
电 子 科 技 大 学
UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

硕士学位论文

MASTER THESIS



论文题目 铁路视频监控系统的研究与实现

学 科 专 业 计算机系统结构

学 号 201021060124

作 者 姓 名 李海涛

指 导 教 师 雷 航 教 授

分类号 _____ 密级 _____

UDC ^{注 1} _____

学 位 论 文

铁路视频监控系统的研究与实现

(题名和副题名)

李海涛

(作者姓名)

指导教师

雷 航 教 授

电子科技大学 成 都

(姓名、职称、单位名称)

申请学位级别 硕士 学科专业 计算机系统结构

提交论文日期 2013. 03 论文答辩日期 2013. 4. 28

学位授予单位和日期 电子科技大学 2013 年 6 月 29 日

答辩委员会主席 _____

评阅人 _____

注 1: 注明《国际十进分类法 UDC》的类号。

RESEARCH AND IMPLEMENTION OF VIDEO SURVEILLANCE SYSTEM ON RAILWAY

A Master Thesis Submitted to

University of Electronic Science and Technology of China

Major: Computer Architecture

Author: Li Haitao

Advisor: Lei Hang

School : School of Computer Science and Engineering

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：_____ 日期：_____ 年 _____ 月 _____ 日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将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、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名：_____ 导师签名：_____

日期：_____ 年 _____ 月 _____ 日

摘 要

随着人们对人身安全的关注和智能化办公需求的增加，视频监控技术，特别是视频监控系统的应用越来越多，所发挥的作用也越来越重要。随着近些年来铁路系统的发展，特别是在万里的铁路线上和密集的大客站上，对于视频监控系统的需求越来越高，因此，在铁路系统中实现视频监控就成了必然的趋势。

本文正是在这样趋势的促使下，结合新技术新思想，在对当前视频监控系统的理论和实现技术分析和研究的基础上，对成熟的视频监控系统设计和实现案例进行了总结，对铁路视频监控系统项目进行了需求分析，设计并实现了视频监控系統平台。完成了如下的主要工作：

1. 对视频监控系统结构、功能和性能完成详细的需求分析工作。特别是针对所要实现的铁路环境和物理网络条件的分析，对实现的多种功能，如实时播放、视频回放、权限控制和带宽管理等的分析以及对系统的性能包括实时性和流畅性，可靠性和响应效率等方面的分析工作。这些工作提供了大量的数据和信息，对系统的设计和实现具有巨大的帮助作用。

2. 结合需求分析的要求，对系统进行了设计与实现。主要分为三大部分内容，第一，对系统整体结构的设计和分析，要在铁路系统中建设视频监控系统就需要设计满足铁路环境要求和业务逻辑结构的具体实现。第二，对系统工作流程和实现功能的设计，其中对于实时播放、视频回放、带宽管理和权限控制的设计是系统设计和实现的重心。第三是对针对流程和功能的设计，对系统具体模块的设计和实现工作。结合性能需求，还需要对系统运行的可靠性和视频播放的流畅性与实时性等的实现工作进行完善和改进。这部分工作是系统完成的实际操作，因此决定了系统运行的好坏。

3. 设计并实现的系统还要经过测试的验证，所以需要对系统功能和性能进行有效的测试和分析工作，分析测试数据，对系统设计和实现的效果给予反馈。经过测试，系统的设计和实现符合需求分析提出的要点，达到了可靠性的要求，在实时性和流畅性方面也具有不错的表现效果。

关键词：视频监控，实时播放，视频回放，权限控制

ABSTRACT

With the attention of personal security and the need of intelligent office, Video surveillance technology, especially the system, has more and more use and influence. In recent years when the railway has more development, the need for video surveillance in railway and stations is more and more. So its achievement must be carried on.

Under this trend, this dissertation summarizes and analyzes the new ideas and the perfect system examples to design and implement the system of video surveillance by using its requirement analysis based on the research and analysis of the technology and principle on video surveillance .The works has been done as following:

1. By means of the requirement analysis of the structure, function and performance of the system, especially the environment and conditions of railway and network, the many functions including the real-time, playback, level control and bandwidth management and the performance including the real-time, fluency, reliability and timeliness should be analyzed. A lot of data and information from these works give much help for the designing and implementing of the system.

2. The design and implement have three parts. Firstly, the total structure of the system needs to satisfy the condition and the logic level of the railway station. Secondly, the flow of work and the implement of function mentioned on the four parts should be finished conscientiously. Last based on the second part, the special module should be designed and implemented logically. The work for improving the performance of the system including the reliability, real-time and fluency should be treaded carefully for the reason that these works are important for the finality of the system.

3. Testing is important for checking out the function and the performance of the system .The data and the result of the tests must be analyzed to describe the conclusion of the system .Through by the test, the design and realization of the system meets the requirements analysis, achieving the requirement of reliability, and also has good performance on real-time and fluency.

Keywords: video surveillance, real-time, playback, level control

目 录

第一章 绪 论	1
1.1 研究的背景和意义	1
1.2 国内外研究现状	2
1.3 本论文的研究内容	4
1.4 本文层次结构安排	5
第二章 相关技术和工具	6
2.1 相关技术	6
2.2 相关工具	8
2.3 本章小结	10
第三章 视频监控系统的的需求分析	11
3.1 铁路系统结构的分析	11
3.1.1 物理网络结构	11
3.1.2 逻辑结构	14
3.1.3 业务结构	15
3.2 功能需求分析	15
3.2.1 实时播放	17
3.2.2 视频回放	18
3.2.3 权限控制	18
3.2.4 带宽管理	19
3.3 性能需求分析	20
3.3.1 视频播放的流畅性和实时性	20
3.3.2 服务器运行的可靠性	21
3.3.3 系统服务器的响应效率	21
3.4 本章小结	22
第四章 视频监控系统的设计与实现	23
4.1 系统整体设计与实现	23

4.1.1 系统物理结构的设计与实现.....	24
4.1.2 系统业务逻辑结构的设计与实现.....	25
4.2 视频监控主程序的设计与实现.....	27
4.3 实时播放的设计与实现.....	29
4.3.1 实时播放的流程设计与实现.....	29
4.3.2 主连接的建立和处理.....	30
4.3.3 视频连接的建立和处理.....	32
4.4 视频回放的设计与实现.....	37
4.4.1 视频回放的流程设计与实现.....	37
4.5 客户交互模块的实现.....	38
4.5.1 连接的初始化与处理.....	39
4.5.2 客户连接交互模块.....	40
4.5.3 信息查询和视频连接交互模块.....	41
4.6 数据处理模块的实现.....	42
4.6.1 设备与通道的连接和处理.....	42
4.6.2 设备连接和处理的设计与实现.....	43
4.6.3 通道连接和处理的设计与实现.....	47
4.6.4 权限控制和设备维护.....	50
4.6.5 视频回放数据处理的设计与实现.....	53
4.6.6 有关视频播放时的连贯性和实时性优化的设计与实现.....	56
4.7 本章小结.....	58
第五章 系统的测试与运行.....	59
5.1 系统测试和运行环境.....	59
5.1.1 硬件环境.....	59
5.1.2 软件环境.....	59
5.2 系统部署和运行测试.....	59
5.2.1 系统的部署.....	59
5.2.2 系统的运行.....	64
5.3 系统功能测试.....	66
5.3.1 实时播放.....	66
5.3.2 视频回放.....	69

5.3.3 云台控制和权限管理.....	70
5.4 性能测试.....	72
5.5 本章小结.....	73
第六章 总结与展望.....	74
致 谢.....	76
参考文献.....	77
攻硕期间取得的研究成果.....	79

第一章 绪 论

铁路在我国的经济发展和现代化建设中发挥着重要作用，国民经济的发展需要铁路来贯穿南北西东，国家建设需要铁路来连通省市，交通出行需要铁路来联系你我他之间的距离^[1]。随着社会的发展、经济的繁荣和现代化水平的提高，在现阶段我国基本国情的条件下，铁路所能发挥的作用尤为突出。在四川地震，南方暴雪这样恶劣的自然灾害面前，几乎所有的运输方式都已经瘫痪的情况下，唯有铁路撑起了挽救生命、保障货资的希望，在中国千万里的铁路线上，上演了令世界为之赞叹的运输能力，一次又一次的展现了中华儿女那不屈不挠的精神。而为了让铁路能够在这新时代具有中国特色的社会主义现代化建设中发挥更大的作用，现代化、网络化的视频监控系统的建设势在必行。

1.1 研究的背景和意义

科学技术的进步和管理方式的转变给新时期的铁路系统注入了高效、安全、稳定的新鲜血液。随着近几年国民生产力水平的提高，科技水平的腾飞，综合国力的日益攀升，铁路系统也紧跟时代进步的步伐，从过去分散的、效率低下的管理模式转向综合调度指挥、综合维修、集中养护等高度集中高效的作业管理体制。随着网络技术的发展，无线电、图形图像等高科技技术的普遍应用，铁路这个钢铁巨人也越来越多的吸收了新的科技元素。而这些的加入，为铁路安全高效的运行提供了更加有效的保障。随着电子化信息化所带来越来越多的好处，对于可视化的管理运营模式的需求在不断攀升。视频监控系统作为构建智能化、科技化、系统化铁路系统不可或缺的一个环节，可以为车辆的统筹安排、抢险保障的统一调度、灾害的防护和安全保障等提供直观先进有效的决策辅助，给保障铁路的安全运营，提升服务质量提供了有力的手段^[2,3]。

视频监控系统一般由前端视频设备，传输线路，后台服务控制系统和客户端组成^[4]。随着网络技术的发展，传统的 TCP^[5,6]，UTP^[5,7]协议进行网络数据传输已经不能满足流媒体数据越来越高的传输要求，取而代之的是对流媒体更有针对性的 RTP^[8,9]，RTSP^[8,10]，RTCP^[8]等协议，提供了更好的保障和更优异的表现。由于视频图像往往数据量很大，在网络上传输，必然占用大量带宽和资源，随着视频编解

码技术的成熟，特别是 H. 264/AVC 标准^[11,12]的使用，使得同等码流带宽的情况下能够传输更多的视频画面，这大大提高了视频监控的作用。鉴于铁路系统自身地域广、环境复杂、条件有限等诸多不利因素的特点，使得通常的视频监控系统难以发挥其作用，因而，对于日益重要的综合化、智能化、科技化的铁路系统建设而言，实现满足铁路需求的完善的视频监控保障势在必行。

1.2 国内外研究现状

视频监控系统的发展经历了三代，从早期的模拟闭路电视监控系统向数字视频监控系统演变，而随着网络时代的来临，IP 技术逐步统一全球的今天，满足网络需求的具有网络化、智能化特点的视频监控系统将被越来越多的应用到各个领域之中^[13]。

视频监控是由摄像、传输、控制、显示和存储等五大部分组成，而其中摄像、传输和显示是必不可少的部分。

摄像，顾名思义，通过摄像头、摄像机等专业的设备捕获并收集图像视频信息，而获得的视频图像信息经过数字化处理后就是用来传输的原始视频数据信息，这些数据的质量，特别是清晰程度从源头决定了视频监控系统视频播放画面的清晰度。而随着技术的发展，特别是近几年成像设备和光学材料的进步给予摄像设备的发展带来了新的春天，由原始的模拟信号摄像头，到数字信号摄像头；由原始单一的摄像头，到现在应用于不同领域的专业摄像头。例如，高清摄像头，带三百六十度旋转的球机摄像头，可变焦摄像枪，红外线摄像头等。随着需求的拓展和技术水平的提高特别是摄像成像技术的发展，视频监控系统的应用领域越来越广。

传输，这是视频监控系统得以广泛使用的基本保障。由于视频图像需要显示的地方往往与摄像设备所在位置相隔很远，将视频数据保质保量的传输成为了视频监控实现的关键。传统的模拟信号由于其抗干扰性差，传输距离短，设备消耗大成为视频监控系统广泛使用的瓶颈。但随着对视频监控系统应用的越来越多，其使用的范围也不再仅仅局限于几十、几百米，更多的是对远距离数据的访问，而这是模拟信号传输所无法实现的。随着数字信号的使用，大大提高了视频数据信息的传输距离和传输效率，为视频监控系统的发展奠定了基础。而网络时代的来临，更使得视频监控系统这种可视化的安防手段越来越多的被重视。网络化和智能化给视频监控系统带来了新的生命力。如今，随着光纤技术的发展，庞大视

频数据的传输所引起的设备消耗和信号干扰的影响也在逐渐变小。

控制、显示和存储为视频监控系统提供了更加灵活和有效的手段。对于视频监控系统而言，显示是根本目的，只用通过这种可视化的监控方式，视频监控系统的优点和特性才能更好的展现出来。从传统的显示器到视频矩阵，甚至如今的平板电脑和移动终端都能成为视频监控系统的载体，而这极大的增加了可视化终端的灵活性和方便性。随着设备和技术的发展，控制云台能够为传统固定的摄像设备提供实时按需的灵活监控，为突发情况的捕捉和实时定位跟踪都提供了基础保证。存储矩阵的加入使得视频监控系统具有了“生命期”，传统的视频监控随着实时的观看就失去了其视频图像，而存储数据的实现使得重要数据的保存和回放成为了可能，极大提高了视频监控系统的灵活程度和可靠性。

硬件平台设备的更新发展仅仅是为视频监控系统在各个领域内发挥作用奠定了基础，众所周知，真正的把这些设备统一在一起协调工作的还是软件系统，随着图形图像技术和流媒体协议的发展，在网络化信息化的浪潮中，视频监控系统一次次的完善更新着自己。

随着内容的丰富，需求的增加，技术的进步和设备的更迭，视频监控系统慢慢的由本身单一独立的系统软件逐步的转变为由具有多样的应用需求和灵活扩展性的视频监控平台^[14,15]。在平台上，视频监控和观看仅仅成为了基础，在之上所加入的各种新技术和新思想不断的丰富和完善着系统本身，也让系统具有了更多个性化的实现。

智能视频分析技术是在视频监控平台上应用最多也是最重要的实现，它使得传统意义上简单的单纯的视频监控系统平台变的更具智能，更加灵活，更符合人性化的要求^[16]。通过对视频图像特征的分析，不但可以快速的找出所感兴趣的视频画面，还可以对人员进行全方位的动态追踪而无需人工的参与。对于智能视频分析技术的研究和应用是最近几年发展的方向，国内外有很多的学校和企业都在致力于这方面在实际中的应用和实现工作。例如，由美国国防高级研究计划局（DARPA）资助，卡内基梅隆大学等几家著名研究机构合作开发的，用于未来城市和战场的识别和监控的一种自动视频理解技术；英国的雷丁大学开展的对车辆和行人的跟踪及其交互作用识别的相关研究等。此外，视频手势等智能视频技术的发展，特别是在手持终端上的良好表现力，给智能视频分析技术在视频监控系统平台的实现提供了可能。如今，国内的很多知名大学和企业也同样在致力于智能视频分析技术的研究，特别是在行为模式识别和运动目标检测等方面都取得了一定的成果。

单机设备的加入，使得视频监控不再是固定的监测，而提供了一种实时直播的手段，特别是对于突发情况的区域，没有视频设备存在，那么就需要先进的单机设备及时的捕捉视频画面，通过视频监控系统实现数据的传输。

另外，随着视频搜索技术的发展，对于海量的视频数据信息的查找效率和精确度将大大提高，这将会对视频的分析工作带来前所未有的推动作用，那么对于视频监控平台的巨大效益将是不可估量的。

微电子技术和计算技术的进步，特别是针对不同领域的应用需求，视频监控平台可以和越来越多的其他领域相结合产生更加高效的产品。例如同webGIS相结合的视频监控系统平台能够更加有效和直观的在地图上标明视频监控的区域和位置；利用微电子技术和视频监控系统平台结合，实现更精确的定位和联动，在突发情况发生时能够及时的响应并捕获场景信息等。

可见，硬件设备的更新，技术手段的加强和功能业务需求的丰富对视频监控的发展是一针有力的强心剂，更给予了它不断前进和发展的保障，为视频监控系统在铁路系统中实现，完成铁路的现代化建设夯实了基础。

1.3 本论文的研究内容

本文课题源自于“成都铁路局视频监控系统”项目。该项目是成都铁路局的重点项目，旨在实现成都铁路局辖区内的全方位全站点的远程视频监控。针对成都铁路局系统的特点，特别是在物理设备受限，网络资源有限的条件下，结合项目需求，设计了较为合理的系统结构，完善了相关设计和算法，实现视频数据的本地存储和远程分发。此外提出了有效的权限控制判断，为网络带宽的合理利用和系统的稳定运行提供了保证。

本文课题主要针对成都铁路局辖区的站点环境，因而满足成都铁路局的现有设备和条件是必须具备的前提，由于网络线路的改造依然在进行中，所以导致在现有的网络线路的条件下只能提供 2M 的视频传输带宽，因而如何确保权限和带宽保证视频的稳定播放，这是系统设计的首要考虑的问题，一方面通过权限控制，保证视频带宽的合理化使用，另一方面，通过服务器分发，避免视频带宽的重复利用造成资源的浪费，更高效的合理利用有限的带宽资源。铁路系统中，站点和线路存在着变化性，同时业务逻辑关系也同实际的物理链路结构存在着很大的冲突，通过将物理结构同业务逻辑结构的耦合度降低，设计满足铁路变化需求的视频监控系统结构。另外由于设计的要求，需要对两路码流和多种设备提供支持，

使得信息查询和数据获取的方式都有新的变化，也需要在尽可能短的时间内完成系统的设计与实现。

1.4 本文层次结构安排

本文的层次结构安排按照如下的方式进行：

第一章是绪论，主要是对本文所介绍的视频监控系统的项目来源和实现意义做了主要的介绍和阐述，同时对相关背景和环境做了分析，目的是对本文有一个总体直观的了解。

第二章是对本文所涉及内容的相关技术做简单的分析和了解工作，特别的由于项目的实现需要开发工具的帮助，所以还对系统实现所用到的工具做了简要的分析。

第三章是系统的重点，是对视频监控系统的需求分析工作，通过铁路系统结构的分析，功能需求分析和性能需求分析三部分对系统所要实现环境和需要完成的工作有个比较具体的了解。分别从物理网络结构和业务逻辑结构入手对铁路系统结构的进行了分析，分四个方面对系统功能需求进行了介绍，同时还对系统性能所要实现的效果做了阐述，为系统的设计与实现打下了基础。

第四章是整篇文章的核心，更是系统的核心，此部分通过对系统整体的设计和实现，对数据处理模块的设计和实现以及对客户交互模块的设计和实现，从三方面由浅入深，由整体到局部，结合需求分析的要求对整个系统特别是服务器程序的设计和实现做了详细的阐述。此外还对提高系统性能和友好性的几个方面也做了完善与优化工作。

第五章是系统的测试，主要是对系统功能和性能的测试，根据测试的结果查看系统是否能够按照设计的需求正常工作，是对系统的检验也是对系统能否稳定运行的保证。

第六章是对工作的总结，在下一步工作中需要改进不足，优化系统的性能。

第二章 相关技术和工具

2.1 相关技术

视频监控系统能够实现远距离传输的一个重要方面就是数字视频信号数据的高效压缩。视频压缩技术就是在数字视频数据的基础上，基于视频图像在空间和时间上的冗余，利用一定的算法和原理在保证图像质量的前提下将空间和时间中冗余的信息去除，只保留非相关信息的技术^[17]。通过这种方式传输视频数据，可以在有限的数据流条件下传送更多的视频画面。从 20 世纪 80 年代以来，国际标准化组织 ISO/IEC 和国际电联 ITU-T 都在不断追求在尽可能低的码率下获得尽可能好的图像质量。

从 90 年代初具有高效的编码特性、高质量的图像表现性、高可靠的处理性以及较低的实现延迟特点的 H.261^[18]编码标准运动图像编码算法出现在世人面前以来，它已在实时会议，运动图像处理的编解码实现过程中一次次的得到证明。随着 H.263^[19]等编码标准的过度，如今由 ITU-T 的 VCEG（视频编码专家组）和 ISO/IEC 的 MPEG（活动图像编码专家组）的联合视频组（JVT）开发的一个新的数字视频编码标准 H.264 已经是多媒体实时领域的主流，越来越多的视频数据采用这种方式压缩编码，特别是它良好的网络适应性在网络流媒体传输方面具有显著的优势。H.264 编码方式是它的前辈们优点的继承和发扬，在这基础上其适应性和兼容性大大增强，这在实现的功能和领域的范围将更加广泛，同时编码之后的数据格式能够很好的在网络数据传输的过程中保证信息的安全和稳定，这就使得 H.264 编码格式与生俱来的具有同网络更好的亲和力和表现性能^[20]。H.264 编码方式应用目标范围更广，满足了不同速率、不同解析度以及不同传输场合的需求，使得其自身在通过网络传输的过程中具有更强的错误分析能力和自我修复能力，因此在网络状态不稳定的环境里 H.264 编码方式将有更突出的优点，这些是以往的编码方式所无法比拟的^[20-22]。

H.264 编码格式的视频数据，一般的会包含三种帧数据格式。I 帧，称之为关键帧，之所以这样命名，原因在于它是所有帧数据的基础，由于 H.264 是一种有损的编码方式，其高效的视频压缩技术必然的要求在数据处理上有所侧重，那么在实现的时候就无法保证所有的视频帧都是完整的信息。I 帧是个例外，它包含了

一帧图像画面的全部信息，之所以这样做，原因在于它是其他所有视频帧的参考，如果参考都不完整，所实现的视频画面将无法获得很好的效果。

B 帧和 P 帧是用于补偿和预测的数据帧，简单的来讲这些帧数据发挥的作用就是在 I 帧的基础上或者是当前播放画面的基础上对视频图像进行微调，而随着这种微调的累积所带来的效果就是视频的动态变化，也就实现了视频数据的压缩和编码。因此每个 B 帧和 P 帧都有它们对应的 I 帧数据，而这就是它们用于微调的基础。所以在视频编码的过程中从理论上讲，只需要一帧的 I 帧数据，后边全是其他帧数据就可以实现视频画面的播放，但在实际的应用中，一般的对于摄像设备编码的过程中，为了更好的容错和更新都会采用一段间隔时间一个 I 帧的方式来实现。比较多的采用 2 秒的时间间隔。

在使用 H.264 进行解码的过程主要的步骤有：经过熵编码解码后的数据要进行重新的排序，得到的宏块在进行反量化和逆变化之后，再根据参考帧的数据重新构建视频帧数据，重新构建的视频帧数据再经过滤波处理将会得到可以用于播放的良好的视频画面。由于在 H.264 的标准中所定义的更多是方法和工具，而在真正的具体实现中会根据具体的实现要求有选择性的使用这些工具和方法。为了统一，H.264 定义了三种较为常用和合理的模式，称之为 profile，解码时只需要根据具体的 profile 就可以确定解码的过程了。例如，BaseLine profile 为低延迟的应用设计，它应用于低处理能力和高丢包率的环境中，所以它的编码效率是最低的；Extended profile 是 BaseLine 的超集，增加了更多的工具，有了更好的效率，它的目的在于流媒体视频的应用；Main profile 提供了最好的编码效率。在项目的实际环境中，根据具体的网络环境等条件，视频编解码所使用的 profile 等级为 Main。

在网络中实现数据的接收和发送以及控制离不开稳定高效的传输协议及标准的实现，在视频播放越来越成为主流的今天，传统的网络数据传输协议由于其共通性的特点对于视频数据的处理并没有优势，因此已经不能完全满足越来越庞大的视频数据的需求和灵活多样的实现^[23,24]。稳定可靠的 TCP 虽然能增加安全性和可靠性，但对于网络资源的开销太过巨大，其较低的效率也为视频监控的实现提出了不小的挑战；高效轻便的 UDP 虽然去除了“三次握手”等耗费资源的开销，但没有了可靠性和安全性的保证，对于时序性具有很高要求的 H.264 编码来说，却提出了很大的难题。针对流媒体的特点，由 Real Networks 和 Netscape 共同提出的 RTSP（Real-Time Streaming Protocol，实时流协议）的出现，对控制声音或影像，并允许同时多个串流需求控制提供了可能。该协议定义了一对多应用程序如何有效地通过 IP 网络传送多媒体数据^[25]。RTSP 在体系结构上位于 RTP 和 RTCP

之上,传输时所用的网络通讯协定并不在其定义的范围内,因而它可以使用 TCP 或 UDP 完成数据传输^[26]。随着这几年对实时流媒体数据需求的增加,RTSP 被越来越多的使用,如视频监控,实时直播,视频点播等,特别是移动平台对于 RTSP 的良好支持,使得其更多的成为了一种规范,更加促进了它的发展。

2.2 相关工具

C#^[27]是由微软开发的高级程序语言,它具有面向对象编程语言的特点,很好的汲取了 Java 语言的优点,对于项目开发和再利用有很好的实现效果。特别是其中标准类库的实现是在长期经验总结的基础上不断地进行优化改进,在项目开发的过程中减少了对它们使用时的检测和修改的周期,提高了实现的效率。

此外,在对 C 和 C++代码的兼容上和实现上,C#语言也有很好的表现。特别是对于指针的处理和实现,不同于 Java 语言完全的封装起来,C#语言为用户提供了可以直接访问内存的指针实现,这在高级面向对象语言中是不多见的。对于函数参数的处理,C#语言也支持 out 和 ref 的标示,即可以明确地指示参数的性质和作用,同时在实现时对于不同的标示有不同的处理和实现。例如对于 out 标示来讲,所修饰的参数必须要在函数返回之前进行赋值。

特别的,作为 Windows 平台上开发的系统,C#语言与生俱来的占有优势,作为微软一脉相承的软件工具,它可以更好的、更有效的实现与 Windows 系统的交互,实现对 Windows 系统底层和内部的调用和实现,这是其他的高级语言所达不到的^[28]。

MySQL^[29]是开源项目所实现的数据库,相较来说,它具有更灵活和轻便的特点,同时又不失功能的实现,所提供的功能满足了对数据库实现的基本需求,在对于数据库需求不算巨大的项目开发中,MySQL 开源性的特点是实现项目需求的不二之选。

CoreAVC^[30]是比较著名的 H.264 视频解码库,其他的还有如 ffdshow^[31]等。作为现阶段仅需要实现高效快速的解码工作,无需对大型开源项目 ffmpeg^[32]进行分析和研究,当然其中大量的优秀的 Codec 还是对于以后的项目实现具有重要的意义。

使用的测试图像大小为 D1 (720x480), profile 等级为 main, level 等级为 5 的四路实时数据流分别对 CoreAVC 和 ffdshow 进行测试。两者在实时性的表现上都很好,画面的延迟在半秒左右(考虑视频编码耗时和网络传输延时),且解码画

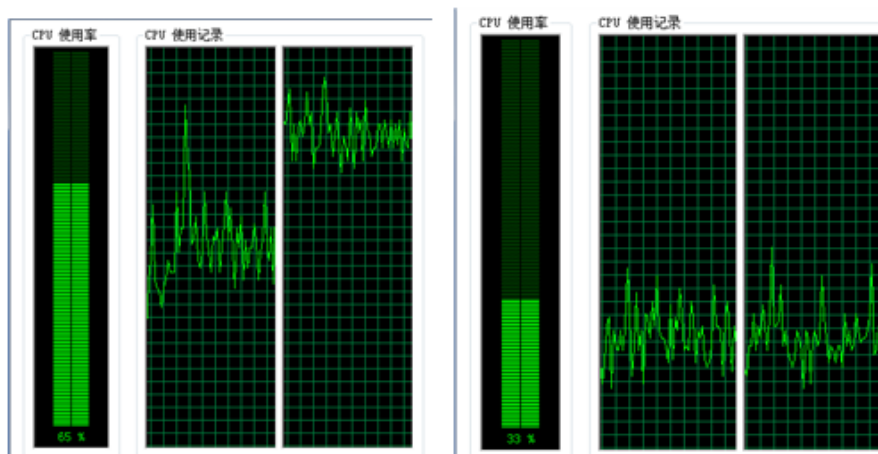
面的质量也很清晰，长时间播放并没有出现花屏和跳帧的现象，如图 2-1 所示。



左边使用的是 ffmpeg，右边使用的是 CoreAVC

图 2-1 实时播放效果比较

但由图 2-2 可见，CoreAVC 在解码过程中对 CPU 的整体使用情况要远远好于 ffmpeg，其一，在 CPU 的占有效率上，CoreAVC 只有 30%-40%，而 ffmpeg 则达到了 60%-70%，明显高出了前者一倍；其二，在 CPU 核心的使用情况上来看，对于测试所使用的双核 CPU，CoreAVC 对于两个核心的负载比较均衡，区别不大，但 ffmpeg 不但有明显的负载不均匀且波动剧烈，这对 CPU 效率和性能的影响是显而易见的。另外经测试 CoreAVC 可以实现九路的 D1 正常播放，而 ffmpeg 当使用六路 D1 进行播放时，就已经出现画面抖动，显示不稳定的情况了。



左边使用的是 ffmpeg，右边使用的是 CoreAVC

图 2-2 实时播放 CPU 效率比较

可见 CoreAVC 具有更好的解码性能和更低的 CPU 负载，在对视频解码播放时具有更高的表现性。

2.3 本章小结

本章对视频监控系统实现所需要的相关技术做了简要分析，特别是对 H.264 编码格式做了详细的叙述。这对于系统的实现具有重要的辅助作用。另外还对系统实现所用到的工具做了介绍，对解码库的选择和使用还通过实验测试进行了分析，选取了能够更好的为视频监控系统服务的动态解码库，这些对于系统运行的稳定和系统性能的提升具有重要意义。

第三章 视频监控系统的需求分析

本章对铁路系统的物理环境和实施条件做了详尽的分析，并对其业务逻辑结构和物理线路结构等做了详细的介绍，同时对在铁路系统中需要实现的视频监控系统所要完成的功能和性能进行了切实的需求分析，对可能存在的问题和难点做了详细的介绍。以期对后边系统的设计和实现做好充分的准备工作。

3.1 铁路系统结构的分析

视频监控系统，就是要实现视频的采集、处理、传输和播放。在铁路系统实现视频监控系统不同于传统意义上的视频监控系统，它所覆盖和实现的范围将更加广泛，所需要考虑问题涉及的方面也更加多样，因此要实现视频监控系统良好的稳定性和灵活的扩展性，对系统整体设计需求的详细分析将尤为重要。

3.1.1 物理网络结构

视频监控系统功能的实现是建立在视频监控系统良好的框架之上的，因此在对具体功能需求分析之前，要对视频监控系统所要实现的环境和整体结构进行详细深入了解，这样才能为实现视频监控系统架构的合理性和功能的灵活扩展性给予充分的补充。

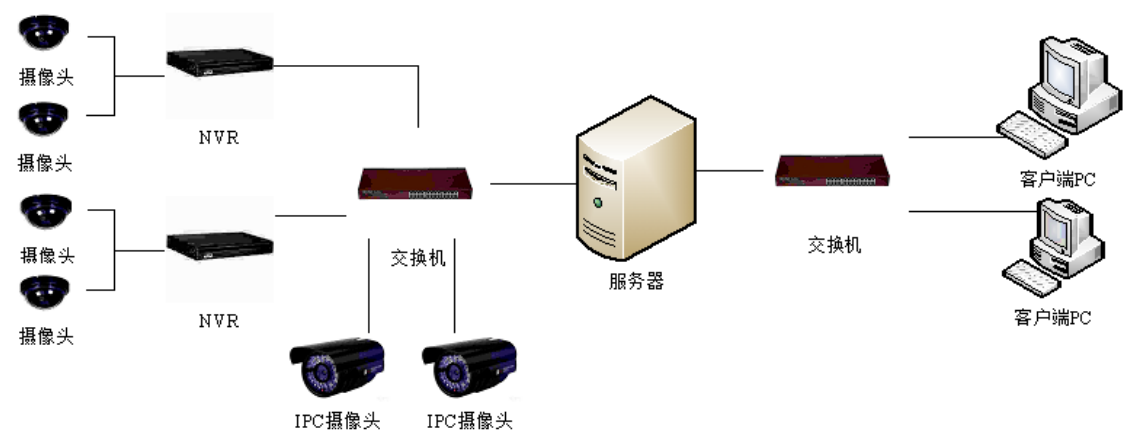


图 3-1 视频监控系统的一般实现

视频监控系统一般由前端（视频设备）、传输（一般为网络）、服务器和客户

端所组成，几部分相互协作，组织成一个整体如图 3-1 所示，在此基础之上实现对视频的具体功能的实现^[33,34]。

前端，就是视频设备，在此次的视频监控项目中所使用的视频设备有：DVR（Digital Video Recorder，数字视频录像机）^[35]、NVR（Network Video Recorder，网络视频录像机）^[36]和 IPC（IP CAMERA，网络摄像头）^[37]通过自身或所连接的摄像头（一般通过视频数据线直接连接，但有些为了更好的效果也会经过光电转换以光信号的形式通过光纤进行传递）获得视频数据。这些设备通过对专有的动态库的调用以网络作为媒介实现服务器同它们之间的交互。由于摄像设备种类的千差万别导致每种设备都有其各自不同的动态库和调用过程，同时还由于不同的提供厂商基于种种商业原因的考虑对视频数据进行封装处理，也会给视频数据的获取和统一增加了不小的难度。

服务器是视频监控系统的大脑，在整个系统结构中占据着核心的位置，是整个系统实现功能的关键，也是本文所叙述和实现设计的主要部分。对服务器而言，一方面它通过网络与视频设备连接，通过调用视频设备厂商所提供的动态库来实现视频实时数据的获取和视频设备的交互。获得的实时视频数据经过服务器的分析与处理成为可供客户端播放的数据源。另一方面，服务器同客户端之间建立起网络连接，通过连接实现双向的交互，把客户所需要的视频数据传送给客户端，同时对客户的发起的命令和请求实现响应和处理。

客户端是与用户打交道的最直接的一端。可以说客户端表现的性能的好坏是给予用户最直接的感受。客户端所实现最大的功能就是视频的播放。同时为了更好的友好性和灵活性，在客户端也需要实现交互功能，例如，PTZ 云台控制，通话对讲等。

实现远端视频监控数据的访问是保证对铁路系统整个辖区全方位统一管理和监督的关键，因此在一般的视频监控系统实现的基础上，设计构建立体化、网络化的视频监控系统架构就成为了实现所必须解决的问题，而这之前就需要对成都铁路局现阶段提供的视频网络结构有详细的了解。

在铁路系统中，视频监控所要监控的视频设备所在点往往都是铁路沿线的站点，而在这些站点中，由于所处的环境和业务逻辑关系其地位和作用都是不同的。成都中心作为成都铁路局的调度和指挥中心，发挥着绝对核心的作用，因此，把成都中心看作是整個成都铁路局物理网络的核心所在。铁路系统的管理往往是根据铁路线路划分的，由于铁路线路环境复杂，条件恶劣的特点，在现阶段的条件下，难以实现将铁路沿线的所有站点接入到成都中心来，因此往往是在业务逻

辑上处于管理层次的站点或者是重要的大客站作为物理网络的节点接入到成都中心来，如图 3-2 所示。比如在成渝线上，有多达十多个站点，这些站点分布在铁路沿线上。其中永川和江津两个大站由于其位置和业务逻辑的重要性，在铁路系统中占据着重要的作用，因此为了保证线路的通畅和信息传递的及时，这两个站点有单独的物理网络线路与成都中心相连接。因此，将这样的有独立网络连接链路连通上级层次站点的站点称之为汇聚点。

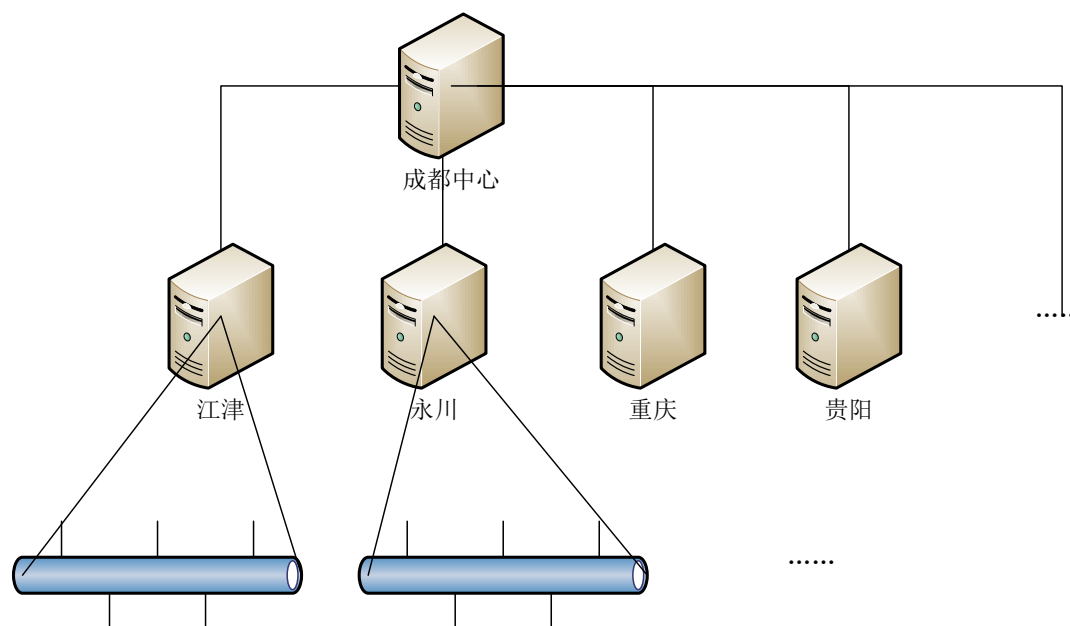


图 3-2 成都铁路局物理网络概图

沿线的站点中，还有大量的其作用和地位相对大客站要弱很多的小站点，它们并不会也没有必要接入到成都中心。一方面线路铺设的投入太大，另一方面所涉及的功能可以有相近的汇聚点所统筹管理，这样的情况也是符合铁路系统中业务逻辑结构划分的。如图 3-3 所示，是江津段物理网络连接的示意图。

可见，在成渝线江津段所连接的物理网络站点中，并非是星型结构的实现，即在站点之间并没有独立的物理线路与汇聚点相连接，而是环状的连接结构。需要注意的是，在铁路视频改造还未完工的条件下，物理链路都只能提供双向 2M 的物理带宽，所以对于视频数据的传输就显得捉襟见肘了，同时，对于连接上的每个站点其本身都不会对物理链路的连通有所影响，也就是说，对于环上的所有站点对于物理链路带宽的占用将是共享的。

因此对于每个站点来讲都可以看作是一道闸门，而这道闸门的限制是一定的，对于通过这道闸门的数据流量是分为上行流量和下行流量，两者之间是独立的。

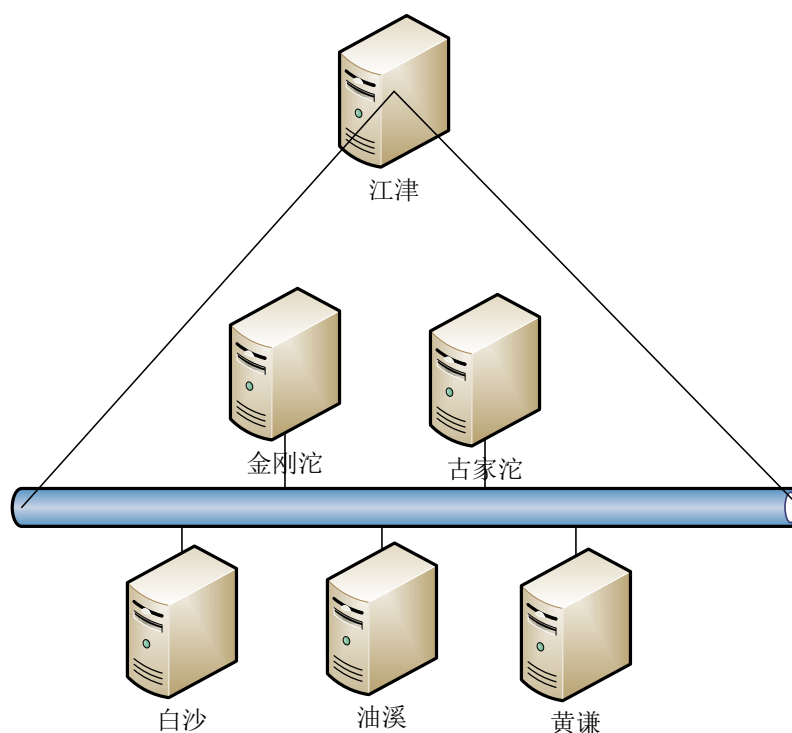


图 3-3 江津段物理网络连接的示意图

例如，当金刚沱的数据经由白沙流向江津时，会占用白沙到江津的数据带宽，因此为了保证视频数据的传输质量，对于每一道闸门都需要有判断机制来确保带宽的使用情况。这带来的后果就是必然导致视频数据的可访问的数量的限制，就需要提高视频数据的利用效率，特别的对于视频数据就需要实现对相同访问资源的共享，来提高网络带宽的使用效率。

3.1.2 逻辑结构

在铁路系统中，业务逻辑不同于物理结构的分布。在逻辑结构中，可以将成都铁路局系统划分为三级：局，段，站。

1) 局，对成都铁路局而言，这是唯一的，这也是最高的管理部门，对于整个成都铁路局辖区具有统一调度，统筹安排的职能。

2) 段，由于铁路的特殊情况，所有站点都是沿铁路沿线设置，而如果将所有的站点都设置在局之下，不光管理起来有难度，也会大大增加系统的冗余，更将对系统的效率大大降低。所以将有联系的一些站点组合在一起统一管理，而这些就是段。

3) 站点，顾名思义，就是铁路沿线所存在的站点，每个站点都有自己独立的

管辖区域，对其所在区域负责。而对于视频监控系统而言，每个站点都会成为系统中的一个节点。

3.1.3 业务结构

根据业务功能的不同，铁路系统可以划分为车务部门、电务部门、机务部门等，而这些部门又有各自独立的结构划分。这种划分既依赖于物理结构，更与系统的逻辑结构密不可分，在铁路系统中，业务工作和职能履行的实现往往是以这种结构的划分作为基础。由于各自部门间业务的独立性，部门之间几乎没有关联，明确职责也使得各部门之间业务更有针对性。所以在系统的设计时，对于不同区域的摄像设备的访问就需要具有业务的针对性和独立性。例如，电务部门是管理供电相关的业务，所以对供电室的摄像设备就需要有观看的权限，但对于站台上的摄像设备，电务部门应当由于权限限制，将不具有查看的能力。因此由于业务部门的不同，它所能观看的摄像设备也将有所区别。

3.2 功能需求分析

视频监控系统最基本的功能就是实现实时视频的观看，没有这点视频监控系统也就失去了它的意义。在铁路系统中，突发情况和特殊需求往往很多，对于视频回放也有很高的需求。此外，由于铁路系统的特点，在成都铁路局现阶段的设备条件和环境下，也需要系统实现权限判定和带宽控制的功能。

因此在服务器的设计过程中，用例描述如下：

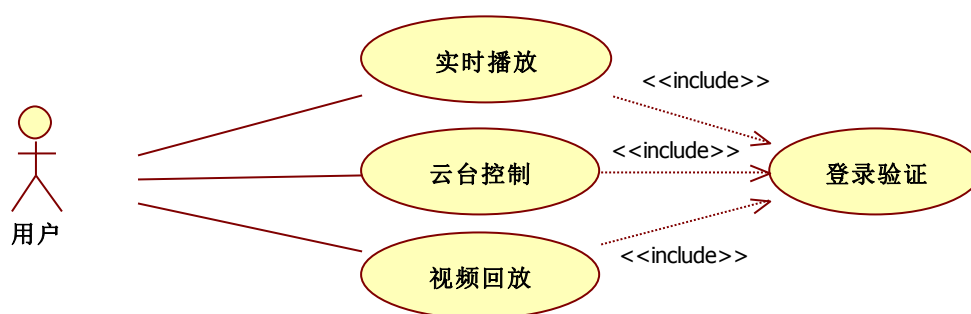


图 3-4 服务器功能用例图

用户（这里可能是实际的登录用户，也可能是远程服务器的虚拟用户）能够通过服务器完成实时播放和视频回放的功能，具体的细节将在 3.2.1 小节和 3.2.2

小节来详细介绍，当用户发起请求时，服务器会对用户请求的信息进行分析，对所需要的数据进行处理，反馈给所需要的用户。此外，对于用户，还可以通过服务器对设备进行云台控制的能力。而这些都是需要在经过了登录验证的条件下完成的。

对于设备的处理也是服务器需要重点考虑的内容，如图 3-5 所示：

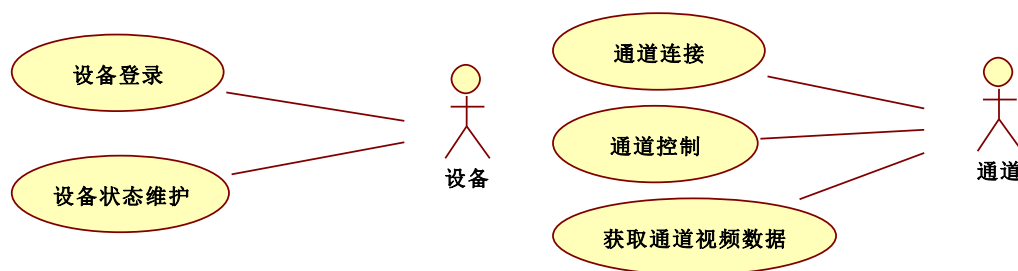


图 3-5 服务器设备和通道用例图

设备和通道是服务器视频数据的主要来源，因此对于设备和通道的参与与交互成为了服务器实现至关重要的部分。设备登录实现了与设备的连接通信，设备状态维护保证了与设备之间连接通信的正常，这其中既包括了设备本身的信息还包括设备所包含通道的状态信息。通道是实现视频监控的基本对象单位，因此不仅仅要实现连接的正常化和视频数据的获取，同时作为云台控制的载体，它也需要保证对于控制方面联系和实现的工作。

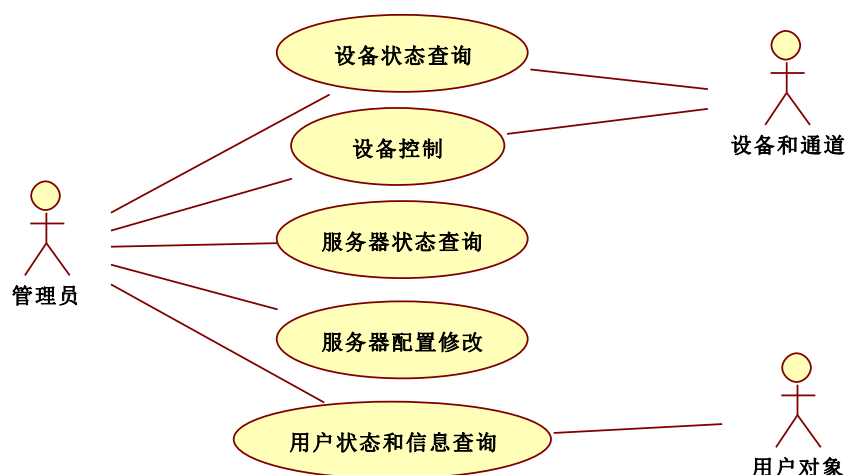


图 3-6 服务器状态维护用例图

服务器还需要实现良好的交互界面，那么对于服务器状态也需要有比较全面的查询和维护功能，如图 3-6 所示：

作为服务器的管理员，应当必须能够通过服务器查看当前运行的状态信息，并对服务器进行有针对性的配置和处理工作，而这也是可视化界面所要求的。因此这就需要通过设备状态查询和设备控制来实现与设备和通道的交互工作，另一方面，对于用户的状态信息可以通过查询过程来实现交互。此外，对于服务器来讲，还需要对于服务器本身的运行和状态信息实现查看和配置的功能。

3.2.1 实时播放

在视频监控系统中，实时播放占据着无可替代的地位，它关系到视频监控系统存在意义，同时也是视频回放必不可少的重要前提。服务器与视频设备连接并从其获得实时视频数据，再对这些数据进行处理之后，将他们发送出去，并在客户端流畅稳定的播放。显然这就需要如下的过程：

1) 与设备的连接，系统中所使用的设备是多种多样的，为了更好的体现视频监控系统的性能，那么就需要对视频设备的连接和数据的处理有长远性的考虑。由于各式各样的、各种厂商的视频设备都会不断的接入系统之中，设备的连接和数据协议并没有统一的格式，因此在系统设计时就需要考虑系统对于设备连接的包容性、统一性、灵活性和扩展性。

2) 数据通道的连接，一般的在设备连接之后将会进行数据通道的连接，而这也是实现视频数据获取的最直接最根本的途径。对于现在的视频设备而言，硬件性能已经大大加强，特别是针对视频图像处理技术的加强，对一个摄像头所捕获的视频画面可以直接通过编码实现两路通道的视频输出，一路为主码流，这也是决定了视频的最高质量，而这一路视频也将会被保存下来用于视频回放的数据源；另一路为辅码流，是对主码流的数据进行抽取压缩后所获得的低质量的视频数据。由于较低的数据量，一般的对于远端的数据传输将会选择使用。

3) 数据的处理和发送，来自于本地视频设备或远端服务器的实时视频数据经过统一的格式化处理后，将不再具有视频设备的独立特性。通过对其进行加解密处理之后，通过网络连接发送到客户端用于播放。由于每个客户进行实时播放时对数据的请求会有相同的可能，因此实现数据的分发增加数据的利用效率，有利于节约网络带宽。另外，由于 H.264 编码的特性，在数据发送时为了保证流畅性需要对发送的数据进行判定和控制。

4) 客户端的播放和交互，视频播放是视频监控系统实现功能的最根本的目的。而保证视频播放的流畅性和实时性是视频监控系统性能的最直接的评判标准。因

此需要对接收数据缓冲区的控制和播放速率的调节进行设计和实现，来满足良好的视觉效果。

3.2.2 视频回放

如今，生活节奏越来越快，信息获取方式越来越便捷，时间变得越发珍贵，因此对于视频信息数据的存储和多次利用就变得日益突出，特别是对地域广阔的铁路系统而言，存在人力不足，环境恶劣，条件简陋等诸多不利条件下，要做到实时的预测和关注几乎难以实现。那么当问题发生时，能够及时的回放当时的过程则会给管理工作和调度工作提供有效的辅助手段。所以视频回放功能的实现对于视频监控系统的意义不亚于实时播放。

1) 回放，就是把过去发生的视频再次的呈现在众人面前，同时还要为这种呈现的方式提供有效的控制手段，如暂停，加速等。特别的，这种回放方式不仅需要实现本地的回放，还要对于远端回放也具有实现的可能，即，如实时播放一样可以去回放远端的视频数据。

2) 回放的视频数据，是存储在本地的视频数据，这些数据来自于实时播放。由于之前所述，视频设备的多样性也导致了所得到的视频数据格式的多样性。每个厂商为了自己的商业利益必然的对视频数据做了个性化的处理，例如在 T 厂商的摄像头所提供的数据是按照固定的数据包的大小来传输的，而 D 厂商的数据是按照一帧帧的数据发送的，因此每个数据包的大小都是不固定的。有些厂商使用的是 8 个字节的数据包包头来描述数据包信息，而另一些则使用的是 16 个。这些种种都需要进行统一化的处理，这样在实现视频回访时对于视频的处理过程才能简化且更有效率。

3) 在进行回放时，提供了速度控制的功能，那么在实际的网络传输过程中，由于播放速度的变化，会导致视频在传输时的码流发生变化。在速度变慢时，码流也跟着降低，而在速度升高时，码流也会随着单位时间内发送的帧的数量的增加而增加，因此会对有限的带宽造成巨大的影响。

3.2.3 权限控制

权限控制是系统得以正常运行的保证，它存在于系统设计的各个方面，只有确定了合理的权限结构，才能够在系统的正常使用过程中不会出现混乱和不明确的情况。同时权限还有区别用户等级重要作用。

1) 用户的访问权限,应当同用户所属的部门相一致。即,对于某个部门而言,它只能使用属于它管辖的摄像设备。当摄像设备加入到 A 部门后,那么所有属于 A 部门的用户都有权利来访问此摄像设备。

2) 用户的所属部门是可以多个的,这样也灵活的实现了用户权限的控制。同时,摄像设备的所属部门也是可以灵活添加的,这样可以将某个摄像设备实现灵活的部署。

3) 用户权限控制应当符合用户的逻辑等级,即,铁路局的局长用户所能看到的 A 部门的摄像设备应当包含某一段长用户所有权限的 A 部门的摄像设备。

4) 不同用户在同一业务部门中,同一用户在不同的部门中,都需要提供不同的权限等级,这样才能保证在出现资源竞争时可以灵活的控制资源的取舍方式。

5) 不同的用户,由于其需求不同,对于他们应当提供有限制的交互操作能力。例如,对于远程的用户,为了保证带宽,一般不具有回放功能。

6) 权限的设计与实现应当满足物理结构和逻辑与业务结构的灵活性。即保证关联和相互之间的耦合不高,便于扩展和修改。

3.2.4 带宽管理

远程视频数据的访问最大的难度在于带宽处理,特别是在现阶段,铁路视频改造还没有完成,站点之间的链路只提供双向 2M 的物理带宽。因此,对于视频播放这种极大占用带宽的功能实现,就需要良好的带宽处理和控制机制,来保证数据流量的正常,以确保视频监控的稳定实现。

1) 对铁路系统物理网络结构了解与分析是实现带宽处理的根本,特别是,当物理链路上存在带宽共享时,就需要对经过节点的上行和下行带宽做重要的分析,来保证带宽流量在最大带宽允许的范围内。

2) 视频设备共提供两种编码格式的视频数据,一路高质量的,一路低质量的。在测试的过程中发现,不同的码流格式数据所占用的带宽的平均值是不一样的。具体的测试结果如表 3-1 所示,其中 D1, 720P, 1080P 表示设备所能提供的最高视频质量,而低质量的视频选取采用 CIF 格式来确定。可见对于 2M 的物理链路带宽而言,只能实现 4 路 CIF 格式的画面或者一路 D1 格式画面的播放,为了实现更好的监控表现,对于远端的视频访问,都采取 CIF 格式的数据码流。因此,将带宽处理的控制基数以 512K 作为 1 路视频, D1 为 4, 720P 为 8, 1080P 为 16。因此 2M 的带宽所能提供的最大视频播放为 4。

表 3-1 视频设备码流质量

	高质量	低质量
D1	2M	512K
720P	4M	512K
1080P	8M	512K

3.3 性能需求分析

系统功能的实现是基本，而为了使这些功能能够流畅稳定的运行，提高客户端的友好性，增加灵活性和信息处理的效率，还需要对系统在很多方面进行优化和完善工作。

3.3.1 视频播放的流畅性和实时性

视频播放，特别是在监控系统中，为了实现对突发情况及时有效的处理，这也是实现视频监控系统的初衷之一，因此对于视频播放画面的实时性具有比较高的要求，视频的延迟不应当超过 2 秒钟。为了实现高实时性的视频播放，那么最好的实现效果就是在收到视频数据的时候立刻对其进行播放。但往往这样做的结构将是对画面播放的流畅程度产生严重影响，特别是在网络环境不好的情况下，这种影响尤为突出，严重的阻碍了视频监控的观感。

其原因，有以下几点：

1) 视频数据是按照一帧帧的数据进行网络发送的，那么对于多用户的连接过程时，数据的发送并非是完全的按照稳定的循环调度的顺序发送的，一旦发送给某个客户的数据由于调度周期的影响产生了延迟就会导致视频出现不流畅的情况。

2) 网络的传输环境会有波动，一段时间内，如果网络拥塞，则会使得数据包需要经过较长的时间才能到达客户端；反之，则会导致数据包迅速的到达客户端。这种波动的情况将会对视频画面的播放产生时快时慢的卡顿现象。

3) 视频数据在客户端接收到的速率无法也不可能保证满足视频数据的帧率，如果对数据采用直接播放的方式必然会导致视频的不连贯。

因此综上所述，就需要一方面在客户端设计视频播放的控制方式和策略来保证视频播放时的连贯性，另一方面，在服务器发送数据时设计满足 H.264 编码数据特性的控制方式，保证视频数据的实时性。

3.3.2 服务器运行的可靠性

服务器需要保证 24 小时不间断的运行，同时服务器在运行的过程之中不可避免的会发生错误，一旦错误出现导致服务器崩溃，就需要及时的将服务器唤起，另外视频设备的连接状态和数据通道的连通情况也需要保证在系统运行的过程之中稳定的运行，而这些工作无法依靠人力来完成实现，就需要有专门的程序来保证服务器运行的可靠性。

一般的对于服务器的检测守护进程能够很好的完成这项任务，它在运行过程之中可以不间断的查看服务器的运行状态，测试服务器的功能运转的正常性。特别是在出现问题时，能够及时的将服务程序唤醒，同时将信息记录下来并通知管理员检查。

守护进程在保证系统稳定可靠的同时，还能够发现系统运行出现的 Bug。由于程序的编写必然存在这样或那样的不合理或者是考虑不周到，但这些隐含的问题，是通过简单的测试工作无法真正的发现的，只有在真正的运行过程之中才能够发现问题。所以在程序出错崩溃时记录程序的错误状态有利于对系统功能的改进。

在 Windows 平台上，将守护进程以服务的形式存在，在系统启动时，自动的加载服务启动守护进程，将守护进程的维护工作交由系统来完成，有更好的稳定性。

设备和通道的检测由于动态库本身的实现不具有绝对了稳定性，因此也需要在服务器的设计和实现中对于连接通道的数量和每个通道的实时状态予以判断，来保证在系统运行的过程中不会出现由于设备连通问题而导致的视频数据访问失败的情况。

3.3.3 系统服务器的响应效率

服务器要完成大量的工作，包括对数据的处理，同客户的交互等。那么当客户请求和视频请求增加时，需要保证系统服务器的响应效率，来确保数据信息及时有效的传输。而系统的相应效率需要考虑如下几个方面：

- 1) 客户命令的处理过程需要确保及时有效。
- 2) 数据信息的查找和用户信息的查找由于存在对共享资源的访问，需要保证互斥性，因此需要对查找过程进行优化以尽量减少系统由于查找耗时等待而导致的连接超时情况。
- 3) 系统应当保证对大量客户连接处理的及时响应。

3.4 本章小结

本章对视频监控系统所实现的环境做了详细的分析，特别是对铁路系统的物理网络链路结构和逻辑业务结构做了具体的阐述，这在系统设计时，对整个系统框架的实现都有很大的帮助作用。同时还对系统所实现的功能做了主要的分析工作，分为实时播放、视频回放、权限控制和带宽管理四个部分进行阐述。功能的实现是基础，要实现系统的稳定、可靠良好的运行还需要对系统性能作充足的分析工作，这些工作的完成，对系统设计和实现的作用是显而易见的。

第四章 视频监控系统的设计与实现

4.1 系统整体设计与实现

视频系统采用 C/S 结构进行设计，基于 C#语言开发，使用开源数据库 MySQL 作为数据库访问。整体结构如图 4-1 所示。

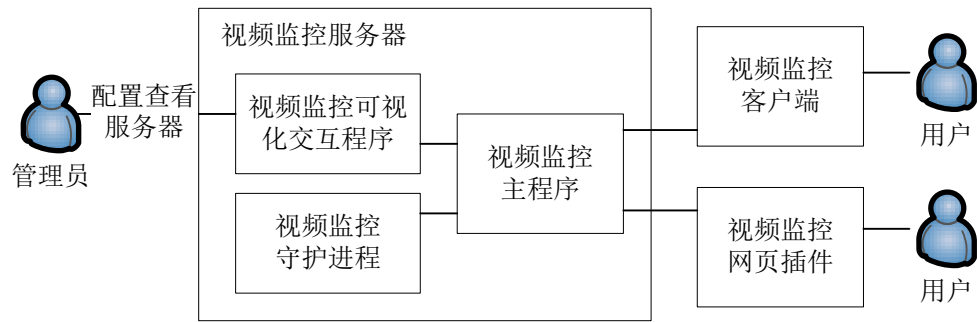


图 4-1 系统总体设计结构图

服务器端的设计，主要由三个部分组成：

1) 视频监控主程序，这是视频监控系统实现功能和提供服务的重点所在，它完成了对实时播放、视频回放、权限控制和贷款管理等主要功能和其他的一些辅助功能的实现。这也是本文所叙述的最重要的部分，其主要的设计和实现将在后边详细阐述。

2) 视频监控守护进程，是视频监控系统长期运行的安全保障，它能够在系统出现问题时及时的恢复系统的状态并将信息记录。特别的，视频监控系统要在服务器启动时一起运行，将以系统服务的方式来实现。

3) 视频监控可视化交互程序，是管理员与视频监控主程序交互的直接界面显示端，一方面为了交互的友好性，可视化是最好的实现方式；另一方面，以系统服务实现的守护进程会导致视频监控主程序在启动后桌面交互失效，所以设计单独的可视化交互程序来实现视频监控系统的配置和查看。

客户的访问提供了两种方便的途径，独立的客户端程序和 Web 端的 IE 视频监控插件。其实现的主要功能分为两部分，与服务器的连接与交互和对视频的播放与控制。

4.1.1 系统物理结构的设计与实现

根据 3.1.1.1 节对系统物理网络结构的分析和 3.1.2.3 节与 3.1.2.4 节对权限控制和带宽管理的分析。在对服务器整体物理结构的设计与实现中，需要在物理链路上保证带宽的控制和设备访问权限的控制，因此对于系统整体物理结构的设计如图 4-2 所示。

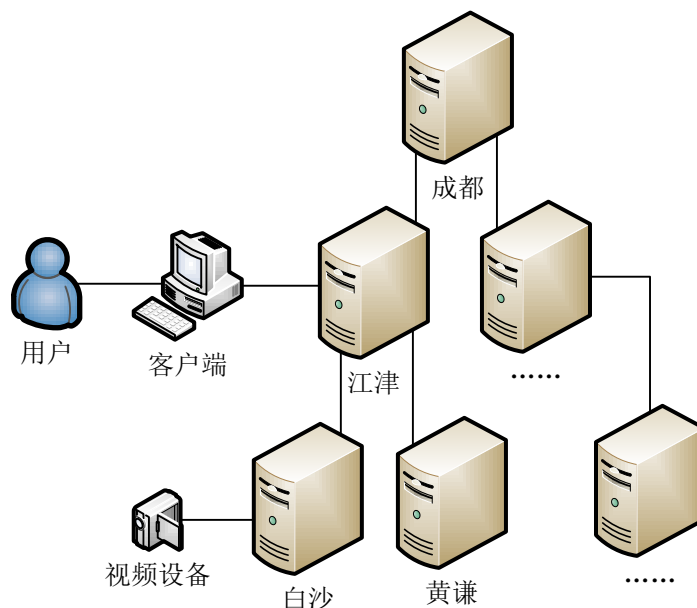


图 4-2 系统物理结构设计图

在物理网络中，每条单独的物理链路都应当具有单独的带宽管理控制判断，对于成都中心而言，与它相连接的每个汇聚点都应当具有独立的 2M 带宽，而且成都中心更多的是作为总的汇聚点而非转发点存在，因此，将星型结构的网络设计成树状结构的实现。

对于如图 3-3 所示的物理网络，将江津作为父节点实现，而它有两个子节点，分别是白沙和黄谦，金刚沙和古家沱分别作为两者的子节点再被添加进来。也就是说，将环从中间分开，线性的向两边添加子节点，通过抽象上级和下级服务器来实现带宽的管理工作。

因此，每个节点都只存在一个父亲节点和多个子节点，每个父亲结点就是服务器所对应站点的“上行”服务器站点，而每个“下行”服务器则是用子节点表示，而对于那些有多条独立线路的服务器而言，它就存在多个子节点。每个服务器就会将自己上下级的关系存在所在服务器的数据库中，这样每个节点都只与自身的服务器信息有关，而无需设计单独的中心服务器。树中的每个节点服务器所

在的内部网络都自成一个独立的视频监控系统，可以不与其他的节点进行交互实现本地视频设备的访问。这样，在网络出现状况时，每个子节点独立存在的网络内仍然可以保证其自身的正常使用。

当用户想访问视频设备，在登录江津服务器的情况下，只有通过江津服务器间接的访问白沙服务器来实现对于设备的访问，而这种间接的访问，称之为代理模式，即江津服务器作为一个特殊的客户端（这样，对于服务器而言，它对客户的请求可以统一化，也保证了，在系统的任何时刻，其它的服务器都可以随时接入进来）连接至白沙服务器，白沙服务器将用户所需要的数据返回给江津服务器，这样接入江津服务器的客户就可以得到视频设备的数据了。通过这种方式，可以在江津服务器上加入判断机制，通过计算发出的远程数据访问请求的量化带宽来确保带宽的控制。

此外，设计代理模式的另一个目的是实现数据的分发。如果此时还有用户通过江津服务器要获取相同设备的数据（可以是连接江津服务器的其他客户，也可以是连接江津服务器的其他服务器），江津服务器就可以通过分发来实现数据的共享，这样可以有效的实现资源复用的效果，对提高数据和网络带宽的利用率具有巨大作用。

4.1.2 系统业务逻辑结构的设计与实现

系统中业务逻辑结构是通过两方面来实现的，其一，通过设计数据库来实现用户，部门和设备的业务关联关系；其二，通过服务器的层次结构来实现用户等级和设备访问权的逻辑结构关系。

4.1.2.1 业务关联关系的设计与实现

在数据库的设计中，将部门作为中间的桥梁，“连接”用户和设备。每一个用户都可以确定自己所属的部门，部门的数量并没有限制，在每一个部门中由于用户的职位不同他所拥有的能力就有所差别，因此所能执行的操作就有限制；而将设备，具体来说，是视频通道信息添加到具体的部门中，这样凡是属于此部门的用户都将能够对这些通道具有访问的权限。

如图 4-3 所示，`t_user` 表是用户的个人信息表，它是所有与用户数据相关的所有表信息结构的核心和基础，其中所记录的是用户最基本的个人数据信息。`t_directionary` 是部门表，它所记录的是铁路系统的所有部门信息，所有和部门相关的数据信息都将这个表作为所有处理业务的基础和核心。`t_channel` 是通道信息

表，记录的是所有设备的通道信息，其中包含外键来确定所属的设备信息。系统中所有与设备有关的信息都将此表作为基础。

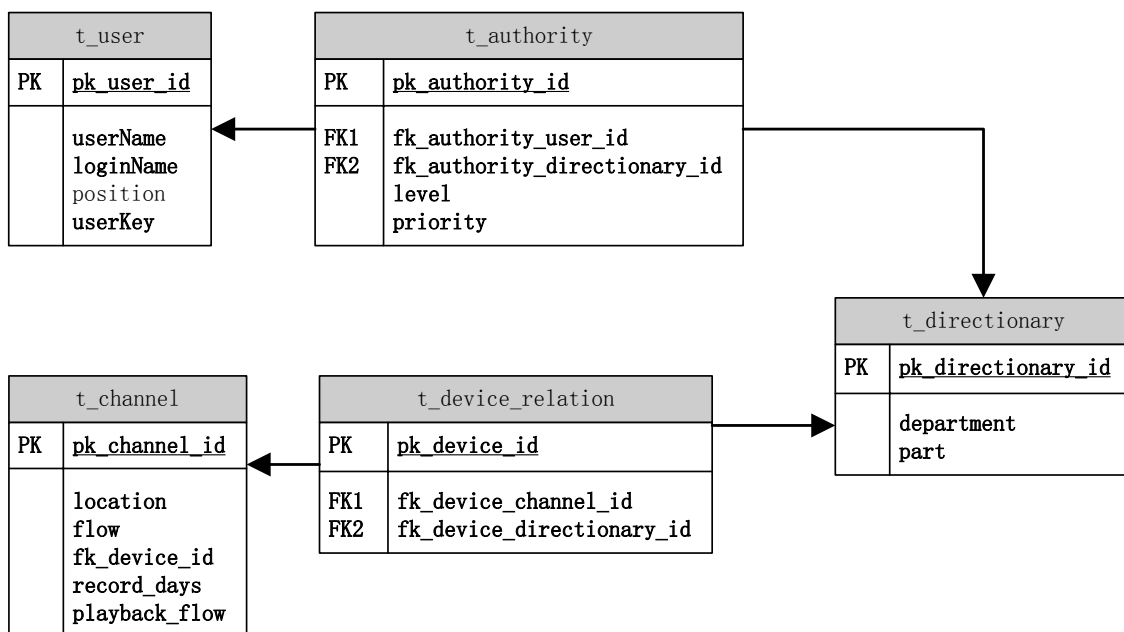


图 4-3 数据库业务关联关系的设计和实现

为了实现业务逻辑的关联，就需要实现用户表与部门表，通道表与部门表的关联关系，通过部门表来实现两者的统一。其中 $t_authority$ 是用户的权限表，表中的每条记录表明用户所属的部门和拥有的权限。 $t_device_relation$ 是通道的归属表，其中的每条记录表明此通道被设置于哪些部门能够访问。当通过对同一部门的用户和通道查询时就可以确定有哪些用户可以实现对此部门下的哪些通道有访问的权利了。

4.1.2.2 逻辑关系结构的设计与实现

在逻辑结构层次中，上级站点对下级的站点拥有管理的统筹安排的能力，因此在系统的设计时，也应当保证这样的逻辑关系存在。

用户的注册会在不同的服务器来实现，其对应的业务逻辑在哪一层服务器层次，那么就将用户注册在此服务器的数据库上。上级站点注册的用户对此站点所属的下级站点具有设备的访问权限，而此访问权限的实现取决于用户所属的部门在下级服务器中包含摄像设备。

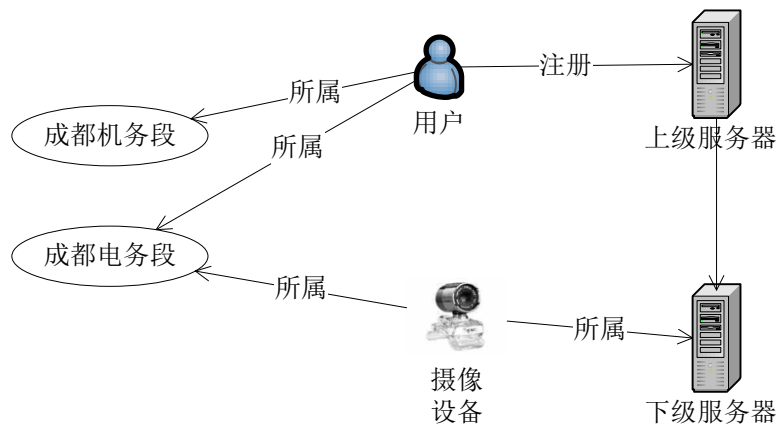


图 4-4 逻辑关系示例图

如图 4-4 所示，用户是注册在上级服务器数据库上的，他属于成都机务段和成都电务段两个部门（这两个部门都是整个系统公有的），摄像设备是注册在下级服务器上的，是下级服务器的本地设备，而这个设备只注册在成都电务段上。因此用户在成都电务段下能够找到摄像设备，对它具有访问权限，而在成都机务段中则没有摄像设备。

通过这样的设计可以灵活的实现设备的可见性，方便的调配用户对于部门设备的统一管理，降低了耦合性，利于信息的修改和扩展。

4.2 视频监控主程序的设计与实现

视频监控主程序是实现视频监控系统服务端的核心，是本文介绍的主要工作，视频监控系统将实现两大基本功能，实时播放和视频回放，同时权限控制和带宽管理将融入其中，具体的设计与实现将分节详细说明。

对于服务器而言，它是连接设备与客户之间的纽带，一方面与设备或文件进行交互，从其中获得数据并对所得到的数据进行操作和发送；另一方面，从客户也得到交互的命令和消息，将他们解析后会反馈到对设备或文件上。因此将系统的设计划分为两大模块，数据处理模块和客户交互模块。结合功能分析的内容，系统服务端视频监控主程序的模块设计如图 4-5 所示。

客户交互模块是系统实现功能的主体部分，完成了实时播放、视频回放、权限控制和带宽管理等主要功能。在客户交互模块中划分为六个功能子块：网络连接监听和处理、客户主连接的交互、客户视频连接的交互、客户信息的维护和获取、客户视频信息的维护和获取以及视频设备和数据的交互。这六部分各自完成

自己负责的操作，在业务上相互独立，在设计上共同协作来实现系统主体功能和流程同客户之间的交互工作。

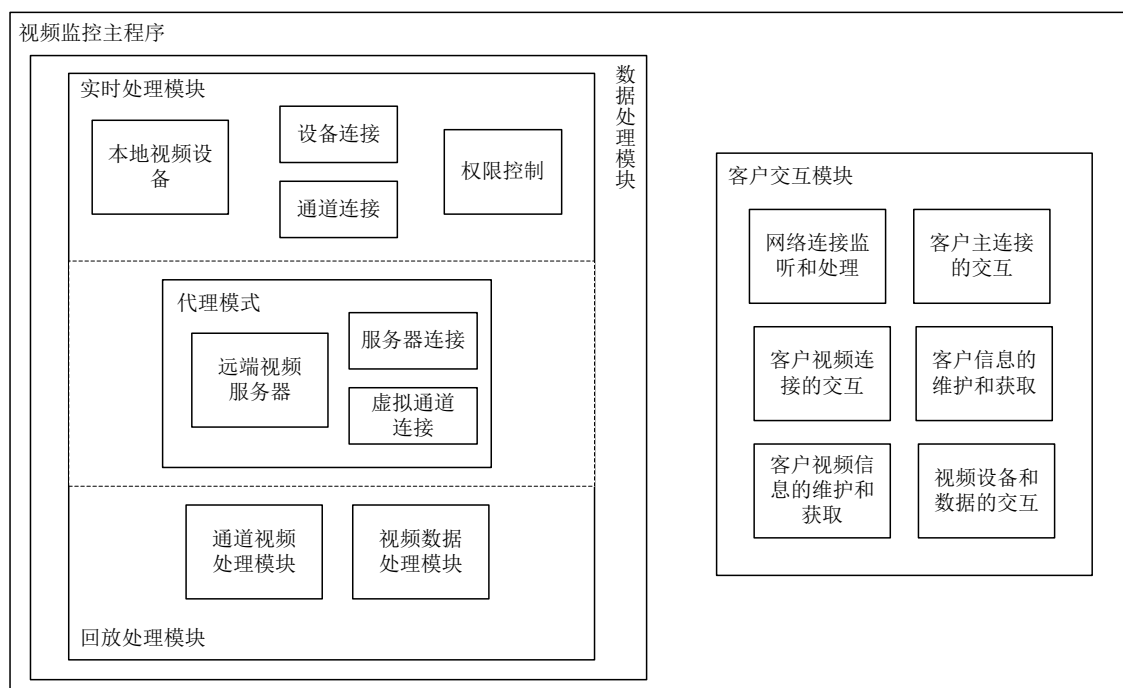


图 4-5 视频监控主程序模块设计

数据处理模块是系统对数据操作的主要实现部分，它完成了系统对于功能实现所需要的视频数据的管理和处理，根据不同的划分，数据处理模块有不同的组合形式。依据数据所实现的功能不同，可以将数据处理模块划分为实时处理模块和回放处理模块，两部分在实现上对于数据处理的对象是不同的，前者由于保证实时性需要从设备获得实时视频数据流，后者则需要根据客户的需要对保存在本地的视频数据文件进行操作和控制。依据数据来源的不同，又可以将数据处理模块划分为本地数据处理和远端数据处理，前者的对象是本地服务器所连接的视频设备或本地的视频文件，而后者则是向远程服务器发起请求，通过远程服务器来获得所需要的视频数据。

下面分别从设计流程和处理过程入手，对系统所需要实现的实时播放和视频回放功能做详细的分析与阐述，以期达到对详细的业务流程在总体上有比较系统的了解，在细节上有比较深入的分析的目的。在此基础上，分别对客户交互模块和数据处理模块进行详细的实现设计，分别从结构和类图入手完成系统的设计和实现工作，满足需求分析的要求。

4.3 实时播放的设计与实现

4.3.1 实时播放的流程设计与实现

客户要实现实时播放的功能将需要两个阶段，如图 4-6 所示，由于功能的实现需要两个 TCP 连接的建立，因此将两个连接的建立作为系统的对于功能实现最主要的部分：主连接的建立和处理过程以及视频连接的建立和处理过程。

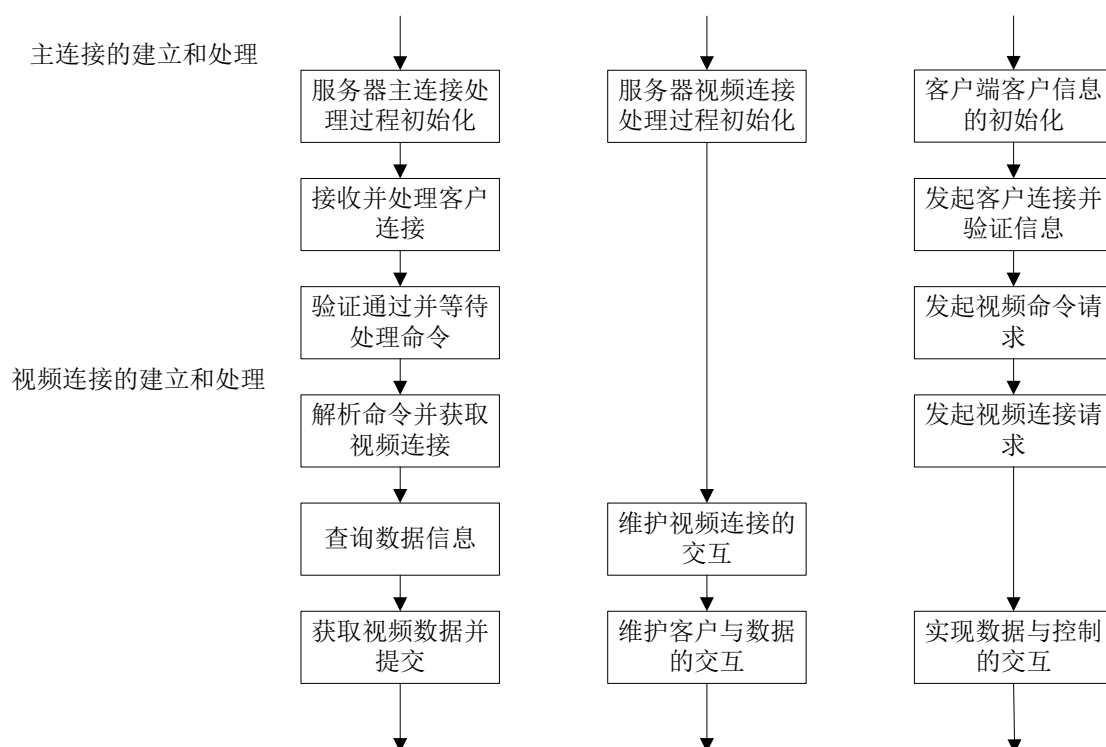


图 4-6 实时播放流程设计图

主连接的建立和处理：主连接部分是服务器与客户之间的基础连接通道，又称之为客户连接，在主连接中系统分为两部分的工作，服务器和客户端。在服务器端，随着系统的启动，服务器会进行主连接的初始化工作，当初始化处理过程完成后，系统才会对客户的连接请求实现监听，此时客户的连接请求会被系统所捕获，接收到的客户连接请求将会在系统中进行信息的验证以确保信息的有效性和客户连接的合法性。之后系统和客户将会进行有效信息的交互，当工作完成，此时主连接的建立过程将完成，系统会等待接收服务器与客户之间的交互，来实现系统和客户之间的会话。

客户连接的建立过程：这是客户与服务器真正的实现具体功能的部分。服务

器依然需要实现对客户视频连接请求的监听和处理。当客户通过主连接发送来的命令，经过解析后就进入了客户连接的建立和处理过程，服务器根据解析后的命令信息将会完成两部分的工作，一部分是对视频数据的获取，需要根据命令信息的参数分析查找到所需要的视频数据源；另一部分将会对获取到的视频连接进行验证和确认。在两部分工作都完成之后，服务器将会将两部分关联在一起，这样对于客户所需要的视频数据将发送到客户端，同时客户的控制命令也会被服务器所响应。

4.3.2 主连接的建立和处理

在主连接中，服务器的过程分为：服务器主连接处理过程的初始化，接收并处理客户连接和验证通过等待处理命令。

4.3.2.1 服务器主连接处理过程的初始化

随着服务器的启动，服务器的主连接处理过程将会初始化。初始化的主要过程如图 4-7 所示：

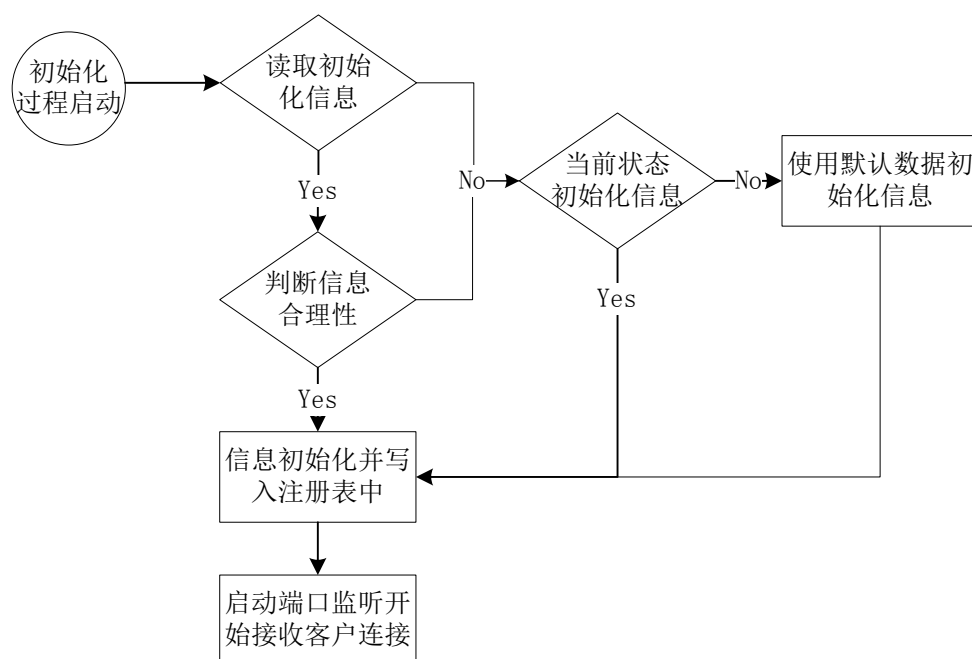


图 4-7 主连接处理过程的初始化图

在系统中，服务器的配置信息存储在了注册表中，这样对于注册表的数据，服务器能够不依赖于其他信息，仅通过本身系统所提供的接口可以方便的实现存储和修改，同时也避免了以文件形式存储信息而导致的误删情况的发生。所以当

服务器启动后，在初始化过程中，先对注册表中的信息查询，如果存在有效信息，那么会把这些信息读取出来。由于一般的对于服务器而言都有多个网卡，而在之前也分析了，服务器中的每个网卡都有不同的作用，因此，需要判断得到的初始化信息是否符合系统的状态，是否所对应的网卡满足需求。否则两者有任意的一步不成立，那么系统将会从服务器的状态中获取当前的设备信息，并以此作为初始化的信息数据。这样，获得的初始化数据将会初始化系统，然后把它们写入注册表中，以后可以直接从注册表中获得。主连接初始化之后，服务器将进入下一步过程，接收并处理客户连接。

4.3.2.2 接收并处理客户连接的过程

当用户发起主连接请求后，服务器端的监听将会响应，进入接收并处理客户连接的过程，如图 4-8 所示：

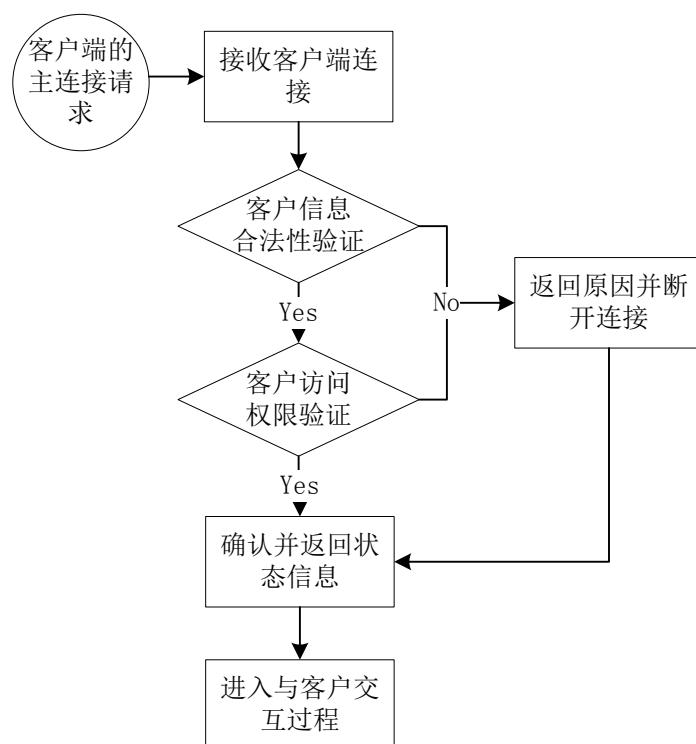


图 4-8 接收并处理客户连接的过程

当用户向主连接监听端口发起建立请求时，服务器会触发接收主连接的处理过程。从监听端口获得的主连接会话中获取的用户信息，将会被服务器用来与经过数据库查询或远程服务器访问得到的用户数据进行比对来验证客户的合法性。当用户的合法性检查通过后，则会根据用户访问的来源数据信息来确定用户是否符合白名单的要求具有对该服务器的访问权限。如果验证过程也通过，那么服务

器会给客户返回确认信息，并将客户的权限信息和可访问内容数据信息返回给客户，这样客户端收到确认后，通过权限信息决定用户所能使用的功能，通过可访问内容数据信息决定实时播放所能访问的视频通道。反之，当两者任意一个无法得到验证时，会将原因返回给用户，并结束此次连接，释放处理过程所申请的相关资源。

当服务器对客户进行了检查和判定过程后，将等待客户端用户命令的发起，此时服务器已确保了与客户端之间连接的建立，实时播放将进入视频连接的建立和处理过程。

4.3.3 视频连接的建立和处理

视频连接的建立和处理过程是实现服务器与客户端之间视频数据的传输和命令信息的送达的基本。当用户发起视频请求后，经过服务器的处理，之后两者之间的交互工作将交由此连接完成。

视频连接的建立与主连接相似，服务器在启动时，也会调用视频连接的初始化工作，服务器将会从注册表中获得连接的初始化信息，创建监听等待客户的连接。当客户端需要创建视频连接请求时，将通过主连接将视频请求信息送往服务器端，同时向服务器的视频连接监听端口发起连接请求，等待连接的建立成功。

4.3.3.1 视频连接的建立过程

当服务器视频连接监听端口收到视频连接请求时，将会调用视频连接的处理程序，对视频连接进行分析和处理，并获得视频连接的信息数据。

如图 4-9 所示，服务器通过获得的客户视频连接接收来自客户对视频连接的描述信息，其中包括视频连接的唯一描述信息，IP 地址和请求时间等。为了保证视频连接描述的唯一性需要从两方面来保证。一对于服务器而言，视频连接发起的客户需要区分，这样才能和发起请求的客户关联在一起；另一方面是对于每个客户需要区分发起请求的视频通道，这样才能同具体的视频通道数据相关联。通过这两方面的确定，保证了视频连接描述信息的唯一性。由此可以创建连接的描述信息，并将其按照时间顺序加入到描述信息队列中。除此之外，还要对描述队列进行检索，因为一旦队列中存在超时连接是就需要将其从队列中清除，同时如果队列数据达到上限，也需要告知系统此时的服务器的处理能力已不足以接收新的连接请求，停止对新连接的接收。

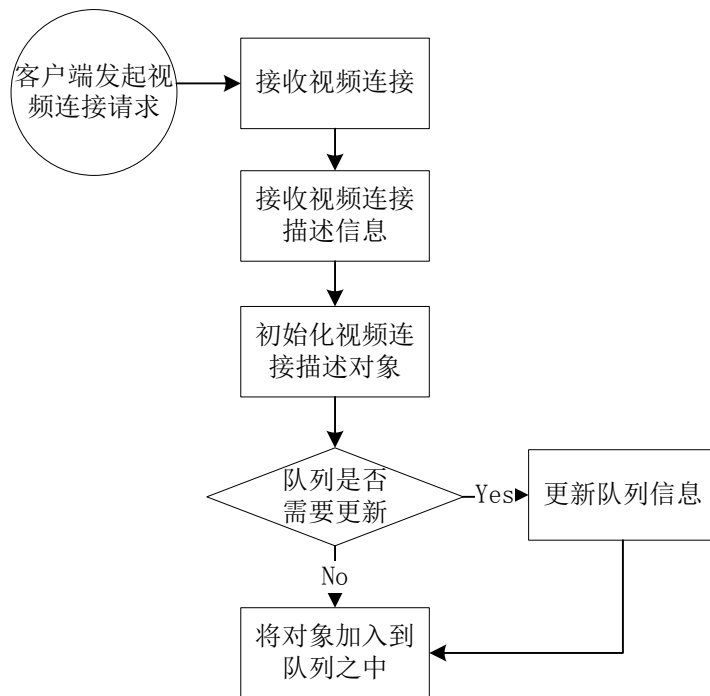


图 4-9 视频连接的建立过程

4.3.3.2 用户命令的处理过程

服务器收到来自客户的视频请求命令后，命令解析过程会将命令分解成统一的命令参数格式，通过对每一部分的分析，主程序将会调用相应过程来完成功能的实现。实时播放的命令解析后将会告知系统实现两方面的工作：数据信息的查询和视频连接的查询。

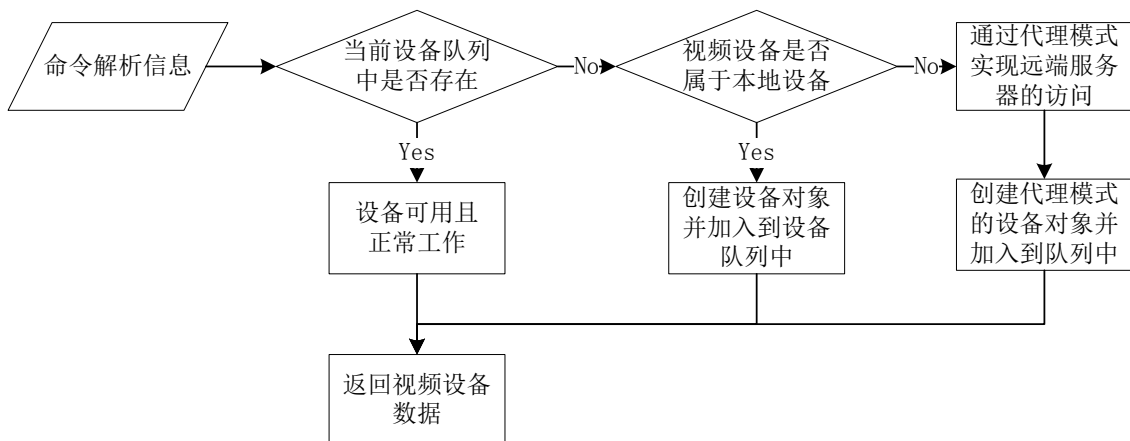


图 4-10 视频设备信息的查询过程

视频连接的查询工作，是对在视频连接描述队列中，将视频命令所对应的连接根据唯一描述信息查询出来，并创建用户的交互对象等待与视频数据相关联的

过程。将每个视频用户交互对象与视频数据关联在一起，这样就可以实现数据的分发，增加数据的利用率，减小带宽消耗。

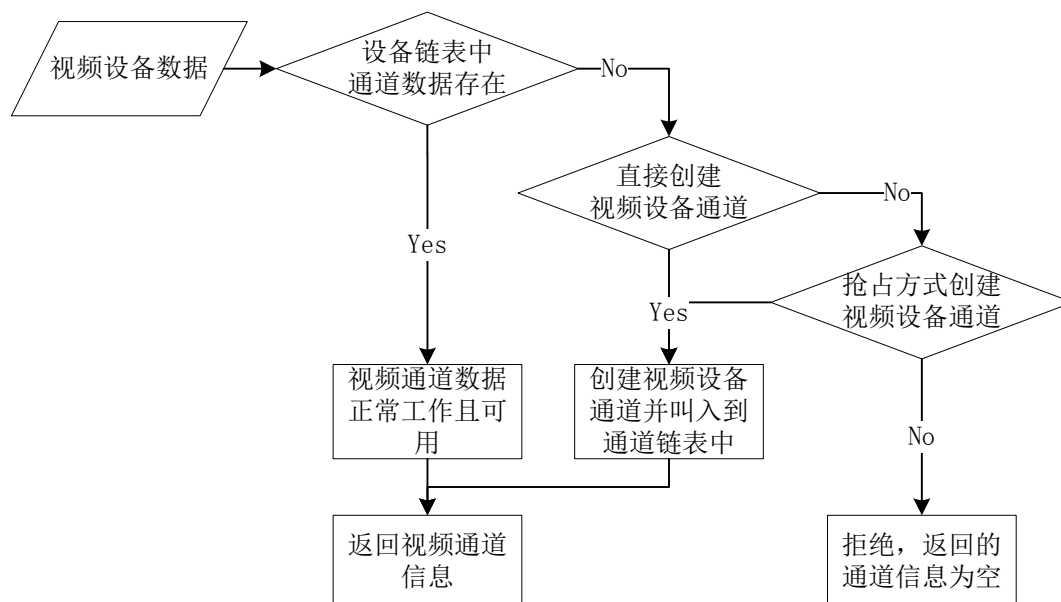


图 4-11 视频通道信息的查询过程

视频数据信息的查询过程，包含了两部分的实现，视频设备的查询和视频通道的查询，如图 4-10 和图 4-11 所示。具体的详细过程将会在 4.3.3.3 小节和 4.3.3.4 小节中详细的叙述。

4.3.3.3 视频设备信息的查询过程

在视频设备信息查询的过程之中，服务器收到的命令经过解析处理后，会根据统一的格式将其划分成命令处理所识别的命令参数，其中对于设备的描述参数将会得到。在服务器中，存在视频设备信息队列，在其中保存的是系统当前所连接上的视频设备，而这些设备可以是本地实际的物理视频设备，也可以是通过“代理”模式所实现的虚拟视频设备。根据设备信息参数，在视频设备信息队列中进行查找工作，如果队列中存在视频设备信息，对设备信息所对应的设备进行测试，判断其可用后将返回视频设备数据，否则，在队列中如果不存在视频设备，那么会进行两种情况的处理：

1) 视频设备是本地设备，但是在系统的初始化过程中并没有创建这个设备。由于系统需要支持设备的动态创建，那么当在系统启动后，有新设备添加且新设备还未被系统所连接，此时系统将会根据数据库的信息获得设备的连接描述信息，创建视频设备连接，并将设备的描述信息添加到视频队列之中。

2) 视频设备不是本地设备, 需要通过远端的服务器获得, 这里提出一种解决方法, 称之为代理模式:

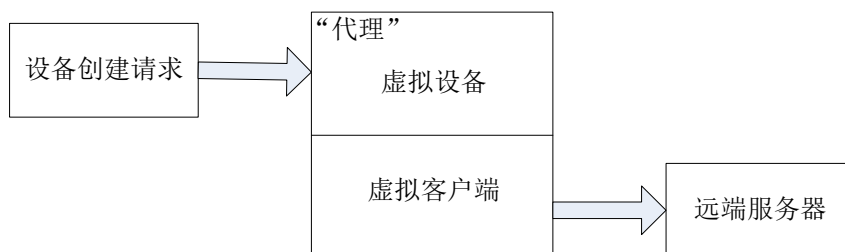


图 4-12 代理模式示意图

如图 4-12 所示, “代理”的功能是实现远端服务器的连接, 但对服务器内部的请求而言, 它表现为设备的形式。由于其不具备真实设备的特征, 因此将其定义为虚拟设备, 而在虚拟设备的内部实现中, 具体的功能则是由被称为虚拟客户端的部分来完成的。一方面实现对远端服务器的连接, 另一方面就是实现作为设备所需要实现的接口功能, 来为设备调用时提供统一的对外访问接口。

不管是那种方式, 系统对于设备都是采用统一的处理方式, 将其加入到设备信息队列中, 以便以后查询的使用。得到的视频设备信息, 将会返回给查询过程, 查询过程根据设备的信息进行视频通道信息的查询过程。

4.3.3.4 视频通道信息的查询过程

在视频通道信息的查询过程中, 由于每个设备对象内部都会维护视频通道信息链表, 当获得设备信息之后, 查询过程将会根据通道信息参数来查询其中是否存在符合条件的视频通道信息。

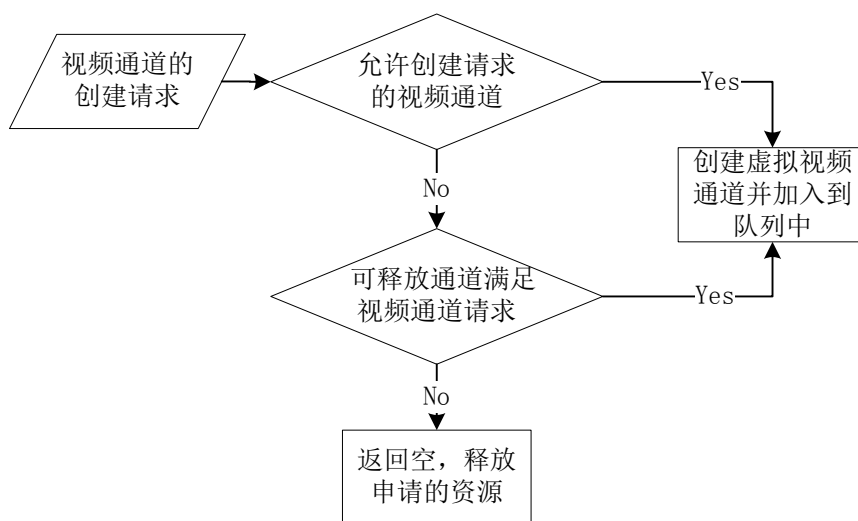


图 4-13 虚拟通道的创建和处理

当找到信息时，系统会返回视频通道信息，而这就是用来关联视频数据和用户请求的关键。如果在链表中不存在需要的信息，那么就需要再进行创建的工作，一般对于本地设备如果不存在信息时，那么将会有单独的处理过程，设备会维护自身的设备通道，因此这里对这种情况不作处理。而当设备是虚拟设备，即进行的是远端视频访问时，其查询和处理的详细过程如图 4-13 所示。

在需求分析中特别强调的一点是带宽处理，来保证视频在播放时的稳定，而在实际的设计中，将每个通道量化为标准带宽的对象，在需求分析中已经叙述过，将一路 CIF 质量的视频量化为一路标准的带宽对象，量化值是 1。对于一路 2M 的视频数据链路，它所提供的带宽量化值为 4，因此，对于虚拟设备而言，它作为虚拟客户去连接的远端服务器之间的链路带宽的量化值为 4。带宽控制的实现，就是来控制当它所创建的虚拟视频通道的带宽量化值之和不允许超过 4。因此，通过计算设备的最大量化带宽 F_{\max} 与当前设备的实际占用的量化带宽 F_{current} 之间的差值，记为 F_{free} ，表示当前的空闲带宽，是否能够满足新的请求视频通道量化带宽 F_{need} 的需求。见式 (4-1)。

$$F_{\max} - F_{\text{current}} = F_{\text{free}} \leq F_{\text{need}} \quad \text{公式 (4-1)}$$

如果满足，那么则可以创建新的通道并加入到通道信息链表之中，否则将进行优先级的判断。

由于每个客户都拥有优先级权限，所以对于每一个视频通道，它所用于判定时需要的权限等级理所应当的由访问此通道用户的优先级权限来决定。因此，每个视频通道的权限取决于其上用户的最高优先级权限，见式 (4-2)。

$$L_{\text{channel}} = \text{Max}(L_{\text{user}}) \quad \text{公式 (4-2)}$$

在虚拟设备的空闲带宽不足以满足要求时，就需要判断可释放带宽是否足够满足系统的请求，见式 (4-3)。

$$F_{\text{free}} + F_{\text{release}} \geq F_{\text{need}} \quad \text{公式 (4-3)}$$

其中，可释放带宽 F_{release} 是计算所有设备上的权限低于请求通道权限的虚拟通道的量化带宽之和，见式 (4-4)。

$$F_{\text{release}} = \sum f_{\text{release}} (L_{\text{channel}} < L_{\text{need}}) \quad \text{公式 (4-4)}$$

那么将这些虚拟通道和其上访问的客户所占用的资源释放，然后创建新的虚拟通道，并将其加入到链表之中，以便以后的查询和处理工作。这样就得到了用户所需要的视频通道信息。如果此时可释放的带宽依旧无法满足系统的要求，将不进行资源的释放，会返回特殊的视频通道信息，告知请求的客户，带宽资源已经被占满，无法接受新的请求。

至此，将同客户的视频数据的连接请求加入到所得到的视频通道信息对应的视频通道数据的用户链表中，就可以实现视频数据对用户的分发工作。

4.4 视频回放的设计与实现

4.4.1 视频回放的流程设计与实现

视频回放的实现过程如图 4-14 所示，可见整个流程与实时播放基本一样，也是分为主连接和视频连接的建立和处理过程。由于客户要进行视频的请求必然要实现主连接的建立，所以此过程是共同的，在此不再赘述。而对于视频连接的建立过程，由于视频回放现阶段还无法实现视频数据的分发和共享，所以在虚拟通道权限判断的过程当中将只有一个用户的访问。而整个过程中最大的不同点，就是增加了获取视频目录文件信息和获取视频数据文件描述信息两部分。

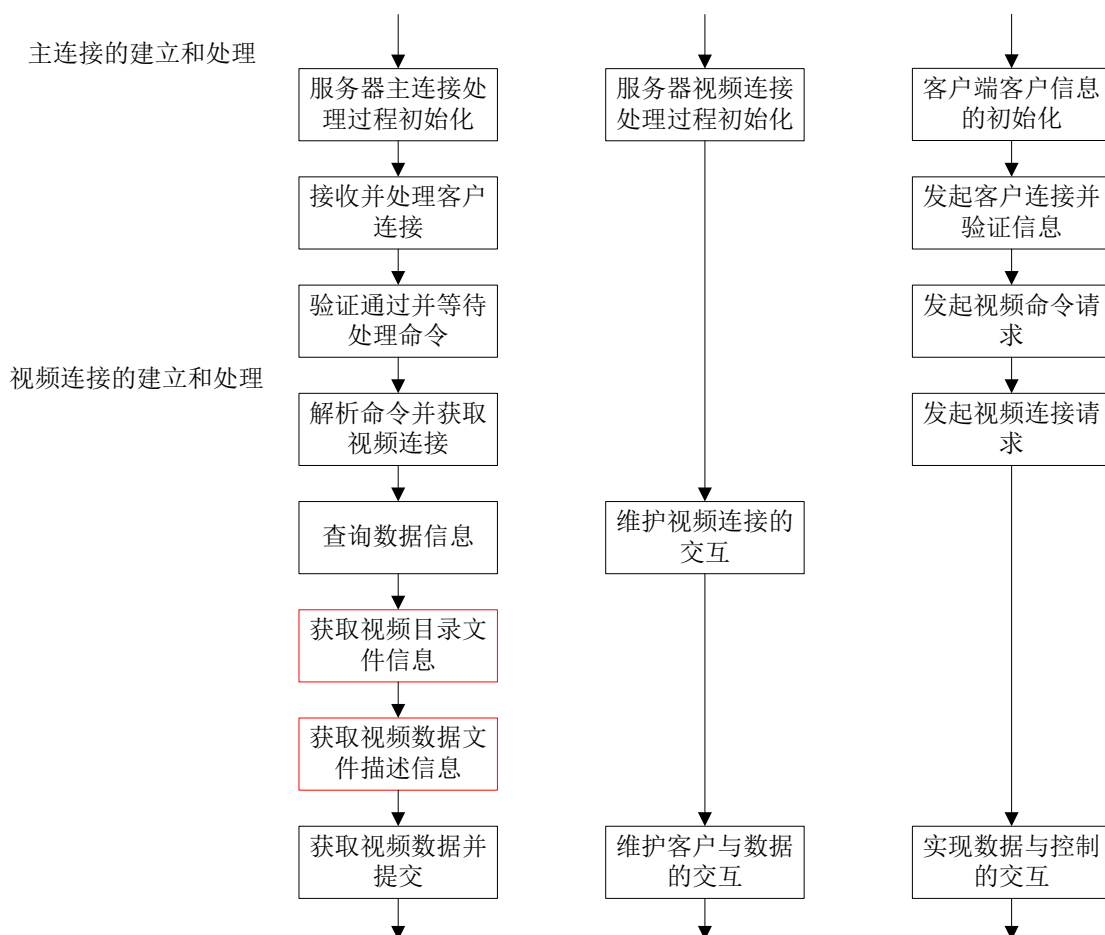


图 4-14 视频回放流程图

4.4.1.1 获取视频目录文件信息

视频文件在本地按照设备 IP 地址，以目录的区分方式存储在本地硬盘上。当用户发起视频回放请求时，就需要获取请求的视频通道的目录信息。由于存储文件按照时间顺序记录的，那么就需要将这些信息按照时间的顺序发送给客户端，以方便客户端对文件的选取，确定所要回放的视频文件是哪一个。只有选择了需要回放的视频文件，系统才能实现视频数据文件描述信息的获取和视频数据的读取。

4.4.1.2 获取视频文件描述信息

视频描述信息文件是实现视频回放控制的关键，它记录了视频文件数据中的视频帧的地址信息，以及视频帧画面的分析数据信息，这些数据保证了视频数据帧的随机访问，同时还将提高客户端回放的友好性。

4.4.1.3 视频回放时的控制和处理

由客户交互模块得到的视频回放控制命令传递给数据处理模块实现，在数据处理模块中有单独的线程来实现视频数据文件的读取和控制工作。当收到暂停命令后，线程将被挂起，视频数据读取工作暂停，将不再调用交互模块的发送过程将数据发送给客户端，这样此时就实现了视频的暂停。需要恢复时，只需要重新唤起视频文件处理线程即可。

加速和减速的实现是通过控制线程循环时的间隔周期，同时对于间隔时间的动态控制也是用于对于带宽控制的实现。当加速播放时，随着速度的增加需要对视频帧的读取增加间隔数量以变相的提高视频画面的播放速度，为了保证视频画面的稳定，I 帧将不会被跳过，那么随着间隔帧数量的增加，将会导致 I 所占的比例越来越大，这就会使得网络视频的传输带宽增加，因此此时需要增加循环的间隔周期使得单位时间内发送的视频数据降低，稳定系统网络上的带宽传输。

当减速时，仅需要增加间隔周期，那么发送到客户端的数据频率也会随之降低，此时在客户端播放时所看到的画面就会有减速的效果。

4.5 客户交互模块的实现

4.3 节和 4.4 节中已经通过对于实时播放和视频回放流程设计的分析，详细的描述了系统在实现实时播放和视频回放功能时的实现过程。因此，系统客户交互

模块的具体实现结构如图 4-15 所示：



图 4-15 客户交互模块具体结构图

4.5.1 连接的初始化与处理

连接的初始化和处理模块主要包含客户主连接和客户视频连接的初始化和处理过程，具体函数实现基于 4.3.2.1 节和 4.3.3.1 节对两者的分析和设计工作。当系统启动后，创建此功能模块的实例，来启动客户连接和视频连接的监听，并对接收到的连接请求实现处理，完成基本的功能操作。

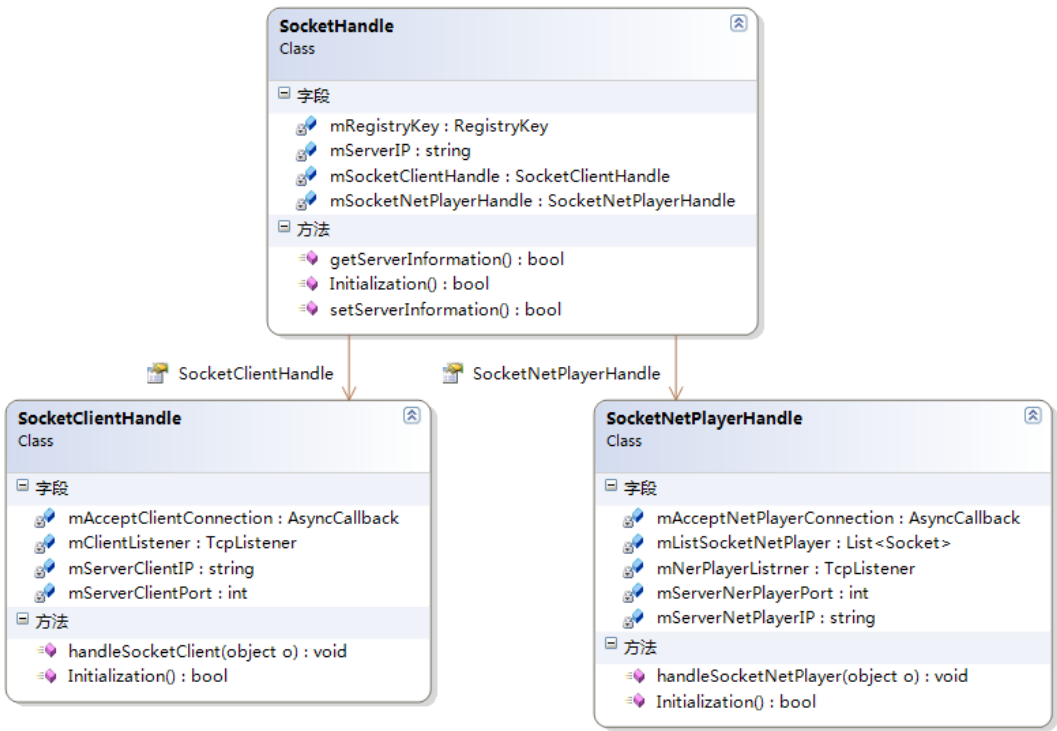


图 4-16 连接初始化与处理类图

如图 4-16 所示，是模块实现的类图，SocketHandle 是系统用于初始化连接操作的类，mSocketClient 和 mSocketNetplayer 分别是 SocketClient 和 SocketNetPlayer 的实例，分别用于完成主连接和视频连接的建立和处理工作。在 SocketHandle 类中的 Initialization 接口将会调用 mSocketClient 和 mSocketNetplayer 的 Initialization 方法实现客户连接和视频连接的初始化和监听工作。当监听启动后，收到的连接请求将会激活各自的回调函数 handleSocketClient 和 handleSocketNetPlayer 来实现对于连接的处理工作。

4.5.2 客户连接交互模块

客户连接交互模块，是服务器实现与客户端用户连接建立和维护功能的主体，在其中将主连接的主要过程分为三个子模块来实现，如图 4-17 所示：

ClientConnection 是实现客户连接维护和保障功能的，其中 heartBeat 函数定时的向客户端发送心跳信息，来实时确定连接的存在性，一旦当发现客户连接断了，那么系统将进入废弃状态，此时将保留一段时间的用户信息。如果客户短时间内再发起了新的连接，那么可以直接通过此来得到用户的数据信息而无需重新申请，这样减少资源的消耗并减少了系统对于客户的响应时间。在子模块中还提供了与客户端的交互接口 Sent 和 Receive，将会对数据进行加解密操作来保证数据在网络上传输的安全性，这样保证了用户的个人信息。

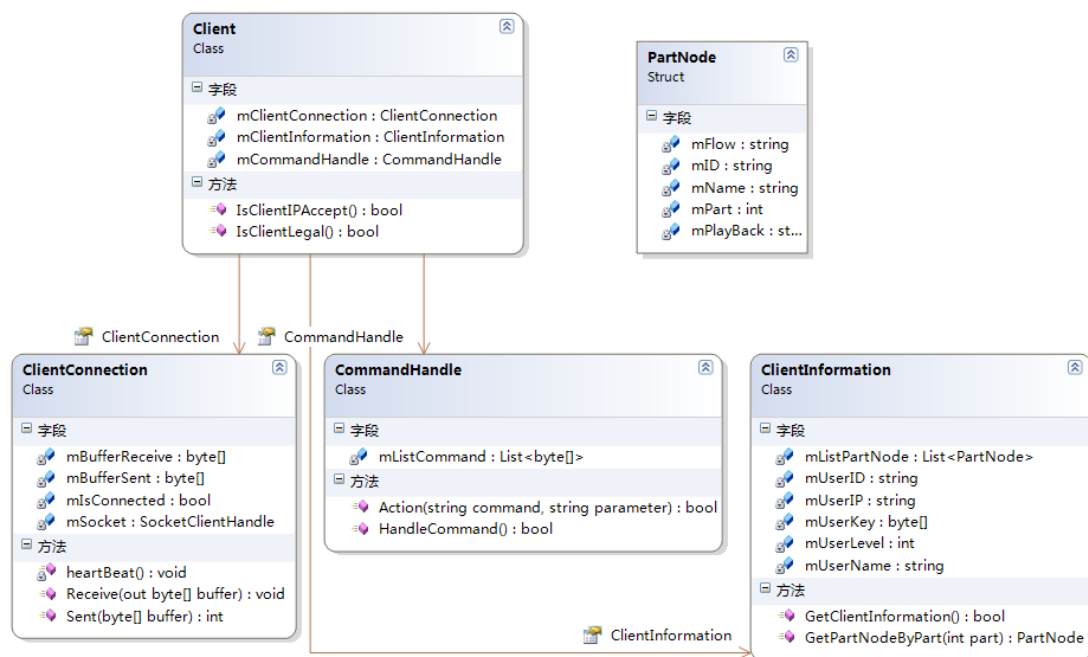


图 4-17 客户连接交互模块类图

CommandHandle 是对命令的处理模块，当从客户端发送的命令被服务器接收后，需经过 utf-8 的编码处理转换成标准的字符串编码命令格式，按照统一的命令格式划分后执行 Action 函数实现具体命令的处理过程。

ClientInformation 是用户信息的描述，其中除了包含用户的基本信息外，还含有用户权限和部门下所能够访问的视频通道，一方面为了保证客户部门信息的及时更新，在部门的设备信息发生变化时，所对应的 ClientInformation 对象的 GetClientInformation 方法会被调用，因此保证了对于每个客户而言部门视频设备信息的及时性，而用户的访问可以根据对应的部门通过 GetPartNodeByPart 方法来获得视频设备信息。

4.5.3 信息查询和视频连接交互模块

视频连接根据设计的叙述，需要信息查询模块来实现视频通道，视频设备和视频连接的查询，而对于实时播放和视频回放，对于查询的过程都有自己不同的处理，因此要实现统一的接口就要通过集成来实现多态。在系统的关联模块中，将视频通道和客户连接结合在一起，但对实时和回放来说，存在不同的处理过程，由于现阶段的视频回放还无法实现视频的分发功能，而且对于两者控制功能的实现也是有各自的特点。例如在实时播放中，控制的对象是云台功能，是对视频摄像头的控制；而视频回放时是对视频文件读取的控制，因此对于不同的控制对象在实际的实现过程之中就有其各自的处理特点。

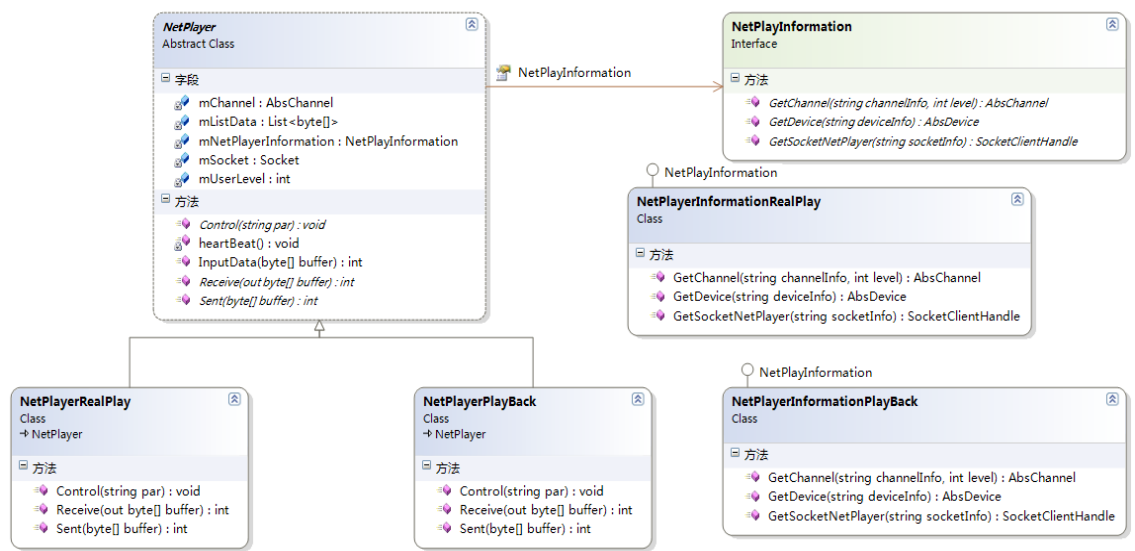


图 4-18 信息查询与视频连接交互模块设计图

还有一点需要注意的是，对于视频数据的发送，实时播放和视频回放也存在着区别，由于实时播放的视频需要保证视频的实时性和流畅性，因此对于发送的数据需要有条件的调整；而在视频回放时，数据的发送需要考虑的是在用户发起速度控制时所带来的码流的变化，而这种变化是需要通过调整来保证网络带宽的稳定的。因此对于两者的实现也需要具体的区别化实现过程，因此实现如图 4-18 所示：

在图中 NetPlayer 是视频数据和用户的关联类，其中 mSocket 是用户对应的视频连接，mChannel 是对应的视频通道数据对象，NetPlayerRealPlay 和 NetPlayerPlayBack 继承了 NetPlayer，并针对实时播放功能和流程的特点和视频回放的性质分别实现了父类中的抽象方法。这样，当外部通过 NetPlayer 的引用调用抽象方法时，无需了解具体的对象实例的性质，对于外部来讲是透明的，这也降低了对具体实际功能实现时的耦合性。另外，信息的查找过程也是与具体的实现能够相关联的，这在之前的设计中已经详细的叙述了，因此为了将这种“绑架”去除，对于不同的查询过程实现相同的查询接口，在 NetPlayer 中添加对接口的引用，当子类创建时同样的初始化对应的子类查询方法。通过这样的设计查询过程不仅将不再依赖于具体的功能实现，而且还具有很好的扩展性，当系统添加新的功能时，只需要继承 NetPlayer 实现新的关联类同时实现 NetPlayerInformation 接口就可以实现新的客户功能的实现。

4.6 数据处理模块的实现

设备与通道连接处理模块是整个系统数据来源的根本，根据之前详尽的需求分析，连接过程分为设备的连接和处理以及通道的连接和处理，此外为了减少操作和判断的复杂性，不使用分级分层的权限判断和控制，而是设计在具体的设备和通道所在端实现。由于视频的控制必然的需要对具体设备发起控制命令，那么就需要保证与设备的连接状态，因此对于控制权限的实现也需要在具体的模块内实现。

4.6.1 设备与通道的连接和处理

在系统启动运行时，调用设备连接模块实现设备的连接，每个设备再连通它所拥有的视频通道，创建的通道会根据关联性统一的由控制权限模块来实现对通道的云台控制。如图 4-19 所示：

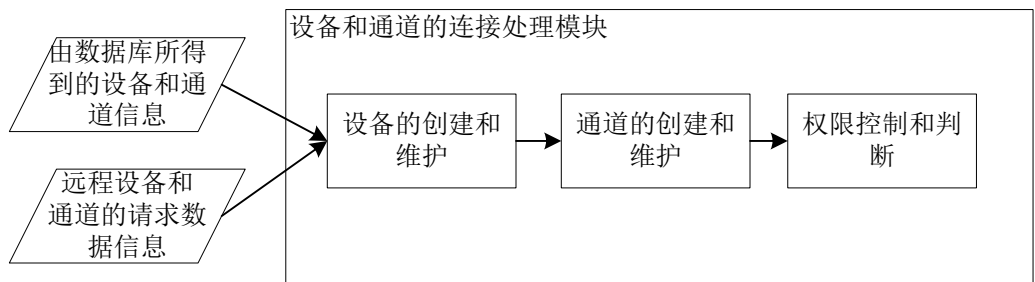


图 4-19 设备和通道的连接处理

由数据库所得到的设备信息，这些信息包括设备的类型，设备的 IP，设备的端口，设备的登录名和密码等，以及设备所包含的通道信息，通过这些来完成设备的和通道的创建过程。

4. 6. 2 设备连接和处理的设计与实现

4. 6. 2. 1 设备连接和处理的流程设计

设备的连接是实现视频监控数据获取的基础，通过 SDK 的调用，实现服务器与设备会话的建立，通过会话实现其他的请求操作，设备连接和处理的过程如图 4-20 所示。

由于连接设备使用的是厂商所提供的动态库，所以在进行设备连接之前需要对设备的 SDK 动态库进行初始化操作。由于每种设备所使用的动态库不同，为了节约系统的资源在运行时按需要的进行动态库的加载工作，同时又由于动态库的初始化工作不能重复进行。因此在系统中对每种动态库设计了标志位，当标志位被设置时表明此种动态库无需再进行初始化工作，当设备初始化时只需要查看标志位是否已经设置来决定是否进行初始化工作即可。

当设备对象创建时，通过数据库所获得的用于设备连接的信息和设备需要初始化的信息都会保存在对象的实例中。当设备出现意外中断时，由无法通过动态库本身的回调函数来处理时，就需要对设备进行重新连接工作，而此时无需再次查询数据库，只需要从设备实例的属性中获取即可，因此节约了时间的开销，提高了系统响应的效率。

调用 SDK 实现设备的连接过程，对于不同的动态库的处理方式有所不同，可以通过函数的返回值或回调函数来获得设备连接的句柄，一般的正确的获得了设备句柄之后就可以通过此会话句柄来实现与设备的交互操作。

在设备成功的连接后，还需要对设备的参数进行初始化操作，如设备状态的重置，防止设备因处在错误状态而导致无法正常工作。特别的，还需要对设备的

工作时间与服务器的本地时间同步，这是为了保证视频的画面时间（由设备的工作时间确定）和视频文件的修改时间（由服务器的本地时间来决定）相一致。当这些工作处理完毕后就可以进行通道的创建过程。

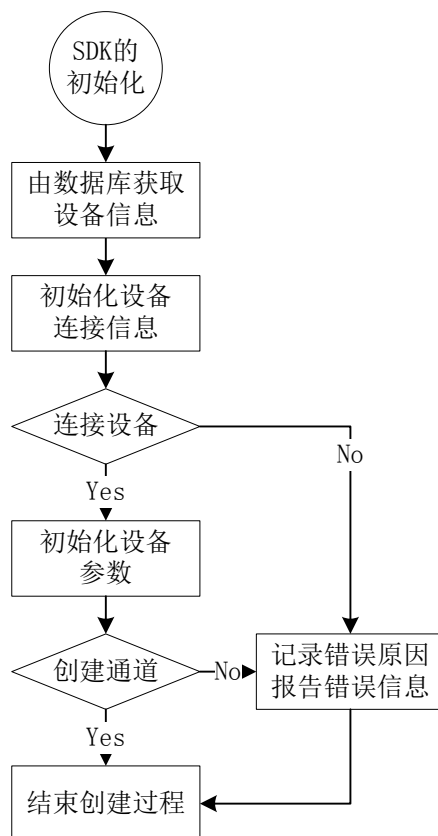


图 4-20 设备的连接和处理过程

通道的创建过程由单独的线程来处理，这样一方面保证了在设备连接会话创建后进行通道创建时不会由于等待而导致设备连接缓慢的问题，特别的对于一些大客站而言，两三百个摄像头的初始化过程如果使用顺序等待的方式将是极度耗费时间的。另外通过这种方式，设备连接与通道连接也将会更好的解耦，对于通道单独的处理也将更加灵活。具体的实现在 4.6.3 节中叙述。当通道成功创建后整个连接过程结束。

在设备的连接和处理过程中必然会出现错误的情况，当错误发生时，系统会将这些错误信息记录下来并通知管理员及时的检查，保证设备的正常工作以确定系统的稳定工作。

4.6.2.2 设备连接和处理模块实现

在设备连接时，由于设备不同所使用的动态库就有所差别，即使相同的动

态库对于不同的设备也有各自不同的处理方式，所以在设计实现时一方面要考虑到作为设备应有共同的对外提供的接口，提供统一的调用而无需关注具体的内部实现；另一方面，又必须要保证设备拥有各自的具体实现，满足自身连接和处理需求并且相互之间没有干扰。此外，还应有利于设备的添加和修改，所以设备的结构如图 4-21 所示。

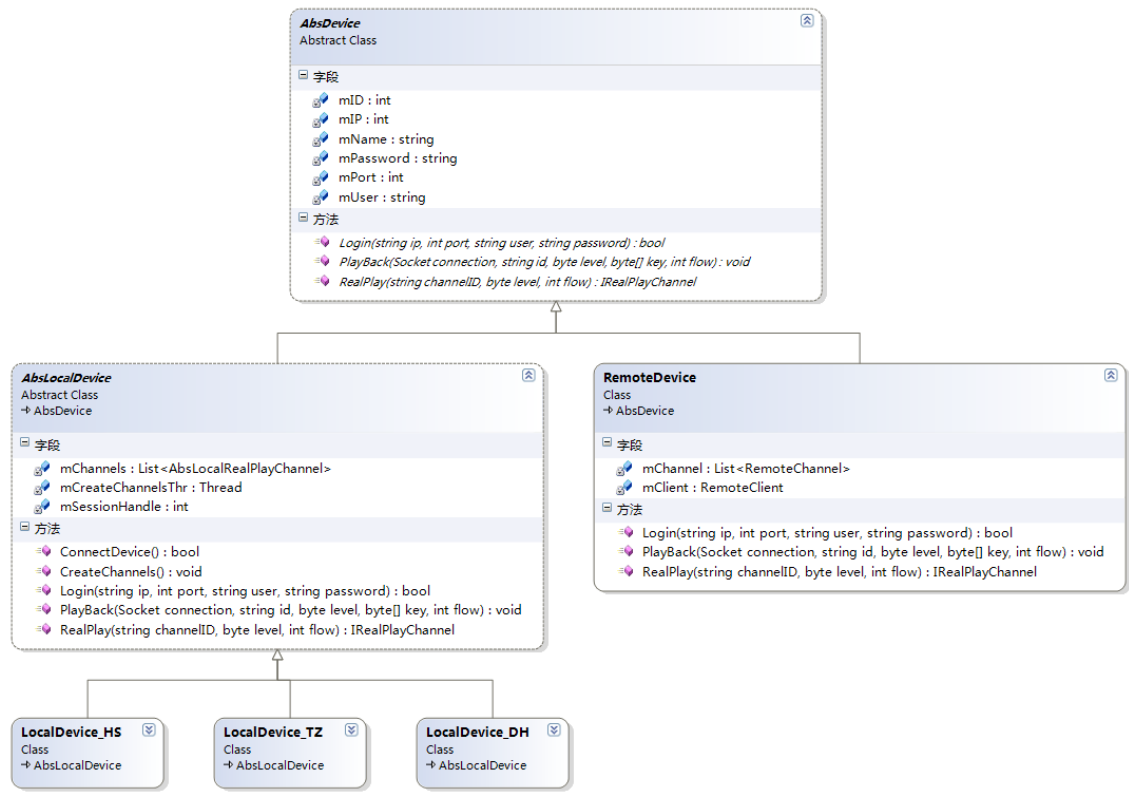


图 4-21 设备连接和处理类实现图

由于设备对象实例需要记录设备的 IP，登录名等信息，所以将 AbsDevice 以抽象类的形式定义而非接口的形式，以便通过属性可以直接访问设备的描述信息，在 AbsDevice 中定义了三个抽象方法 Login、RealPlay 和 PlayBack，其中 Login 用于设备的连接，RealPlay 和 PlayBack 分别用于获得所需要的实时播放和视频回放所需要的数据信息。

在之前分析过，在系统中存在本地设备和虚拟远端设备，两者既有区别，又存在着联系。两种设备都需要作为设备拥有统一的对外接口，所以都将继承 AbsDevice 抽象父类。但在具体的实现时还是有很大的区别的。

在需求分析中提到，由于设备种类的多样性，导致动态库的多样，并且每种设备的连接方式都有各自不同的特点，因此将这种变化和不确定性封装起来，通

过子类的不同实现来处理这种不同。系统利用同一的构造方法，根据参数的不同创建不同的子类实例，而在系统运行时只需要调用统一的接口，系统会根据具体的子类实现来明确的实现不同设备的连接和处理过程。因此 `AbsLocalDevice` 继承了 `AbsDevice` 并实现了与具体子类无关的 `Login`、`RealPlay` 和 `PlayBack` 方法，并定义了 `ConnectDevice` 和 `CreateChannels` 抽象方法需要具体的子类根据设备的特性来实现。这样对于本地设备而言，当新种类的设备加入时，只需要继承 `AbsLocalDevice` 类实现抽象方法而对其他的具体子类没有任何的修改和影响就可以实现系统的要求，因此降低了结构之间的耦合性，同时提高了系统的扩展性。

如之前设计中分析，虚拟远端设备类 `RemoteDevice` 通过虚拟用户来实现了抽象父类 `AbsDevice` 的三个抽象方法，虚拟用户的 `RemoteClient` 的实现类图如图 4-22 所示。

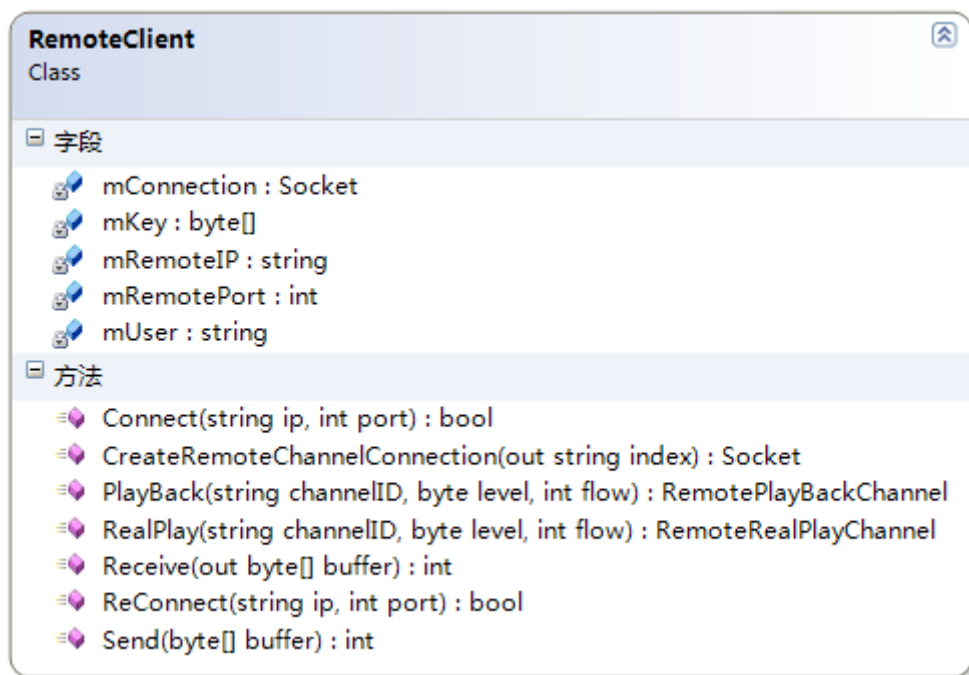


图 4-22 RemoteClient 类结构图

在 `RemoteDevice` 的 `Login` 方法实现中，通过对 `RemoteClient` 实例调用 `Connect` 方法实现以特殊的虚拟用户对远程服务器的连接，如果连接成功将返回，否则将调用 `ReConnect` 方法实现远端服务器的重连。

在远程虚拟设备的实现中，由于所连接的是远程服务器，因此在实现 `RealPlay` 和 `PlayBack` 获得所需要的虚拟通道信息时，就需要进行带宽管理的判断。实现过程如 4.3.3.4 小节中所叙述的流程。而所需要的虚拟通道信息则由 `RemoteClient` 实

例对象的 RealPlay 和 PlayBack 方法分别得到。这样与远程服务器的连接和交互工作都由 RemoteClient 类来完成，RemoteDevice 类只需要从其获得所需要的数据和信息即可，它对外表现为设备的属性。这样对远端的访问也同样的统一为设备来处理，将远程服务器抽象成设备资源，通过虚拟用户实现“代理”模式的功能，这样设备的连接过程整体都将是统一化处理了，在系统运行的时候设备类的会实现动态绑定。

4.6.3 通道连接和处理的设计与实现

4.6.3.1 通道连接和处理的流程设计

设备的连接是为了实现数据的获取和云台控制，因此需要对每个通道建立连接来实现功能，如图 4-23 所示，是通道连接和处理的流程图。

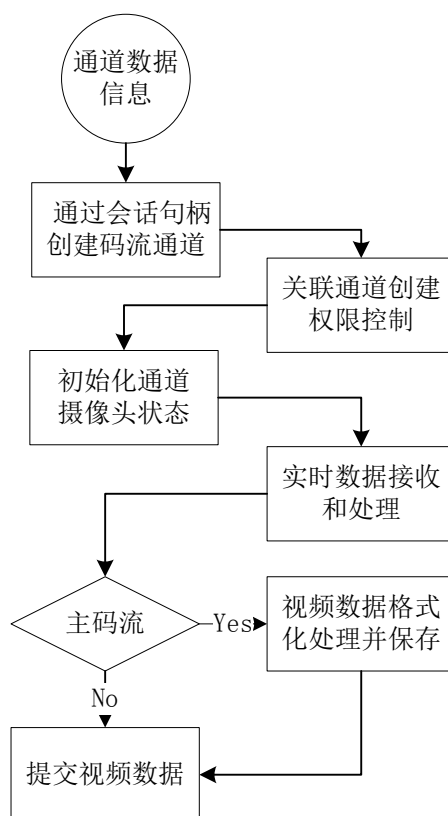


图 4-23 通道连接和处理流程图

由数据库得到的通道信息通过设备类的处理过程分析后对设备的通道进行创建。实时数据是视频监控系统的核心，把每个摄像头称之为一路视频，例如，DVR 可以连接四个摄像头，那么它就有四路视频；IPC 枪机只有自己，那么它就只有一

路视频。每一路的视频，根据画面质量和来源，又可以分为来自于主码流的高质量视频数据和来自于辅码流的低质量视频数据，为了在逻辑上明确的可以区分两者之间的差别，将两种码流视频数据看作两路不同的通道。

所以根据需求分析的要求，通过连接设备的会话句柄，分别进行主码流和辅码流通道的创建，当成功的创建通道后，将进行数据和关系的处理工作。

两路通道从根本上说所对应的是一路视频信息，所以它们所对应的云台控制和权限管理应当是相关联的。例如，当由用户通过主码流通道对摄像设备进行云台控制请求时，对应的辅码流通道也应当收到相应的权限请求，因此需要将两者关联在一起并通过单独的权限控制实现对云台的访问和控制请求。

新建立的通道，云台之前状态是不确定性的，特别是需要对视频数据进行保存，所以摄像头所记录的视频位置至关重要，因此就需要对通道进行预置，保证视频画面的位置。

另一方面，通道的创建，根本目的还是获取实时视频数据，因此当建立视频流连接后，系统将通过回调函数，获得具体的视频数据包，再对视频数据包进行处理后就可以实现数据的发送和保存。具体的实现过程，将在回放数据处理中详细讨论。

4.6.3.2 通道连接和处理的模块实现

为了实现通道的标示意义，首先需要确定在数据库中的标示形式。

pk_channel_id	location	flow	fk_device_id	record_days	playback_flow	fk_sort_id
1-0	迎祥街信号机械室	4	1	7	4	19
1-1	迎祥街信号机械室	1	1	7	4	19
1-2	迎祥街通信机械室	4	1	7	4	18
1-3	迎祥街通信机械室	1	1	7	4	18
2-0	迎祥街内勤	4	2	7	4	2
2-1	迎祥街内勤	1	2	7	4	2
2-2	迎祥街外勤	4	2	7	4	3
2-3	迎祥街外勤	1	2	7	4	3
2-4	迎祥街重方咽喉	4	2	7	4	4
2-5	迎祥街重方咽喉	1	2	7	4	4
2-6	迎祥街成方咽喉	4	2	7	4	4
2-7	迎祥街成方咽喉	1	2	7	4	4

图 4-24 通道信息在数据库中的表示

如图 4-24 所示，每个视频数据源将看作是一条记录，其中字段 pk_channel_id 表示每一路不同的通道，而每一路通道除以 2 所得的商，即是每一路视频所对应的编号（编号从零开始），这样，数字 0 到 3 则表示了连接的设备中的视频编号。

编号的余数，即如果是偶数，则表示这路视频所对应的视频通道是高质量主码流视频数据，反之如果是奇数，那么则是低质量辅码流视频数据。

在通道的连接和处理的过程中，同样要使用不同种类的设备动态库，这就必须在实现时有所区别。对于不同的设备通道连接所使用的设备连接会话句柄的数量也是不同的，所以，在设计实现时依然如设备类使用继承的方式通过多态来实现设备的动态绑定。

所有的通道将被看做是一种 Channel 实例，都继承自 AbsChannel，由于不同的通道实现的作用不同，因此将通道设计成用于本地数据处理的 AbsLocalChannel 和用于远程数据处理的 AbsRemoteChannel。在需求分析中对于功能的分析可以发现，数据的处理主要实现两种实时播放和视频回放，因此按照功能的划分再将通道设计为实现了实时播放 IRealPlayChannel 接口的 AbsLocalRealPlayChannel 和 RemoteRealPlayChannel 以及用于视频回放的 LocalPlaybackChannel 和 RemotePlaybackChannel，如图 4-25 所示。

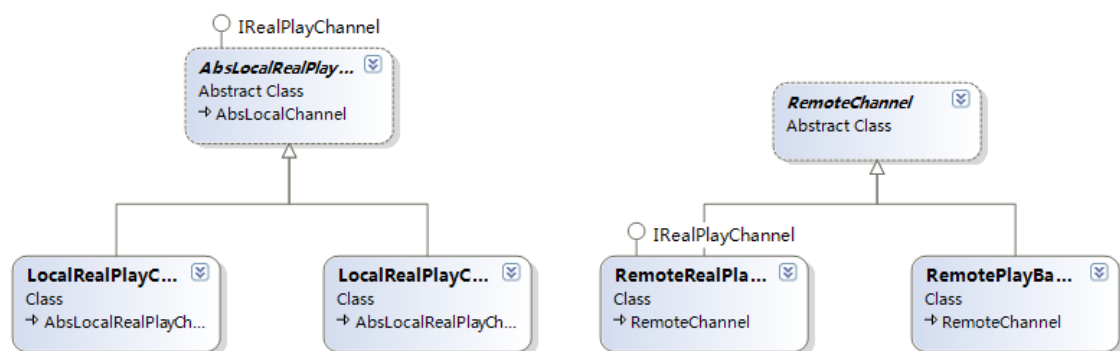


图 4-25 通道模块类设计图

对于本地通道数据，与设备类的本地设备有相似的处理，都需要对不同的设备有不同的子类实现。因此，都需要继承并实现 AbsLocalRealPlayChannel 的 CreateRealTimeStream、PTZControl 和 HandleData 方法。

CreateRealTimeStream 是用来实现与具体设备进行通道创建的方法，其实现的过程依赖于具体的设备种类，因此需要在子类中有针对性的调用。

PTZControl 是实现云台控制的方法，由于不同的设备所采用的云台控制协议不同，因此所对应的云台控制命令有很大的别，为了对外实现统一的接口调用，在系统的实现过程中，需要在具体的子类设计时完成相应的工作。

HandleData 是对获取的实时数据包的处理工作，在需求分析中已经提到过，设备的种类不同，那么它们所封装的数据格式千差万别，因此也需要针对具体的

设备来实现。

不同于设备的设计，在实现实时播放功能时，系统通过 `IRRealPlayChannel` 接口来实现对本地实时通道和远程虚拟实时通道的统一化管理，而对远程虚拟通道统一管理时则使用 `AbsRemoteChannel` 来实现。这样一方面实现了功能上的分离，降低了耦合性，另一方面也利于系统结构的扩展，增加系统的灵活性。

4.6.4 权限控制和设备维护

4.6.4.1 权限和云台控制的设计与实现

在需求分析中提到的很重要的一方面就是来自于网络带宽的限制和不同等级用户之间的权限控制，在物理上看分离的主辅两个码流通道，其在实际的应用上存在着将两者看作是一体的许多场合，特别是对于具有权限判定的云台控制。因此权限控制设计的好坏对于系统的实现影响很大。

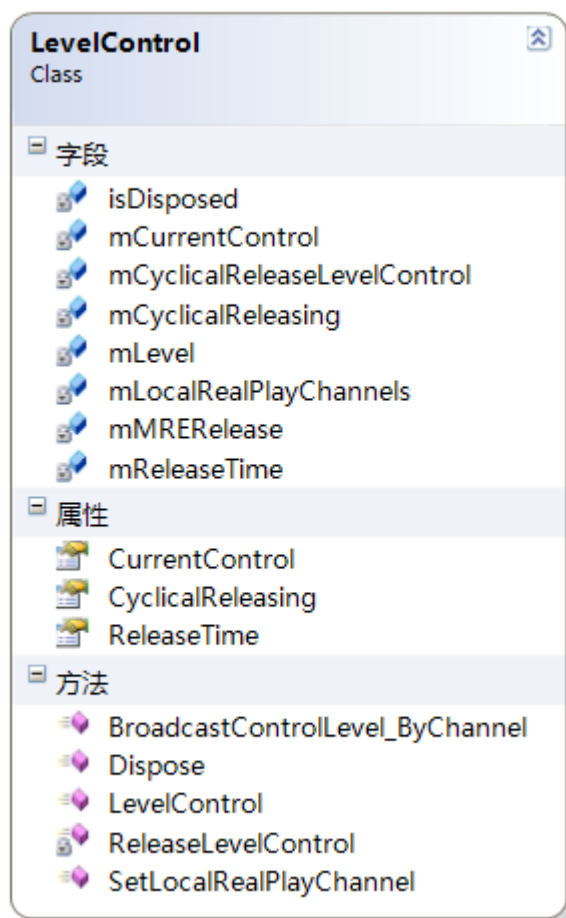


图 4-26 权限控制类图

将控制权限的判定分离出来单独处理，有利于系统的变化，云台的控制与权限的控制本身依赖于每个视频通道，但与具体所使用的动态库并无关联。因此，对于所有的通道，都应有统一的处理方式，所以将这部分功能单独的分离出来，在功能的实现上是合理的。

在 `LevelControl` 类中，将多个通道通过 `mLocalRealPlayChannels` 数组相关联，所关联的通道是本地实时通道实例。类中有且仅有唯一的获得权限控制的用户 `mCurrentControl` 和当前用户的权限 `mLevel`，以此来保证控制的唯一性，如果新的客户请求与当前的不一致，将根据用户的权限等级与当前的用户权限 `mLevel` 进行比较，如果满足要求新用户将会获得控制权限并更新控制用户 `mCurrentControl` 和用户权限 `mLevel`，否则将会拒绝请求并通知请求的用户访问受限的原因。如果客户的请求与当前的一致，那么客户的权限控制时间 `mReleaseTime` 将会被重置，以延长用户的控制的释放时间。在当前用户或者当前权限发生变化，那么将调用 `BroadcastControlLevel_ByChannel` 方法，通过对每个关联通道上的用户进行权限广播来更新权限的信息，以保证权限控制的实时性。

由于网络的延迟性，客户端与服务器端的数据交互必然存在的很多不确定性，因此减少造成干扰的因素和简化处理的逻辑，对于用户的权限判断只在本地实时通道实例端进行。但这样的设计方式也带来了另外一个问题，一旦某个用户因某些意外原因，错误地一直占用着通道的控制权限导致其他低权限客户无法抢占，那么就会使得系统由于权限控制问题将无法使用云台控制造成系统运行的障碍。所以，在客户每进行一次控制操作请求时，都会将释放时间 `mReleaseTime` 重置并触发权限控制机制 `ReleaseLevelControl` 函数运行，当规定的时间到达（不同的用户的控制时限与其权限相关），系统将自动释放当前用户，并归零控制权限。通过这种方式，保证了系统的权限控制的稳定。

在 `ReleaseLevelControl` 函数中以独立线程循环的方式实现，`mCyclicalReleasing` 用来控制循环的结束，保证在通道正常工作的时间内，权限控制可以合理运行，并在资源释放时，实现循环结束，保证线程正常的执行完毕。信号量 `mMRERelease` 的信号变化，使得循环的线程在需要的时候才被唤起，这样节约了 CPU 的消耗并降低了无用资源的浪费。`mReleaseTime` 是权限控制和释放的核心变量，它决定了用户所能获得的控制时间，当在超时可对资源释放。所以，当客户保持控制权限的前提下，其每次的控制命令都会保证控制权限时限的延长，而一旦规定时间内没有新的控制命令发出，即控制时限达到，那么系统将会给所关联的通道发送停止命令，同时释放控制权限，清空控制客户，并向所连接的客户广播新的可控

信号。

通过这样的设计，一方面实现了权限控制与多个通道的关联，同时降低了权限控制与通道的耦合性；另一方面，解决了高权限客户由于不合理情况的发生而长时间占用通道权限导致其他客户无法控制的问题。

4.6.4.2 设备维护的设计与实现

视频数据是系统的关键所在，那么保证视频数据的稳定获取更成为了系统设计的重中之重。而实现视频数据稳定获取的保证存在两个方面：

1) 要保证系统和设备的正常连接和通道连接的正常创建，这些信息往往能从设备驱动所使用的动态库的返回函数中直接获取，但也存在着一些特殊情况，例如，在大量的测试和长期的实验过程中发现，有些设备无法正确的返回一些关键信息，当动态库驱动多台设备时，或者设备突然断电时，这样的情况尤为明显。设备通道的信息是通过数据库获取的，因此对于设备本身而言无法判断通道的数量信息，在测试的过程中发现通道的数量经常由于所连接设备的增加频繁的调用动态库而导致通道数量创建不足情况的发生，但是这样的情况一旦发生，需要有效的处理方法，否则，将会对视频监控系统的运行带来灾难性的后果。

2) 视频设备的质量千差万别，无法保证每台设备在长期运行过程中不出现任何的差错，特别是这样的差错通过设备本身提供的信息根本无法得到通知。最明显的一点，在长时间的测试过程中发现，特别是在回访时发现有些摄像头的数据是没有图像的。经过查找发现，虽然系统与设备的所有连接都是正常的，但是回调函数所得到的所有视频数据包都是没有图像信息数据的，但这一切的原因却并不明确，因此只能通过外部手段来解决。由于这些不确定情况的发生，导致系统与设备的连接进行了重新的设计，那么对系统的设计提出了更大的挑战。

考虑到以上两种需要处理的情况，最好的处理而且也是唯一可行的处理就是通过销毁会话清理资源重新进行设备和通道的连接，因此，如何将两个需要考虑的方面放在一起处理同时尽量减少对原来结构设计的修改成为了实现的关键。

而这里的实现将会使用单独的线程来实时的进行检测。在循环中，检测过程先对设备的通道数量进行统计，如果创建成功的通道数量与数据库中所记录的信息不符，那么就需要对通道进行重新的创建工作，由于失败的原因并不明确，这里所采取的方法就是销毁设备资源，释放会话连接，重新进行设备和通道的连接和创建工作。当通道数量检测成功后，将会对每个通道数据进行检测，特别的在通道的正常工作的过程中，无论画面变化多小，每秒钟都会有超过 5K 的数据流

量，而测试发现在视频没有图像，出现错误的情况下，数据流量一般会降低到 2K 左右，因此对每个通道采取每十秒钟一次画面统计，如果此时通道所获得的视频数据量达不到 50K，那么就认为它发生了错误，需要对此通道进行重新创建的工作。通过这样的方式，在一定程度上保证了视频设备和视频数据的正常连接和获取，为系统稳定运行和正常的交互提供了前提基础。

4.6.5 视频回放数据处理的设计与实现

4.6.5.1 实时数据的处理与保存

根据需求分析中的叙述，设备的不同会造成数据格式的千差万别，因此需要对数据进行统一的处理工作，才能用于视频回放操作来保证在回放时高效的响应时间和更容易的读取方式。特别是对于高性能的视频监控服务器而言，硬盘的响应速度已成为系统效率提升的巨大瓶颈，更好的对视频原始数据做优化对系统效率的提升也具有明显的作用。

鉴于 H.264 编码格式文件的特性，视频将会分成两个文件存储，其一是纯粹的视频数据其中也包括对应的音频数据，另一个是对视频数据的描述信息文件。在数据存储文件中，从实时视频数据中去除了包头的描述信息剩余的将是具体的视频或者音频数据，并将视频数据的帧的开始统一为 0001 开头的信息。这样视频数据在读取时，就可以使用统一的处理过程来进行回放的工作。

此外，每个数据帧的头部需增加了八个字节，用于存储视频包的描述信息，包括视频包的长度，因为在回访时依然是按照一帧帧的进行读取的，那么必须有描述每一帧的长度信息。对于不同的设备而言，它的帧率是不同的，这意味着对于视频编码而言，它是经过多长时间采集的视频图像然后将之编码成视频数据再进行 H.264 编码，那么对于每一帧与下一帧之间的间隔就是当前这一帧在视频解码播放时在画面渲染时所持续的时长，同样的对于音频数据也有相对应的处理。因此，描述信息中还包括这一帧的持续时间。

视频描述信息所存储的是视频数据中每一帧的地址信息，通过地址信息可以直接的确定到具体帧在硬盘中的存储位置，通过这个位置信息，系统可以快速的读取视频数据。由于视频数据按照一小时进行一个文件的存储，那么在实际的运行中将其整个的放在内存之中将需要大量的内存空间，这对于系统而言是难以接受的，因此通过这样的方式可以将文件中视频帧的位置描述信息放于内存中，用数组来保存，这样对于视频数据帧就可以做到随机读取了

在系统运行和测试的过程中发现，对于视频数据的大小缺少预估，很容易将磁盘空间写满，带来的后果就是系统崩溃。而视频数据存储硬盘上不可能手动地去进行删除操作，因此就需要设计文件的更新策略。

由于不同的视频通道所对应的监控位置不同，因此它所提供的功能和重要性就大不一样，那么对于每个视频通道都为其设置一个更新周期，即在存储空间不足时所更新掉的视频文件的时间间隔。通道初始化时，会从数据库中获取更新周期。

每个视频文件按照一个小时的长度来存储，这样对于回放时的时间信息提供了相对明确的保证。同时创建误差为一分钟，即对于由于系统重启等原因导致的视频数据文件时长小于一分钟的情况将重新记录新文件。

特别的，由于现阶段铁路系统设备改造还在进行，设备资源有限，那么对于硬盘空间不足时也要保证数据的即使更新。所以在创建新文件之前要对硬盘进行空间查询，当剩余空间小于 10% 时，将对每一个通道，根据它更新周期来确定它所删除多少天的视频数据，来保证有足够的空间保存新的视频数据。

4.6.5.2 回访数据的获取与处理

对于视频的来源，同实时播放一样，也分为本地和远端，对于远端的处理，依然是通过“代理”模式来实现的，服务器在本地创建一个虚拟的数据源来为客户提供视频的访问和控制接口，这样对于客户而言视频的来源将会被封装起来，所以对于客户而言，视频的内部实现是透明的，客户只需要知道，能通过接口来实现所需要的视频数据，这样视频源的变化对于外部而言就是透明的了，因此降低了耦合性，更重要的也为以后功能的扩展，特别是回放视频的数据分发提供了基础。

对于本地的视频通道，系统将会根据视频通道在本地的存储信息对于通道存储的所有文件数据进行信息的收集，这包括文件的时间信息，大小等。对于远端数据，代理数据源将会通过远端服务器的转发，来获得视频通道的文件信息。这些信息将会经过编码发送到客户端，客户端经过信息交互的显示，接受到客户的选取后将所选择的视频信息返回给服务器，服务器在收到具体的视频数据的请求后，将会把对应的数据描述信息文件装入内存中，初始化视频处理模块，进行视频数据的处理工作。

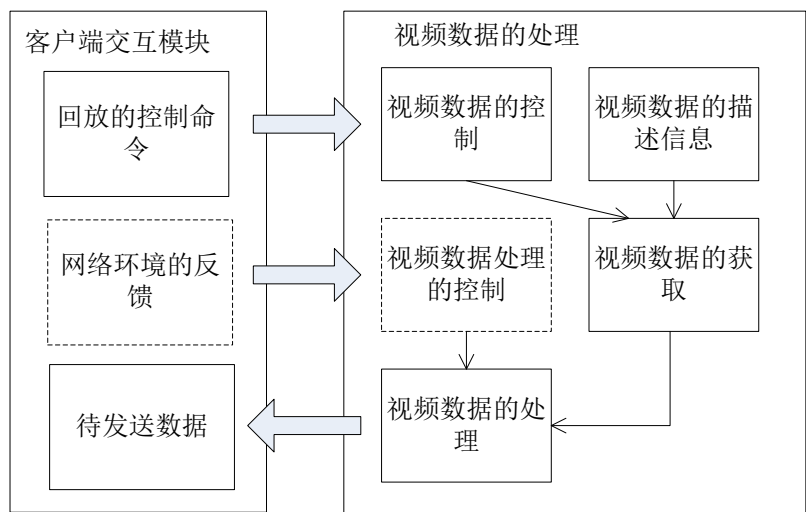


图 4-27 视频处理模块

如图 4-27 所示，在视频数据处理模块中，可以分成五个部分。经过初始化之后，视频数据的描述信息将会为视频数据的获取提供依据，依靠客户端的控制命令，视频数据的控制会对视频数据的获取过程进行针对性的访问控制，获取到的视频数据，经过处理得到满足网络数据发送需求的视频数据，提交给交互模块。特别的，其中由于 H.264 编码对于网络具有很多好的适应性，因此便于以后的扩展，提供了根据网络的当前状态对视频数据进行可控的编码处理，保证流畅的情况下，实现动态的变化。但由于 H.264 的编码处理依然在开发之中，这里的控制功能实现的还不完善。

视频数据的控制：对于回放而言，客户在播放视频时，会对视频进行加速、播放、暂停和点击等操作，那么要实现这些操作相应的要对视频的获取进行与之相关的实现和控制，而这些控制的实现就通过视频数据的控制模块来完成的。客户的控制命令，通过此模块转化为一步步的对视频数据获取控制操作，通过改变视频获取模块中的控制参数，有效的实现视频数据获取的控制。

视频数据的获取：这是核心实现，是与视频源文件直接打交道的实现。通过视频数据描述信息，视频数据的获取可以快速的定位到视频的任意位置，同时经过对控制命令参数的分析和判断可以对获取数据的过程实现所需的功能。

视频的描述信息：顾名思义，是对视频数据文件描述信息的结构体，其中包含视频数据的帧信息，还有视频只能分析的数据信息。这些数据保证了视频数据的随机访问，同时还将提高客户端回放的友好性。

视频数据的处理：从视频原始文件所读取的视频数据并不能直接通过网络发送到客户端，因此要对视频数据进行根据网络状态或者是带宽要求的控制处理，

才能实现满足需求的视频获取工作，这样所得到的数据提交给交互模块之后，能够实现很好的稳定需求。

视频数据处理的控制：如图，由于此功能的实现更多的是为了网络状态而保证回放的流畅性所实现的对于视频数据处理的控制功能。在对 H.264 进行新的处理和编码工作来实现和提高视频的流畅程度，但还处于开发之中。

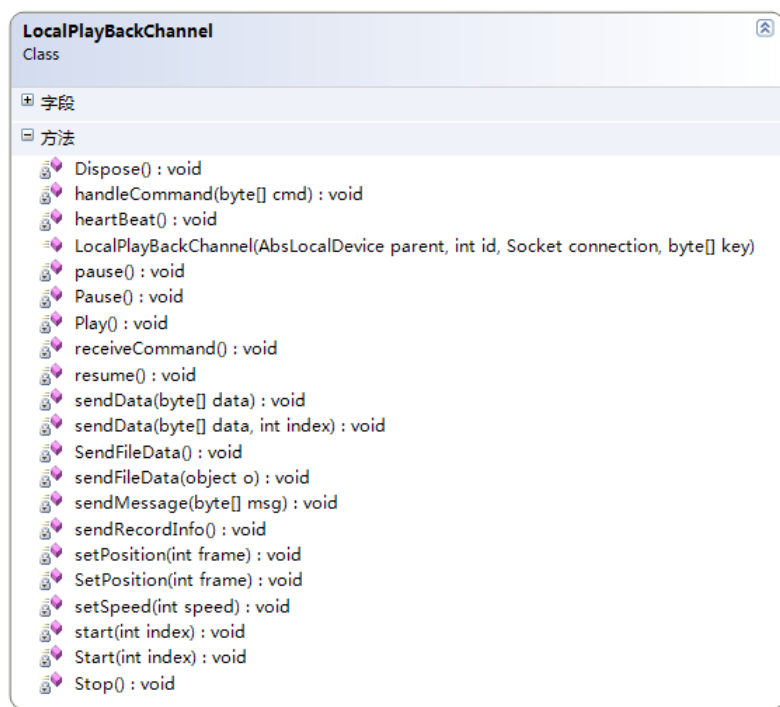


图 4-28 回放数据处理结构图

如图 4-28 所示，视频回放数据处理中，实现了视频数据文件的主要操作，视频数据从文件中获得，将他们传递给请求的 NetPlayerPlayBack 对象，NetPlayerPlayBack 对象再把数据发送到具体的请求客户端。同时，NetPlayerPlayBack 对象会受到客户的控制请求，通过调用 LocalPlayBack 所提供的接口，来实现具体的控制操作。

通过这样的设计，使得数据端与客户交互端能够很好的分割开来，在实现时，两者不必关心对方如何实现的功能，只需要按照规定的接口调用，就能够实现具体的控制和交互。

4.6.6 有关视频播放时的连贯性和实时性优化的设计与实现

在视频监控系统中，播放的实时性和连续性是必须要考虑的两个重要方面。

实时性决定了视频监控系统的可靠性，视频监控必然要求了所观看的实时画面是要最近阶段发生的现场状况。连续性决定了用户的使用感受，如果视频画面发生了卡顿那，必然的对视频监控系统的观感产生了极大的影响，而这对于视频监控的直接评价将会大大降低。

实时性和连续性之间既有矛盾又有联系，由于数据在网络上传输的波动性，视频数据帧无法确保按照规定的 40 毫秒的时间差来持续稳定的送达。在实际中很可能会因为网络的波动导致视频帧数据传输的时快时慢，高实时性的要求就是要保证所收到的视频数据及时的播放出来，这对于收到数据帧的快慢极其敏感很容易出现画面的不连贯，影响用户观看的使用感受。反之，如果为了保证视频播放的稳定和连贯，那么就需要通过缓冲区来“中和”视频数据由于网络播放所带来的抖动性，但过小的缓冲区不足以中和这种波动，过大的缓冲区则无法实现实时性的要求。因此需要对两者的平衡点作合理的设计。

根据实验的总结和需求分析，处理过程将分为客户端和服务端两方面来设计和实现。

在服务端，保证数据发送的实时性。因此就需要在进行数据分发时给予每个客户交互模块最新的数据块，但由于 H.264 视频数据编码的特点，帧与帧之间有密切的联系，特别是 I 帧作为所有帧的基础，其重要性不言而喻。如果使用单一的数据发送缓冲区，一旦数据发送缓慢时会导致后边的帧数据将前边的帧数据覆盖掉，那么一旦 I 帧数据被覆盖了，则后边的所有非关键帧的数据信息将无效，即解码之后也无法实现视频播放，所以缓冲队列的使用是不可或缺的。在队列中一旦队列的长度过大，则说明有数据压码了，而压码带来的后果就是数据的实时性受到影响，此时需要做的就是将最新的数据传送给交互模块，而不再考虑队列中的数据是否已经发送给客户了，因为这些数据已经是有很大的延迟性的了。因此，当 I 帧到来时，将队列中的缓冲区清空，由于 I 帧保存的是完整的一帧图像画面，当 I 帧解码播放时能够迅速的更新视频画面，这样能够保证发往客户的数据都是相对实时性很高的。而对于长度的参考值，是根据两个 I 帧之间的最小值来确定的。

在客户端，同样需要缓冲队列来实现“中和”客户端接收数据过程中的波动性变化。而对于视频播放的流畅性将通过 DirectShow 来控制，对于每一帧数据，DirectShow 可以控制其渲染时间，即这一帧图像的显示时间是 40 毫秒，下一帧可以是 39 毫秒。因此通过这个时间可以保证视频的连续性，而这个时间的获取一般的是在视频设备编码时所携带的，我们可以这样理解，即经过了这些时间后，视频设备又进行了一次视频图像的捕获和编码，因此这一帧图像是经过这些时间

变化后的结果。测试发现并非所有的视频设备都提供这个持续时间，因此就需要通过帧率来模拟那些没有帧的持续时间数据信息，但这样的结果必然会导致视频在播放时的持续时间与真实的时间有误差，所以这里就需要缓冲队列将这种误差所带来的累积平衡掉。同时还可以根据对缓冲区中帧数据的平均到达时间来优化视频播放时的连贯性。

4.7 本章小结

本章从系统的整体设计结构出发，对系统的物理组织结构和逻辑层次结构针对需求分析中的要求进行了详细的设计和实现的阐述。在对系统服务主程序的整体设计和思想进行了分析之后，分别从用户交互模块和数据处理模块两方面对系统的实现做了详细的分析工作。

在用户交互模块的设计和实现中，从功能实现的角度，分别对实时播放和视频回放从流程设计和结构实现两方面进行了详细的设计与实现的分析工作。并针对具体的问题给予了合理的解决方案，达到了预期的效果。

在数据处理模块的设计与实现中，在对设备和通道的连接和处理的设计与实现分析的过程中，分别从流程和结构两个方面入手，特别是对于本地和远端设备和通道的联系与区别做了详细的讨论，将整个数据处理模块的架构完整的建立起来，实现了数据处理模块的低耦合、高扩展的特性。此外还对保证系统稳定运行的权限控制和设备维护两方面做了详细的讨论与实现工作。最后，对视频播放时连贯性和实时性作了设计和优化，提高了系统的表现性能，增加了客户的使用感受。

总的来说，系统实现了需求分析中所提的要求和功能，在功能实现和性能需求方面都做到的良好而有效的设计与实现。

第五章 系统的测试与运行

5.1 系统测试和运行环境

5.1.1 硬件环境

在服务器端，使用主要使用的服务器设备为：IBM System x3850

5.1.2 软件环境

服务器端所使用的配置如下：

.Net 版本	.Net Framework v2.0
MySql 版本	MySql 5.5
Windows 平台	Windows Server 2008

客户端的配置如下：

平台	Windows 7
解码库	CoreAVC 3.0

5.2 系统部署和运行测试

5.2.1 系统的部署

系统的运行是在在 Windows Server 2008 平台下，需要.Net Framework v2.0 和 MySql 5.5 的支持。在所需软件正确安装之后，先要对运行环境进行配置和初始化工作。

1) 数据库的初始化过程创建和配置工作的实现是通过脚本程序来完成实现的，过程如图 5-1 所示。



图 5-1 数据库脚本文件执行

当执行完数据库的脚本文件后，数据的初始化创建工作将会完成，将会按照要求在系统中创建 spjk 数据库，并初始化所需要的表结构、视图和函数等。

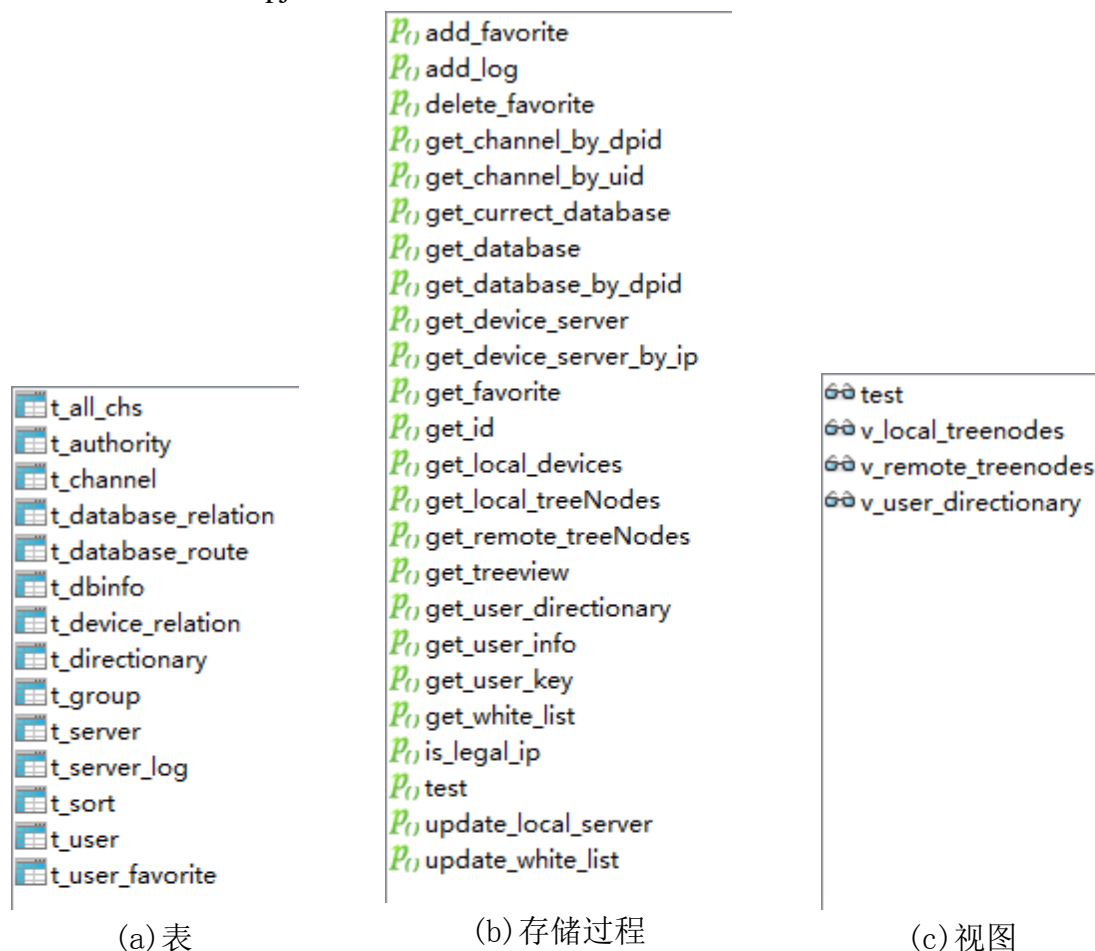


图 5-2 数据库初始化图

如图 5-2 所示，是经过脚本初始化创建之后的 spjk 数据库，图中 (a) 是系统运行所需要的表，(b) 是需要的存储过程，(c) 是视图结构，经查看都已经按照设计的要求成功创建。下一步是需要将具体的环境数据配置到数据库中，来保证系统的正常运行。

2) 通过数据库管理程序对系统数据库进行配置工作。

如图 5-3 所示，是对数据库所在级别对应服务器的用户人员的配置工作，当将用户信息和数据添加的数据库中之后，就表明此用户在当前服务器注册成功。注意，这里的密钥是不允许修改的，而具体的修改过程有专门的实现工具来配置。

如图 5-4 所示，是对数据库所在服务器连接设备的配置和注册界面，主要是对服务器所连接设备的信息进行添加和配置，以及具体通道信息的描述，将这些信

息写入注册表中，以方便在系统运行的过程之中灵活使用。

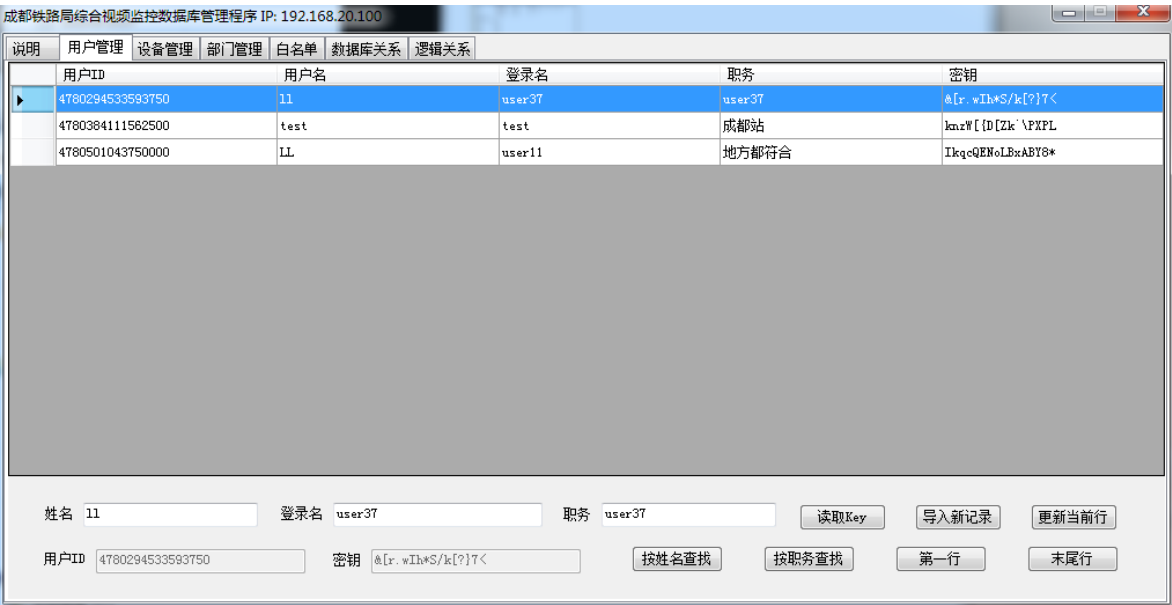


图 5-3 数据库用户管理配置



图 5-4 数据库设备和通道配置

在对设备和人员注册之后，还需要将两者的信息关联起来，就是将逻辑结构和业务权限的整合。如图 5-5 所示，将通道数据和用户信息添加的对应的部门之中，来实现两者数据的有效联系。通过点击图片，来切换人员管理和设备管理。

如图 5-6 所示，服务器还需要配置上下级关系，这是实现远程访问和整个体系结构的关键。所以需要谨慎操作，在根据具体的环境要求实现了上下级关系的配

置之后，数据库的基本信息的配置工作就完成了。

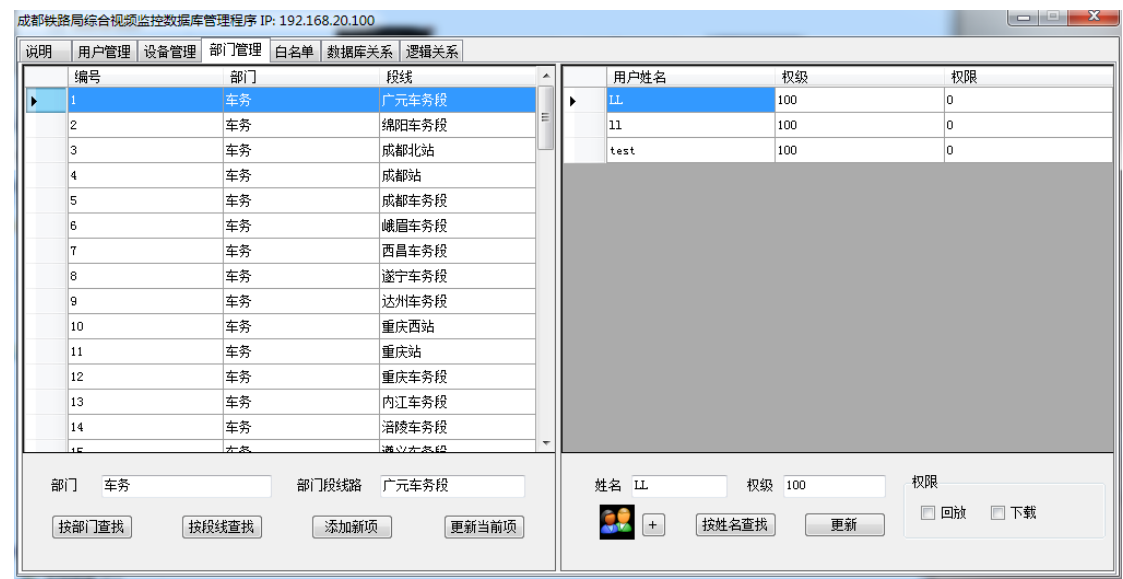


图 5-5 用户和设备关系配置



图 5-6 上下级关系配置

3) 服务端程序被装载到服务器上之后，由于系统在 Windows Server 2008 平台上运行会导致桌面交互的失效，无法通过自身的实现可视化交互，所以服务器的配置工作将由独立的配置程序来实现，如图 5-8 所示。服务器配置程序所配置的分两方面：数据库的配置是描述系统运行时所对应的数据库服务器的信息，服务器配置是服务器在运行时相关的属性信息，这些数据都将写入到注册表中方便操作，如图 5-9 所示。由于服务器往往有多个网卡，所以在服务器配置中需要对网卡进行

选择，同时按照需要填写上报 IP 用于系统运行时特殊状态的信息报警。



图 5-7 服务器信息配置

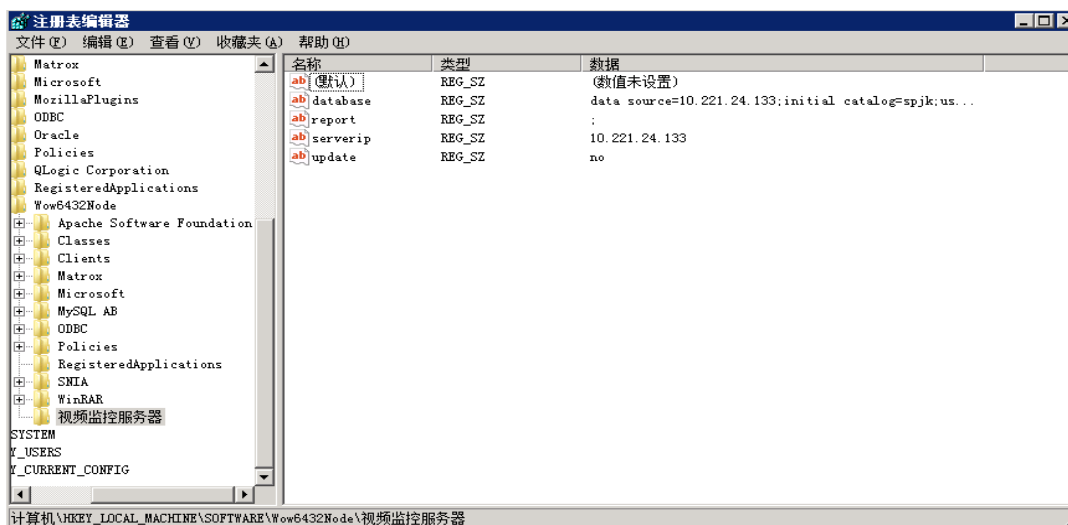


图 5-8 注册表信息

4) 点击确定之后，视频监控主程序将会启动并启用视频监控守护服务，连通设备获取数据，接收客户的请求实现视频监控系统的功能。同样由于 Windows Server 2008 的桌面交互的问题，对于服务器运行状态的信息也需要独立的显示程序来查看。如图 5-9 所示，是视频监控系统运行一段时间之后的状态，左边所记录的是服务器的简单运行日志，主要是显示用户的访问记录。中间表示的是当前连

接到服务器上的客户连接有哪些, 对其中每一条点击将会在最右侧显示用户操作的详细信息。用户操作的详细信息包括用户的发起的命令和请求的权限变化, 通过这些可以对系统运行的状态作简要的分析。

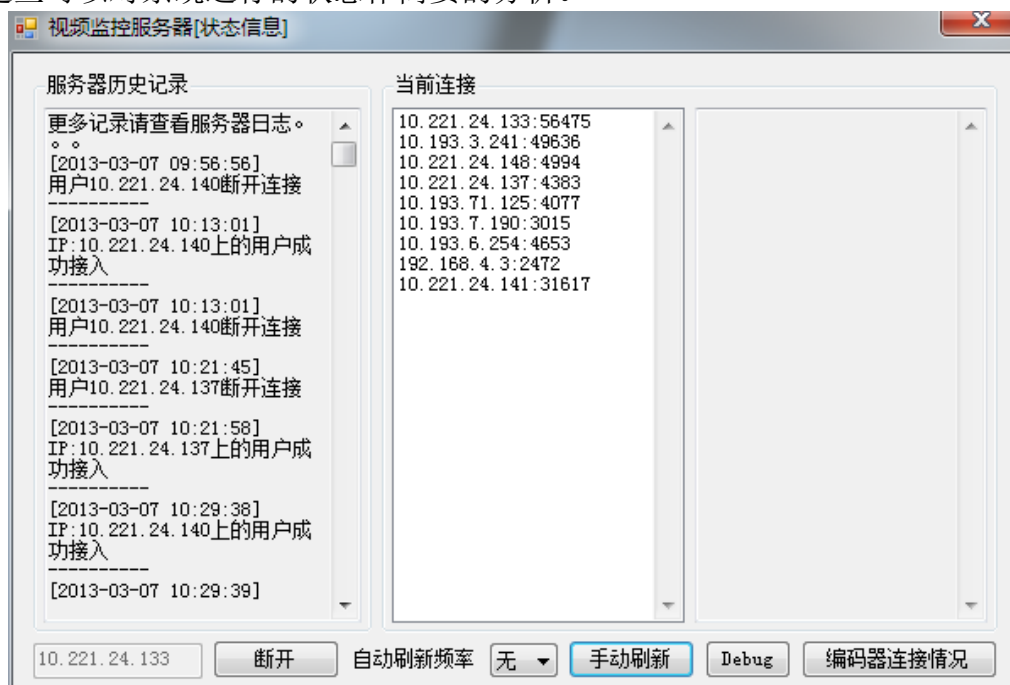


图 5-9 服务器运行状态信息

至此, 服务器的配置工作结束, 服务器正常启动运行
客户端的配置则通过客户端安装软件直接运行即可, 这里不再详细叙述。

5.2.2 系统的运行

服务器可以通过独立的客户端和网页方式访问。

5.2.2.1 客户端的运行

双击客户端执行程序, 弹出登陆界面, 如图 5-10 所示:

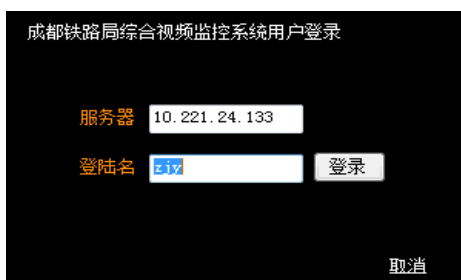


图 5-10 客户端登陆界面

在输入正确的服务器信息和用户名后，系统进入主界面，左侧是所有站点通道的逻辑层次树形结构，鼠标点击播放视频正常如图 5-11 所示，表明系统正常运行能够接受来自独立客户端的连接，并响应视频数据请求。



图 5-11 客户端运行主界面

5.2.2.2 网页端的运行

通过网址访问 web 服务器打开登录网页如图 5-12 所示，在输入正确的用户名后，系统进入主界面。



图 5-12 网页端登陆界面

选择需要访问的通道，点击播放，通过 IE 插件，与服务器相连，如图 5-13 所

示，画面播放正常，表明系统能够正常响应来自 IE 插件的视频数据请求



图 5-13 网页端视频播放

5.3 系统功能测试

5.3.1 实时播放

主要对服务器实时播放的实现是否满足系统分析所完成的设计和要求，对视频数据的处理是否做到了服用分发的效果以达到高效利用带宽的目的，同时对远程数据的访问是否符合设计的逻辑以及数据传输的实现工作进行测试。还对播放画面的流畅度的进行了简要分析。

以最高权限用户访问为了保证不会受到贷款和权限的限制，分别从中心服务器、汇聚服务器和子服务器登录，访问不同层次、不同节点服务器的本地设备视频。

表 5-1 根服务器实时播放测试结果

通道名称	服务器相对级别	测试运行结果
贵阳北内勤	下级服务器	正常
马场通信机械室	下级服务器	正常
内江售票大厅	下级服务器	正常
双流昆方咽喉	下级服务器	正常
遵义二站台	下级服务器	正常

表 5-2 中间汇聚服务器测试结果

通道名称	服务器相对级别	测试运行结果
自贡出站口	其他子节点服务器	正常
凯里票房 1	其他子节点服务器	正常
贵阳通勤口	本级服务器	正常
小西堡外勤	子级服务器	正常
重庆北 VIP 通道	其他汇聚服务器	正常

表 5-3 子节点服务器测试结果

通道名称	服务器相对级别	测试运行结果
蔡家运转室	本级服务器	正常
重庆站一侯	上级服务器	正常
井口职工通道	其他子节点服务器	正常
江津候车室	其他汇聚点服务器	正常
荣昌广场	其他子节点服务器	正常

由表 5-1、表 5-2 和表 5-3 的测试结果可以看出，通过不同节点服务器对不同位置的数据进行访问，其视频播放都能够正常实现。所以系统层次结构设计合理，上下级关系符合逻辑，能够有效的实现视频数据的转发功能。

使用 A 用户登录江津服务器，访问下级白沙服务器的视频数据，视频数据应当通过江津服务器实现“代理”功能，江津服务器的状态如图 5-14 所示。

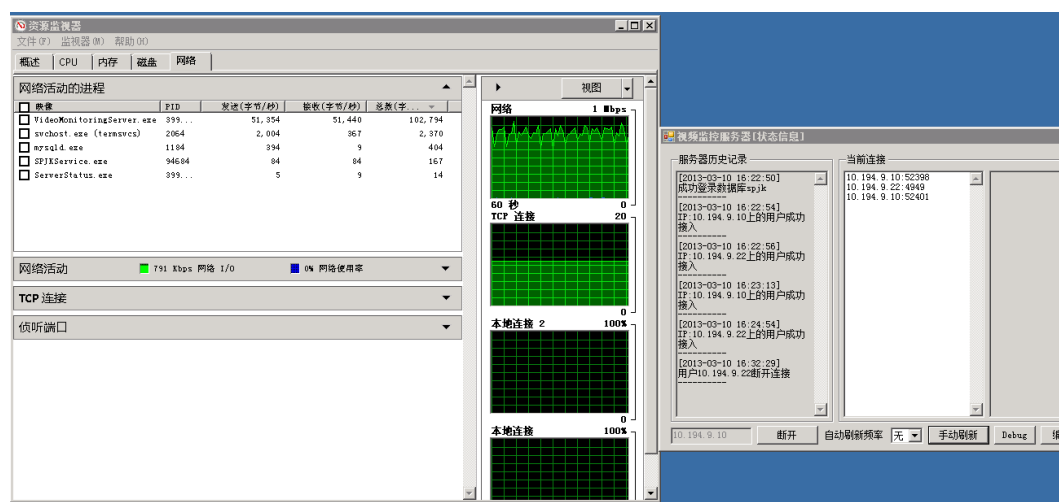


图 5-14 只有 A 用户访问视频数据

可见此时的服务器程序所接受到的网络数据所占用的带宽约为 512K，正好满

是一路 CIF 视频的数据大小。符合系统设计实现的要求。

此时使用 B 用户也从登录江津服务器，对下白沙级服务器请求与 A 相同的视频数据进行访问，江津服务器的状态如图 5-15 所示。

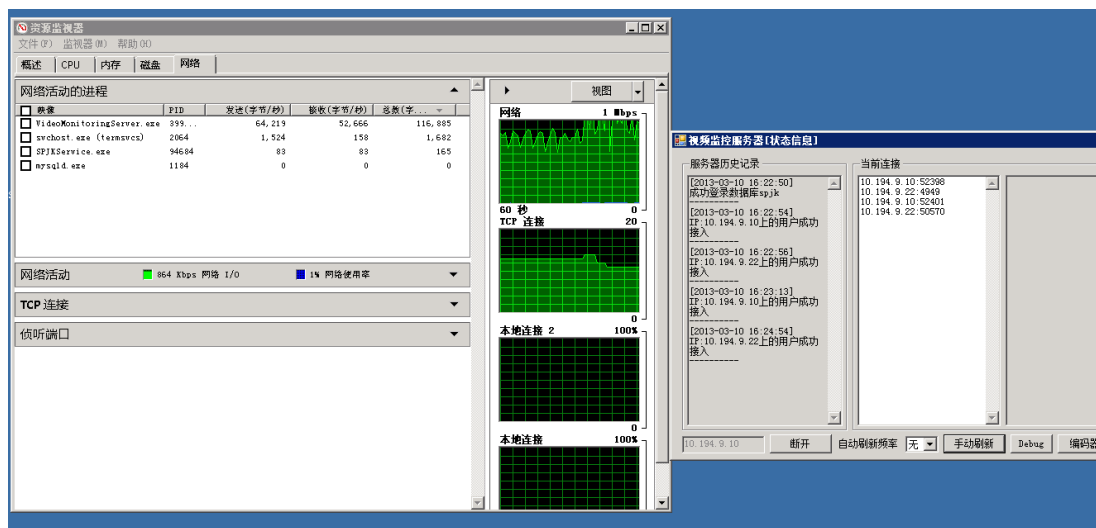


图 5-15 A 和 B 用户同时访问相同的视频数据

关闭 B 用户的视频请求后，江津服务器的网络带宽请求并没有减少，说明 A 用户仍在访问数据使得江津服务器并没有中断与下级服务器的数据请求，运行符合设计实现的要求。B 用户分别对下级不同的服务器视频数据和白沙服务器与 A 用户访问所不同的视频请求进行测试，此时的服务器状态如图 5-16 所示。

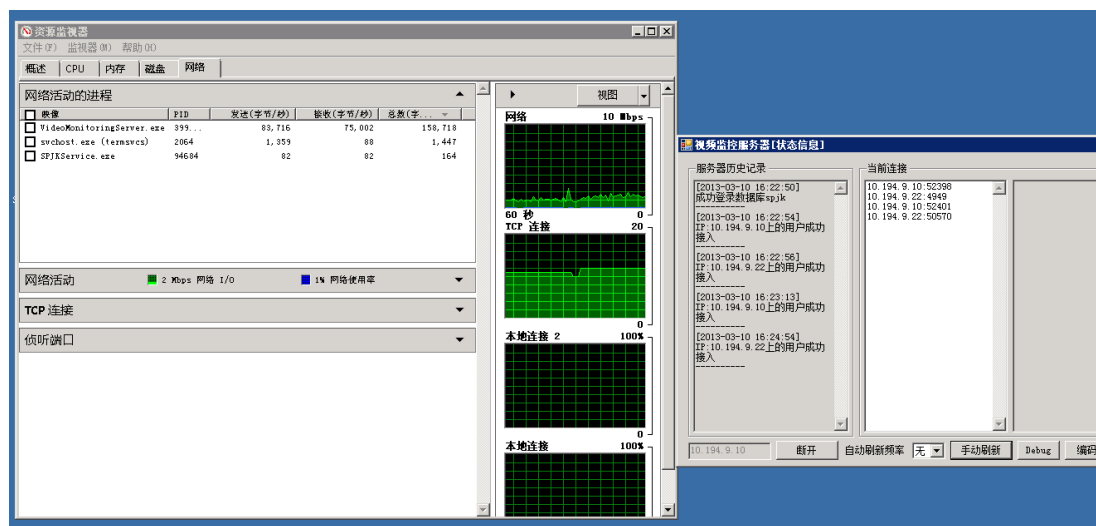


图 5-16 A 和 B 用户同时访问不同的视频数据

可见，当访问的数据相同时，汇聚点服务器接收到的数据带宽使用并没有增加，说明对于相同的视频数据请求，服务器起到了分发视频的作用。而访问不同

的数据时，服务器收到的网络带宽流量有明显的增加，说明服务器对于不同的数据无法实现分发功能，符合系统运行的要求。

由此可以得出结论，测试的结果表明在服务器能够实现实时播放的基本功能，同时在整体结构的设计和实现上符合需求分析的要求，对于数据的转发能够实现有效和及时，这在视频画面的流畅程度上可以明显的感受得到；另外对于经过服务器的数据流，可以根据用户的请求实现分发功能，符合项目设计的初衷，提高了网络带宽的利用效率。整体来说，实时播放功能的实现满足了需求分析中的要求，符合铁路系统的业务逻辑，对现阶段的物理网络状况也具有很好的适应性，达到了系统设计的预期目标。

5.3.2 视频回放

为了避免其他因素的干扰，使用最高权限用户，分别从中心服务器、汇聚服务器和子服务器访问不同节点服务器视频数据的测试结果，如图 5-17 所示。



图 5-17 视频回放实现过程

图中 (a) 所示是对通道进行视频回放请求时，客户端所得到的通道录像信息，当选取了具体需要回放的视频文件后，如图中 (b) 所示，客户端进行视频回放，此时是以正常速度播放的。可见，系统视频回放功能能够正常的实现视频回放的基本功能，对视频数据的远程访问也说明回放功能的转发工作是合理的符合系统设计和实现要求的。

对视频回放所提供的控制，例如暂停，加速，减速，定位操作的测试结果如下：

表 5-4 视频回放功能测试

操作	测试结果
暂停	画面停止，流量为零
加速	画面快速，流量略有增加
减速	画面缓慢，流量明显减少
定位	有效实现定位操作，切换播放画面

可见，系统能够有效的实现视频数据的回放，并且对提供的操纵功能有比较好的实现。符合系统设计的要求。

5.3.3 云台控制和权限管理

云台控制测试，主要是对系统对于用户所发起的云台控制命令的响应和处理功能的实现效果的测试，检查云台能否按照预定的设计完成相应的功能实现。



图 5-18 云台测试结果

测试结果如图 5-18 所示，其中中间的图像时云台未经过控制前的原始位置，分别对其发起上、下、左、右等八个方向的控制命令，云台的转动位置根据要求

很好的实现，同时对焦距的控制也能够变化视频设备的前后景深。所以，系统对于云台控制命令的解析正确，对于云台功能实现正常，满足需求分析的要求，符合设计的初衷。

A 客户是具有高权限客户，B 客户的权限要低一些，所登录的服务器对于远端视频服务器请求的量化带宽为 4，所访问的每一路视频通道都为 1。当单独使用 B 客户请求视频时，如图 5-19 所示，当请求第五路远程访问的视频数据时，系统给出了“带宽耗尽”的信息提示。此时第五路视频的带宽请求将超过服务器所接受的最大量化带宽值 4，这表明，系统的设计对于单用户的请求符合需求分析的要求，满足设计的初衷。

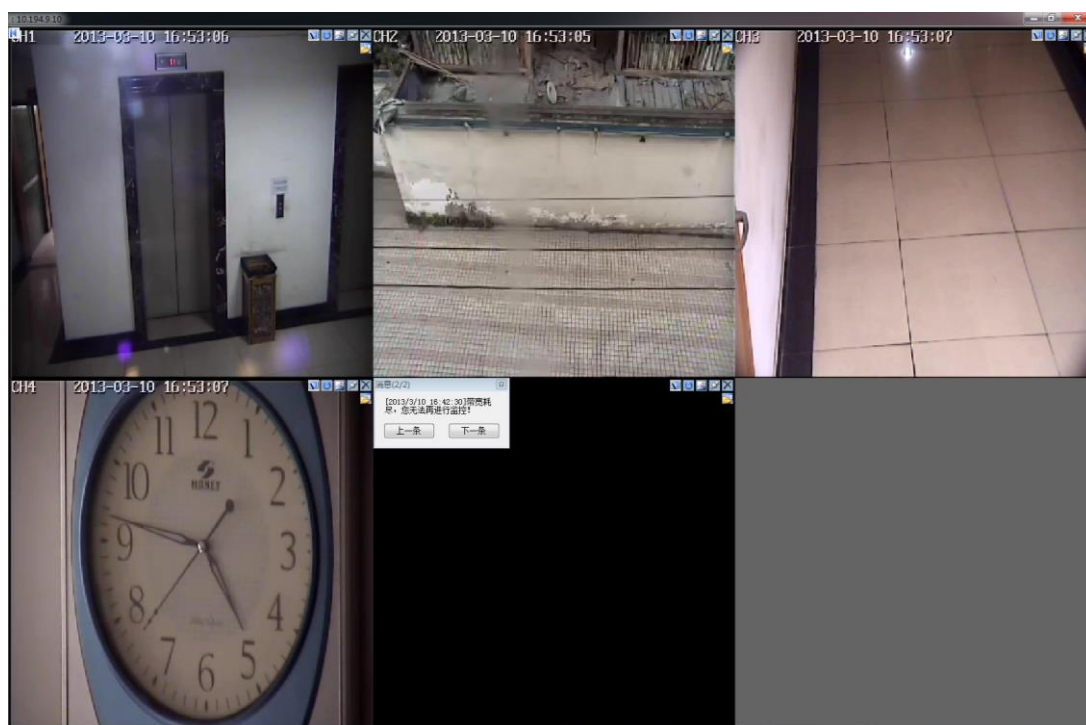


图 5-19 单用户视频请求带宽控制

当 B 客户播放四路视频画面时，此时 A 客户请求与此四路不同的视频数据，即服务器不会实现分发功能，将会对远程访问的虚拟通道进行权限判断。此时 B 用户的一路视频数据停止，并提示“带宽耗尽”的信息，表明此路视频数据所占用的网络带宽被高级用户占用，如图 5-20 所示。这表明，由于 B 用户的权限低，那么他所访问的视频通道的权限就相对较低，因此在权限判断的过程中，A 所请求的新的视频通道的权限将高于 B 用户，那么对于占满网络带宽的远程虚拟设备而言就需要进行判定工作，显然 B 用户的一路数据被 A 所抢占了。实验的结果符合分析的过程，满足需求分析的要求，表明权限控制能够实现系统设计所预期的

效果。



图 5-20 低权限用户带宽被抢占

当客户端进行控制时，一段时间不进行操作后，会受到提示信息“您的控制时间已到”，满足系统设计的需求，证明系统能够正常实现功能；同时对不同等级的客户进行同样的操作，释放控制的时间根据客户等级的提高时间加长，证明功能实现正常，符合系统设计的要求。

5.4 性能测试

对服务器进行用户连接和视频请求的冲击测试，测试发现，在视频请求的频率在两秒钟时，系统能够保证视频画面的稳定获取，随着冲击频率的增加，CPU 和内存的增长很快，特别是内存的消耗对系统的影响非常大，经过一到两天的持续冲击后，会出现内存错误导致系统崩溃。因此，当用户产生大量冲击时，只能通过将用户加入黑名单的方式来避免客户不友好的冲击访问。

当对服务器程序通过多种不同的方式关闭时，系统守护进程都能有效的将程序唤醒，保证了系统的稳定运行，因此满足系统设计的要求。

5.5 本章小结

通过对系统实现的部署对系统的主要步骤进行的测试工作，达到了与预期的效果。特别是针对实时播放和视频回放的功能测试，符合预期的效果，满足系统的要求。对云台控制和权限管理的测试，结果正常，验证了系统的良好设计。对系统性能的测试，保证了系统能够在恶劣条件下稳定运行。总的来说，系统的实现符合项目的需求，合理的完成了项目的任务，有针对性的很好的解决了遇到的难题。

第六章 总结与展望

本论文是基于“成都铁路局视频监控系统”项目，主要是在需求分析的基础上，设计实现了符合基本要求的视频监控系统整体架构，并通过对数据处理模块和客户交互模块的设计与实现，完成了实时播放、视频回放、权限控制和带宽管理等功能的实现。同时对系统的性能，特别是可靠性、响应效率和视频画面播放的实时性和流畅性的完善与改进，增强了系统良好的用户使用感受。

本文完成的主要工作有三方面：

1) 对视频监控系统项目进行需求分析。从铁路系统的整体结构入手，分别对现阶段设备所运行的物理网络环境和铁路系统的业务逻辑关系进行了详细的阐述和研究工作。这对于系统整体结构的设计和实现具有重要的参考价值，是系统合理性、扩展性和稳定性的关键。对于系统功能和性能的分析，明确了系统设计的要求，有针对性的分析和总结了可能遇到的问题和难点，对于设计过程中所要注意的问题做了总结，为系统功能的实现和性能的优化做了铺垫作用。

2) 在需求分析的基础上对系统进行了详细的设计和实现工作，这对整个项目的实现具有决定性作用。在对系统整体结构设计的基础上，特别是结合要求确定了系统的设计框架后，分别从处理流程和模块结构入手完成了系统的设计与实现。在流程设计中，针对所实现的实时播放和视频回放功能，分析了它们具体的实现细节，构建了合理的结构，并将权限控制和带宽管理的功能融入其中，符合设计逻辑和项目要求。在流程分析的基础上，对于模块的设计从用户交互模块和数据处理模块两个方面进行讨论，通过不断的优化和修改，模块结构的设计和实现不但具有稳定的表现效果，还具有低耦合，高扩展，利于灵活的变化和修改的特点，对系统的长期稳定可靠运行具有重要意义。

3) 对系统的设计和实现通过测试工作得以验证。分别从功能测试和性能测试两方面完成了对系统的分析工作。对于需求分析所提及的实时播放、视频回放、云台控制和权限管理的实现，测试结果都表明，系统的设计和实现是合理的。对于系统性能测试的结果表明，系统实现过程中的优化工作是有有效的，为系统的运行增加了砝码。

但系统在设计过程中依然还有很多不足，这是下一步工作需要完成的：

1) 视频回放没有实现分发功能，对于一次性的回放对数据的，特别是网络资

源的浪费将是巨大的，因此在下面的工作里，将会对视频回放的处理过程进行重新的设计和优化。

2) H.264 数据重新编码的分析和实现还需要结合网络带宽的情况，进行优化来更好的满足视频数据在网络中的传输。因此在下一步的工作中，需要结合 H.264 编码的特点根据网络状况的反馈，动态的实现带宽控制的功能，这将使得网络带宽的利用效率大大提高。

视频监控系统已经实现了产品化，在成都铁路局正式投入使用，现阶段，安装的视频设备有上千套，接入站点的两百多个，涵盖了成都铁路局的主要线路。在投入使用以来，系统稳定运行，可靠性高，表现性好，经受住了春运的考验。

视频监控系统平台的建立，仅仅只是视频监控领域很小的一步，如今随着智能分析技术的发展，有越来越多的特性和功能在视频监控平台上实现。特别是能够在解放人力资源的同时，保证高效和精确的监控职能，将在铁路这样的密集的公共场所发挥越来越重要的作用。可以预见，视频监控将会有越来越广阔的舞台，将会实现越来越美好的未来。

致 谢

研究生的三年时光转眼即逝，三年的时光有辛苦，有惆怅，有无助，有压力，有快乐，有喜悦，但无论如何，真心的感谢那些帮助过我的人，是你们让我学会了坚强，学会了感恩，学会了以一颗坚强的心来面对挑战和困难。

感谢您，雷航老师，我的导师，您在开学是对我的教导还萦绕在耳边，那份对学问和技术的钻研精神一直激励我不断前进，即使是我接手项目最困难的时候，依旧坚定了我坚持和努力的决心与信心。

感谢您，郝宗波老师和江维老师，我的项目组老师，是您在项目中给予我了巨大的帮助和照顾，让我能踏实安心的完成项目和研究，即使您在国外访问期间仍然对我的关系和鼓励让我深受感动。

感谢您，邓卫东工程师，是您在工作和生活上给予我的关怀和帮助，特别是与您一起完成项目的这段时间里，我从您身上学到了很多，无论是技术上的还是为人处事上的对于我今后的成长都有着无可估量的作用。

感谢你们，我的师兄张玉生、焦运磊，我的同学王旭鹏、文肇国、何吉、张伟丽、王景丹，我的学弟储典、黄园刚等，是你们陪我度过了三年的时光，在生活和项目上的相互协助所建立起来的友谊我是今后难得的财富。

最后，感谢我的父母，是你们一直在背后坚定的支持我，鼓励我，安慰我，照顾我，我如今能取得的成就离不开你们谆谆教导。

再一次感谢给于过我帮助的所有人，谢谢你们。

参考文献

- [1] 朱春明, 王培华. 铁路多经业与运输主业建立合作伙伴关系的路径选择[J]. 理论学习与探索, 2008: 57-58
- [2] 孟巍. 铁路客运专线综合视频监控系统建设方案[J]. 铁道通信信号, 2007, 43(10): 41-43
- [3] 付强. 高速铁路综合调度系统研究[D]. 成都: 西南交通大学, 2005
- [4] 高惠燕. 浅析网络流媒体视频监控系统的架构[J]. 浙江万里学院学报, 2006, 19(5): 37-40
- [5] 赵飞, 叶震. UDP 协议与 TCP 协议的对比分析与可靠性改进[J]. 计算机技术与发展, 2006, 16(9): 219-221
- [6] Postel J. Transmission Control Protocol[S]. RFC 793, DARPA. 1981
- [7] Postel J. User Datagram Protocol[S]. RFC 768. 1980
- [8] 周晓华. 基于 RTSP 协议的流媒体自适应系统的设计与实现[D]. 上海: 上海交通大学, 2012
- [9] H Schulzrinne, A Rao, R Lanphier. Real-Time Streaming Protocol[S]. RFC 2326, 1998.4
- [10] H Schulzrinne, S Casner, R Frederick et al. RTP: A Transport Protocol for Real-Time Application[S]. RFC 1889, 1996.1
- [11] Soon-kak Kwon, A Tamhankar, K R Rao. Overview of H.264/MPEG Part 10[J]. Visual Communication and Image Representation, April 2006, vol.17: 186-216.
- [12] I E Richardson. The H.264 advance video compression standard 2nd Edition[J]. Wiley, 2010.
- [13] 张可义, 岳秀江, 韩立新. 视频监控系统工程应用研究[J]. 制造业自动化, 2007, 29(3): 96-97
- [14] 杨建全, 梁华, 王成友. 视频监控技术的发展与现状[J]. 现代电子技术, 2006, 29(21)
- [15] 郑世宝. 智能视频监控技术与应用[J]. 电视技术, 2009, 33(1)
- [16] 刘治红, 骆云志. 智能视频监控技术及其在安防领域的应用[J]. 兵工自动化, 2009, 28(4)
- [17] 张春田, 苏育挺, 张静. 数字图像压缩编码[M]. 北京: 清华大学出版社, 2006
- [18] CCITT. CCITT Recommendation H.261. Video Codec for Audio Visual Services at px64kb/s[S]. 1990
- [19] ITU-T Recommendation H.263[S], Video Coding for Low Bit Rate Communication, 1998
- [20] 余兆明. 图像编码标准 H. 264 技术[M]. 人民邮电出版社, 2006.
- [21] Lu C H. Robust Error-resilient H. 264/AVC video coding[J]. 2012.
- [22] 毕厚杰. 新一代视频压缩编码标准: H. 264/AVC[M]. 人民邮电出版社, 2005.

- [23] 蔡安妮, 孙景鳌. 多媒体通信技术基础[M]. 电子工业出版社, 2000.
- [24] 王军. 多媒体网络传输的研究与实现[D]. 中国人民解放军国防科学技术大学, 2002.
- [25] H Schulzrinne. RTP, Profile for Audio and Video Conferences with Minimal Control [S] . RFC 1890, 1996.
- [26] ITU Telecommunication. Standardization Sector of ITU, Video Coding for Low Bit Rate Communication[R] . Draft ITU2T Recommendation. 263 Version 2, 1997.
- [27] 内格尔, 伊夫杰, 格林, 等. C# 高级编程[M]. 清华大学出版社, 2006.
- [28] Williams M. Visual C# .NET 技术内幕[J]. 2003.
- [29] Vaswani V. MySQL 完全手册[J]. 2004.
- [30] CoreCodec Inc., CoreAVC. Available: <http://www.corecodec.com/products/coreavc.html>
- [31] 黄婧芝, 付显峰, 吴国平. 基于 DirectShow 的 H. 264 视频采集与播放的实现[J]. 电子技术, 2010, 2: 025.
- [32] FFmpeg. <http://www.oschina.net/p/ffmpeg/>
- [33] 颜菲菲, 高胜法, 刘晓兰. 远程视频监控系统的安全可靠性研究[J]. 计算机工程与设计, 2005, 26(9): 2494-2496.
- [34] 刘舒, 沈军辉. 流媒体技术及其在视频监控系统中的应用[J]. 中国人民公安大学学报 (自然科学版), 2007, 1.
- [35] DVR, <http://baike.baidu.com/view/837.htm>
- [36] NVR, <http://baike.baidu.com/view/1739928.htm>
- [37] IPC, <http://baike.baidu.com/view/63675.htm>
- [38] 何青林, 陈朝武, 卢煜, 等. 基于 SIP 的视频监控联网系统的设计与实现[J]. 电视技术, 2010 (5): 116-118.
- [39] 宗文杰, 余青松, 张敏, 等. 基于 GIS 的视频监控系统研究[J]. 计算机工程与设计, 2011, 32(002): 745-748.
- [40] 余腊生, 刘勇. 基于网络的智能视频监控系统的设计与实现[J]. 计算机工程与设计, 2009 (16): 3879-3882.

攻硕期间取得的科研成果

- [1] 李海涛. H.264/AVC 视频编码技术的分析与其编解码器的比较. 计算机学院硕士研究生学术交流论坛