计算机网络实验报告3-1

姓名: 高祎珂 学号: 2011743

实验要求

利用**数据报套接字**在用户空间实现面向连接的可靠数据传输,功能包括:**建立连接、差错检测、确认重传**。流量控制采用**停等机制,完成给定测试文件的传输**。

功能设计

1、数据报套接字UDP:

UDP是User Datagram Protocol的简称,中文名是用户数据报协议,是OSI参考模型中的传输层协议,它是一种无连接的传输层协议,提供面向事务的简单不可靠信息传送服务.

2、建立连接:

根据TCP的三次握手和四次挥手原则,进行发送端与接收端的连接和断连。

3、差错检测

计算校验和,实现可靠数据传输。本次实验设计数据的UCP报文头是分开传输的,在握手,挥手期间,只传UDP报文头,在数据传输过程中,要加入数据段计算校验和。

4、确认重传

使用rdt3.0实现,设计一个定时器,根据序列号确认,以及超时重传。超时重传指的是在发送数据报文段后开始计时,到等待确认应答到来的那个时间间隔。如果超过这个时间间隔,仍未收到确认应答,发送端将进行数据重传。

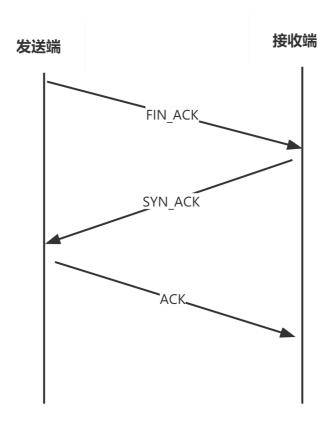
协议设计

• 报文格式

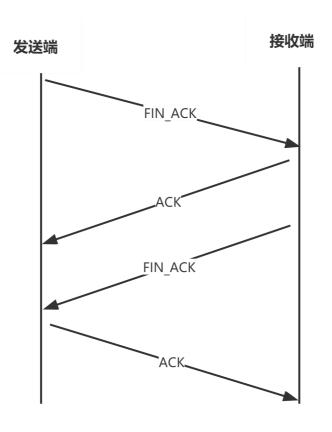


前16位为数据长度,用于记录数据区大小,17-32位位校验和,用于检验传输的正确性,33-40为标志位,40-48位为传输的数据包的序列号。

• 连接与断开



。 四次挥手



• 数据传输

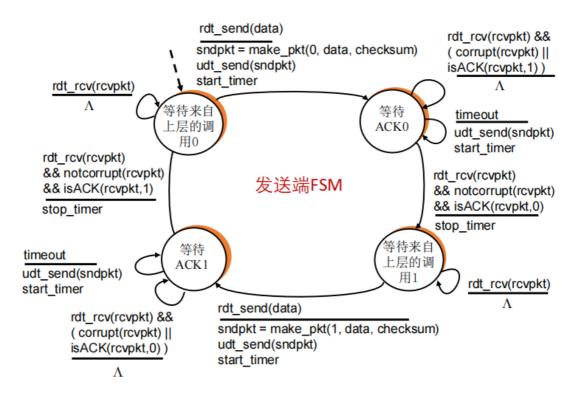
数据传输发怂端和接收端军参照rdt3.0进行实现。数据在传输时,将一个文件分为数个包进行分段传输,每个包内容为数据头+数据,且考虑到文件内容较大,每次只读取文件的一部分到缓存进行传输。

发送端在传输时,需要接受到上一个发送包序号的ACK=1旦确认无误后才能发送下一个数据包,若超时或者校验和不对则进行重传,如果未收到想要的数据包则等待;接收端接收到了一个数据包,先要进行校验,如果检查无误,则向发送放返回该序列号的ACK=1。

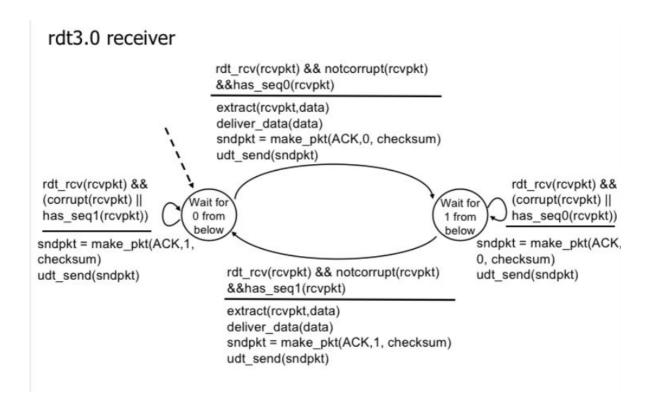
发送端状态机

使用rdt3.0实现可靠数据传输

■ rdt3.0: 发送端状态机



接收端状态机



代码实现

计算校验和

```
u_short cksum(u_short* UDP_header, int size)
{
   // size以字节计数,但是这里要用16byte
                                       u_short
   // 这里使用的函数都是以字节计数的
   unsigned short* a = (unsigned short*)UDP_header;
   unsigned int sum = 0;
   for (int i = 0; i < size / 2; i++) {
       sum += (unsigned int)a[i];
       if (sum & (0xFFFF0000)) {
           sum &= 0xFFFF;
           sum++;
       }
   }
    return ~(((unsigned short)sum) & OXFFFF);
}
```

数据头

```
struct HEADER
{
    u_short sum = 0;//校验和 16位
    u_short datasize = 0;//所包含数据长度 16位
    unsigned char flag = 0;
    //八位,使用后三位,排列是FIN ACK SYN
    unsigned char SEQ = 0;
    //八位,传输的序列号,0~255,超过后mod
    HEADER() {
        sum = 0;//校验和 16位
        datasize = 0;//所包含数据长度 16位
```

```
flag = 0;

//八位, 使用后三位, 排列是FIN ACK SYN

SEQ = 0;

}

};
```

发送端发包

```
void send_package(SOCKET& socketClient, SOCKADDR_IN& servAddr, int& servAddrlen,
char* message, int len, int& order) {
   HEADER header;
    char* buffer = new char[MAXSIZE + sizeof(header)];
   for (int i = 0; i < MAXSIZE + sizeof(header); i++) {</pre>
        buffer[i] = 0;
    }
   header.datasize = len;
   header.SEQ = unsigned char(order);//序列号
   memcpy(buffer, &header, sizeof(header));
   memcpy(buffer + sizeof(header), message, sizeof(header) + len);
    u_short check = cksum((u_short*)buffer, sizeof(header) + len);//计算校验和
   header.sum = check;
    memcpy(buffer, &header, sizeof(header));
    sendto(socketClient, buffer, len + sizeof(header), 0, (sockaddr*)&servAddr,
servAddrlen);//发送
    cout << "Send message " << len << " bytes!" << " FLAG:" <</pre>
state[int(header.flag)] << " SEQ:" << int(header.SEQ) << " CHECKSUM:" <<</pre>
int(header.sum) << endl;</pre>
    clock_t start = clock();//记录发送时间
   //接收ack等信息
   while (true) {
        u_long mode = 1;
        ioctlsocket(socketClient, FIONBIO, &mode);
        // 超时机制
        int times = 0;
        while (times < 5 && recvfrom(socketClient, buffer, MAXSIZE +
sizeof(header), 0, (sockaddr*)&servAddr, &servAddrlen) <= 0)</pre>
        {
            // 未收到消息,超时重传
           if (clock() - start > MAX_TIME)
                sendto(socketClient, buffer, len + sizeof(header), 0,
(sockaddr*)&servAddr, servAddrlen);//发送
                cout << "TIME OUT! ReSend message " << len << " bytes! FLAG:" <<</pre>
state[int(header.flag)] << " SEQ:" << int(header.SEQ) << endl;</pre>
               times++;
                start = clock();//记录发送时间
           }
        }
        //cout << "收到了消息" << endl;
        if (times == 5) {
            cout << "发送出错" << endl;
            return;
        }
        // 得到了数据包
        memcpy(&header, buffer, sizeof(header));//缓冲区接收到信息,读取
```

```
//检查是否是想要的ACK以及差错检验
       //cout << header.SEQ <<" "<< header.flag << " " <<
cksum((u_short*)buffer, sizeof(header) + len) << endl;</pre>
       if (header.SEQ == u_short(order) && header.flag == ACK)
       {
           //修改flag,检查校验和
           header.flag = INIT;
           memcpy(buffer, &header, sizeof(header));
           if (cksum((u_short*)buffer, sizeof(header) + len) == check)
               header.flag = ACK;
               cout << "Send has been confirmed! FLAG:" <<</pre>
state[int(header.flag)] << " SEQ:" << int(header.SEQ) << endl;</pre>
               break;
           }
       }
       // 序列号不对,或者检验和不对,等待
       else {
           // 收到的消息不对,超时重传
           if (clock() - start > MAX_TIME)
               header.datasize = len;
               header.SEQ = u_char(order);//序列号
               header.flag = u_char(0x0);
               memcpy(buffer, &header, sizeof(header));
               memcpy(buffer + sizeof(header), message, sizeof(header) + len);
               u_short check = cksum((u_short*)buffer, sizeof(header) + len);//
计算校验和
               header.sum = check;
               memcpy(buffer, &header, sizeof(header));
               sendto(socketClient, buffer, len + sizeof(header), 0,
(sockaddr*)&servAddr, servAddrlen);//发送
               cout << "Message Error! ReSend message " << len << " bytes!</pre>
FLAG:" << state[int(header.flag)] << " SEQ:" << int(header.SEQ) << endl;</pre>
               start = clock();//记录发送时间
           }
           continue:
       }
   }
   u_long mode = 0;
   ioctlsocket(socketClient, FIONBIO, &mode);//改回阻塞模式
}
```

接收端收包

```
int RecvMessage(SOCKET& sockServ, SOCKADDR_IN& ClientAddr, int& ClientAddrLen,
    char* message) {
    long int all = 0;//文件长度
    HEADER header;
    char* buffer = new char[MAXSIZE + sizeof(header)];
    int seq = 0;
    int index = 0;
    while (true) {
```

```
for (int i = 0; i < MAXSIZE + sizeof(header); i++) {</pre>
            buffer[i] = 0;
        int length = recvfrom(sockServ, buffer, sizeof(header) + MAXSIZE, 0,
(sockaddr*)&ClientAddr, &ClientAddrLen);//接收报文长度
        //cout << "收到了文件" << length;
        memcpy(&header, buffer, sizeof(header));
        //判断是否是结束
        if (header.flag == OVER && cksum((u_short*)&header, sizeof(header)) ==
0)
        {
            cout << "文件接收完毕" << endl;
            break;
        }
        // 数据包flag是初始化状态,为0 确认校验和
        if (header.flag == unsigned char(0) && cksum((u_short*)buffer,
length)==0&& seq == int(header.SEQ)) {
            cout << "Rece message " << length - sizeof(header) << " bytes!Flag:"</pre>
<< flagmap[int(header.flag)] << " SEQ : " << int(header.SEQ) << " CHECKSUM:" <<</pre>
int(header.sum) << endl;</pre>
            // 把数据写入message
            char* temp = new char[length - sizeof(header)];
            memcpy(temp, buffer + sizeof(header), length - sizeof(header));
            memcpy(message + all, temp, length - sizeof(header));
            all = all + int(header.datasize);
            // 校验和重新置0,返回ACK
            header.flag = ACK;
            header.sum = 0;
            memcpy(buffer, &header, sizeof(header));
            // 发送ACK
            sendto(sockServ, buffer, sizeof(header)+ MAXSIZE, 0,
(sockaddr*)&ClientAddr, ClientAddrLen);
            cout << "Send to Clinet ACK:" << (int)header.SEQ << " SEQ:" <<</pre>
(int)header.SEQ << " CHECKSUM:" << int(header.sum)<< endl;</pre>
            seq++;
           if (seq > 255)
               seq = seq - 256;
            }
        //校验和不对,或者SEQ不对
        else {
            //说明出了问题,返回ACK
            header.flag = ACK;
           if (cksum((u_short*)buffer, length) != 0)
                header.flag = INIT;
            memcpy(buffer, &header, sizeof(header));
            //重发该包的ACK
            sendto(sockServ, buffer, sizeof(header)+MAXSIZE, 0,
(sockaddr*)&ClientAddr, ClientAddrLen);
            cout << "Some ERROR:" << (int)header.SEQ << " SEQ:" <<</pre>
(int)header.SEQ << endl;</pre>
```

```
continue;//丢弃该数据包

}

//发送OVER信息
header.flag = OVER;
header.sum = 0;
u_short temp = cksum((u_short*)&header, sizeof(header));
header.sum = temp;
memcpy(buffer, &header, sizeof(header));
if (sendto(sockserv, buffer, sizeof(header), 0, (sockaddr*)&ClientAddr,
ClientAddrLen) == -1)
{
    return -1;
}
return all;
}
```

结果展示

握手展示

```
7 D:\VS2022\Project\Server\x64\Debug\Server.exe
初始化成功
建立监听socket成功
绑定成功
排入监听状态。等待客户端上线
成功接收第一次握手信息,FLAG:SYN
成功建立通信!可以接收数据,FLAG:ACK
```

M D:\VS2022\Project\Server\x64\Debug\Client.exe

```
份始化成功
收到第二次握手信息, FLAG=ACK_SYN
服务器成功连接:可以发送数据
请输入文件名称
```

客户端

数据传输展示

```
D:\VS2022\Project\Server\x64\Debug\Client.exe
                                                                                                                                                                                                                                                                                                                                  D:\VS2022\Project\Server\x64\Debug\Server.exe
       end message 7 bytes! FLAG;INIT SEQ:0 CHECKSUM:433
end has been confirmed! FLAG;ACKSEQ:94
end End!
計方已成功統敗文件!
                                                                                                                                                                                                                                                                                                                                              nd to Clinet ACK:9 SEQ:9 CHECKSUM:0
                                                                                                                                                                                                                                                                                                                                 文件接收完毕
                                                                                                                                                                                                                                                                                                                             Rece message 7 bytes!Flag:INIT SEQ: 0 CHECKSUM
Send to Clinet ACK:0 SEQ:0 CHECKSUM:0
文件接收完毕
                                                                                                                                                                                                                                                                                                                         Send to Clinet ACK:0 SEQ:0 CHECKSUM:0
文件接收完毕
Rece message 1024 bytes!Flag:INIT SEQ: 0 CHECKSUM:29986
Send to Clinet ACK:0 SEQ:0 CHECKSUM:0
Rece message 1024 bytes!Flag:INIT SEQ: 1 CHECKSUM:62646
Send to Clinet ACK:1 SEQ:1 CHECKSUM:0
Rece message 1024 bytes!Flag:INIT SEQ: 2 CHECKSUM:58318
Send to Clinet ACK:2 SEQ:2 CHECKSUM:0
Rece message 1024 bytes!Flag:INIT SEQ: 3 CHECKSUM:41596
Send to Clinet ACK:3 SEQ:3 CHECKSUM:0
Rece message 1024 bytes!Flag:INIT SEQ: 4 CHECKSUM:5272
Send to Clinet ACK:4 SEQ:4 CHECKSUM:0
Rece message 1024 bytes!Flag:INIT SEQ: 5 CHECKSUM:16760
Send to Clinet ACK:5 SEQ:5 CHECKSUM:0
Rece message 1024 bytes!Flag:INIT SEQ: 6 CHECKSUM:51790
Send to Clinet ACK:6 SEQ:6 CHECKSUM:0
Rece message 1024 bytes!Flag:INIT SEQ: 7 CHECKSUM:31670
Send to Clinet ACK:7 SEQ:7 CHECKSUM:0
Rece message 1024 bytes!Flag:INIT SEQ: 8 CHECKSUM:19841
Send to Clinet ACK:8 SEQ:8 CHECKSUM:0
Rece message 784 bytes!Flag:INIT SEQ: 9 CHECKSUM:18209
Send to Clinet ACK:8 SEQ:9 CHECKSUM:0
Rece message 784 bytes!Flag:INIT SEQ: 9 CHECKSUM:18209
Send to Clinet ACK:8 SEQ:9 CHECKSUM:0
Rece message 784 bytes!Flag:INIT SEQ: 9 CHECKSUM:433
Send to Clinet ACK:0 SEQ:0 CHECKSUM:0

$\frac{\partial Check Che
   Send message 1024 bytes! FLAG:INIT SEQ:0 CHECKSUM:29986
Send has been confirmed! FLAG:ACK SEQ:0
     Send message 1024 bytes! FLAG:INIT SEQ:1 CHECKSUM:62646
Send has been confirmed! FLAG:ACK SEQ:1
Send message 1024 bytes! FLAG:INIT SEQ:2 CHECKSUM:58318
     Send message 1024 bytes! FLAG:ANT SEQ:2 CHECKSUM:58318
Send has been confirmed! FLAG:ACK SEQ:2
Send message 1024 bytes! FLAG:INIT SEQ:3 CHECKSUM:41596
     Send has been confirmed! FLAG:ACK SEQ:3
Send has been confirmed! FLAG:ACK SEQ:4
Send has been confirmed! FLAG:ACK SEQ:4
Send has been confirmed! FLAG:ACK SEQ:4
Send message 1024 bytes! FLAG:INIT SEQ:5 CHECKSUM:16760
Send message 1024 bytes! FLAG:ACK SEQ:5
Send message 1024 bytes! FLAG:ACK SEQ:5
Send message 1024 bytes! FLAG:ACK SEQ:5
    Send has been confirmed! FLAG:ACK SEQ:5
Send message 1024 bytes! FLAG:INIT SEQ:6 CHECKSUM:51790
Send has been confirmed! FLAG:ACK SEQ:6
Send message 1024 bytes! FLAG:INIT SEQ:7 CHECKSUM:31670
Send has been confirmed! FLAG:ACK SEQ:7
Send message 1024 bytes! FLAG:INIT SEQ:8 CHECKSUM:19841
Send has been confirmed! FLAG:ACK SEQ:8
Send message 784 bytes! FLAG:INIT SEQ:9 CHECKSUM:18209
Send bas been confirmed! FLAG:ACK SEQ:9
Send End!
 Send message / bytes! FLAG:INIT SEQ:0 CHECKSUM:433
Send has been confirmed! FLAG:ACK SEQ:0
Send End!
对方己成功接收文件!
     对方已成功接收文件!
                                                                                                                                                                                                                                                                                                                                 文件接收完毕
                                                                                                                                                                                                                                                                                                                                Rece message 1024 bytes!Flag:INIT SEQ : 0 CHECKSUM:28496
     D:\VS2022\Project\Server\x64\Debug\Client.ex
                                                                                                                                                                                                                                                                                                                                  D:\VS2022\Project\Server\x64\Debug\Server.exe
                                                                                                                                                                                                                                                                                                                               Send to Clinet ACK:9 SEQ:9 CHECKSUM:0
文件接收完毕
      Send has been confirmed! FLAG:ACK SEQ:0
                         End!
对方已成功接收文件!
对方已成功接收文件!
Send message 1024 bytes! FLAG:INIT SEQ:1 CHECKSUM:28496
Send message 1024 bytes! FLAG:INIT SEQ:1 CHECKSUM:44651
Send message 1024 bytes! FLAG:INIT SEQ:1 CHECKSUM:44651
Send message 1024 bytes! FLAG:INIT SEQ:2 CHECKSUM:10990
Send has been confirmed! FLAG:ACK SEQ:2
Send message 1024 bytes! FLAG:INIT SEQ:3 CHECKSUM:5720
Send message 1024 bytes! FLAG:INIT SEQ:3 CHECKSUM:5720
Send message 1024 bytes! FLAG:INIT SEQ:4 CHECKSUM:31685
Send message 1024 bytes! FLAG:INIT SEQ:4 CHECKSUM:26965
Send has been confirmed! FLAG:ACK SEQ:5
Send message 1024 bytes! FLAG:INIT SEQ:6 CHECKSUM:43889
Send has been confirmed! FLAG:ACK SEQ:6
Send message 185 bytes! FLAG:INIT SEQ:7 CHECKSUM:9498
Send has been confirmed! FLAG:ACK SEQ:6
Send message 185 bytes! FLAG:INIT SEQ:7 CHECKSUM:9498
Send has been confirmed! FLAG:ACK SEQ:7
Send End!
对方已成功接收文件!
                                                                                                                                                                                                                                                                                                                              Rece message 7 bytes!Flag:INIT SEQ : (
Send to Clinet ACK:0 SEQ:0 CHECKSUM:0
文件接收完毕
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                0 CHECKSUM: 433
                                                                                                                                                                                                                                                                                                                           X件接收完毕

Rece message 1024 bytes!Flag:INIT SEQ: 0 CHECKSUM:28496
Send to Clinet ACK:0 SEQ:0 CHECKSUM:0
Rece message 1024 bytes!Flag:INIT SEQ: 1 CHECKSUM:44651
Send to Clinet ACK:1 SEQ:1 CHECKSUM:0
Rece message 1024 bytes!Flag:INIT SEQ: 2 CHECKSUM:10990
Send to Clinet ACK:2 SEQ:2 CHECKSUM:0
Rece message 1024 bytes!Flag:INIT SEQ: 3 CHECKSUM:5720
Send to Clinet ACK:3 SEQ:3 CHECKSUM:0
Rece message 1024 bytes!Flag:INIT SEQ: 4 CHECKSUM:31685
Send to Clinet ACK:4 SEQ:4 CHECKSUM:0
Rece message 1024 bytes!Flag:INIT SEQ: 5 CHECKSUM:26965
Send to Clinet ACK:5 SEQ:5 CHECKSUM:0
Rece message 1024 bytes!Flag:INIT SEQ: 6 CHECKSUM:43889
Send to Clinet ACK:6 SEQ:6 CHECKSUM:0
Rece message 185 bytes!Flag:INIT SEQ: 7 CHECKSUM:9498
Send to Clinet ACK:7 SEQ:7 CHECKSUM:0
文件接收完毕
    Send End!
对方已成功接收文件!
Send message 3 bytes! FLAG:INIT SEQ:0 CHECKSUM:37271
Send has been confirmed! FLAG:ACK SEQ:0
Send End!
对方已成功接收文件!
```

文件接收完毕

文件传输完成

名称	修改日期	类型	大小
.vs	2022/11/16 21:44	文件夹	
<mark>⊩</mark> x64	2022/11/18 0:20	文件夹	
🔁 1.jpg	2022/11/19 18:32	JPG 图片文件	1,814 KB
🔁 2.jpg	2022/11/18 18:32	JPG 图片文件	5,761 KB
🖺 3.jpg	2022/11/18 18:32	JPG 图片文件	11,689 KB
helloworld.txt	2022/11/19 14:01	文本文档	1,617 KB
server.cpp	2022/11/19 13:45	C++ source file	11 KB
₽ Server.sIn	2022/11/17 0:10	Microsoft Visual	3 KB
Server.vcxproj	2022/11/17 0:10	VC++ Project	7 KB
Server.vcxproj.filters	2022/11/17 0:10	VC++ Project Fil	1 KB
Server.vcxproj.user	2022/11/16 21:44	USER 文件	1 KB

Rece message 3 bytes!Flag:INIT SEQ: 0 CHECKSUM:37271 Send to Clinet ACK:0 SEQ:0 CHECKSUM:0 文化抄址产生比

传输总时间为:13s 吞吐率为:142873byte/s

青输入文件名称

挥手展示

```
请输入文件名称

quit

Send End!

对方已成功接收文件!决定断开

收到第二次挥手信息,FLAG= ACK

四次挥手结束,连接断开! FLAG=FIN_ACK
```

可以看到基本功能进行了实现。