

华中科技大学
网络空间安全学院

《计算机通信与网络》课程设计报告

姓 名 钱俊卓
班 级 本硕博 2101

学 号 U202112213

联系方式 18224607294

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（段落、行间距、缩进、图表、编号等）		10		
感想（含思政）		10		
意见和建议		10		
验收时间		10		
Socket 编程	代码可读性	10		
	注释	10		
	软件体系结构	30		
	问题描述及解决方案	10		
实验报告总分		100		
教师签名			日 期	

目 录

目录

一、 实验概述	1
1.1 实验名称	1
1.2 实验目的	1
1.3 实验环境	1
1.4 实验内容	1
1.5 实验要求	2
二、 实验过程	1
三、 实验测试与分析	1
3.1 系统测试及结果说明	1
3.2 遇到的问题及解决方法	3
3.3 设计方案存在的不足	3
四、 实验总结	4
4.1 实验感想	4
4.2 意见和建议	5

Socket 编程实验

一、 实验概述

1.1 实验名称

Socket 编程实验。

1.2 实验目的

通过 socket 程序的编写、调试，了解计算机网络可靠传输协议，熟悉基于 UDP 协议的 socket 编程方法，掌握如何开发基于 TCP/UDP 的网络应用。

1.3 实验环境

操作系统：Windows/Linux

编程语言：C，C++

1.4 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） 能够实现两种不同的传输模式 netascii 和 octet；
- （3） 能够将文件上传到 TFTP 服务器；
- （4） 能够从 TFTP 服务器下载指定文件；
- （5） 能够向用户展现文件操作的结果：文件传输成功/传输失败；

- （6） 针对传输失败的文件，能够提示失败的具体原因；
- （7） 能够显示文件上传与下载的吞吐量；
- （8） 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；
- （9） 人机交互友好（图形界面/命令行界面均可）；
- （10） 额外功能的实现，将视具体情况予以一定加分。

1.5 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。

二、 实验过程

2.1 系统结构设计

1、模块框图如图 2-1 所示

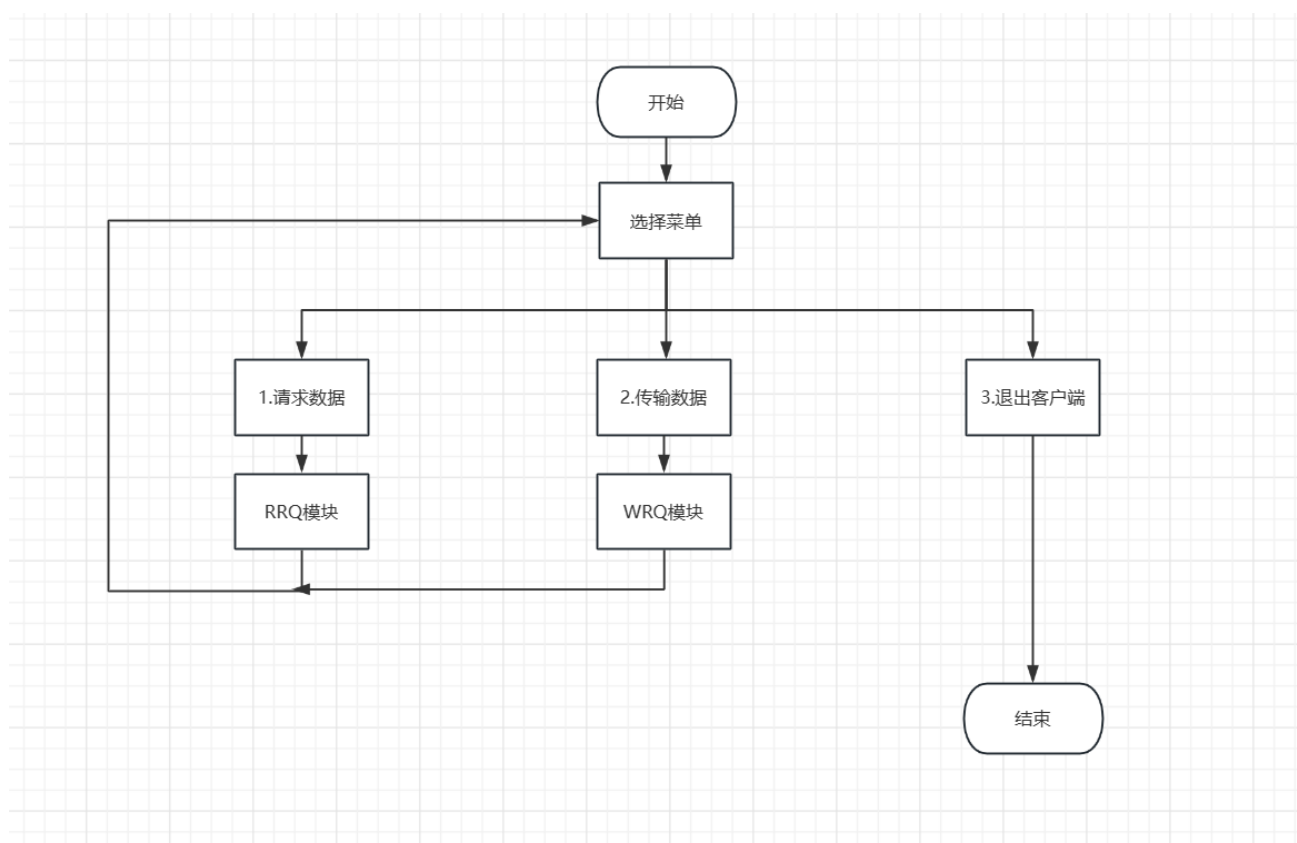


图 2-1 模块框图

2、模块功能说明

菜单模块是一个永真循环，其功能为循环接收用户输入的指令、选择数据收发模式，达到收发数据的功能，并且可以手动退出客户端。

RRQ 模块功能：向服务端发送读取请求包，并且在收到回应包之后开始接收数据包，将数据写入指定文件中，在收到结束包之后退出该模块，该模块设计了超时重传以及最大重传次数，确保了可靠传输。

WRQ 模块功能：向服务端发送写入请求包，并且在收到回应包之后构造数据包并开始向服务端发送数据包，将数据写入指定文件中，在收到结束包之后退出该模块，该模块设计了超时重传以及最大重传次数，确保了可靠传输。

3、模块之间的接口说明

主菜单（main）分别通过 RRQ 与 WRQ 函数调用 RRQ 模块和 WRQ 模块。接口参数均为 type（数据传输的类型）以及 connectSocket(创建的套接字)。

4、数据处理流程

读取数据：

- ① 打开文件
- ② 将数据包的数据读入缓冲区（最多为 512 字节）
- ③ 将缓冲区数据写入文件

写入数据：

- ① 打开文件
- ② 将文件中的数据读取到缓冲区
- ③ 用缓冲区数据构造数据包
- ④ 发送

2.2 详细设计

1、核心函数流程

RRQ 函数流程如下图 2-2 所示：

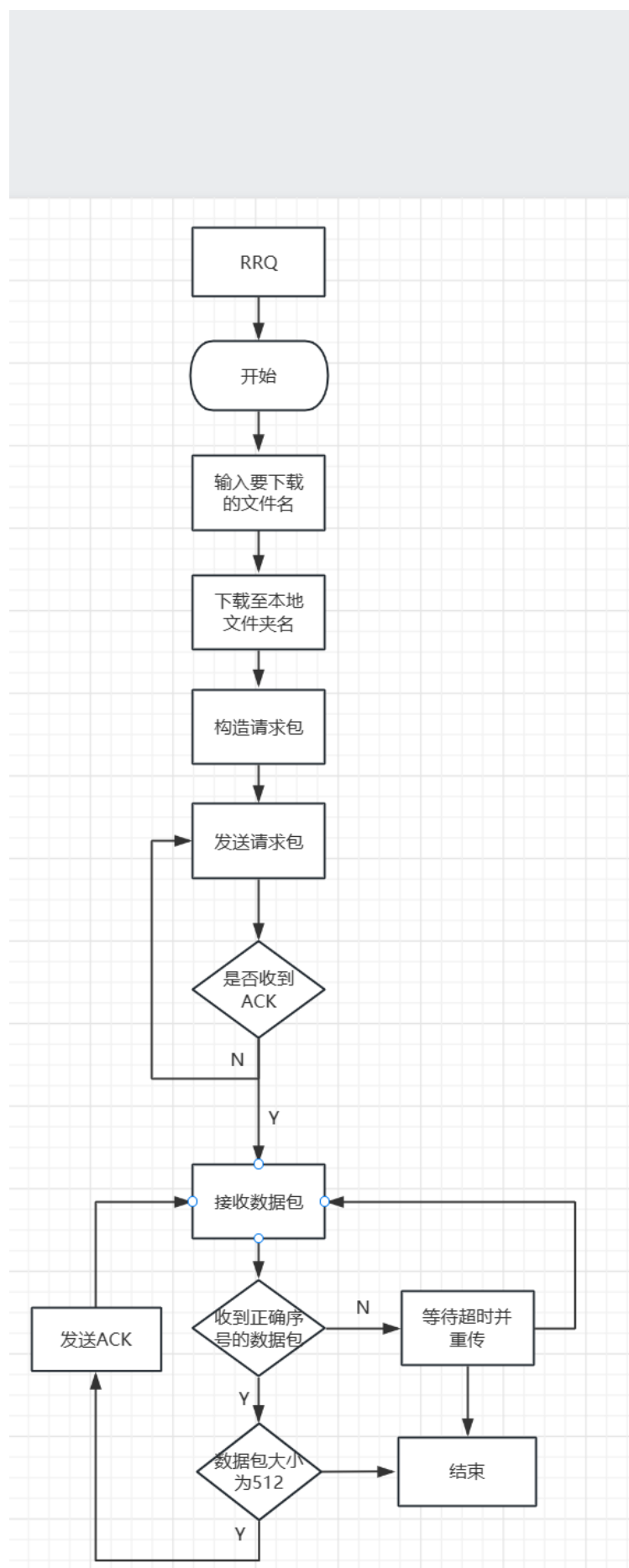


图 2-2 RRQ 流程图

WRQ 函数流程图如图 2-3 所示：

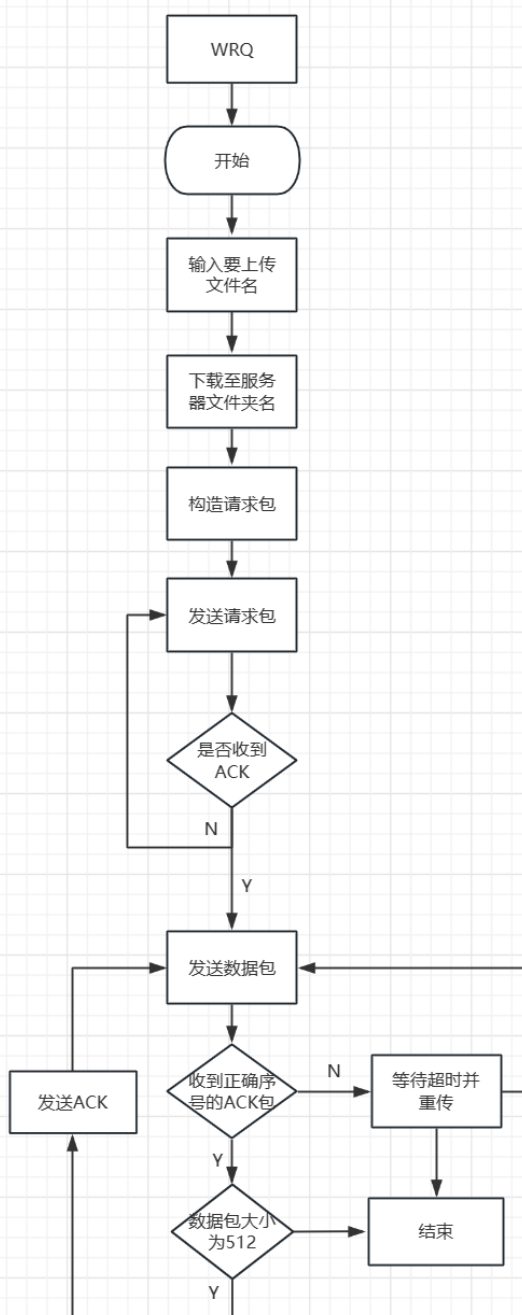


图 2-3 WRQ 流程图

2、核心数据结构

TFTP 共定义了五种类型的包，包的类型由数据包前两个字节确定，我们称之为 Opcode（操作码）字段。这五种类型的数据包分别是：

- 读文件请求包：Read request，简称为 RRQ，对应 Opcode 字段值为 1
- 写文件请求包：Write request，简称为 WRQ，对应 Opcode 字段值为 2
- 文件数据包：Data，简称为 DATA，对应 Opcode 字段值为 3
- 回应包：Acknowledgement，简称为 ACK，对应 Opcode 字段值为 4
- 错误信息包：Error，简称为 ERROR，对应 Opcode 字段值为 5

RRQ 和 WRQ 的数据包格式一样，只不过某些值域设置有差别，剩下的三种数据包格式各不相同。

（1）RRQ 和 WRQ 数据包的格式，首先是 2 字节表示操作码，它用来表示当前数据包的类型（取值 1 表示该数据包是个读请求，2 表示该数据包是写请求）；接下来是可变长字段，它用来表示要读取或上传的文件名，它使用 ASCII 码并以 0 表示结尾；第三个字段叫 Mode，也是可变长字段，用来表示传输文件的数据类型，如果传输的是字符串文件，那么它填写字符串“netascii”，如果传输的是二进制文件，那么它填写字符串“octet”，这些字符串都以 0 结尾，其结构如图 2-4 所示：

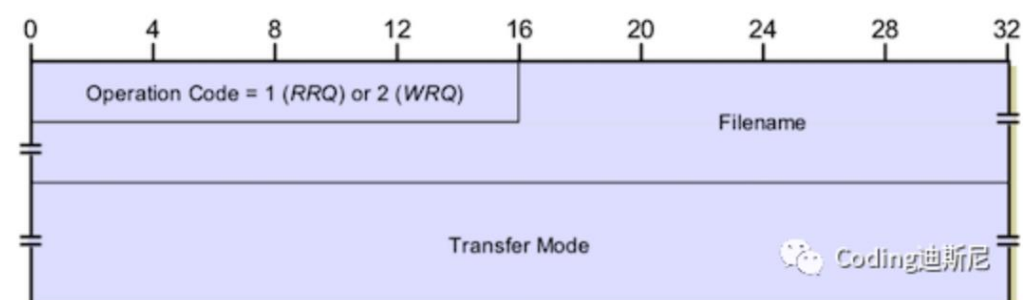


图 2-4 TFTP 协议 RRQ/WRQ 数据报格式

（2）传输数据块的 DATA 数据包，它头 2 字节也是操作码，取值 3 用于表示数据包用于数据块传输，接下来的 2 字节用于表示数据块编号，最后是可变长字段 Data，用于装载数据块，

该数据包的格式如图 2-5 所示：

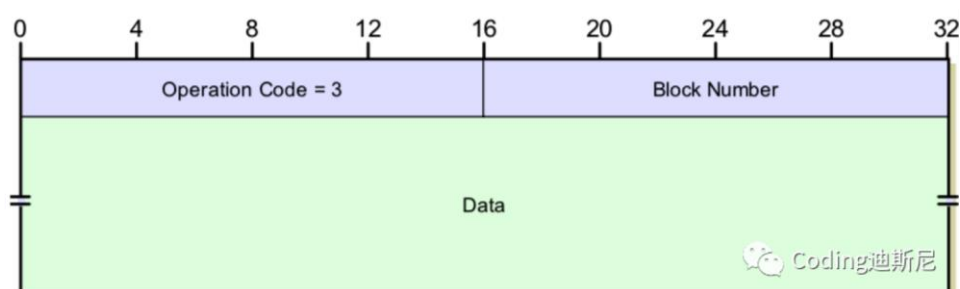


图 2-5TFTP 协议 DATA 数据包格式

(3) 应答 ACK 数据包，它开始的 2 字节也是操作码，取值 4；接下来 2 字节表示接收到的数据块编号，相应结构如图 2-6 所示：

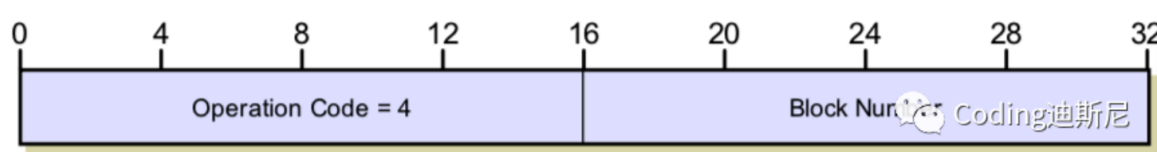


图 2-6TFTP 协议 ACK 数据包格式

(4) 错误 ERROR 数据包，它开始的 2 字节表示操作码，取值 5；接下来 2 字节表示错误码；最后的是可变长字段，它用字符串的形式描述具体错误，该数据包的结构如图 2-7 所示：

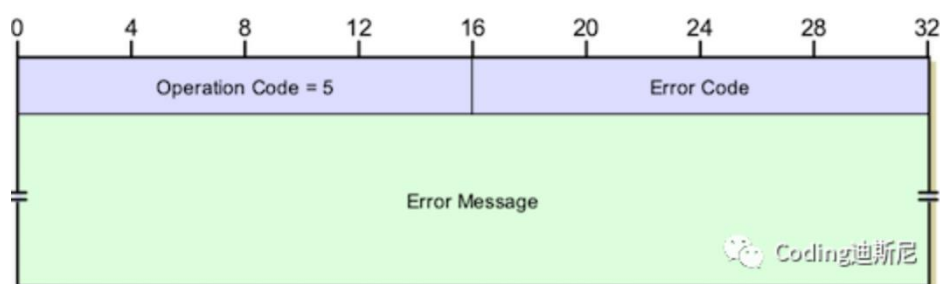


图 2-7 TFTP 协议 ERROR 数据包格式

2.3 代码实现

以下为 RRQ 代码及解释，WRQ 同理：

```
void RRQ(string type, SOCKET connectSocket){
```

```
    sockaddr_in serverAddress;
```

```
    //将服务端信息存储在结构体中
```

```
int size = sizeof(serverAddress), bytesnum, num = 1, bytesRead;

serverAddress.sin_family = AF_INET;

serverAddress.sin_addr.s_addr = inet_addr(IP_ADDRESS);

serverAddress.sin_port = htons(PORT_NUMBER);


//构造请求包

char buf[516], buf1[516];

string filename, filename1;

cout << "请输入要下载的文件名:";

cin >> filename;

cout << "下载至:";

cin >> filename1;

ZeroMemory(buf1, sizeof(buf1));

buf1[1] = READ_REQUEST;

strcpy(buf1 + OP_LEN, &filename[0]);

if(type == OCTET || type == NET_ASCII) strcpy(buf1 + OP_LEN +
filename.size() + 1, &type[0]);

else {

    printf("illegal type\n");

    return ;

}
```

```
int sendNum = 1;
```

```
//发送请求包
```

```
sendto(connectSocket, buf1, 2 + filename.size() + type.size() + 2, 0, (struct  
sockaddr*)&serverAddress, size);
```

```
//打开文件
```

```
FILE *in;
```

```
in = fopen(&filename1[0], "a+");
```

```
while(true){
```

```
//记录重传数目 若超过规定次数则通信失败
```

```
if(sendNum > MAX_SEND_TIME){
```

```
    printf("超过最大重传数\n");
```

```
    break;
```

```
}
```

```
ZeroMemory(buf, sizeof(buf));
```

```
bytesnum = recvfrom(connectSocket, buf, FILE_BYTES, 0, (struct  
sockaddr*)&serverAddress, &size);
```

```
//判断接受的是否为 error 包
```

```
if(isError(buf)){
```



```
printf("Error at : %d\n", WSAGetLastError());

closesocket(connectSocket);

WSACleanup();

exit(1);

}

//错误处理

//第一个包未接收成功的错误处理

if(bytesnum == SOCKET_ERROR && num == 1){

    printf("Error at : %d\n", WSAGetLastError());

    sendto(connectSocket, buf1, 2 + filename.size() + type.size() + 2, 0,

(struct sockaddr*)&serverAddress, size);

    sendNum++;

    continue;

}else if(bytesnum == SOCKET_ERROR && num != 1){//未接收到指定包的错

误处理

    printf("Error at : %d\n", WSAGetLastError());

    buf[1] = ACK; buf[3] = num;

    sendto(connectSocket, buf, ACK_BYTES, 0, (struct

sockaddr*)&serverAddress, size);

    sendNum++;

    continue;

}else {//正确情况
```

```
fprintf(in, "%s", buf + 4);

sendNum = 1;

//发送 ACK

if(buf[3] == num){

    buf[1] = ACK; buf[3] = num;

}else{

    num++;

    buf[1] = ACK; buf[3] = num;

}

do{

    byteRead = sendto(connectSocket, buf, ACK_BYTES, 0, (struct
sockaddr*)&serverAddress, size);

}while(byteRead == SOCKET_ERROR);

}

if(bytesnum < FILE_BYTES){

    //关闭文件

    fclose(in);

    //退出循环

    break;

}

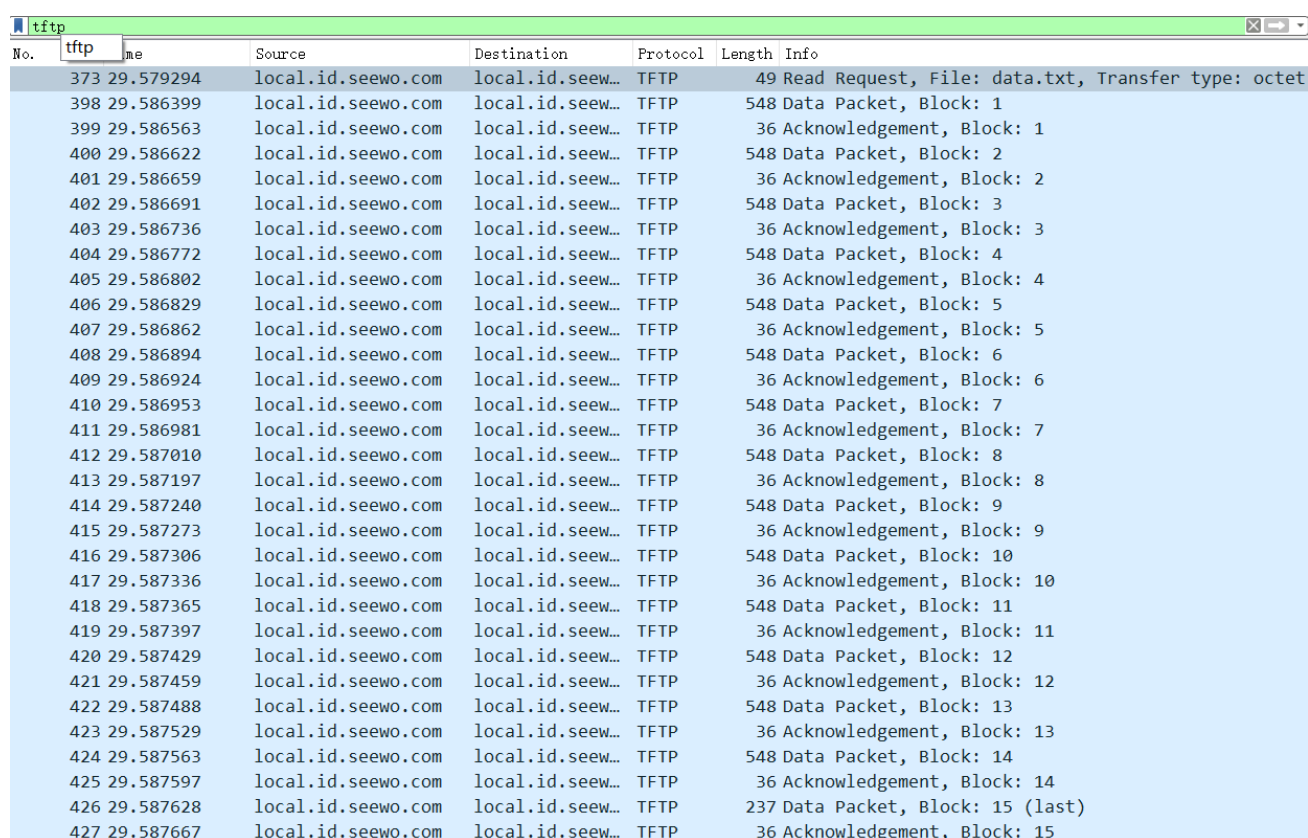
}
```

```
printf("receive succeed!\n");  
}
```

三、实验测试与分析

3.1 系统测试及结果说明

代码完成后在个人 PC 上进行测试，使用的服务端为 tftpd64。首先进行接受数据的测试案例，图 3-1 为 wireshark 抓包情况。



No.	Time	Source	Destination	Protocol	Length	Info
373	29.579294	local.id.seewo.com	local.id.seewo.com	TFTP	49	Read Request, File: data.txt, Transfer type: octet
398	29.586399	local.id.seewo.com	local.id.seewo.com	TFTP	548	Data Packet, Block: 1
399	29.586563	local.id.seewo.com	local.id.seewo.com	TFTP	36	Acknowledgement, Block: 1
400	29.586622	local.id.seewo.com	local.id.seewo.com	TFTP	548	Data Packet, Block: 2
401	29.586659	local.id.seewo.com	local.id.seewo.com	TFTP	36	Acknowledgement, Block: 2
402	29.586691	local.id.seewo.com	local.id.seewo.com	TFTP	548	Data Packet, Block: 3
403	29.586736	local.id.seewo.com	local.id.seewo.com	TFTP	36	Acknowledgement, Block: 3
404	29.586772	local.id.seewo.com	local.id.seewo.com	TFTP	548	Data Packet, Block: 4
405	29.586802	local.id.seewo.com	local.id.seewo.com	TFTP	36	Acknowledgement, Block: 4
406	29.586829	local.id.seewo.com	local.id.seewo.com	TFTP	548	Data Packet, Block: 5
407	29.586862	local.id.seewo.com	local.id.seewo.com	TFTP	36	Acknowledgement, Block: 5
408	29.586894	local.id.seewo.com	local.id.seewo.com	TFTP	548	Data Packet, Block: 6
409	29.586924	local.id.seewo.com	local.id.seewo.com	TFTP	36	Acknowledgement, Block: 6
410	29.586953	local.id.seewo.com	local.id.seewo.com	TFTP	548	Data Packet, Block: 7
411	29.586981	local.id.seewo.com	local.id.seewo.com	TFTP	36	Acknowledgement, Block: 7
412	29.587010	local.id.seewo.com	local.id.seewo.com	TFTP	548	Data Packet, Block: 8
413	29.587197	local.id.seewo.com	local.id.seewo.com	TFTP	36	Acknowledgement, Block: 8
414	29.587240	local.id.seewo.com	local.id.seewo.com	TFTP	548	Data Packet, Block: 9
415	29.587273	local.id.seewo.com	local.id.seewo.com	TFTP	36	Acknowledgement, Block: 9
416	29.587306	local.id.seewo.com	local.id.seewo.com	TFTP	548	Data Packet, Block: 10
417	29.587336	local.id.seewo.com	local.id.seewo.com	TFTP	36	Acknowledgement, Block: 10
418	29.587365	local.id.seewo.com	local.id.seewo.com	TFTP	548	Data Packet, Block: 11
419	29.587397	local.id.seewo.com	local.id.seewo.com	TFTP	36	Acknowledgement, Block: 11
420	29.587429	local.id.seewo.com	local.id.seewo.com	TFTP	548	Data Packet, Block: 12
421	29.587459	local.id.seewo.com	local.id.seewo.com	TFTP	36	Acknowledgement, Block: 12
422	29.587488	local.id.seewo.com	local.id.seewo.com	TFTP	548	Data Packet, Block: 13
423	29.587529	local.id.seewo.com	local.id.seewo.com	TFTP	36	Acknowledgement, Block: 13
424	29.587563	local.id.seewo.com	local.id.seewo.com	TFTP	548	Data Packet, Block: 14
425	29.587597	local.id.seewo.com	local.id.seewo.com	TFTP	36	Acknowledgement, Block: 14
426	29.587628	local.id.seewo.com	local.id.seewo.com	TFTP	237	Data Packet, Block: 15 (last)
427	29.587667	local.id.seewo.com	local.id.seewo.com	TFTP	36	Acknowledgement, Block: 15

3-1 RRQ 测试截图

图 3-2WRQ 测试截图：

83	12.173705	local.id.seewo.com	local.id.seew...	TFTP	46 Write Request, File: a.txt, Transfer 1
108	12.174810	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 0
109	12.175089	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 1
110	12.175257	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 1
111	12.175289	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 2
112	12.175341	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 2
113	12.175369	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 3
114	12.175398	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 3
115	12.175420	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 4
116	12.175451	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 4
117	12.175472	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 5
118	12.175503	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 5
119	12.175524	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 6
120	12.175552	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 6
121	12.175573	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 7
122	12.175602	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 7
123	12.175636	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 8
124	12.175669	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 8
125	12.175691	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 9
126	12.175753	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 9
127	12.175775	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 10
128	12.175806	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 10
129	12.175828	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 11
130	12.175857	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 11
131	12.175879	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 12
132	12.175907	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 12
133	12.175928	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 13
134	12.175956	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 13
135	12.175980	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 14
136	12.176014	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 14
137	12.176040	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 15
138	12.176071	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 15
139	12.176105	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 16
140	12.176136	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 16
141	12.176160	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 17
142	12.176203	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 17
143	12.176231	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 18
144	12.176269	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 18
145	12.176297	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 19
146	12.176338	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 19
147	12.176370	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 20
148	12.176417	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 20
149	12.176451	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 21
150	12.176498	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 21
151	12.176531	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 22
152	12.176573	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 22
153	12.176635	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 23
154	12.176677	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 23
155	12.176721	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 24
156	12.176800	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 24
157	12.176832	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 25
158	12.176868	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 25
159	12.176890	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 26
160	12.176919	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 26
161	12.176941	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 27
162	12.176970	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 27
163	12.177009	local.id.seewo.com	local.id.seew...	TFTP	548 Data Packet, Block: 28
164	12.177053	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 28
165	12.177095	local.id.seewo.com	local.id.seew...	TFTP	466 Data Packet, Block: 29 (last)
166	12.177130	local.id.seewo.com	local.id.seew...	TFTP	36 Acknowledgement, Block: 29

图 3-2 WRQ 测试截图

3.2 遇到的问题及解决方法

问题一：开始时构造请求包错误，通过 wireshark 发现发送的包为错误请求包，于是查找资料发现在数据传输模式后面需要以 0 结尾。

问题二：超时重传问题不能解决，一开始想使用多个线程启用计时器来完成超时重传，后来发现 winsock2 库中有自带的库可以设置超时时间。

问题三：无限重传，设置最大重传数后解决。

3.3 设计方案存在的不足

不足一：没有设计日志功能。

不足二：没有设计吞吐量功能。

不足三：没有图形界面。

不足四：超时时间没有经过实验确定，可能不是最佳超时时间。

四、 实验总结

4.1 实验感想

TFTP (Trivial File Transfer Protocol) 是一种简单的文件传输协议，通常用于在计算机网络上进行文件传输。在进行了 TFTP 客户端实验后，我获得了一些有关该协议和网络通信的深刻认识，下面是我的实验感想。

通过这个实验，我更深入地理解了 TFTP 的工作原理。TFTP 是一个面向 UDP 的协议，它使用简单的请求响应模式来传输文件。TFTP 客户端向 TFTP 服务器发送读取或写入请求，然后服务器响应。这个过程的简单性使得 TFTP 成为一种适用于特定用途的轻量级文件传输协议。

我学到了 TFTP 客户端如何与服务器进行通信。实验中，我们需要编写一个 TFTP 客户端程序，它能够构建 TFTP 请求报文并将其发送到服务器，然后接收并处理服务器的响应。这个过程让我更加了解了网络编程和工具的基本概念。

TFTP 实验还让我认识到了网络中可能出现的错误和异常情况。在实验中，我们必须考虑到文件不存在、权限问题以及网络连接等情况，并编写相应的错误处理代码。这个过程教会了我如何编写健壮的网络应用程序，以应对各种不可预测的情况。

通过这个实验，我更加深刻地理解了协议的重要性。网络通信必须遵循一定的协议，以保证数据的可靠传输。TFTP 虽然简单，但它在特定的场景中仍然有其用途，例如位于轮胎内快速传输硬件或配置文件。

总的来说，TFTP 客户端实验为我提供了宝贵的网络编程经验，并让我更好地理解协议和网络通信的重要性。我相信这些知识和经验将在未来的学习和职业生涯中对我产生积极的影响。

4.2 意见和建议

在进行 TFTP 客户端实验时，我有一些关于实验设计的建议，希望能够提高实验的效果和学习体验。

在建议实验前提供更详细的背景知识和学习材料。在深入实验之前，提供有关 TFTP 协议的更多信息，包括其工作原理、用途和常见应用场景，可以帮助学生更好地理解实验的目标和重要性。

增加实验的互动性和挑战性。通过引入一些较复杂的网络情感或附加任务，可以激发学生的兴趣，并促使他们深入思考和解决问题。这有助于提高学生的问题解决能力和实践应用技能。

考虑提供更多的实验支持和资源。这包括示例代码、调试技巧和常见错误的解决方法。这些资源可以帮助学生更轻松地完成实验，并在遇到困难时提供支持。

鼓励学生进行实验后的反馈。收集他们的意见和建议，促进不断改进实验设计和教学方法。这种反馈循环有助于提高实验的质量，并确保其与学生的需求和期望保持一致。

总之，通过更深入的背景知识、更严格的任务、更多的支持资源和反馈机制，可以改进 TFTP 客户端实验的设计，从而获得更多的教育性和吸引力。

原创性声明

本人郑重声明本报告内容，是由作者本人独立完成的。有关观点、方法、数据和文献等的引用已在文中指出。除文中已注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品成果，不存在剽窃、抄袭行为。

已阅读并同意以下内容。

判定为不合格的一些情形：

- （1） 请人代做或冒名顶替者；
- （2） 替人做且不听劝告者；
- （3） 实验报告内容抄袭或雷同者；
- （4） 实验报告内容与实际实验内容不一致者；
- （5） 实验代码抄袭者。

作者签名： 钱俊卓