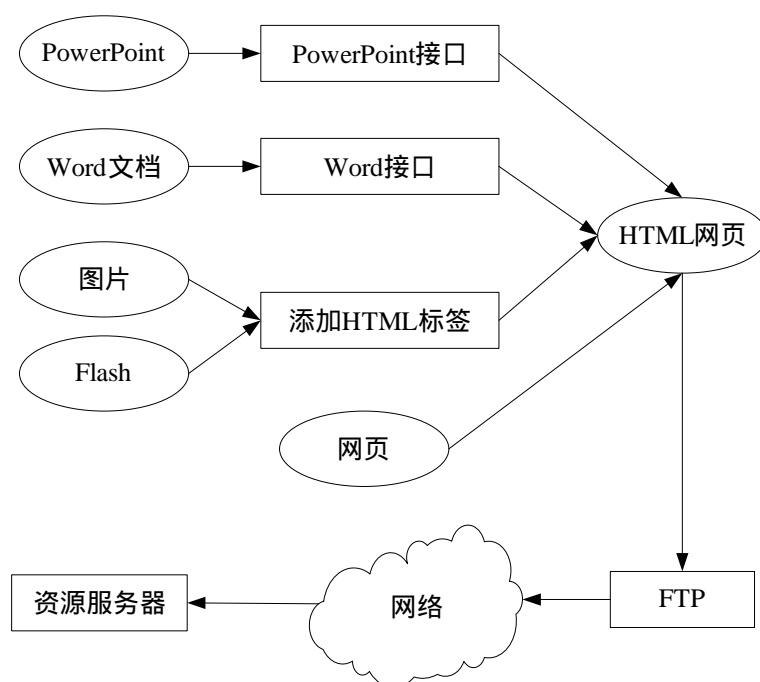


图 3-4 资源格式转换示意图

### 3.3.2 数据库数据获取

远程教学系统中存在着很多数据，主要分为用户数据和课程数据，这两种数据在终端系统和 MCU 服务器中均有应用。本系统使用数据库对这些数据进行统一管理。

终端在登录系统时，须验证用户身份，在授课过程中可查询其他用户的公开信息，使用这些功能时终端需获取用户数据；在登录系统后，用户须从所有课程中选取要上的课程，获取所有课程信息列表时需获取课程数据。

MCU 服务器在某一用户进入该服务器维护的课程时，服务器需获取该用户的身份信息以便维护，此时需获取用户信息；服务器在启动时需获取所有课程的信息以便维护，此时需获取课程信息。

一般而言，如图 3-5 所示终端或 MCU 服务器应直接连接数据库获取以上数据

并在本地进行数据的相关处理。

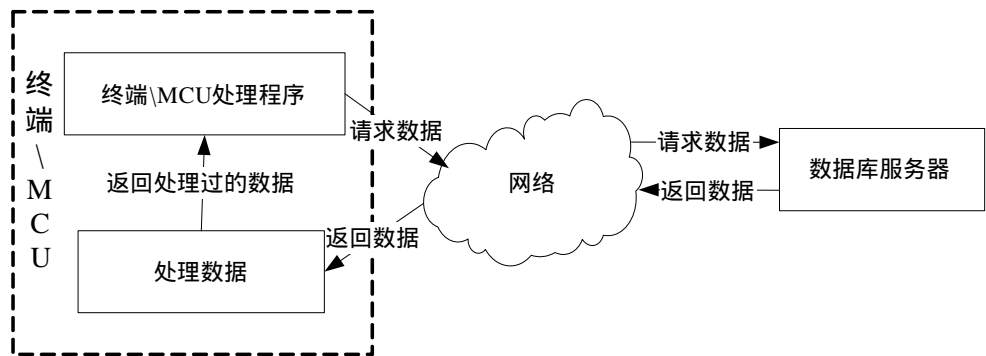


图 3-5 一般查询数据库方式

但这些数据数量不多，例如验证身份信息时终端只需获得对用户登录的确认或者拒绝；对于用户的公共信息也只有姓名、性别等少数几项；课程列表信息也只是课程名、任课教师等少量信息。为获取如此少量的信息而连接数据库，不但消耗大，且不安全。因此，如图 3-6 所示，本系统中采用 Web Service<sup>[xxxii]</sup> 技术查询数据库。

终端或 MCU 服务器向数据库服务器发出 Web Service 请求<sup>[xxxiii]</sup>，Web Service 服务端接收到 Web Service 请求后在本地查询数据库，本地处理程序接收数据库返回的数据，处理后将处理结果返回给终端或 MCU 服务器。这种方式不需要将数据库访问的帐号、密码等信息暴露在网络中，也不需返回大量待处理的数据，大大增强了系统的安全性<sup>[xxxiv]</sup>。

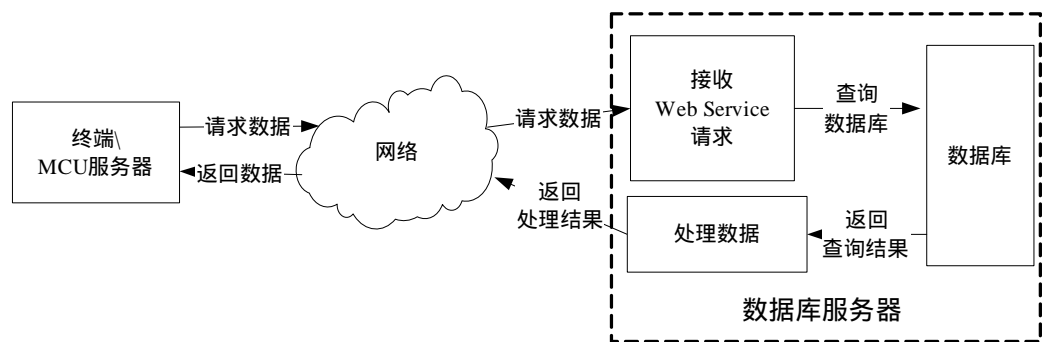


图 3-6 Web Service 式查询数据库方式

### 3.4 H.323 层设计方案

H.323 层主要负责信令和音视频的传输。本层完全基于 H.323 协议，采用 Openh323 协议栈实现教学会议的初始化、教学过程信令及音视频数据的传输<sup>[xxxv]</sup>，使用 C++ 语言进行开发。

本层使用集中式多点会议模式，如图 3-7 所示，所有媒体流及控制信令均由 MCU 服务器中转。

终端的 H.323 层处理媒体流的采集和发送，会议的发起和建立等工作；MCU 服务器则处理多路媒体流的接收、混合处理及发送，会议的建立及维护及控制信令的处理和转发工作。因此，终端不包括 MC 及 MP，MCU 服务器同时包含 MC 和 MP<sup>[xxxvi]</sup>。

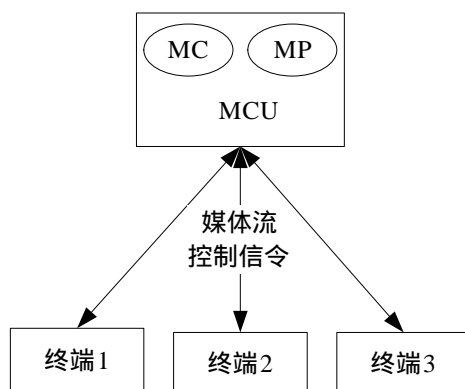


图 3-7 H.323 层整体构架

### 3.4.1 自定义信令系统结构

由于本系统是不对等权限系统。教师拥有较高的权限，可以要求学生发言，对正在发言的同学可停止学生发言，对于申请发言的同学可同意或拒绝学生发言；而学生则权限较低，学生仅能申请发言和停止发言。各终端也需要同所有进入课程的终端的状态。因此，本系统设计了一套专用于远程教学的信令。

#### 1. 自定义信令系统

如图 3-8 所示，自定义信令系统分为三部分：文本交流信令、地址信令和控制信令。图中细框所示即是所有的自定义信令。

文本交流信令 CHAT 是负责文本交流的信令，本系统中的文本交流功能即使用此信令实现。CHAT 信令中携带了用户的用户名和文本交流信息，由 MCU 服务器中转至所有用户，达到一人输入文字，所有人可见的目的。此信令还拥有一个目的地址字段，若发送方设置了该字段，则 MCU 服务器将只向该地址转发此信令，达到私聊的目的。

地址信令携带的是地址信息，本系统中有两种地址信令：HTTP 地址信令和 FTP 地址信令。本系统中的课件浏览和网页共享功能使用 HTTP 地址信令

HTTP\_ADDRESS 实现。教师端资源层将所有教学资源转换成 HTML 格式上传至资源服务器，当教师需要切换教学资源或展示网页时，教师端向 MCU 服务器发出 HTTP\_ADDRESS 信令，该信令携带了教学资源或网页的 URL 地址，其它终端收到 MCU 服务器转发的该信令后，其资源层即可使用此 URL 请求该网页获取资源。本系统中的文件共享功能使用 FTP 共享文件，每个会议有一个分配的 FTP 地址。在终端加入某一会议时并不了解该 FTP 地址，因此需向 MCU 服务器请求该 FTP 地址，MCU 服务器使用 FTP\_ADDRESS 信令发回该地址。在终端需要使用 FTP 共享功能时即可使用该地址开始 FTP 连接过程。

控制信令是控制教学过程的信令，该信令是整个自定义信令系统的核心。控制信令又分为确认信令 ACK，拒绝信令 NAK，MCU 信令及终端信令：

- 1) ACK 信令是对请求的确认。它携带了原请求信令的内容。
- 2) NAK 信令是对请求的拒绝。它同样携带了原请求信令的内容。
- 3) MCU 信令是 MCU 服务器特有的信令，它向各终端传递终端无法获取的信息。
- 4) GIVETOKEN 信令是告知 TOKEN（令牌）信令，TOKEN 是 MCU 服务器为每个终端生成的唯一标识，因此本系统中均以 TOKEN 来唯一标识各个终端。但这个标识只在服务器端生成，终端无从得知，因此定义了此信令。服务器使用此信令告知终端其对应的 TOKEN。
- 5) ENTER 信令是用户进入信令，当一个用户进入该课程时服务器向所有终端发送此信令，更新所有终端的用户列表。该信令携带了进入用户的用户名、TOKEN 等信息。
- 6) LEAVE 信令是用户离开信令，当一个用户退出该课程时服务器向所有终端发送此信令，更新所有终端的用户列表。该信令同样携带了退出用户的用户名、TOKEN 等信息。
- 7) MCULIST 命令是用户列表更新命令。MCU 服务器为每个课程都维护了一个详细的用户列表，该列表包括所有用户的用户名、TOKEN、当前状态、音频能力、视频能力。MCU 服务器定时向所有终端发送此列表，更新所有终端的用户列表。

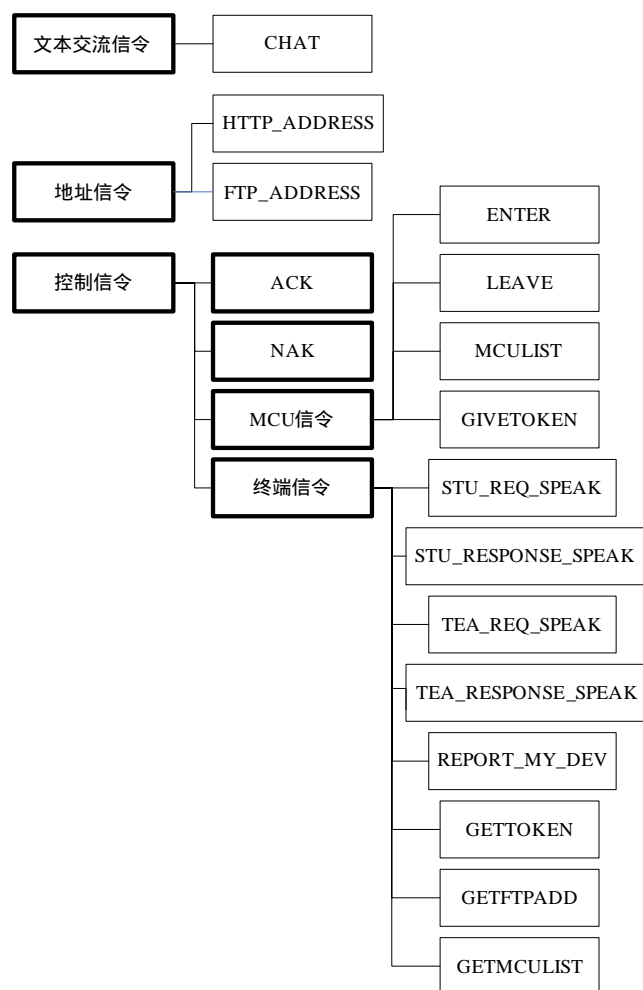


图 3-8 自定义信令系统结构

终端信令是终端用来控制教学的信令：

- 1) GETTOKEN 信令是请求 TOKEN 信令。终端连接到服务器之后向服务器发送此信令索取自己的 TOKEN。
- 2) GETFTPADD 信令是请求 FTP 地址信令。如前文所述，终端在初始化时需要向服务器请求该会议的 FTP 地址以完成 FTP 连接。
- 3) GETMCULIST 信令是请求 MCULIST 信令。终端在初始时没有任何用户列表信息，因此初始化时向 MCU 发送此请求获取当前的列表，创建终端的用户列表。
- 4) REPORT\_MY\_DEV 信令是设备命令。MCU 并不知道终端音视频设备是否可用，因此终端在初始化时向 MCU 发送此命令报告自己的音视频设备能力。
- 5) STU\_REQ\_SPEAK 信令是学生申请发言信令。在学生申请自己发言时使用。

- 6) STU\_RESPONSE\_SPEAK 信令是学生响应信令。该信令分为 ACK 确认信令和 NAK 拒绝信令两种，是对学生发言成功与否的响应。终端收到此信令则根据确认或拒绝信息更新该用户的状态。
- 7) TEA\_REQ\_SPEAK 信令是教师要求发言信令。在教师要求某位学生发言时发送。
- 8) TEA\_RESPONSE\_SPEAK 信令是教师响应命令。该信令同样分为 ACK 确认信令和 NAK 拒绝信令两种，是教师对学生发言的响应。

## 2. 自定义信令结构

文本交流、地址信令、控制信令三种信令不仅功能不同，其信令结构也不尽相同。

文本交流信令结构如表 3-1 所示，目的 Token、源 Token 标识接收和发送方，ISCHAT 表示信令种类为文本交流信令，随后是交流文本和附加信息。

表 3-1 文本交流信令结构

目的 Token	源 Token	ISCHAT	交流文本	附加信息
----------	---------	--------	------	------

地址信令结构与文本交流格式有所不同，以 HTTP 地址为例，如表 3-2 所示，目的 Token、源 Token 标识接收和发送方，ISADDRESS 表示信令种类为地址信令，随后是 HTTP 地址和附加信息。

表 3-2 地址信令结构

目的 Token	源 Token	ISADDRESS	HTTP	HTTP 地址	附加信息
----------	---------	-----------	------	---------	------

控制信息信令格式比以上两种信令格式复杂，一般而言控制信息的信令格式如表 3-3 所示。目的 Token、源 Token 同样标识接收和发送方。ISSIGNAL 表示信令种类为控制信令。信令属性分为以下几种：确认(ACK)、拒绝(NAK)、终端控制(CTRL)及 MCU 信令(MCU)。对于控制信令一般要携带其音视频能力信息。随后是附加信息。

表 3-3 控制信令基本结构

目的 Token	源 Token	ISSIGNAL	信令属性	信令名	音视频能力	附加信息
----------	---------	----------	------	-----	-------	------

## 3.4.2 自定义信令流程

### 1. 终端加入课程初始化流程

终端加入课程的初始化过程中，有大量 H.323 协议规定外的信息需要与 MCU 服务器交互，因此使用了大量上节所述的自定义信令。



信令流程如图 3-9 所示。首先终端与 MCU 之间建立 H.323 连接，连接建立完成后即开始自定义信令初始化流程。在进行所有操作之前，终端必须先获取自己的 TOKEN，因此终端向 MCU 发送 GETTOKEN 信令，MCU 以 GIVETOKEN 发回终端的 TOKEN。随后，终端向 MCU 发送 REPORT\_MY\_DEV 信令报告自己的音视频能力，MCU 若收到则发回该信令的 ACK 确认，若终端在一定时间内没有收到 ACK，则继续发送 REPORT\_MY\_DEV 信令。最后，终端需要建立自己的用户列表，故终端向 MCU 发送 GETMCULIST 信令请求列表，MCU 以 MCULIST 信令发回当前的用户列表。创建了用户列表后，初始化流程结束，终端与 MCU 开始正常的交互流程。

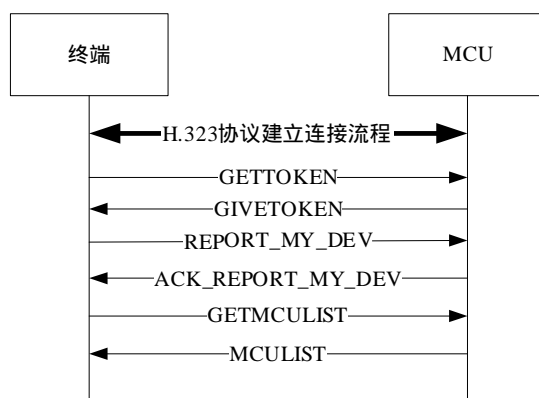


图 3-9 终端加入课程初始化流程

## 2. 学生申请发言流程

学生端可以主动申请发言，教师可以批准学生发言，也可以拒绝学生发言，学生发言后所有终端需更新该学生的状态信息。

学生申请发言流程如图 3-10 所示，学生先向 MCU 发送 STU\_REQ\_SPEAK 信令，教师端接收到 MCU 转发的此信令后，可发出 TEA\_RESPONSE\_SPEAK 信令的 ACK 确认信令或 NAK 拒绝信令。若学生端收到 MCU 中转回的 NAK 信令，则之前发送的 STU\_REQ\_SPEAK 信令失效；若学生端收到 MCU 中转回的 ACK 信令，则可以打开音视频通道，向 MCU 发送媒体流。若音视频通道打开成功，则学生端向 MCU 发送 ACK\_STU\_RESPONSE\_SPEAK 信令表示学生端已发言，MCU 向所有终端转发 ACK\_STU\_RESPONSE\_SPEAK 信令，收到该信令的终端更新此用户信息为正在发言，若音视频通道打开失败，则学生端向 MCU 发送 NAK\_STU\_RESPONSE\_SPEAK 信令表示学生端发言失败，之前发送的 STU\_REQ\_SPEAK 失效。

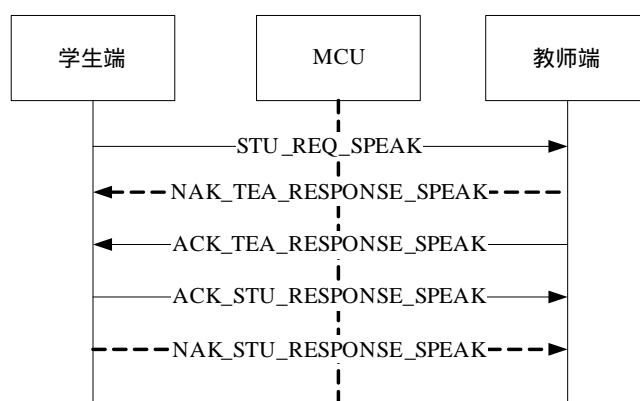


图 3-10 学生申请发言流程

### 3. 教师端要求学生发言

教师端要求学生发言的流程与学生申请发言颇为不同。教师要求发言，学生不能拒绝发言，若设备允许则自动发言。

教师端要求学生发言流程如图 3-11 所示，教师端向 MCU 发送 TEA\_REQ\_SPEAK 信令，学生端接收到 MCU 转发的该信令后，自动打开音视频通道，向 MCU 发送音视频。若通道打开成功，则学生端向 MCU 发送 ACK\_STU\_RESPONSE\_SPEAK 信令，MCU 转发给所有终端，收到确认信息的终端更新发言学生的状态为正在发言；若通道打开失败，则学生端向 MCU 发送 NAK\_STU\_RESPONSE\_SPEAK 信令，TEA\_REQ\_SPEAK 信令失效。

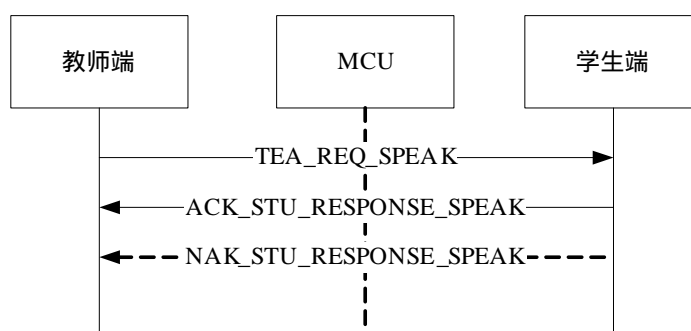


图 3-11 教师要求学生发言流程

### 4. 停止发言流程

停止发言流程比发言流程简单得多。教师停止某学生发言，只需向该学生发送 NAK\_STU\_RESPONSE\_SPEAK 信令，学生即暂停音视频通道，并向 MCU 发出 NAK\_STU\_RESPONSE\_SPEAK 信令，所有终端收到后更新该学生信息为未发言。若学生停止自己发言，则只需暂停音视频通道，并向 MCU 发送 NAK\_STU\_RESPONSE\_SPEAK 信令，所有终端收到后更新该学生信息为未发言。

### 3.5 本章小结

本章第一节首先介绍了远程实时教学系统的需求分析，本节就系统的功能需求和希望达到的设计目标展开对需求的讨论。第二节介绍了系统构架及两层框架结构，本节给出了两层框架结构的设计原理图，并根据此原理图介绍了本系统的设计原理。第三节中介绍了资源层的设计方案。本层采用 C#.NET 进行开发，本节介绍了资源层资源处理技术和数据库数据获取技术的设计及具体实现。第四节中介绍了 H.323 层的设计方案。本节主要说明了 H.323 层的整体构架，介绍了自定义信令系统及自定义信令流程。本层采用 C++ 语言及 Openh323 协议栈进行开发。

文章描述的两层框架结构一定程度上解决了现代远程教学系统中存在的数据量过大的问题，为远程教学系统的发展和改善做出了尝试和贡献。后文中笔者将分别从 H.323 服务器及终端系统两方面具体介绍本系统的设计及实现。

## 4 H.323 服务器设计与实现

本远程教学系统的服务器分为 H.323 服务器、网页服务器及数据库服务器，H.323 服务器是最重要的一个，他完成了所有信令及音视频的传输工作，是系统不可缺少的一部分。H.323 服务器完全遵循 H.323 协议，相当于 H.323 协议中的多点控制单元 (MCU)，本系统对 Openh323 协议栈 MCU 部分作了一定的改进，加入了自定义信令的处理，使 MCU 具备了服务远程教学系统的能力。

### 4.1 H.323 服务器处理流程设计

本系统采用集中多点式会议模式<sup>[xxxvii]</sup>，所有终端需与 MCU 建立连接，终端通过建立连接与 MCU 交互控制信息和媒体流。每个终端维护一个连接，而 MCU 则同时维护多个连接。在一个大型系统中，一个 MCU 往往需要维护上万个连接。对于 MCU 来说，一个连接就是一个用户，MCU 对于用户数据的处理是通过对连接的维护实现的<sup>[xxxviii]</sup>。

MCU 从初始化到关闭的流程如图 4-1 所示。MCU 在初始化之前需要导入存放在指定地点的配置文件，此配置文件中存放了 MCU 的所有配置信息，包括 IP 地址、监听端口号等等。MCU 根据此配置信息初始化整个系统。

系统初始化完毕，MCU 还不能正常使用，还必须在界面上手动开启要开通的课程，只有在课程开通之后，用户才可以使用该课程。

由于连接可能随时到来，故 MCU 始终运行着一个监听线程，负责处理所有到来连接。连接到来时，系统要先检查连接字符串，该连接字符串有多种表达形式，主要信息包括：MCU 名\MCU 地址、房间 ID、用户名及密码。系统首先检查此字符串是否是规定格式，若不是则立即拒绝连接。检查完格式后，系统继续检查 MCU 名或 MCU 地址，若该项表示的是自己，则字符串合法。对于非法字符串，系统立即拒绝连接。然后系统查看请求进入的房间是否已开启，若未开启则拒绝连接，否则可以接收连接。

系统对于每个已接收的连接都要处理控制信息和媒体流直至系统关闭。

图 4-1 即是 MCU 的基本处理流程。但是 MCU 可能同时开放多门课程，如果 MCU 在所有连接范围内处理控制信令和媒体流，则对于多门课程的处理将非常混乱。因此，图 4-1 中的“处理各连接控制信令和媒体流”的过程对于所有连接并不是统一的。

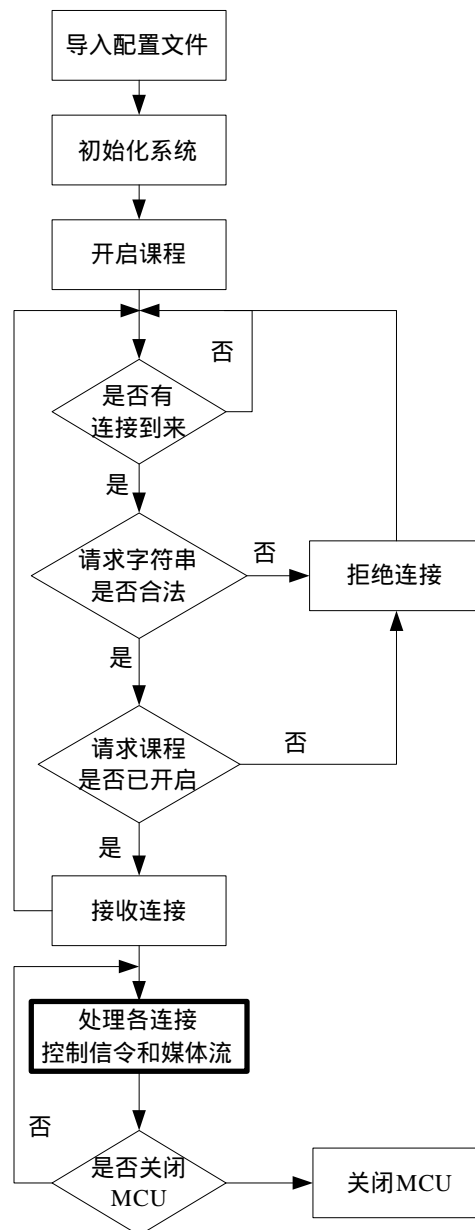


图 4-1 服务器处理连接流程图

自定义信令处理功能部分关键代码如表 4-1 所示：

该代码实现的处理流程与图 4-1 所描述的处理流程一致。代码省去了对各种信令的具体处理，这部分在后文将详细介绍。

表 4-1 信令处理部分函数代码

```

BOOL H323Connection_ConferenceMember::ProcessCommand(PString & result, const PString & str)
{
    CmdClient cmd;
    if (!cmd.ParseCmdPDU(str))
    {

```

```

        PTRACE(1, "Invalid command: " << (const char*)str);
        return FALSE;
    }
    OpenMCUH323Connection * conn = (OpenMCUH323Connection *) GetH323ConnectionWithLock();
    if (conn == NULL) return FALSE;
    PString userID = conn->GetRemoteName();
    PString roomID = conn->GetRequestedRoom();
    PString callToken = GetH323Token();
    if (!ep.GetConferenceManager().HasConference(conn->GetConferenceIdentifier()))
    {
        PTRACE(1, "ROOM does not exist.");
        return FALSE;
    }
    BOOL isTeacher = GetConference()->IsTeacher(userID);
    PString role = isTeacher ? TEACHER : STUDENT;

    // 索要自己 token
    if ( cmd.GetCmdType() == CmdClient::CMD::CTRL_CLIENT_GETTOKEN )
    {
        CMD_PDU pdu = cmd.BuildMCUGiveTokenPDU( callToken, role );
        conn->SendUserInput(pdu);
        return FALSE; // 不向其他人转发该信令
    }
    // 源与实际 token 不同, 拒绝转发
    if ( cmd.GetSouToken() != callToken )
        return FALSE;
    // 处理所有的命令, 维护 MCU 内部数据
    switch ( cmd.GetCmdType() )
    {
        case CmdClient::CMD::CTRL_CLIENT_REPORTMYDEVICE:
        case CmdClient::CMD::ACK_CLIENT_REPORTMYDEVICE:
        case CmdClient::CMD::NAK_CLIENT_REPORTMYDEVICE:
        case CmdClient::CMD::CTRL_CLIENT_TEAREQUESTSPEAK:
        case CmdClient::CMD::ACK_CLIENT_STURESPONSESPEAK:
        case CmdClient::CMD::NAK_CLIENT_STURESPONSESPEAK:
        case CmdClient::CMD::CTRL_CLIENT_STUREQUESTSPEAK:
        case CmdClient::CMD::ACK_CLIENT_TEARESPONSESPEAK:
        case CmdClient::CMD::NAK_CLIENT_TEARESPONSESPEAK:
        case CmdClient::CMD::CHAT:
        case CmdClient::CMD::CTRL_CLIENT_GETMCULIST:
        case CmdClient::CMD::CTRL_MCU_PERSONLEAVE:
        case CmdClient::CMD::FTP_ADDRESS:
    }
}

```

在 MCU 系统中课程以房间标识, 一个课程就是一个房间。房间的数据结构示意图如图 4-2 所示。一个房间中包含了 N 个用户的信息, 而每个用户又包括用户名、密码、房间号及 TOKEN 等信息。

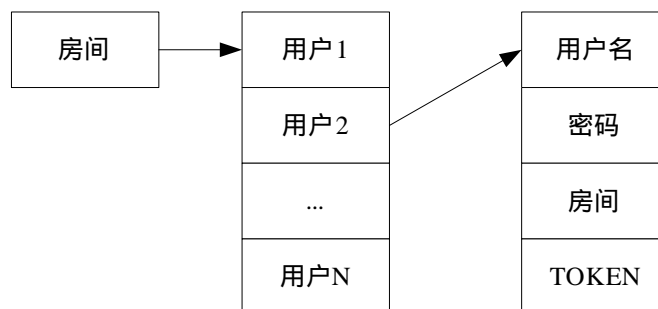


图 4-2 房间数据结构示意图

MCU 中控制信令和媒体流的处理都是以房间为单位的。当 MCU 接收一个连接后，系统即将此连接归类到某一个房间，之后对连接传输数据的处理都是以房间为组进行。例如，某终端向 MCU 发送控制信令，MCU 需要向所有终端转发，这里的转发是以房间为单位的，所有终端指的是这个房间内的所有终端，而不是整个 MCU 处理的所有终端。对于媒体流也一样，后文将介绍的音频和视频的混合处理、发送，都是在一个房间的单位内进行的。

## 4.2 H.323 服务器信令处理功能实现

MCU 对自定义信令的收发都是使用用户输入通道，所有的自定义信令都作为用户输入传输。系统对这些把输入转换为与自有信令一样的 ASN.1 格式传输。MCU 对自定义信令的处理流程如图 4-3 所示：

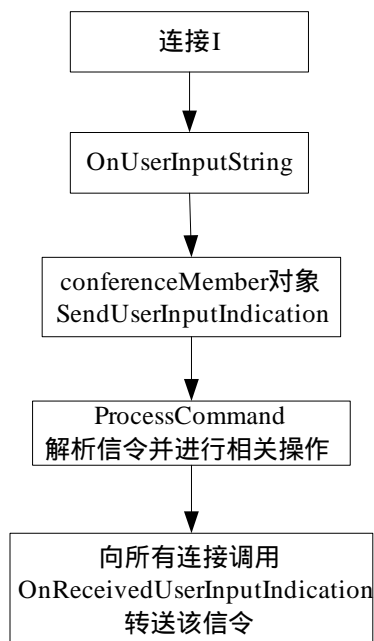


图 4-3 MCU 自定义信令处理流程

当连接 I 上接收到自定义信令是，该连接的 OnUserInputString 方法被调用，这个方法是所有自定义信令的输入接口。OnUserInputString 方法将调用 conferenceMember 对象的 SendUserInputIndication 方法。SendUserInputIndication 方法首先调用 ProcessCommand 方法，此方法首先对自定义信令进行解析，然后根据不同的信令种类进行不同的操作，必要时还需要对该信令进行更改。随后 SendUserInputIndication 方法将依次调用连接列表中所有连接的 OnReceivedUserInputIndication 方法，将信令发送到所有连接上<sup>[xxxix]</sup>。

第三章详细介绍了各种自定义信令及其信令结构，MCU 针对于不同的信令，进行的处理也不同。

对于地址信令，MCU 服务器不作任何处理，直接向同一课程中的所有终端转发。

对于文本交互信令，若目的 TOKEN 字段填写符号“\*”，则 MCU 向同一课程中的所有终端转发；若填写的是同一课程中其他终端的 TOKEN，则向该终端转发。

对于 GETTOKEN 信令，MCU 发回 GIVETOKEN 信令；对于 REPORT\_MY\_DEV 信令，MCU 在用户信息表中更新该终端的用户信息。

对于 TEA\_REQ\_SPEAK 信令及 TEA\_RESPONSE\_SPEAK 信令的确认或拒绝信令，MCU 直接向该终端转发。

对于 STU\_REQ\_SPEAK 信令，MCU 直接向教师端转发。

对于 STU\_RESPONSE\_SPEAK 信令的确认或拒绝信令，MCU 向同一课程中所有终端转发，并同时更新该终端的状态为正在发言或未发言。

对于 GETMCULIST 信令，MCU 向该终端发回当前的 MCULIST。

当有成员进入或离开课程，MCU 向该课程中的所有成员发送 ENTER 或 LEAVE 信令。

另外，MCU 还将定期向所有终端发送当前的 MCULIST。

当然，终端在处理各种信令之前需要先检查其源 TOKEN 是否与该连接的 TOKEN 相同，若不同则拒绝处理此信令。这是出于安全性的考虑，防止有终端冒充其他终端向 MCU 发送信令。

### 4.3 H.323 服务器音视频处理功能实现

音视频模块的处理对于 MCU 来说也是比较重要的一个部分，MCU 服务器音视频模块主要是指其中 MP 的部分，该模块接收所有连接传来的音视频流，并将这些音视频流混合，并发给所有终端。在此模块中，音频和视频有不同的处理和混合方



式。

4.3.1 音频处理

在系统中，同一时刻有多个终端与 MCU 存在连接，这些终端有的将向 MCU 传递音频流，有的则不传递。MCU 音频混合的目的是以特定的策略选择几路音频流，将这几路音频流混合成一路音频，再传递给所有终端。

MCU 为每个终端开辟了一大块音频缓冲区，如图 4-4 所示，每个终端对应一个连接。终端 A 至终端 N 均有独立的音频缓冲区。其中每大块缓冲区分别存放了除该对应终端以外其他终端的音频缓冲数据，例如，连接 I 的音频缓冲区中存放了除 I 以外连接 A 至 N 的音频缓冲数据。因此，在 MCU 中存在着  $N*(N-1)$  份音频缓冲数据。每个连接拥有一个字典和  $N-1$  份音频缓冲数据。

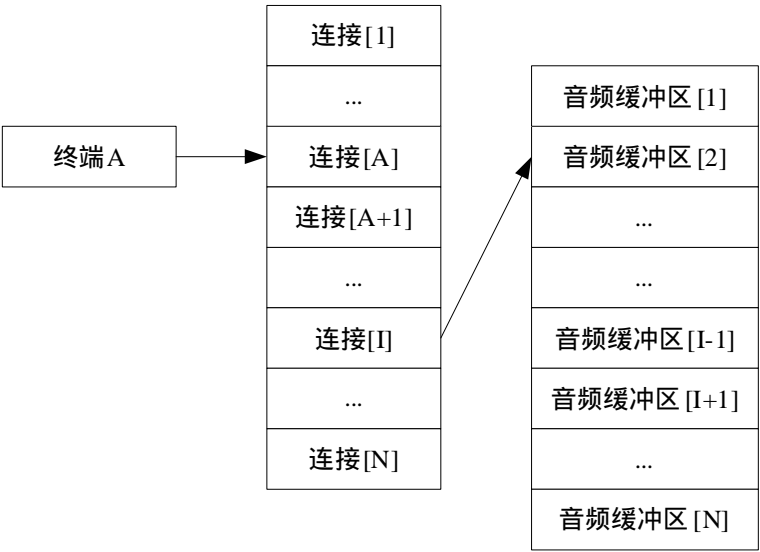


图 4-4 MCU 音频缓冲区结构示意图

当终端 I 的音频数据到来时，系统中 IncomingAudio 对象将处理到来的音频数据：

- 1. 音频编解码器将从网络上获取的音频数据解码并写入 IncomingAudio 通道。
- 2. IncomingAudio 根据将数据分发到连接 I，此时连接 I 上即收到了已解码的音频数据。
- 3. 连接 I 将数据写入 endpoint 对象。endpoint 对象是 MCU 系统中的主要对象，在现实中对的是 MCU 端点，整体上的操作基本上有 endpoint 完成。
- 4. endpoint 对象将数据拷贝到除了连接 I 外该房间的所有连接。

5. 除 I 外的所有连接将数据拷贝到对应的音频缓冲区。

接收数据处理流程完毕, 最终将视频数据存放于连接对象控制的音频缓冲区中。在整个接收过程中, 连接 I 收到的一份数据被拷贝 N-1 次。

当音频编码器 I 要求发送音频数据时, 由 OutgoingAudio 对象处理即将发送的音频流:

1. 音频编解码器向 OutgoingAudio 通道请求音频数据。
2. OutgoingAudio 通道向相应的连接 I 请求音频数据。
3. 连接 I 向 endpoint 请求发送数据。
4. endpoint 的 ReadAudio 方法得知与音频编解码器 I 相关的连接 I 请求过数据。
5. 连接 I 中的 ReadAudio 方法被调用。
6. ReadAudio 方法混合音频缓冲区中的 N-1 份数据。

其中音频混合的方法比较简单, 如图 4-5 所示, 某一连接的缓冲区里有三份音频缓冲数据, 连接中的 ReadAudio 方法将这三份音频数据同时放入混合器混合。而混合器往往仅仅是根据一定策略, 将几份数据相加混合。混合后的数据即可发向对应的连接。

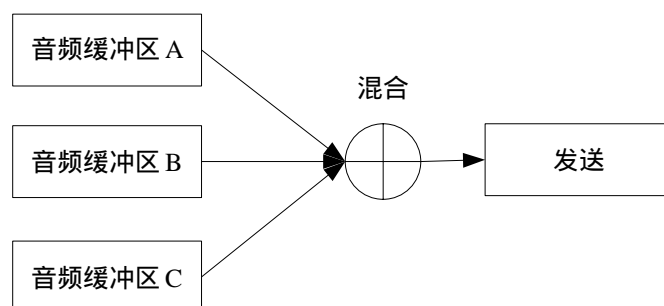


图 4-5 音频混合示意图

### 4.3.2 视频处理

本系统中存在多个用户, 但 MCU 不必将所有用户的视频混合, 只需混合发言同学的视频, 因此, 本系统采用 4 个人 (包括教师) 同时发言的设计。

当视频数据到来时, 网络将视频帧输出给系统, 被视为视频输出设备。该设备的视频编解码器将该数据解码, 并发送给该连接的 OnIncomingVideo 方法处理, 处理方法如图 4-6 所示。

连接的 OnIncomingVideo 方法接收到该视频帧。OnIncomingVideo 方法调用对应 conferenceMember 对象中的 WriteVideo 方法。WriteVideo 方法调用相应 Conference

会议对象中的 WriteMemberVideo 方法。WriteMemberVideo 方法调用该会议对应的视频混合器 videoMixer 对象将此帧数据写入会议的视频缓冲区中。



图 4-6 MCU 视频接收

当需要发送视频流时，数据从系统当向网络输入，被视为视频输入设备。该设备向连接请求发送视频流，处理流程如图 4-7 所示。

连接的 OnOutgoingVideo 方法接收到该视频流。OnOutgoingVideo 方法调用 conferenceMember 对象的 ReadVideo。ReadVideo 方法调用 conference 对象的 ReadMemberVideo 方法。ReadMemberVideo 方法调用会议对应的视频混合器 videoMixer 对象从会议的视频缓冲区中读出该视频帧并发送。

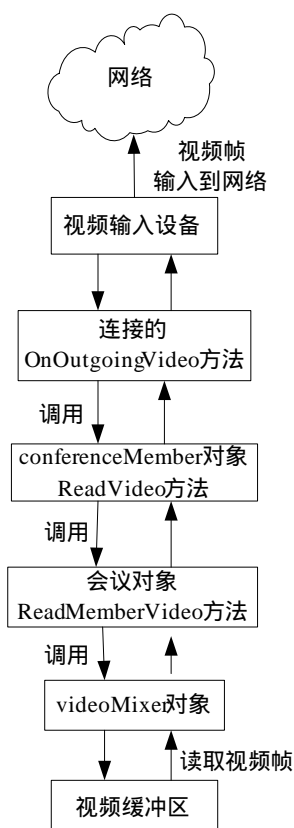


图 4-7 MCU 视频发送

在 MCU 中有一块视频缓冲块 VideoFrameStoreList，所有会议的视频缓冲帧以一定的顺序存放于此缓冲块中。当某连接的一帧视频数据到来时，对应通道的视频编解码器对其进行解码。若该连接是 MCU 允许的四个发言用户中的一个，则将解码后的该帧拷贝到视频缓冲区的对应区域。如图 4-8 所示，连接 A 到来的视频帧存放在左上角 A 的区域，连接 B 到来的视频帧存放在右上角 B 的区域，以此类推。

此方法并没有对视频数据进行真正意义上的混合，只是将四帧图拼接成一帧，然后再编码，发向网络。这种方法保证了所有终端的视频在服务器端同步。终端接收到视频流后，不需要再对视频流进行同步，只需将四幅图从一幅中截取出来即可。

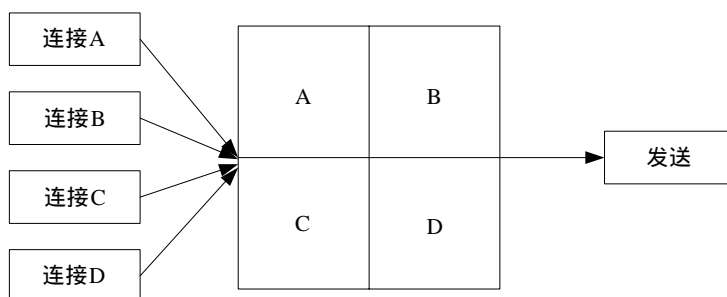


图 4-8 视频混合示意图

这种方法也存在着一定的弊端，服务器端需要对视频数据进行一次编码、解码和拼接，系统资源消耗较大；如果需要混合的图片很多，则使用此方法将组合成过大的图片，过于消耗网络带宽。这都是未来需要改进的地方<sup>[x1]</sup>。

## 4.4 本章小结

H.323 服务器只具有 H.323 层，本章首先介绍了 H.323 服务器的流程设计，由于 H.323 服务器只负责转发工作，故该流程相对于终端部分的流程简单得多。另外本章还介绍了 H.323 服务器的信令功能模块和音视频功能模块的设计，在对 Openh323 协议栈的分析和改进中完善了这两个模块的设计。

H.323 服务器是本系统中最核心的服务器，该服务器承担了系统中音视频数据和控制信令的处理和传输。H.323 服务器控制着终端系统，终端系统则在 H.323 服务器的控制下完成系统需求的功能。本文下一章将详细介绍在 H.323 服务器控制下的终端系统的设计与实现。

## 5 终端设计与实现

本远程实时教学系统分为终端、H.323 服务器、网页服务器及数据库服务器几部分。终端系统是用户直接面对的部分，终端的设计和实现直接影响本系统的应用效果及用户评价，因此终端是本系统中最重要的组成部分，也是本文着重阐述的部分。由于远程实时教学系统的应用要求，终端又分为教师端及学生端两种。

由于本文着重描述 H.323 协议与远程实时教学系统的结合，因此本章着重介绍 H.323 层信令处理功能和音视频处理能的实现。终端系统采用 OpenH323 协议栈实现 H.323 层的大部分功能，因此本章将在 OpenH323 协议栈的基础上作介绍。

### 5.1 终端处理流程设计

终端系统是一个完整的体系，它融合资源层和 H.323 层构成了远程实时教学系统。终端系统分为教师端和学生端，教师端与学生端 H.323 层的实现比较相似，但在资源层的处理上，教师端与学生端有很大的不同。教师端必须处理教学资源、控制教学进度，因此学生端的功能是教师端的一个子集，因此，此节将以教师端为例介绍终端基本处理流程。教师端工作流程如图 5-1 所示。

系统在开启之初必须进行初始化，与 MCU 的处理方式相同，终端系统也同样需要导入存放在指定位置的配置文件，该文件包括了终端系统的 IP 地址、音视频配置信息等等系统运行必需的参数。

系统初始化之后，用户可以点击界面登陆按钮，填写用户名、密码和数据库服务器地址登陆。登陆时系统将使用前文所述的 Web Service 技术从数据库获取登陆许可。终端系统向数据库服务器的 Web Service 服务传送登陆所需的用户名和密码。数据库服务器在本地查找数据库，返回用户是否允许登陆的信息。若不允许登陆则返回初始界面。

若数据库服务器允许登陆，则终端系统开始向数据库服务器索要课程列表，这个过程同样使用 Web Service 技术实现。数据库返回用户可选的课程及课程的相关信息，例如课程名、授课教师、该门课程使用的 MCU 地址及资源服务器地址等。

用户一旦选择一门课程，则终端 H.323 层向该课程的 MCU 地址发起连接，MCU 服务器接受连接后，课程在 H.323 层上已经开通。此时学生端已可以进入课堂学习。由于教师端必须在进入课程之前处理好教学资源，因此，此时教师端的资源层限制了教师进一步使用课程的功能。

教师可以点击界面资源向导按钮，在资源向导中添加教学资源。该模块按照第三章所述的方法分别处理各种教学资源，将之转换成 HTML 格式，并生成课程包。在资源向导处理完成时，系统自动使用 FTP 将生成的课程包上传到该资源服务器供学生下载浏览。

处理完资源后，教师只需点击开始授课，便可以使用全部的课程资源。教师可以点击教学资源列表切换教学资源，教师端的 H.323 层将通过 MCU 向所有学生发送此资源在资源服务器上的 URL 地址，学生端的 H.323 层接收到此地址后将通知其资源层获取此 URL 上的课件资源。同时，教师端还可以通过发送各种自定义控制信令控制学生的发言情况...

在教学过程完毕后，教师只需点击停止授课按钮即可停止课程。此时 H.323 层并未断开连接，系统只是在资源层上停止了课程进行，此状态下，教师只需重新点击开始授课即可重新授课。

若教师希望完全退出该课程，则可点击退出课程按钮，此时 H.323 层与 MCU 断开连接，若教师希望重新开始课程，则需要重新与服务器建立连接。

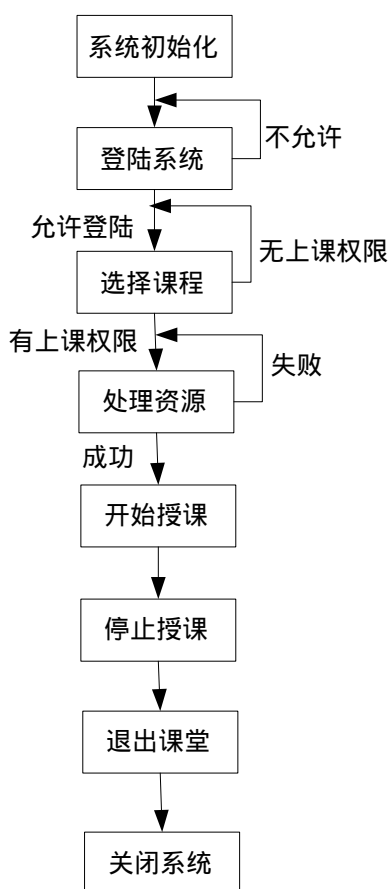


图 5-1 教师端工作流程图

## 5.2 终端信令处理功能实现

终端与 MCU 一样，都是使用用户输入通道传递自定义信令，因此也是适用的 OnUserInputString 接口处理接受的自定义信令。自定义信令的基本接收处理流程如图 5-2 所示。

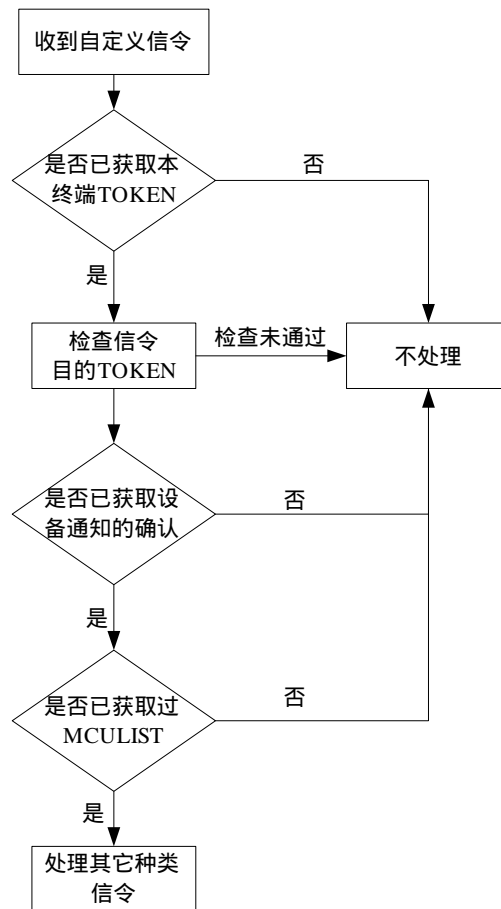


图 5-2 终端系统接受信令流程

前文已经介绍过终端初始化时终端需要进行自定义信令流程，如图 3-9 所示，终端需向 MCU 发送 GIVETOKEN、REPORT\_MY\_DEV 及 GETMCULIST 三个信令以完成初始化过程，因此信令的接收流程也必须判断这些信令是否已被回复。

由于所有终端都是使用 TOKEN 唯一标识，在收到信令时需要对 TOKEN 进行判断。但由于终端最初无法了解本地 TOKEN，故在从 MCU 接收到 GIVETOKEN 信令之前，终端不应也无法处理其它种类的自定义信令。所以如图 5-2 所示，在终端收到信令后，首先就要判断本地是否已获得 TOKEN，若为获得则不处理该信令。

终端获得本地 TOKEN 后可以开始处理信令，为了不处理误发给自己的信令，终端应对所有信令检查目的 TOKEN。有效的目的 TOKEN 只有两种，若目的 TOKEN



与本地 TOKEN 一致，则表示该信令是发向自己的，该信令有效；若目的 TOKEN 字段填写的符号“\*”，且源 TOKEN 字段填写的是 MCU，则表示此信令是 MCU 群发给所有终端的控制信令，该信令有效。除此之外，其他信令都为无效信令，系统不作处理。

检查完目的 TOKEN，按图 3-9 所示自定义信令流程，终端下一步应检查是否获得 MCU 对于 REPORT\_MY\_DEV 信令得确认信令，若未获得则不再处理该信令，系统一直 MCU 等待该信令的确认。

获得了确认信息之后，终端必须创建本课程用户列表，因此需要等待 MCU 对于 GETMCULIST 信令的回复，即 MCULIST 信令，若接收到该信令则可建立用户列表，课程的初始化完毕，终端可以进一步处理其它各种信令。

从 MCU 接收到信令后流程处理部分代码如表 5-1 所示。该代码与图 5-2 所述流程一致。

表 5-1 流程处理部分代码

```
void MyH323Connection::OnUserInputString(const PString & value)
{
    CmdClient cmd;
    if(cmd.ParseCmdPDU(value) == TRUE)
    {
        if(cmd.GetDesToken() == this->myEndpoint.myToken || cmd.GetDesToken() == "")
        {
            //如果没有得到过自己的 token，则不解析其他的命令知道得到自己的 token
            if (getToken == FALSE)
            {
                if (cmd.GetCmdType() == cmd.CTRL_MCU_GIVETOKEN)
                {
                    this->myEndpoint.myToken = cmd.GetMyToken();
                    getToken = TRUE;
                    this->myEndpoint.waitForToken.Signal();
                    return;
                }
            }
        }
        else
        {
            if(getReportAck == FALSE)
            {
                //收到报告硬件信息的 mcu 回复
                if(cmd.GetCmdType() == cmd.ACK_CLIENT_REPORTMYDEVICE)
                {
                    if(cmd.GetSouToken() == "mcu")
                    {
                        this->myEndpoint.waitForReportDevice.Signal();
                        this->getReportAck = TRUE;
                        if(myInfo == NULL)
                        {

```

```

        return;
    }
    else
    {
        BOOL VideoOK;
        BOOL AudioOK;
        cmd.GetVideoEnable()?VideoOK = TRUE : VideoOK = FALSE;
        cmd.GetAudioEnable()?AudioOK = TRUE : AudioOK = FALSE;
    }
}
return;
}
}
else
{
    if(getMcuList == FALSE)
    {
        if(cmd.GetCmdType() == cmd.CTRL_MCU_MCULIST)
        {
            //收到 MCULIST 后的处理
            .....
            return;
        }
    }
    else
    {
        switch(cmd.GetCmdType())
        {
            //收到聊天信息的处理
            case cmd.CHAT :
                return;
            //收到 http 地址的处理
            case cmd.HTTP_ADDRESS :
                return;
            //收到 ftp 地址的处理
            case cmd.FTP_ADDRESS :
                return;
            //收到学生请求发言的处理
            case cmd.CTRL_CLIENT_STUREQUESTSPEAK :
                return;
            //收到 McuList 信息的处理
            case cmd.CTRL_MCU_MCULIST :
                return;
            //收到 mcu 有成员加入会议的信息
            case cmd.CTRL_MCU_PERSONENTER :
                return;
            //收到 mcu 有成员离开会议的信息
            case cmd.CTRL_MCU_PERSONLEAVE :
                return;
            //收到 MCU 确认老师要求的学生已发言信息
            case cmd.ACK_CLIENT_STURESPONSESPEAK :

```



分这两种地址。

终端收到 STU\_REQ\_SPEAK 信令，表示有终端要求发言，故应修改该用户状态为申请发言，并通知资源层更新该终端状态。

终端收到 TEA\_REQ\_SPEAK 信令，表示教师要求本终端发言，终端应打开音视频通道，并向 MCU 回送 STU\_RESPONSE\_SPEAK 的确认信令，表示终端已发言，若失败则发送 STU\_RESPONSE\_SPEAK 的拒绝信令。

终端收到 STU\_RESPONSE\_SPEAK 的确认\拒绝信令，应修改对应用户状态为正在发言\未发言，并通知资源层更新该用户的状态。

终端收到 TEA\_RESPONSE\_SPEAK 确认信令，表示教师已允许本终端发言，故终端应打开音视频通道并发送 STU\_RESPONSE\_SPEAK 确认信令，失败则发送拒绝信令。

终端收到 TEA\_RESPONSE\_SPEAK 拒绝信令，表示教师已拒绝或停止本终端发言，暂停音视频通道并发送 STU\_RESPONSE\_SPEAK 拒绝信令。

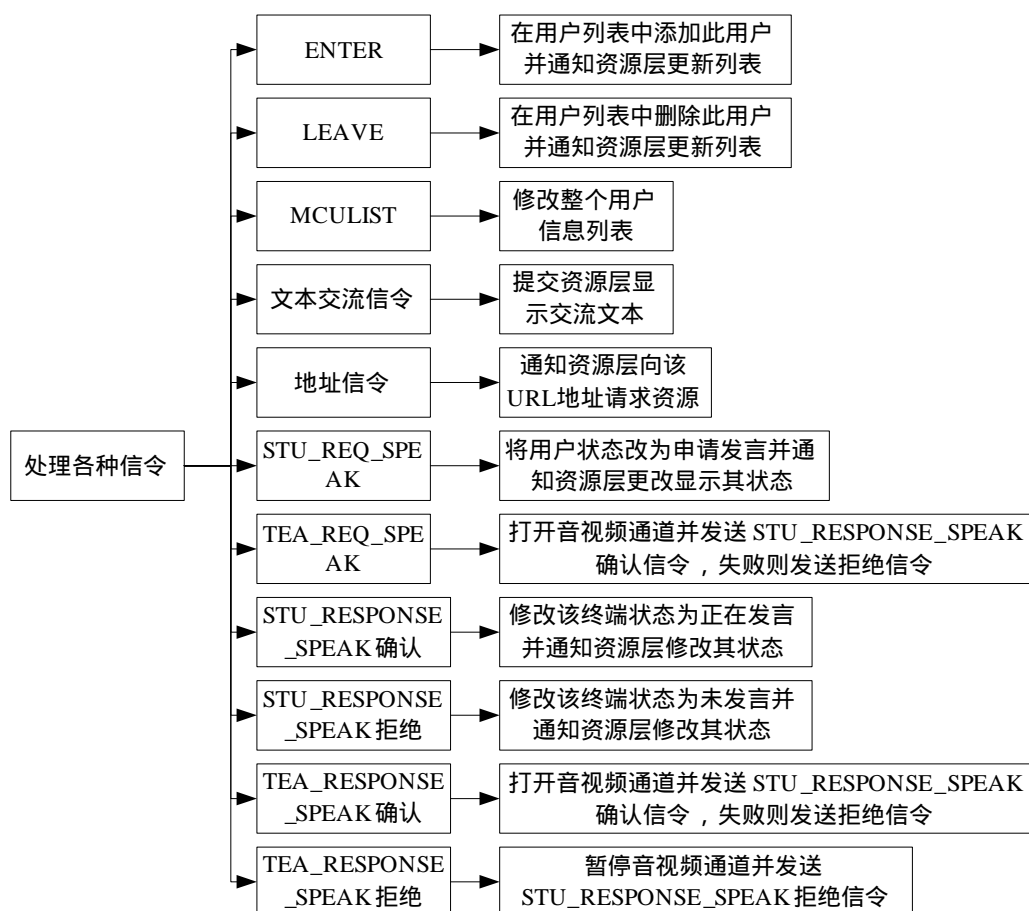


图 5-3 终端对于多种自定义信令的处理

### 5.3 终端音视频处理功能实现

终端的音视频模块负责音频和视频的采集、处理、发送、接收和显示。由于本系统采用了两层框架结构，导致本系统终端音视频模块与 OpenH323 音视频模块不同，本系统在这个模块上作了较新颖的设计。

第三章介绍过，两层框架结构中资源层使用托管代码 C#实现，而 H.323 层使用非托管代码 C++实现。这样的设计不仅是系统与 H.323 通信协议结合，还解决了现代远程实时教学系统中普遍存在的问题。但是这样的设计也给具体实现提出了更高的要求。

托管代码与非托管代码在内存管理方面的机制完全不同，因此长时间在两层之间传输大数据量的数据会使内存管理者频繁更替，除了在处理速率方面会受到影响，更易引起内存管理的混乱，导致系统的不稳定。使用非托管代码编写 H.323 层从网络获取视频数据，需要使用托管代码编写的上层界面中视频显示模块显示，其中必然涉及到托管代码和非托管代码之间大数据量的传输问题。

在这个问题中，本系统对使用内核模式解决用户模式出现的问题作出了尝试。解决方式如下：编写非托管代码接口，由用户模式运行的非托管代码使用 DirectShow Filter 机制向内核模式运行的虚拟摄像头写入数据；用户模式的托管代码则利用 DirectShow 提供的托管代码接口直接捕获虚拟摄像头中的数据显示。使用这种方式后，非托管代码和托管代码之间不再需要大数据量的传输，也不会再出现内存管理方面的问题。

为了提高系统的稳定性，Windows 操作系统对硬件操作进行了隔离，应用程序一般不能直接访问硬件。这个问题可以用 DirectShow 技术解决<sup>[xli]</sup>。

DirectShow 是微软公司提供的一套在 Windows 平台上进行流媒体处理的开发包。它基于模块化，每个功能模块都采取 COM 组件方式，称为 Filter<sup>[xlii]</sup>。DirectShow Filter 工作在用户模式，而硬件工作在内核模式。为实现交互，DirectShow 为采用 WDM 驱动程序的硬件设计了 KsProxy Filter(如图 5-4 所示)<sup>[xliii]</sup>。这种 Filter 跟其它普通的 DirectShow Filter(例如，自定义 Filter)处于同一个级别。它通过 Stream Class 控制硬件的驱动程序 minidriver（在系统中对应虚拟摄像头驱动程序）<sup>[xliv]</sup>。

在本系统中，DirectShow 技术不仅用于摄像头的捕获显示，也应用于向虚拟摄像头驱动中传送数据，由于 DirectShow 可以自定义 Filter，而自定义的 Filter 可以留出写入数据的接口，因此，在本系统中实现由应用程序向驱动程序写入数据的原理如图 5-4 所示。准备传输数据时，调用 DirectShow 提供的 API 函数将自定义 Filter

和 Ksproxy Filter 连接起来；传输数据时，调用自定义 Filter 的数据接口，向自定义 Filter 中写入视频数据，视频数据将从自定义 Filter 传至 Ksproxy Filter。绑定了虚拟摄像头设备的摄像头捕获程序，使用 DirectShow 机制，通过 Ksproxy Filter 从虚拟摄像头驱动中获取数据，既可得到从自定义 Filter 传入的数据。对于处于用户模式的摄像头捕获程序而言，即捕获了由用户模式传入内核模式的数据。

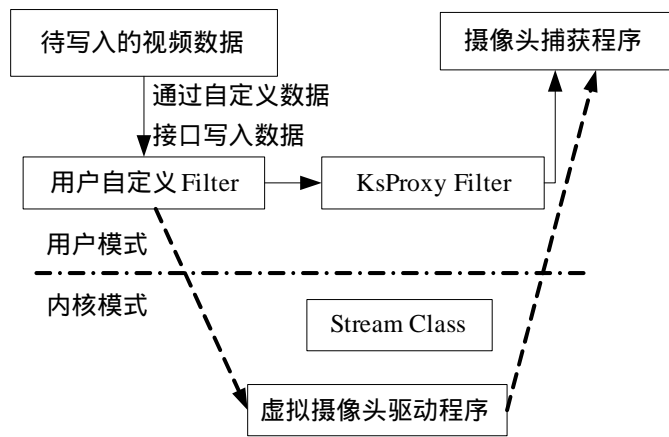


图 5-4 应用程序向硬件驱动写入数据原理图

在自定义 Filter 中可以加入视频处理模块，如图 5-4 所示，摄像头捕获程序(远程教学系统程序)将获取经过处理后的视频图像，在系统运行过程中，可随意更换视频处理形式，体现了较好的封装性、复用性和模块性独立。

虚拟摄像头在系统中应用的实现比较复杂，主要分为数据传输的准备、视频数据处理及数据传输的清理这几部分。

#### 1. 数据传输的准备和清理

在数据传输时，需要通过自定义 Filter 向虚拟摄像头驱动中写入数据，因此数据传输前，应使用 DirectShow 机制完成初始化工作：

- 1) 初始化 DirectShow 提供的 COM 接口
- 2) 初始化虚拟摄像头，将自定义 Filter 与虚拟摄像头驱动连接起来
- 3) 对自定义 Filter 的初始化，主要包括自定义 Filter 中默认图片的初始化以及一些视频处理功能的初始化工作。

在具体应用中根据功能需求的不同，上述第三步可以忽略。

在数据传输结束时需要相应的释放占用的资源。

#### 2. 视频数据处理

在 OpenH323 协议栈中，开始传输视频数据时，H323Channel 的子类 H323UnidirectionalChannel 调用 Start 方法，启动了一个后台线程

H323LogicalChannelThread，这个线程用于接收和发送媒体数据。该线程调用 H323\_RTPChannel 中的发送(Transmit)和接收(Receive)方法来传递媒体数据<sup>[xiv]</sup>。

发送媒体数据时使用的是 H323\_RTPChannel 类中 Transmit 方法：首先调用 H323\_RTPChannel 类中的 Read 方法，从 Codec 中读出一帧媒体数据，再调用 WriteData 将该数据包封装成标准的 RTP 分组，写入 UDP 套接字传送到对端主机。

接收媒体数据时使用的是 H323\_RTPChannel 类中的 Receive 方法：首先调用 H323\_RTPChannel 类中的 ReadBufferedData 方法从网络读取一帧媒体数据，再调用 Write 方法将该数据写入 Codec。

H.323 中读取或写入数据都是通过 Codec 完成，Codec 是编解码器，与 PChannel 绑定。PChannel 是定义了 IO 通道的抽象类。PVideoChannel 是 PChannel 的子类，用于从指定 IO 通道中读取或写入视频数据。视频 Codec 绑定的即是 PVideoChannel。

当发送视频数据时，如图 5-5 所示，H323\_RTPChannel 类中的 Read 方法，从视频 Codec 中读出一帧视频数据，视频 Codec 通过 PVideoChannel 的 Read 方法从绑定的视频输入设备中读入视频数据至缓冲区 buf，对该缓冲区中数据进行编码，然后返回给 H323\_RTPChannel 类的 Read 方法。同样，接收视频数据时，如图 5-6 所示先使用视频 Codec 对发送给 Codec 的数据进行解码，然后存储到缓冲区 buf 中，PVideoChannel 类中的 Write 方法将存储在缓冲区 buf 中的数据写入绑定的视频输出设备显示。

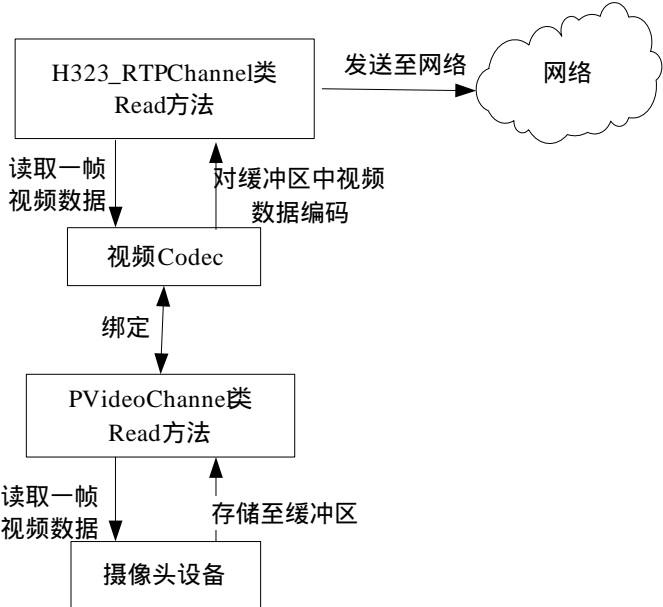


图 5-5 OpenH323 协议栈发送视频数据示意图

如上所述，PVideoChannel 与视频设备存在着直接的联系，从 IO 通道中读取或

向 IO 通道中写入视频数据都要通过 PVideoChannel 类，虚拟摄像头数据接口是 IO 通道的一种，因此需要在 PVideoChannel 类中对虚拟摄像头数据接口进行操作。

PVideoChannel 类中的 Read 方法是从 IO 设备中读取视频数据，因此，如图 5-6 所示，应将 Write 方法 buf 中的数据通过虚拟摄像头数据接口写入虚拟摄像头中，供 DirectShow 捕获显示网络接收的视频数据。

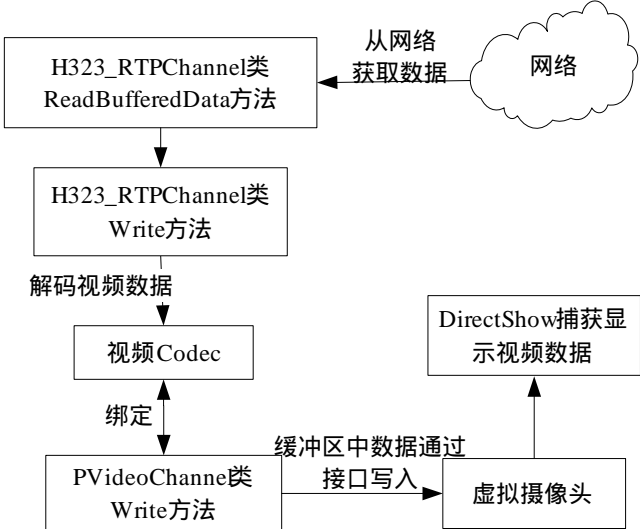


图 5-6 发送、接收视频数据时虚拟摄像头的应用

### 5.4 终端资源层与 H.323 层接口

由于终端系统同时拥有资源层与 H.323 层，而这两层分别使用 C#和 C++这两种截然不同的语言实现，因此层接的接口成为了一个重要的问题。

终端系统中的界面显示、流程控制及资源处理部分都是资源层的处理工作，而信令及音视频处理则是 H.323 层处理的工作。资源层接收用户操作，须向 H.323 层发送用户指令，H.323 层接收此命令即向网络发送相应的控制信令。当终端用户从网络上接收到某些控制信令、文本交互信令及地址信令时，都须向资源层提交相应数据，供资源层更新、显示。

资源层与 H.323 层间交互信息涉及到的自定义信令如图 5-3 所示，H.323 层从网络接收到 ENTER、LEAVE 及 MCULIST 信令时，都需要通知资源层更新当先显示的用户列表；当收到文本交流信令时，需要向资源层提交该文本数据供资源层显示；当收到地址信令时，需要通知资源层向该地址请求资源；当收到 STU\_REQ\_SPEAK、STU\_RESPONSE\_SPEAK 的确认\拒绝信令时，须通知资源层修改相应用户的状态为申请发言、正在发言或未发言；当学生端申请或停止发言时，



资源层须通知 H.323 层发送 STU\_REQ\_SPEAK 或 STU\_RESPONSE\_SPEAK 的拒绝信令；当教师端要求或停止发言时，资源层须通知 H.323 层发送 TEA\_REQ\_SPEAK 或 TEA\_RESPONSE\_SPEAK 的拒绝信令；当教师允许某人发言时，资源层须通知 H.323 层向该终端发送 TEA\_RESPONSE\_SPEAK 的确认信令。

由于本系统中资源层与 H.323 层分别采用不同的语言实现，因此函数接口在两层中均要定义，且参数的数据类型要求一致。

H.323 层发送信令函数定义如表 5-2 所示，这些都是在 H.323 层中定义的接口函数，因此都是使用 C++ 语言定义。

表 5-2 H.323 层发送信令接口函数代码

```
extern "C" ENDPOINT_API BOOL SendChatPDU(MyH323Connection * connection,char * desToken,char * chatString);
extern "C" ENDPOINT_API BOOL SendHttpPDU(MyH323Connection * connection,char * desToken,char * httpAddress,char * message);
extern "C" ENDPOINT_API BOOL SendFtpPDU(MyH323Connection * connection,char * desToken,char * ftpAddress,char * message);
extern "C" ENDPOINT_API BOOL SendTeaRequestSpeakPDU(MyH323Connection * connection,char * desToken,bool videoEnable,bool audioEnable,char * message);
extern "C" ENDPOINT_API BOOL SendStuRequestSpeakPDU(MyH323Connection * connection,bool videoEnable,bool audioEnable,char * message);
extern "C" ENDPOINT_API BOOL SendNakStuResponseSpeakPDU(MyH323Connection * connection,bool videoEnable,bool audioEnable,char * message);
extern "C" ENDPOINT_API BOOL SendAckTeaResponseSpeak(MyH323Connection * connection,char * desToken,bool videoEnable,bool audioEnable,char * message);
extern "C" ENDPOINT_API BOOL SendNakTeaResponseSpeak(MyH323Connection * connection,char * desToken,bool videoEnable,bool audioEnable,char * message);
```

资源层向 H.323 层发送信令函数定义如表 5-3 所示，这些都是在资源层中定义的函数接口，因此都是使用 C#语言定义的。

表 5-3 资源层向 H.323 层发送信令接口函数代码

```
[DllImport("EndPoint.dll", CharSet = CharSet.Ansi)]
public static extern int SendHttpPDU(IntPtr connection,string desToken,string httpAddress,string message);
[DllImport("EndPoint.dll", CharSet = CharSet.Ansi)]
public static extern int SendFtpPDU(IntPtr connection,string desToken,string ftpAddress,string message);
[DllImport("EndPoint.dll", CharSet = CharSet.Ansi)]
public static extern int SendChatPDU(IntPtr connection,string desTokenPtr,string chatStringPtr)
[DllImport("EndPoint.dll", CharSet = CharSet.Ansi)]
public static extern int SendTeaRequestSpeakPDU(IntPtr connection,string desTokenPtr,int videoEnable,int audioEnable,string message);
[DllImport("EndPoint.dll", CharSet = CharSet.Ansi)]
public static extern int SendStuRequestSpeakPDU(IntPtr connection,int videoEnable,int audioEnable,string message);
[DllImport("EndPoint.dll", CharSet = CharSet.Ansi)]
public static extern int SendNakStuResponseSpeakPDU(IntPtr connection,string desTokenPtr,int videoEnable,int audioEnable,string message);
```

```
[DllImport("EndPoint.dll", CharSet = CharSet.Ansi)]
public static extern int SendAckTeaResponseSpeak(IntPtr connection, string desTokenPtr, int
videoEnable, int audioEnable, string message);
[DllImport("EndPoint.dll", CharSet = CharSet.Ansi)]
public static extern int SendNakTeaResponseSpeak(IntPtr connection, string desTokenPtr, int
videoEnable, int audioEnable, string message);
```

H.323 层接收信令并通知资源层的接口函数定义如表 5-4 所示。这些都是在 H.323 层中定义的接口函数，因此都是使用 C++ 语言定义的。前三个是教学资源显示、文本交流及 FTP 地址接口，后三个是人员添加删除及状态修改接口。这些接口的参数是资源层中定义的 C# 函数的指针，在 C# 中调用 C++ 中的特定函数将 C# 中处理函数的指针存储到 C++ 的指定指针中，在 C++ 中即可调用该函数指针实现 C++ 调用 C# 的函数接口。

表 5-4 H.323 层通知资源层接口函数代码

```
extern "C" ENDPOINT_API BOOL SetHtmlActive(HtmlActive * func);
extern "C" ENDPOINT_API BOOL SetChatShow(ChatShow * func);
extern "C" ENDPOINT_API BOOL SetFtpActive(FtpActive * func);
extern "C" ENDPOINT_API BOOL SetRemovePerson(RemovePerson * func);
extern "C" ENDPOINT_API BOOL SetRefreshTreeView( RefreshTreeView * func);
extern "C" ENDPOINT_API BOOL SetClearList(ClearList * func);
```

在系统实现中还有一些简单的接口函数，主要实现对会议初始化、结束、音视频通道的维护等功能，这些函数只是通知函数，不需要传递任何参数信息，因此不再详述。

## 5.5 本章小结

本章第一节首先介绍了终端部分的基本处理流程。对于不同的终端种类，该流程分为教师端流程和学生端流程，本节以教师端为例介绍了教师端流程，包括了从系统初始化到系统关闭所经历的所有过程。这部分是资源层完成的工作之一。然后在第二节和第三节着重介绍了终端中 H.323 层的详细设计，第二节介绍信令模块的设计，主要介绍了终端的自定义信令接收过程及对于自定义信令的处理。第三节介绍了基于虚拟摄像头的音视频模块<sup>[xlvi]</sup>的设计。这两节涉及到的 Openh323 协议栈的开发技术，使用 Openh323<sup>[xlvii]</sup>及 Pwlib<sup>[xlvi]</sup>库，这也是本系统的研究重点。这些属于 H.323 层。第四节介绍了资源层与 H.323 层之间的接口，列举了部分接口函数。

终端是系统中最重要的一部分，它给用户呈现了最直观的使用效果。终端系统的是否能令用户满意是评价系统是否成功的一个重要的标准。

## 6 系统测试与应用

现阶段本系统已经结束了设计和开发的工作，进入测试和部署的部分。本章将展示部分的系统界面图。结合界面图，本章将分类介绍本系统已实现的各部分功能，这些功能均是前文所述需求的实现。另外，由于本系统已进入测试和部署阶段，本章还将具体介绍本系统的现实应用，包括系统测试结果，部署情况等结果分析。

### 6.1 系统整体效果

本系统经过一年的设计和编码现已完成。终端系统界面图如图 6-1 所示。



图 6-1 系统界面图

#### 6.1.1 视频交互区

左侧四个方形区域是视频区域。此区域显示四个用户（一个教师 and 三个学生）的视频图像。

第一个图像永远是教师视频，其后是发言学生的视频。系统界面图中右两路视频，一路教师视频，一路学生视频，这两路视频对所有用户可见。

6.1.2 资源浏览区

中上区域是资源浏览区域，此区域负责教学资源的展示工作。

如图 6-2 所示，显示的是教师向学生展示的课件资源。本系统可与各种常用办公软件协同作战，包括 Word、Power Point 等。教师可将 Word 文档、Power Point 文档、html 网页、Flash、媒体文件及各种格式的图片处理成课件，通过系统在授课讲述的同时进行播放，给学生最直接的知识输入和认识。

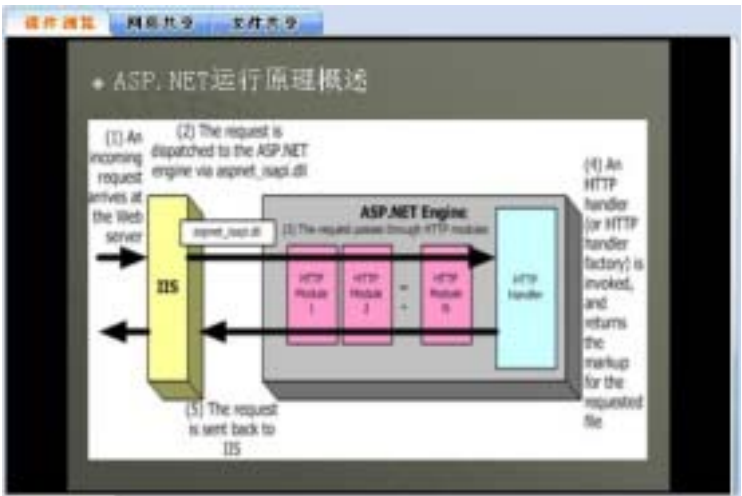


图 6-2 课件浏览窗口

当然教师浏览网页时，如图 6-3 所示，此区域将切换成网页共享展示该网页。通过使用该功能，教师可向学生展示网页，丰富了教学内容；同时，学生也可使用该功能浏览网页，查找与课程有关的资料，使学习更加多样化。



图 6-3 网页共享窗口

在使用过程中，如图 6-4 所示，用户还可点击文件共享按钮启动文件共享功能。

通过使用该功能，教师上传教学资料，学生直接下载即可得到，学生即使身在远方，也可以收到老师的教学资料；同时，学生亦可以上传学习资料（作业等）供教师或其他学生下载，即使没有身在教室，也可以提交作业。该功能为教师与学生，学生与学生之间架起了沟通的桥梁。



图 6-4 文件共享窗口

6.1.3 文本交流区

中下区域是文本交流窗口，如图 6-5 所示，用户在下方框中输入文字，使用中间按钮条中按钮调整文字格式，之后可以点击发送按钮发送给其他终端。用户可通过下拉菜单选择所有人或单个用户名来选择文本发送对象，可向所有终端发送，也可以只向某个终端发送。上方框显示终端接收到的文字。多人文字发言可配加多种动作、变换文字字体及颜色。文字发言框有自动记忆自动填充功能，发言可保存历史记录，可导入已有记录发送，并且支持“悄悄话”功能。

通过该功能，使教师和学生之间，学生与学生之间进行文本方式的交流。在不占用宝贵的网络带宽的前提下，实现课堂交互的实时性，进一步保证了教学的质量。

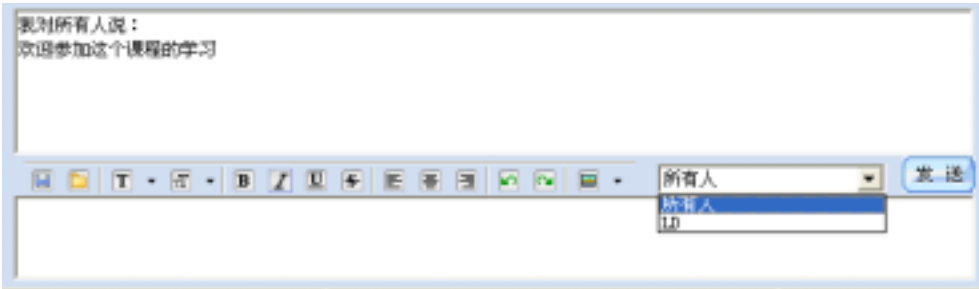


图 6-5 文本交流窗口

#### 6.1.4 用户列表区

右上方区域是用户列表区域，前文所述资源层的用户列表指的即此列表。此列表将所有用户分状态显现。用户状态分别为教师、正在发言、申请发言和普通用户状态。

学生发言过程如图 6-6 所示。教师可右击某用户普通用户头像要求其发言，右击申请发言用户头像可同意其发言或拒绝其发言，右击正在发言用户头像，可停止其发言；学生右击自己头像要求发言，也可停止发言。用户还可右击任意用户头像察看该用户的公开身份信息。



图 6-6 学生发言过程

#### 6.1.5 课件资源区

右下方区域为课件资源区域，该区域为教师端所独有。教师端处理完资源后，在此区域显示所有资源名。这些资源将根据不同的资源类别（如 Word 文档、PowerPoint 文档等）分类列出。教师可通过点击某一资源类别的任一个资源的名称以向学生展示该资源。学生端将通过前文所述的方式获得此资源并浏览。

以上界面图包含了系统用户所能使用的所有功能，这也是 3.1.1 节所述的系统需求的实现。这些功能的具体实现都在前文讨论过。

### 6.2 系统测试及现实应用

上节所述的功能可以满足现代远程教学系统的大部分需求。整个系统的开发和调试都已基本结束。本系统已在某大学工程中心部署测试，系统在支持约 40 人的小型网络中已持续运行了一个多月，并取得了令人满意的效果。

本系统主要数据处理主要在终端和服务端这两部分，因此主要对这两部分进行了集中的测试。测试过程中主要对以下部分功能进行了集中的测试：

#### 1. 课件列表

教师可将上课所需的图片、各种 office 文档、网页课件等多媒体资源装入课件列表，在上课时实现主控式的同步浏览。主要测试的是资源处理向导和资源上传部分的功能。测试要求各种系统兼容的资源格式都必须能经过资源向导的正确处理，并能正确的上传到资源服务器。

#### 2. 课件实时同步

系统能将课件自动压缩上传到资源服务器，教师和学生可以同步以网页的方式浏览。主要测试的是教师和学生同步浏览教学资源的功能。为了保证教学效果，在局域网内要求每个学生端每次都必须能根据教师的教学命令浏览到教学资源。由于实时性要求，教师点击课件资源后，学生必须在 1 秒钟内浏览到该资源。

#### 3. 音视频交互系统

教师能够把自己的视频广播到教室中，或者广播某个学生端的视频；在教师广播自己视频时，也可以监视某个学生的视频。教师能把自己的声音广播出去，同时可以允许某个学生说话，或者同时允许两个学生广播声音。教师端有学生举手状态的显示框，教师可以选择举手的学生，并允许发言。学生在被允许发言时，系统会弹出一对话框，提醒学生发言。这部分测试的是教师对学生发言的控制和课堂内音视频传输的质量，这部分是测试的重点。教师必须可以准确的控制学生的发言情况，所有音视频必须按正确的状态传输。

#### 4. 音视频延时

由于本系统为实时系统，因此终端间音视频数据必须能实时被传输。测试要求终端之间音视频延时不能超过 1 秒。

#### 5. 文本交流系统

在文字讨论区，教师、学生可以与全体人员或某个人进行文字交谈，当选择对一个人发送信息时，此课堂内其他成员则看不到。文字交互系统中还有关于老师操作信息的系统提示和教室公告，如教师允许某人发言、教师允许某人录制课件等。在局域网内要求文字数据每次都能正确的被传输。

#### 6. 文件共享

可以通过 FTP 共享一些文档资料。在局域网内要求文件的上传和下载都能正确的进行。

#### 7. 选择教室

支持多个教室同时上课，具有多课堂、帐号等管理功能。这是对服务器端课程管理功能的测试。要求服务器开启的课程终端可以进入，未开启的终端不可进入。

经过一个多月的系统测试，以上各功能都已达到测试要求。测试结果如下：

1. 各种课件资源均能被正确的处理及上传。
2. 教师的教学资源展示命令均能被正确的传输，学生端均能获得正确的教学资源，局域网内教师发出命令与学生获取资源之间延时不超过 0.5 秒。
3. 教师能准确的控制学生的发言情况。
4. 且局域网内音视频的最长延时不超过 0.5 秒。
5. 文字交流时所有文字信息（包括文字格式信息）均能正确的被传输。
6. 文件的上传和下载一般不会出现问題。
7. 服务器可以正确的控制多课程多账号。各个课程可同时运行，不互相干扰。

测试结果表明本系统稳定性能很好，运行时不会出现很大的问题，在音视频传输质量、网络带宽消耗及系统的稳定性方面均可满足教学需求。

本系统在局域网内已通过测试，并已进行了初步的部署。本系统实现了远程实时教学系统的基本功能，如音视频交互，教学资源浏览，文本交互，教学过程控制等。各种操作都有权限设置，例如学生发言控制只能是教师才能执行。如果有非法操作或错误操作，系统会提示用户哪里出了问题，真正达到人机交互。但本系统在大型网络中并未进行过测试，故并未得到相关的测试数据，仍有待进一步测试。

### 6.3 本章小结

本章首先展示了本系统的终端界面图，并根据界面图对终端实现的各个功能一一做了介绍；然后，在第二节中主要介绍了本系统目前的应用状况和有待进行的工作；最后，就系统的测试结果给出了结果分析。

本章是对本系统的现阶段实现状况的介绍部分。本系统虽已完成并取得了一定的成果，但仍有需要完善的地方，本系统也将继续改进，不断完善。



## 7 总结

### 7.1 研究成果

远程实时教学系统，是现代教育产业和通信技术融合的产物。目前，我国的远程教育体系正在不断地发展当中，本系统的实现对与国际标准结合的教学系统的发展有着一定的重要意义。本文基于系统实现进行的研究和做出的贡献如下：

第一，将通信标准与远程教育的体系相结合，提出了用 H.323 协议实现实时教学平台的方案，并给出了其技术实现过程。将通信标准的实现和远程教育系统的实现相结合，对远程教学系统进行了国际化的实践，为远程教学的发展做出了有益的尝试和推动。

第二，本系统提出了两层框架结构，一定程度上解决了现代远程实时教学系统中存在的问题，为现代远程实时教学系统开创了新的发展思路。

第三，本系统提出了将虚拟摄像头应用到远程教学系统的设计，对于利用内核模式解决用户模式问题的设计思路作了一定的研究和实践。为相关问题提供了参考和借鉴。

国外在远程教育系统和通信计算机的普及方面比国内进行的早，有很多成熟的远程教学系统的实现，这些都是值得今后学习和借鉴的。希望通过大家的共同努力不断的推动我国教育和通信的发展。

目前国内虽然已经有了“天地网”这样的大型实时教学系统，但不是所有的地方都需要卫星教学，实时教学系统正在小型化和普及。本系统可以作为有效的补充，但是还需要进一步的完善。同时，由于国内教育在通信标准化起步较晚，并且不是基于成熟的国际通信标准。因此，国内存在着各种各样的非标准的学习系统，但远程教育通信标准化的过程正被越来越多的重视。本系统给出了一个可参考的通信标准化实现，为共同的学习系统标准化的目标做出努力的尝试。

### 7.2 发展方向

本系统实现了教学平台的基本功能，保证了教学中师生交互的多媒体展现方式。同时实现了远程教学系统与国际通信标准的结合，能够和其他的符合该标准的网络设备协同工作，而且本文就此也给出了一定的具体实现。但是，本系统还有进一步需要完善的地方和研究的方向，包括以下几点：

第一，本系统在媒体流和教学资源同步方面仍存在欠缺。现在本系统仍然是将教学资源信息与媒体流分开传输，分开显示。若能够将两者间建立联系，保持两者间的同步，这本系统将更为完善。

第二，在系统的技术方面，本系统采用 H.323 协议进行多媒体通信的实现还需进一步扩展。一方面，在 H.323 协议与多种类型网络的兼容方向上还有很大的可挖掘的地方；另一方面，H.323 协议穿越网络地址转换( Network Address Transfer ,NAT )的实现以及 H.323 协议同 SIP 协议和 H.248 协议的互通的实现上还需要进一步工作，以保证系统在不同网络环境上的使用。

第三，由于本系统并没有考虑在超大范围内使用，所以没有加入关守与网关等设备的支持，但若系统需要大量应用，关守和网关的加入将必不可少，因此这方面仍然是有待扩展的方面。

我国近年来虽然在远程教育领域发展迅速，但与国外相比仍存在一定的差距，国外的成熟系统不仅有通信标准的依托，其系统的设计和处理效率方面也比我国领先不少，这些都是值得借鉴的方面。希望本系统这些方面的尝试能够为我国远程教育技术做出贡献。

## 致 谢

三年的硕士生活即将结束，本篇论文包含着我这三年的个人的总结，包含着各位师长和同学的帮助和教导，包含着家长的哺育和国家的期盼。借本文脱稿之际，在此我想向所有在我学习期间帮助过我成长的人表达我最诚挚的谢意。

无师难以成学，首先我要感谢我的导师杜旭副教授和吴砥讲师。传道、授业、解惑，杜老师和吴老师用自己丰富的知识和开阔的眼界，不断对我进行教育，打开我的视野，为我创造了很好的学习研究环境，使我受益匪浅。

还要感谢实验室的各位老师，他们在硕士学习期间给予我的帮助，像一个领路人，指导我学习的方方面面。

感谢严鹤博士，从我初入项目组至今，不断给我学术上的指导和生活上的帮助。在学术上严博士是一位严师，没有他的指导，我在学术上无法达到今天的高度；在生活上他是一位益友，他帮助我融入实验室这个大家庭。在此对他致以诚挚的谢意。

感谢马寅秋、胡小莉、徐玮、董毅、胡鑫和白红梅这几位同学，我们共同在项目中战斗着、生活着、成长着，完成了系统的实现，创造了一个又一个成绩。这是我论文得以写成的基础。

感谢魏东生、项杰、姜宇清、李德刚、邓飞这几位师兄。他们乐于分享自己的硕士学习和项目开发经验，是益友也是良师。

感谢熊琦、田好美、王媛、陈飞、龚后山这几位同窗。共同的学习环境，共同的学习经历，使我们相互学习、共同进步，结成了深厚的友谊，愿他们在以后的人生道路上越走越辉煌。

感谢万萍、侯婷以及实验室里其他所有成员的积极配合和努力工作，感谢大家在学习、工作中给予我的支持与帮助。

感谢我的父母多年以来对我的培养和教育，十年树木，我会用自己最大的能量回报社会，以报养育之恩。

借此机会在这里，对所有在我攻读硕士学位期间给予我热情支持和帮助的老师 and 同学致以最诚挚的谢意！

袁聃

2007 年 1 月 20 日

## 参考文献

- [i] 中国互联网网络信息中心. CNNIC. [Online] <http://www.cnnic.net.cn/>. Last Visited 2007.1
- [ii] 侯专政. 多媒体技术对远程教育的影响. 长沙通信职业技术学院学报, 2004, 3(02): 78~80
- [iii] 周俊, 冯军焕, 李有梅. 高校网络学堂建设与管理. 西南交通大学学报, 2006.12, 7(6): 23~27
- [iv] 王晶, 史东林, 孙胜娟. 远程教育在现代化教学中的应用. 河北体育学院学报, 2006, 20(02): 93~94
- [v] Korpi. M, Kumar. V. Supplementary services in the H.323 IP telephony network. Communications Magazine, 1999, 37: 118~125
- [vi] E. A. Thrush, N. E. Young, Hith. Processing putting courses on the web. Tech. Commun, 1999, 8(1): 49~59
- [vii] 戴国梁. 远程教育技术模式及启示[J]云南电大学报, 2004, 6(01): 21~23
- [viii] 沈中南, 史元春. 现代远程教育技术规范简介[J]计算机工程与应用, 2003, (05): 66~69
- [ix] 张俊霞. 现代远程教育的技术手段及其实现. 河南教育学院学报, 2003, 22(01): 134~135
- [x] 高春鸣, 丁一坚, 兰秋军. 远程教学平台的研究. 长沙电力学院学报, 2001, 16(2): 19~22
- [xi] 虞永超. H.323 标准入门. 现代电信科技, 1998, (12): 15~19
- [xii] 吴晓红, 黄永峰. H. 323 标准下会议系统关键技术分析. 数据通信, 2006, (5): 11~14
- [xiii] 姜宇清. 基于 H.323 的标准化实时教学平台的设计与实现:[硕士学位论文]. 保存地点: 华中科技大学图书馆, 2006.
- [xiv] 张永光. 基于 H.323 协议的视频会议系统研究. 视频技术应用与工程, 2006, (8): 128~130
- [xv] ITU-T. ITU-T Recommendation H.323 Packet-based multimedia communications systems, 2000.11
- [xvi] Nakamura. K, Funaki. K. An Improvement of G.711 PLC Using Sinusoidal model. Computer as a Tool, 2005, 2: 1670~1673

- [xvii] Niranjana Shetty , Jerry D. Gibson. Improving the Robustness of the G.722 Wideband Speech Codec to Packet Losses for Voice Over Wlans. *Acoustics, Speech and Signal Processing* , 2006 , 5 : 365~368
- [xviii] Wachtler. A. , Hoffmann. R. Detection of synchronisation errors for the G.728 decoder. *Telecommunications Symposium* , 1998 , 2 : 454~457
- [xix] Salami. R , Laflamme. C. , Bessette. B. Description of ITU-T Recommendation G.729 Annex A: reduced complexity 8 kbit/s CS-ACELP codec. *Acoustics, Speech, and Signal Processing* , 1997 , 2 : 775~778
- [xx] 张学武, 杨学星, 江冰. 基于 H.263 的视频编码、解码的研究及软件实现[J] *计算机工程与设计* , 2005 , 26(09) : 2491~2493
- [xxi] 洪小富, 陈晓锋, 张国焯. 基于 H.323 协议的企业级 VoIP 语音网关的设计和实现. 2003 , 29(11) : 132~134
- [xxii] 张剑寅. 基于嵌入式系统的 H.323 协议网守的研究. *计算机工程* , 2003 , 29(7) : 135~136
- [xxiii] 朱建新, 周毅. H.323 体系结构的一个改进模型. *计算机工程* , 2003 , 29(2) : 258~268
- [xxiv] Cheng-Yue Chang , Ming-Syan Chen , Pai-Han Huang. An H.323 gatekeeper prototype: design, implementation, and performance analysis. *Multimedia* , 2004 , 6(6)
- [xxv] ITU-T. ITU-T Recommendation H.225.0 Call Signaling protocols and media packet for packet based network
- [xxvi] ITU-T. ITU-T Recommendation H.245 Control protocol for multimedia communication , 2001.7
- [xxvii] 孙晓慧. 多媒体通信控制协议 H.245 及其关键技术. *无线电工程* , 2003 , 33(9) : 48~50
- [xxviii] 张卫, 沈富可. 基于低速率线路的多媒体通信控制协议 H.245 及其实现. *计算机工程* , 1998 , (6) : 21~25
- [xxix] 廖湘华, 李治柱. 基于 IP 的虚拟教室交互模式剖析与实现. *计算机仿真* , 2005 , 22(5) : 239~243
- [xxx] 王子荣, 杨贯中, 胡峰松. 基于 IP 的多媒体网络教学系统的研究与开发. *湖南大学大学学报* , 2000 , 27(6) : 96~102
- [xxxi] Equivalence Pty Ltd. Openh323 Project. [Online] <http://www.openh323.org/>.

Last Visited 2006.6

- [xxxii] Feng Liu , Gesan Wang , Li Li. Web Service for Distributed Communication Systems. Service Operations and Logistics, and Informatics , 2006 : 1030~1035
- [xxxiii] 罗胜涛. 基于 Web Services 的统一权限系统. 计算机工程 , 2006 , 32(24) : 34~36
- [xxxiv] 杨兴凯 , 刘宏. 基于 Web Services 跨平台存取数据库[J]. 电脑学习 , 2002 , (06)
- [xxxv] 齐小谦. 如何运用 Openh323 协议栈来开发 VOIP 软件. 无线电通信技术 , 2004 , 20(6) : 55~57
- [xxxvi] Friedland. G , Pauls. K. Architecting multimedia environments for teaching. Computer , 2005 , 38(6) : 57~64
- [xxxvii] 刘瑞芳 , 曾晓轩 , 孟庆昌等. H. 323 视频会议中 MCU 的设计与实现. 北京邮电大学学报 , 2003 , 26(2) : 77~81
- [xxxviii] 施吉鸣. 基于 IP 网络的视频会议系统及其应用. 宁波职业技术学院学报 , 2004 , 8(05) : 80~81
- [xxxix] Langi. A. Z. R , Yonata. Y. A performance and interoperability study of an OpenH323-based multimedia over IP (MOIP) system. Circuits and Systems , 2002.10 , 1 : 427~430
- [xl] R. E. Mayer , Multimedia Learning. New York : Cambridge Univ. Press , 2001 , 15~38
- [xli] 李智芳 , 黄超. 基于 DirectShow 流媒体采集和播放系统设计和实现. 中山大学学报 , 2005 , 44(11) : 166~169
- [xlii] 陆其明. DirectShow 实务精选[M] . 北京: 科学出版社 , 2004.
- [xliii] 方波 , 曾致远. 基于 DirectShow 的流式立体视频播放器的设计与实现[J] . 宽带网络与传输 , 2004 , 10 : 39~42
- [xliv] 陆其明. DirectShow 开发指南. 北京: 清华大学出版社 , 2003.
- [xlv] 卢政. 如何成功的运用 OPENH323 来开发商业的 H.323 协议栈 [EB/OL]. 2002.11 : 61~72
- [xlvi] 袁聃 , 程文青 , 吴砥. 虚拟摄像头在基于 H.323 的远程教学系统中的应用. 计算机工程 , 2007 , (11)
- [xlvii] Equivalence Pty Ltd. Openh323 Source Code. [Online] <http://www.openh323.org/code.html>. Last Visited 2006.6

[xlvi] Equivalence Pty Ltd. PWLib Source Code. [Online] <http://www.openh323.org/code.html>. Last Visited 2006.6

## 附录一 攻读硕士期间发表论文目录

[1]袁聃，程文青，吴砥. 虚拟摄像头在基于 H.323 的远程教学系统中的应用. 计算机工程，已录用. 署各单位：华中科技大学