华中科技大学
网络空间安全学院

《计算机通信与网络》实验报告

姓　　名　吴锦添

班　　级　信安 1805

学　　号　U201810398

联系方式　15972019448

分　　数

实验报告及代码和设计评分细则

评分项目		满分	得分	备注
文档格式		10		
感想（含思政）		5		
意见和建议		10		
Socket 编程	代码可读性	5		
	注释	5		
	软件体系结构	5		
	问题描述及解决方案	5		
局域网组网实验	设备选型合理性	5		
	遇到的问题描述及解决方案	5		
	对理论的理解	5		
	对工程的理解	5		
广域网组网实验（路由器实验）	遇到的问题描述及解决方案	5		
	对理论的理解	5		
	对工程的理解	5		
综合组网实验	拓扑结构	5		
	设备选型	5		
	经济、环境、管理的考虑	5		
	对工程的理解	5		
总分				

综合得分

操作得分		**实验报告得分**	
综合得分			
教师签名		**日期**	

目　　录

1 Socket 编程实验

1.1 环境

操作系统：Windows/Linux
编程语言：C

1.2 实验内容

完成一个 TFTP 协议客户端程序，实现一下要求：
（1） 严格按照 TFTP 协议与标准 TFTP 服务器通信；
（2） 能够将文件上传到 TFTP 服务器；
（3） 能够从 TFTP 服务器下载指定文件；
（4） 能够向用户展现文件操作的结果：文件传输成功/传输失败；
（5） 针对传输失败的文件，能够提示失败的具体原因；
（6） 能够显示文件上传与下载的吞吐量；
（7） 能够记录日志，对于用户操作、传输成功，传输失败，超时重传等行为记录日志；
（8） 人机交互友好（图形界面/命令行界面均可）；
（9） 能够实现两种不同的传输模式 netascii 和 octet；
（8）和（9）为额外功能，将视具体情况予以一定加分。

1.3 实验要求

（1） 必须基于 Socket 编程，不能直接借用任何现成的组件、封装的库等；
（2） 提交实验设计报告和源代码；实验设计报告必须包括程序流程图，源代码必须加详细注释。
（3） 实验设计报告需提交纸质档和电子档，源代码、编译说明需提交电子档。
（4） 基于自己的实验设计报告，通过实验课的上机试验，将源代码编译成功，运行演示给实验指导教师检查。

1.4 系统设计及实现（软件体系结构）

1.4.1 系统结构设计

系统模块划分：

在 Socket 实验中根据系统所需要满足的功能分为两大部分，分别是 Write_Request 功能和 Read_Request 功能。Write_Request 函数在调用过程中使用辅助函数 Make_Write_Request 函数和 Make_Data_Package 函数实现向服务器端上传文件。Read_Request 函数在调用过程中使用辅助函数 Make_Read_Request 函数和 Make_ACK 函数实现在服务器端下载文件。

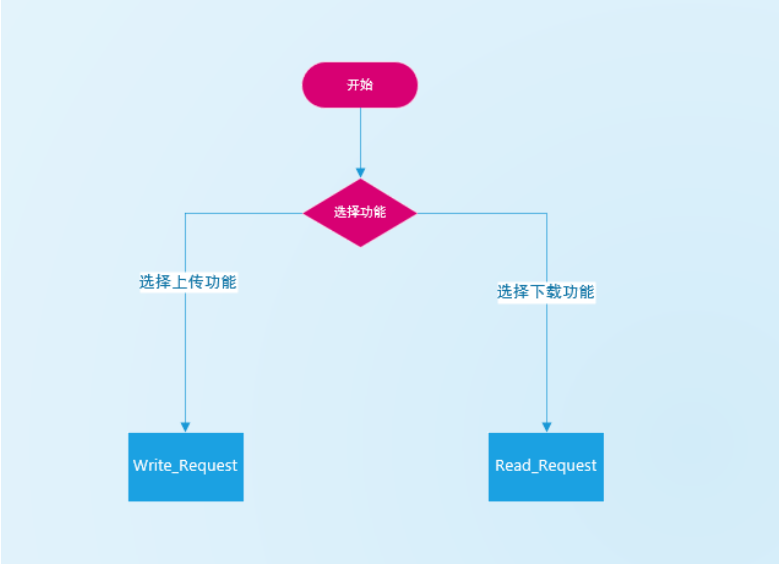

图 1.1　模块功能

模块关系：

对于两大模块从服务器下载文件和向服务器上传文件，这两个大模块是并列的，但其中封装调用的内部函数是有重叠的，两个模块都要先向服务器的 69 好端口发起连接（伪连接）请求，而后收到请求响应后再执行对应的操作。

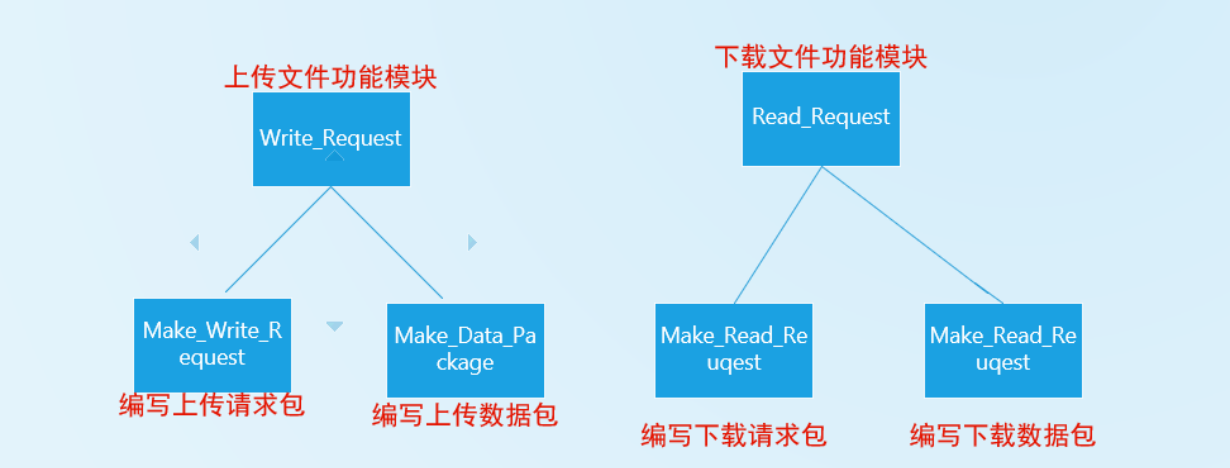

图 1.2　模块关系

接口设计描述：

在编程的过程中为了实现系统的两大模块功能，编写了一些函数接口供给使用，以下

表格就是核心函数的参数、返回值和功能：

表 1.1　函数接口描述

函数名	输入参数	返回值	功能
Timeshow	无	无	在日志文件中输出时间
Write	Receive 数组：及收到的数据 数据长度 l	无	实现 linux 版本服务器的写操作
Make_Write_Request	Buffer 数组：存储请求包的数组	返回写请求的请求包大小	生成一个写请求包
Make_Data_Package	Buffer 数组：存储数据包数据 Block_Num：数据的块号	返回数据包的字节大小	将输入的 Buffer 数组填充数据包数据
Make_Read_Requesr	Buffer 数组：存储请求包的数组	返回读请求的请求包大小	生成一个读请求包
Make_ACK	Buffer 数组：存储返回的 ACK 报文 block：数据的块号	无	生成接收块号对应的确认报文
Main　主程序	无	无	调用对应函数让用户进行功能选择

1.4.2 系统数据处理流程

上传数据流程：

首先在主程序中使用 WSAStarup 函数将 Winsock 初始化，并在每一次循环的开始设置服务器端的端口为 69，根据地址创建对应的 socket。

用户进入上传功能模块提示输入上传文件以及上传数据方式。在 Make_Write_Request 函数中根据用户输入的信息编写对应的写请求报文，Opcode 为 1，文件名为用户输入文件名，数据类型由用户输入决定。并返回请求包的数据大小，这个数据大小必须要获取，因为使用 sendto()函数的第二个参数必须得是发送文件的实际数据大小。

当服务器端接收到写请求报文后回应确认报文，并给服务器端分配端口，之后服务器和客户端就在这个端口上进行相应的数据上传。客户端每收到一个返回包就解析返回包的类型和含义。

下载数据流程：

在完成相应的初始化之后，用户输入下载的文件名和数据类型，在 Make_Read_Request 函数中完成读请求报文，随后在 Read_Request 函数中发送读请求报文。解析服务器返回的数据包，并根据返回包的类型做出相应的动作。

1.4.3 系统详细设计

文件上传：

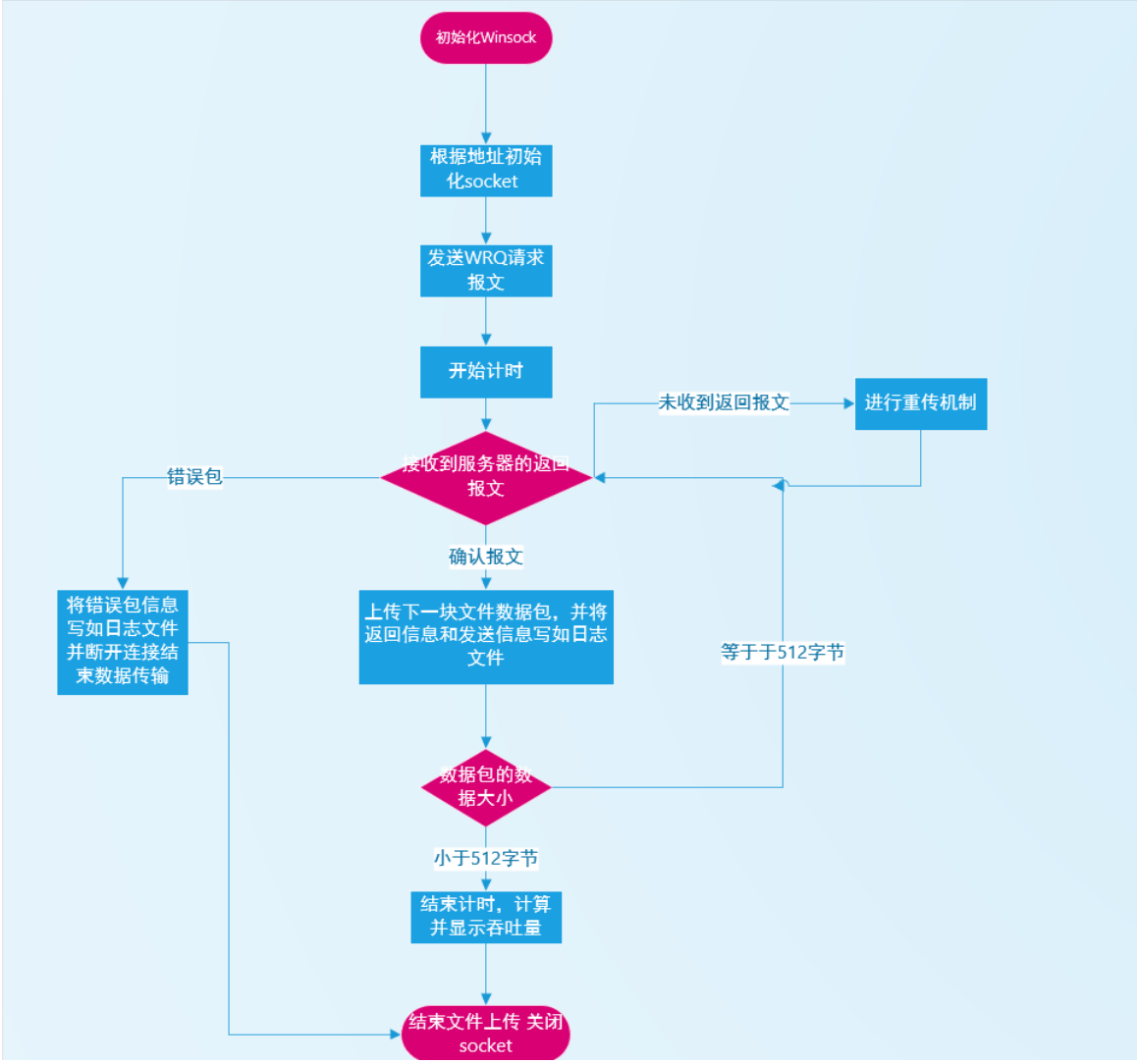

图 1.3　文件上传功能流程

说明：

在用户进行功能选择前初始化 Winsock 并根据地址设置客户端的 socket，同时使用 bind()完成 socket 的绑定。客户端向服务器发起写请求数据包，用户根据服务器返回的数据包进行相应的操作。

如果用户在设定时间内接收到服务器端的确认报文，那么调用 Make_Data_Package()函数生成下一块数据包并发送给服务器，同时将接收到的确认报文信息和发送的数据包信息写日

志文件中，更新数据吞吐量。

如果用户在设定时间内还未接收到服务器的返回报文（确认报文或错误报文），那么进行重传机制，将上一次发送的数据包再次发送给服务器，同时在日志文件中记录进行的动作。

如果接收到错误包，那么打印出错误信息并将错误信息记录在日志中，同时断开连接（伪连接），关闭 socket。

文件下载：

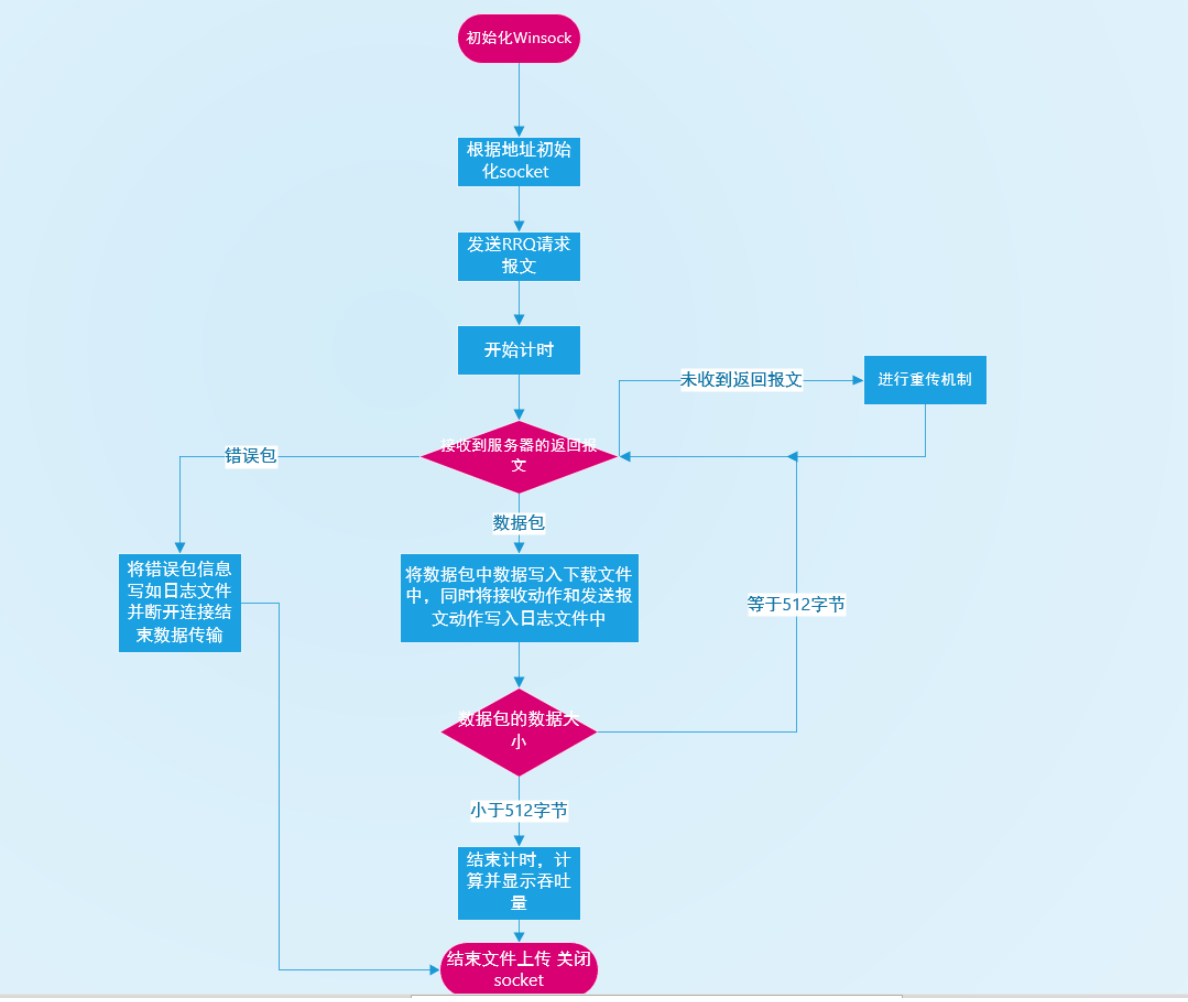

图 1.4　下载文件流程

说明：

下载文件和上传文件一样，对于从服务器端接收的数据包进行解析，分析数据包的类型做相应的动作即可。

1.4.4 系统测试方案

测试方案：

1. 不开启 clumsy 应用进行上传下载测试

2. 尝试上传客户端不存在的文件
3. 尝试下载服务器端不存在的文件
4. 开启 clumsy 应用进行上传下载测试
5. 分别选择 ASCII 码格式和二进制格式上传下载文件

1.5 系统测试及结果说明

在不开启 clumsy 应用进行上传或下载文件操作，文件的下载上传速度大约几百 kBps，当开启 clumsy 应用进行上传下载操作的时候只有几 kBps 的速率。主要的原因是为了实现丢包重传机制、超时重传机制，将 SOCKET 的 I/O 设置为了非阻塞式 I/O 模式，同时每次的接收操作后都使用 Sleep 函数停等 2s 后继续接收，在设置的次数内没有接收到服务器的数据包那么进行重传。这样子即使在没有丢包的情况下也得多停等 2s，同时发现在控制台的打印信息的时间也被记录在吞吐量时间的统计中，这样使得计算所得的速率远远的小于真实的速率。在未设置重传机制之前，文件操作的吞吐量大约未 8~9MBps。

在上传客户端不存在的文件时，会提示客户端不存在该文件，并退出程序，返回值为-1。如果下载服务器端不存在的文件时，则会解析服务器端返回的错误包，打印出 File not Found!，然后断开连接。

本程序能够正确的完成 ASCII 码和二进制的上传和下载机制，下载的文件都能够正常打开。在开启 Clumsy 下使用二进制下载 EasyX.exe，能够正确的打开该应用文件：

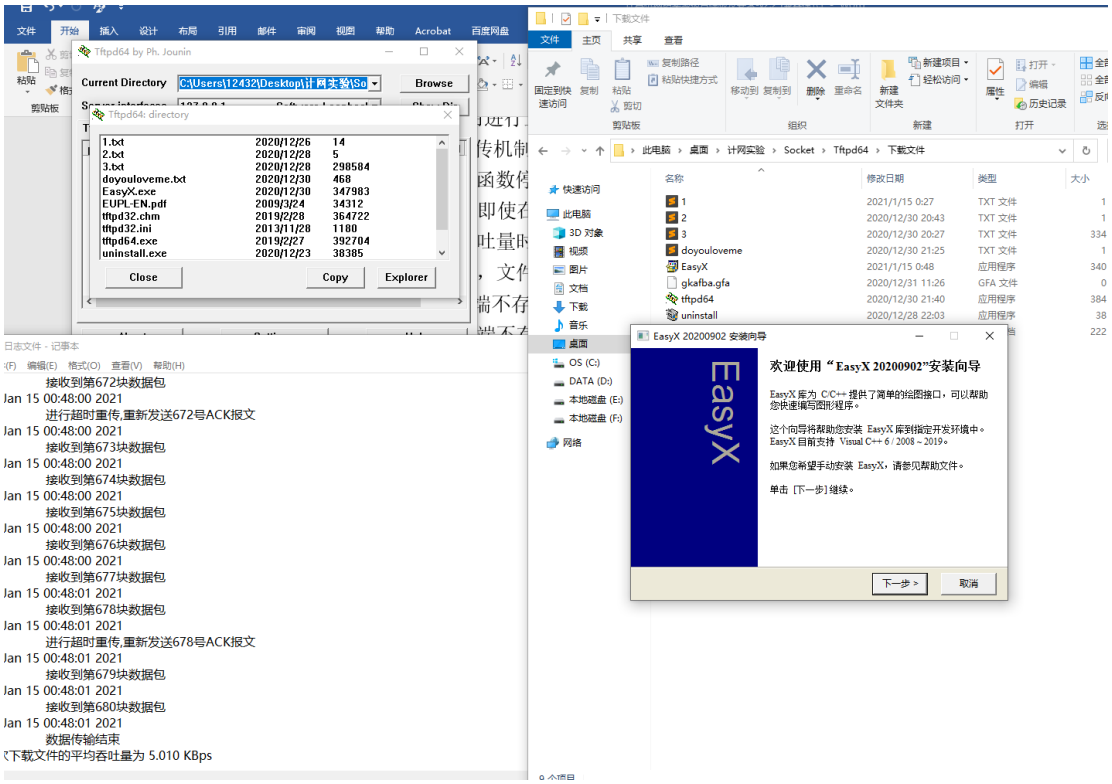

图 1.5 测试结果

1.6 遇到的问题及解决方法

1. 在实验过程中第一个遇到的问题就是对接口函数参数的定义不是很明确，在初步编写下载请求的时候，明明写出的函数调试几次都没发现任何问题，同时使用 wireshark 抓包也能发现自己发送出去的请求，但是始终无法接收到服务器端的回应。在查阅资料的时候，发现了一位博主遇到过和我一样的问题，这是对 sendto 函数参数的理解错误，sendto 函数的第二个参数一定要是实际发送给服务器的报文大小，而不是数组的大小。

2. 在解决丢包问题的时候，一开始的想法是使用多线程中的信号量来解决问题，同时使用多线程进行实验。对于出现丢包状况，使用一个信号量来对等待时间进行判断，只有当超时时，使用 ReleaseSemaphore 函数将信号量 Mutex 加一，这时才进行重传。但是在这里使用的是阻塞的 I/O，一旦主程序的等待时间超时，即使进行了重发，也还是阻塞在那里不能够接收后继续执行。随后在老师所发的指导手册上看到了非阻塞 I/O 和阻塞 I/O 的说明后，使用 ioctlsocket 函数将 socket 设置为非阻塞 I/O，同时利用 Sleep 函数在每次接收后停等 2s。这样进行循环就能实现超时重传机制。但是这样的方法还是有些不足的地方，就是大大的降低了上传、下载时的吞吐量。

3. 一开始就是使用简单的 fopen()、fwrite()两个函数进行 ASCII 和二进制读写的实现，但是在查看使用 ASCII 下载的文件时，发现文件每行都会多一个换行符，这其实就说明没有完全实现 ASCII 码格式的传输。在使用十六进制格式查看文件内容后，猜测服务器可能是在 Linux 系统下实现的，使用的是‘\n’(0x0A)表示换行，而 windows 版本下的换行是以’\r\n’（0x0D0A）来表示的。于是在进行读操作的时候对读取的字符进行判断，而后再进行转化即可实现文本的一致下载。

2 局域网组网实验

2.1 环境

设备名称	设备型号	默认用户名	密码
无线 AP	华为 AP4050DN	Admin	admin@huawei.com
路由器	华为 AR1220C	Admin	Admin@huawei
三层交换机	-华为 S5720-28X-LI-AC	admin	admin@huawei.com
二层交换机	-华为 S5720-28P-LI-AC	admin	admin@huawei.com
	24 端口网络配线架		

2.2 实验要求

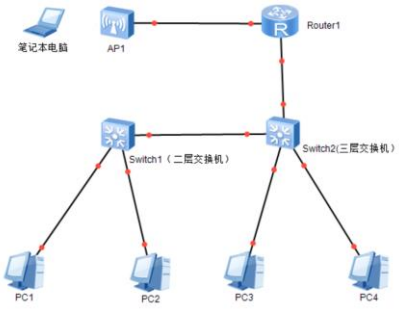

图 2.1

利用华为真实网络设备，完成图 2.1 所示的网络配置，实现如下组网要求：

（1） PC1 与 PC3 在同一个 VLAN（VLAN2）中，能互相访问；

（2） PC2 与 PC4 在同一个 VLAN（VLAN3）中，能互相访问；

（3） 通过配置交换机和路由器，使得 PC1、PC2、PC3、PC4 可以相互访问；

（4） 笔记本电脑通过 AP 无线上网，跟 PC1、PC2、PC3、PC4 之间互通。

2.3 实验步骤说明

1. 检查电源，连接设备

首先检查实物机的各个设备的电源是否连接好，而后使用网线和 console 线按照指导手册的图片进行连接。连接完后，查看各个接口是否有绿色指示灯正常亮起或者闪烁，由此判断接口是否出问题、电源是否接好。

2. 配置 PC 端 IP

首先在设备管理器中禁用 PC 端的一个网卡，而后在网络管理中打开设备的 IPV4 设置进行配置。配置的主机 IP、子网掩码、网关如下表所示：

表 2.1　主机配置

设备名	ip 地址	子网掩码	网关
PC1	192.168.2.2	255.255.255.0	192.168.2.1
PC2	192.168.3.2	255.255.255.0	192.168.3.1
PC3	192.168.2.3	255.255.255.0	192.168.2.1
PC4	192.168.3.3	255.255.255.0	192.168.3.1

3. 配置交换机

设备加电以后，我们主要采用串口线连接设备的 console 口来进行管理。将串口线的水晶头那一端连上设别的 console 口，另一端接入 PC 的 USB 接口。在设备管理器里面查看 COM 口的编号。启动串口终端管理软件 secureCRT 连接对应的 com 口，波特率设置为 9600，然后启动设备，输入初始化账号密码。

连接后进行交换机配置，各项配置指令在指导手册中已经给出。

4. 配置路由器

路由器的配置过程与交换机一致，按照指导手册即可。

5. 配置 AP

6. 在各主机 PC 上相互 ping 进行测试

在连接完成后，在四台主机上互相 ping，根据结果判断是否正确地完成各个设备的配置。而后使用自己的主机连接上配置的 AP 无线网，测试跟 PC1、PC2、PC3、PC4 之间互通性。

2.4 结果分析

在配置路由器前，由于交换机中配置了两个 VLAN，分别是 192.168.2.0/24 和 192.168.3.0/24，VLAN 内的 PC 可以互通，即 PC1 与 PC3（在 VLAN2 中）可以互通，PC2 与 PC4（在 VLAN3 中）可以互通，而子网间不互通。

在配置路由器后和交换机，实现了不同网络的连接，四台 PC 间任意两台 PC 可以互相通信。

无线路由器 AP 配置后，使用个人笔记本电脑连接到该无线网络，测试与四台 PC 的互通性，个人笔记本连接该网络后均能实现对四台主机的通信，实验成功。

2.5 遇到的问题及解决方法

在第一次实验的时候，遇到一个很傻的问题，我们组将所有设备全部配好了，在secureCRT上也完成了配置，但是在四台主机上进行ping指令测试的时候两个VLAN的主机始终不能完成互通。而后我们开始怀疑是设备出现了问题，果然发现一个路由器的接线口的灯没有亮。在检查了一遍各个设备的电源连接后，发现这个指示灯没有亮的路由就是后面的电源没有接好。一定要在完成设备连接后查看各个指示灯是否闪烁，避免因电源连接问题耽误时间。

在实验过程中也出现了接口的未匹配问题，所以一定要注意各个路由器和交换机的端口号要与输入命令中所配置的端口号一致，适配器的 POE 接口连接无线路由器 AP 的 POE 接口，DATA 接口连接路由器对应接口。

3 广域网组网实验（路由器配置）

3.1 环境

华为虚拟仿真平台 eNSP。

3.2 实验要求

1. 路由器实验

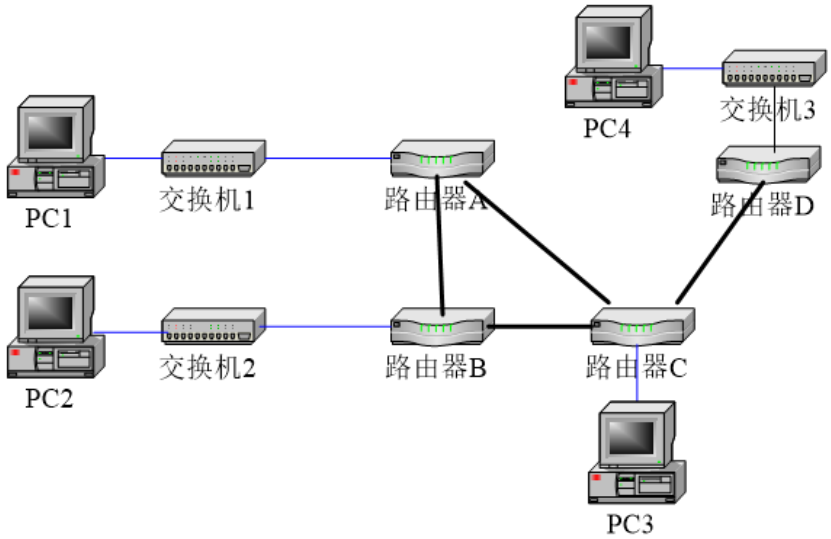

图 3.1

根据图 3.1 的拓扑结构进行组网，要求如下：

（1） PC1：192.168.1.0/24 网段；PC2：192.168.2.0/24 网段；PC3：192.168.3.0/24 网段；PC4：192.168.4.0/24 网段

（2） 路由器上配置 RIP 协议，使各 PC 机能互相访问

（3） 路由器上配置 OSPF 协议，使各 PC 机能互相访问

（4） 路由器上配置 RIP/OSPF 协议，使各 PC 机能互相访问之后，再对对路由器 1 进行访问控制配置，使得 PC1 无法访问其它 PC，也不能被其它 PC 机访问

（5） 路由器上配置 RIP/OSPF 协议，使各 PC 机能互相访问之后，再对对路由器 1 进行访问控制配置，使得 PC1 不能访问 PC2，但能访问其它 PC 机

3.3 路由配置实验简要说明及结果分析

实验设计：

按照图 3.1 创建拓扑图连接设备，各个端口的 ip 地址都分配在 192.168.0.0/16 网络范围内，

其中路由器 B、C、D 都使用 router，A 路由器使用 AR2240。

实验步骤：

1. 绘制拓扑图

交换机选择，选择交换机 S5700，共三台。路由器选择，路由器 B、C、D 都使用 router，A 路由器使用 AR2240。他们的连接线选择的是普通的双绞线。将每一个设备和交换机按照图 3.1 所示的拓扑图连接。

2. 配置主机网络

各主机的网络配置如下表所示：

表 3.1　PC 参数配置表

主机名	ip 地址	子网掩码	网关
PC1	192.168.1.1	255.255.255.0	192.168.1.2
PC2	192.168.2.1	255.255.255.0	192.168.2.2
PC3	192.168.3.1	255.255.255.0	192.168.3.2
PC4	192.168.4.1	255.255.255.0	192.168.4.2

3. 配置路由器各端口 IP 配置如下：

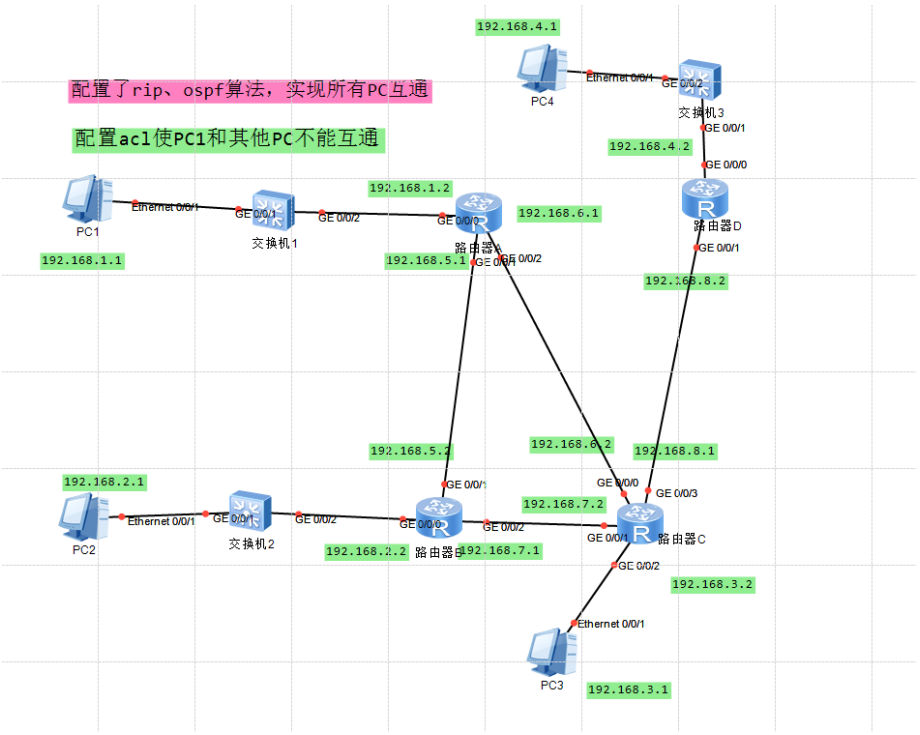

图 3.2　拓扑图

4. 配置 rip 或 ospf 协议

配置路由协议使路由器各个端口间的地址可以互访，从而使不同网络号的设备可以互访。Rip 路由协议（左）和 ospf 路由协议（右）配置指令如下：

	ospf 1 router-id 192.168.1.2
rip	area 0
network 192.168.1.0	network 192.168.1.0 0.0.0.255
network 192.168.5.0	network 192.168.5.0 0.0.0.255
network 192.168.6.0	network 192.168.6.0 0.0.0.255

5. 设置访问控制策略

在本实验中需要实现两个访问控制策略，策略一是使得 PC1 无法访问其它 PC，也不能被其它 PC 机访问；策略二是使得 PC1 不能访问 PC2，但能访问其它 PC 机。显然两个策略是不能同时实现的，所以以下给出两次配置访问控制策略的相关配置表：

表 3.2 访问策略一配置表

规则号	配置路由	功能	指令
3001	路由器 A	过滤所有来自于 192.168.1.0/24 的数据包（PC1 不能访问其他 PC）	rule deny ip source 192.168.1.0 0.0.0.255
3002	路由器 A	过滤所有到达 192.168.1.0/24 的数据包（PC1 不能被其他 PC 访问）	rule deny ip destination 192.168.1.0 0.0.0.255

表 3.3 访问策略二配置表

规则号	配置路由	功能	指令
3001	路由器 A	过滤所有从 192.168.1.0/24 到达 192.168.2.0/24 的数据包（PC1 不能访问 PC2）	rule deny ip source 192.168.1.0 0.0.0.255 destination 192.168.2.0 0.0.0.255

结果分析：

在配置完 rip 或 ospf 路由协议之后，除了第一次 ping 可能超时，之后整个拓扑图的所有 PC 主机都能都实现互通。这可能就是路由协议的学习过程，第一次进行 ping 的时候路由表都是空的，但在自学习后，路由算法填入信息进入路由表。

完成访问控制策略一之后 PC1 主机不能和任何其他的 PC 互通，完成访问控制策略二后 PC1 不能访问 PC2，但是能访问其他的 PC，其他 PC 也能访问 PC1。

遇到的问题及解决方法：

问题 1：在配置 rip/ospf 路由算法的时候误以为能够同时配置两个算法，然后进行选择。实际上在配置完后，两个算法只会使用一个。

解决方案：重新一个一个的配置了相应的算法。

4 综合组网实验

4.1 环境

华为虚拟仿真平台 eNSP。

4.2 实验要求

某学校申请了一个前缀为 211.69.4.0/22 的地址块，准备将整个学校连入网络。该学校有 4 个学院，1 个图书馆，3 个学生宿舍。每个学院有 20 台主机，图书馆有 100 台主机，每个学生宿舍拥有 200 台主机。对这些主机进行组网，要求：

（1） 图书馆能够无线上网（选做，AP+AC 配合使用）；

（2） 学院之间可以相互访问；

（3） 学生宿舍之间可以相互访问；

（4） 学院和学生宿舍之间不能相互访问；

（5） 学院和学生宿舍皆可访问图书馆。

过程要求：

（1） 网络拓扑结构的设计并在仿真软件上进行绘制；

（2） 要求具有足够但最少的设备，不需要考虑设备冗余备份的问题；

（3） 对全网的 IP 地址进行合理的分配；

（4） 在绘制的网络拓扑结构图上对各类设备进行配置；

（5） 测试是否满足组网需求，如有无法满足之处，请结合理论给出解释和说明。

4.3路由配置实验步骤说明及结果分析

实验设计：

首先根据实验要求进行子网分配，给各个区域分配相应的子网和满足条件的 IP 地址数目，各区域的网络分配表如下：

表 4.1 网络配置表

区域	子网网段	子网掩码	网关	IP 地址数目
学生宿舍 1	211.69.5.0/24	255.255.255.0	211.69.5.1	256
学生宿舍 2	211.69.6.0/24	255.255.255.0	211.69.6.1	256
学生宿舍 3	211.69.7.0/24	255.255.255.0	211.69.7.1	256
图书馆	211.69.4.0/25	255.255.255.128	211.69.4.1	128
学院 1	211.69.4.128/27	255.255.255.224	211.69.4.129	32

学院 2	211.69.4.160/27	255.255.255.224	211.69.4.161	32
学院 3	211.69.4.192/27	255.255.255.224	211.69.4.193	32
学院 4	211.69.4.224/27	255.255.255.224	211.69.4.225	32

交换机选择，选择交换机 S5700，共八台。路由器选择，三台 AR2200，连接线选择的是普通的双绞线。

实验步骤：

1. 绘制拓扑图

按照自己的思路画出相应的拓扑图，具体结构如下：

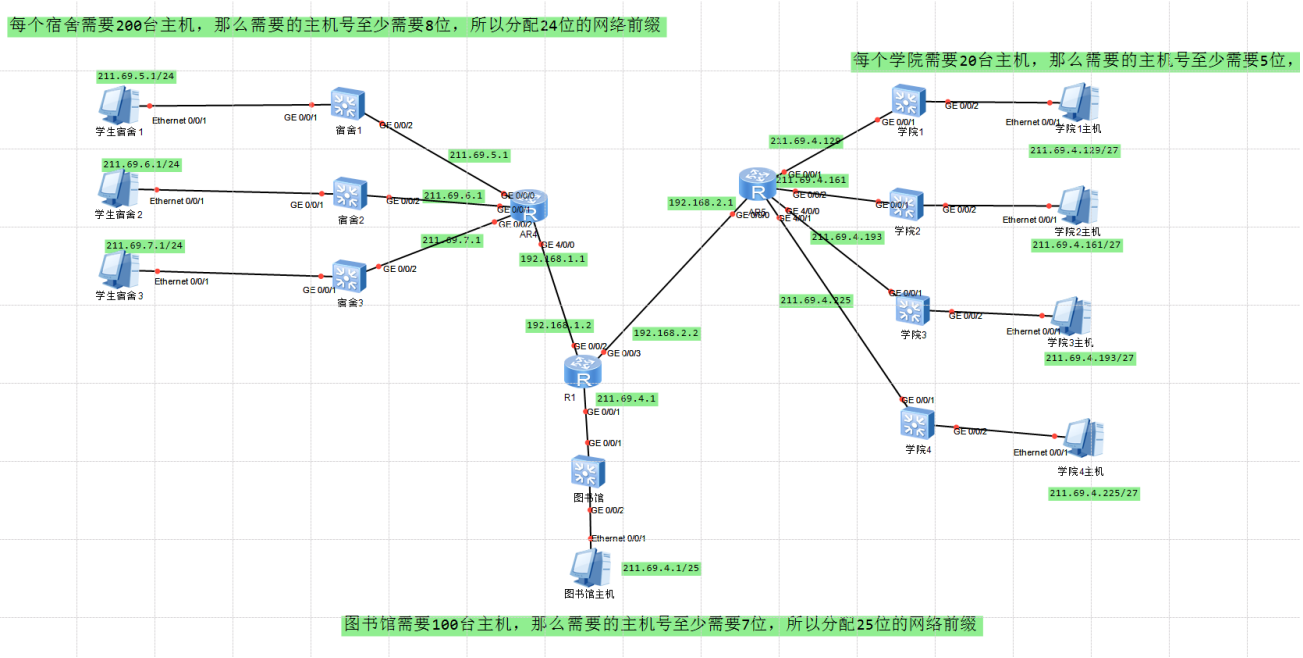

图 4.1　拓扑图

2. 配置端口IP

将实验设计实现的对应的子网划分到每一个区域，并选取一个样例IP地址赋予PC机，并打开并配置的路由器的端口IP，各端口的配置结合表4.1和图4.1。

3. 配置路由算法

打开各个区域的路由器，并配置ospf路由算法，使得各个子网能够相互访问。进行相互ping的操作，保证各个区域的网络在目前是可以相互联通的。

4. 添加访问控制策略

编写限制条件ACL，为了完成宿舍与学院之间的不可相互连接操作。与宿舍直接连接的路由器中加入限制条件限制，限制从学院发往宿舍的数据包。同时，在与学院直接相邻的路由器中设置限制条件，禁止从宿舍发来的数据包，这样我们就可以保证宿舍与学院之间不能相互连接，同时宿舍与学院可以与图书馆进行连接。具体配置如表4.2所示：

在配置完成后，在宿舍和学院之间相互发送ping指令，测试访问控制策略的配置是否成功。

表 3.3　访问策略二配置表

规则号	配置路由	功能	指令
3001	学生宿舍路由器	过滤所有发送到子网 211.69.4.128/25 的数据包（拒绝宿舍访问学院）	rule deny ip destination 211.69.4.128 0.0.0.127 int g0/0/0
3002	学院路由器	过滤来自 211.69.4.128/25 网络，发送至 211.69.5.0/24 到 211.69.7.0/24 的数据包（阻止学院的访问）	rule deny ip source 211.69.4.128 0.0.0.127 destination 211.69.5.0 0.0.0.255 rule deny ip source 211.69.4.128 0.0.0.127 destination 211.69.6.0 0.0.0.255 rule deny ip source 211.69.4.128 0.0.0.127 destination 211.69.7.0 0.0.0.255

结果分析：

在配置完各个端口 IP 和路由器算法后，所有的主机都能互通。而后配置完两个访问控制策略，使得宿舍和学院之间不能相互访问，同时其他的所有主机均能互相访问。

遇到的问题和解决方法：

问题 1：在使用 ESpn 的过程中，开启路由器的速度实在太慢了。

解决方法：尝试停止再开启，能够很好的解决等待太久但还是未能开启路由器的问题。

问题 2：在实验过程中明明 save 了指令，但是再次进入配置的指令没有能够保存。

解决方法：保存配置文件 + 存放配置指令在文档中。

五 感想与建议

5.1 感想

在第一次的实验在机房中做网线，其实现在想想当时做网线还是挺有趣的，最后做出了一根 8 个全通的网线，一根通四个的网线，尤其是和同组同学一起去做网线感觉很好玩。

在第二次的实验中，我们小组进行事物设备的连接，第一次做的时候不是很懂，而且老师的理论课刚刚讲完不是对于一些原理没有太了解，在连线的时候也连错了。不过后来尝试着弄了 eNSP 的模拟实验，对着拓扑图和课本去理解虚拟局域网和路由算法真的确实学到很快很深刻。第二次去机房再做事物实验的时候就很顺利的完成了。

在做组网实验的时候，感觉对着课本ppt来一边做一边学习真的是很有帮助，在做组网实验的时候查看路由表信息，来体会路由算法的自学习。在这个实验中，我对在交换机中VLAN的理解更加深刻。VLAN可以把同一个物理网络划分为多个逻辑网段。即便在同一个交换机上，处于不同VLAN的端口也是不能通信的。这样一个物理的交换机可以当作多个逻辑的交换机使用。

在贯穿整个实验中的 Socket 编程中对于 Winsock 中内置的几个 API 函数有了比较深刻的了解（主要是错了之后查了很久）。在实验的过程中查阅了很多的资料，尤其是发生错误时，在网上即使错误号相同大家的解决方法也不一样，甚至使用网友的解决方案不一定能够解决自己的问题。在整个的编写过程中我认为有难度的就是实现重传机制，我编写的重传机制是在非阻塞 I/O 下完成的，只要设置循环次数和停等时间就能够实现 rdt 中的超时重传。在查阅资料的过程中也学习了其他的很多网络层的相关知识，对于每一次的数据交换也有了深刻的理解。

最后，各位老师的指导与讲解都十分认真细致，老师们都好有耐心的帮助我们解决问题！不管是课堂还是实验期间都学到了很多，在这里感谢学院提供的实验平台与材料，也感谢老师的教导！

5.2 意见和建议

感觉可以将实物的组网配置和 eNSP 的组网练习调整一下次序，感觉在对配置指令和相应的知识点学习过后再去用实物去做会很有意思。

再就是建议老师们在做网线的实验课结束的时候提醒一下同学们收拾一下垃圾，感觉这样大家应该都会去收拾的（可能是有些同学忘记了）。