



南开大学
Nankai University

南 开 大 学

网 络 空 间 安 全 学 院

计算机网络实验报告

基于 UDP 服务设计可靠传输协议并编程实现

李潇逸 2111454

年级：2021 级

专业：信息安全、法学

指导教师：张建忠

2023 年 12 月 15 日

目录

一、 实验要求	1
二、 原理	1
(一) client 端	1
(二) server 端	1
三、 代码分析	1
四、 展示	3
五、 总结	3

一、 实验要求

在实验 3-1 的基础上，将停等机制改成基于滑动窗口的流量控制机制，发送窗口和接收窗口采用相同大小，支持选择确认，完成给定测试文件的传输。

二、 原理

(一) client 端

- 为了实现消息的同时收发，提高效率，使用多线程编程
- 每一个包都内置一个计时器，当超时后就进行重传

(二) server 端

- 按顺序接收对方发来的消息 msgseq (收到的消息序号) == recvnexseq, 且校验和正确, 返回对应的 ACK (ackseq=msgseq)
- 所有的都进行重传

三、 代码分析

发送

发送

```
1 int sendOneMsg(message msg, int num) {
2     if (num != messagenum - 1) {
3         memcpy(msg.data, buffer[num], 8192);
4         msg.len = 8192;
5     }
6     else {
7         memcpy(msg.data, buffer[num], lastlen);
8         msg.len = lastlen;
9     }
10    msg.seq = num;
11    sendmessage(msg);
12    cout << "len=" << msg.len << ",checksum=" << msg.checksum << ",flag="
        << msg.flag << ",seq=" << msg.seq << endl;
13    clock_t start = clock();
14    while (1) {
15        clock_t end = clock();
16        if (end - start > TIMEOUT) {
17            cout << "触发重传机制" << endl;
18            sendmessage(msg);
19            start = clock();
20        }
21        if (state[num] == 1) {
22            return 1;
```

```

23     }
24 }
25 }

```

接收线程

接收线程

```

1  int recvThread() {
2      message recv = recvmessage();
3      if (!recv.isEXT()) {
4          thread recvThr(recvThread);
5          if (recvThr.joinable()) {
6              recvThr.detach();
7          }
8          return 0;
9      }
10     //recv.output();
11     if (recv.seq == 65534) {
12         cout << "收到第二次挥手，文件传输结束！" << endl;
13         system("pause");
14         exit(0);
15         system("pause");
16     }
17     int ack = recv.seq;
18     if (status[ack] == 1 || ack < recvbase) {
19         cout << "收到无效包裹" << endl;
20         message send;
21         send.setACK();
22         send.ack = ack;
23         sendmessage(send);
24     }
25     if (status[ack] == 0) {
26         message send;
27         send.setACK();
28         send.ack = ack;
29         sendmessage(send);
30         memcpy(buffer[ack], recv.data, recv.len);
31         status[ack] = 1;
32         if (ack == messagenum - 1) {
33             lastlen = recv.len;
34             /*return 0;*/
35         }
36     }
37     if (isEnd) {
38         return 0;
39     }
40     thread recvThr(recvThread);
41     if (recvThr.joinable()) {
42         recvThr.detach();

```

```
43     }  
44     return 0;  
45 }
```

四、 展示

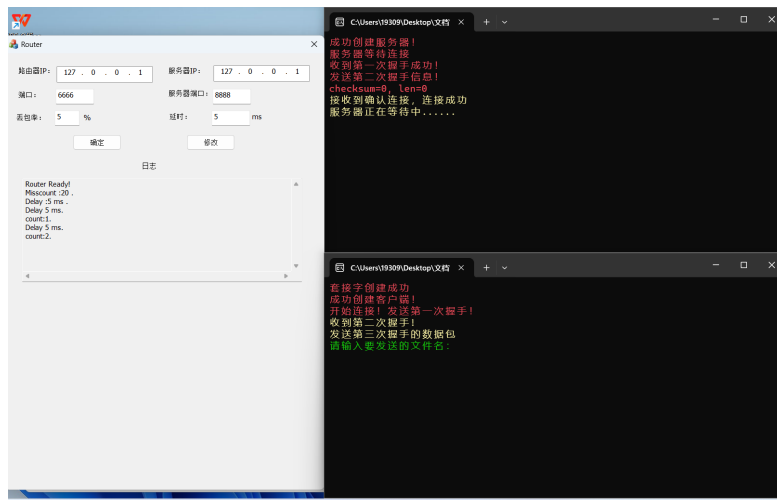


图 1: 初始设置

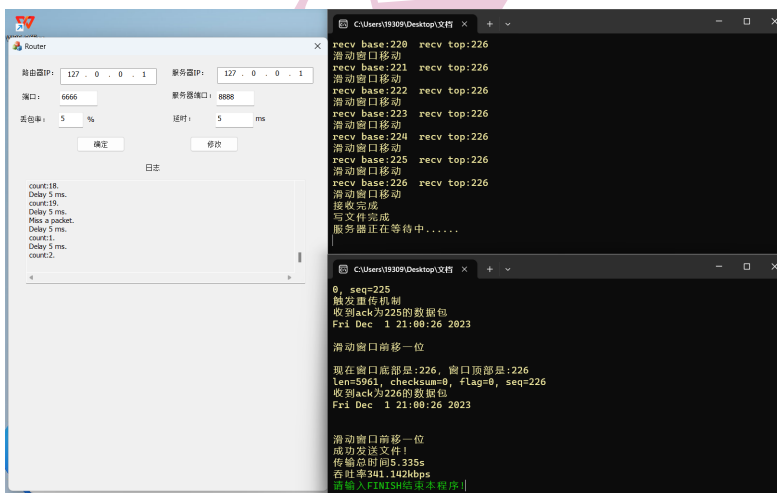


图 2: 传输过程

server.cpp	2023/12/1 17:43	C++ Source	11 KB
server.exe	2023/12/1 17:46	应用程序	35 KB
1.jpg	2023/12/1 21:00	JPG 图片文件	1,814 KB

图 3: 结果

五、 总结

实现了 UDP 传输滑动窗口，将研究如何加快传输速度