

学校代码：10255

学号：2121063

实时同步虚拟协同实验平台的设计与实现
**DESIGN AND IMPLEMENTATION OF REAL-TIME AND
SYNCHRONOUS COLLABORATIVE VIRTUAL
EXPERIMENT PLATFORM**

专业：信息与通信工程

作者：孔德凤

指导教师：陈 光

日期：2015 年 1 月 8 日

东华大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

东华大学学位论文版权使用授权书

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

保密 ☐，在 _____ 年解密后适用本版权书。

本学位论文属于

不保密 ☐。

学位论文作者签名：

日期： 年 月 日

指导教师签名：

日期： 年 月 日

实时同步虚拟协同实验平台的设计与实现

摘 要

随着计算机科学和信息技术的不断发展,虚拟实验教学已成为高校实验教学的一种重要形式。尤其对于计算机网络的实验教学,由于传统的网络实验需要大量昂贵的网络设备,但高校日渐庞大的招生规模使得购买网络设备的资金越发紧缺,无法满足教学需求,所以,虚拟计算机网络实验成为解决此问题的有效途径。而协同虚拟实验是一种特殊的虚拟实验,它可以在网络空间中建立一个虚拟的交互式学习环境,通过模拟和实现传统课堂的教学功能来为地域分散的网络用户提供一个共享协作式教学实验环境,协同虚拟实验的创新与开发也成为近几年高校实验教学的重要研究方向。

本文以本课题组开发的远程网络教育实验公共服务平台和平台上一系列单人基础组网实验为基础,以提高实验的效率、加强学生在实验过程中的合作精神为目标,提出了虚拟协同实验平台上的实验操作全部配置化的设计思想,利用网络编程技术、C#编程语言、.NET类库及 Sql Server 数据库等关键技术,设计和实现了允许多人实时同步合作完成同一实验的虚拟协同实验平台。文中首先对实时同步虚拟协同实验进行需求分析和总结,在此基础上,提出了实时同步虚拟协同实验平台的设计目标和功能设计。参照真实计算机网络的相关实验案例,本文分别从实时同步实验操作、同步搭建实验拓扑和设备操作的并发控制等几个关键点进行了分析和研究,设计并实现了实时同步

虚拟协同实验的各项功能。文中提出了实时同步连线算法，利用此算法一方面完成了协同搭建实验拓扑时拓扑结构的实时同步，另一方面实现了学生自主选择网络拓扑的功能；在实验操作全部配置化的思想下，提出了实时同步实验操作的方法，实现了配置信息的实时同步。文中还设计了适合本实验平台的“改进的高优先级两阶段封锁法”并发控制机制，实现了对学生操作实验设备的并发控制。

最后，本文以两人的虚拟 rip 路由选择协议实验为例，对实时协同虚拟实验平台的各项功能进行了测试，并进一步说明了本文方案的有效性和应用价值。

关键字：虚拟协同实验，实时同步，连线算法，并发控制，C#

DESIGN AND IMPLEMENTATION OF THE REAL-TIME SYNCHRONOUS COLLABORATIVE VIRTUAL EXPERIMENT PLATFORM

ABSTRACT

With the continuous development of computer science and information technology, virtual experiment teaching has become an important form of experimental teaching, especially for experimental teaching of computer networks. Due to the traditional network experiment requires a lot of expensive network equipments, but the shortage of university funds are unable to meet the huge growing enrollment, so the virtual computer network experiments become an effective way to resolve this contradiction. Collaborative virtual experiment is a special virtual experiment, it can achieve the establishment of a virtual interactive learning environment in cyberspace, through simulation and implementation capabilities of traditional classroom teaching to provide a shared collaborative teaching、learning and experimented environment for geographically dispersed network users. Collaborative innovation and development of virtual experiments has become an important research direction of the university experimental teaching in recent years.

In this context, our research group has developed a remote network education experiment in public service platform and a series of single people basic networking experiments. In order to improve efficiency and enhance students' experimental cooperative spirit, the author designed and accomplished an virtual collaborative experiment platform which is based on the existing single virtual experiment platform by using of the proposed operation of the entire configuration of the experimental design, network programming, C # programming language, .NET class libraries and Sql Server databases, to allow people to complete the same experimenta in the same time. Firstly, the author analyse and summarize the reuquests of real-time synchronization of virtual collaborative experiment, on this basis, it proposed the construction and functions of virtual collaborative experiment platform. On the basis of computer network real test case and experimental data, this paper do researchs on real-time synchronization experimental operation, synchronous lab topology building and concurrency control of device operation, to design and implemente various functions of real-time synchronization of virtual collaborative experiment. In this process,this paper proposed the connection algorithm, the one hand, the algorithm achieve real-time synchronization when people build structures together, on the other hand, the algorithm achieve the function of choosing the network topology independently;with the concept of all

operations in the experimental are configured, this paper design the Real-time synchronization method, combining this method to achieve a real-time synchronization of configuration information; this paper designed "Improved two-stage high-priority blockade law" to control concurrent operation of the equipment. This process gives the text in detail.

Finally, real-time collaborative virtual experiment platform's functions were tested by rip protocol experiment with two people,it further illustrates the effectiveness and value of this study program.

Kong Defeng (Signal & Information Processing)

Supervised by Chen Guang

KEY WORDS: virtual collaborative experiments, real-time synchronization, connection algorithms, concurrency control, C #

目 录

第一章 绪 论.....	1
1.1 课题的背景与意义.....	1
1.2 国内外研究现状	2
1.3 课题的研究内容	4
1.4 论文组织结构	4
第二章 相关实现技术	5
2.1 .NET Framework	5
2.1.1 公共语言运行时 (CLR)	5
2.1.2 类库(FLC)	6
2.1.3 Windows 窗体控件	7
2.2 C#编程	7
2.2.1 C#语言概念及优势	7
2.2.2 C#网络编程方法概述	8
2.2.3 C#Delegate 编程	9
2.3 C/S 结构.....	9
2.4 SQL SERVER 2008	10
2.5 本章小结	11
第三章 实时同步虚拟协同实验的设计方案.....	12
3.1 单人虚拟组网实验平台概述	12
3.2 虚拟协同实验平台的设计目标	13
3.3 虚拟协同实验平台的功能设计	14
3.4 本章小结	14
第四章 实时同步虚拟协同实验的实现	16
4.1 实时同步实验操作.....	16
4.1.1 真实实验过程中的实时同步协同	16
4.1.2 实时同步实验操作的实质	18
4.1.3 实时同步实验操作的设计	19
4.1.4 实时同步实验操作的实现	21
4.1.5 异地实验操作的实时显示	25
4.2 同步搭建实验拓扑.....	27
4.2.1 实际实验拓扑的同步搭建	27
4.2.2 虚拟拓扑的设计与实现	28
4.2.3 实时同步连线算法的设计	30
4.2.4 实时同步连线算法的实现	31

4.3 并发控制机制	34
4.3.1 实验用户的注册和登录	34
4.3.2 虚拟协同实验组的创建	35
4.3.3 并发控制机制的设计	38
4.3.4 并发控制机制的实现	39
4.4 本章小结	40
第五章 实时同步虚拟协同实验的运行	41
5.1 实时同步实验操作的实例	41
5.2 同步搭建实验拓扑的实例	42
5.3 并发控制机制的实例	44
5.3.1 学生注册和登录实例	44
5.3.2 学生选择实验和实验组	44
5.3.3 学生选择实验和实验组	45
5.4 本章小结	47
第六章 总结与展望	48
6.1 工作总结	48
6.2 研究展望	48
参考文献	49
攻读学位期间的研究成果	51
致 谢	53

第一章 绪论

1.1 课题的背景与意义

在信息技术和计算机网络飞速发展的今天,远程教育获得了蓬勃的发展,远程教育是综合应用一定时期的技术开发和利用各种教育资源为学生提供教育服务的总称^[1],其教育模式和教学方法日趋成熟,开设的学科越来越丰富,学科体系越来越完善。但是受时间、空间、成本等多方面的限制,如何在远程教育中高效地开展实验教学仍然是困扰教育管理者的一大难题,虚拟实验的出现,为解决这一难题提供了一种有效的解决方案^[2]。

虚拟实验室是一种基于 VR 虚拟现实技术、Web 技术构建的开放式网络化的虚拟实验教学系统,是现有各种教学实验室的虚拟化和数字化。虚拟实验室由虚拟器材库、开放式实验室管理系统和虚拟实验台组成^[3]。虚拟实验室的出现为开设各种虚拟实验课程提供了一个全新的教学环境。虚拟实验台类似于真实的实验台,可供学生亲自动手配置、连接、调节和使用每一个实验仪器设备。教师也可以利用虚拟器材库中已有的器材进行自由搭建完成各种典型实验案例,这一特性是虚拟实验室区别于一般实验教学课件的重要特点^[4]。目前国内外在远程教育实验教学的研究和应用方面虽然已经获得了一些成果,但是虚拟实验网络教学平台多采用的是单机版的虚拟实验软件,学生与学生之间交互性较差,学员独自进行大型实验时会有很多的冗余工作,这严重制约了实验教学质量的提高。

计算机支持的协同工作的提出有效的解决了虚拟实验网络教学平台的搭建问题。计算机协同工作(Computer Supported Cooperative Work)是利用计算机和通信技术建立一个可以多人协同完成的工作环境,这一概念最早是由美国 MIT 的艾琳·格雷夫(Iren Grief)和 DEC 的保罗·卡什曼(Paul Cashman)于 1984 年在一个专题讨论会上首次提出的,当年他们提出这一概念是为了描述他们当时正在组织研究一个关于如何利用计算机技术支持交叉学科的研究人员可以共同工作的课题,他们利用通信技术和多媒体技术建立了一个可以协同工作的环境。在此环境下参与者可以相互合作,共同解决在一个研究领域、一个项目、一个产品或者求解学术上的同一个难题^[5]。

多人协作虚拟实验室是一种特殊的虚拟实验室,是指应用虚拟现实技术与多媒体技术,在计算机上提供近似真实的虚拟实验环境,当学生登陆系统加入一个协作实验组后,同组的多名学生开始共同做一个实验,实验组的所有操作都是同步的,通过相互之间的协作实验组成员可以共同完成一个实验。它的出现实现了在网络空间中搭建一个虚拟的交互式的学习环境,在这个环境中通过模拟实

现传统课堂上的教学功能,从而为地域分散的网络用户提供了一个共享协作式的教学及实验环境,使其不仅具有一个自由的教学空间,用户还可以在这个环境中进行各种实时的交流和协作实验操作,满足 CSCW 所要求的 WYSIWIS(What You See Is What I See)功能,综上所述,我们可以看出对虚拟协同实验系统进行深入的研究具有十分重要现实意义^[6]。

本课题组根据《计算机网络》教学需求,在本学院网络通信实验室的实验设备基础上,设计开发了远程网络教育实验公共服务平台。该课题采用 B/S 模式,学生只需通过浏览器,访问此网站,就可以不受时间和空间的限制进行虚拟的实验操作^[7]。但此课题现有的虚拟实验还存在着一定的不足,譬如只有单人实验,不能多人协同完成实验。本文将以单人虚拟组网实验平台为基础,进一步深入研究并开发了实时同步虚拟协同实验平台,目的在于可以使多名学生同步协作完成同一项实验。在此虚拟协同实验平台下,学生不仅能够操作基本的虚拟实验,巩固计算机网络知识,提高自身的实验动手能力,更能通过协同合作增强学生的合作意识、提高实验效率。

1.2 国内外研究现状

虽然提出虚拟实验的概念只有二十余年,但是由于它具有广阔的应用前景,各国均在该领域进行大力的研发并且已经取得了一些进展。对于虚拟协同实验系统,目前国内外还没有出现作为正式产品的同步协同虚拟实验室,但对虚拟实验室和 CSCW 两方面的研究都已开展得如火如荼^[8]。其中国外比较著名的有:

(1)1997 年德克萨斯洲大学的 PatriciaJ.Tene:先生设计了一个虚拟协同实验系统,其目的是为了能够获得协同实验环境中涉及的一些关于物理科学实验室所固有的科学方法,使用这样的实验室,使得学生必须进行协作实验,从而让学生可以在协作学习小组中加强彼此之间的信任。

(2)2007 年 Lianguan Shen 等在异步协同环境下工程设计教育^[9]中提出了通过多种角色的概念来区分不同的操作权限。不同的角色提供与该角色相对应的权限和义务。除此之外,文章中定义的操作实体不仅具有几何属性,比如尺寸、类型等,还包括协作属性,用于包含操作中生成的协作信息。

(3)2009 年 Jara 等在在线实时虚拟协同实验室^[10]中实现了一个可以多人异地轮流操作同步观看的实时协作虚拟实验室,文中使用加锁法对实验室进行并发控制,在教师演示的过程中学生是不能参与操作的,如果学生对实验过程存在疑问要首先经过老师同意才能操作实验,并且同一时刻只能有一个学生可以操作实验面板,所以该实验室不支持异地多人同时操作。

(4)2009 年 Qin Jing;Choi,Kup 一 Size 等在基于集群混合网络架构的虚拟协同

手术^[11]中提出了一个支持协作虚拟外科手术的完整框架。该框架主要是基于duster混合网络体系结构完成的,在该框架中通过采用多路传输传送参与者的所有相关更新信息的方式来减少系统的网络延迟,由管理服务器维护系统的一致性。协作虚拟环境(CVES)的研究使得在外科手术中进行模拟协同工作成为可能,但由于在保持最快网络存取状态一致性上存在技术瓶颈,所以要实现一个高效能协作外科手术模拟仿真系统仍然是一项十分艰巨的任务。

与国外相比,国内的虚拟协同实验研究起步较晚,但是仍取得了一定成果,较具代表性的虚拟协同实验室研究有:

(1) 2000年湖南科技大学的李仁发在基于虚拟原型的虚拟实验室实现原理研究中,采用三层结构模式实现了一个支持计算机协同操作的虚拟实验系统,但系统只实现了应用层上的网络协同工作。

(2) 2004年来自于湘潭师范学院的彭晖等人把电路实验作为实验操作模型,使用Java语台开发出了一套基于web的同步协同虚拟实验室系统^[12]。该系统采用令牌环+消息队列的设计思想来控制系统的并发问题,主要的思想是通过应用服务器维护一个令牌环,然后该令牌环以极短的时间间隙在各个用户之间进行传递,从而使组内的实验成员可以依次执行自己的实验操作,最终共同完成一项实验任务。

(3) 2006年郑耿忠在基于web的协同电路虚拟实验室中采用串行化法的思想完成了一个关于电路的协同虚拟实验室的设计^[13],这个系统通过使用一个并发控制器为每个共享资源建立对应的动态进程队列,当用户向并发管理器提出数据操作请求时,并发控制器就会检查该共享资源所对应的进程队列。如果该队列中有其他进程,则对该请求不作应答,否则作应答。

(4) 2011年马艳芳在《虚拟协同实验的研究与设计》一文中提出了一种使用“多种令牌”+“操作队列”进行并发控制的方法、用于协同信息共享的分布式网络传输模式,面向对象建模及与并发控制机制的协作这三个关键技术实现《计算机网络》虚拟协同实验系统中的异步方式教案实践^[14]。

由此可见,虚拟协同实验系统已广泛应用在国内外多个高校的多个学科之中,许多虚拟实验室已投入到了实际应用中。同时,在计算机网络通信方面,许多高校开始关注计算机组网虚拟实验系统的建设和开发,并且取得了一定的成果。然而,就虚拟协同计算机组网实验系统而言,国内目前还是停留在理论研究阶段,实际都是采用异步的方式实现协同。

针对这一不足,本文在本实验组研究和开发的单人虚拟组网实验平台的基础上,结合实际需求进行了深入的研究并最终搭建了实时同步虚拟协同实验平台,使学生可以自由组合共同完成同一项实验,并且很大程度的提高了实验效率,与此同时也让虚拟计算机网络实验过程更贴近实际实验流程^[15]。

1.3 课题的研究内容

本文基于网络编程技术、并发控制机制、.NET 类库、C#和思科网络设备，在本课题组单人虚拟组网实验的基础上，研究和开发了实时同步虚拟协同实验平台，此实验平台严格模拟实际实验的流程^[16]，学生按小组完成指定的实验。主要的研究内容有：

（1）对本课题组开发的单人虚拟组网实验的特点进行分析，提出实时同步协同虚拟实验的设计要求和设计目标。

（2）对实时同步协同虚拟实验进行总体的设计，包括实时同步实验操作、网络拓扑的同步构建和并发控制机制的设计。

（3）分析计算机网络实验过程，提出将所有实验操作进行配置化的设计思想。

（4）分析计算机网络实验拓扑构建的实现方法，提出并设计了一种连线算法，实现了虚拟协同网络拓扑的自主划分。

1.4 论文组织结构

论文共分六个章节，具体安排如下：

第一章 绪论。主要介绍了本文的研究背景与意义，并通过研究分析国内外的虚拟协同计算机网络实验的研究现状，提出了本文的研究内容。

第二章 相关实现技术。简要阐述本文所涉及的关键性技术。

第三章 实时同步虚拟协同实验的设计方案。对本课题开发的虚拟实验平台进行了简要的介绍，并以该平台中一个具体的基础组网虚拟实验为例，阐述了实验的基本过程和特点，分析了该实验在实际应用当中的不足之处。在此基础上，提出了协同虚拟实验平台的设计方案。

第四章 实时同步虚拟协同实验的实现。主要针对实时同步实验操作、同步搭建实验拓扑和并发控制机制等几个方面，分别提出实验操作配置化的设计思想、连线算法及“改进的高优先级两阶段封锁法”的并发控制机制，并通过大量的效果图和代码示例对开发实现过程进行了详尽的阐述。

第五章 实时同步虚拟协同实验平台的运行。主要针对实时同步实验操作、同步搭建实验拓扑和并发控制机制等几个方面，对虚拟协同实验的功能进行实例测试，并进一步阐述各个功能实现的过程和特点。

第六章 总结与展望。对本文所做的研究和成果进行总结，并对进一步的研究工作进行了展望。

第二章 相关实现技术

本章主要介绍了本实验平台所使用的开发平台、开发工具及实现语言，并通过对比和分析阐述了选择开发平台和实现语言的理由。除此之外本章还介绍了涉及到的关键性技术。最后，针对本文的研究内容给出了技术层面的定义。

2.1 .NET Framework

.NET Framework 是一种用于 Windows 的新托管代码编程的模型，该模型支持生成和运行下一代应用程序和 XML Web Services^[17]。它将强大的功能与新技术结合起来，用于构建具有视觉上引人注目的用户体验的应用程序，实现跨技术边界的无缝通信，并且能支持各种业务流程，主要功能体现在：统一的程序设计模式、多平台应用程序、多语言集成、对 web 应用和服务的支持、自动资源管理和轻松进行部署。.NET Framework 包含两大组件：公共语言运行时 Common Language Runtime(CLR)和类库 Framework Class Library(FCL)^[18]，其结构如图 2.1 所示：

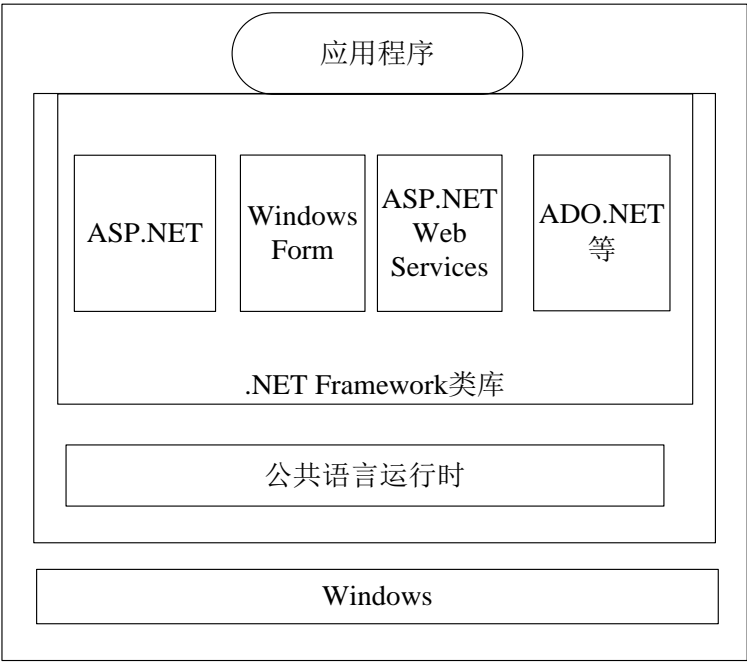


图 2.1 .NET Framework 结构

2.1.1 公共语言运行时（CLR）

公共语言运行时（CLR）是.NET Framework 框架的基础，可以把运行时看作是一个在程序执行时负责管理代码的代理，它主要有线程管理、内存管理和远

程处理等职责,除此之外它还强制实施严格的类型安全以及其他形式的可提高可靠性和安全性的代码准确性^[18]。实际上,代码管理是运行时的基本原则,通常我们把在 CLR 的控制下运行的代码称为托管代码。CLR 执行编写好的源代码之前,需要编译它们,在.NET 中,编译分为两个阶段:

(1) 把源代码编译成中间语言(IL)。

(2) CLR 把 IL 编译为平台专用的代码,两阶段如下图 2.2 所示。

这两阶段很重要,Microsoft 中间语言是提供.NET 的许多优点的关键。Microsoft 中间语言与 java 字节代码共享一种概念:它们都是低级语言,语法简单,可以非常快速的转换为内部机器码。对于代码来说,这种精心设计的通用语法有平台无关性、提高性能和语言的互操作性的优点^[19]。

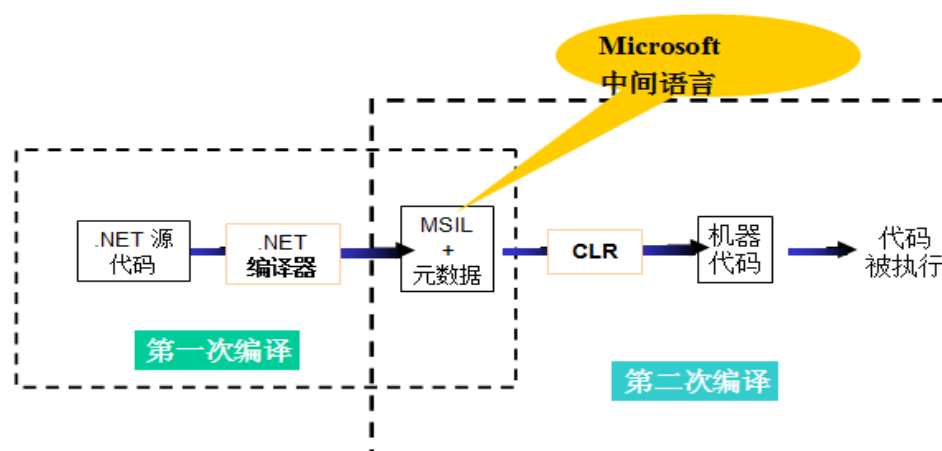


图 2.2 .NET 编译过程

2.1.2 类库(FLC)

.NET Framework 类库是一个与公共语言运行时紧密集成的可重用的类型集合。该类库是面向对象的,并提供从中导出托管代码功能的类型,它可以完成以前要通过 Windows API 来完成的绝大多数任务^[18]。这些类派生自与中间语言相同的对象模型,也基于单一继承性。.NET 类库的一个优点是它们非常直观和易用,VB 和 Java 开发人员非常熟悉这种方式,对于 C++开发人员来说这是极大地解脱。另一方面,C++开发人员总是很容易访问整 Windows API,而 VB 和 Java 开发人员只能访问其语言所能访问的基本操作系统功能。.NET 类库将 VB 和 Java 库的易用性和 Windows API 函数的丰富功能结合起来。类库足以应付日常工作使用,如果需要调用 API 函数,.NET 提供了所谓的“平台调用”,来确保对数据类型进行正确的转换^[19]。.NET Framework 类型使您能够完成一系列常见的编程任务(包括诸如数据收集、文件访问、字符串管理、以及数据库连接等任务)。除这些常规任务之外,类库还包括一些可以支持多种专用开发方案的类型,例如:

控制台应用程序、Windows GUI 应用程序(Windows 窗体)、Windows Presentation Foundation (WPF)应用程序、ASP.NET 应用程序、Windows 服务、WCF 等^[20]。

微软公司发布了众多版本的.NET，最高的版本是今年 2 月发布的.NET Framework 4.5，但考虑到实际的操作环境，本文选用的是.NET Framework 2.0。

2.1.3 Windows 窗体控件

在.NET Framework 开发中，Windows 窗体是一种特殊的类，其本身可以像普通类一样使用。典型的 Windows 窗体包括标准 SDI 窗体（单文档窗体）、MDI（多文档窗体）、对话框以及图形窗口等，这些窗体的创建和添加、窗体的大小和比例、窗体外观等形状，事实上都是 Windows 窗体类的属性^[21]。Windows 窗体提供执行许多功能的控件和组件，绝大多数 Windows 窗体控件都是派生于 System.Windows.Forms.Control 类，该类定义了控件的基本功能，因此为众多 Windows 窗体控件所继承。在 Windows 窗体控件中，许多类本身都是其他控件的基类，但每组控件都有一组属性、方法和事件，用于特定的目的。当设计和修改 Windows 窗体应用程序的用户界面时，需要对控件进行添加、对齐和定位等操作。控件是包含在窗体对象内的实例，每种类型的控件都具有自己一些特定的属性集、方法和事件，从而实现与其他控件的区别^[22]。在使用 Windows 窗体控件时，开发者既可以在设计器界面中操作控件，也可以在编写代码时动态地添加控件。简单的说，控件是一种特殊的对象，是显示数据和接受数据输入的相对独立的用户界面元素。

2.2 C#编程

2.2.1 C#语言概念及优势

C#是可用于创建要运行在.NET CLR 上的应用程序的语言之一，具有语法简洁、面向对象设计、与 web 紧密结合、完整的安全性和错误处理、版本控制、兼容性及灵活性的特点。它是从 C 和 C++演化而来的，是 Microsoft 为使用.NET 平台而量身打造的，它综合考虑了其他各种编程语言的优缺点，在保留各种编程语言的优点的同时尽量解决其他编程语言存在的问题。因为 C#语法比较简单，所以使用 C#开发应用程序比使用 C++要简单一些，与此同时 C#是一种十分强大的编程语言，用 C++可以完成的任务几乎全部可以通过 C#完成。虽然一般情况下执行相同的任务的时候，用 C#编写的代码量通常比 C++要大一些，但是 C#代码更健壮，调试起来也更简单，.NET 始终可以随时的跟踪数据类型。C#虽然只是用于.NET 开发的一种语言，但是它是最好的一种语言，它是唯一一个彻头

彻尾为.NET Framework 设计的编程语言，是可以移植到其他操作系统上的.NET 版本中使用的主要语言，它能使用.NET Framework 代码库提供的每种功能。综上所述，本实验的开发选择使用 C#编程语言来完成^{[23] [24]}。

2.2.2 C#网络编程方法概述

计算机网络的 TCP/IP 模型将计算机网络通信定义为一个四层框架模型，分别为链路层、网络层、传输层及应用层，从名字上看，TCP/IP 包括传输控制协议和网际协议，但实际上指的是 Internet 协议系列，常称为 TCP/IP 协议族，具体模型结构及各层所包含的主要协议如图 2.3 所示^[25]。.NET 框架也为网络开发提供了两个顶层命名空间：System.Net 和 System.Web，C#通过命名空间中封装的类和方法实现网络通信编程、Web 应用编程以及 Web Service 编程，常使用的命名空间有：System.Net、System.Net.Sockets 及 System.Web，常使用的类有：IP 地址类、DNS 类及 Socket 类。

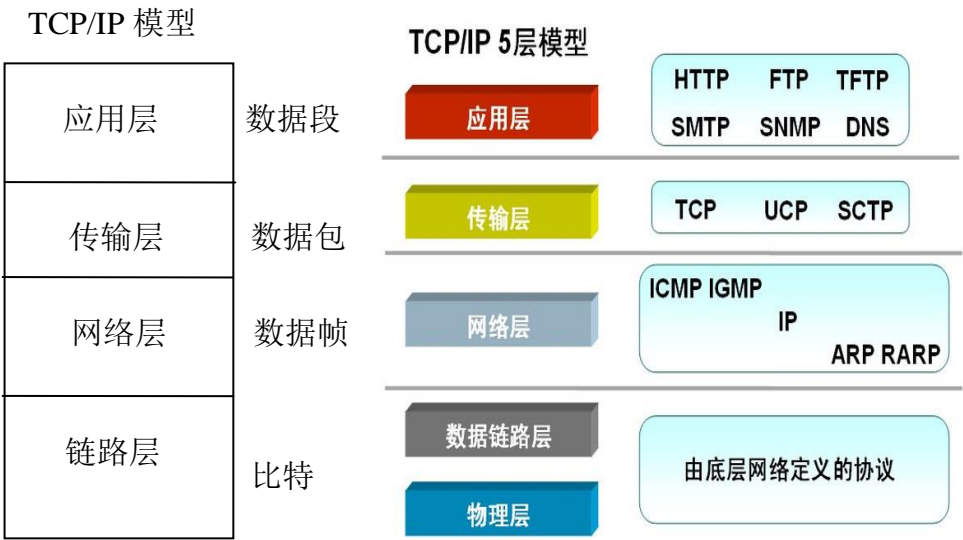


图 2.3 TCP/IP 模型结构（见左图） TCP/IP 协议族（见右图）

其中 Socket 套接字是通信的基石，是支持 TCP/IP 协议的网络通信的基本单元。可以将套接字看做不同主机间的进程进行双向通信的端点，它构成了单个主机内及整个网络间的编程界面。套接字同常和一个域中的套接字交换数据，各种进程使用这个相同的域互相之间通过 Internet 协议族来进行通信，具体关系如图 2.4 所示。因此在 Windows 编程中使用套接字来解决网络中的数据传输问题^[26]。

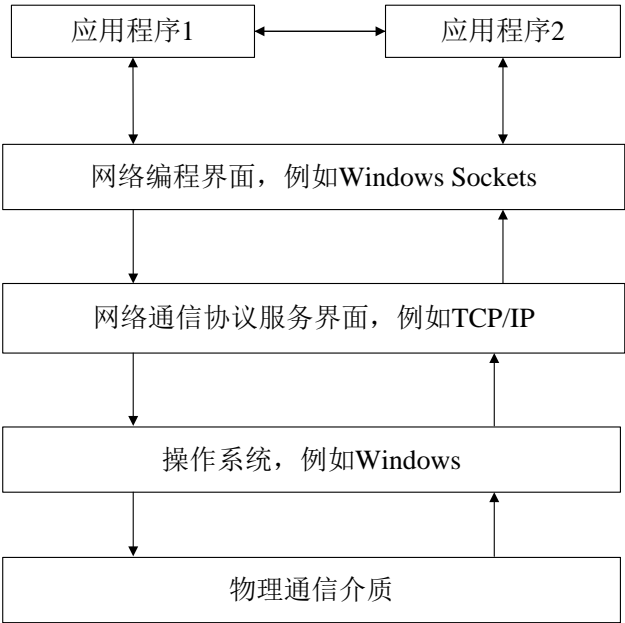


图 2.4 应用程序与 Windows Sockets 关系图

2.2.3 C#Delegate 编程

Delegate 是在 .Net1.0 版本中就已经存在的特性，Delegate 中文翻译为“委托”。MSDN 中对 Delegate 的解释是：C# 中的委托类似于 C 或 C++ 中的函数指针。使用委托使程序员可以将方法引用封装在委托对象内。然后可以将该委托对象传递给可调用所引用方法的代码，而不必在编译时知道将调用哪个方法。与 C 或 C++ 中的函数指针不同，委托是面向对象、类型安全的。我们经常在设计模式中的观察者模式（观察者模式又称 Source/Listener 模式）中使用 Delegate，其实在需要使用回调函数时都可以考虑使用 Delegate，譬如：子线程调回主线程函数^[27]。

2.3 C/S 结构

C/S 结构，即 Client/Server（客户机/服务器）结构，一般采用两层结构，结构图如图 2.5 所示，它由两部分构成：前端是客户机，通常是 PC；后端是服务器，运行数据库管理系统，提供数据库的查询和管理。

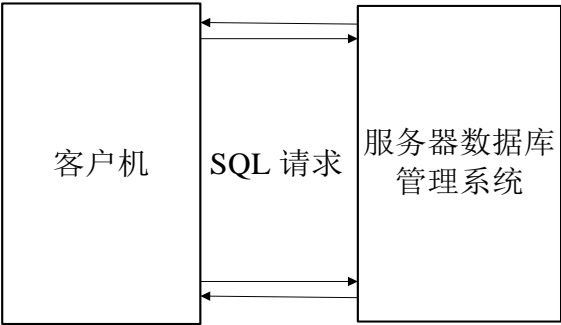


图 2.5 两层结构 C/S

现在 C/S 结构建立在中间件产品上，是一种三层结构，核心概念是利用中间件将应用分为表示层、业务逻辑层和数据存储层三个不同的层次，层级结构图如图 2.6 所示，三个层次的划分是从逻辑上来分的，具体的物理分发有很多种组合。中间层作为构造三层结构应用系统的基础平台，提供了以下主要功能：负责客户机与服务器之间、服务器与服务器之间的连接和通信；实现应用与数据库的高效连接；提供一个三层结构应用的开发、运行、部署和管理的平台。

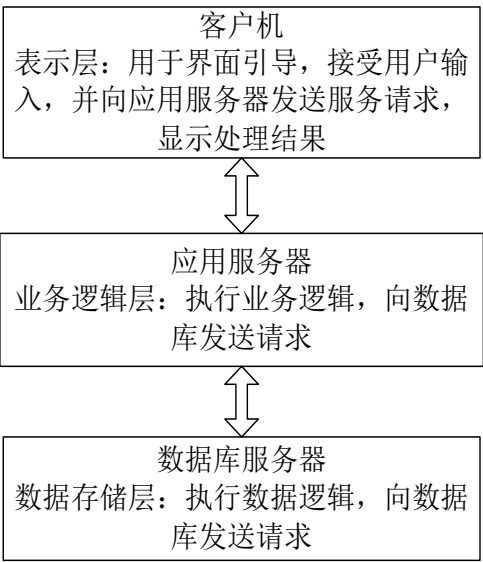


图 2.6 三层结构 C/S

C/S 结构具有强壮数据操纵和事务处理能力，并且具有数据的安全性和完整性约束，由于本文所研发的系统对数据安全性、系统稳定性、响应速度有较高的要求，所以本文选择使用 C/S 结构。

2.4 SQL SERVER 2008

数据库是数据存放的地方，是需要长期存放在计算机内的有组织并且可共享的数据集合。数据库中数据按照一定的数据模型组织、描述和存储，具有较小的冗余度，具有较高的数据独立性和易扩展性，可以让不同的用户共享使用。

常见的数据库模型有：层次模型、网状模型和关系模型 3 种，关系模型是一种重要的数据库数据模型，是最有效率的数据库组织方式之一，它建立在集合论坚固的数学基础之上，是集合论在数学组织领域的应用，对于权限控制，并发控制有着很好的应用。而由 Microsoft 发布的 SQL SERVER 产品是一个典型的关系型数据库管理系统，以其功能的强大性、操作的简便性、安全的可靠性，得到了广泛的应用，特别是新发布的 SQL SERVER 2008，它在 SQL SERVER 2005 的强大功能之上，为用户提供了一个更完整的数据管理和分析解决方案^[28]。经过综合考虑，本实验选择使用 SQL SERVER 2008 来配合实现并发控制算法。

综合以上所述，本文研究内容的实质是在.NET 开发平台下，将网络编程技术、C#编程语言和数据库技术相结合，设计并实现的一个 Windows 窗体控件应用程序。

2.5 本章小结

本章介绍了本文开发所涉及到的部分关键性技术，主要包括：.NET Framework 开发框架、Windows 窗体控件、C#编程知识（C#网络编程、Delegate 编程）以及 SQL SERVER 2008。通过介绍相应技术的优缺点及其用途，并与其他同类技术进行了多方面的比较，结合实际需求，说明了本文选定相应开发技术的原因。最后对本文研究内容进行了技术层面的概括：实时同步虚拟协同实验是在.NET 平台下，利用网络编程技术、C#编程语言及 Sql Server 数据库设计并实现的一个 Windows 窗体控件应用程序。

第三章 实时同步虚拟协同实验的设计方案

本章以一个具体的基础虚拟组网实验为例简要介绍了单人虚拟组网实验平台，阐述了实验的过程和特点，由此提出了实时同步虚拟协同实验的设计方案。

3.1 单人虚拟组网实验平台概述

单人虚拟组网实验平台主要开发了针对思科、华三、神州数码和锐捷等网络设备为原型的一系列基础组网实验^[29]。以该平台中 RIP 实验为例，简要的介绍单人虚拟组网实验平台以及实验的过程，单人虚拟组网实验平台见图 3.1。

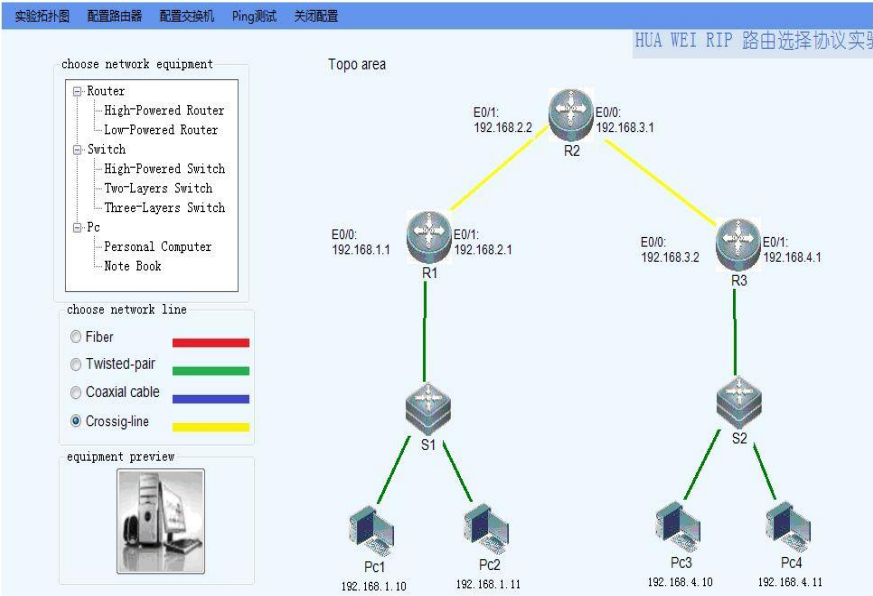


图 3.1 单人虚拟组网实验平台截图

由图 3.1 可见，单人虚拟组网实验包括四个区域：设备选择区（choose network equipment）、连线选择区（choose network line）、设备预览区（equipment preview）以及拓扑展示区（Topo area）。实验主要有四个操作过程^[15]：

第一步：在设备选择区通过点击网络设备图标，便可在设备预览区查看自己选择的网络设备缩略图。双击设备预览区，每双击一次拓扑展示区就会增加一个网络设备图标。

第二步：选择完你需要的所有网络设备之后，在连线选择区选择网络设备之间需要的连线，右键单击网络设备图标，选择相应端口进行设备的连接。

第三步：设备连接后，同样右键单击拓扑展示区的 PC 机图标，为其配置 IP 地址，网络设备的配置是通过双击拓扑展示区的图标来实现的，设备配置界面的截图如图 3.2 所示。

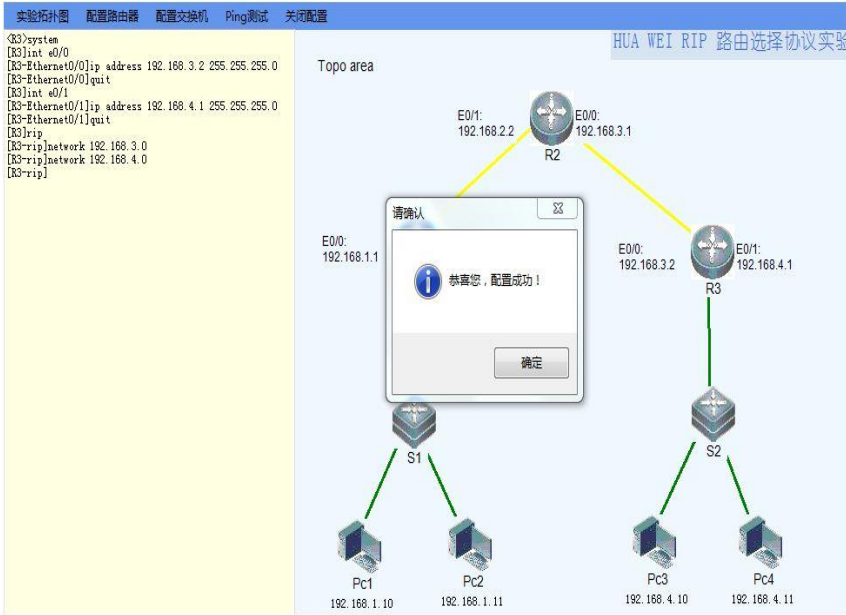


图 3.2 设备配置界面截图

第四步：设备配置完毕后连通性的测试是通过双击 PC 机图标，书写 ping 命令实现的。

单人虚拟组网实验使学生可以在自己的 PC 机上，随时进行计算机网络实验的练习，但在实际的应用还存在着一定的不足：（1）单人实验没有交互性，且相同网络设备的配置几乎相同，单人操作导致实验效率降低，对于大型的网络实验尤为明显。（2）单人虚拟组网实验的实验拓扑单一，用户只知道本实验的拓扑结构，了解不到整个网络结构，也学习不到如何自主划分网络拓扑的相关知识，从而导致学生无法完整理解计算机网络知识。（3）网络拓扑的搭建完全通过拖拉的方式实现，与现实计算机网络实验操作不符。

3.2 虚拟协同实验平台的设计目标

针对上节所提出的单人虚拟组网实验的几个不足，并结合实际的实验要求和网络实验室条件，提出了虚拟协同实验平台的设计目标：

- （1）同一项组网实验以及网络拓扑的搭建可以以小组为单位共同完成，以节省单人重复配置网络设备的时间和精力，同时也培养学生的团队合作意识。
- （2）共同完成实验的过程中，所有同组成员可以同时操作，并且每个组员的操作结果以及整体的网络拓扑结构都会实时的显示在每个组员的实验平台上。
- （3）每个组员在进入实验平台后可以随意选择网络设备，被其他组员占用的设备除外，实现设备占用的并发控制。

- (4) 每个实验组可以按照自己的意愿通过配置交换机,划分 vlan 来获取实验需要的网络结构,使得虚拟实验过程与真实的组网实验过程完全相符,也使得学生更加清楚整个机柜的网络结构,提高学生认知真实网络结构的能力。

3.3 虚拟协同实验平台的功能设计

虚拟协同实验平台是以单人虚拟组网实验为基础,但是在界面布局和实验操作步骤方面都有自己的定义,结合 3.2 节虚拟协同实验平台设计目标,重新进行界面设计和后台代码编写,实现具有多人共同搭建实验拓扑,并发选择控制和实时同步实验操作等功能的虚拟协同实验平台。平台的功能设计方案如下:

(1) 实时同步实验操作

虚拟协同实验平台的重点就是协同,现如今研究的虚拟协同实验平台都是采用异步的方式实现协同,而本文的协同不仅可以满足组员共同合作完成一项实验的需求,更重要的是在同步操作的前提下完成的。实验平台大致分为拓扑显示区、设备配置区以及配置查看区。在实验的过程中,任何组员的操作都会实时的同步到其他组员的实验平台上,组员可以查看网络结构中任意网络设备现在的配置情况,进而了解整个实验的进展情况。

(2) 同步搭建实验拓扑

网络结构的构建是组网实验的学习重点之一,单人虚拟实验只有单一固定的网络结构,学生在搭建网络拓扑过程中只是死板的背诵网络结构,无法自主的在整个网络结构中选择自己的网络搭建方案。本文给出了小组实验中整体机柜的网络图,允许小组成员以划分 vlan 的方式得到自己需要的网络结构,并且同组成员可以同时协同搭建网络结构,每个人的搭建结果都会实时展现在所有组员的界面上,虚拟协同实验也就真正的实现了实时同步。

(3) 并发控制机制

虚拟协同实验要实现多人同时操作不同的网络设备,协作完成同一项实验,那么设备的占用或者说实验平台的并发控制便成为了虚拟协同实验的重中之重。本文设计的虚拟协同实验支持某个组员对某个网络设备进行配置的同时,其他组员对于此网络设备只有查看配置信息的权限,不可以进行配置,并且组员对于网络设备的占用以时间为判断基准,时间越早优先级越高,优先级最高者可以执行配置操作。

3.4 本章小结

本文详细介绍了单人虚拟组网实验平台、单人虚拟实验的实验流程以及单人

虚拟组网实验平台的缺陷，并在此基础上提出了虚拟协同实验平台的建设目标，以及对于实时同步实验操作、共同搭建实验拓扑、并发控制机制这三个关键问题的设计方案。

第四章 实时同步虚拟协同实验的实现

本章主要从实时同步实验操作、共同搭建实验拓扑、并发控制机制等几个关键点，设计并实现了实时同步虚拟协同实验平台，并通过代码、表格、算法流程图及实验截图详细的阐述了实验平台的实现过程。

4.1 实时同步实验操作

目前，虚拟协同实验采用的是异步的方式实现共同完成同一项实验的要求，所以没有涉及到在组员平台上实时同步实验操作的技术难题。本文设计的虚拟协同实验平台允许学生共同操作，学生的操作信息会通过网络编程技术传送到同组的其他组员那里，其他组员通过委托技术将接收到的操作信息显示到自己的操作界面上。

4.1.1 真实实验过程中的实时同步协同

本文设计的实时同步虚拟协同实验平台上的网络设备是以本院网络通信实验室中的设备为原型的。实验室中包括 H3C 和华为两种机柜，本文网络设备模拟的是 H3C 机柜中的设备，本文模拟的 H3C 机柜物理结构如图 4.1 所示，机柜从上至下，分别是三台路由器、三台交换机（其中最下面的一台为三层交换机）以及两个配线架。网络设备连接到配线架上，学生通过插拔配线架来控制网络设备之间的连通情况，配线架的使用是为了防止学生经常插拔网络设备接口导致接口的老化与损坏。与物理结构相对应的是机柜的网络结构，通过配线架的转接，路由器和交换机之间形成的网络拓扑图如图 4.2，其中二层交换机和三层交换机 sw1、sw2 和 sw3 端口标识均为三字段 e0/1/x，x:1~24，路由器连接交换机的接口为 GigabitEthernet 0/0 和 GigabitEthernet 0/1，可简写为 g0/0 和 g0/1，机柜的交换机 SW1 的 1、5 号口与 SW2 的 3、7 号口分别连接了四台计算机，学生可以通过华为开发的应用软件操作机柜的设备。

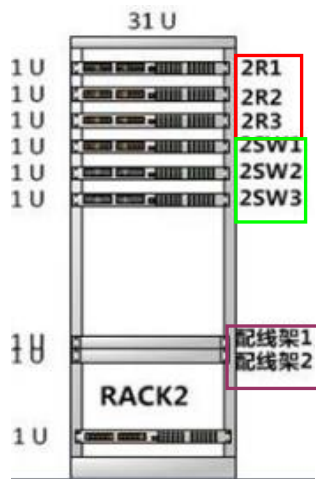


图 4.1 H3C 机柜图

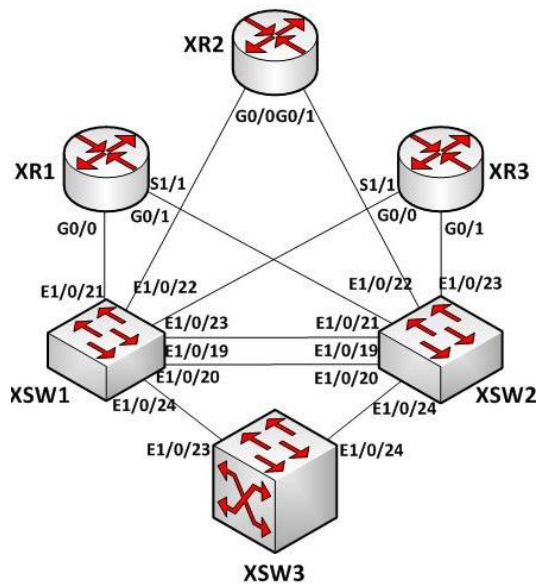


图 4.2 H3C 机柜网络拓扑图

学生进行计算机网络课程学习时，除了基本知识的理解之外，还需要到网络通信实验室进行硬件设备的操作与练习，网络通信实验室中学生是通过基于 B/S 架构的 NEMS 平台来完成实验的操作的。学生四人划分为一个小组，使用一个机柜的设备，自由选择 PC 机准备操作网络设备。学生首先要登录 NEMS 平台的网址，在登录以后学生可以在网站上看到实验拓扑图、实验原理等，并通过登录设备操作进入硬件设备的配置窗口，登入设备界面如图 4.3，进入硬件设备的配置窗口后学生便根据实验要求利用配置代码的形式完成自己那台设备的应完成的操作，当本组所有学生都完成自己控制的硬件设备的配置时，就可以用实验要求的验证方法，验证机柜设备整个实验的完成效果。由于是真实硬件设备，任何配置都会反应到设备上，任何一个组员只需要登录一台没有被占用的设备，利用查看命令就可以查看设备当前的配置情况。

实验设备列表					
设备管理控制台	接口	设备标签	设备名称	设备类型	实验操作
DMC3	Async1/0	MSR30-20	MSR30-20	路由器	登录设备 监控实验
DMC3	Async1/1	MSR30-20	MSR30-20	路由器	登录设备 监控实验
DMC3	Async1/2	MSR30-20	MSR30-20	路由器	登录设备 监控实验

图 4.3 登入设备界面图

4.1.2 实时同步实验操作的实质

对应于实际实验的实验设备，本课题中的单人虚拟组网实验主要有三种网络设备：路由器、交换机和计算机终端。其中，路由器承担了网络数据包的交换工作，是整个组网实验的核心，而交换机和计算机终端只是起到一个验证网络连通性的作用^[15]。本文设计的虚拟协同实验为了能够严格的对真实协同实验进行模拟，在实验平台上设计了三台路由器、三台交换机、四台计算机，即是一个 H3C 机柜的网络设备，也是实际实验中一个小组成员可以操作的所有设备。但是本文设计的交换机不仅可以用来验证网络连通性，还可以和现实中的交换机一样划分局域网^[16]，这是和单人虚拟组网实验不同的。

在单人虚拟组网中实验设备的选择、实验拓扑的搭建都是采用拖拉、点击控件的方式实现的，本文设计的虚拟协同实验完全模拟真实的实验流程，整个过程中实验拓扑的搭建、网络协议的配置、连通性的测试都是通过对设备的配置来实现的，所以本文中提到的实验操作实际上是对设备进行配置，设备配置的效果图如图 4.4 所示。

```
配置窗口
<SW1>sys
[SW1]int E1/0/22
[SW1-Ethernet1/0/22]undo shutdown
[SW1-Ethernet1/0/22]quit
[SW1]int E1/0/23
[SW1-Ethernet1/0/23]undo shutdown
[SW1-Ethernet1/0/23]quit
[SW1]vlan 2
[SW1-vlan2]port E1/0/22 to E1/0/23
[SW1-vlan2]quit
[SW1]int E1/0/21
[SW1-Ethernet1/0/21]undo shutdown
[SW1-Ethernet1/0/21]quit
[SW1]int E1/0/1
[SW1-Ethernet1/0/1]undo shutdown
[SW1-Ethernet1/0/1]quit
[SW1]vlan 1
[SW1-vlan1]port E1/0/1
[SW1-vlan1]port E1/0/21
[SW1-vlan1]
```

图 4.4 设备配置效果图

本文对设备的配置方式是双击设备图标，出现对应配置框后输入配置信息，得到相应的设备反应，完成配置。实现方式如下：

- (1) 使用 PictureBox 控件的 initialimage 属性显示设备图标，设备图标分别有路

由器、交换机和 PC 机三种类型，因为平台有三台路由器、三台交换机及四台 PC，则设备图标名称分别以后缀 lr1、lr2、lr3、ts1、ts2、ts3、pc1、pc2、pc3、pc4 来区分。

(2) 使用 Textbox 作为设备的配置框及查看配置的文本框，对应于不同的设备有不同的配置框和查看配置的文本框，配置和查看配置的文本框分别以后缀 confi 和 checki 来区分，其中 i 与设备相对应，对应关系如表 4-1 所示，

表 4-1 i 与网络设备之间的关系

i	1	2	3	4	5	6	7	8	9	10
i 代表的设备	lr1	lr2	lr3	ts1	ts2	ts3	pc1	pc2	pc3	pc4

则 textBox_confi 和 textBox_checki 便为表 4-1 中 i 对应网络设备的配置框和查看框。

(3) 对设备图标 PictureBox_x 进行双击，按照表 4-1 的逆映射关系，找到设备 x 对应的 i 值，显示 textBox_confi，隐藏其余配置文本框，从而将设备图标与配置或查看文本框一一对应。学生双击设备图标，触发的是 PictureBox 控件的 DoubleClick()事件，事件的主要实现语句如下所示：

```
private void pictureBox_lr1_DoubleClick(object sender, EventArgs e)
{
    textBox_conf1.Visible = true;
    textBox_conf2.Visible = false;
    textBox_conf3.Visible = false;
    textBox_conf4.Visible = false;
    textBox_conf5.Visible = false;
    .....
}
```

综上所述，实时同步实验操作的实质是实时同步设备的配置信息，而设备的所有配置信息都存在于对应的 Textbox 配置文本框中，所以实时同步 Textbox 框中键入的配置信息成为了实现实时同步实验操作的关键。

4.1.3 实时同步实验操作的设计

简单地说，套接字就是不同的计算机之间为了满足各自进程间通信的需要所架设的一条数据通道，是一个通信链的句柄。具体来讲，套接字出现的动机是：应用层通过传输层进行数据通信时，TCP 和 UDP 会遇到同时为多个应用程序进程提供并转发服务的问题。多个 TCP 连接或多个应用程序进程可能需要通过同一个 TCP 协议端口传输数据。为了区别不同的应用程序进程和连接，许多计算机操作系统为应用程序与 TCP/IP 协议交互提供了称为套接字（Socket）的接口，用以区分不同应用程序进程间的网络通信和连接。生成套接字主要有 3 个参数：

通信的目的 IP 地址、使用的传输层协议（TCP 或 UDP）和使用的端口号。把这个 3 个参数结合起来并与 Socket 绑定，应用层就可以和传输层通过套接字接口，区分来自不同应用程序进程或网路连接的通信，实现数据传输的并发服务^[25]。

C#中的套接字类的原型是：

```
System.Net.Sockets.Socket
Public Socket(AddressFamily addressFamily, SocketType socketType,
               ProtocolType protocolType)
```

在此构造函数中，SocketType 和 ProtocolType 要自己根据实际情况进行对应，对应关系如表 4-2。

表 4-2 套接字类型与协议对应关系

套接字类型	协议对应关系
Dgram(UDP)	无连接通信
Stream(TCP)	面向连接的通信
Raw(ICMP)	ICMP 协议
Raw(Raw)	简单 IP 包

.NET 中的 Socket 提供了下列三种类型套接字：

- (1) 流式套接字（Stream）提供了一个面向连接、可靠的数据传输服务，数据无差错、无重复的发送，而且按发送顺序接收。内设流量控制，避免数据流超限；数据被看做字节流，无长度限制。流非常适合处理大数据量，文件传送协议（FTP）就是使用流式套接字实现的。
- (2) 数据报式套接字（Dgram）提供了一个无连接服务。数据包以独立包形式被发送，不提供无错保证，数据可能丢失或重复，并且接收顺序混乱。网络文件系统（NFS）使用的是数据报式套接字。
- (3) 原始式套接字（Raw）允许对较低层协议（如 IP、ICMP）直接访问。常用于检验新的协议实现或访问现有服务中配置的新设备。

其中，数据报式套接字是以 UDP 协议为基础的，由于 UDP 是一种无连接的通信协议，因此消耗资源小，数据传输的速度得到大幅度的提高，因此，通常视频、音频和普通数据的传输基本上使用的都是 UDP，即使偶尔丢失一两个数据包，也不会对接收效果产生很大的影响，平时我们使用的聊天工具 QQ 使用的就是这种具有实时特点的 UDP 协议完成的^[26]，面向无连接的套接字并不需要发送连接信息，它的工作流程图如图 4.5 所示，有如下几个重要步骤：

- (1) 通过套接字的构造函数方法创建套接字对象，采用 bind（）方法对本地地址和套接字绑定，此处需要强调的是，只有正确的完成了绑定，才能够进行下面的数据传送。
- (2) 绑定完成之后即开始传输数据，无连接方式采用 SendTo 方法和

ReceiveFrom 方法，因为无连接方式不进行连接，所以一定要指定目标主机的地址。

(3) 最后只要释放连接，关闭套接字即可。

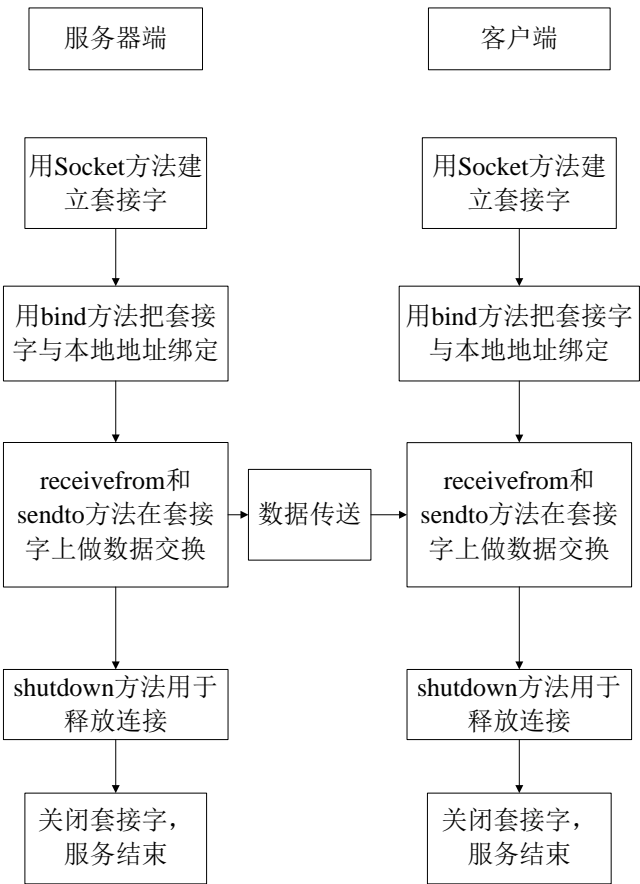


图 4.5 面向无连接的套接字工作流程图

本文研究虚拟实验平台的重点是实时同步配置信息，所以对配置信息的快速传输具有较高的要求，综上所述，本文采用面向无连接的套接字来实现配置信息的同步功能。

4.1.4 实时同步实验操作的实现

本文用面向无连接的套接字发送和接收其他组员对实验设备的配置信息，由于本文采用 C/S 模式，所以没有严格的服务器端和客户端之分，因此组员相互之间可以直接发送和接收配置信息。实时同步实验操作功能的实现主要分为实时发送和实时接收两个部分，下面将分别详细介绍各自的具体实现过程：

(1) 实时发送：

本文主要利用配置文本框 Textbox 的 KeyPress() 事件来确保发送的实时性。学生一旦在配置框中键入配置信息，便触发了文本框的 KeyPress() 事件。KeyPress() 事件是当用户按下和松开一个 ANSI 键时才会触发，具有焦点的

对象接收该事件,在截取 TextBox 或 ComboBox 控件所输入的击键时是非常有用的,它可立即测试击键的有效性或在字符输入时对其进行格式处理。本文的 KeyPress () 事件中主要有两个功能:第一,根据本地用户的配置情况模拟真实设备给出反应;第二,将本地学生控制的设备配置信息实时的发送给同组的其他同学。

对于功能一,本文在 KeyPress () 事件中使用参数 e 的 e.KeyChar () 属性获取与按下的键对应的字符,其中 enter 键的键值为 13,退格键的键值为 8。此外,从 TextBox 控件文本内的开始字符位置检索到输入的行号 i,主要实现方法是在 KeyPress () 中添加以下语句:

```
string a;  
a =  
(textBox_conf1.GetLineFromCharIndex(textBox_conf1.SelectionStart)).ToString();  
int i = int.Parse(a);
```

在获取到配置的行号 i 及按下的键值后,根据学生配置的内容使用选择语句 (if-else),按照交换机和路由器配置响应流程(见图 4.6)做出相应的反应。设备配置响应流程的核心是:根据 textBox_conf.Lines[i]的内容按照实际设备应有的反应执行相应的操作,这也严格模拟实际实验流程。

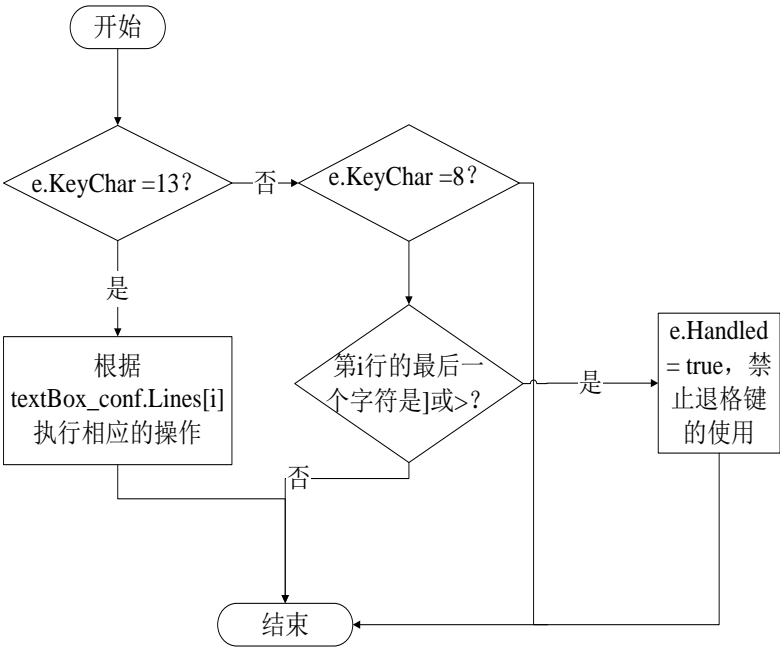


图 4.6 网络设备配置响应流程

本文中学生主要用到划分 vlan、设置端口 IP 地址、添加路由协议等配置命令,根据这些配置命令虚拟网络设备会给出与真实设备相似的反应,这也是图 4.6 根据 textBox_conf.Lines[i]执行相应操作的依据,涉及到的主要网络设备命令如表 4-3 所示。

表 4-3 网络设备命令

命令	说明
system	进入系统视图模式
undo shutdown	打开以太网端口
int ethernet (0/1/2)	选择配置的端口
int g0/0	g0/0 接口的 Mac 地址
vlan	虚拟局域网
port	端口
ip address	设置设备 IP
rip	内部网关协议
ping	Send echo messages.

对于功能二，本文在 `KeyPress()` 事件中，添加了 `conf_Click()` 事件，该事件目的是为了获取设备现阶段的配置信息，并触发配置信息的发送事件。学生每配置完一行按下 `enter` 键的同时，后台就会进行判断并根据配置内容给出反应，然后触发 `conf_Click()` 事件，类似于 `KeyPress()` 事件，`conf_Click()` 事件使用同一个方法获取本配置行 `textBox_conf.Lines[i]` 的内容，并触发 `sendData()` 函数将本配置行内容发送至其他同组成员。

`sendData()` 函数获取到配置行内容后，使用 C# 中的 `Socket` 类向其他组成员传输配置行内容，`Socket` 类常用属性和方法 UML 图如图 4.7 所示。

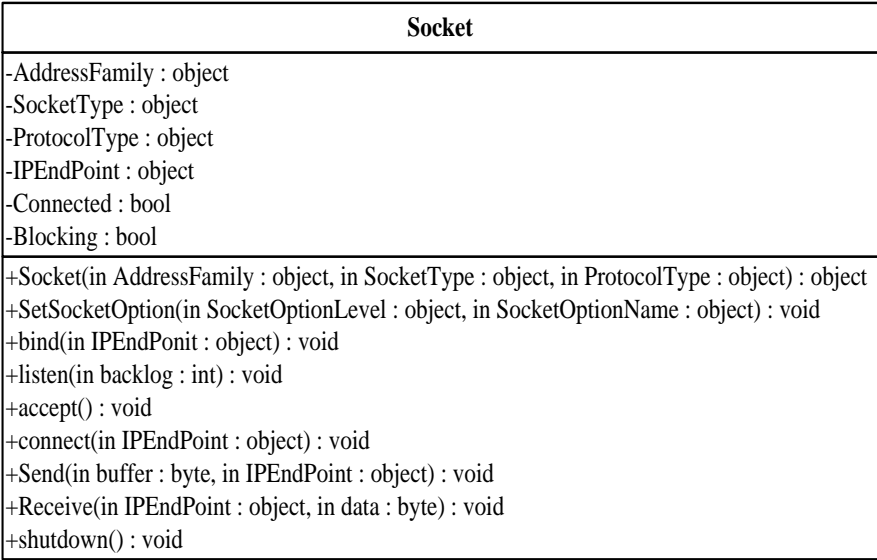


图 4.7 Socket 类 UML 图

在 4.1.3 节中详细介绍了无连接套接字（`Socket`）的工作流程，结合图 4.5 可知实时发送的实现流程是：首先使用 `Socket` 构造函数声明定义一个 `Socket` 类，鉴于本文将发送和接收信息端口设置为同一个，所以设置 `Socket` 为端口可复用，之后将 `Socket` 与网络端点绑定，开始发送配置信息，发送完毕后关闭 `Socket`。

对于绑定的网络端点，本文使用 `EndPoint` 类将网络端点表示为 IP 地址和端口号。`EndPoint` 类对象表示指定的 IP 地址和端口的组合^[30]。IP 地址采用的是广播地址，可以将配置信息广播给局域网内其他人员。端口采用临时端口，计算机网络中任何 TCP/IP 实现所提供的服务都用知名的 1~1023 之间的端口号，这些知名端口号由 Internet 号分配机构（Internet Assigned Numbers Authority）来管理，大多数 TCP/IP 实现给临时端口分配 1024 到 65535 之间的端口号^[31]。本文选择临时端口号 9050 用于发送配置信息。实时发送部分实现如下：

```
Socket sock = new Socket(AddressFamily.InterNetwork, SocketType.Dgram,
    ProtocolType.Udp);//初始化一个Socket实例，采用UDP传输
sock.SetSocketOption(SocketOptionLevel.Socket,
    SocketOptionName.ReuseAddress, true);
EndPoint iep= new EndPoint(IPAddress.Broadcast, 9050);
byte[] buffer = Encoding.UTF8.GetBytes(sendmessage);//编码
sock.SetSocketOption(SocketOptionLevel.Socket, SocketOptionName.Broadcast,
    1);
sock.SendTo(buffer, iep);
sock.Close();
```

综上所述，本文使用 `KeyPress()` 事件完成设备实时智能反应，结合 `SendData()` 函数实现配置信息在局域网内的实时发送。

(2) 实时接收：

本文通过使用线程结合无连接套接字技术来实现信息的实时接收功能。本文在后台开启一个线程在实验过程中始终监听并获取实时发送来的局域网内配置信息。

线程是程序执行的最小单元，C#中使用 `Thread` 类来创建线程，MSDN中对 `Thread` 类的解释是创建并控制线程，设置其优先级并获取其状态^[32]。`Thread` 类有几个至关重要的方法，如表4-4所示，

表 4-4 Thread 类的主要方法

方法	说明
<code>Start()</code>	启动线程
<code>Sleep(int)</code>	静态方法，暂停当前线程指定的毫秒数
<code>Abort()</code>	通常使用该方法来终止一个线程
<code>Suspend()</code>	该方法并不终止未完成的线程，它仅仅挂起线程，以后还可恢复
<code>Resume()</code>	恢复被 <code>Suspend()</code> 方法挂起的线程的执行

C#中初始化的 `Thread` 实例，通过 `ThreadStart` 委托来指明方法从何处开始运行，

Thread实例调用Start方法后，线程开始运行，线程一直到它所调用的方法返回后结束。

本文为了在学生实验过程及时的获取网络中传来的配置信息，在学生进入小组实验界面的同时，便开启线程开始监听。具体实现是在platform_Load（）函数中添加以下语句：

```
public static Thread td;
td = new Thread(new ThreadStart(this.StartListen));
td.Start();
```

本文中td实例从StartListen（）函数开始运行，StartListen（）函数中定义了一个Socket实例，此实例可以复用9050端口，因此Socket实例绑定的网络端点IP为局域网内任意IP，端口为9050端口。StartListen（）函数使用while(true)死循环利用Socket实例的ReceiveFrom（）方法，对9050端口传递过来的信息持续性的接收，实时接收部分的实现代码如下：

```
Socket socket = new Socket(AddressFamily.InterNetwork, SocketType.Dgram,
    ProtocolType.Udp);
socket.SetSocketOption(SocketOptionLevel.Socket,
    SocketOptionName.ReuseAddress, true);
EndPoint iep = new IPEndPoint(IPAddress.Any, 9050);
socket.Bind(iep);//绑定这个实例
EndPoint ep = (EndPoint)iep;
while (true)
{
    byte[] data = new byte[1024];
    int recv = socket.ReceiveFrom(data, ref ep);
    string GetMessage = Encoding.UTF8.GetString(data, 0, recv);
    .....
}
```

在学生进入实验组实验界面的同时，便开启了负责监听配置信息的线程，一旦有新的配置信息发送过来，便会被马上接收，从而实现了配置信息的实时接收。

4.1.5 异地实验操作的实时显示

客户端在9050端口接收到其他客户端发送来的配置信息后，将其存放在一个字节数组中，由于配置信息仅仅保存在后台，所以这个时候在本客户端界面查看配置Textbox上并不能显示出其他组员的配置信息，并且C#中禁止跨线程直接访问控件，那么上节中提出的通过另外添加监听线程实现实时接受配置信息，把查

看配置的Textbox 设置为Windows窗体的主线程的方法就无法直接将配置信息写到查看配置的Textbox中了。

针对出现的跨线程访问控件的问题，本文提出了通过Invoke方法与控件的Control.InvokeRequired属性相结合的解决方案。

MSDN中对于Control.InvokeRequired属性的解释是：获取一个值，该值指示调用方在对控件进行方法调用时是否必须调用Invoke方法。Windows窗体中的控件被绑定到特定的线程，不具备线程安全性。因此，如果从另一个线程调用控件的方法，那么必须使用控件的一个Invoke方法来将调用封送到适当的线程，该属性可用于确定是否一定要调用 Invoke 方法。如果不存在控件句柄，则InvokeRequired就要沿控件的父级链进行搜索，一直到它搜索到有窗口句柄的控件或窗体为止。假如找不到适合的句柄，InvokeRequired方法就会返回false。这意味着如果不需要Invoke（调用发生在同一线程上），或者如果控件是在另一个线程上创建的但尚未创建控件的句柄，则InvokeRequired可以返回false。当一个控件InvokeRequired属性值为true时，说明有一个创建它以外的线程想访问它，此时它将会在内部调用Invoke方法来完成访问，这个做法保证了控件的安全^[33]。而Control.Invoke方法是在拥有此控件的基础窗口句柄的线程上执行委托（详见2.2.3节）[34]。在本文2.2.3节中对委托做了相应的介绍，委托是一种将引用封装成匿名函数或命名的类型，委托的声明制定了一个函数签名，包括一个返回类型和参数列表，声明时要用到关键字delegate，然后声明委托类型的函数变量，初始化之后，使用委托变量调用这个函数。

本文StartListen（）实时接收到配置信息，做出本地客户端的后台逻辑处理后，根据配置信息调用相应的prif函数，判断是否在同一线程中，如果在同一线程中则将配置信息添加到相应的TextBox中；否则将调用Invoke函数唤醒主线程，使用委托技术将配置信息添加到相应的TextBox中。主要实现代码如下：

```
delegate void del(string t);
private void prif1(string t1)
{
    if (!InvokeRequired)
    {
        textBox_check1.AppendText(t1);
    }
    else
    {
        del dele = new del(prif1);
        Invoke(dele, new object[] { t1 });//执行唤醒操作
    }
}
```

```
    }
}
```

至此，本小节利用Invoke函数，用委托将实时接收到的配置信息写到窗体主线程中的可以查看配置信息的Textbox中，其他组员对设备的配置信息将会实时的显示到客户端界面上。

4.2 同步搭建实验拓扑

单人虚拟组网实验中实验拓扑采用的是拖拉控件、点线连接的动态方式实现的，但是如果把这种方式用在搭建多人虚拟组网实验中，实验拓扑的同步操作将会很难实现。本文在 4.1.2 节中讲到实时同步实验操作的本质就是实时同步设备的配置信息，这也说明搭建实验拓扑这一实验操作也是完全通过实验设备的配置实现的。事实上，这也与实际实验的流程完全吻合，实际实验中学生是通过配置交换机来实现本组实验拓扑划分的。本文中交换机的配置可以通过 4.1 节的网络通信技术进行传递，每个组员客户端读取到交换机配置信息后，在后台做出逻辑分析处理，将处理结果显示到前端界面上，从而达到同步搭建实验拓扑的目的。

4.2.1 实际实验拓扑的同步搭建

实际小组实验中学生在划分小组，登录 NEMS 平台选择自己要操作的网络设备之后，不需要对设备进行连线的插拔，只需要配置图 4.2 中的二层交换机，划分 vlan 就可以实现小组网络结构。

vlan 是当前构建区域网普遍采用技术，被称之为“虚拟局域网”，传统局域网组网方式是以交换机为主连接设备，并且基于物理网络的布局来搭建网络平台，例如与同台交换机相连的用户的所有主机会组成一个小型的局域网，这些主机在同个广播区域内。而虚拟局域网是在逻辑上搭建网络结构的一种方式，可把同台交换机上的不同用户划分在不同虚拟局域网中，逻辑上它们就像连接在几台互不相连的交换机上一样，当然，也可以把不同交换机上的用户划分在同一个虚拟局域网中，从逻辑上这些主机就像连接在一台交换机上一样。二层交换机只能搭建单一的广播域，这可能影响到网络整体传输性能，在使用虚拟局域网后，能够将网络分割成多个广播域，从而提高网络的传输性能^[35]。

vlan 在交换机上大致有 4 类实现方式：

(1) 基于端口划分虚拟局域网

这种划分vlan的方法是以以太网交换机端口作为判断条件的，属于同一vlan的端口可不连续，若有多台交换机，同一vlan可以跨越多台以太网交换机，目前根据端口划分是划分vlan使用最多的方法。

（2）基于MAC地址划分虚拟局域网

这种划分vlan的方法是以每个主机的MAC地址作为判断条件，对每个具有不同MAC地址的主机分别配置好各自所属的组。

（3）基于网络层划分虚拟局域网

这种划分vlan的方法是以每个主机的协议类型（若支持多协议）或网络层地址作为判断条件，虽然此划分方法以网络地址（如IP地址）作为判断条件，但是它并不是路由，与网络层的路由没有任何关系。

（4）根据IP组播划分虚拟局域网

实际上IP组播也是一种vlan的定义，它认为一个组播组就是一个vlan，这种划分方式将vlan扩大到了广域网^[36]。

实际进行网络通信实验时，学生使用的是基于端口的划分方式，H3C的二层交换机划分vlan的语句很简单：`vlan xx`，`port Ex/x/xx`这两个配置语句即可完成端口vlan的划分，进而完成小组网络结构的搭建。

本文将实际搭建实验拓扑的方式虚拟化，小组成员通过配置交换机划分网络拓扑，后台经过一定的逻辑处理之后，将划分的网络拓扑实时显示在本机和同组其他组员的客户端界面上。

4.2.2 虚拟拓扑的设计与实现

实际实验中设备的连接是不需要通过插拔线来完成的，网络设备的端口本身是连接好的，只是因为设备的端口全部关闭，所以导致网络设备没有连通，学生只需要用配置命令打开想要打开的设备端口即可。虚拟协同实验为了模拟真实实验过程，在学生进入实验组的虚拟实验平台时，机柜的网络拓扑便展现在学生面前^[37]。图 4.2 展示的是实际实验的机柜网络拓扑图，本文使用 `pictureBox` 及数组的关系模拟真实机柜网络拓扑图，虚拟的机柜网络拓扑图如图 4.8 所示。

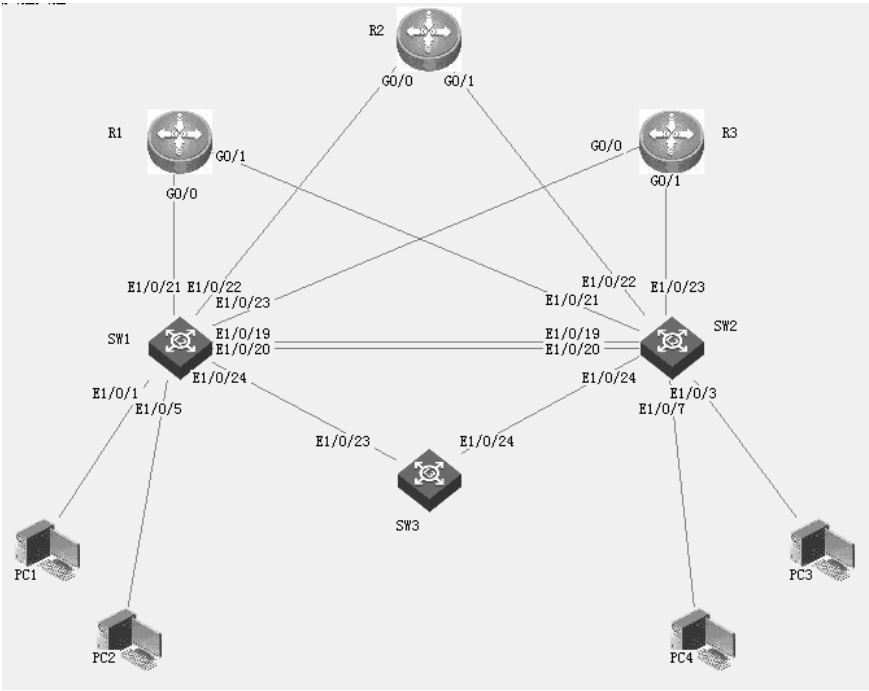


图 4.8 虚拟机柜网络拓扑图

图 4.8 网络拓扑图中有 3 台路由器、3 台交换机及 4 台计算机，机柜拓扑图是通过一维数组 Router_Switch 和二维数组 ConnectPoint 实现的。Router_Switch[i] 用来表示网络设备的编号，初始化的值见表 4-5，

表 4-5 Router_Switch 数组初始化

i	R1	R2	R3	S1	S2	S3	PC1	PC2	PC3	PC4
Router_Switch[i]	0	1	2	3	4	5	6	7	8	9

ConnectPoint[m,n]用来表示两个设备之间是否有连接线：0 代表设备间没有连接线，1 代表设备间有连接线。数组的横纵坐标值皆为 Router_Switch[i]的值，范围为：[0~9]，为了清晰地展现出设备之间连线的初始状态，本文利用函数的映射关系图，用表 4-6 表示出设备连线的初始化值。

表 4-6 设备连线的初始化值

i	R1	R2	R3	S1	S2	S3	PC1	PC2	PC3	PC4
Router_Switch[i]	0	1	2	3	4	5	6	7	8	9
R1	0	0	0	1	1	0	0	0	0	0
R2	1	0	0	1	1	0	0	0	0	0
R3	2	0	0	1	1	0	0	0	0	0
S1	3	1	1	1	1	1	1	1	0	0
S2	4	1	1	1	1	1	0	0	1	1
S3	5	0	0	1	1	0	0	0	0	0
PC1	6	0	0	1	0	0	0	0	0	0
PC2	7	0	0	1	0	0	0	0	0	0
PC3	8	0	0	0	1	0	0	0	0	0
PC4	9	0	0	0	1	0	0	0	0	0

与实际实验相同，本文的初始连线仅代表设备之间有双绞线连接，但是设备的端口并没有打开，这也意味着实际上设备之间并没有连通。本文将设备之间连线的状态分为三种：（1）设备之间有双绞线，但并没连通；（2）设备之间连线已经连通；（3）设备之间存在虚拟的连线，即可以通过两个及两个以上交换机连通的设备之间的连线。对应于三种不同的连线状态， $\text{ConnectPoint}[m,n]$ 有不同的值，拓扑图的连线有不同的连线，具体关系如图 4.9 所示。

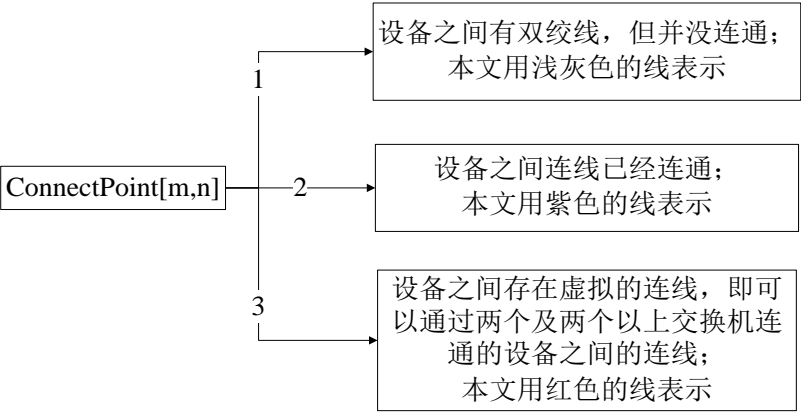


图 4.9 $\text{ConnectPoint}[m,n]$ 值、连接线的颜色及连线状态的关系

4.2.3 实时同步连线算法的设计

设计连线算法的目的有两个：（1）在整个机柜的网络拓扑结构基础上使用连线算法使得学生可以通过配置交换机划分 vlan ，从而达到由学生主动的组建属于本实验组的网络结构的目的，这也就避免了长久以来学生需要死记硬背单一网络拓扑结构的问题，不论是从实验的可操作性还是从学生学习知识的角度考虑都是一个质的飞跃；（2）连线算法可以将每个组员划分的网络拓扑结构实时的显示到其他组员的实验平台上，在网络拓扑搭建这一重要实验步骤上实现实时同步功能。

连线算法的流程是：学生通过双击交换机图标，触发交换机的 $\text{DoubleClick}()$ 事件， $\text{DoubleClick}()$ 事件有两个功能，第一个功能是将显示设备配置信息的 Textbox 显示出来，隐藏其他 Textbox ，第二个功能是将设备锁定，仅允许占有的组员进行写操作。通过更改 Textbox 中的配置信息便可以完成对交换机的配置以及 vlan 的划分。划分完 vlan 后，后台就会进行以下两个方面的处理：

- (1) 后台会根据配置信息，找到 vlan 内相对应的端口之间的连线，即 $\text{ConnectPoint}[m,n]$ 数组中的值，并根据图 4.9 中的对应关系相应地改变 $\text{ConnectPoint}[m,n]$ 数组中对应的值，通过画图技术将拓扑图用不同颜色的连接线展示出来；
- (2) 后台会按照本文 4.1 节实时同步实验操作的方法将交换机配置 vlan 的信息发送给其他组员。其他组员的后台根据接收到配置信息做出逻辑分析，然后根

据图 4.9 中的对应关系相应的改变 ConnectPoint[m, n]数组的值，最后通过画图技术将拓扑图用不同颜色的连接线展示在各自的终端上。

连线算法流程图如图 4.10 所示，

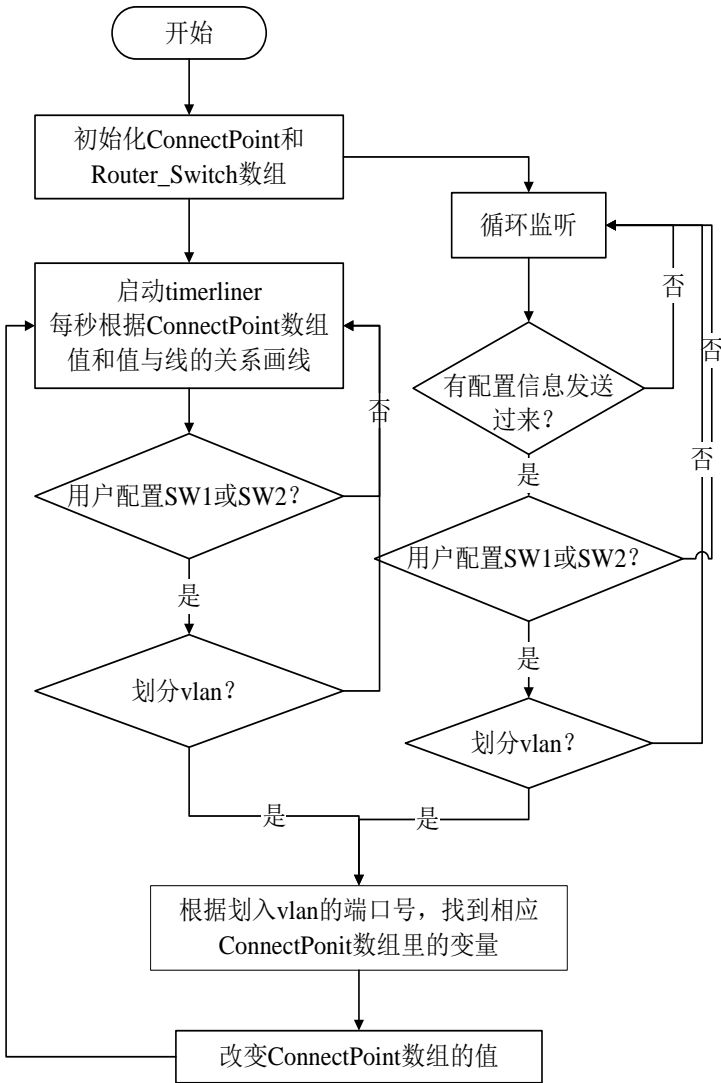


图 4.10 连线算法流程图

图 4.10 中，连线算法分为左右两个方面，左侧流程是根据学生操作网络设备，后台做出相应的逻辑处理，在自己的平台上展示出自己搭建的小组网络拓扑，右侧流程是学生接收到其他远程终端组员的交换机配置信息后，后台做出逻辑处理并在平台上展现出远程端组员搭建的网络拓扑，通过左右两个流程，完成小组实验拓扑的搭建以及实验拓扑的实时同步。

4.2.4 实时同步连线算法的实现

综合考虑设计的连线算法特性以及实际实验过程，目前主要有以下几个工作需要完成：

(1) 根据 ConnectPoint[m, n]画连线。

本文网络拓扑中的设备连线是使用 Pen 类和 Graphics 类来实现的。Pen 类用于绘制直线和曲线的对象，Graphics 类封装了一个 GDI+绘图图面，这两个类皆封装于 System.Drawing 命名空间中^[38]。Pen 类的构造函数 Pen(Color, Single)可用来构造指定颜色和宽度的实例，Graphics 类中的 DrawLine(Pen, Int32, Int32, Int32, Int32)方法绘制一条连接由坐标对指定的两个点的线条^[39]，本文使用这两个方法根据设备图标的位置坐标完成了设备之间不同颜色连接线的绘制，具体实现如下：

```
Pen myPen = new Pen(Color.Gray,1); //灰色线
Graphics g = this.CreateGraphics();
g.DrawLine(myPen, Router_Switch[m].Location.X + 20,
Router_Switch[m].Location.Y + 25, Router_Switch[n].Location.X + 20,
Router_Switch[n].Location.Y + 25);
```

为了保证连线能够实时的显示，本文使用timerliner定时器，设置Interval值为1，即每秒钟执行一次timerliner_Tick()事件。timerliner_Tick的作用是根据图4.6中线与ConnectPoint[m, n]值的对应关系，使用循环语句和判断语句根据ConnectPoint[m, n]的值，每秒钟画出Router_Switch[m]和Router_Switch[n]之间的连线，从而实现了实时展现设备之间的连线关系。

(2) 虚拟化 vlan 的划分。

结合实际实验，划分 vlan 不仅仅是创建 vlan 并将端口加入其中，在配置 vlan 之前还必须将要划分的交换机端口打开，因此本文设计了 undo 数组保存端口的打开和关闭状态，0 为关闭状态，1 为打开状态。undo 初始值为 0，学生打开端口，相应的 undo 值改成 1，学生输入创建 vlan 的代码，选中要加入 vlan 的端口后，后台会判断要加入 vlan 的端口是否打开，打开则找到端口相应的连线，改变 ConnectPoint[m, n]的值，实现的核心代码如下：

```
case "[SW1-vlan2]port E1/0/22 to E1/0/23":
    if (undo[0] == 1 && undo[1] == 1)
    {
        textBox_conf4.AppendText("\n");
        textBox_conf4.AppendText("[SW1-vlan2]");
        ConnectPoint[1, 2] = 3;
        ConnectPoint[3, 1] = 2;
        ConnectPoint[3, 2] = 2;
    }
    else
    {
```

```

        textBox_conf4.AppendText("The port is closed!");
        textBox_conf4.AppendText("\n");
        textBox_conf4.AppendText("[SW1-vlan2]");
    }

```

从上面的实现中可以看出划分一次 vlan 要改变三条线的颜色，这是本文特意为了使学生能够更加深刻的理解 vlan 而设计的，拓扑上分别展示出实际使用的连线和虚拟连接线的效果图，使得学生可以更加直观的看到自己设计的网络拓扑，从而更加深入的理解 vlan 的原理和作用。

(3) 异地同步网络拓扑

图 4.10 右侧的流程图便是异地同步网络拓扑，目的在于可以把学生在本地划分的网络拓扑实时同步到其他组员的平台，从而实现小组每个成员的平台都可以实时显示最新的完整拓扑图。本文使用 4.1 节介绍的网络编程技术将交换机 vlan 的配置信息发送给其他组员，同时也将相应的配置文本框名称，以及设备端口状态 undo 数组发送给其他组员。实验平台后台接收到信息后，先根据配置文本框名来找到需要配置的设备，再根据 undo 数组值打开相应设备端口，最后使用连线算法根据收到的配置信息改变 ConnectPoint[m, n]数组的值，从而画出本实验平台的网络拓扑图，接收到信息的后台处理部分实现如下：

```

while (true)
{
    byte[] data = new byte[1024];
    int recv = socket.ReceiveFrom(data, ref ep);
    string GetMessage = Encoding.UTF8.GetString(data, 0, recv);
    string textno = GetMessage.Substring(0, 13);
    switch (textno)
    {
        case "textBox_conf4":
            string t4 = GetMessage.Substring(26, GetMessage.Length - 26) + "\n";
            switch (GetMessage.Substring(26, GetMessage.Length - 26))
            {
                case "[SW1-vlan2]port E1/0/22 to E1/0/23":
                    if (GetMessage.Substring(13, 2).Equals("11"))
                    {
                        ConnectPoint[1, 2] = 3;
                        ConnectPoint[3, 1] = 2;
                        ConnectPoint[3, 2] = 2;
                    }
                }
            }
    }
}

```

```

        }
        break;
        .....
    }
}
}

```

因为配置信息是实时发送的，而且配置信息的处理也是实时进行的，至此，其他组员的网络拓扑可以实时的显示在自己的平台上，整个小组实验的拓扑也实现了实时同步。

4.3 并发控制机制

本文设计的虚拟协同实验是一个标准 CSCW (Computer Supported Cooperative Work) 系统，作为 CSCW 系统，为了提高系统响应时间，一般将共享对象复制到个用户本地平台，各用户对共享对象的操作以消息的方式通知其他用户，其他用户在接收到消息后在本地执行相应的操作，从而保持各用户端共享对象的一致性^{[40][41]}。在 CSCW 系统中，并发控制是关键技术之一，它直接影响着 CSCW 系统的效率，本节在对传统并发控制技术进行分析和比较之后，结合本平台的特点，提出了“改进的高优先级两阶段封锁法”的并发控制算法，并结合 Sql Server 有效的实现了实验的并发控制。

4.3.1 实验用户的注册和登录

为了确定小组实验中组员的身份及相关信息，学生在进入本平台小组实验之前，需要进行账号的注册，注册成功后以注册身份登录，选择要加入的小组实验进行实验。注册和登录模块的主要流程图如图 4.11 所示，

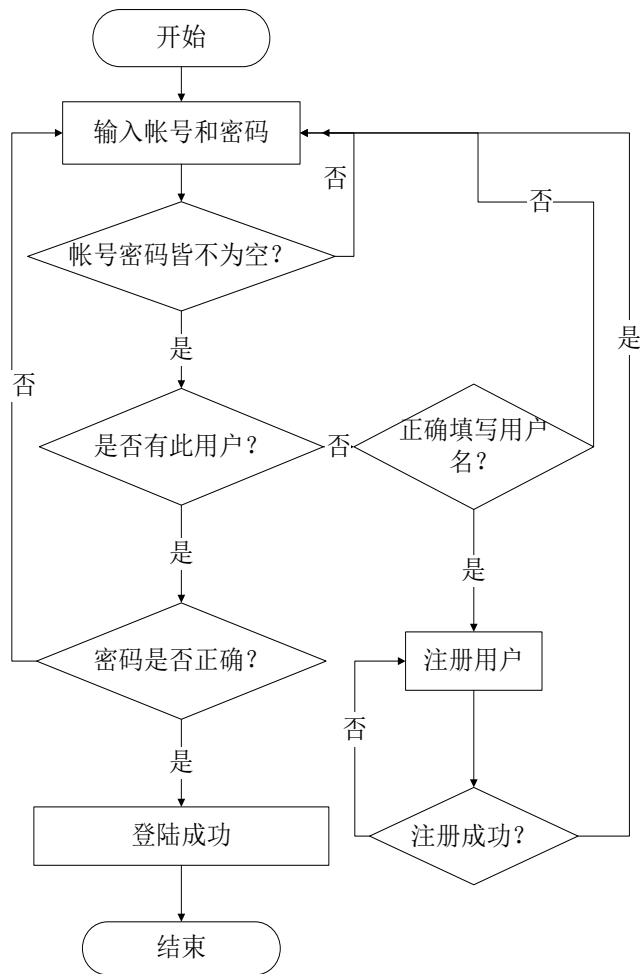


图 4.11 注册登录流程图

注册和登录模块是通过 sql server 辅助实现的，本文在 joint 数据库中设计了 user 表保存用户注册信息，使用 ADO.NET 技术，Insert 或 Select 数据表 user 中的数据，完成学生注册和登录。user 表的结构如表 4-7 所示，表中主要保存了学生的用户名，密码，id 以及注册 pc 的 ip。

表 4-7 user 表

列名	数据类型	长度	允许空	字段说明
id	int	4 字节	主键不许空	记录编号
username	varchar(MAX)	2 ³¹ - 1 字节	允许为空	用户名
password	varchar(MAX)	2 ³¹ - 1 字节	允许为空	用户密码
ip	varchar(50)	50 字符	允许为空	用户 ip

4.3.2 虚拟协同实验组的创建

学生在注册并成功登录平台之后，会进入选择实验和实验组的界面，此界面

主要分为三个部分，一部分显示学生的个人信息，一部分显示可以选择的实验，最后一部分显示可以加入或创建的实验组。只有选择好了要做的实验和要加入的实验组，学生才能进入虚拟协同实验平台正式开始与其他同学合作完成实验。学生加入实验组的流程图如图 4.12 所示，

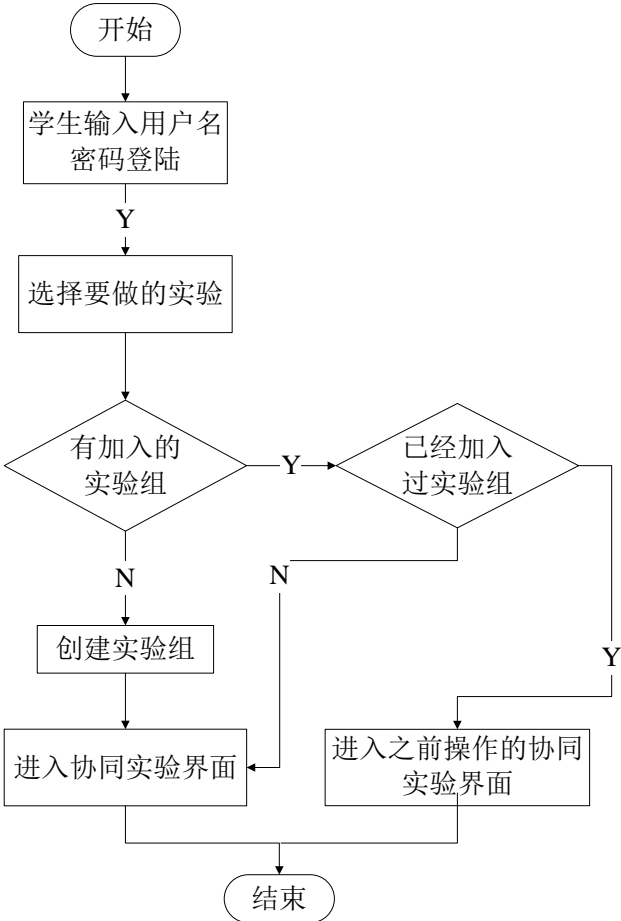


图 4.12 学生加入实验组流程图

本文对界面的三个主要功能的实现方式分别是：

第一、展示学生的个人信息。

本文中学生login界面与join界面是两个不同的Form窗体，因此，如何将一个Form窗体的信息传递到另一个Form窗体中是实现本功能的关键。本文中Form窗体之间的传值是利用类的构造函数实现的，本文用join类继承login类，join类中有包含user和ipadd两个参数的构造函数，通过下面的语句将join界面展示出来，并在此过程中将用户注册的用户名和用户的ip地址传递到join类中。

```
Form frm = this;
join join= new join(user,ipadd);
join.Show();
```

join 类使用构造函数获取用户名和 ip，界面上使用三个文本框，保存并显示用户的信息，具体实现如下：


```
public partial class join : Form
{
    public join(string user,string ipadd)
    {
        InitializeComponent();
        this.label1.Text="欢迎您 "+user+"，请创建或加入您的RIP路由协议  
实验小组";
        this.label2 .Text=user;
        this.label4.Text = ipadd;
    }
    .....
}
```

第二、选择要做的实验。

本文使用 **GroupBox** 将学生可以选择的实验聚集在一起，每个实验使用 **RadioButton** 按钮来显示，学生只要点击想要加入的实验，就会触发 **RadioButton** 的 **click（）** 事件，**click（）** 事件可以为学生展示出此实验现在所有的实验组和实验组成员，以帮助学生选择要加入的实验组。

第三、选择要加入的实验组。

本文在选择要做的实验中讲到，**RadioButton** 按钮的 **click（）** 事件可以显示出本实验的所有实验组和实验组成员，这主要是依靠 **DataGridView** 结合 **Sql Server** 实现的。

DataGridView 是一个网格控件。它取代了 window form 之前版本中 **DataGrid** 控件，易于使用、高度可定制、支持很多用户需要。使用 **DataGridView** 控件，可以显示和编辑来自多种类型数据源的数据。**DataGridView** 控件的数据绑定非常简单，大多数情况下，只需设置 **DataSource** 属性即可。本文使用它来显示数据库中 学生创建和加入实验小组的数据。

本文在 **joint** 数据库中设计了 **team** 表用于保存学生创建及加入实验组的数据，**team** 表的结构如表 4-8 所示，

表 4-8 team 表

列名	数据类型	长度	允许空	字段说明
exname	varchar(MAX)	2 ³¹ -1 字节	允许为空	实验项目名
teamname	varchar(MAX)	2 ³¹ -1 字节	允许为空	实验组名
leadername	varchar(MAX)	2 ³¹ -1 字节	允许为空	实验组创建者

membername	varchar(MAX)	2 ³¹ -1 字节	允许为空	实验组成员
------------	--------------	-----------------------	------	-------

学生进入 join 界面选择要做的实验后, 触发 RadioButton 控件 Click() 事件, 在 Click () 事件中本文采用了 ADO.NET 技术, 在 team 表中 select 所有学生选择的实验的实验组信息, 并将查询到的数据源绑定到 DataGridview 中展现出来。学生查看实验组信息之后, 决定加入还是创建一个实验组。

学生在 TextBox 中填入自己要选择的组, 点击创建或加入 button 按钮后, 便会分别触发 button 按钮的 click () 事件, click () 事件将学生创建或加入实验组的信息 insert 或 update 至 team 表中。由于每个时间点都可能会有学生加入实验, 本文使用 timer 控件每秒钟获取一次 team 表中 RadioButton 选中实验的实验组信息, 展示到 DataGridview 中^[42], 具体实现如下:

```
string str = "server=(local);Database=joint;Uid=sa;Pwd=123456";
SqlConnection sql;
private void timer1_Tick(object sender, EventArgs e)
{
    sql = new SqlConnection(str);
    sql.Open();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "select * from team where exname='"+exname+"'";
    cmd.CommandType = CommandType.Text;
    DataSet ds = new DataSet();
    SqlDataAdapter sda = new SqlDataAdapter(cmd.CommandText, sql);
    sda.Fill(ds);
    dataGridView1.DataSource = ds.Tables[0].DefaultView;
    dataGridView1.Visible = true;
    sql.Close();
}
```

至此, 学生完成了要加入的实验和实验组的选择, 进入相应虚拟协同实验组进行协同实验的操作。

4.3.3 并发控制机制的设计

在传统的分布式系统中, 一般是通过事务处理机制之间的串行性来满足并发控制的一致性要求, 用的比较多的方法有: 时间邮票顺序法、多版本时间戳顺序法、二步封锁法(2PL)以及无锁机制法等^[43]。CSCW 系统的并发控制主要有两种极端: 依靠对共享对象的协作感知解决冲突和令牌传递协议, 目前常用的并发

控制的方法一般是在这两种极端之间，本文对常用的方法进行详细介绍与对比，如表 4-9 所示^[44]。

表 4-9 并发控制方法的对比

并发控制方法	定义	优点	缺点
加锁法	对共享对象进行加锁和解锁	设计和实现简单	申请和释放锁消耗时间； 用户难以控制申请和释放锁的时间和数据粒度
集中控制法	集中控制进程管理共享对象的操作	实现简单	会引起与集中式系统相关的问题； 系统响应时间延长
串行化法	将共享对象操作序列化，用队列来执行	避免了资源冲突	不支持多进程； 系统工作量大、效率低
令牌环	各站点得到令牌的用户可以操作	传输速度高	空闲令牌易丢失； 令牌易重复
依赖检测	以操作的时间戳检查冲突	响应时间快	有冲突的要人工干预

本文结合本虚拟实验平台的特点，提出了“改进的高优先级两阶段封锁法”的并发控制方法。高优先级两阶段封锁法使用锁调度机制，事务要访问某节点数据对象时，该节点的锁调度机制对数据对象加锁，如果该数据对象已经被其他事物加锁，就按照优先级顺序，优先级高的事务获得操作权，其他事务重启^[45]。本文对其进行改进，改进算法的基本原理为：①当某一个用户对共享对象进行写操作的时候，其他用户对共享对象仅有读的权限；②对同一个共享对象有多个用户同时申请对其进行写操作的时候，根据用户对共享对象的操作时间设定优先级，时间早的优先级高。

4.3.4 并发控制机制的实现

学生在进入虚拟协同实验界面后，首先要做的操作便是占用自己想要操作的设备，本文中采用双击设备图标的方式完成设备的占用操作，在双击鼠标的同时会出现本设备的配置窗口。按照“改进的高优先级两阶段封锁法”的设计思想，先双击设备的学生会获得设备的写操作权限，设备锁定，此时其他组员便只有读取此设备配置信息的权限。权限的控制是依靠 Sql Server 中的 user_owner 表实现

的，user_owner 的表结构如表 4-10 所示。

表 4-10 user_owner 表

列名	数据类型	长度	允许空	字段说明
username	varchar(MAX)	2 ³¹ -1 字节	允许为空	用户名
exname	varchar(MAX)	2 ³¹ -1 字节	允许为空	实验项目名
teamname	varchar(MAX)	2 ³¹ -1 字节	允许为空	实验组名
R1	int	4 字节	允许为空	R1 是否锁定
R2	int	4 字节	允许为空	R2 是否锁定
R3	int	4 字节	允许为空	R3 是否锁定
S1	int	4 字节	允许为空	S1 是否锁定
S2	int	4 字节	允许为空	S2 是否锁定

从表 4-10 中可以看到，R1、R2、R3、S1、S2 列标记用户对于设备的锁定情况：值为 1 为未锁定，值为 2 为锁定。学生创建或加入实验组进入协同平台后，后台使用 `insert into user_owner(username,exname,teamname,R1,R2,R3,S1,S2,Wr) values ('" + user + "','" + exname + "','" + teamname + "',1,1,1,1,1,0)` 语句在表 4-10 添加一条记录，每条新添加的记录设备锁定的初始值皆为 1。学生在双击设备图标时触发 `pictureBox_lr1_DoubleClick` 事件，令相应 `textbox` 的 `visible` 为 `true`，展现出相应设备的 `textbox` 配置框，与此同时使用 `update user_owner set R1=2 where username <>'" + label2.Text + "' and exname='" + label5.Text + "' and teamname='" + label3.Text + "'` 语句，将学生双击设备的锁定值更新为 2。`timerliner` 计时器每秒钟除了画连接线之外，还监视是否有锁定值变为 2 的设备，一旦检测到就将 `pictureBox_lr` 的 `Enabled` 设置为 `false`，其他组员的此设备图标不可再点击，同时还将 `labelcheck` 的 `visible` 设置为 `true`，在其他组员相应设备旁显示出“正在配置，点击查看代码”的提示。此时，若其他组员想查看设备配置情况，双击 `label` 图标即可出现属性为 `readonly` 的 `textbox` 文本框及现在设备的配置情况。至此，实现了本实验平台“改进的高优先级两阶段封锁法”的并发控制机制。

4.4 本章小结

本章主要对实时同步虚拟协同实验平台进行了具体的功能设计与实现，针对实时同步实验操作、同步搭建实验拓扑以及实验操作的并发控制等几个关键技术的实现做了详细的阐述。

第五章 实时同步虚拟协同实验的运行

本章主要从设备配置的实时同步、实验拓扑的同步搭建、并发操作的控制等几个方面，对本文设计搭建的虚拟协同实验平台的主要功能进行了测试，并在测试的实例中给出了效果截图与详细说明。

5.1 实时同步实验操作的实例

本文假设学生 kdf 占用 SW1（交换机 1），学生 sharpay 占用 SW2（交换机 2），本文以在 sharpay 的实验平台查看 kdf 对 SW1 的配置来验证实验平台的实时同步功能，为了能够突显出平台的实时性，本文在查看和配置文本框上加上了时间显示器，并在同一台 PC 机上对比实验效果图。学生 kdf 配置 SW1，此时学生 sharpay 右键双击自己平台 SW1 设备图标旁的“正在配置，点击查看代码”label，出现查看 SW1 配置情况的 textbox，为了能够集中检测实时同步配置信息的功能，本文只给出关键部分的实验效果图，如图 5.1 所示。

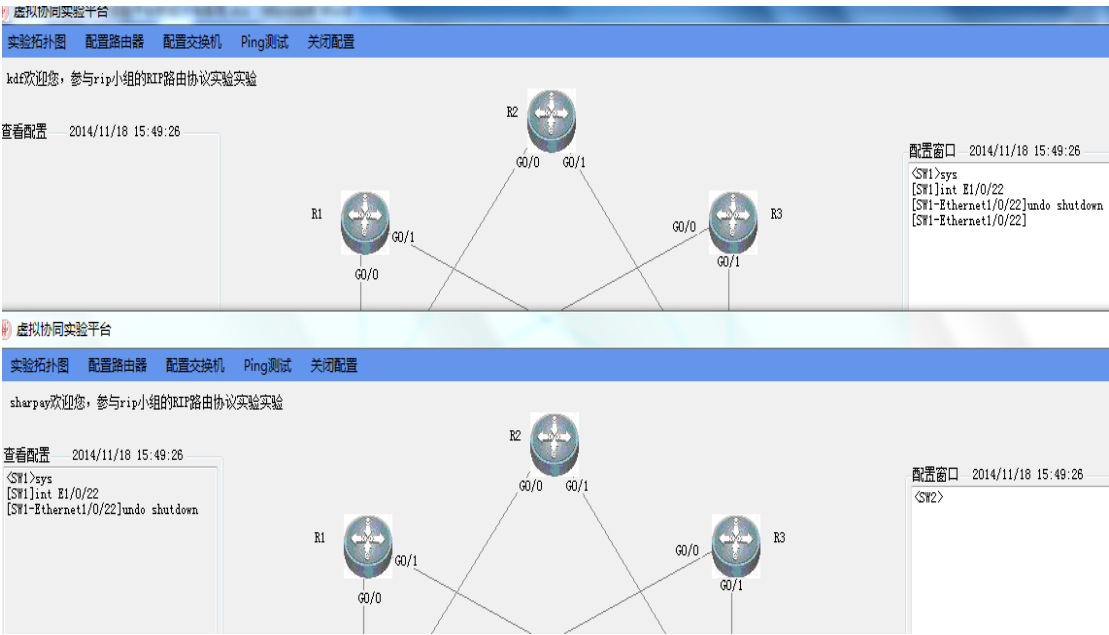


图 5.1 实时同步配置信息实例截图

图 5.1 中有 kdf 与 sharpay 两个学生用户共同完成 rip 路由协议实验，kdf 配置窗口可以看到 15:49:26 这个时刻 kdf 对 SW1 的配置情况，此时 sharpay 的查看配置窗口中 15:49:26 时的配置信息与 kdf 相似，但是由于提示行的命令对于实验是没有操作意义的，所以 sharpay 查看配置与 kdf 配置窗口的命令行在同一时刻是相同的，这就证明本平台实现了实时同步设备配置信息的功能，本文第 4 章讲

到本实验设备操作的实质是对实验设备的配置，所以本实例表明本实验可以实时同步实验操作。

5.2 同步搭建实验拓扑的实例

本实例同样假设学生 kdf 占用 SW1（交换机 1），学生 sharpay 占用 SW2（交换机 2），由于本小节主要验证实验拓扑的实时同步搭建，因此本文在实验拓扑上加上一个时间显示器。本文按照计算机网络 rip 路由选择协议的网络结构图搭建网络拓扑，rip 路由选择协议网络拓扑图如图 5.2。

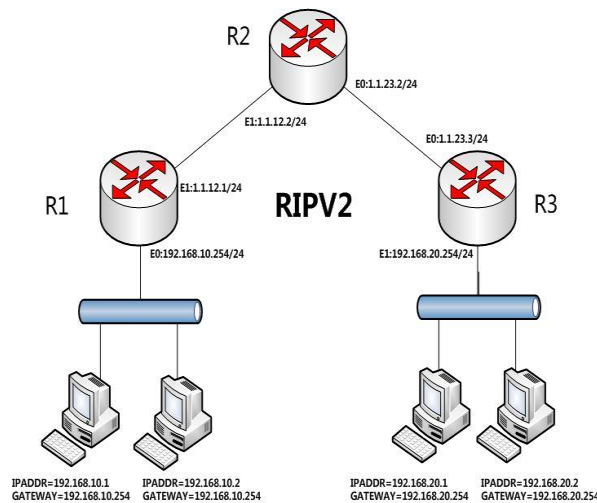


图 5.2 rip 路由选择协议网络拓扑图

学生 kdf 配置 SW1，划分 vlan，与 sharpay 共同搭建图 5.2 的网络拓扑，如果 kdf 划分的网络拓扑能够实时显示到 sharpay 的实验平台上那就证明本平台实现了网络拓扑的同步构建。学生 kdf、sharpay 网络拓扑的实验截图分别如图 5.3、5.4 所示。

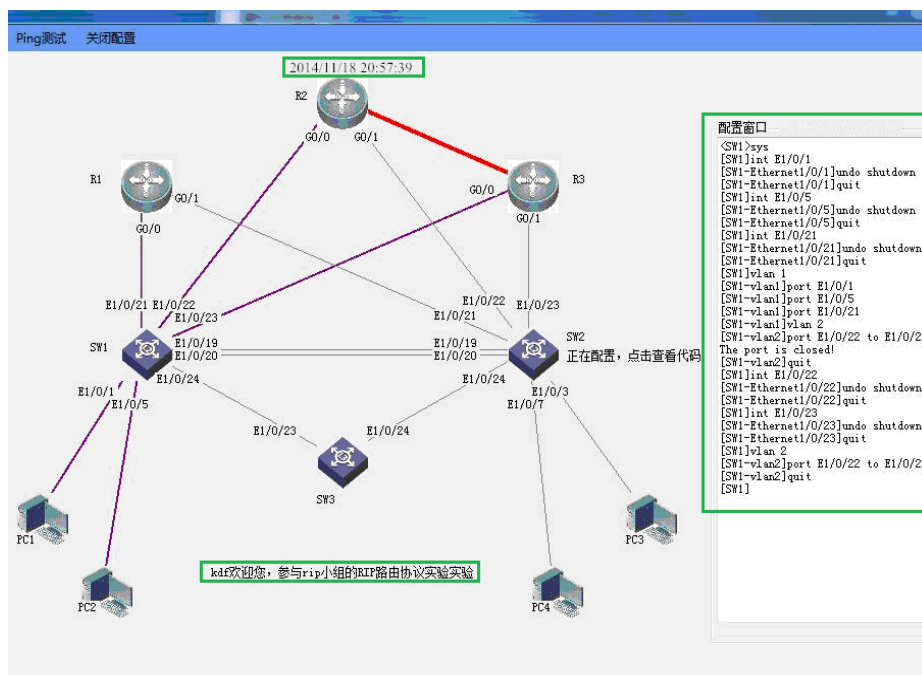


图 5.3 kdf 网络拓扑截图

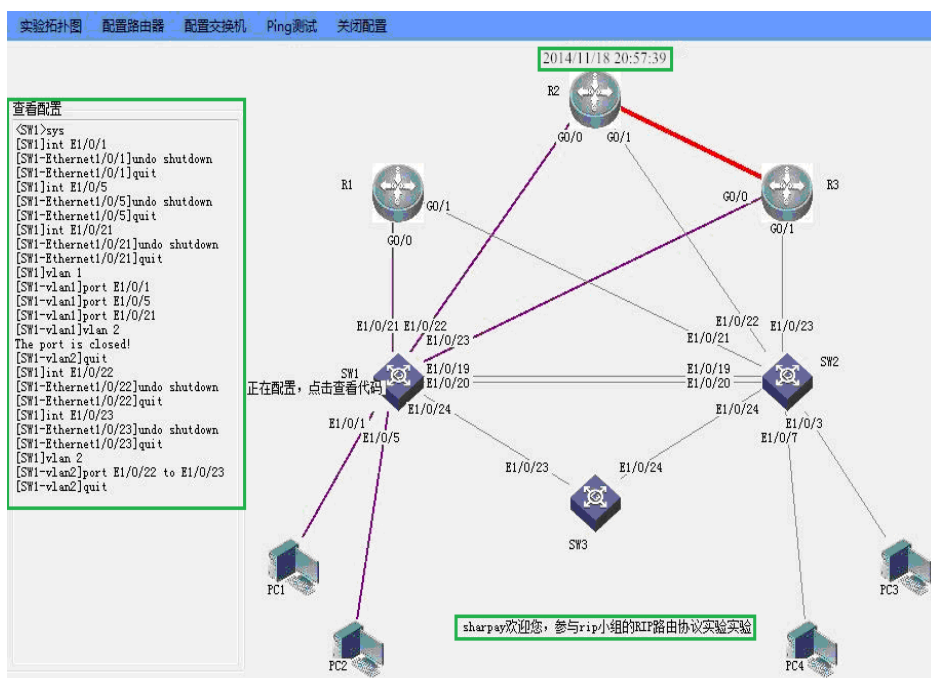


图 5.4 sharpay 网络拓扑截图

图 5.3 是 kdf 通过在右侧配置窗口配置 SW1，将 SW1 的 1、5、21 端口划分到 vlan 1 中，将 22 和 23 端口划分到 vlan 2 中，则在 kdf 平台上出现了三种颜色的连接线：灰色代表连接但没有连通、紫色代表连接且连通、红色表示划分 vlan 后可以通过交换机连接的路由器虚拟连线。图 5.4 左侧查看配置窗口显示的是 kdf 配置 SW1 的信息，在 20:57:39 时图 5.3 和图 5.4 呈现出相同的网络拓扑结构，这个结构是来自于 kdf 的配置结果，这也就验证了本平台可以同步搭建网络拓扑。

图 5.3 可以看到配置框中在未打开 22 和 23 端口的时候划分 vlan 2，设备给出了 The port is closed 的错误，这说明在划分 vlan 之前必须保证交换机端口是打开的。

5.3 并发控制机制的实例

并发控制的验证实例主要分为三个部分：第一部分是学生注册和登录，第二部分是实验和实验组的选择，第三部分是并发占用实验设备。本小节仍然假设学生 kdf 与学生 sharpay 共同完成实验，分别给出三个部分的测试实例。

5.3.1 学生注册和登录实例

本小节分别以 kdf 注册，sharpay 登录作为测试用例。kdf 同学注册成功后的界面如图 5.5 所示，sharpay 同学登陆成功界面如图 5.6 所示，

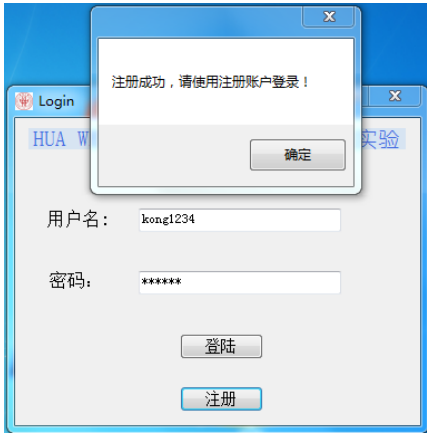


图 5.5 注册成功界面



图 5.6 登陆成功界面

5.3.2 学生选择实验和实验组

kdf 点击 RIP 路由协议实验按钮，DataGridView 实时显示数据库 team 表中

rip 实验组的数据。kdf 输入小组名创建一个实验，实验界面如图 5.7 所示，kdf 的小组信息马上显示在 DataGridView 中。

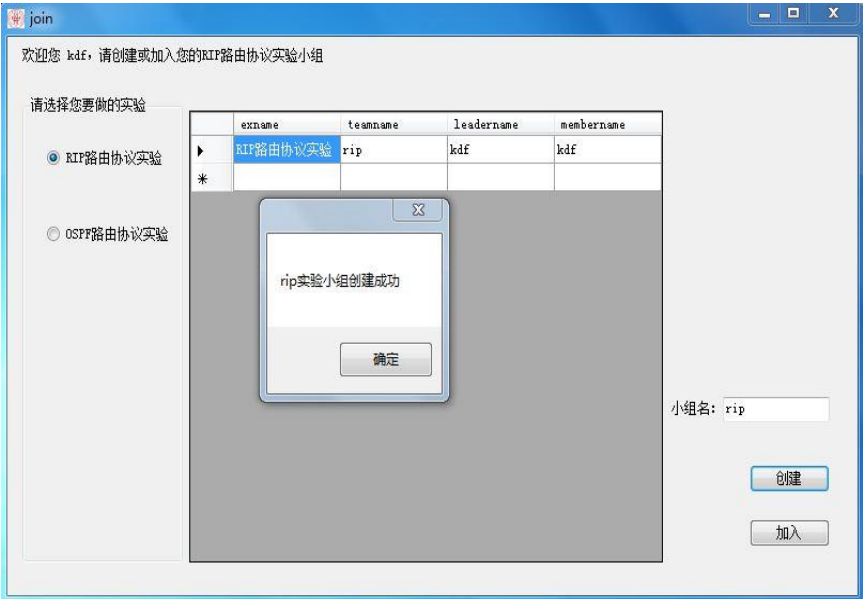


图 5.7 kdf 创建 rip 实验组截图

sharpay 点击 RIP 路由协议实验按钮，DataGridView 实时显示数据库 team 表中 rip 实验组的数据，由于 sharpay 比 kdf 晚操作，所以显示出 kdf 创建的小组信息，sharpay 输入要加入的小组名，sharpay 加入的实验组信息便立刻显示在了 DataGridView 中，这样 rip 小组就有两个成员，如图 5.8 所示。

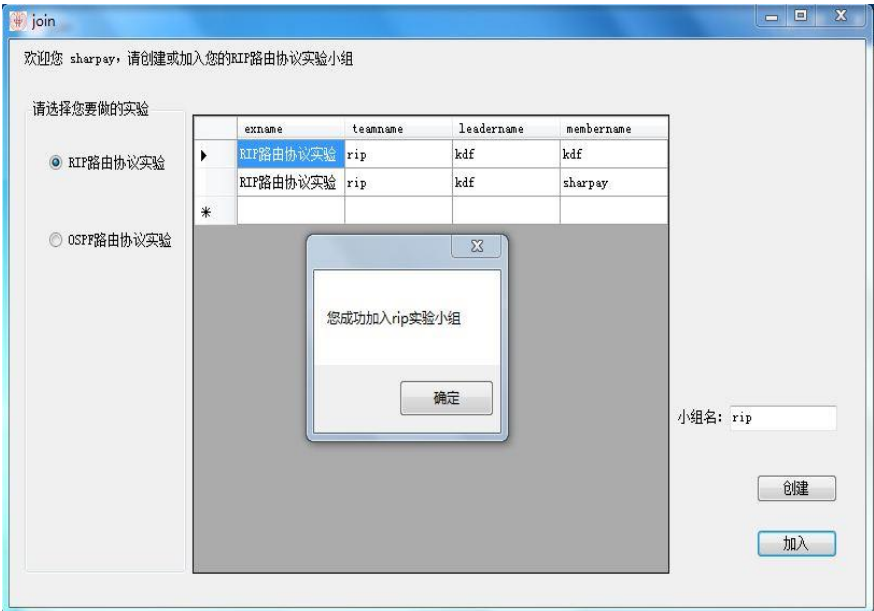


图 5.8 sharpay 加入 rip 实验组截图

5.3.3 学生选择实验和实验组

并发占用实验设备的功能验证上，本文假设学生 kdf 先占用 SW1(交换机 1)，

kdf 占用设备界面截图如图 5.9 所示，

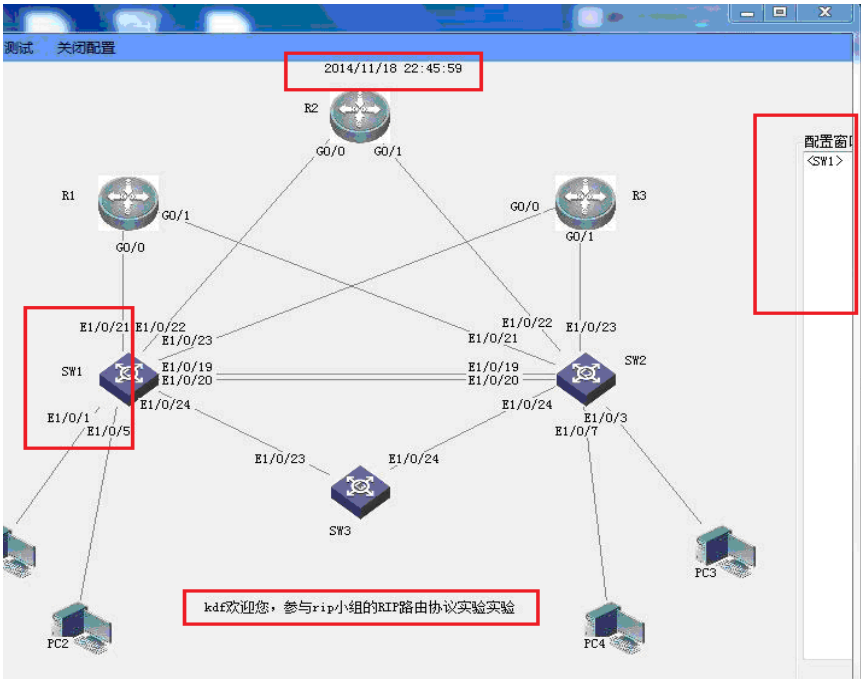


图 5.9 kdf 占用设备界面截图

sharpay 占用设备界面截图如图 5.10 所示，

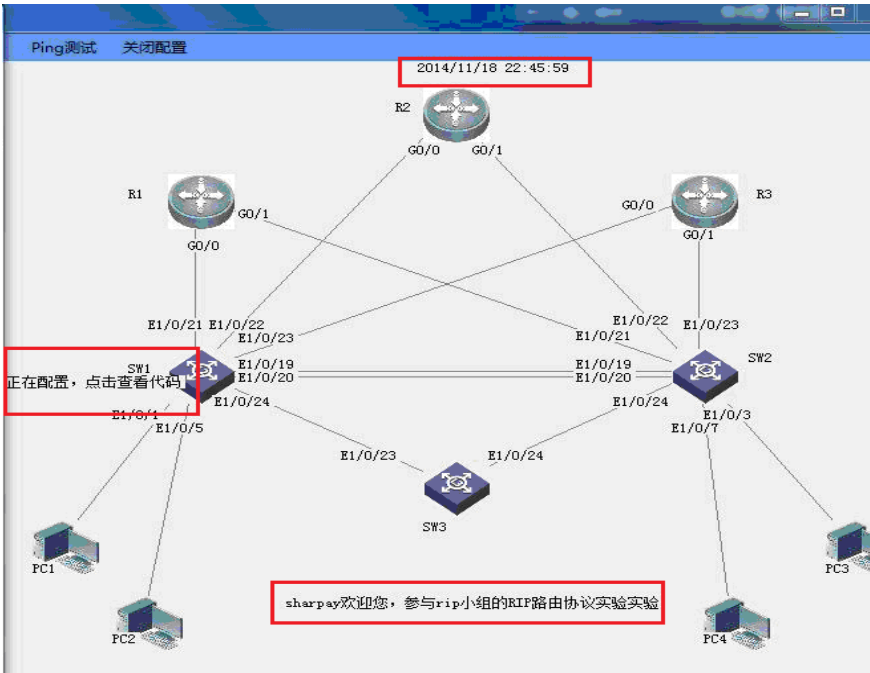


图 5.10 sharpay 占用设备界面截图

kdf 双击 SW1 的同时会出现配置 SW1 的文本框，该文本框是可写入的，并且在图 5.10 中可以看到在 kdf 双击 SW1 的时刻（22:45:59），sharpay 界面 SW1 旁会出现“正在配置，点击查看代码”的 label 标签。此时 sharpay 双击 SW1 是没有反应的，双击 label 标签出现图 5.4 左侧的查看配置文本框，配置文本框是灰色

的，不可写只可读。至此成功验证了本实验平台并发控制机制的有效性。

5.4 本章小结

本章主要在两人操作的虚拟协同平台上，对实时同步实验操作、同步搭建实验拓扑、并发操作的控制三个关键功能给出了具体的测试实例，对相应功能进行了具体的演示，给出了功能的实验效果图。结合这些测试实例的实验效果图更加充分的展示了本实验平台的实现过程和功能特性。

第六章 总结与展望

6.1 工作总结

本文在单人虚拟组网实验平台基础组网实验的开发思想下,结合多人协作虚拟实验的项目需求及项目特点,设计并开发了实时同步虚拟协同实验平台。该平台不仅能够允许多人共同完成同一项组网实验,增强了学生之间的合作意识,还严格模拟真实的实验流程及实验操作,更好的帮助学生熟悉实验操作流程,全面地学习和掌握计算机网络知识。

本文对实时同步实验平台的需求进行分析和总结,提出了虚拟协同平台的设计目标,通过网络编程技术、C#编程语言、.NET 类库及 Sql Server 数据库等关键技术,搭建了一个可以满足同步搭建网络拓扑、实时同步实验操作及控制设备并发操作的虚拟协同实验平台。

经过运行实例证明,该虚拟协同实验平台基本满足多人协同完成同一项虚拟实验的需求,它的设计和开发,为本课题进一步实现智能化在线虚拟协同实验平台提供了重要理论依据和技术基础。

6.2 研究展望

为了使能够方便的在不同地域进行协同实验,以及能够更加自由的协同操作虚拟设备,本研究所开发的实时同步虚拟协同实验还有以下几个方面有待提高和完善:

- (1) 完成在线虚拟协同实验平台的搭建
- (2) 实现对同步配置相同设备时的并发控制
- (3) 优化交换机和路由器的配置信息使其更加智能化

参考文献

- [1] <http://baike.baidu.com/view/8864.htm?fr=aladdin>
- [2] 叶小英. 虚拟实验教学系统在远程教育中的应用与评价研究[D].北京:北京邮电大学, 2010.
- [3] <http://baike.baidu.com/view/1515402.htm?fr=aladdin>
- [4] 王有宾,何仲. 高校虚拟实验室建设的研究[J]. 数字技术与应用,2011,11:119+121.
- [5] 蔡旻,王浩强. 计算机支持下的协同设计技术研究[J]. 成都信息工程学院学报,2004,04:500-502.
- [6] 甘茂华,阮丽娜,李昌国,杨春. 多人协作虚拟实验室综述[J]. 计算机应用与软件,2010,05:130-132+143.
- [7] 潘淼. 虚实互动的在线组网实验平台的研究和设计[D].东华大学,2013.
- [8] 叶步伟. 虚拟实验研究现状及其前景分析[J]. 中小学电教,2010,Z1:41-44.
- [9] Lianguan Shen,Yu Hao,Mujun Li,Wei zhao,Jinjin Zheng,A Synchronous Collaborative Environment for Engineering Design Education, Proceedings of the 2007 11th International Conference on Computer Supported Cooperative Work in Design.
- [10] Carlos A. Jara, Francisco A.Candelas, Fernando Torres. Real-time collaboration of virtual Laboratories through the Internet. Journal article Compute Education,Volume 52, Issue1. Jan 2009.P126~140
- [11] Qin Jing, Choi Kup-SzeZ, PoonWai-Sang and so on. A framework using cluster-based hybrid network architecture for collaborative virtual surgery. Computer Methods and Programs in Biomedicine.Vol.96(3).Dec2009. P205 — 216
- [12] 崔光佐, 程旭, 杨芙清. 基于 WWW 的网上虚拟实验室设计与实现[J]. 北京大学学报, 2004(6): 43-46.
- [13] 郑耿忠. 基于 Web 的协同电路虚拟实验室的设计及实现[J]. 计算机时代,2006,04:50-52.
- [14] 马艳芳.协同虚拟实验的研究与设计[D].北京:北京邮电大学教育技术学, 2011.
- [15] 戴玮烨. 在线自由组网虚拟实验的设计与实现[D].东华大学,2013.
- [16] 张胜兵,吕养天.专业课程计算机网络工程实验教程[M].西北:西北工业大学出版社,2006:22-24.
- [17] <http://msdn.microsoft.com/library/zw4w595w.aspx>
- [18] 陈国顺,付卫华. .NET Framework 类库概述[J]. 现代情报, 2003, (12):214-215.
- [19] 李铭.C#高级编程(第6版)[M].北京:清华大学出版社,2008:1-11.
- [20] <http://baike.baidu.com/view/4294.htm?fr=aladdin>
- [21] 陈承欢.Windows 应用程序设计案例教程[M].北京:人民邮电出版社,2009:10-25.

- [22] 李乃文.C#2008 程序设计实践教程[M].北京: 清华大学出版社,2009:100-140.
- [23] 陈锐,张蕾,李绍华.C#程序设计[M].北京: 清华大学出版社,2011:136-140.
- [24] Jesse Liberty, Donald Xie. Programming C# 3.0 5th Edition[M]. Massachusetts: O'Reilly Media, Inc, 2009: 253-271.
- [25] 梅晓冬,颜焯青.Visual C#网络编程技术与实践[M].北京:清华大学出版社,2008:21-25.
- [26] 金华,华进.C#网络编程技术教程[M].北京: 人民邮电出版社,2008:110-138.
- [27] <http://www.cnblogs.com/hydddd/archive/2009/07/26/1531538.html>
- [28] 郭郑州, 陈军红.SQL SERVER 2008 完全学习手册[M].北京: 清华大学出版社,2011:1-40.
- [29] Wendell Odom, Thomas Knott. Networking Basics CCNA 1 Companion Guide[M]. Massachusetts: Cisco Press, Inc , 2006: 133-140.
- [30] 明廷堂..NET 网络编程技术原理与开发实例[J].电脑编程技巧与维护,2013,19:56-64+70.
- [31] <http://baike.baidu.com/view/642103.htm?fr=aladdin>
- [32] <http://msdn.microsoft.com/zh-cn/library/system.threading.thread.aspx>
- [33] <http://www.cnblogs.com/songxingzhu/p/3677307.html>
- [34] <http://msdn.microsoft.com/zh-cn/library/system.windows.forms.control.invokerequired.aspx>
- [35] 万小强. VLAN 技术的研究与实现[D].北京邮电大学,2006.
- [36] 洪军,韩燮. VLAN 技术及在交换机上的实现[J]. 电脑开发与应用,2007,02:11-13.
- [37] 伍孝金. 计算机网络[M]. 北京: 清华大学出版社, 2007: 153-162.
- [38] Microsoft. Graphics Class [EB/OL].
<http://msdn.microsoft.com/en-us/library/system.drawing.graphics.aspx>.
- [39] http://blog.163.com/gis_warrior/blog/static/19361717320119227034181/
- [40] 冯晨华,徐捷,王鸿谷,庄钊文. CSCW 系统中的并发控制机制的研究[J]. 计算机工程与应用,1999,04:21-23+41.
- [41] 彭晖,王宇栋,刘金旺. 基于 Web 的同步协同虚拟实验室设计与实现[J]. 计算机工程与应用,2004,07:155-157.
- [42] 刘增杰,臧顺娟.Windows 应用程序开发(C#版)[M]. 北京: 清华大学出版社, 2011: 114-127.
- [43] Dourish P.Consistency Guarantees: Exploiting Application Semantics for Consistency Management in a Collaboration Toolkit. Proceedings of CSCW, Cambridge MA,USA:Now. 1996:pp.268-277
- [44] 古明家.分布式数据库系统的并发控制算法综述[J]. 现代计算机,2008,03:147-149.
- [45] Greeberg S and Marwood D.Real Time Groupware as a Distributed System:Concurrency Control and its Effect on the Interface, Proceedings of CSCW,pp.207-217

攻读学位期间的研究成果

参与项目

[1] 2012-2013 上海市教育科学 2011 年度重点项目(A1119): 基于虚实互动的网络实验教学新方法的研究.

发表论文

[1] 孔德凤 韩秀玲 陈光 任卓君. 实时同步协同虚拟实验平台的研究与设计. 实验室研究与探索 (已录用).

致 谢

饮其流时思其源，成吾学时念吾师，经过两个多月的磨练，终于将论文完成，论文完成之时，向我的导师陈光教授和指导教师韩秀玲副教授表示真诚的谢意，没有两位导师的帮助和教导，我是无法取得现在的成绩的。

从我进入东华大学读研深造以来，陈老师和韩老师就对我严格要求，作为转系生的我在很多专业知识方面都有欠缺，但两位老师对我的学习和课题研究进行耐心的指导和帮助，并推荐好的书籍供我提升自己的专业知识能力。在本文的撰写过程中，从项目的开启、程序的开发与调试、项目的试用一直到论文完成，每一步两位老师都倾注了大量的心血。两位老师渊博的专业知识、丰富的项目经验、忘我的工作作风都使我受益匪浅。

感谢课题组的师兄师姐们，将课题组的前期工作做得很出色，铺垫好的基石让后人的工作做得更无阻，更顺畅。课题组的其他同学的团队精神也深深地感染着我。还要向我的父母，帮助过我的老师、同学和朋友们表示衷心的感谢。

最后衷心地感谢在百忙之中评阅论文和参加答辩的各位专家，感谢您们对本文提出的宝贵建议！