摘　要

在当今的操作系统中，虽然 Windows 占据统治地位，但是 LINUX 经过十几年的发展，也已逐渐成熟，使用 LINUX 的 Web 服务器随处可见。与 Windows 相比，LINUX 自身的稳定性以及优秀的网络性能使它的应用日益广泛，而且 LINUX 提供了友好的图形用户界面，已经直接向主宰个人电脑的 Windows 系统提出了强大的挑战。

而随着个人计算机和 Internet 技术的普及，集文本、图形、图像、动画、音频和视频等多种信息为一体的多媒体业务正在迅速渗入人们的生活，并成为网络通信的主流。但是数字视频信息的数据量巨大，无论存储还是传输都很困难，因此视频压缩技术就成为多媒体通信中最为关键的技术。国际上通用的图像视频压缩标准有 JPEG、H.261、H.263 以及 MPEG 系列。MPEG 标准兼顾了图像质量和压缩费用，在图像、伴音、存储媒体和传输四个方面制定了统一的标准，有效的将计算机、通信和电视广播三方面结合了起来。在本文中简单阐述了 MPEG 视频压缩编码算法及其软件实现。

视频业务在网络上的传输应用包括视频点播、视频会议、可视电话、远程教学、远程医疗等等。本文针对台湾 LEADTEK 公司的客户需求，介绍了局域网内视频点播系统的方案设计和实现，重点阐述了本人完成的客户端部分。

文章首先在第一章中介绍了多媒体通信和课题背景。第二章中介绍视频压缩的基本技术以及国际通用的标准。第三章中具体介绍 MPEG-1 和 MPEG-2 标准以及它们的编解码软件实现，包括 MPEG-1 和 MPEG-2 的基本编解码以及 MPEG-2 数据分割分层编解码。第四章论述我们的视频点播系统方案设计及实现，重点介绍其中客户端的具体设计和工作流程。第五章总结研究生期间所做的工作，并指出今后应继续开发和完善的工作。

关键词： 视频，MPEG，LINUX，数据分割，视频点播，媒体服务器，客户端

Abstract

Among all the operating systems today, although Windows has taken up the dominion place, LINUX system has grown up gradually after decades of progress, and it is used on Web servers everywhere. Compared with Windows, LINUX system is used increasingly broad because of its stability and excellent network performance. And, withal LINUX has provided a friendly graphical user interface, and it has already turned to be a strong challenge to Windows system directly, which is dominating the PC world.

With the development of PC and Internet, multimedia services with text, graphics, image, audio and video are being widely used in our life, and become the main service on network. But the data of digital video is so huge both for storage and for transport, that video compression becomes the key technology in multimedia communications. There are many international standards for video compression, e.g. JPEG, H.261, H.263 and the series of MPEG. MPEG considers picture quality and compression cost, establishes uniform standards for image, audio, storage media and transport, and combines three field of computer, communication and television broadcast. In this paper, we describe MPEG video compression arithmetic, and implement it with software.

There are many applications of video transport on network, e.g. video on demand (VOD), video conferencing, visual telephone, distance education, etc. In this paper, we introduce the design and implement of a VOD system in LAN, which is demanded by Co. LEADTEK. And Most of it is about the client accomplished by me.

The first chapter introduces the multimedia communication and the background of the project. The second chapter introduces the basic technologies and international standards of video compression. The third chapter introduces MPEG-1 and MPEG-2 in detail, then describes their implement with software, specially describes the Data Partition (DP) scalable encoding and decoding. In the fourth chapter, we describe the design of our VOD system, and then describe the client in detail, which is designed by the author. The last chapter summarizes the work during the three years, and points out what to continue and perfect.

Keywords: video, MPEG, Data Partition (DP), VOD, media server, RTP, client

第一章　多媒体通信及课题背景概述

1.1　多媒体通信概论与发展方向

在技术发展史上，计算机、通信和广播电视一直是三个相互独立的技术领域，各自有着互不相同的技术特征和服务范围。但是，近几十年来，随着数字技术的发展，这三个原本各自独立的领域相互渗透、相互融合，形成了一门崭新的技术——多媒体技术。而随着个人计算机和 Internet 技术的普及，集文本、图形、图像、音频、动画和视频等多种信息为一体的多媒体业务正在迅速渗入人们的生活，并成为网络通信的主流。多媒体技术的应用与发展，反过来进一步加速了这三个领域的融合，使多媒体通信成为通信技术今后发展的主要方向之一。

多媒体使用具有划时代意义的“超文本”思想与技术组成了一个全球范围的超媒体空间，通过只读光盘存储器(compact disc read-only memory，CD-ROM)、数字多功能光盘(digital versatile disc，DVD) 、多媒体计算机和融合后的三大网络，使得人们表达、获取和使用信息的方式和方法发生重大变革，对人类社会产生了长远和深刻的影响。

新技术的产生与发展往往与其特定的技术背景密切相关，而且大多数都是以其它相关技术的发展作为基础。实际上，多媒体技术之所以能够在 20 世纪 80 年代末期出现，并且立即在世界范围内得到迅速发展，主要得益于下述几个方面的技术发展。

- 图像压缩编码技术的成熟

在通信领域中，人人都知道数字信号具有模拟信号所无法比拟的优越性。但与此同时，数字信号的传输对带宽的要求也要远远高于模拟信号。一路按分量进行编码的彩色电视信号，按照国际标准，编码后的数据率为 R = 216Mbit/s。因此，要以数字方式传输电视信号，必须要解决数据率的压缩问题。

对各种图像的压缩编码问题人们已经进行了几十年的深入研究，进入 20 世纪 80 年代后，这项技术已经相当成熟，而近几年来，人们又相继提出了很多的压缩方法与标准，如 MPEG 系列，H.323 系列等，这些技术对各种图像及视频信号的

传输都起了极大的推动作用。

将图像压缩编码技术应用到计算机领域则直接导致了新技术的诞生。当 DVI（Digital Video Interactive）技术与世人见面时，它已经能将图像信号和伴音信号压缩 100 倍以上(包括适当地将电视信号的图像分辨率降低)，其速率为 1.2～1.4Mb/s，这使得活动图像数据能够在计算机总线上传输，从而成为计算机可以处理的数据类型之一。

- 大规模集成电路技术的发展

在多媒体技术发展的初期，CPU 的处理能力还很低，那时数据的压缩和解压缩运算还要靠专门的芯片来完成。这主要是因为当时的集成电路技术还不是很发达。

可是，进入二十世纪 90 年代以来，大规模集成电路技术得到了迅猛的发展。现在，在一个 CPU 芯片内，集成的晶体管数可达到 500 万门以上。这也为多媒体技术的发展提供了极为必要的前提条件。

- 大容量数字存储技术的发展

本世纪 70 年代出现的激光视盘(LD)技术，可以在一张盘上存储大约 30 分钟的电视节目，在 80 年代出现了 CD 盘，在一张盘上可纪录超过 70 分钟的音乐节目。

随着光盘技术的发展和随机访问问题的解决，1984 年出现了纪录计算机数据的 CD-ROM, 在一张 CD-ROM 上已经能够满足存储一个电影节目(一小时左右)的要求。与此同时，计算机硬盘和磁盘阵列的容量也得到了飞速的增长。这一切都为多媒体技术的全面发展和实际应用提供了充分的条件。

多媒体涉及的技术范围很广，技术很新，研究内容也很深，是多种学科和多种技术交叉的领域。目前，多媒体技术的研究和实际应用开发主要在下列几个方面：

(1) 多媒体数据的表示技术：包括文字、声音、图形、图像、动画、影视等媒体在计算机中的表示方法。由于多媒体的数据量大得惊人，尤其是声音、图像和影视，包括高清晰度数字电视(High Definition Television，HDTV)这类的连续媒体。为了克服数据传输通道带宽和存储器容量的限制，世界各国已经投入了大量的人力和物力来开发数据压缩和解压缩技术；而人-机接口技术（如语音识别和文本-语音转换(text to speech，TTS)）、虚拟现实(Virtual Reality，VR)也是当今多媒

体技术研究中的热点技术。

(2) 多媒体创作和编辑工具：使用工具将会大大缩短提供信息的时间。将来人人都要会使用多媒体创作和编辑工具，就像现在我们使用笔和纸那样熟练。

(3) 多媒体数据的存储技术：这包括 CD 技术，DVD 技术等。

(4) 多媒体的应用开发：包括多媒体 CD-ROM 节目(title)制作，多媒体数据库，环球超媒体信息系统(Web)，多目标广播技术(multicasting)，影视点播(video on demand，VOD)，电视会议(video conferencing)，远程教育系统，多媒体信息检索等。

1.2 课题背景及研究内容介绍

多媒体是指多种信息传播媒体，如文本、图形、图像、音频、动画和视频等信息的组合。而多媒体通信就是指集这些通信媒介为一体的，具有集成性、同步性与交互性的通信方式。多媒体通信近年来发展迅速，特别是在宽带通信网中，多媒体业务将占据相当大的比重，作为多媒体主流业务的视频业务传输正越来越受到用户和研究者的重视，而视频点播则是这项业务的典型代表。

1.2.1 VOD 的发展得到技术支持

虽然数字存储技术和网络得到不断地发展和完善，但是由于数字视频信息的数据量十分巨大，无论存储或传输都还是很困难，因此视频压缩技术就成为多媒体通信中最为关键的技术。国际上通用的图像视频压缩标准有 H.263、H.261、JPEG 以及 MPEG 系列。其中 MPEG 标准兼顾了图像质量和压缩费用，对图像、伴音、存储媒体和传输四个方面制定了统一标准，有效的将计算机、通信和电视广播三方面结合了起来。

视频压缩编码一系列国际标准的提出标志着视频编码技术已经成熟，开始由学术研究走向产业化，前景十分诱人。早在 1991 年就有人预言，视频编码技术的突破具有十分巨大的意义，其意义之大已到了可以促使现有信息产业的结构发生巨变的程度，它使通信、广播、计算机产业的界限变得模糊了。近年来的实践也证实了这个预言。现在的情况已经不是简单的满足某些用户的图像压缩的要求，而是正在以视频编码为核心技术之一，大规模地积极开拓新的产品和新的领域，

并且已经开始影响到人们的生活方式。电视广播、VCD 和 DVD、视频点播、视频会议、可视电话、远程教学、远程医疗……视频业务正在迅速渗入人们生活的各个方面，可以预计，充满魅力的视频业务将在未来的信息社会中扮演举足轻重的角色。

目前基于 IP 寻址的多媒体通信异军突起，发展势头极为迅猛。而 Internet 已经是家喻户晓，Internet 的 Web 技术也已广为使用。随着 Web 技术的发展，特别是 JAVA，JAVA Script 和 plug-in 技术的不断引入，Web 系统的能力越来越强大，它已经不只是提供简单的文本、图片等的信息检索服务，它还可以提供声音、动画乃至实时运动图像的服务。

当前世界各国、各大厂商都在研究基于 IP 寻址的多媒体通信。国际标准化组织，包括 ISO（国际标准化组织）和 ITU-T（国际电联），也都在积极地进行基于 IP 寻址的多媒体通信的研究，制定相应的标准。

由于基于 IP 寻址的多媒体通信技术的飞速发展，有一大批原来必须在 B-ISDN 中开展的业务，业已证明可以很好地在 IP 网上开展。最有代表性的业务之一就是视频点播（VOD，Video 0n Demand），一直以来，它在各国的 B-ISDN 实验网中几乎无一例外地得到应用。而这项业务目前在 IP 网上开展得很好，国内外已有这样的 VOD 设备，并已投入商业运行，其使用方便程度和服务质量都很好。所以在 Internet 上的实践是三网合一后实现视频点播的第一步。

1.2.2 LINUX 操作系统简介

在 LINUX 操作系统出现之前，由于 UNIX 系统十分庞大，所需硬件也较为昂贵，所以在 PC 机上 Windows 操作系统占据了垄断地位。1991 年，Linus B. Torvalds 在网络上组织人员为 PC 机编写了第一个免费的 UNIX 内核(Kernel)，发展至今已经成为一个能在 PC 机上稳定工作的 UNIX/X-win 操作系统。这个系统被命名为 LINUX 系统。

LINUX 是一个可以在 PC 机上运行的 UNIX 系统，不但具有最新 UNIX 的全部功能，而且还拥有独特的良好性能。LINUX 是一个开放的，多用户、多任务的操作系统，不但性能优越，而且有着良好的用户界面和丰富的网络功能。其它特性还包括设备独立性、可靠的系统安全性、良好的可移植性及按需求定制的灵活性。这些特性使得 LINUX 得到极快的发展和普及。LINUX 系统及其发展均符合

POSIX标准。其内核支持Ethernet，PPP，SLIP，NFS等协议；系统应用包括telnet，rlogin，ftp，mail，gopher，talk，term，news等全套UNIX工具包；X图形库包括xterm，fvwm，xxgdb，mosaic，xv，gs，xman等全部X-win应用工具；中文工具有cxterm，celvis，cemasc，cless，hztty，cytalk，ctalk，cmail等，可以处理GB，BIG5文件。此外还有DOS模拟软件，可以运行DOS/Win下的软件。

归功于上述一系列优点，LINUX操作系统近年来发展迅猛。世界各地有很多程序员和工作小组在积极地为LINUX编写各种各样的应用软件，从DOS环境摸拟到图像、声音信号的处理，从游戏到中文软件，无所不包。而各大软件公司也已纷纷推出LINUX版本的商业软件。由于得到了诸如英特尔、IBM、惠普以及戴尔等业界巨头的支持，LINUX现在正在成为主流产品。从戴姆勒克莱斯勒公司到华尔街的大多数重量级经纪公司，LINUX都在为自己赢得一席之地。谁能想到，就在3年前还是两手空空的LINUX现在已经占据了价值509亿美元的服务器市场13.7%的份额。据IDC市场研究公司预计，上述数据到2006年将攀升到25.2%，使得LINUX在服务器市场上坐稳第二把交椅。IDC公司还预计，从2003年开始，占据服务器市场头把交椅的微软公司的市场份额将从现在的59.9%开始下降，而与此同时LINUX则将在诸如“游戏工作站”——视频游戏机和电视录像机等多种消费电子产品中亮相。毫无疑问，Linux作为一个新兴事物已经日渐成熟。

与Windows操作系统相比，LINUX自身的稳定性以及优秀的网络性能使它的应用日益广泛。从大的方面说，LINUX的自由软件思想是微软公司专业软件思想的挑战者。自由软件彻底推翻了微软王国的商业模式，它主张源代码公开，使得程序员和用户一起融入产品开发中，可以根据自己的需要进行修改和完善。而且用户可以自由拷贝软件，充分享受信息技术革命带来的好处。以Windows为代表的封闭式产权形态受到了以LINUX为代表的开放式产权形态的挑战，而国际互联网的崛起则注定了软件产业从封闭到开放的大势所趋；从小的方面说，LINUX是最适合计算机爱好者的操作系统。首先，LINUX是一个UNIX操作系统在个人电脑上的完整实现。有了它，用户可以在个人电脑上运行各种UNIX命令，使用各种UNIX软件，享有从Internet上获得的各种为UNIX编写的免费软件、工具乃至游戏。其次，LINUX是免费的，即使我们不能从网上下载也只需要花费工本费就可以得到它的光盘安装版。它完全公开的内核及软件源代码对于从事计算机科研的工程师们来说，更是难得的宝贝。而且LINUX提供了友好的图形用户界面，已

经直接向主宰个人电脑的 Windows 系统提出了强大的挑战。

LINUX 正走向主流!

1.2.3 课题研究内容

基于以上介绍的视频点播业务的迅速发展和 LINUX 操作系统的广泛应用，我们实验室选择了 LeadTek 的这个项目，在 LINUX 操作系统上实现基于 IP 的视频点播功能，将支持 MPEG 流的传输与播放。而我个人承担的课题是在 LINUX 上实现视频点播系统的客户端。

我在研究生期间主要所做的工作为：研究 MPEG 标准，在多媒体计算机平台上用软件方式实现 MPEG 流数据的编解码算法；在多媒体计算机平台上用软件方式实现局域网内的视频点播系统，重点完成系统的客户端，即 MPEG 流的解码播放程序。

选择在多媒体计算机平台上进行研究开发的原因在于：它能将多媒体数据的采集，编解码，传输，显示等功能集于一身，可以不需要其它硬件设备而直接在其已有的硬件平台上很容易地进行系统开发。而计算机在社会生活中各个领域都得到了广泛的应用，普及率也在快速增长，在不久的将来就可以达到甚至超过电视机的普及率。所以，以多媒体计算机为平台开发的系统会有广阔的市场前景。

选择以软件作为多媒体通信系统的开发手段的原因在于：成本低；便于版本升级和功能增强；更容易推广和占领市场等。但是软件开发的缺点就是对计算机硬件的依赖很强，知识产权保护能力弱。随着计算机技术的不断发展，硬件设备的升级和更新换代，软件多媒体通信系统的优越性将逐渐展露出来，缺点也会得到克服。每一次计算机硬件性能的提高，都是对软件系统的自然升级。

1.3 论文内容安排

本论文的主要内容安排如下：

第一章介绍多媒体通信及其发展方向；讨论了课题背景中视频点播业务的发展和 LINUX 操作系统的一般特性；简单介绍了研究生期间的工作内容，并解释了选择此课题的缘由；最后是本论文的内容安排。

第二章简单介绍了多媒体通信中的关键技术----视频压缩技术，包括视频压缩

中常用的基本技术以及国际通用的视频压缩标准，并作了简单的理论分析。

第三章论述研究生期间对 MPEG 视频压缩技术的研究，首先介绍 MPEG-1 和 MPEG-2 标准，然后介绍它们的编解码软件实现工作，包括 MPEG-1 和 MPEG-2 的基本编解码以及 MPEG-2 数据分割分层编解码。

第四章主要介绍 MPEG 视频流在 IP 网上的传输应用——局域网内的视频点播系统。首先介绍系统的总体方案设计，然后具体介绍客户端的结构设计和功能模块划分，最后详细介绍我在其中具体软件实现的功能模块。

第五章是对整个研究生期间所做工作的总结，以及对将来接续工作的建议。

第二章　多媒体通信中的关键技术----视频压缩技术

视频，就是活动图像。我们所看到的视频信息实际上是由许多单一的图像所组成的，每幅图像称为一帧。由于人眼的视觉惰性，每秒 24 帧的电影画面就形成了连续活动影像感觉的电影。

图像是多媒体携带的信息中极为重要的媒体，人类接受的信息约有 70%来自视觉系统，实际上就是图像和视频。但是，图像和视频数字化之后的数据量非常大。以一路电视信号为例，按 CCIR601 标准，数字化后的分辨率为 720*576，每秒 25 帧，Y:U:V 为 4:2:2。若用 8bit 数据表示 Y 信号，则每像素占 16bit，数码率为 165.9Mbps。以 64kbps 作为一个数字话路，若不加压缩，为传输一路电视要占用 2592 个有效数字话路，这在实际应用中是难以接受的。若用一个容量为 1GB 的硬盘或者 CD-ROM 来存储这样的数据，则只能存储不到 1 分钟的视频，并且所需的高数据吞吐量是一般的硬盘和 CD-ROM 难以达到的。若不加压缩，HDTV 信号的数码率甚至可达到 1Gbps。因此必须要对图像和视频数据进行压缩，这样才能够满足存储容量和传输带宽的要求。几十年来，许多科技工作者一直在孜孜不倦地寻找更有效的方法，以尽量少的数据量表征图像和视频，同时保持复原图像和视频的质量，使它符合预定应用场合的要求。

视频数据可以进行压缩有几方面的原因。首先，原始图像是高度相关的，存在很大的冗余度。数据冗余造成比特数浪费，消除这些冗余可以节约码字，也就是达到了数据压缩的目的。大多数图像内相邻像素之间有较大的相关性，即空间冗余度；图像序列（即视频）前后帧之间也有较大的相关性，即时间冗余度。

其次，若用相同码长表示不同出现概率的符号也会造成比特数的浪费，这种浪费称为符号冗余度，如果采用可变长编码技术，对出现概率高的符号用短码字表示，对出现概率低的符号用长码字，就可以消除符号冗余度，从而节约码字。

允许图像编码有一定程度的失真也是图像可以压缩的一个重要原因。在许多应用场合，并不要求经压缩及复原以后的图像和原图完全相同，而允许有少量失真。只要这些失真并不被人眼所察觉，在许多情况下是完全可以接受的。这就给压缩比的提高提供了十分有利的条件。图像质量允许的损失越多，可以实现的压缩比就越大。这种有失真的编码称为限失真编码。在多数应用中，人眼往往是图

像信息的最终接收者（信宿），如果能充分利用人眼的视觉特性，就可以在保证所要求的图像主观质量的前提下实现较高的压缩比，这就是利用了视觉冗余度。

2.1 视频压缩的基本技术

基本的视频压缩技术包括：预测编码、正交变换、最佳量化与矢量量化、熵编码以及运动补偿预测等，下面对它们简要的加以介绍。

1、预测编码

预测编码也称为差分脉冲编码调制（DPCM），既可在一帧图像内进行帧内预测编码，也可以在多帧图像间进行帧间预测编码。预测编码的基本技术是信号的最佳预测和最佳量化。

帧内进行预测编码的理论依据是二维图像中相邻像素之间存在很强的相关性，因此可以用已知的前面几个像素值来预测当前像素的值。然后对实际值和预测值之间的差值（预测误差）进行量化和编码。帧内预测编码的优点是方法简单，硬件实现容易，缺点是对信道噪声及误码很敏感，会产生误码扩散，同时帧内 DPCM 的压缩比较低，通常为 2-3 倍。随着正交变换编码的广泛应用，帧内编码的作用已经很有限。目前主要使用帧间预测方法来压缩视频信号。

帧（场）间预测编码的理论依据是视频信号的相邻帧（场）之间存在极强的相关性。利用这种时间相关性来进行帧间编码，可以获得比帧内 DPCM 高得多的压缩比。因此，帧间编码被广泛应用于视频压缩编码标准，如 H.263、MPEG 标准等。帧间预测编码方法包括帧间统计特性、帧（场）重复、阈值法、帧内插、运动补偿预测、自适应帧内/帧间编码等。

2、正交变换编码

预测编码的任务是要使预测值尽可能接近实际样值，也就是要寻找一种尽可能接近原信号统计特性的预测方法，通过相差来除去视频图像信号的相关性，从而达到数据压缩的目的。一种更有效的除去视频图像信号相关性的方法是进行信号变换。通常采用正交变换编码，其主要优点有：

- 在变换域里描述视频图像要比在空间域里简单。
- 视频图像的相关性明显下降，信号的能量主要集中在少数几个变换系数

上，采用量化和熵编码可以有效的压缩其数据。

• 可以充分利用人眼的视觉特性，例如空间频率特性、视觉心理和视觉现象等。

• 具有较强的抗干扰能力，传输过程中的误码对图像质量的影响远小于预测编码。通常，对高质量的图像，DPCM 要求信道误码率<10^{-6}，而变换编码进要求信道误码率<10^{-4}。

K-L 变换是最佳的正交变换，其变换后的系数之间是互不相关的。但就实现的成本与实时性来说，K-L 变换是最困难的一种变换。由于它的理论价值高于实际价值，因此通常被作为衡量其他变换方法性能的一个标准。

在视频压缩中，最常用的变换方法是离散余弦变换，即 DCT，它被认为是性能最接近 K-L 变换的准最佳变换。DCT 变换有快速算法，能够实现实时视频压缩。

DCT 变换矩阵的大小可以从去相关程度和实现难易等方面来综合考虑，8*8 通常被认为是一种较好的选择。其二维 DCT 变换为：

正变换：$F(u,v) = (1/4)*C(u)*C(v)*$

$$\{\Sigma\ \Sigma f(x,y)*\cos[(2x+1)u\pi/16]*\cos[(2y+1)v\pi/16]\}$$

反变换：$f(x,y) = (1/4)*$

$$\{\Sigma\ \Sigma C(u)C(v)*\cos[(2x+1)u\pi/16]*\cos[(2y+1)v\pi/16]\}$$

其中：$C(u) = 1/\sqrt{2}$　当 $u = 0$

1　其他

$C(v) = 1/\sqrt{2}$　当 $u = 0$

1　其他

经 DCT 变换得到的系数，其能量主要沿对角线分布，且在左上角集中了主要能量，这反映了能量以低频成分为主的客观现实，即图像大部分区域变化不大，亮度突变部分占少数。

3、量化

量化是将具有连续幅度值的输入信号转换成只具有有限个幅度值的输出信号的过程。在压缩编码中，量化的目的是在确保一定图像质量的前提下，舍弃一些对视觉效果影响不大的次要信息，从而达到进一步的压缩。量化对压缩后的码率和重建图像质量均会产生重要影响。

最佳量化器的设计方法有两种，一种是客观准则设计法，采用量化均方误差最小为约束条件；另一种是主观准则设计法，它根据人眼的视觉特性来设计量化器。通常采用主观准则设计方法，利用人眼视觉特性中的视觉掩盖效应，当图像边缘相邻两侧的亮度值相差很大时，采用比较大的量化步长，这样可以在相同数据量的情况下得到比较高的图像质量。

对于各种编码的量化器来说，都有自己独特的量化方式。采用哪种量化方式是由所要量化的系数的特点来决定的。在 DPCM 预测编码中，常常使用非均匀量化器，其目的不在于使量化误差的均方值最小，而在于使量化误差的能见度最小。在 DCT 变换后，一个系数的量化误差可以对整个块的图像质量产生影响，直流分量的量化误差将扩大成块与块之间灰度均值的差异。所以对变换系数量化的处理办法一般分为分带编码（Zonal coding）和阈值编码（Threshold coding）两种。

4、熵编码

熵编码是一种基于量化系数统计特性所进行的无失真编码。常用的熵编码方法有游程长度编码、哈夫曼编码和算术编码。在视频编码中，通常采用游程长度编码和哈夫曼编码。

游程编码目前已广泛应用于视频编码中，例如 DCT 编码中，经过量化的 DCT 系数通常会出现很多零系数。在这种情况下，与其传送大量的零系数，不如告知接收端那些非零系数，并告知两个非零系数之间有多少个零，恢复时插入零系数即可。为了增加相连零系数的个数，通常对已量化的系数按“Z”字形排列。

哈夫曼编码是图像压缩中最重要的编码方法之一，是 1952 年由哈夫曼（Huffman）提出的一种非等长最佳编码方法，其基本原则是给出现概率较大的符号（事件）分配较短的码字，给出现概率较小的符号分配较长的码字，从而提高编码效率。

5、运动补偿预测

如前所述，采用帧间预测编码可以减少时间域上的冗余度，提高压缩比。如果将上一帧相同空间位置处的像素值作为待编码的当前帧的预测值，这种预测对图像中的静止背景部分是很有效的，但是对于运动部分，这种不考虑物体运动的简单的帧间预测效果并不好。如果有办法将对当前帧某像素（或像素块）进行预

测时知道这个像素（或像素块）是上一帧的哪个位置移动过来的，在做预测时以那个位置上的像素值作为预测值，则预测的准确性将大为提高。也就是说采用运动补偿帧间预测可以使预测差的方差大大减小，从而减低码率，提高压缩比，也就是说通过运动补偿帧间预测可以更好的利用序列图像的时间冗余度。运动补偿技术的主要内容包括：

• 图像分割。将视频图像分割成静止部分和运动部分。

• 运动检测与估值。检测运动类型（平移，旋转和缩放等），估计运动物体的位移值。通常采用块匹配算法进行运动估值。最优匹配的搜索算法有全搜索法、三步搜索法、正交搜索法等。运动估值是运动补偿的关键。

• 运动补偿。用位移估值进行运动补偿预测。

• 预测信息编码。对预测信息（例如运动矢量）进行编码，作为边信息传送给接收端。

2.2 视频压缩标准简介

视频压缩编码经过四十多年，尤其是近二十多年的发展，有些技术已十分成熟，其中以离散余弦变换(DCT)为代表的一类算法得到了广泛的应用和认可，并被采纳为国际标准。在本节，我们简要地介绍一下这些标准，这对了解视频压缩编码的现状是和了解本论文研究背景是必不可少的。

国际上视频压缩主要遵从以下几个标准：JPEG 标准、JPEG-2000 标准、H.261 建议、H.263 建议、MPEG-1 标准视频部分、MPEG-2 标准视频部分、MPEG-4 标准等。

1、JPEG 标准

JPEG 标准是联合图像专家小组为连续色调（灰度或彩色）静止图像压缩制定的通用国际标准，其算法要点在于 DCT 和可变长编码（VLC）压缩技术，其编解码过程如图 2-1(a) (b)所示。

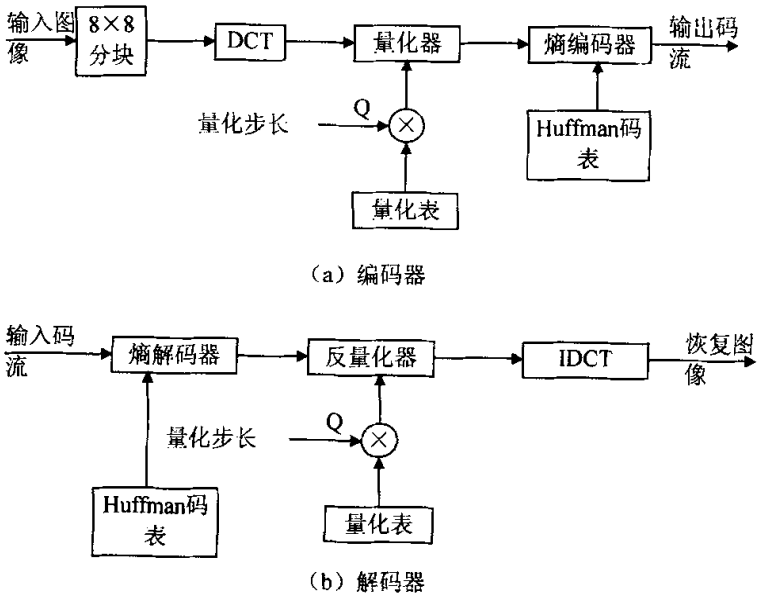


图 2-1　JPEG 基本编解码系统原理图

输入的 8×8 图像块经前向 DCT 变换后，其变换系数首先经过标量量化，然后进入变字长编码器对经 Zig-Zag 扫描得到的幅度和游程进行熵编码。解码器完成以上各步骤的相反操作，得到重建图像。

只要硬件处理速度足够快，JPEG 也能用于实时视频压缩。

2、JPEG-2000 标准

随着图像通信的应用领域的增加，人们对图像压缩的标准又提出了更高，更多的要求，为满足这些要求，自 JPEG 标准公布以来，人们对彩色静止图像的压缩编码又展开了很多研究。JPEG-2000 就是集合这些研究成果构成的一个新标准，它不仅包括新的压缩算法，也包括灵活的压缩结构和格式。JPEG-2000 可在低比特率下提供比现有标准更高的主观质量，同时，它是一种基于结构的标准，可以通过下载软件来集成新的压缩算法部件，而不需增加新的标准定义。JPEG-2000 主要采用小波子带编码。

JPEG-2000 相对于 JPEG 的主要优点有：

- 高压缩比，可用于大图像的压缩；
- 支持既具有连续值部分又具有二值部分的混合图像的压缩，可用于带有文

字的图像的编码；

- 基于像素精确度和清晰度的逐级传输，允许逐步增加图像的清晰度；
- 对比特差错有鲁棒性，可用于无线信道传输；
- 为不同的应用系统提供优化的开放的结构体系；
- 基于内容的描述，支持在大型图像数据库中查找图像的需求；
- 可通过数字水印，数字标签等实现图像的保密性。

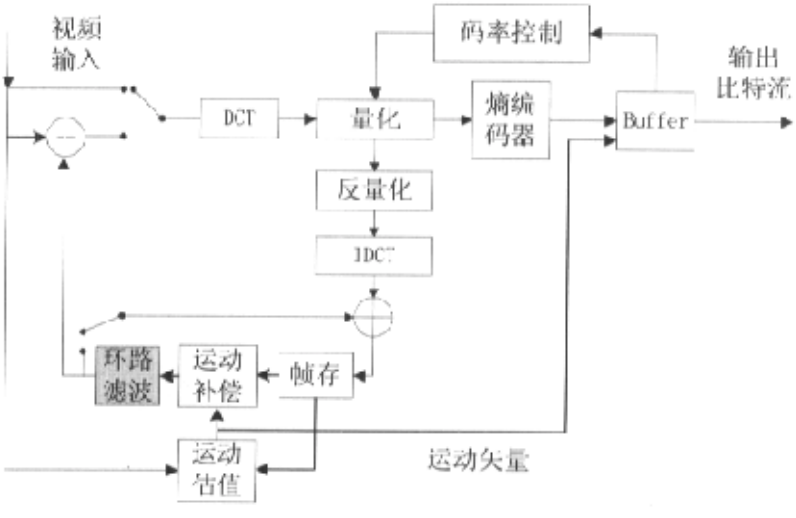
JPEG-2000 的应用领域包括：文件图像，医学图像，传真，Internet/WWW 图像，遥感，数字照片图书馆，电子照片等。

3、H.261 建议

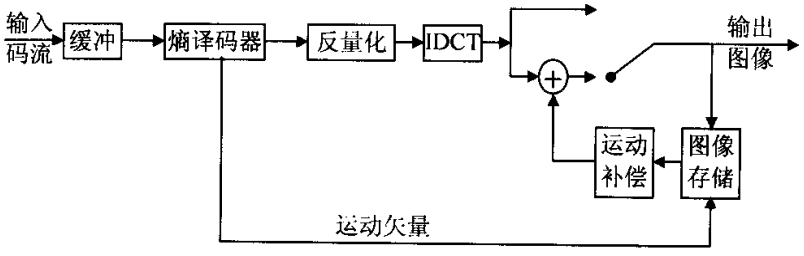
H.261 标准面向 ISDN 上的可视电话，会议电视等应用。它可工作于单路 B 信道(64kb/s)、多路 B 信道、单路 H 信道(384kb/s)，多路 H 信道等环境。

H.261 标准是世界上第一个得到广泛承认的，并产生巨大影响的数字视频压缩编码标准，此后国际上制定的 MPEG-1，MPEG-2，H.263 等数字视频编码标准都以它为基础和核心。

H.261 视频压缩编解码算法的原理框图如图 2-2 所示，其基础和核心是混合编码技术，即：带有运动补偿的帧间 DPCM+二维 DCT 变换+熵编码。



(a) 编码器（注：阴影部分为可选项）



(b) 解码器原理框图

图 2-2 H.261 视频压缩编解码

与 JPEG 编解码器相比，图中除了 DCT 变换、量化、熵编码外，新增了两个关键部分。(1) 运动估值、运动补偿单元。该单元以宏块（16×16 的像素块）为单位作帧间预测，运动补偿，且当预测效果不理想时，切换到帧内方式进行编码。(2) 输出缓冲区 Buffer 以及码率控制单元，当编码器工作于恒定比特率模式时，需要在编码器和信道之间放置一个缓冲器以平滑编码器输出的变速率比特流。码率控制单元的作用在于根据缓冲器的状态调整量化参数以避免 Buffer 产生上溢或下溢。其中，运动补偿是运动图像编码的关键所在，旨在去除图像序列在时间轴方向上固有的冗余度。

4、H.263 建议

H.263 建议是为了满足低码率活动图像的编码制订的。H.263 同 H.261 标准一样，采用运动补偿和 DCT 编码的方法。但是为了提高压缩比，H.263 较 H.261 又采取了一些新的措施。H.263 相对于 H.261 的主要改进如下：

• 半像素精度的运动补偿。运动矢量精度的提高使经运动补偿后的帧间误差减小，从而降低了码率。

• 不受限的运动矢量（可选项）。运动矢量可以指向图像以外的区域，此时指向图像外部的参考像素用最近的边缘像素替代。此模式可以改善边缘宏块的预测效果，适合较小的图像格式。

• 用基于句法的算术编码代替霍夫曼编码（可选项）。算术编码可以在较小的比特率下获得同等的图像质量，但编译码器的复杂度有所提高。

• 先进的预测模式（可选项）。对宏块内中的 4 个 8*8 亮度块分别进行运动

估值获得 4 个运动矢量，如果利用 4 个运动矢量所得到的预测误差要比用整个宏块估值所得到的单个运动矢量的预测误差小得多，则传送 4 个运动矢量。虽然此时传送运动矢量所花费的比特数增加了一些，但由于预测误差的大幅度降低，仍然使总码率降低。

• PB 帧模式（可选项）。H.263 采用 P 帧和 B 帧作为一个单元来处理的方式，即将 P 帧和由该帧与上一个 P 帧所共同预测的 B 帧一起编码。H.263 中只有第一帧是 I 帧，其余都是 P 帧或 PB 帧。采用这种模式可以在编码比特率增加不大的情况下使编码图像的帧频有较大的提高。

5、MPEG-1 标准

MPEG 是 Moving Picture Experts Group 的简写，该组织成立于 1988 年，MPEG-1 是其推出的第一个标准，主要应用于视频信号(如电影节目)的数字化压缩存储，其图像质量与家用录像机相近。MPEG-1 的目标是对逐行扫描的视频进行编码，比特率约为 1.5Mbit/s。MPEG-1 用到的主要编码技术是帧内 DCT 编码和帧间预测运动补偿。MPEG-1 标准将在第三章中详细阐述。

6、MPEG-2 标准

MPEG-2 是主要针对数字视频广播（DVB）、高清晰度电视（HDTV）和数字视盘（DVD）等应用而制定的 4 Mbit/s～9 Mbit/s 运动图像及其伴音的编码标准。MPEG-2 视频是 MPEG-1 视频的扩展。MPEG-2 与 MPEG-1 相比功能上有了很大的扩充，图像质量也有了很大的提高，它不仅支持普通的 CIF、CCIR-601 等分辨率格式，还可支持高清晰度分辨率；不仅支持面向存储媒介的应用，还广泛地支持各种通信环境下数字视频信号的编码与传输，如卫星广播、数字地面广播等等。MPEG-2 另一个重要特色是其比特流的可分级性（Scalability），这意味着解码器可以忽略比特流中的增强部分，只解码全部比特流中的基本部分，仍可得到有用的图像序列，只不过这时所得到的图像分辨率低一些，或者帧速率低一些，或者质量低一些。比特流的分级能力可通过空间分级（Spatial Scalability）、时间分级（Temporal Scalability）、信噪比分级（SNR Scalability）、数据分割(Data Partitioning) 四种方法来实现。MPEG-2 标准也将在第三章中详细阐述。

7、MPEG-4 标准

MPEG-4 是面向多媒体、面向网络应用、采用面向对象技术的新一代通用信息处理技术。它采用更有效、更灵活的手段来处理图像、声音以及其它类型的数据。它不仅可以处理自然影像、声音，而且还可以处理合成影像、声音。它允许用户根据自己的需要，将 MPEG-4 提供的一些标准算法/工具进行灵活的配置，以适应不同的应用场合。它实现了基于内容的压缩编码，具有良好的兼容性、伸缩性和可靠性。MPEG-4 的视频部分提供一组算法和工具来实现不同的功能：如高效的压缩编码技术、空间域的分级、时间域的分级，对象分级以及优良的抗误码能力等。

MPEG-4 另有一部分工作称之为“合成/自然复合编码”（Synthetic/Natural Hybrid Coding），它主要研究合成数据的编码。SNHC 的目的是在 MPEG-4 的框架内提供合成的视频/音频素材。SNHC 的研究内容包括脸部与躯体的动画、文本到语声的合成，等等。

第三章 MPEG视频压缩技术研究

3.1 MPEG-1标准----ISO/IEC11172

MPEG标准，是由国际标准化组织ISO和国际电工委员会IEC共同制定的的视频压缩标准，制定时称为MPEG标准，在MPEG-2及以后的标准出台之后，为清楚起见，一般将其称为MPEG-1标准。运动图像专家小组（Motion Picture Expert Group）于1990年提出标准草案，1992年正式出版，标准的编号为ISO/IEC11172，标准的题目为“具有1.5Mbit/s数据传输率的数字存储媒体运动图像及其伴音的编码”。此编号内共有三部分内容：第一部分为系统，编号为11172-1，阐述几种伴音压缩数据和图像数据的复用，以及加入同步信号后的整个系统；第二部分为视频，编号为11172-2，阐述视频数据的压缩；第三部分为音频，编号为11172-3，阐述数字伴音的压缩。这里对前两部分简单地加以介绍。

MPEG-1系统层数据流是由一系列包组成的，由包起始码(pack-start-code)0x000001BA标识。每个包由包头和多个分组包组成。包头中包含系统首部（由起始码(system-header-start-cede)0x000001BB标识）。分组包起始码(packet-start-cede-prefix+stream-id)从0x000001BC到0x000001FF，不同的起始码代表不同的分组包类别，其中值得我们注意的是0xE3代表视频分组包，其中的负荷为视频数据，按照MPEG-1第二部分进行编码；0xC0和0xD0代表音频分组包，其中的负荷为音频数据，按照MPEG-1第三部分进行编码。MPEG-1系统层的数据包结构如图3-1所示。其中图（b）中分组数据可能是视频或音频或其他，根据分组包的起始码来判定。

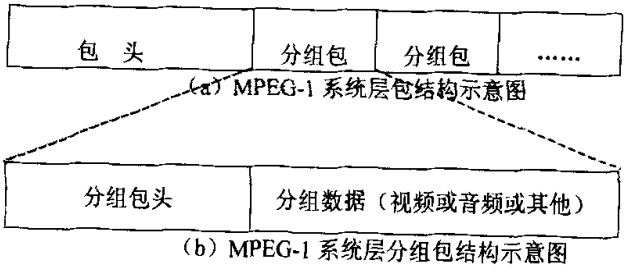


图3-1 MPEG-1系统层结构示意图

MPEG-1第二部分即视频部分中指出，ISO/IEC定义的编码表示可获得高的压

缩比，同时可保持良好的图像质量。由于在编码过程中并不是保存精确的像素值，所以算法不是无损的。编码技术的选择是基于要求高质量的图像、高压缩比与对编码比特流的随机操作需求之间的权衡。以比较满意的位速率达到良好的图像质量需要很高的压缩比，对此仅用帧内编码是无法完成的。然而对随机操作的需求用纯粹的帧内编码却能获得最满意的结果。这需要在帧内和帧间编码之间、递归和非递归缩减时间冗余之间进行仔细的权衡。

为达到高压缩比，MPEG-1 视频部分采用了一系列技术，首先是为信号选择一个合适的分辨率，再一个就是利用基于块的运动补偿来减少时间冗余的算法。运动补偿用来根据前序图对当前图进行因果预测，根据后继图对当前图进行非因果预测，或者根据前序图和后继图对当前图进行差值预测。运动矢量是针对每一16*16（像素*线）的图像区域定义的。差值信号，即预测差，利用离散余弦变换（DCT）作进一步压缩以消除空间相关性，然后进行量化。在量化过程中要丢掉一些不太重要的信息，因而这是不可逆的过程。最后，运动矢量与 DCT 信息相结合，用变长码进行编码。

我们采取自上而下的方式来描述 MPEG-1 视频比特流，以对其有一个总体的概念。序列是视频编码的最高层次，序列从序列头开始，序列头中定义了解码器所需的重要参数。序列头后面跟有一个或多个图组（GOP）。序列可以包含有额外序列头。序列以序列结束码（sequence-end-code）0x000001B9 结束。从图 3-2 中可以清楚的看出 MPEG-1 视频流的层次结构。

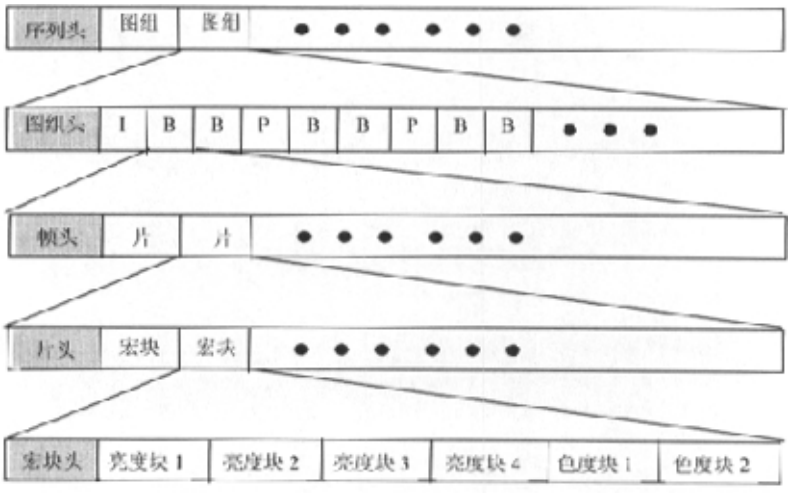


图 3-2　MPEG 视频层结构

序列头：表示上下文的随机存取单元。序列头中定义了许多参数，如图像的宽高尺寸、帧速率、位速率、缓冲器大小以及帧内和非帧内的量化矩阵等，这些在解码时是非常重要的。序列头由起始码 0x000001B3 标识。

图组（**GOP**）：表示视频的随机存取单元。有一个或多个有助于对序列随机操作的编码图像组成的序列称为图组或 GOP（Group of Pictures）。一个图组中的帧序列存在两种明显的图像顺序：解码输入顺序（即它们在视频比特流中的出现顺序）和显示顺序（也就是编码输入顺序）。图组是一组显示顺序上相连续的图。一个图组至少包含一个 I 帧，且必须以 I 帧作为解码顺序的开头。其后可以跟随任意多个 I 帧和 P 帧。在每对 I 或 P 帧之间（亦可在位于第一幅 I 图之前）可插入任意多个 B 帧。在要求能随机操作、快速播放、快速或正常逆放的应用场合，可使用相对较短的图组。图组亦可以从景物剪裁处或其它运动补偿失效的地方启动。常规的图组结构可用两个参量来描述：N 和 M，N 是图组中的帧数，M 是相邻 I/P 帧的间距。图 3-3 显示出一个典型的图组的解码顺序和显示顺序，其 N 值为 9，M 值为 3。图组由起始码 0x000001B8 标识。

时间序号	1	2	3	4	5	6	7	8	9	10	11	12	13
编码输入顺序	I	B	B	P	B	B	P	B	B	I	B	B	P
时间序号	1	4	2	3	7	5	6	10	8	9	13	11	12
解码输入顺序	I	P	B	B	P	B	B	I	B	B	P	B	B
时间序号	1	2	3	4	5	6	7	8	9	10	11	12	13
显示顺序	I	B	B	P	B	B	P	B	B	I	B	B	P

图 3-3　典型的 MPEG 视频流的一个图组

帧（亦称图）：初始编码单元。源图或重构图由三个矩阵组成。矩阵中的数是 8bit 的。三个矩阵是：亮度矩阵（Y）和两个色度矩阵（C_b 和 Cr）。Y 矩阵需具有偶数的行和列，而 C_b 和 Cr 矩阵水平和垂直方向均为 Y 矩阵尺寸的一半。帧层包含一幅图全部的编码信息，其头用于识别该帧的时间参考、编码类型、视频缓冲校验器（VBV）的延迟以及使用的运动矢量的范围（如果合适的话）。在 MPEG-1 中定义了三种视频帧类型，规定了可用哪些预测方式来对每个像块编码：帧内编码帧（I 帧）编码时无需参考其他帧，它给编码序列的解码起始提供操作点，但仅能获得中等的编码压缩比；前向预测帧（P 帧）用从过去 I 帧或 P 帧得来的运动补偿预测，减少了空间和时间冗余度，与 I 帧相比压缩效率更高，并可用

作进一步预测的参考；双向预测帧（B帧）同时采用过去和将来的I或P帧作运动补偿预测，具有最大程度的码率压缩。为了可从将来的帧作后向预测，编码器重新排列了图像，从自然显示顺序转变成解码顺序，因此 B 帧在它参考的过去和将来帧之后传送。这样就产生一个重新排序延时，此延时与连续的 B 帧数有关。帧由起始码 0x00000100 标识。

片（亦称子图）：重同步单元。每个片包含整数个按光栅扫描顺序排列的宏块。一帧内各片的大小可以不同，一帧的分割也不需要与其他帧相同。一帧中的片可以从任一宏块起始或结束；第一个片由帧的左上第一个宏块开始，最后一个片必须结束于帧的右下角最后一个宏块；片之间不能有间隙，也不能重叠；一帧中片的个数最少为一个，最多可以同宏块的个数相同。片由起始码 0x00000101～0x000001AF 标识。后 8 位表示该片的首宏块在帧中的垂直位置。片的首宏块的水平位置可以由片头中的宏块增量得到，于是片的位置可以确定，而无须参考前面的片或宏块。因此，解码器可解图片中的任一片而无须先解图片中的其他片。这一特征允许解码器通过搜索下个片的起始码来克服位错误，并继续解码。一帧中各片的起始码从 0x00000101 开始，此后为非递减顺序，两个或者更多片有可能有相同的起始码。起始码最大可到 0x000001AF，也就是说一帧最多可以分为 176 片。片以上的层次（包括片层）是字节对齐的。

宏块：运动补偿单元。选择 16*16 的宏块作为运动补偿单位是在使用运动信息所带来的编码效率的提高与其需要的附加存储开销之间折衷的结果。每一宏块可以是多种不同类型中的一种。例如，帧内编码、前向预测编码、后向预测编码以及双向预测编码宏块均允许存在于双向预测编码帧中。依照宏块类型，运动矢量信息和其他附加信息同压缩的预测误差信号一样存储于每一宏块中。宏块是非字节对齐的。每个宏块包括四个亮度块和两个色度块。

块：DCT 单元。在 MPEG 标准中，采用一种基于块的视觉加权量化和游程编码的 DCT 方法。作为帧内编码宏块的原始图像以及作为预测编码宏块的预测误差的 8*8 块先变换到 DCT 域并进行比例变换，然后进行量化。量化后许多系数为零，于是利用二维游程和变长码来对剩余的系数进行有效化编码。块也是非字节对齐的。

编码流程如图 3-4 所示：

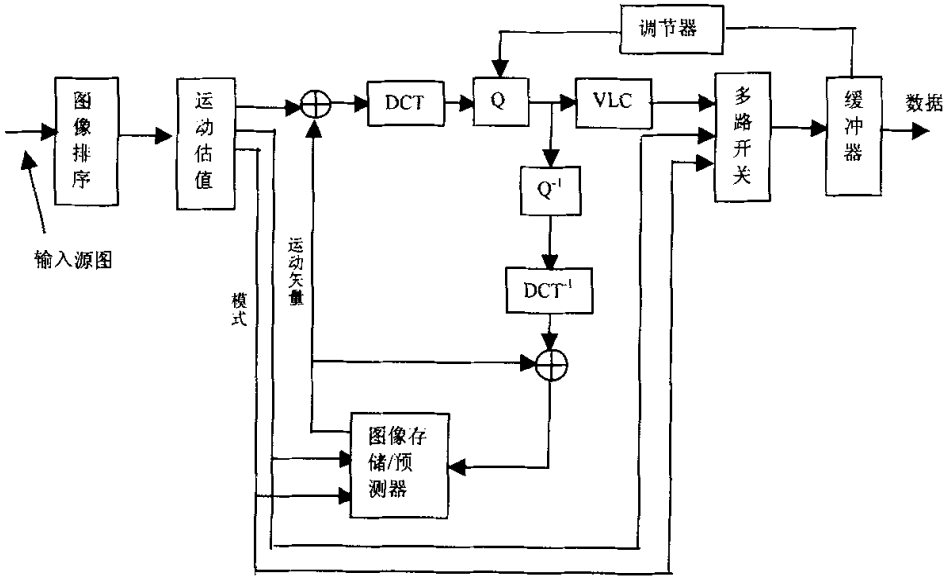


图 3-4　MPEG 视频编码流程框图

下一页是由字节组成的系统流结构图：

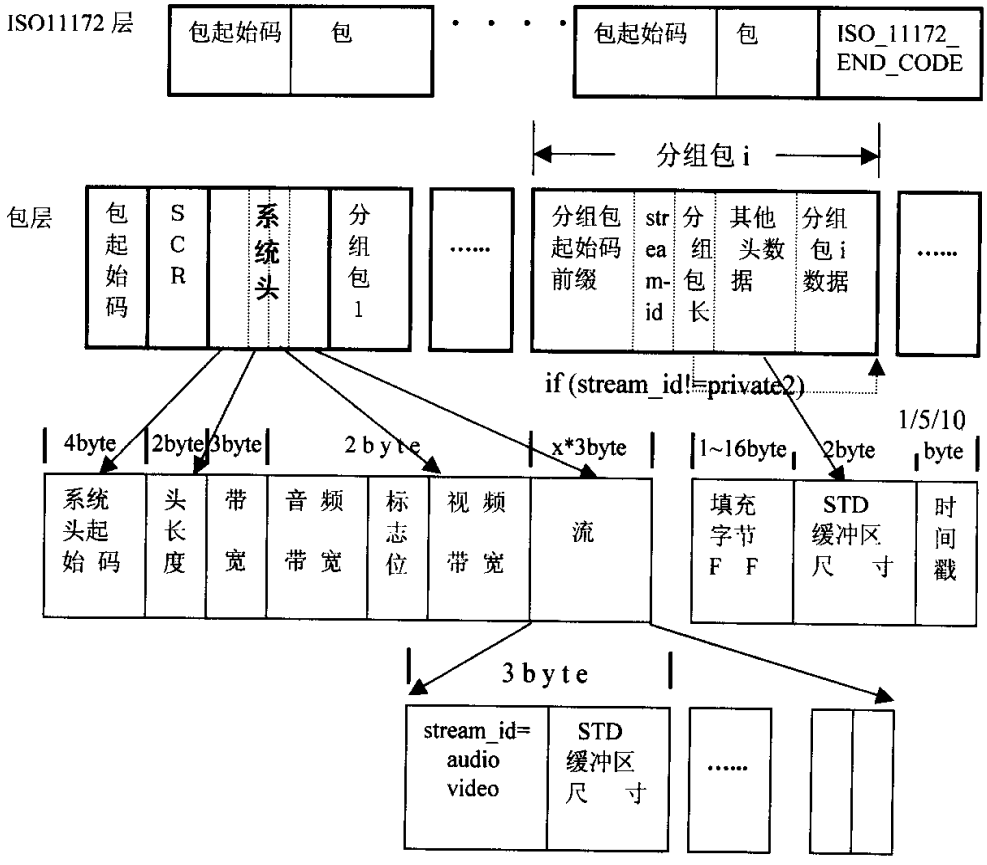
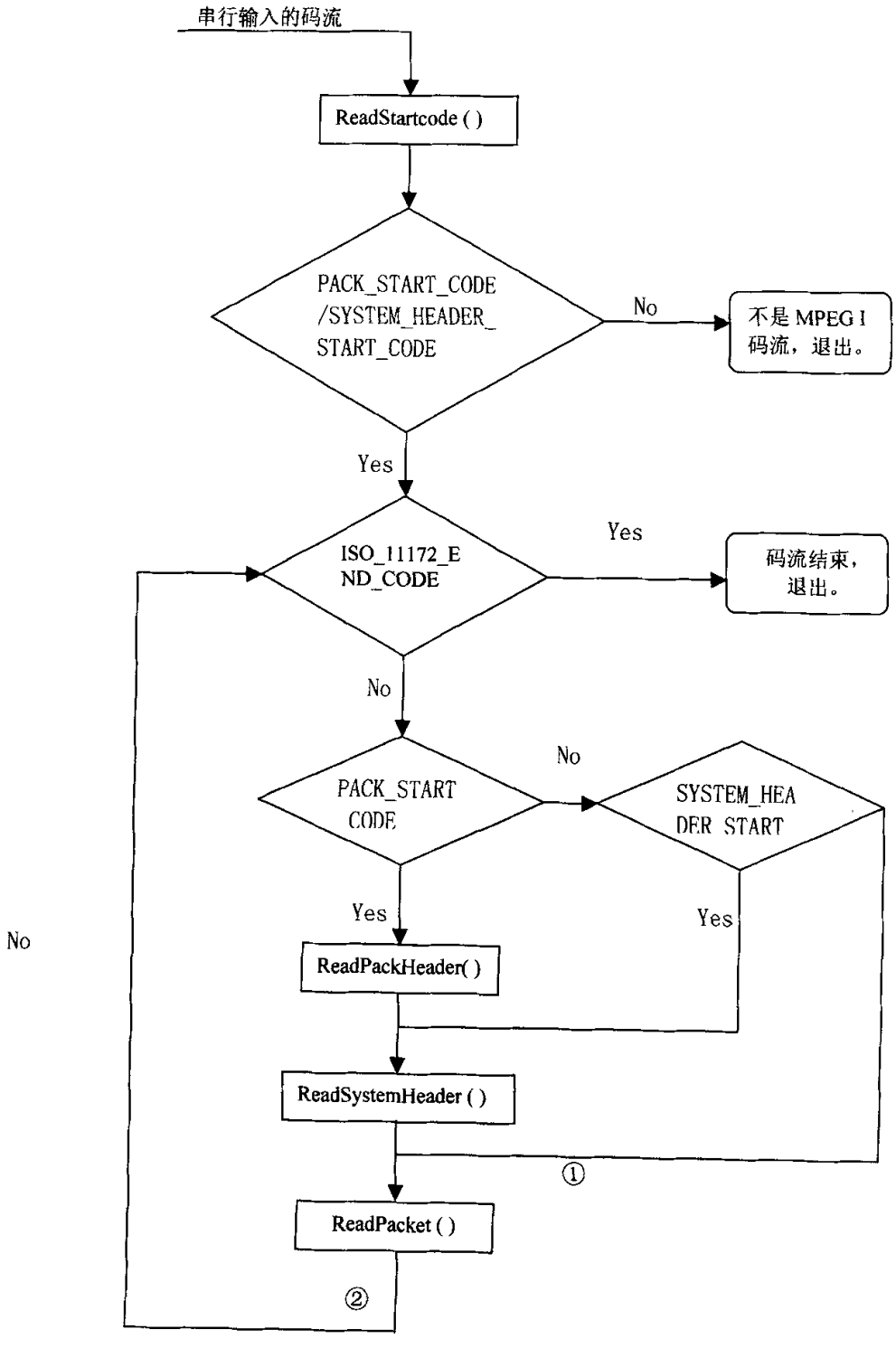


图 3-5　MPEG-1 系统流结构图

通过以上框图，我们可以清楚地了解 MPEG-1 的系统层码流结构，再根据 ISO / IEC11172 的语法说明，即可设计出其系统层解码程序的流程图。这里以解出其音频、视频包为目的，列出标准化的程序流程，（为明了起见，对程序函数作以简要说明）如下：

附：函数说明文档

skip ( )　　：跳过一个字节
storevideo ( )　　：储存视频序列
storeaudio ( )　　：储存音频序列
ReadPacket ( )　　：读分组包数据
ReadSystemHeader ( )　　：读系统头数据
ReadPackHeader ( )　　：读包头数据
ReadStartcode ( )　　：读入数据字节


串行输入的码流
ReadStartcode ( )
PACK_START_CODE /SYSTEM_HEADER_ START_CODE
No
不是 MPEG I 码流，退出。
Yes
ISO_11172_E ND_CODE
Yes
码流结束，退出。
No
PACK_START CODE
No
SYSTEM_HEA DER START
Yes
Yes
No
ReadPackHeader( )
ReadSystemHeader ( )
①
ReadPacket ( )
②

真正的工作在 ReadPacket ( )函数中完成，其流程如下：

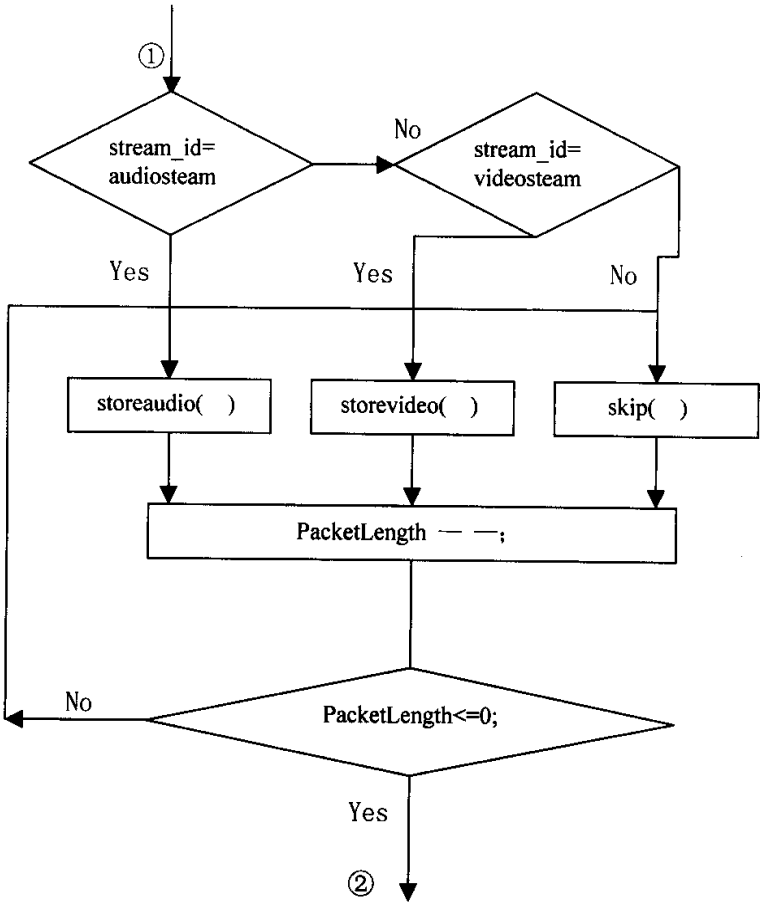


图 3-6 解码流程框图

以上就是 MPEG-1 系统层解出音频、视频包的全过程。

3.2 MPEG-2 标准----ISO/IEC13813

MPEG-2 标准是由运动图像专家小组于 1994 年制定的，标准的编号为 ISO/IEC13813。MPEG-2 是一个用于视频、音频和有关数据的编码和复用的规格，希望它能适用于一系列的应用、码率等级、质量等级和多种业务。MPEG-2 作为公认的压缩方案，具有标准的开放性、技术的低成本、成员间的互操作性和灵活性、比特率的可选择扩展性及众多厂商的支持等优势，在网络、通信、卫星链路等更广阔的领域被采纳。在未来数字电视广播、HDTV 及更广阔的网络视频传输、多媒体通信等综合业务中，MPEG-2 作为标准的地位已经不可动摇。MPEG-2 以 MPEG-1 标准（ISO/IEC11172）为基础，是对 MPEG-1 的改进和扩充。下面我们仍是只讨论用到的第一部分（系统部分）和第二部分（视频部分）。

MPEG-2 标准系统部分中规定了两种系统流：节目流（Program Stream，简称 PS）和传送流（Transport Stream，简称 TS）。

节目流与 MPEG-1 中的系统流类似，它主要是用于在一个不易出错的环境下（如多媒体数据存储）来传送和保存一道节目的编码数据。例如 DVD 中存储数据，采用的就是 MPEG-2 节目流。为保持与 MPEG-1 的兼容性，MPEG-2 节目流包结构与 MPEG-1 系统流的单元层包结构非常相似，只是由于系统时钟由 90KHz 变为 27MHz，使得某些比特位的结构和大小发生了变化。节目流的数据包是变长的。MPEG-2 节目流包的结构示意如图 3-7（a）所示。

而传送流是用于在有可能发生错误的环境下（如网络传输）进行一道或多道节目编码数据的存储和传送，广泛应用于数字视频广播（DVB）和多媒体通信中。传送流由一个个传送分组包组成，每个传送分组包固定为 188 个字节，其中标识其开始的同步字 0x47 占一个字节，其他头信息占 3 个子节，有效负载占 184 个字节。MPEG-2 传送流分组包的结构示意如图 3-7（b）所示。其中数据字节可以来自 PES 分组或者私有数据。数据字节可以赋为任何值。

MPEG-2 节目流和传送流都是以 PES（原始流分组）为基础的，它们都建立于 PES 之上。PES 与 MPEG-1 中的分组包类似。原始流数据作为负载加入 PES 分组中，而 PES 分组又作为负载加入节目流或传送流的分组包中。每个 PES 分组首部的第一个字节就是传送流分组有效负载的第一个字节。PES 分组的结构示意如

图 3-7（c）所示。其中 PES 分组数据可能是视频或音频或其他，根据分组包的起始码来判定。

包头	PES 分组	PES 分组	……

（a）MPEG-2 系统层节目流包结构示意图

同步字	其他头信息	数据字节	数据字节	……

（b）MPEG-2 系统层传送流包结构示意图

分组包头信息	分组数据

（c）MPEG-2 系统层 PES 分组结构示意图

图 3-7　MPEG-2 系统层结构示意图

下面我们来讨论 MPEG-2 的第二部分，即视频层。它与 MPEG-1 相比，主要增加了以下功能：

（1）处理隔行扫描的视频信号的能力

MPEG-1 只能处理逐行扫描的视频信号，而 MPEG-2 中增加了四种对隔行扫描图像更为有效的预测方式：

• 用于场图像的场间预测：这种方式与帧间预测相似，只是被预测的宏块是从某一场图像（而不是帧图像）中取出的，参考图像则可以选择为该场图像的前一场、或者与之极性（即奇数场或偶数场）相同的前一场图像。

• 用于帧图像的场间预测：将帧图像中的宏块分成 2 个场宏块，2 个场宏块分别由奇数场和偶数场的像素构成，然后对这 2 个场宏块分别进行场间运动估值。这样，对前向预测而言，每个帧宏块有 2 个运动矢量，对双向预测而言，每个帧宏块有 4 个运动矢量。这种预测方式主要用于图像剧烈的情况，例如球赛等。

• 用于 P 帧的双基预测：这种预测值用于隔行扫描的图像，并且前一帧不是 B 帧的情况。首先将帧图像中待预测的宏块分成奇数场和偶数场 2 个场宏块，然后从前一帧的奇、偶场分别对这 2 个场宏块做场间运动估值，得到 4 个运动矢量。每个场宏块都是先由同极性场预测，再由不同极性场预测，因此伴随一个场

宏块传送的应该有 2 个（第一次和第二次的）运动矢量。为了用较少的比特数传送运动矢量，采取只传第一个矢量和一个校正矢量（即两个运动矢量的差）的方法。校正矢量的水平和垂直分量数值限制为+1、0 和-1（半个像素），用 2 个比特就能表示。对于每一个场宏块，分别用这 2 个运动矢量做具有运动补偿的场间预测，将 2 个预测值按像素取平均值，作为该场宏块的预测值，用实际值减去预测值，将差值编码传送。在接收端，将第一个运动矢量在时间上扩张或压缩到极性相反的场内，再加上校正矢量，就可以恢复出第二个矢量来。这种预测方式可以达到与双向预测相比拟的性能，而又不会引起像 B 帧那样的编码延时。它的缺点是在编码过程中要占用较多的存储单元。

• 用于场图像的 16*8 预测：这种方法是将场图像的宏块分成上下 2 个 16*8 的半宏块，对每个半宏块分别进行上面所说的场间预测。

对隔行扫描的块，采用图 3-8（b）所示的扫描顺序将 DCT 系数矩阵转化为一维的序列。

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

（a）Z 字形扫描顺序

0	4	6	20	22	36	38	52
1	5	7	21	23	37	39	53
2	8	19	24	34	40	50	54
3	9	18	25	35	41	51	55
10	17	26	30	42	46	56	60
11	16	27	31	43	47	57	61
12	15	28	32	44	48	58	62
13	14	29	33	45	49	59	63

（b）交替扫描顺序

图 3-8　两种图像扫描顺序

（2）更高的色度取样模式

MPEG-1 使用 4:2:0 模式，即色度信号的取样率无论是在水平方向还是垂直方向上都是亮度信号样点数的 1/2。而 MPEG-2 除了 4:2:0 之外，还支持 4:2:2 和 4:4:4 模式，前者色度信号的样点数在垂直方向上与亮度信号相同，只在水平方向上是亮度信号的 1/2；后者的色度信号的样点数则与亮度信号完全相同。

（3）可分级的视频编码方式

所谓可分级的（Scalable）视频编码是指编码所产生的码流具有下述特性：对

部分码流解码和对码流的全部进行解码能够分别获得不同质量的重建图像。对部分码流解码获得的图像比对全部码流解码获得的图像分辨率（或帧率、信噪比等）要低。MPEG-2 所支持的可分级的视频编码方式有空间可分级性编码、时间可分级性编码、信噪比可分级性和数据分割（Data Partition）四种，我们将在下一节中详细介绍。

为了适应不同应用的需要，MPEG-2 用类（Profile）和等级（Level）的形式来定义 MPEG-2 视频规范中语法和语义的子集，由此也就定义了针对某一特定比特流的解码器能力。类是本规范所定义的整个比特流语法的一个限定的子集，而等级是影响比特流中参数的限制集合。一致性检测将针对限定的类，在限定的等级实现。以类和等级的形式定义一致性点的目的是使不同的应用之间交换比特流变得容易。MPEG-2 定义的类、等级以及所推荐的类/等级组合分别见表 3-1、3-2 和 3-3。

表 3-1　MPEG-2 类

类标识	类	特征
110 至 111	保留	-
101	Simple	4:2:0 取样，仅用 I/P 帧，无可分级性
100	Main	以上参数，加上 B 帧
011	SNR 可分级	以上参数，加上 SNR 可分级性
010	空间可分级	以上参数，加上空间可分级性
001	High	以上参数，改为 4:2:2 取样
000	保留	-

表 3-2　MPEG-2 等级

等级标识	等级	最高分辨率
1011 至 1111	保留	-
1010	Low	352*288　　亮度信号样值，30Hz
1001	保留	-
1000	Main	720*576　　亮度信号样值，30Hz
0111	保留	-
0110	High 1440	1440*1152　亮度信号样值，30Hz
0101	保留	-
0100	High	1920*1152　亮度信号样值，30Hz
0000 至 0011	保留	-

表 3-3　推荐的类/等级组合

类	等级			
	Low	Main	High 1440	High
Simple		SP/ML		
Main	MP/LL	MP/ML	MP/H1440	MP/HL
SNR 可分级	SNR/LL	SNR/ML		
空间可分级			SSP@H1440	
High		HP/ML	HP@H1440	HP@HL

在表 3-3 的组合中，Simple/Main 组合适用于视频会议的应用，不使用 B 帧减少了编解码的延时；Main/Main 是 MPEG-2 的主要应用形式，适合于大多数广播电视级别的应用；两种高等级是为高清晰度电视（HDTV）而用的。

3.3　MPEG-2 可分级性编码

如上节所述，可分级的视频编码是 MPEG-2 相对于 MPEG-1 的一个改进功能。在许多情况下，由于用户要求不同、终端能力不同、异构网络的不同支路所能提供的 QoS 不同、或网络传输条件（噪声、拥塞）的变化等原因，需要提供不同质量的视频信号。如果每个内容都存放两种质量的版本显然是不经济的，解决这类问题的最好方法是，用单个编码器产生分层次的已压缩码流，对不同层次的码流解码可以获得不同的图像质量。这也就是 MPEG-2 标准中所提出的可分级性编码功能，也可称作分层编码。

分层编码的优势主要表现在两方面，一是可以适应不同用户终端，低档次终端只对码流的一部分（基本层）进行解码，而处理能力高的终端对整个码流解码，获得高质量的图像。例如对于高清晰度电视节目，如果采用没有分级的普通编码方式，那么如果接收端是分辨率为 1920*1152 的高清晰度电视接收机，当然可以毫无障碍的接收节目并解码播放；但是如果用户使用的只是 720*576 分辨率的普通电视接收机，就无法接收观看这些高清晰度电视节目了。假如对这些高清晰度电视节目采用空间域分层编码，情况就不同了：使用高清晰度电视接收机可以利用基本层和增强层共同进行解码，而使用普通电视接收机的用户也可以只对基本层节目解码，从而也能看到这些节目，只是清晰度不如高清晰度电视接收机播放出来的而已。

分层编码的另一个优势是可以用来在传输条件变坏时改进系统的性能。例如在 ATM 网发生拥塞时，网络节点可能要被迫丢弃一些信元。在一般编码的码流中，数据的丢失不仅造成丢失部位图像的缺损，而且误差还会在空间和时间上扩散。如果采用分层编码，并让基本层数据具有高优先级，增强层数据具有低优先级，那么在网络拥塞时，节点丢弃的只是增强层的数据，接收端仍能收到基本层数据从而得到完整的图像，只不过质量降低一些而已。

MPEG-2 标准中提出了四种分层编码方式：空间分层，时间分层，信噪比分层和数据分割。

1、空间域分层编码

采用空间域分层编码可以提供空间分辨率不同的分层编码码流。MPEG-2 标准支持两层（基本层和一个增强层）的空间域分层编码。原始图像经过亚取样后得到较低分辨率的基本层图像，对基本层图像按一般编码方式进行编码。另一方面，解码后的当前基本层图像经内插构成当前高分辨率图像的预测值。与此同时，像在一般编码器中一样，前一个已编码的高分辨率图像经运动补偿后也产生一个当前高分辨率帧的预测值。在每个宏块进行编码时，选择两个预测值中产生预测误差较小的一个，作为该宏块进行 DPCM 编码的参考（也可以以两个预测值的平均值为参考）。解码器采用与编码器相同的参考宏块，因此出参考宏块与从增强层码流解码得到的预测误差之和，可以正确的恢复出高分辨率图像来。

2、时间域分层编码

采用时间域分层编码的目的是提供帧率在一定范围内变化的分层编码的码流。其具体做法时，首先将原始图像序列沿时间轴进行亚取样，取样后得到的低帧率序列按一般方式编码得到基本流码流，剩余的帧再编码构成增强层码流。构成增强层的帧可以是 I 帧、P 帧或 B 帧。增强层 P 帧的参考帧可以是增强层中的前 1 帧（可能是 I、P 或 B 帧）、基本层中的前 1 帧或者基本层中的后 1 帧。类似的，增强层 B 帧的参考帧可以是增强层前 1 帧和基本层前 1 帧、增强层前 1 帧和基本层后 1 帧或者基本层的前 1 帧和后 1 帧。不用增强层中的帧来作为后向预测的参考帧是为了避免引入过长的延时。用一般的解码器对时间域分层编码产生的基本层码流解码，即可得到低帧率的图像序列。要得到高帧率序列，必须对基本层和增强层同时解码，再将解码后的帧复接到一起，构成与原始帧率相同的序

列。

3、信噪比分层编码

信噪比分层编码提供不同视觉质量的分层编码的码流。基本层的编码方法与一般编码方式相同，只是 DCT 系数的量化器采用了较大的量化台阶，以获得较高的压缩比。单独对基本层码流解码所得到的图像量化噪声比较大。增强层数据所包含的信息在解码时用来增加 DCT 系数的精度，从而达到提高解码图像质量的目的。MPEG-2 标准支持两层的信噪比分层编码。在基本层中，DCT 系数用一个量化台阶比较大的粗量化器量化，然后再进一步用一个细量化器对粗量化后的 DCT 系数和量化之前的实际值之差（即量化误差）进行量化，构成增强层的数据。普通的 MPEG-2 解码器可以对基本层码流解码得到信噪比比较低的图像。要得到高信噪比的图像，必须在解码过程中将增强层携带的 DCT 系数的误差值加到基本层的粗量化 DCT 系数上，再进行 IDCT。

4、数据分割

数据分割是 MPEG-2 所支持的在频率域的分层编码模式，是针对 DCT 系数进行的。每个 DCT 系数矩阵可以划分为几个频带，每个频带包含若干个系数。最低频带内的系数（直流和低频分量）构成基本层码流，其他频带内的系数则分别构成各增强层的数据。当解码器只对频率域分层编码器产生的基本层码流解码时，获得的是一个与原始图像的尺寸相同，但不太清晰的图像，这是因为原始图像中的细节信息在 DCT 中表现为高频 DCT 系数，在只有基本层的情况下这些系数在解码器中被视为 0。如果解码器在做 IDCT 之前将基本层和增强层中所含的 DCT 系数组合在一起，即所有频带的 DCT 系数都没有丢失，那么解码后的图像将有清晰的细节。

比较以上的四种分层编码方式，数据分割分层编码能够在较低的计算复杂性下取得较好的效果。

3.4 视频码流的基础知识和解码过程

编码视频数据由排列有序的视频码流级组成，称之为层。如果仅有一层，则编码视频数据称之为非可分级视频码流。否则，称之为可分级层次。

第一层称为基层（base layer），它总是被单独解码。其他层称为增强层，仅能与低层一起解码（顺序集中的前面层），并且从基层开始。

可分级层次的基层可符合本规范或其他标准，例如ISO/IEC 11172-2，增强层应满足本规范。除数据划分为，基层不包括sequence_scalable_extension()。增强层总是包括sequence_scalable_extension()。

3.4.1 基本语义规则

- 如果序列的第一个sequence_header()后不跟着sequence_extension()，则流（stream）应符合ISO/IEC 11172-2。
- 如果序列的第一个sequence_header()跟着sequence_extension()，则sequence_header()的所有后续出现部分也应紧跟着sequence_extension()。
- sequence_extension()应仅紧接着出现在sequence_header()之后。
- 如果sequence_extension()出现在码流中，则每个picture_header()后应紧跟着picture_coding_extension()。
- picture_coding_extension()应仅紧接着出现在picture_header之后。
- 跟着group_of_picture_header()的第一幅编码图像应为I图像。

除sequence_extension()和picture_header()外，定义了不同的扩展。语法中不同的点所允许的扩展集是不同的。码流中允许扩展的每一点处，可包含来自所定义允许集中任意数目的扩展。但每类扩展不应多次出现。

在解码器遇到一带“保留”的扩展标识的扩展时，解码器应丢弃所有后续数据，知道遇到下一个开始代码。本要求允许定义本规范的可兼容扩展。

- GOP后第一幅图像是一幅I图像。

3.4.2 视频解码过程

解码过程，即解码器应完成从编码码流中恢复图像数据。

除反向离散余弦变换IDCT外，所定义的解码过程要求解码器应产生相同的结果。产生相同结果的任何解码过程应满足本规范。

以下为视频解码过程框图。为了清晰起见，简化了框图。

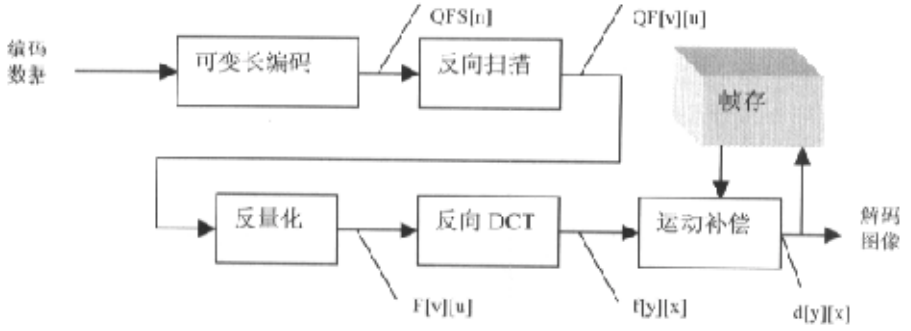


图 3-9 视频解码过程框图

第四章 MPEG 视频压缩技术在 IP 网络上的传输应用----局域网内的视频点播系统

视频点播是二十世纪 90 年代在国外发展起来的，英文称为"Video on Demand"，所以也称为"VOD"。顾名思义，就是根据观众的要求播放节目的视频点播系统。视频点播并没有一个严格的定义，它泛指一类能在用户需要时随时提供交互式视频服务的业务，即“想看什么就看什么，想什么时候看就什么时候看”。它是综合了计算机技术、通信技术、电视技术而迅速发展起来的技术。

视频点播系统主要用于多个用户对网路多媒体文件的共享播放，对提高文件的播放率，以及对硬体储存介质的充分利用，包括多媒体文件的管理都有很多好处。视频点播可以应用于多个领域，如影视欣赏，卡拉 OK，远程教学，远程医疗等等。

4.1 总体性分析

在本节中，我们将分析系统总体架构、硬件设备、软件平台以及传输方式等总体性的选择。

首先，选择客户-服务器模式来实现整个系统是由客户-服务器的特点决定的。客户-服务器模式最重要的特点是非对等作用，即客户相对于服务器处于不平等的地位，服务器拥有客户所不具有的硬件/软件资源和运算能力，服务器提供服务，客户提供请求。客户-服务器模式体现了网络资源信息分布不均等的现象并很好地适应了这种环境。可见，客户-服务器模式的这种非对等作用的特点使其成为网络应用程序相互作用的主要模式的原因。

另一点，网络进程通信与单机进程通信的差别在于，网络通信完全是异步的，相互通信进程之间既不存在父子关系，又不能共享内存缓冲区。因此，需要一种机制为通信进程之间建立联系，为二者的信息交换提供同步。客户-服务器模式完美地解决了上述问题：每次通信均由随机启动的客户进程发起，服务器进程从开机就处于等待状态，这样可以保证服务器随时对客户请求作出响应。另外，

这种请求应答模式为相互通信进程数据传输同步提供了有力支持。因此从技术上说，客户-服务器模式必然成为网络进程间通信的主要模式。

在客户-服务器模式下，一个或更多个客户机和一个或更多的服务器，以及下层的操作系统进程间的通信系统，共同组成一个支持分布计算、分析和表示的系统。在该模式下，应用分为客户机和服务器，分别对应不同的两个应用程序（进程）。客户向服务器发出服务请求，服务器作出响应（一个客户-服务器模式参见图 4-1）。

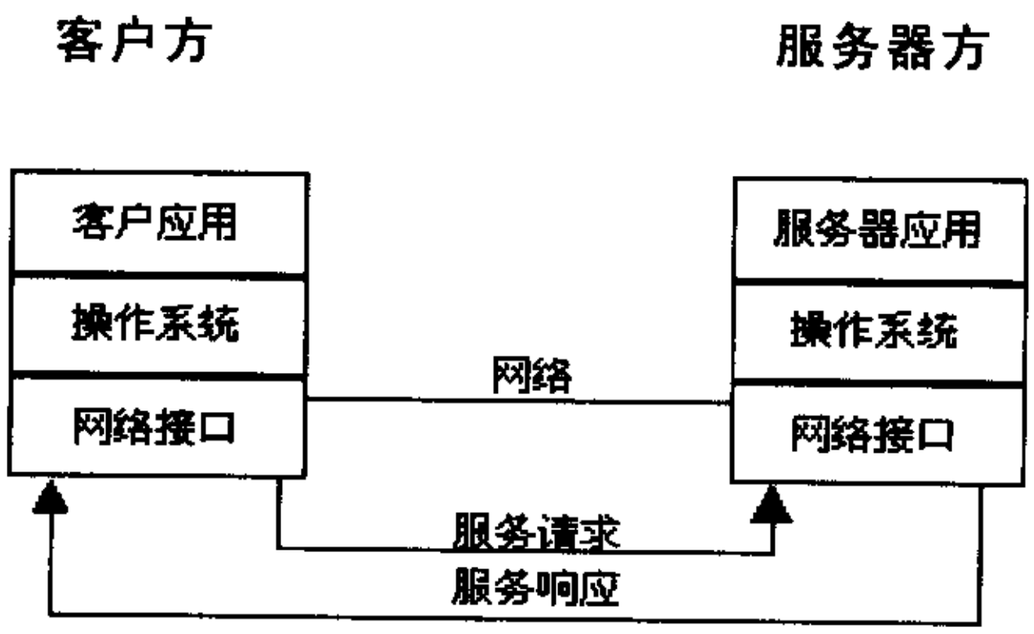


图 4-1 客户机和服务器模式示意图

软件又可以采用传统的客户机-服务器或者基于 Web 的客户机-服务器两种模式。后者的优势在于廉价和快速开发，而缺点是网络传输不理想或者客户端处理能力有限等因素会影响应用程序的速度。如果应用程序可以适应网络框架限制，那么应该考虑 Web 方法。我们采用这种模式来实现我们的方案。

视频点播的终端可以是电视接收机加机顶盒或者个人多媒体计算机。考虑开发单位的现有设备，采用个人计算机作为客户终端。

考虑到客户需求、开发成本以及可重用性问题，我们摒弃了 Windows 平台上微软提供的媒体服务而是采用在 LINUX 平台用 glibc 自主开发整个系统，客户端播放界面采用 Qt 实现。

我们的项目是遵循中华人民共和国通信行业标准 YD/T 1130-2001“基于 IP 网的信息点播业务技术要求”的，在下面两节中，我们介绍标准中与本项目相关的规范。

4.2　视频点播业务的网络组成及各组成部分的功能

根据中华人民共和国通信行业标准 YD/T 1130-2001，基于 IP 网的视频点播业务由信息节点、传送系统（IP 网）、管理系统、导航系统与用户终端系统 5 部分构成，其构成示意如图 4-2 所示。

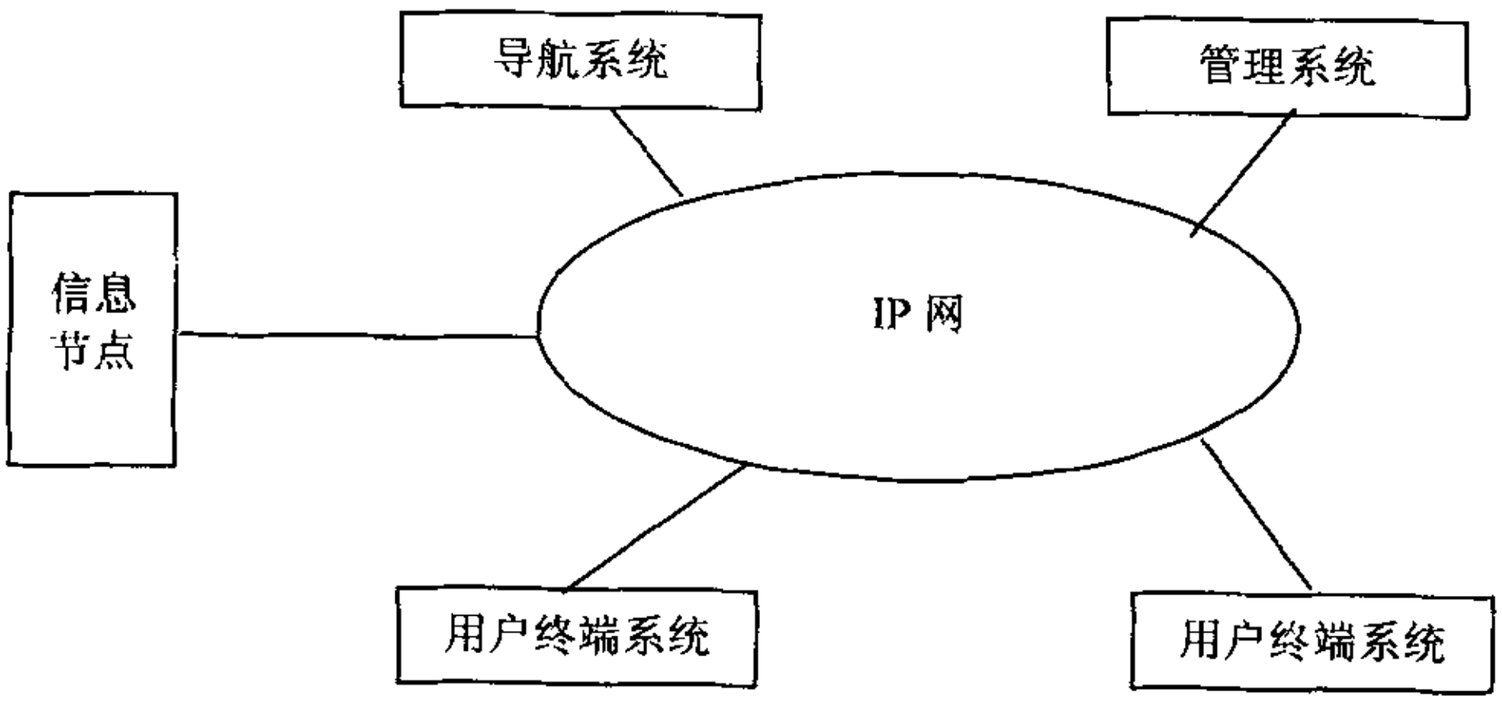


图 4-2　视频点播业务网络组成示意

用户终端作为点播服务的起点和终点，可以发送业务请求信息并显示最终信息。管理系统为信息点播业务系统提供统一的维护和管理平台，管理系统包含认证系统和计费系统。信息节点根据用户请求向用户提供点播信息，并对接入相应信息节点的用户进行二级认证。导航系统为用户提供点播业务系统内各信息节点的目录和地址服务，用户可以据此查找和选择所需信息。

对于局域网内的视频点播系统，只需要用到其中的信息节点、传送系统（IP 网）和用户终端系统三个部分，而对于导航系统和管理系统，仅在广域网中才会用到。因此下面我们仅对这三个部分的功能加以详细介绍。

（1） 信息节点

信息节点是点播业务网向用户提供点播信息服务的系统。主要是由内容子系统、应用服务子系统、通信子系统、认证和计费子系统等组成。其组成示意如图 4-3 所示。

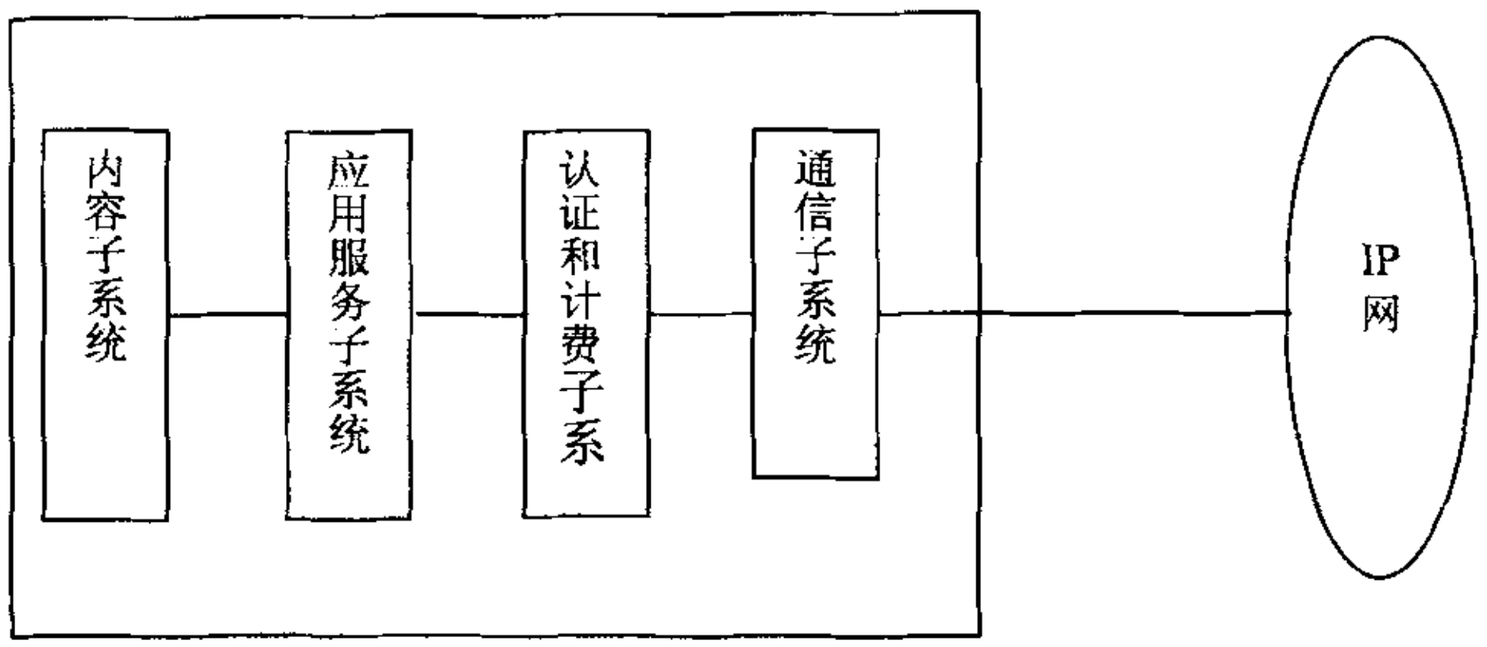


图 4-3 信息节点组成示意

内容子系统主要由视频服务器组成，负责对信息内容的存储和管理，实现对信息内容的采集、压缩和编辑功能。它还可以根据来自应用服务子系统的命令为用户提供点播信息内容。同时，其播放功能可以为用户播放所需信息。在播放过程中，按照用户的要求，提供相应的 VCR 功能，包括选择/删除、启动、停止、暂停、快速前进、后倒、检索快进或后倒等。内容子系统可以由一个或多个物理实体组成，可放置在相同或不同的地点，其中的信息可以来源于其他信息提供系统，如其他网络的信息节点。

应用服务子系统主要负责对信息内容的管理和控制。信息节点在与客户端系统建立快速的传输通道后，应能提供节目检索和服务功能以及应用程序下载功能。

信息节点中的认证和计费子系统主要实现对本地业务的管理，在上级管理系统的要求下，向上级管理系统传送系统管理参数。

通信子系统负责提供与传送网的接口，建立或释放信息节点与用户的双向连接通道。

（2） 传送系统

传送系统负责在用户和管理系统、管理系统和信息节点以及用户和信息节点之间传送内容选择信息、应用下载信息、内容信息、内容控制信息和管理信息。传送系统基于 TCP/UDP/IP 协议。

（3） 用户终端系统

用户终端系统是由用户购置用于节目选择与点播系统交互通信的室内设备，位于点播系统的最靠近用户一端，主要设备是机顶盒和电视机或 PC。

用户终端系统主要由通信子系统、信息解码子系统、信息显示子系统和控制及管理子系统组成。其组成示意如图 4-4 所示。

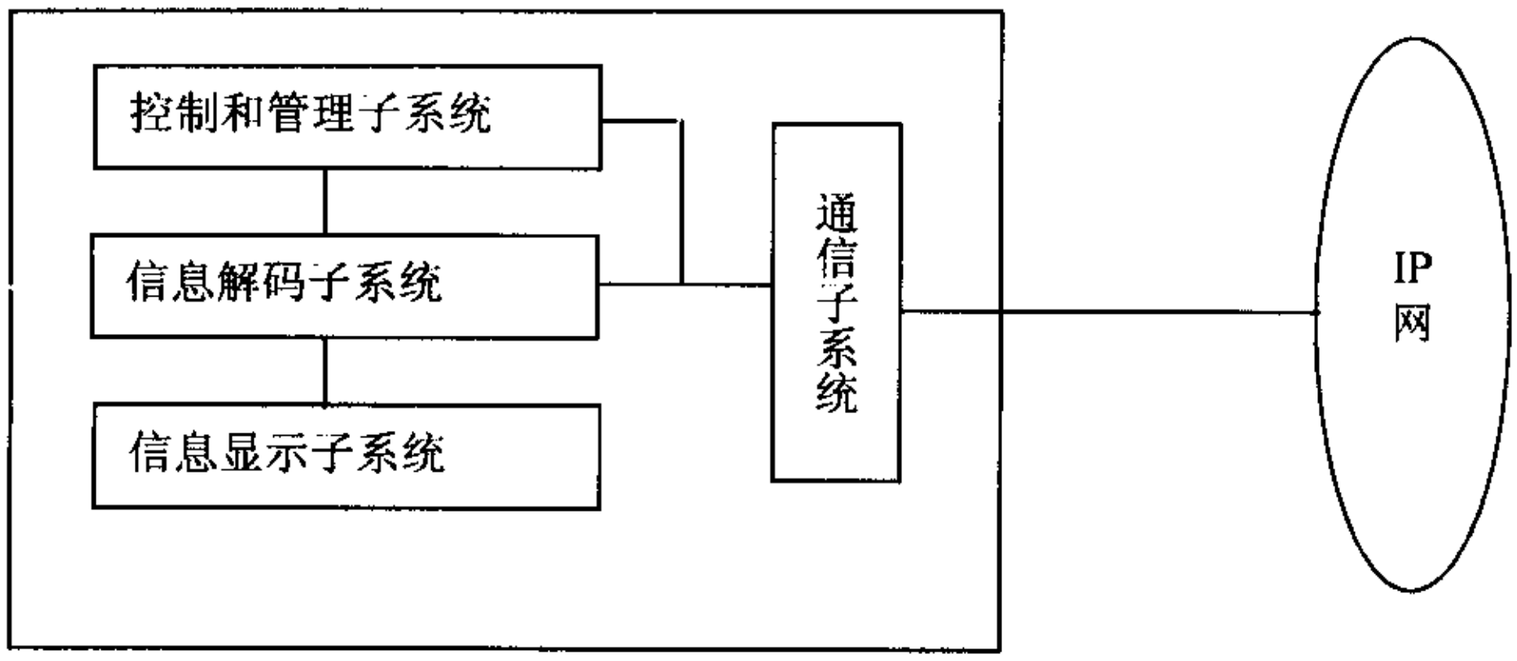


图 4-4　用户终端系统组成示意

控制和管理子系统的功能包括：基本的控制功能，如电源的开/关、点播电视或者标准电视的选择；控制用户的电视和盒式录像机，可输入快进、快退、倒带、播放、暂停、选择、慢放菜单选择和翻页等命令；当用户终端系统为计算机时，采用计算机键盘输入，当用户终端为机顶盒和电视机时，可采用遥控器输入，在机顶盒内应配置相应的软件。

信息解码子系统负责对来自信息节点的信息进行解码，在该子系统中应配置通用的解码软件，同时在信息节点中使用特殊的编码软件时应通过通信子系统和控制子系统将需要的解码软件下载到该子系统中。

通信子系统具有通信功能，完成通信建立、数据传输和通信终止。

信息显示子系统指用户室内设备、网络传输和节目资源的状态，在用户的电视屏幕上显示服务公司以及信息通过者所给出的信息和菜单。

4.3　IP 网上视频点播业务的通信协议

在视频点播业务的实现过程中，通过网络传送的信息流按照功能不同可以分为内容选择信息、应用下载信息、内容信息、内容控制信息和管理信息。

内容选择信息流是用户登录到导航系统和信息节点后，导航系统和信息节点为用户提供内容目录及用户的相应选择信息，该类信息大多为非实时信息流。是否在用户终端和信息节点之间传送该类信息流由用户在使用时选择。该类信息流

对所有的中介实体透明，不引起源对象和目的对象的行为改变。内容选择信息以 HTML 文档的形式存储在导航系统或信息节点处。HTTP 是用户和导航系统及信息节点间所用的协议，在其下层要求使用可靠的面向连接的服务，即 TCP/IP。

应用下载信息流是信息节点向用户下载应用程序时的信息流，该类信息流大多为非实时信息流。是否在用户终端和信息节点之间传送该类信息流由用户在使用时选择。用户在得到二级网关认证之后，可以通过 WWW 服务选择是否进行客户端应用程序下载。如果服务器端采取将播放器嵌入网页中，则用户不需要下载应用程序，只要有浏览器就可以点播节目。

内容信息流为点播系统最终向用户提供的且在用户终端显示的信息。该信息流为从信息节点到用户终端的单向信息流，包括音频、视频信息和/或数据信息，该类信息流大多为实时信息流。该类信息流对所有的中介实体透明，不引起源对象和目的对象的行为改变。当节目源为 MPEG-1 或 MPEG-2 编码方式，在传送时利用实时传送协议（RTP）在 UDP/IP 上传输，并利用实时传送控制协议（RTCP）进行控制。

内容控制信息流是在内容播放时，用户对播放过程的控制信息流，该类信息流大多为实时信息流。当信息节点向用户终端提供服务时，该类节点改变内容信息流，如播放、停止等。内容控制信息的传送方法有实时流协议 RTSP 和 DSM-CC U-U 接口两种，前者在 TCP/IP 上传输，后者需要使用特定的 RPC，然后再在 TCP/IP 上传输。由于我们的实际系统是基于局域网内的，所以对于内容控制信息的误传和丢失的概率较低。考虑到系统的复杂性我们并没有使用上述两种协议，而是使用一套基于 UDP 的自定义消息实现的。具体的消息及实现在下一节中将会详细讲述。

管理信息流是信息节点与管理系统之间以及信息节点与导航系统之间的双向管理控制及数据信息流。该类信息流大多为非实时信息流。该类信息流包括导航信息流、计费信息流等。当信息量较大时，将这些信息存储为文件，在相应站点之间以文件传送协议（FTP）传送，在动态形成这些信息时，使用简单网络管理协议（SNMP）进行传送。

4.4 系统总体方案设计

系统采用基于 Web 的客户机-服务器机制。系统由客户端发起，服务器处于被动状态，等待客户的各种请求，然后通过消息触发和消息响应机制来完成它的工作。软件应用的主要过程是这样的：首先，用户通过 IP 地址或者域名向服务器发出连接请求，服务器向客户机发送用户名和密码文本输入框，用户在输入框中输入用户名和密码。待用户返回信息后，服务器检查当前用户是否已满并进行身份验证，如果未满而且是合法用户，则向该用户发送节目目录；否则，返回拒绝请求页面。用户从收到的节目目录中选择节目进行点播，在点播过程中可对节目进行暂停，快进，倒带等交互操作。当用户要断开连接，服务器返回该用户本次连接的费用。系统现在能支持 MPEG-1，MPEG-2 节目流。

- 系统组成基本模型：

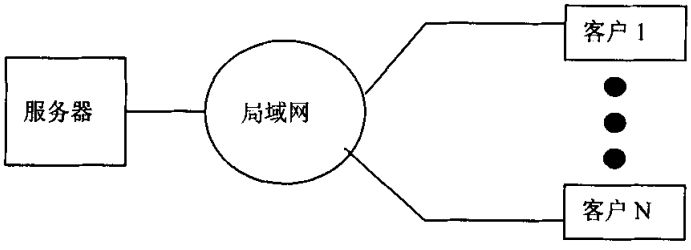


图 4-5 系统组成基本模型

- 外部接口基本模型：

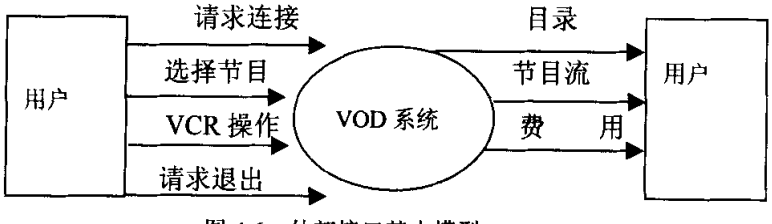


图 4-6 外部接口基本模型

一般约束

A. 运行环境：

（一） 服务器：Pentium III500以上CPU，128M以上内存，20G以上SCSI硬盘。快速以太网适配器或千兆位以太网适配器。

（二） 客户机：标准配置Pentium II 266以上CPU，128M以上内存。图形适配器支持256色显示，分辨率800*600以上。10M以太网适配器或快速以太网适配器。VGA显示器。声卡，与其相连的外围设备包括扬声器或耳机。

（三）操作平台：要求服务器端程序运行在LINUX2.4.x操作平台上；客户端程序可以运行在LINUX & Windows平台上。

（四）开发平台：整个系统程序采用glibc2.1库，LINUX 2.4.x内核，用gcc 2.95编译。基于对象的分布式计算环境和分布式数据库，采用免费的MySQL数据库，用户界面基于Qt。

B. 并行操作的用户数：

考虑最大点播用户数主要需要考虑三方面的因素：网络带宽，硬盘存取速率，以及线程占用内存。

假设用户点播MPEG-1与MPEG-2格式编码节目的概率大致相等，则从统计上讲，平均码流速率约为3Mbps。

如果单考虑网络带宽的限制。如果成本有限制，采用快速以太网适配器，网卡接口为100M。则在单服务器情况下，可允许30个用户同时点播节目，点播同一节目的最大并发用户数为30。如果所有节目均是MPEG-1编码格式，平均码流速率约为1.5Mbps，则单服务器情况下可允许60个用户同时点播节目。而如果采用千兆位以太网卡，则单服务器情况下可允许300个（MPEG-1和MPEG-2都有的情况）或600个（只有MPEG-1的情况）用户同时点播节目。

如果单考虑硬盘存取速率的限制。服务器采用SCSI160高速硬盘，速率可以达到160Mbps，则在单服务器情况下，可允许50个（MPEG-1和MPEG-2都有的情况）或100个（只有MPEG-1的情况）用户同时点播节目。

综合考虑两方面的限制，则在考虑低成本的情况下，采用快速以太网卡加SCSI160硬盘，瓶颈在于网络带宽，可允许30个或60个用户同时点播节目。规定每个用户在同一时刻只允许点播一个节目（即对每个用户只能开一个专用的用户子线程）。

要想提高最大点播用户数指标，可以采用千兆位以太网适配器或者采用多服务器集群。

C. 对实时性的最低要求：

要保证用户方连续播放，并且等待时间在用户可容忍范围内：

用户发出连接请求之后，10秒之内返回节目目录或非法用户对话框

用户发出节目选择之后，10秒之内开始播放正常节目流

用户发出暂停请求之后，2秒之内冻结最后画面

用户发出快进或倒带请求之后，2秒之内开始播放快进或倒带节目流

用户发出继续播放请求之后，2秒之内开始播放正常节目流

用户发出停止请求之后，10秒之内返回节目目录

用户发出断开连接请求之后，10秒之内返回本次连接时间和费用

D. 安全保密方面的考虑：

一方面，考虑网络安全性：当用户请求连接时的用户名和密码，以及当用户要退出时服务器下传的本次连接费用要加密之后再在网上传送。另一方面，考虑资料的安全保密性，各个数据库要设置访问权限，不允许随便访问和修改。

用户界面需求

A．界面需求

服务器端显示当前在线用户列表。包括用户名，用户正在点播的节目，以及当前状态。

客户机的界面包括：

1 主页面。用户通过IP地址或者域名发出请求后，服务器向用户方下传的页面包括用户名文本输入框，密码文本输入框，登录按钮。当用户提交上述信息后，服务器判断当前用户是否已满。

2 目录页面。服务器确定用户合法后向用户下传的页面。包括可供选择的节目名列表，以及一个退出按钮。

3 播放视频窗。包括暂停，快进，倒带，继续播放，停止五个按钮，一个显示视频窗口。

4 正常结束页面。用户正常退出后，服务器向用户下传的页面。显

示本次连接的时间和费用。

B．输出错误信息的格式：

1．拒绝请求页面。用户通过IP地址或者域名发出请求，当服务器判断当前用户已满，向用户方下传的页面。显示：当前用户已满，请稍后重试。

2．非正常结束页面。如果在特定状态下限制时间内没有收到某个用户的任何消息，就释放同该用户的连接，并向用户方下传本页面。显示：没有响应，已断开连接。并显示本次连接的时间和费用。

注：特定状态下限制时间内是指：

正常播放，快进或倒带状态下：1秒之内（没有收到周期请求消息或暂停或停止消息）。

选择节目状态下（包括刚建立连接时以及停止后再次下传目录时）：5分钟之内（没有收到节目选择消息）。

软件属性需求

A． 正确性需求：要求程序满足该软件需求说明书提出的要求。要求满足需求说明书中的功能需求和界面需求。

B． 健壮性需求：在不合理的输入情况下，程序返回给用户错误对话框，程序仍可继续运行。

C． 安全保密性需求：包括两方面的内容：一是资料安全保密性的考虑，对各数据库进行权限管理，不允许随便对它们进行访问及修改。以ROOT方式运行Web服务器。二是网络安全性的考虑，服务器和客户之间传送的数据和控制信息（用户名，口令等）要加密后在传送，对方收到后要有相应的解密处理。

D． 易使用性需求：图形界面，简洁明了，便于理解和操作。包括三个正常页面（主页面，目录页面，正常结束页面），一个视频窗（用于播放节目，包括交互操作按钮），两个异常页面（拒绝请求页面，非正常结束页面）。客户端程序附在线帮助。

E． 可理解性需求：模块化编程，程序中添加注释。在客户机程序中含在线帮助。

F. 可维护性需求：模块化编程，易于维护。

G. 可测试性需求：模块化编程，可分模块进行测试。

满足以上三点需求的关键在于如何模块化编程。按照功能需求的功能模块划分进行编程。

H. 可移植性需求：可移植性需求主要体现在两个方面，一是操作系统方面的可移植性，二是方案选择方面的可移植性。可移植性需求主要通过软件分层（soft layering）来实现。软件纵向的分为三层：应用程序请求层，库接口层和依赖系统层。上面两层是完全独立于底层操作系统的，在移植到其它操作平台（如安全性强的 UNIX，完全免费的 LINUX 等）时，仅需对第三层进行修改。另一方面，目前我们采用的是在计算机以太网上运行，如果将来软件要通过 ADSL 技术运行或通过 Cable Modem 在有线电视网上运行，上两层也不需要变动。同时，软件设计中的客户终端——个人计算机也可以由监视器加机顶盒加遥控器来代替。

客户端放软件功能需求

（一） 用户的接入与退出

A. 用户接入

输入数据：键盘输入用户名，密码；鼠标事件——连接

输出数据：节目目录或失败对话框

B. 用户退出

输入数据：鼠标事件——退出或关闭程序

输出数据：本次连接费用

（二）用户对节目的 VCR 交互操作

A. 选择节目

输入数据：键盘输入节目 ID 号

输出数据：正常播放的 MPEG-1/MPEG-2/MPEG-4 节目流

B. 暂停

输入数据：鼠标事件——暂停

输出数据：NULL

C. 快进

输入数据：鼠标事件——快进

输出数据：只含奇数I帧的MPEG视频流

D. 倒带

输入数据：鼠标事件——倒带

输出数据：只含奇数I帧的倒序MPEG视频流

E. 继续播放

输入数据：鼠标事件——继续播放

输出数据：正常MPEG节目流

F. 停止

输入数据：鼠标事件——停止

输出数据：节目目录

系统运行体系结构

本系统是基于WEB的分布式计算对象的三层体系结构。见图4-7。

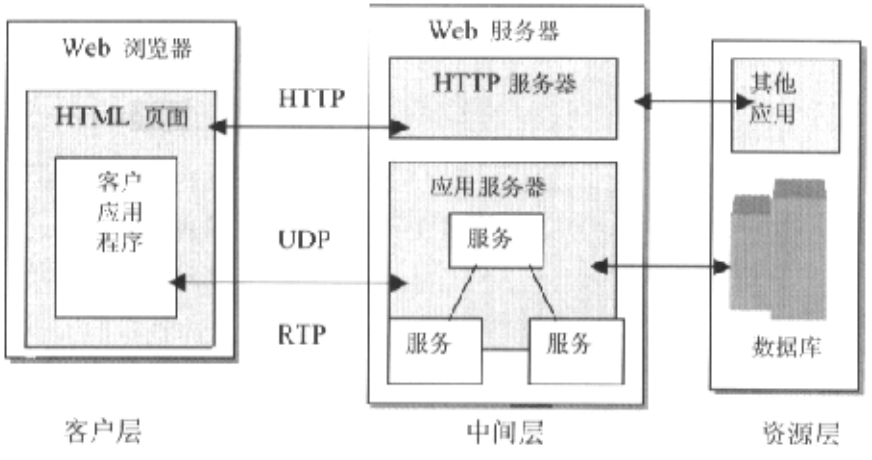


图4-7 基于WEB的分布式计算对象的三层体系结构

本系统采用客户/服务机模式，但是现在还未能将客户应用程序嵌入到HTML页面内。图4-8给出了功能框架的三层结构图，具体内容是：内容服务器、编码器和节目转换器、媒体服务器、服务器端网页和信息数据库和媒体数据库。

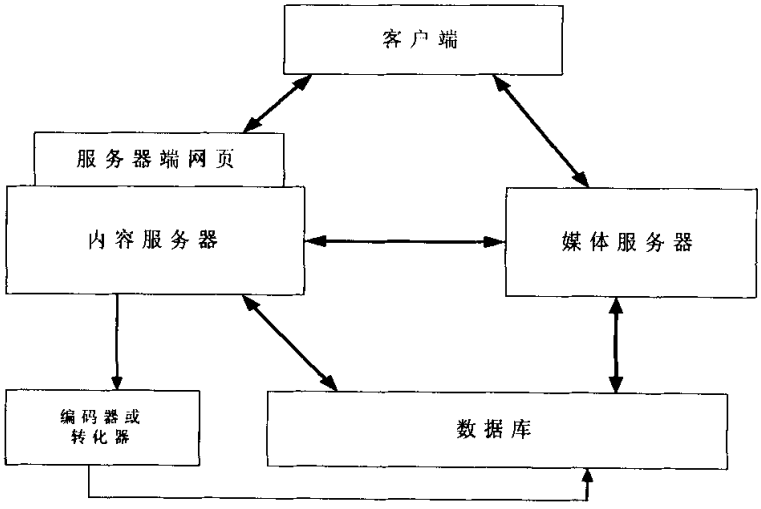


图 4-8　视频点播系统框架结构

各个模块的具体功能如下：

I　服务器端网页要完成的工作包括：

设计各种网页（如节目菜单等）以供客户端点击浏览，接受客户端的请求，并将客户端的请求提交给内容服务器，经过服务端程序处理以后，将处理结果返回给客户端,如果是点播请求，则必须将内嵌有播放器界面的网页传递给客户端。播放器界面有播放，快进，倒带和暂停按钮。此外，还有节目列表框，供选择和搜寻节目。

II　内容服务器要完成的任务是：

监听来自客户的请求，然后调用数据库中的信息来验证此用户的身份，对于合法的用户，将此用户的信息记录进信息数据库；然后给媒体服务器发消息（此消息包括包含有此特定客户的信息），告诉媒体服务器可以为此用户服务，此后的服务交给媒体服务器来完成（即此后客户和媒体服务器直接联系），如果断开连接，媒体服务器应该将此信息告诉内容服务器。

内容服务器应该能够同时监听多个用户的请求，完成对它们的身份验证，并记录用户的信息。此外内容服务器要监控系统负载状况，实现

软件的升级管理，广告信息管理。如果需要添加节目，还要启动编码器或节目转换器将 A/V，DVD，VCD 节目转换成 MPEG 码流，然后放入媒体数据库中。

III 编码器和节目转换器要完成的工作：

搜集各种视/音频节目，然后将其转化成 MPEG 码流，并将其放入媒体数据库中。

IV 媒体服务器的功能如下：

当收到内容服务器发来的为客户服务的消息（包括客户的地址）后，媒体服务器从媒体数据库中读取相应的节目发送到给定的用户；此后媒体服务器直接与客户交互，当收到暂停，快进，倒带和继续播放时，完成相应的操作；当收到断开连接时，必须与给定的用户断开，然后关闭此进程。并且将此情况告诉给内容服务器。此外还要完成广告流的管理，提供负载信息。

V 信息数据库和媒体数据库的工作是：

建立和维护（比如添加，删除和修改等）信息数据库和媒体数据库，主要是有关用户的登陆信息（包括密码，帐户和点播时间信息），节目目录和媒体内容。并且内容服务器随时可以同信息数据库建立联系，向数据库中写入数据或从中读取数据；另外，媒体服务器操纵媒体数据库，从库中找到相应的节目，进行读取操作，直到断开连接时，与数据库断开。

本人在项目中的主要任务是设计和实现客户端，在下一节中具体说明对于客户端的总体设计和相应实现流程。

4.5 客户端

客户端可以分为两大部份：客户端接口模块和播放模块。先由客户端接口模块解开 RTP 包，再把数据交给播放模块进行操作。

4.5.1 客户端接口模块

这个模块的主要作用是提供与远端服务器交互的高效简洁的接口，对播放模

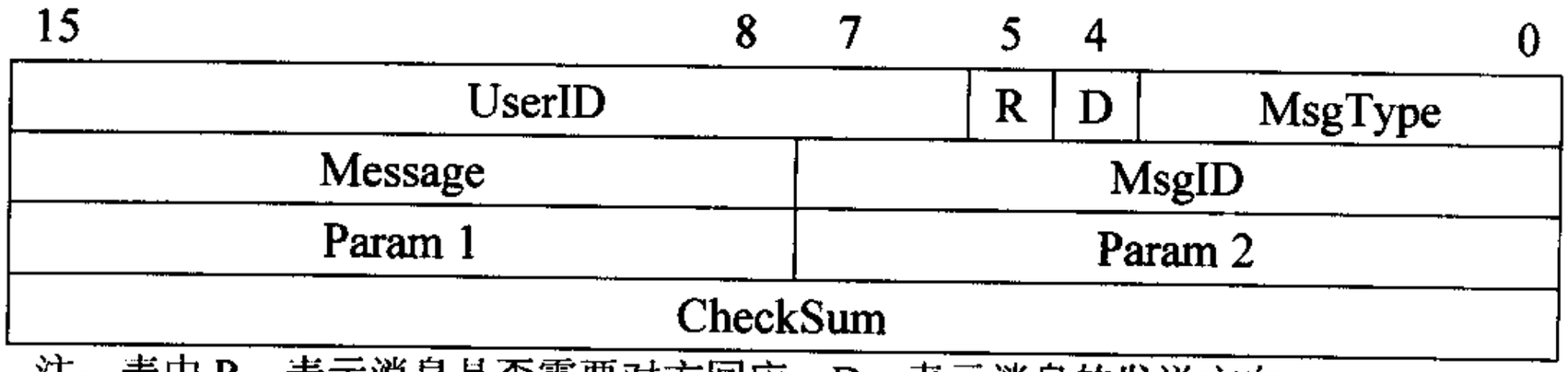
块屏蔽网络的通讯机制，与媒体播放器中的接口模块相对应。在这个模块中使用定义好的媒体服务器和远程客户端的交互过程和交互消息，建立了一套请求和响应的机制，但同时又对客户端内部屏蔽了这些过程。

由于接口模块的隔离作用，对客户端其他部分来讲，系统相当于本机用户的视频播放，不必考虑与网络有关的通信协议等问题，只要发送请求消息并接收媒体服务器的响应就可以了。

客户端接口模块需要完成：1、遵从通信协议的网络消息和可懂的本地指令之间的转换；2、把 RTP 分组包还原成 MPEG 码流；3、屏蔽网络的不稳定性。

本模块所做的主要工作包括：建立包括错误重传在内的消息纠错机制；与服务器进行必要的交互并接收服务器传来的各种消息；向服务器发送周期性 RTP 请求消息，并把接收到的 RTP 分组包还原成 MPEG 码流。

通过上面一些简单的功能描述可以发现在客户端接口模块中我们定义了很多的消息，这些消息基本是为服务器和客户端的交互而设置的，属于内容控制信息，出于简化系统的目的并没有使用实时流协议 RTSP 和 DSM-CC U-U 接口，而是在 UDP 的基础上直接传输控制信息并处理重要信息的错误重传。图 4-9 和表 4-1 分别是消息的格式和具体消息列表。



注：表中 R：表示消息是否需要对方回应；D：表示消息的发送方向

图 4-9　客户端接口模块消息格式

表 4-1　与服务器端交互过程中用到的消息

消息类型	所包含的消息		参数一	参数二	消息功能简单描述
	下行 D=0	上行 D=1			
基本播放控制消息（0H）	/	播放 01H	/	/	这部分消息是视频点播系统所必需的也是最常用的消息，它不需要媒体服务器确认响应。（R=0）
	/	暂停 02H	/	/	
	/	快进 03H	/	/	
	/	倒带 04H	/	/	
	/	选择进05H	/	/	
	/	选择倒06H	/	/	

消息类型	所包含的消息		参数一	参数二	消息功能简单描述
	下行 D=0	上行 D=1			
	/	继续 07H	/	/	
	/	停止 08H	/	/	
	/	RTP 请求 09H	/	/	
快速定位控制消息（1H）	/	获取当前媒体流的分段数 01H	/	/	用于获得初始的媒体流的分段数目，由于不属于重要消息所以不需要对方的接收确认响应。（R=0）
	通知媒体流的分段数目 01H	/	广告部分的分段数	节目部分的分段数	
	/	快速定位到指定节目段 02H	指定段的编号	/	媒体服务器改变 RTP 打包模块所用参数，使 RTP 打包模块从指定节目流位置开始打包。（R=0）
	/	在当前位置设置标签 03H	要设定的标签号（0..15）	/	由于此消息是否成功用户无法感觉到从而可能会影响用户后续操作，所以需要接收确认响应。（R=1）
	/	快速定位到指定标签处 04H	指定的标签标号（0..15）	/	媒体服务器改变 RTP 打包模块所用参数，使 RTP 打包模块从指定标签位置开始打包。（R=0）
	/	快速定位到下一节目段 05H	向前或者向后定位（0/1）	/	用于下一段功能。（R=0）
用户交互消息（2H）	/	进入节目段预览状态 01H	/	/	用于通知媒体服务器播放当前节目的段预览。方便用户选择节目段。（R=0）
	要求用户做出选择 02H	/	选择范围开始值	选择范围结束值	用于服务器对用户做出询问，可用于广告调查等情况中。例如三个选项，选择范围可以是 0~2 或者 1~3。（R=1）
	/	反馈用户的选择 02H	选择的结果		
状态交互消息（3H）	客户端是否仍然存在 01H	/	/	/	在用户停止了请求 RTP 包的时候，可能是因为客户端故障造成的，为了不浪费服务器资源在长时间没有响应的情况下应该释放此用户所占用的资源。（R=1）

消息类型	所包含的消息		参数一	参数二	消息功能简单描述
	下行 D=0	上行 D=1			
	客户端复位消息 02H	/	/	/	在用户选择新的节目后用户端应该复位到原始状态，重新请求菜单等信息。（R=1）
	节目播放状态变化，改变 RTP 请求状态 02H	/	关闭或者重新打开对 RTP 请求的响应(0/1)	/	为了减少不必要的网络传输以及处理，设置了此消息，用于在等待用户响应以及节目播放结束的时候停止客户端的 RTP 请求。当参数一为 0 时，不处理该用户发来的 RTP 请求（R=1）
	/	获取本次已收看的广告时间 03H	/	/	用于用户查询本次点播的时间统计信息。时间的单位为分钟。
	反馈广告时间 03H	/	播放时间高 8 位	播放时间低 8 位	
	/	获取本次已收看的节目时间 04H	/	/	
	反馈节目时间 04H	/	播放时间高 8 位	播放时间低 8 位	
正确接收消息确认（FH）	上行消息正确接收 FFH	下行消息正确接收 FFH	所接受的消息 ID	/	对重要消息的传输进行确认，一段时间如果没有收到确认应从新发送相同消息。对以上列表中 R=1 的消息，如果本表中没有定义对端对该消息的反馈消息类型，就需要对端发送本消息进行确认反馈。

4.5.2 RTP 协议简介

实时传送协议(Real Time Protocol)提供具有实时特征的、端到端的数据传送服务，可以用来传送声音和运动图像数据。在这项数据传送服务中包含了装载数据的标识符、序列计数、时间戳和传送监视。

在 RTP 协议中，没有具体限定底层网络所能提供的传输服务的类型，通常情况下，RTP 协议与 IP 协议栈相结合，运行于 UDP 之上（如图 4-10 所示）。RTP

仅使用 UDP 提供的复用和校验服务。

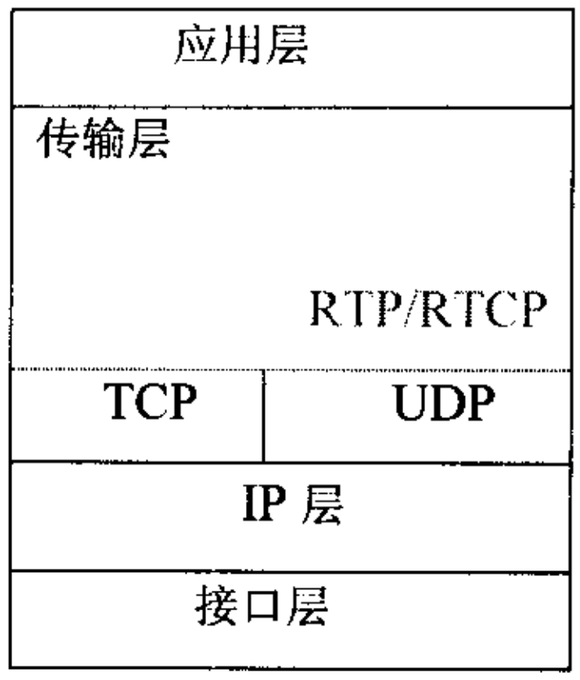


图 4-10 RTP/RTCP 协议在 TCP/IP 协议栈中的位置

由于 UDP 对于报文丢失，次序颠倒等数据传输过程中的差错几乎不做任何处理，而 RTP 本身也不保障对于上述错误的更正，因而接收端应用需要利用固定头中所含的次序号来检测是否存在 RTP 报文的丢失和次序的颠倒。当发现上述两类错误时，应用程序可以进行差错恢复，如要求发方重新传输相应的报文；也可以根据实时数据所采用的编码方式的特点忽略错误。 RTP 报文的次序号也可以用来表述实时数据的定时关系。

RTP 提供多址功能，使用时间戳方式提供在一个源和多个目的之间的同步。如果支持它的网络能提供组播(multicast)功能，则 RTP 也可用组播将数据送给多个目的用户。RTP 不保证数据包按序号传送，包含在 RTP 中的序号可供接收方用于重构数据包序列，也可用于包的定位。

RTP 主要包含了两部分内容，其一为数据报文格式的定义及其使用规则；其二为控制协议，即 RTCP。RTP 传输服务使用者之间的连接被称为 RTP 会话，就每一个会话参加者而言，会话由一对传输层地址（即一个网络层地址加上两个端口地址，一个端口为 RTP 报文的发送/接收所占用，另一个端口为 RTCP 报文的发送/接收所占用）标识。在进行多媒体数据通信时，不同媒体类型数据由不同的 RTP 会话传输。与 RTP 一起使用的控制协议 RTCP 在参加者之间传送控制信息。

一个标准的 RTP 报文是由固定头(fixed header)和用户数据或称数据负载(payload)两部分构成的，其报文结构如图 4-11 所示：

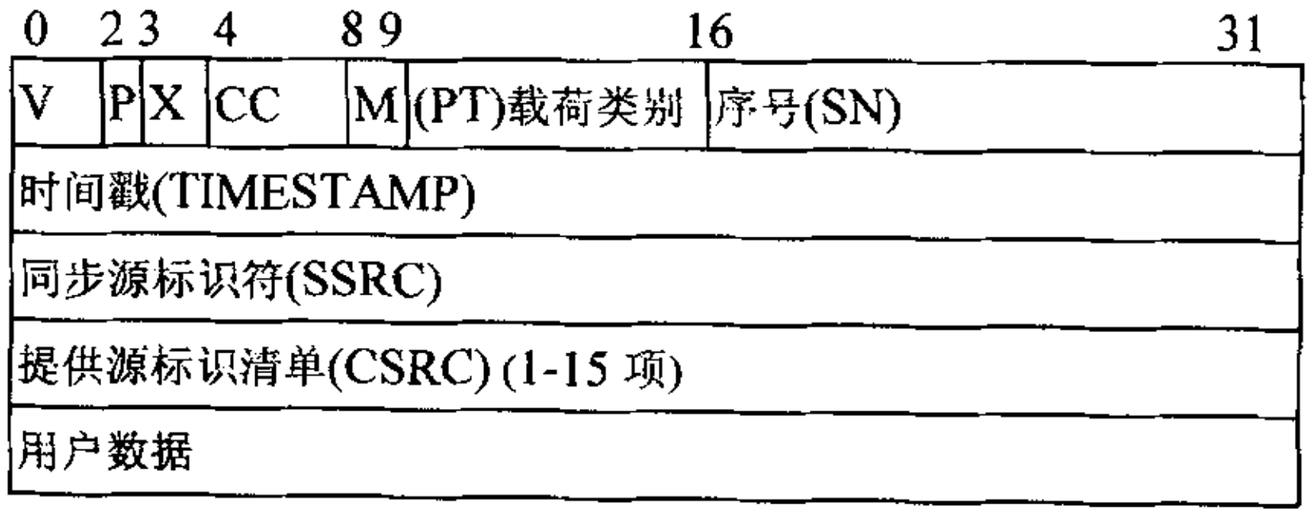


图 4-11 固定报头的 RTP 报文

注：图中各字段的含义如下：

版本(V)：这是一个长度为 2 比特的字段，表示本 RTP 的版本号。

填充(P)：这是一个长度为 1 比特的字段，该位置“1”表示在用户数据字段的最后加有填充位，用户数据字段中的最后一个字节(8 位组)是填充位计数，它表示一共加了多少个填充位。

扩展(X)：这是一个长度为 1 比特的字段，该位置“1”表示 RTP 报头后紧随有一个扩展报头。

CSRC 计数(CC)：这是一个长度为 4 比特的字段，它表示在定长的 RTP 报头后的 CSRC 标识符的数量。

标记(M)：这是一个长度为 1 比特的字段，标记的具体解释由轮廓值来定义。

载荷类别(PT)：这是一个长度为 7 比特的字段，它指示在用户数据字段中承载数据的载荷类别。

序号(SN)：这是一个长度为 16 比特的字段，每发送一个 RTP 数据包该序号增加 1。该序号在接收方可用来发现丢失的数据包和对接收到的数据包进行排序。

时戳(TS)：这是一个长度为 32 比特的字段，它用来表示 RTP 的数据字段中第一个字节的采样时刻。时戳的时间表示应为线性单调递增的，以便完成同步实现和抖动的计算。如果若干个 RTP 数据包的数据是同时产生的(如：一帧图像)，那这几个 RTP 会有相同的时戳值。

同步源标识(SSRC)：这是一个长度为 32 比特的字段，它用来标识一个同步源。

提供源标识清单(CSRC)：它可以有 0——15 项标识符，提供源标识项数由 CC 字段来确定。每一项标识符的长度为 32 比特。

4.5.3 MPEG比特流的RTP净荷格式的制定

RTP 协议的一个重要特点即是它针对不同的应用，定义了多种不同的格式文件和轮廓文件。具体地说，针对不同格式的多媒体数据码流，在使用 RTP 协议形成 RTP 纯负载时，有不同的打包方案。针对 MPEG 格式的数据，RTP 又定义了数种方案可供选择。

不论是哪一种方案，有一点是共同的，考虑到当前 IP 网络的特点和实时数据流的传输要求，主要是为了提高效率起见，RTP 协议特地规定最后形成的 RTP 分组大小最好不要超过下层网络的最大传输单元（MTU）。对于 IP 网来说，MTU 的典型值为 1536 字节。

目前通用的 RTP 净荷格式是 AT&T 实验室的 D. Hoffman 等人提出的 RFC2250 标准，它是 RFC2038 标准的修订版。该标准的基本思路是，把视频和音频数据分别打成 RTP 分组包，然后从两条路径分别传送，按照该标准，无论打包过程还是传输接收过程，视频数据和音频数据都是相对独立的。该标准提出了三点码流切割规则：

- 如果出现 MPEG 视频序列头，必须在 RTP 分组包净荷的开头。
- 如果出现 MPEG 图组头，必须在 RTP 分组包净荷的开头或者跟在视频序列头后。
- 如果出现 MPEG 帧头，必须在 RTP 分组包净荷的开头或者跟在图组头后。

在服务器和客户终端，均需要四个端口。在服务器端，两个端口用于连续的（仅经过 MPEG 标准压缩的）视频和音频数据的输入，两个用于打成 RTP 分组包后的（每个分组包都加上了头信息的）视频和音频分组包的输出；而在客户端，两个端口用于接收到的视频和音频 RTP 分组包的输入，另两个用于重新组装后的视频和音频数据的输出，然后根据 RTP 分组包头中的时间戳，将视频和音频进行同步，就可以进行下面的解码和播放操作了。它的优点在于视频和音频数据相对独立，因而其中某一项数据的错误不会对另一项造成很大影响；缺点是实现比较复杂，因为需要双倍的端口和传输线路。

我们以 RFC2250 标准为基础，参照由 AT&T 实验室的 M. Civanlar 等人提出的 RFC2343，提出了自己的打包方案。它主要是增加了以下原则：

（1） 每个视频 RTP 分组包中必须包含整数个片；

（2） 每个音频分组包中放且只放一个完整的音频帧；

（3） RTP 净荷的头信息中第一个比特（在 RFC2250 中为保留位未使用）标识净荷类型。

（4） 将打包后的视频和音频 RTP 分组包放在一个流中在网上传送，这样可以节省一半的端口

总结一下，按我们的方案，打包应遵循以下原则：

• 如果出现 MPEG 视频序列头，必须在 RTP 分组包净荷的开头。

• 如果出现 MPEG 图组头，必须在 RTP 分组包净荷的开头或者跟在视频序列头后。

• 如果出现 MPEG 帧头，必须在 RTP 分组包净荷的开头或者跟在图组头后。

• 如果出现 MPEG 片头，必须在 RTP 分组包净荷的开头或者跟在帧头后。

• 每个音频 RTP 分组包中包含一个完整的音频帧。

可以看出，在新增了自定义原则之后，仍然满足 RFC2250 标准，因此只要接收端的拆包方法遵循 RFC2250 标准，就一定能够识别我们按自定义方案打包发送的 RTP 分组。

经过上一步的工作，RTP 净荷已经形成，因此下一步的工作就是在各段净荷前再加上 RTP 固定报文头，以形成一个 RTP 分组。

为了方便起见，我们用如下的数据结构来表示一个 RTP 分组：

```
typedef struct {
    char            *data;          //指向存放 data 的指针
    int             data_len        //纯负载长度
    char            *extn;          //指向存放头标扩展区域的指针
    int             extn_len;       //头标扩展的长度
    unsigned short  v:2;            //packet type
    unsigned short  p:1;            // padding flag
    unsigned short  x:1;            // header extension flag
    unsigned short  cc:4;           //CSRC count
    unsigned short  m:1;            //marker bit
    unsigned short  pt:7;           //payload type
    short           seq             // sequence number
```

```
    u_int32        ts          // timestamp
    u_int32        ssrc        // synchronization source
} rtp_packet;
```

在加头过程中需要注意以下几点：

• RTP 报文头中的 PT 值是由第一步所形成的净荷头部中的第一位决定的。PT 值不得等于 200，201，202。这些常数是留给 RTCP 分组的。在本程序中，视频 RTP 分组包的 PT 值为 120；音频 RTP 分组包的 PT 值为 121。

• 在 RTP 协议中，为了使接收端重建分组序，并为了保密起见，在每个 RTP 分组的头标中都有一个序号域。这个序号初始是随机产生的，以后每个 RTP 分组加 1。

• 时间戳是 RTP 报文头中非常重要的部分。正是通过这个域保存了报文中多媒体数据的时域信息，使得在客户端可以将这些多媒体数据组织起来并重建其同步关系。在本程序中， RTP 包中的时间戳代表发送端打包时的系统时间；在对 RTP 包进行接收时，可以使用该时间来近似获得网络的延时抖动(假设发送端和接收端时钟速率相等)。方法如下：假设第 I 个包到达时客户端的系统时间为 C_I，该包的时间戳为 T_I，发送端和接收端的时钟偏差为 R，则网络延时为 $D_I = C_I - T_I + R$。

由上所述，在网络层传输的一个完整的 IP 层数据分组如图 4-12 所示：

UDP Header	RTP Header	负载 Header	PAYLOAD

图 4-12 RTP 分组格式

4.6 客户端播放模块

客户端播放模块是客户端最主要的模块，完成对码流解码播放的功能。数据流在经过客户端接口模块后，RTP 分组包被拆开还原成 MPEG 数据流。与编码系统相应，解码系统要完成对到达的多路数据流的语法分析和分解，并完成解码和基本数据流的显示。一个典型的能完成对 MPEG 码流进行解码的参考框图如图 4-13 所示：

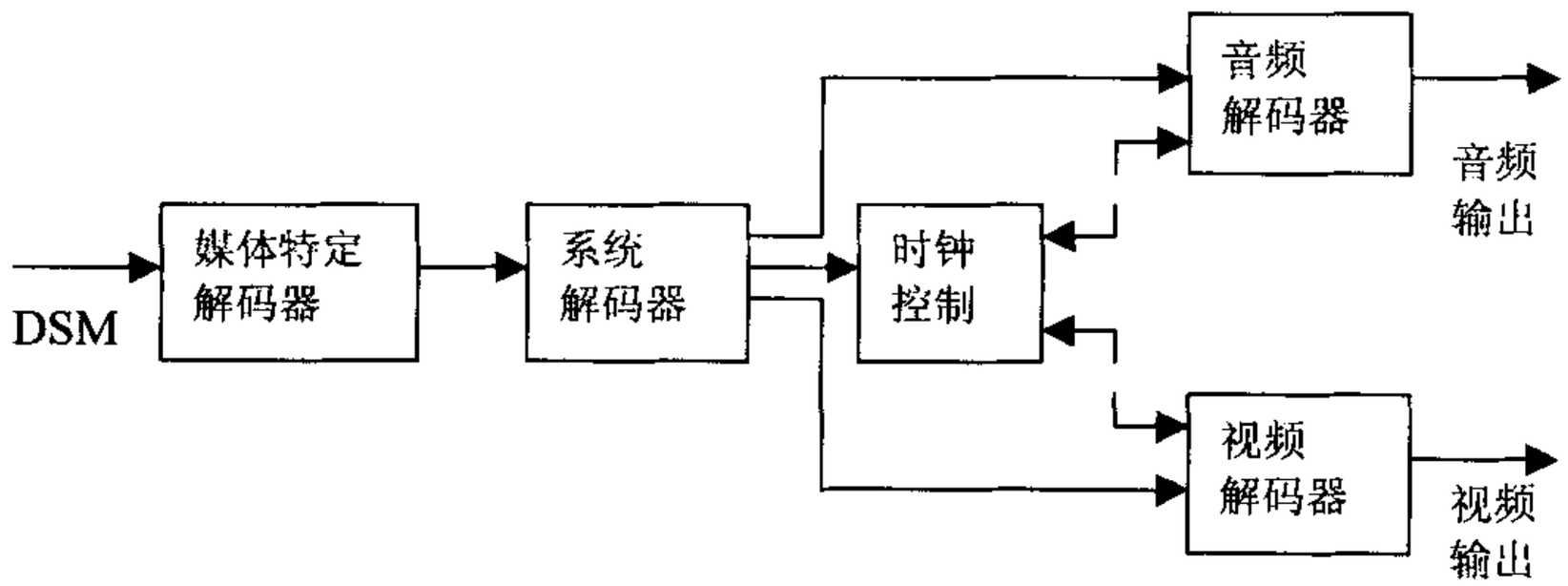


图 4-13　MPEG 解码器参考框图

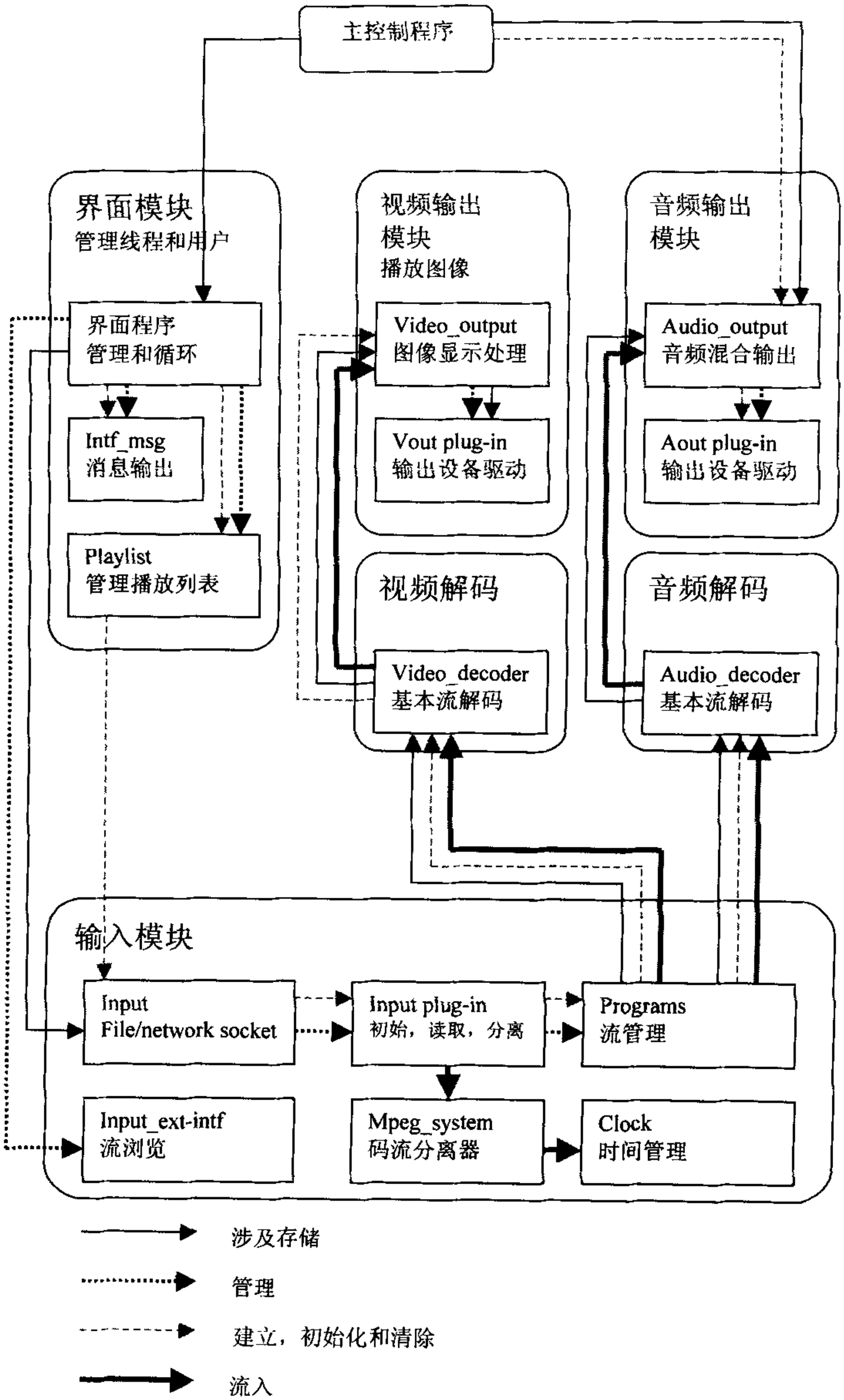
4.6.1　子模块划分

客户端播放模块分成若干子模块和插件。一个模块由一组完成一定功能的 C 程序组成。在编译的时候，各模块和插件会被链接到主应用程序上。各模块分化如下：

- 界面模块：这是整个程序的进入点。它管理所有用户的交互操作和线程的产生。
- 输入模块：它负责开启输入 socket，读取包，解析包，并把重新生成的基本流传递给解码器。
- 视频输出模块：它负责初始化视频显示。然后从解码器中得到画面，把它们从 YUV 格式转化成 RGB 格式，显示在屏幕上。
- 音频输出模块：它负责音频设备初始化，找到正确的播放频率，然后对从解码其中得到的音频帧进行重新取样。
- 杂项模块：在其他模块中的各式各样的应用。这是唯一一个不专门启动线程的模块。
- 解码器模块：包括对各种不同基本流数据进行解码的解码器，主要是视频解码器和音频解码器。

插件放在/plugins 子文件夹中，在运行的时候被装载。每一个插件可以提供最适合一个特定文件或者特定环境的不同特征。例如不同的界面。插件是通过/scr/misc/modules.c 和/include/modules*.h 中定义的函数动态地装载或卸载。

下图给出了各模块的结构图：


主控制程序
界面模块
管理线程和用户
界面程序
管理和循环
Intf_msg
消息输出
Playlist
管理播放列表
视频输出
模块
播放图像
Video_output
图像显示处理
Vout plug-in
输出设备驱动
视频解码
Video_decoder
基本流解码
音频输出
模块
Audio_output
音频混合输出
Aout plug-in
输出设备驱动
音频解码
Audio_decoder
基本流解码
输入模块
Input
File/network socket
Input plug-in
初始，读取，分离
Programs
流管理
Input_ext-intf
流浏览
Mpeg_system
码流分离器
Clock
时间管理
涉及存储
管理
建立，初始化和清除
流入

4.6.2 线程与同步

在多线程的应用程序中，每个线程都要求具有自己的执行堆栈和程序计数器。多线程应用程序中的任何一个线程对于其它的线程来说都是独立执行的单元，它们都要具有自己的资源上下文。一个应用程序中至少应该具有一个线程，这个线程就是应用程序的主线程。我们可以通过线程的函数来开始和停止任何其它线程，但是该主线程只有到应用程序结束时才能停止运行。

我们可以创建一个用户定义的线程，然后在线程体中写入自己定义的函数或完成任务需要的代码段，当线程的函数执行后，该线程就保持执行状态。当线程函数体执行结束后，该线程就终止。

由于功能的需要，程序中使用了多线程。这样的话，解码器的占用和顺序安排就显得比较重要，共享内存的分配，线程间的通信成为难点。这就要通过线程管理来协调。一般来说，在应用程序中线程都需要通信，否则就没有必要来进行线程应用程序的设计了。应用程序的线程通信方法是比较多的。通常有以下几种方法：

• 利用全局变量来通信

当我们的应用程序中要求终止或者唤起线程的执行时，我们需要告诉线程何时终止、何时启动。一种方法就是定义一个全局变量，通过判断全局变量的值来控制线程的执行状态。

• 使用用户定义消息来通信。程序中利用了 C++语言中多线程管理的方法和机制，定义了一系列的函数和消息来进行线程中的通信和管理。

程序中的线程已经进行了模块化，使用了一个相对简单的包装。包括 vlc_thread_create, vlc_thread_exit, vlc_thread_join, vlc_mutex_init, vlc_mutex_lock, vlc_mutex_unlock, vlc_mutex_destroy, vlc_cond_init, vlc_cond_signal, vlc_cond_broadcast, vlc_cond_wait, vlc_cond_destroy, 和如下结构 vlc_thread_t, vlc_mutex_t, vlc_cond_t。这些类分别进行线程的建立、挂起、加入、解除，以及在对内存访问时的互斥初始化、锁定、解锁、消除等。

这种机制的应用较好的解决了线程管理的问题，但是有的时候还是会出现对内存的非法操作，这个问题有待解决。

在多线程的应用程序中，存在而且必须解决的问题是线程之间的同步问题。

比如在本程序中，播放模块的输入缓冲区同时也是解码模块的输出缓冲区，在这两个线程同时运行时，必然出现两个线程同时访问该缓冲区的情况，只是播放线程是读数据，而解码线程是写数据，这样，协调多个线程对缓冲区的访问就不可避免了。只有作好了线程的同步工作，才可以避免出现数据既读不出来也写不进去的情况，使应用程序变得更加安全可靠。

处理多线程访问数据的一种安全有效的方法是使多个线程在同一时刻只能有一个线程访问数据。线程通过 vlc_mutex 对象来访问数据，而该对象在某一时刻仅仅允许一个线程访问它自己，当这个线程返回后，下一个线程才可以访问该数据。这样就做到了在同一时刻多线程只能有一个线程访问数据。

我们的程序已把所有要保护的数据封装在一个类中，该类中同时封装了一个 vlc_mutex 对象，这样类中的数据仅仅可以被单个线程访问。线程通过 vlc_mutex 对象访问数据时，必须对该对象进行包含，可以使用该类中的 vlc_mutex_lock 和 vlc_mutex_unlock 封装。

当其它的线程没有使用自己要访问的 vlc_mutex 对象时，vlc_mutex_lock 便将该对象传递给调用自己的线程。而当使用完 vlc_mutex 对象后，可以使用 vlc_mutex_unlock 来释放该对象，以便其它线程可以访问对象中的数据。

另一个重要问题是解码和播放的同步。解码是由解码的线程完成，而播放则是由专门播放音频或视频的线程完成。为了让音视频在正确播放的同时不会阻塞任何的解码线程，同步问题必须得到解决。这使得在界面、输入、解码器和输出间的通信结构体更加复杂。在码流的系统层里的播放时间戳（PTS）被传递给解码器，然后所有的样值都被依此而标记。输出层就可以根据这个来准确的播放。时间标记被转化成微秒；一个绝对时间标记是从 1970 年 1 月 1 日开始计算的微秒数。而与之对应的 mtime_t 数据类型是一个有符号的 64 位整数类型。时间标记可由 mdate()函数得到。如果有必要，可以暂停线程直到 mwait（mtime_t 类型数据）得到一个特定的时间标记，也可以通过 msleep 来让线程挂起一段时间，当然也是用微秒计算。由此可以达到解码和播放的同步。

4.6.3 基本运行过程简述

当程序被启动，界面线程在/src/interface/main.c 中被启动而变成主线程，然后会经过以下几个步骤：

CPU 检测；消息接口初始化；命令行选项解析；创建播放列表；模块库初始化；启动界面；安装信号处理程序；产生音频输出线程；产生视频输出线程；主循环；事件管理。

4.6.4 消息接口

以下是可能用到的消息输出的函数：

intf_Msg(char * psz_format, ...)：在标准输出上打印一条消息。

intf_ErrMsg(char * psz_format, ...)：在标准错误输出上打印错误信息。

intf_WarnMsg(int i_level, char * psz_format,...)：如果警告级别够低，就把消息输出在标准错误输出上。

intf_DbgMsg(char * psz_format,...)：这个函数用来处理可选择的检测信息。在非跟踪模式下不会用到它。在调试模式下，消息会被写进文件 vlc-trace.log，或者输出在标准错误输出上。

intf_MsgImm，intf_ErrMsgImm，intf_WarnMsgImm， intf_DbgMsgImm：与上述功能一样，只是如果 INTF_MSG_QUEUE 被定义了，那在函数返回前，消息队列就会被清空。

intf_FlushMsg()：如果用得着，就清空消息队列。

4.6.5 管理播放列表

一个接口插件可以在播放列表中添加或删除文件。在 scr/interface/intf_playlist.c 中定义了所用到的函数。intf_PlaylistAdd 和 intf_PlaylistDelete 是最常用的两个典型函数。它们分别完成在播放列表中添加或删除文件的功能。而一旦插件调用了这两个函数，主界面中的管理循环函数 intf_Manage 负责在需要的时候启动或终止输入进程，从而达到在播放列表中控制要播放的文件的目的。

4.6.6 模块库

程序启动后，会给所有可用插件（.so 文件）和内嵌插件在.，./lib，/usr/local/lib/videolan/vlc（插件路径）中建立一个库，以便管理和调用。每个插件都根据功能而被选中。功能如下：

MODULE_CAPABILITY_INTF：界面插件；

MODULE_CAPABILITY_INPUT：输入插件，管理节目流或者 DVD；

MODULE_CAPABILITY_ADEC：音频解码器；

MODULE_CAPABILITY_VDEC：视频解码器；

MODULE_CAPABILITY_MOTION：视频解码器中的运动补偿模块；

MODULE_CAPABILITY_IDCT：视频解码器中的离散反余弦变换模块；

MODULE_CAPABILITY_AOUT：音频输出模块；

MODULE_CAPABILITY_VOUT：视频输出模块；

MODULE_CAPABILITY_YUV：视频输出中的 YUV 转换模块；

其他的线程可以通过 module_Need(module_bank_t * p_bank, int i_capabilities, void * p_data)来取得一个插件的描述符。p_data 是一个为函数 pf_probe()保留的可选参数。返回的 module_t 结构体包含指向插件中的函数的指针。在 include/modules.h 中有更详细的描述和定义。

4.6.7 主循环

界面接口线程会先找到一个合适的界面插件。然后通过插件中的 pf_run()函数进入界面主循环。它会先做一些先期的初始化工作，然后每 100ms 调用一下 intf_Manage。

intf_Manage 将通过卸载不必要的模块来重写模块库，管理播放列表。当要用到消息队列的时候，它还会把等待的消息清空，当模块库和播放列表发生变化时，intf_Manage 会重复上面的工作。

4.6.8 函数及变量命名约定简介

为了便于下一届的同学进行进一步开发，并加强程序的可读性，我觉得有必要把命名规则的约定作一下简介。

1．函数命名根据：模块名＋__＋函数名。例如： intf_FooFunction。静态函数不需要使用模块名 。

2．变量命名根据：

i_ 命名整数型 (l_ 命名长整型)；

b_ 命名布尔型；

d_ 命名双精度实数型 (f_ 命名浮点实数型)；

pf_ 命名函数指针；

psz_ 命名字符串指针；

此外在命名指向某种类型的指针时，会在变量名前加上 p_；如果一个变量不属于基本类（如：复合结构），就不加任何前缀（除非它是一个指针 p_*）。但是以防万一，在前缀后我们会加上变量名。必要时，把它们和下划线一起加入。例如：

```
data_packet_t * p_buffer;
char psz_msg_date[42];
int pi_es_refcount[MAX_ES];
void (* pf_next_data_packet)( int * );
```

4.6.9 界面接口插件的编写

1．界面接口插件需要完成五个功能:

intf_Probe (probedata_t * p_data)：这个函数主要是探测插件能否在现有环境下工作。如果能，函数返回 1－999 之间的一个数，否则返回 0。p_data 在这时是不需要使用的。

intf_Open (intf_thread_t * p_intf)：初始化界面 (例如：打开一个新的窗口). p_intf->p_sys 可以用来存储自己需要的信息。

intf_Close (intf_thread_t * p_intf)：关闭窗口，释放所有占用的资源 (包括 p_intf->p_sys)。

intf_Run (intf_thread_t * p_intf)：进入主循环，直到 p_intf->b_die 被设为 1 时结束。

文件的读取：

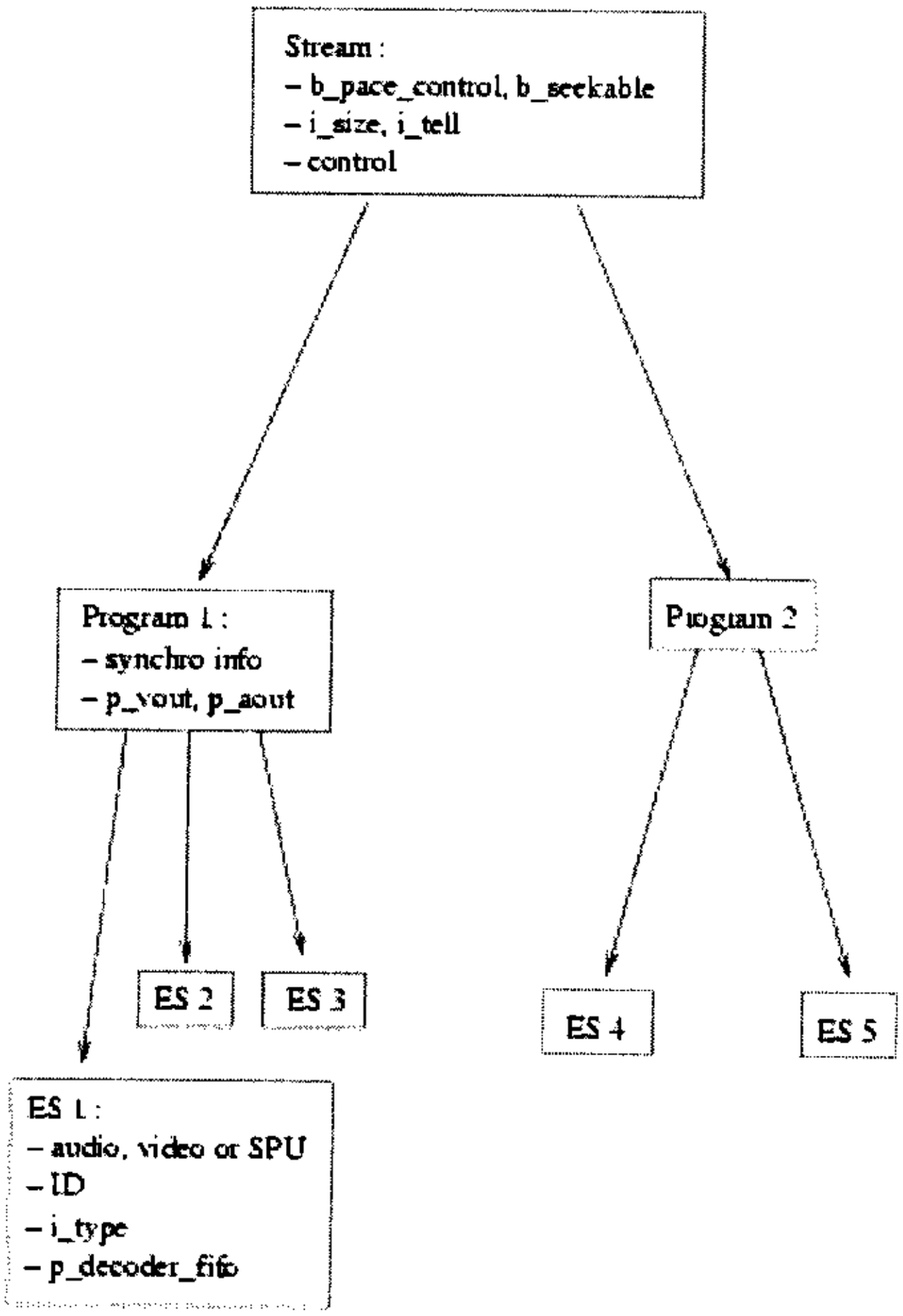
因为媒体流的定义不同，所以播放不同的流时，编码器与输入线程需要重新初始化。界面线程会调用 input_CreateThread 来初始化要用到的线程。

首先找到能读插件的插件程序，如 input_FileOpen，input_NetworkOpen, 或者 input_DvdOpen，来打开套接字。函数会设置两个非常重要的参数：b_pace_control 和 b_seekable 来描述控制分量。接着程序流成回进入输入插件中的 pf_init 函数，循环执行 pf_read 和 f_demux 来读取媒体流。插件负责初始化流结构，管理包缓冲器，读包，解复用输出包。

2．输出到界面接口的结构

请注意 include/input_ext-intf.h，此文件定义了 input_thread_t 结构，stream_descriptor_t 结构和关于 ES 的描述。

首先，input_thread_t 结构定义了两个空类型指针：p_method_data 和 p_plugin_data，用来指向缓冲器管理数据和插件数据。接着，流描述被存储到一个专门用来存储流描述的树型程序描述器中。这个树型结构，我们用下图来描述：



ES（elementary streams）由 ID（a stream_id 真正的 MPEG 流 id），类型（由 ISO/IEC 13818-1 中表 2-29 说明）和文字上的说明共同描述. 还可以携带解复用的相关信息和编码器信息 p_decoder_fifo。如果流不是 MPEG 系统层的，还需要特定的解复用信息。如果需要携带额外信息，还需要使用空类型指针 void * p_demux_data。

输入插件负责启动必要的解码线程，如果它想选择 ES（elementary streams），必须调用 input_SelectES (input_thread_t * p_input, es_descriptor_t * p_es)

3．界面接口使用的方法：

input_ext-intf.c 提供了控制读取媒体流的一些方法：

input_SetStatus(input_thread_t * p_input, int i_mode) ：控制读取速度 i_mode 是 INPUT_STATUS_END, INPUT_STATUS_PLAY, INPUT_STATUS_PAUSE, INPUT_STATUS_FASTER, INPUT_STATUS_SLOWER 中的一个。事实上流的读取速度取决于变量 p_input->stream.control.i_rate 。变量缺省值为 DEFAULT_RATE。

input_ClockManageRef(input_thread_t * p_input, int i_mode)：负责改变速率。由停止输入流实现播放流的暂停。

input_Seek (input_thread_t * p_input, off_t i_position) ：节目播放过程中的随意拖动。位置通常是在 p_input->p_selected_area->i_start 和 p_input->p_selected_area->i_size 之间的一个数值。

input_OffsetToTime (input_thread_t * p_input, char * psz_buffer, off_t i_offset)：相应调整时间的偏移量。

input_ChangeES (input_thread_t * p_input, es_descriptor_t * p_es, u8 i_cat)和 input_ToggleES (input_thread_t * p_input, es_descriptor_t * p_es, boolean_t b_select)：提供函数管理 ES 流。

输入模块从 src/input/input_dec.c 中激活相应的解码器模块。Dec_CreateThread 函数会选择更精确的解码器模块。每一个解码器模块会去找解码器配置类型然后返回一个值。解码器配置类型在 include/input_ext-dec.h 中描述。

4.6.10 数据包结构

数据包结构在 include/input_ext-dec.h 中定义。其中 data_packet_t 是一个指向数据物理位置的指针。解码器读数据从 payload_start 开始直到 p_payload_end 结束。当 p_next 不是空时，将指向下一个包。如果 b_discard_payload 有值，意味着需要丢弃包。

输入模块和解码器模块都需要用到结构 decoder_fifo_t。它规定 PES 包以队列形式编码，控制这个过程的宏包括：

DECODER_FIFO_ISEMPTY,DECODER_FIFO_ISFULL,

DECODER_FIFO_START,DECODER_FIFO_INCSTART,DECODER_FIFO_END,

DECODER_FIFO_INCEND。而下一个包进入解码器需要调用

CODER_FIFO_START(*p_decoder_fifo)。解码结束后需要调用 p_decoder_fifo->pf_delete_pes(p_decoder_fifo->p_packets_mgt, DECODER_FIFO_START(*p_decoder_fifo)) 其中 DECODER_FIFO_INCSTART(*p_decoder_fifo)把 PES 返回用于缓冲区管理。如果队列为空可以再填充，否则不行，一面出现上溢。如果读到文件末尾，p_fifo->b_die 置 1，然后调用 vlc_thread_exit()结束整个过程。

4.6.11 视频解码器使用的方法

视频输出提供一系列方法完成解码器送出解码数据：

picture_t * vout_CreatePicture (vout_thread_t *p_vout, int i_type, int i_width, int i_height)：返回指定帧缓冲器。i_type 可以是 例如 YUV_420_PICTURE。

vout_LinkPicture (vout_thread_t *p_vout, picture_t *p_pic)：增加帧的可关联性，使得这个帧不会在解码器仍然需要时已经不存在了。例如，在播放完 B 帧后，I,P 帧还是有可能需要的。

vout_UnlinkPicture (vout_thread_t *p_vout, picture_t *p_pic)：降低帧的可关联性，在一个现有关联存在前，要断开其他关联。

vout_DatePicture (vout_thread_t *p_vout, picture_t *p_pic)：给每一帧提供说明数据。

vout_DisplayPicture (vout_thread_t *p_vout, picture_t *p_pic)：通知视频输出模块：帧已经被完全解码并准备发送给视频输出模块。

vout_DestroyPicture (vout_thread_t *p_vout, picture_t *p_pic)：对无效帧做标记。

subpicture_t * vout_CreateSubPicture (vout_thread_t *p_vout, int i_type, int i_size)：返回指定子帧图象缓冲器。i_type 可以是 DVD_SUBPICTURE 或者 TEXT_SUBPICTURE。

vout_DisplaySubPicture (vout_thread_t *p_vout, subpicture_t *p_subpic)：通知解码器子帧解码结束可以发送给视频输出模块。

vout_DestroySubPicture (vout_thread_t *p_vout, subpicture_t *p_subpic):标记无效子帧。

4.6.12 视频输出插件

视频输出插件通过调用系统命令实现当前帧并管理输出窗口，需要用到以下函数（在 plugins/x11/vout_x11.c 内）：

int vout_Probe (probedata_t *p_data)：返回 0－999 之间的数值来说明能处理的结构。999 最佳。

int vout_Create (vout_thread_t *p_vout)：初始化打开新窗口。

int vout_Init (vout_thread_t *p_vout): 打开可选帧缓冲器。

vout_End (vout_thread_t *p_vout): 释放可选帧缓冲器。

vout_Destroy (vout_thread_t *p_vout): 关闭窗口释放所有资源。

int vout_Manage (vout_thread_t *p_vout): 管理各种事件，如窗口大小、移动等。

vout_SetPalette (vout_thread_t *p_vout, u16 *red, u16 *green, u16 *blue, u16 *transp): 设置颜色变量。

第五章 总结与建议

在上面两章中，分别介绍了我在研究生期间所做的研究和开发工作。第三章中主要是对 MPEG 编码算法进行研究。第三章中介绍了 MPEG 码流在 IP 网上传输的主要应用——局域网内的视频点播系统。考虑到开发成本、客户需求、版权以及可重用性等因素，我们摒弃了 Windows 平台上微软提供的媒体服务而是采用在 LINUX 平台用 glibc 自主开发整个系统，客户端播放界面采用 Qt 实现。我在整个系统中所做的主要工作是客户端的具体设计以及相应实现流程。

我们的视频点播系统目前还在开发阶段，因此还有许多尚未实现的功能。

首先是对节目格式的支持。目前我们只做了对 MPEG-1 和 MPEG-2 码流的传输与播放，将来还应该扩充到 MPEG-4。我在第三章中所探讨的 MPEG-2 数据分割分层编解码，将来也可以应用到我们的系统中。用户可以选择只点播基本质量的节目，也可以选择点播高质量的节目，这就需要将基本层和增强层的数据同时传送给该用户。另一方面，在用户点播的过程中，如果网络条件变坏，即使增强层的分组包在传送过程中丢失了，用户也可以根据收到的基本层的比特流继续观赏，只是这时质量有所下降而已，而不至于造成节目的不连续性。

另外，我们的视频点播系统目前只支持点播节目库中存储的预先压缩好的节目，将来还应该实现编码器功能，使得用户能够点播实时采集的视音频节目。但是对于实时的压缩编码，靠纯软件的编码器就不行了，需要由硬件来实现。

对于媒体服务器与客户端交互的部分，目前只实现了客户端本地节目的基本功能，包括选择、播放、暂停、快进、倒带、停止等，但是还没有用 RTCP 协议实现实时控制，而像节目分段、插播广告、用户调查、时间统计等服务器的扩充功能，仅仅是在媒体服务器的设计时给出了相关的消息定义，尚没有具体实现，也有待接续者继续完成。

在编写程序的过程中，也遇到一些技术上的问题，但是由于比较琐碎，可讨论性不强，所以就不再论文中一一列出。但是在以后的工作中一定要注意。

参 考 文 献

[1] ISO/IEC 11172，具有 1.5Mbit/s 数据传输率的数字存储媒体运动图像及其伴音的编码——MPEG

[2] ISO/IEC 13813，运动图像及其伴音通用编码国际标准——MPEG-2

[3] DAVIC 1.3 Specification Part 2，System Reference Models and Scenarios

[4] DAVIC 1.3 Specification Part 7，High and Mid Layer Protocols

[5] D. Hoffman, G. Fernando and V. Goyal, "RTP Payload Format for MPEG1/MPEG2 Video", RFC 2250, January 1998

[6] M. Civanlar, G. Cash and B. Haskell, "RTP Payload Format for Bundled MPEG", RFC 2343, May 1998

[7] 中华人民共和国通信行业标准 YD/T1130-2001，基于 IP 网的信息点播业务技术要求

[8] 中华人民共和国通信行业标准 YD/T1131-2001，基于包的多媒体通信系统上的呼叫信令协议及媒体流打包技术

[9] 蔡安妮，孙景鳌，《多媒体通信技术基础》，电子工业出版社，2000

[10] 黎洪松，《数字视频技术及其应用》，清华大学出版社，1997

[11] 余松煜，张文军，孙军，《现代图像信息压缩技术》，科学出版社，1998

[12] [美] A.比特勒等，《客户/服务器环境中的事务处理》，科学出版社，1998

[13] [美] Arthur Griffith，《KDE 2/Qt 编程宝典》，电子工业出版社，2002

[14] 俞文健，刘毅，金超，《Red Hat Linux 6.x 实用大全》，人民邮电出版社，2000

[15] John Goerzen，《Linux 编程宝典》，电子工业出版社，2000

致 谢

在本文完成之际，我要对我的导师黄孝建教授致以最真挚的感谢。在攻读硕士研究生这三年的学习和生活中，在黄教授的谆谆教诲下，我取得了长足的进步。可以说我的每一点进步，每一份成绩和收获，都凝聚着黄教授的心血。在此我要对黄教授说一声：谢谢老师！

感谢实验室各位同学在学习、工作和生活上给我的关心和帮助，尤其感谢周惠文师兄、何海波和吴君同学，张文强师弟。它们在编写论文的整个过程中给我以精神上、生活上的支持和工作上的帮助。

感谢所有关心、帮助我的老师、同学和亲人们！