哈爾濱工業大學
Harbin Institute of Technology

计算机网络
课程实验报告

实验名称						
姓名			院系			
班级			学号			
任课教师			指导教师			
实验地点			实验时间			
实验课表现	出勤、表现得分(10)		实验报告得分(40)		实验总分	
	操作结果得分(50)					
教师评语						

哈尔滨工业大学计算学部
FACULTY OF COMPUTING, HIT

实验目的：

理解可靠数据传输的基本原理；掌握停等协议的工作原理；掌握基于 UDP 设计并实现一个停等协议的过程与技术。

理解滑动窗口协议的基本原理；掌握 GBN 的工作原理；掌握基于 UDP 设计并实现一个 GBN 协议的过程与技术。

实验内容：

1） 基于 UDP 设计一个简单的停等协议，实现单向可靠数据传输（服务器到客户的数据传输）；

2） 模拟引入数据包的丢失，验证所设计协议的有效性；

3） 改进所设计的停等协议，支持双向数据传输；

4）基于所设计的停等协议，实现一个 C/S 结构的文件传输；

5） 基于 UDP 设计一个简单的 GBN 协议，实现单向可靠数据传输（服务器到客户的数据传输）；

6） 模拟引入数据包的丢失，验证所设计协议的有效性；

7） 改进所设计的 GBN 协议，支持双向数据传输；

8）将所设计的 GBN 协议改进为 SR 协议。

实验过程：

1. 分组格式及各个域作用

a） 数据分组格式

采用指导书给出的数据格式

Seq	Data	0

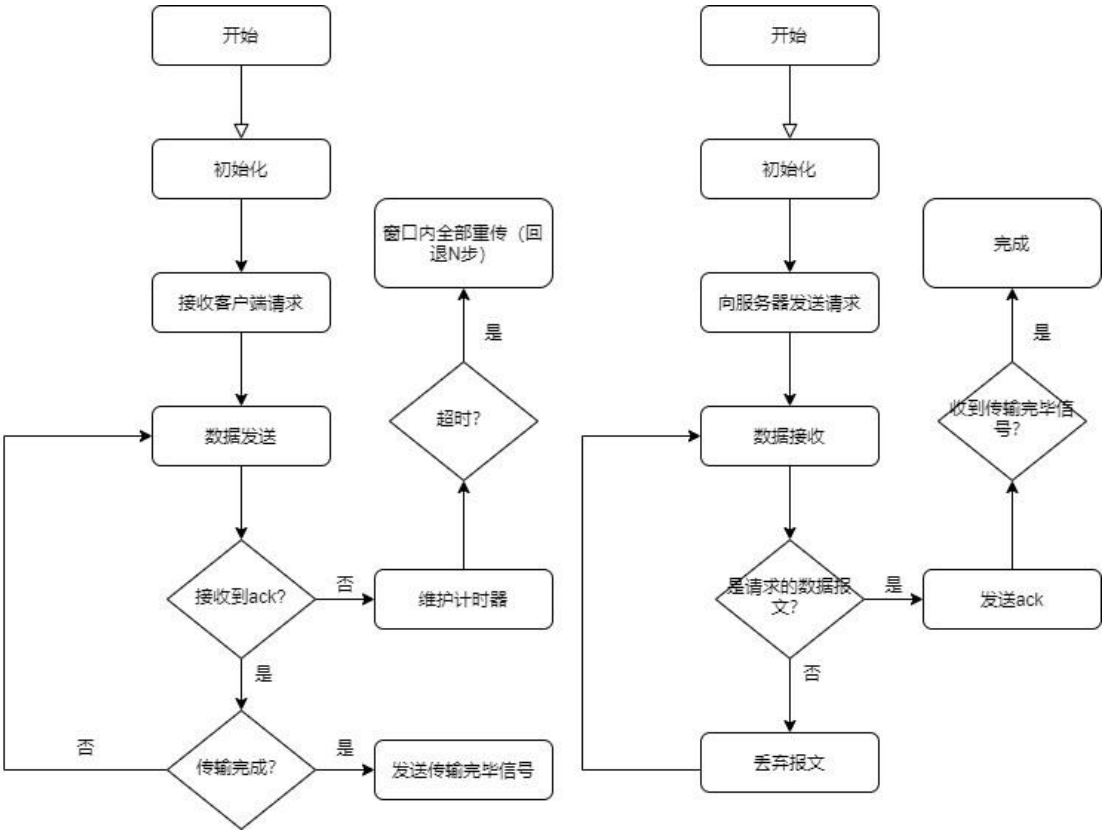
数据帧的 MTU 为 1500 字节，所以 UDP 数据报的数据部分应小于 1472 字节（除去 IP 头部 20 字节与 UDP 头的 8 字节）。Seq 为 1 个字节，取值为 0~255，（故序列号最多为 256 个）；Data≤1024 个字节，为传输的数据；最后一个字节放入 EOF0，表示结尾。

b) 确认分组格式

ACK	0

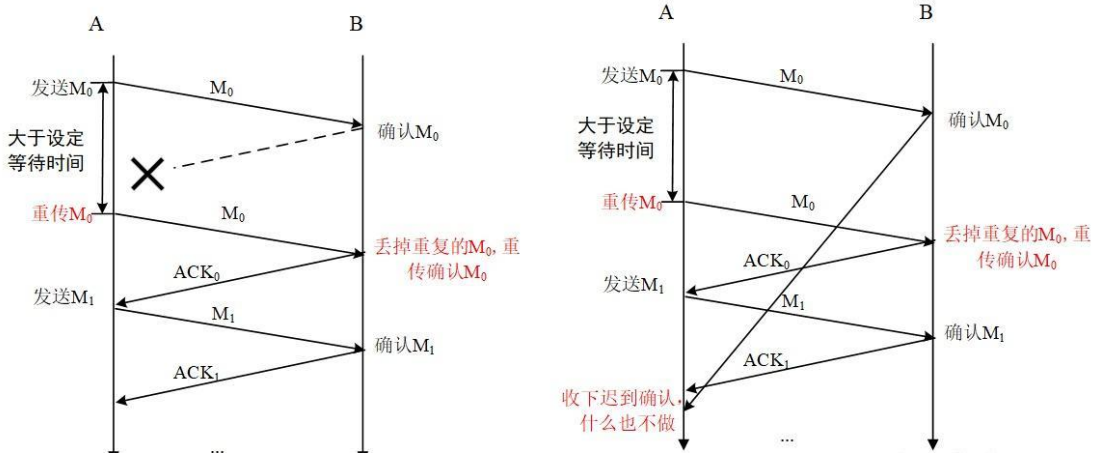
由于是从服务器端到客户端的单向数据传输，因此 ACK 数据帧不包含任何数据，只需要将 ACK 发送给服务器端即可。ACK 字段为一个字节，表示序列号数值；末尾放入 0，表示数据结束。

2. 协议两端程序流程图及协议典型交互过程

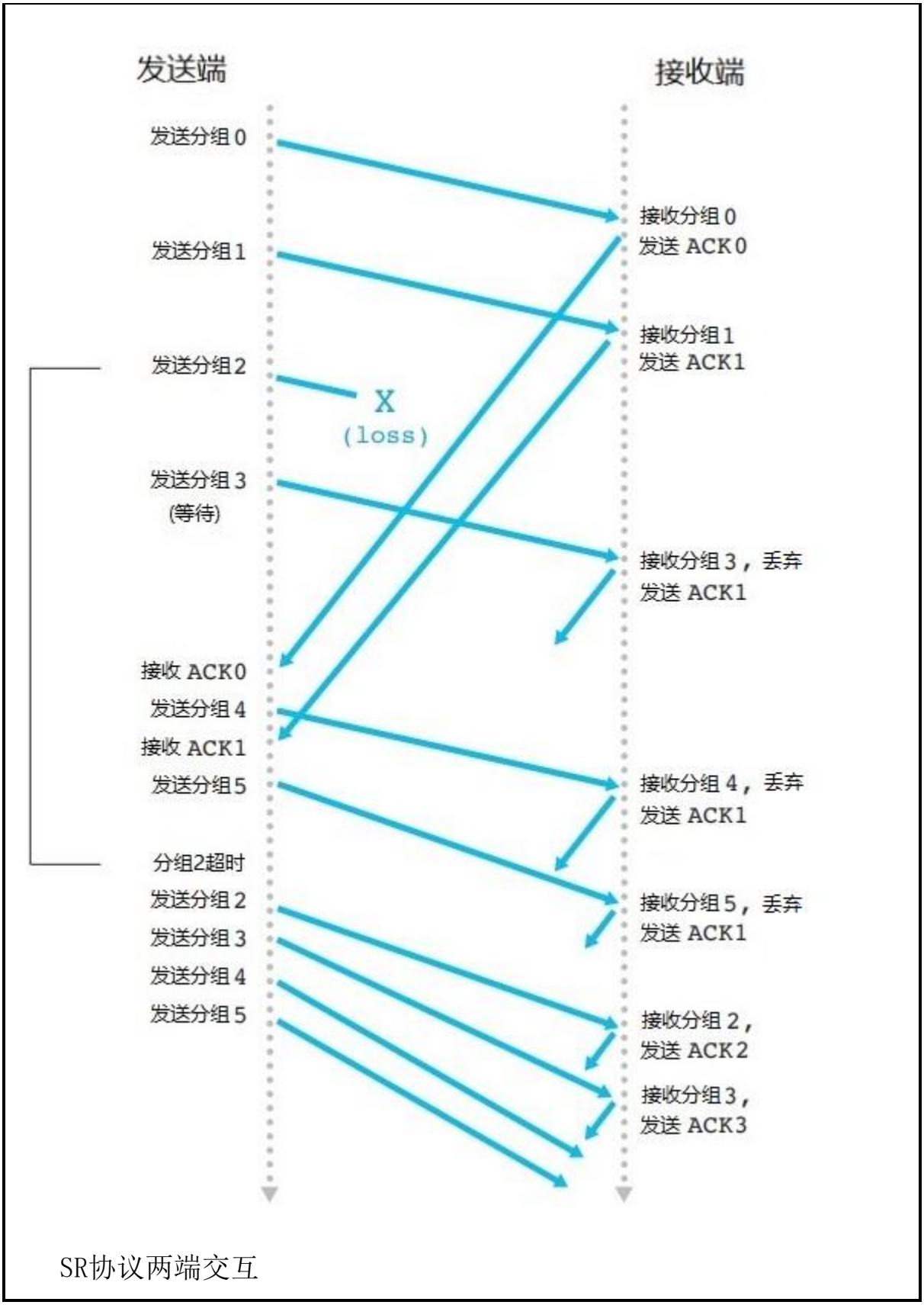


左图为服务器端流程，右图为客户端流程

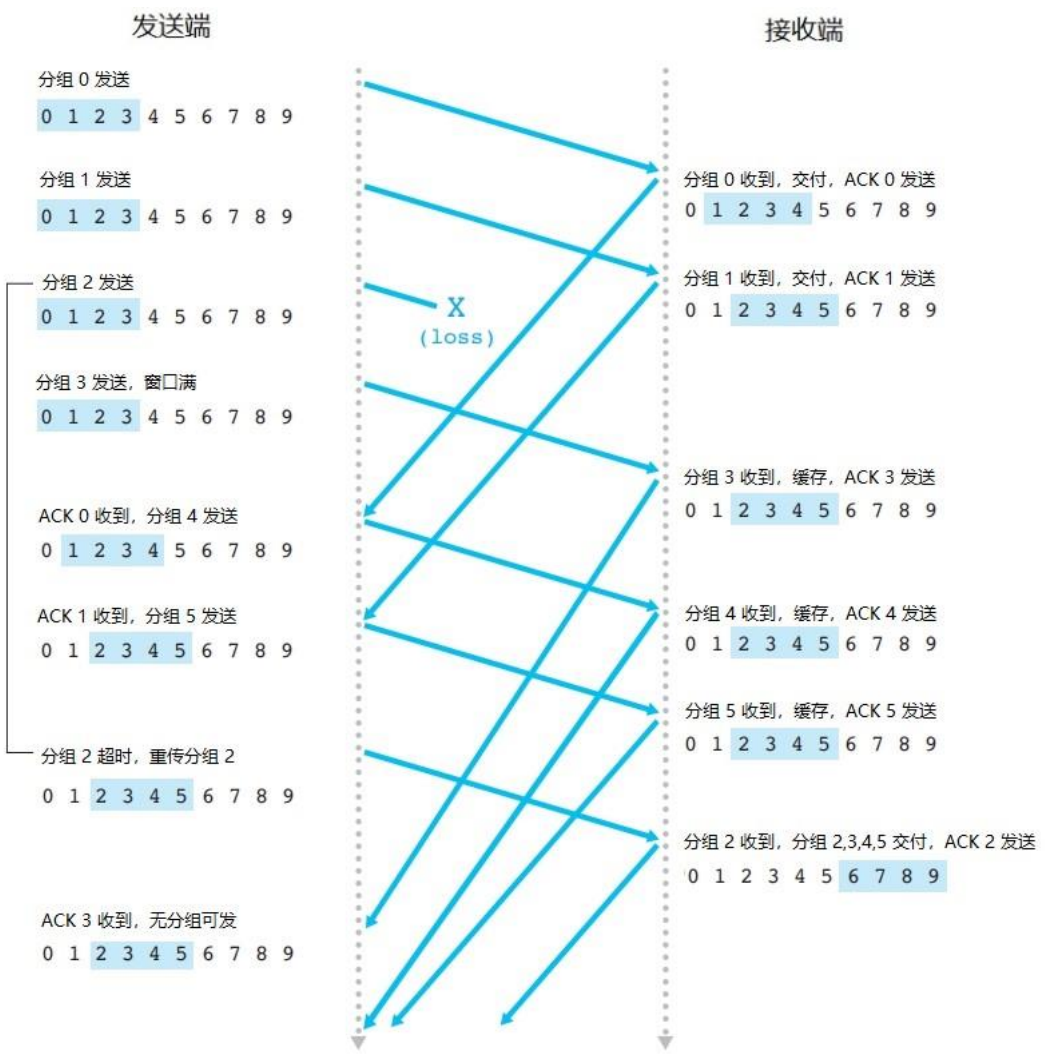
停/等协议两端交互



GBN协议两端交互


发送端
接收端
发送分组0
接收分组0
发送 ACK0
发送分组1
接收分组1
发送 ACK1
发送分组2
X
(loss)
发送分组3
(等待)
接收分组3，丢弃
发送 ACK1
接收 ACK0
发送分组4
接收 ACK1
发送分组5
接收分组4，丢弃
发送 ACK1
分组2超时
发送分组2
接收分组5，丢弃
发送 ACK1
发送分组3
发送分组4
发送分组5
接收分组2，
发送 ACK2
接收分组3，
发送 ACK3

SR协议两端交互



3. 数据分组丢失验证模拟方法

   采用指导书提供的lossInLossRatio函数，根据丢失率随机生成一个数字，判断是否丢失,丢失则返回 TRUE，否则返回 FALSE

4. 程序实现的主要类（或函数）及其主要作用

   void getCurTime(char *ptime)：获取当前系统时间,结果存入 ptime 中

   void printTips()：打印提示信息

BOOL lossInLossRatio(float lossRatio)：根据丢失率随机生成一个数字，判断是否丢失

unsigned int seqIsAvailable()：当前序列号 curSeq 是否可用

```
unsigned int seqIsAvailable()/*此时任意一个seq有三种状态：未发送-0；已发送未确认-1；已确认-2*/
{
    int step;
    step = curSeq - curAck;
    if(step < 0)
    {
        step += SEQ_SIZE;/*轮换*/
    }
    //序列号是否在当前发送窗口之内
    if (step >= SEND_WIND_SIZE)
    {
        return 0;
    }
    if (!ack[curSeq])
    {
        return 1;
    }
    return 2;
}
```

BOOL seqRecvAvailable(int recvSeq)：判断这个序列号是否是所需的

```
/*输入：接收到的序列号
输出：判断这个序列号是否是所需的，布尔变量
*/
BOOL seqRecvAvailable(int recvSeq)
{
    int step;
    int index;
    index = recvSeq - 1;
    step = index - curAck;
    step = step >= 0 ? step : step + SEQ_SIZE;
    //序列号是否在当前接收窗口之内
    if (step >= SEND_WIND_SIZE)
    {
        return FALSE;
    }
    return TRUE;
}
```

void timeoutHandler(SOCKET socketServer, SOCKADDR_IN

addrClient)：超时重传函数，在停等/GBN协议中直接重传窗口内所有报文，在SR协议中只重传对应未被确认超时的报文

```
void timeoutHandler(SOCKET socketServer, SOCKADDR_IN addrClient)
{
    printf("******Time out\n");

    // // 遍历发送窗口内的所有数据包
    // for (int i = 0; i < SEND_WIND_SIZE; ++i)
    // {
        int index = curAck  % SEQ_SIZE;

        // 如果数据包未被确认，则重发
        if (ack[index] == false)
        {
            printf("*****Resend Packet %d\n\n", index + 1);

            // 重发数据包
            char resendBuffer[BUFFER_LENGTH];
            resendBuffer[0] = index + 1; // 序列号从1开始
            memcpy(&resendBuffer[1], dataBuffer[index], BUFFER_LENGTH - 1); // 复制数据包内容

            sendto(socketServer, resendBuffer, BUFFER_LENGTH, 0, (SOCKADDR *)&addrClient, sizeof(SOCKADDR));
        }
    // }
    curSeq = curAck;
```

void ackHandler(char c)：在停等/GBN协议中是累计确认，在SR协议中是对应的报文的确认

```
void ackHandler(char c)
{
    unsigned char index = (unsigned char)c - 1;  //序列号减一
    printf("Recv a ack of seq %d\n", index + 1); //从接收方收到的确认收到的序列号
    int next;

    if (curAck == index)/*当前ack和收到的序列号一致时*/
    {
        totalAck += 1;/*确认的ack增加*/
        ack[index] = FALSE;/*对应的index置为false，表示不再需要ack*/
        curAck = (index + 1) % SEQ_SIZE;
        printf("\n############Windows move to %d###############\n", curAck + 1);/*窗口向前滑动*/
        for (int i = 1; i < SEQ_SIZE; i++)/*遍历窗口内的数据报*/
        {
            next = (i + index) % SEQ_SIZE;
            if (ack[next] == TRUE)/*检查大于当前序号但还未被确认的报文*/
            {
                ack[next] = FALSE;
                curAck = (next + 1) % SEQ_SIZE;
                totalSeq++;
                curSeq++;
                curSeq %= SEQ_SIZE;
            }
            else
            {
                break;
            }
        }
    }
    else if (curAck < index && index - curAck + 1 <= SEND_WIND_SIZE)//要保证是要接受的消息（在滑动窗口内）
```

实验结果：

1. -time、-quit命令的验证

客户端

```
PS E:\计算机网络\lab2\SR> ./client
The Winsock 2.2 dll was found okay
**********************************************
|     -time to get current time                |
|     -quit to exit client                     |
|     -testsr [X] [Y] to test the sr          |
|     -test2sr [X] [Y] to test the sr         |
**********************************************
totalPacket is : 8

-time
2024/5/2 21:1:25
**********************************************
|     -time to get current time                |
|     -quit to exit client                     |
|     -testsr [X] [Y] to test the sr          |
|     -test2sr [X] [Y] to test the sr         |
**********************************************
-quit
Good bye!
PS E:\计算机网络\lab2\SR>
```

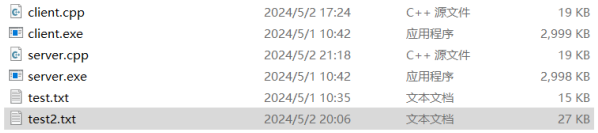
服务器端

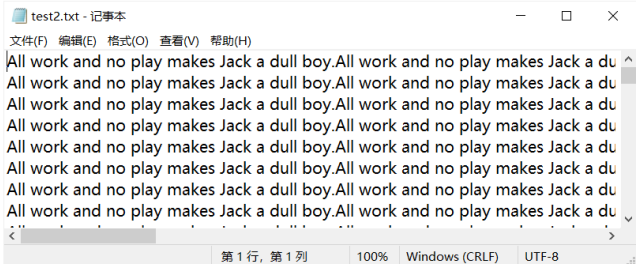
```
PS E:\计算机网络\lab2\SR> ./server
The Winsock 2.2 dll was found okay
totalPacket is: 11

recv from client: -time
recv from client: -quit
```

2. GBN协议的验证

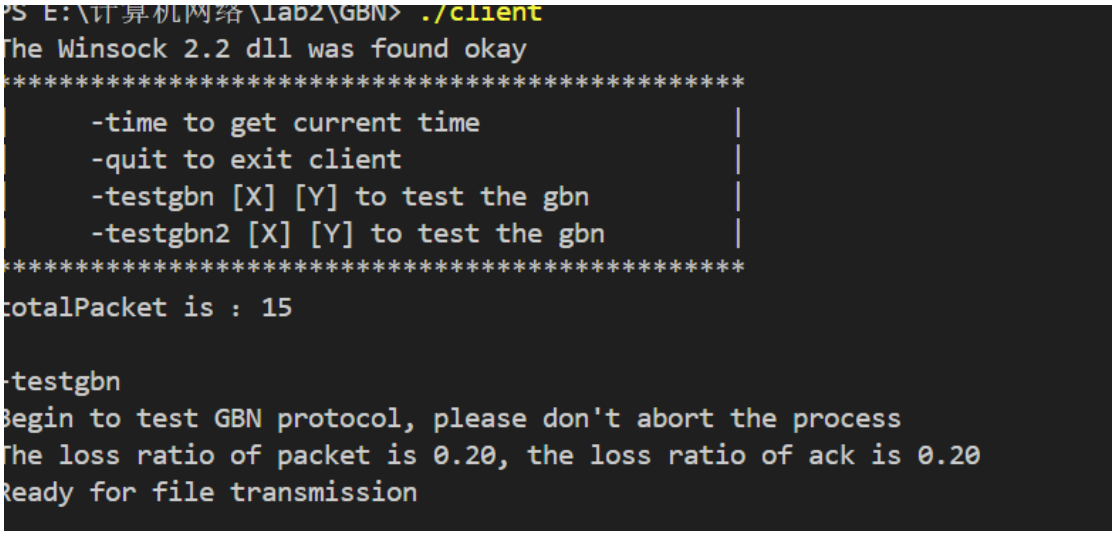
使用默认的丢包率和ack丢失率都是0.2的参数进行模拟，传输这样一

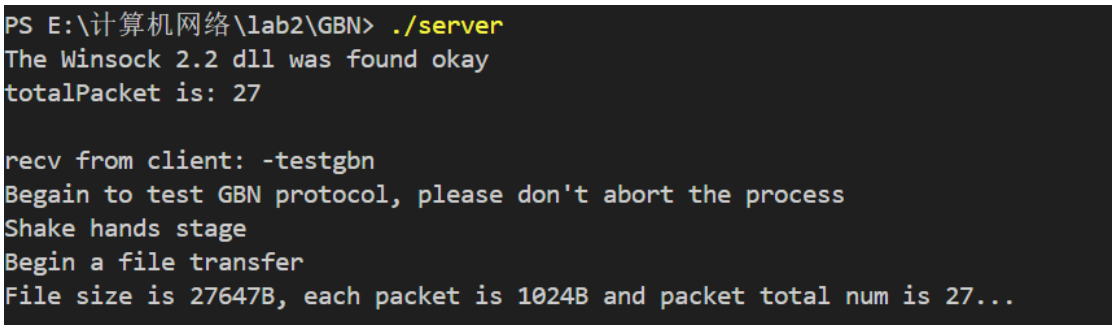
个txt文件



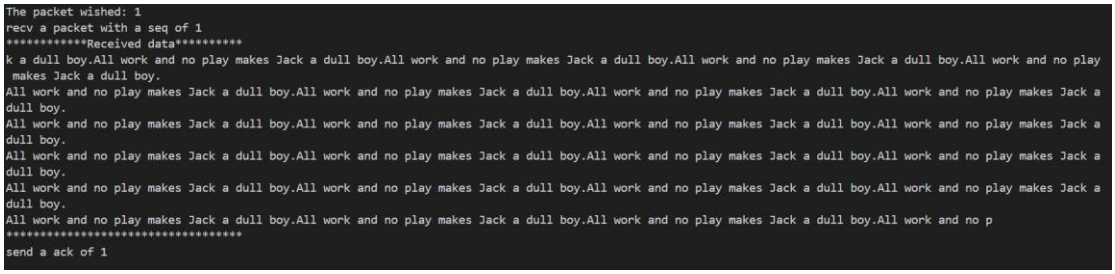


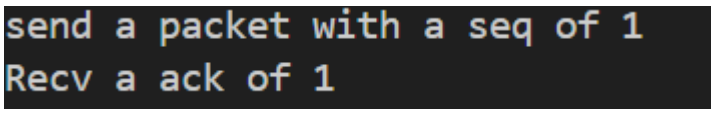
①客户端发起请求并成功通过握手建立起连接



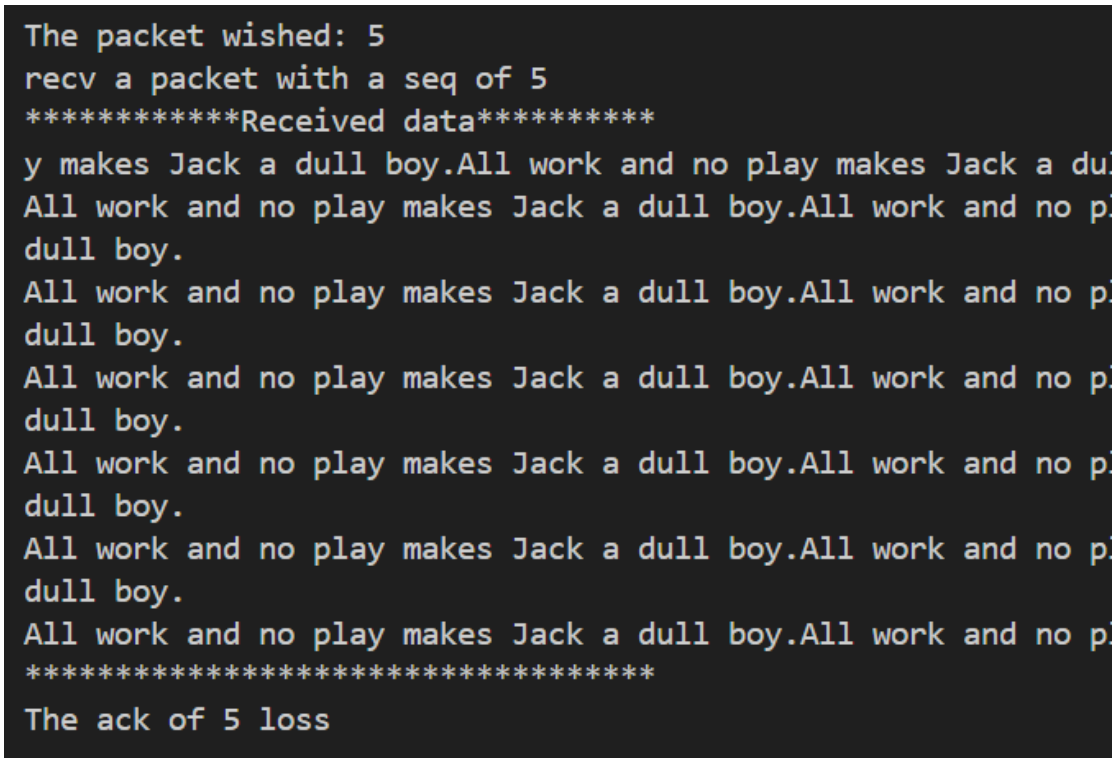


②开始发送和接收报文，数据报和ack都未丢失时，在控制台打印传输的报文，服务器端打印接收到的ack信号



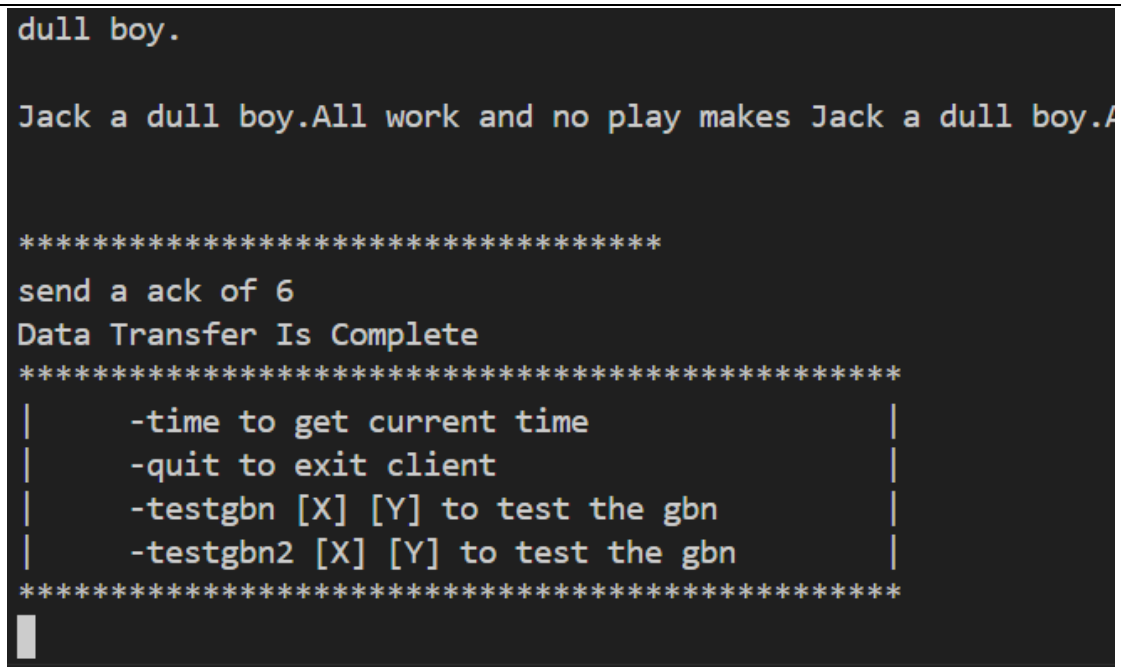


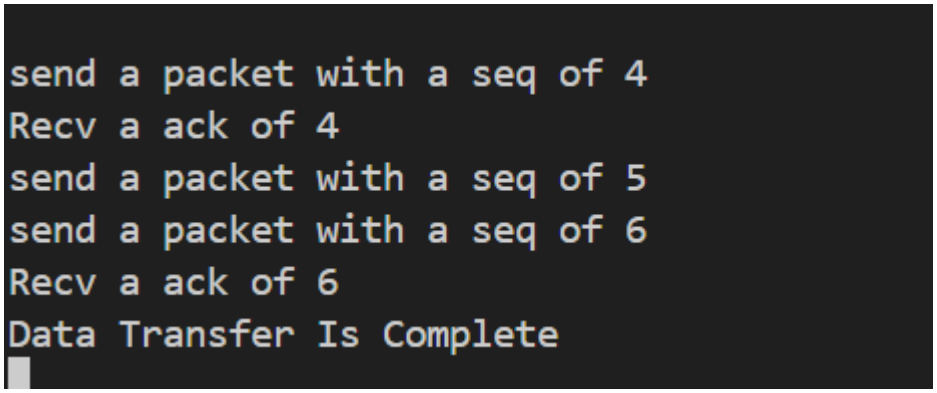
③当数据报和ack其中一个发生丢失时，触发超时重传机制，服务器端打印超时重传的信息



```
Recv a ack of 6
send a packet with a seq of 7
send a packet with a seq of 8
Recv a ack of 8
send a packet with a seq of 9
send a packet with a seq of 10
Recv a ack of 10
send a packet with a seq of 11
send a packet with a seq of 12
send a packet with a seq of 13
Recv a ack of 10
send a packet with a seq of 14
send a packet with a seq of 15
Recv a ack of 10
send a packet with a seq of 16
Recv a ack of 10
send a packet with a seq of 17
Recv a ack of 10
send a packet with a seq of 18
Recv a ack of 10
send a packet with a seq of 19
Recv a ack of 10
send a packet with a seq of 0
Recv a ack of 10
*****Time out
*****Rensend from Packet 11
```

④全部传输完成时，客户端和服务器端都打印完成的信息，客户端重新接收用户的输入，服务器端准备接收客户请求





3. 双向GBN的验证

在第2部分已经完成了从服务器端向客户端传输文件的过程。现在模拟从客户端向服务器端传输文件。

①客户端发起请求并成功通过握手建立起连接

```
**************************************************
|     -time to get current time                  |
|     -quit to exit client                       |
|     -testgbn [X] [Y] to test the gbn           |
|     -testgbn2 [X] [Y] to test the gbn          |
**************************************************
-testgbn2
Begin to test GBN protocol, please don't abort the process
Shake hands stage
Begin a file transfer
File size is 14977B, each packet is 1024B  and packet total num is 15...
```

②开始发送和接收报文，数据报和ack都未丢失时，在控制台打印传输的报文，服务器端打印接收到的ack信号

```
send a packet with a seq of 2
Recv an ack of 2
send a packet with a seq of 3
```

```
The packet wished: 2
recv a packet with a seq of 2
***********Received data**********
rld!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello Worl
llo World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hell
ld!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World
o World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello
!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!H
World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello Wo
ello World!Hello World!Hello World!Hello World!
**********************************
send a ack of 2
```

③当数据报和ack其中一个发生丢失时，触发超时重传机制，服务器端打印超时重传的信息

```
send a packet with a seq of 11
send a packet with a seq of 12
Recv an ack of 4
send a packet with a seq of 13
Recv an ack of 4
send a packet with a seq of 14
Recv an ack of 4
*****Time out
*****Rensend from Packet 5
```

```
The packet wished: 1
recv a packet with a seq of 1
***********Received data**********
o World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello
d!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World
 World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello
Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!H
orld!Hello World!Hello World!Hello World!Hello World!Hello World!Hello Wo
llo World!Hello World!Hello World!Hello World!Hello World!Hello World!Hel
ld!Hello World!Hello World!Hello World!Hello Wo
**********************************
The ack of 1 loss
```

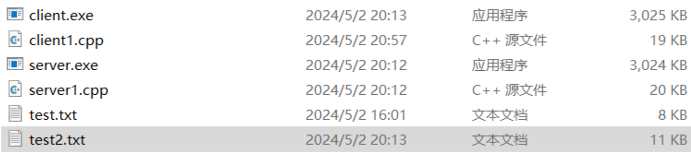
④全部传输完成时，客户端和服务器端都打印完成的信息，客户端重新接收用户的输入，服务器端准备接收客户请求

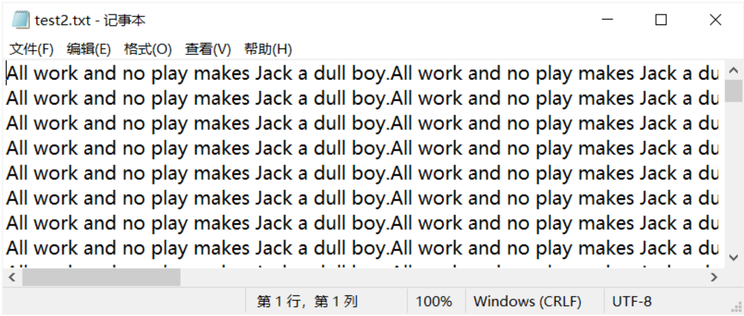
```
send a packet with a seq of 13
Recv an ack of 13
send a packet with a seq of 14
Recv an ack of 14
Data Transfer Is Complete
************************************************
|     -time to get current time                |
|     -quit to exit client                     |
|     -testgbn [X] [Y] to test the gbn         |
|     -testgbn2 [X] [Y] to test the gbn        |
************************************************
```

```
The packet wished: 14
recv a packet with a seq of 14
***********Received data**********
rld!Hello World!Hello World!Hello World!Hello World!Hello World!H
llo World!Hello World!Hello World!Hello World!Hello World!Hello W
ld!Hello World!Hello World!Hello World!Hello World!Hello World!He
o World!Hello World!Hello World!Hello World!Hello World!Hello Wor
**********************************
send a ack of 14
Data Transfer Is Complete
```

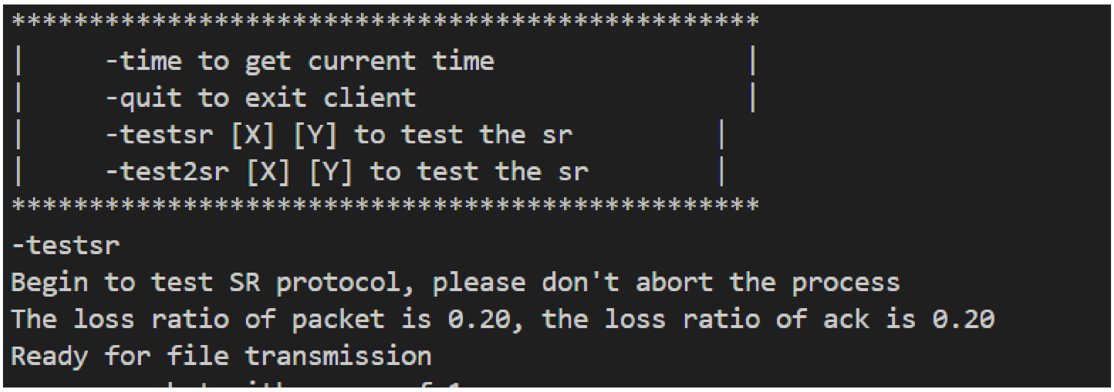
4. SR协议的验证

使用默认的丢包率和ack丢失率都是0.2的参数进行模拟，传输这样一个txt文件

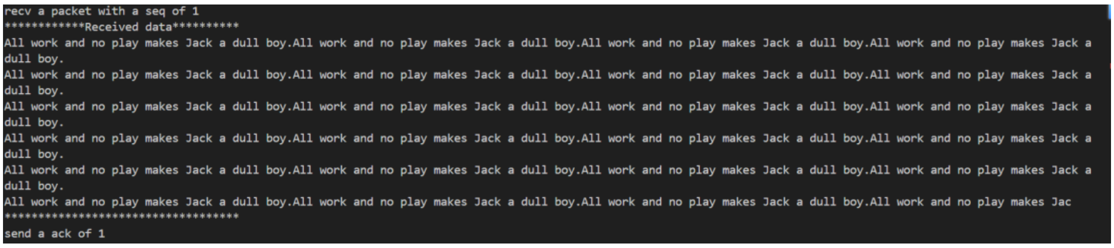




①客户端发起请求并成功通过握手建立起连接



②开始发送和接收报文，数据报和ack都未丢失时，在控制台打印传输的报文，服务器端打印接收到的ack信号



```
recv from client: -testsr
Begin to test SR protocol, please don't abort the process
Shake hands stage
Begin a file transfer
File size is 10910B, each packet is 1024B and packet total num is 11...

send a packet with a seq of 1
totalSeq now is: 1
Recv a ack of seq 1

############Windows move to 2##############
send a packet with a seq of 2
totalSeq now is: 2
Recv a ack of seq 2
```

③当数据报和ack其中一个发生丢失时，触发超时重传机制，服务器端打印超时重传的信息

```
recv a packet with a seq of 3
***********Received data*********
All work and no play makes Jack a dull boy.All work and no play makes Jack a
dull boy.
All work and no play makes Jack a dull boy.All work and no play makes Jack a
dull boy.
All work and no play makes Jack a dull boy.All work and no play makes Jack a
dull boy.
All work and no play makes Jack a dull boy.All work and no play makes Jack a
dull boy.
All work and no play makes Jack a dull boy.All work and no play makes Jack a
dull boy.
All work and no play makes Jack a dull boy.All work and no play makes Jack a
**********************************
The ack of 3 loss
```

```
totalSeq now is: 11
Recv a ack of seq 11
******Time out
*****Resend Packet 3

******Time out
*****Resend Packet 3

Recv a ack of seq 3
```

```
The packet with a seq of 4 loss
The packet with a seq of 5 loss
The packet with a seq of 6 loss
The packet with a seq of 7 loss
```

```
###########Windows move to 4###############
******Time out
*****Resend Packet 4

******Time out
*****Resend Packet 4

Recv a ack of seq 4

###########Windows move to 5###############
******Time out
*****Resend Packet 5

Recv a ack of seq 5

###########Windows move to 6###############
******Time out
*****Resend Packet 6

******Time out
*****Resend Packet 6

******Time out
*****Resend Packet 6

******Time out
*****Resend Packet 6
```

这里运气不好，6号需要反复重传

④全部传输完成时，客户端和服务器端都打印完成的信息，客户端重新接收用户的输入，服务器端准备接收客户请求

```
**********Received data**********
All work and no play makes Jack a dull boy.All
dull boy.
All work and no play makes Jack a dull boy.All
dull boy.
All work and no play makes Jack a dull boy.All
dull boy.
All work and no play makes Jack a dull boy.All
dull boy.
All work and no play makes Jack a dull boy.All
dull boy.
All work and no play makes Jack a dull boy.All
**********************************
send a ack of 7
Data Transfer Is Complete
```

5. 双向SR的验证

在第4部分已经完成了从服务器端向客户端传输文件的过程。现在模拟从客户端向服务器端传输文件。

①文件信息

名称	修改日期	类型	大小
client.exe	2024/5/2 20:13	应用程序	3,025 KB
client1.cpp	2024/5/2 20:57	C++ 源文件	19 KB
server.exe	2024/5/2 20:12	应用程序	3,024 KB
server1.cpp	2024/5/2 20:12	C++ 源文件	20 KB
test.txt	2024/5/2 16:01	文本文档	8 KB
test2.txt	2024/5/2 20:13	文本文档	11 KB

②客户端发起请求，经历握手后完成连接。

```
-test2sr 0.1 0.1
Begin to test SR protocol, please don't abort the process
Shake hands stage
Begin a file transfer
File size is 7184B, each packet is 1024B and packet total num is 8...
```

```
recv from client: -test2sr
Begin to test SR protocol, please don't abort the process
The loss ratio of packet is 0.20, the loss ratio of ack is 0.20
Ready for file transmission
```

注意这回是客户端打印相关文件信息，服务器端准备接收。

③开始发送和接收报文，数据报和ack都未丢失时，服务器端控制台打印传输的报文，客户端打印接收到的ack信号

```
############Windows move to 3################
send a packet with a seq of 4
totalSeq now is: 4
Recv a ack of seq 3

############Windows move to 4################
send a packet with a seq of 5
totalSeq now is: 5
Recv a ack of seq 4
```

```
recv a packet with a seq of 4
***********Received data**********
Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hell
orld!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World
llo World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello
ld!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!H
o World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello Wo
!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hello World!Hel
World!Hello World!Hello World!Hello World!Hell
**********
send a ack of 4
```

注意到SR协议允许乱序接收的特点。

④全部传输完成时，客户端和服务器端都打印完成的信息，客户端重新接收用户的输入，服务器端准备接收客户请求

```
###########Windows move to 9###############
Data Transfer Is Complete
```

```
send a ack of 8
Data Transfer Is Complete!
```

问题讨论：

1.UDP 编程的主要特点

- 无连接：UDP 不需要在发送数据前建立连接，而是直接发送数据，因此其速度快于需要建立连接的协议。
- 无状态：UDP 协议没有保存之前发送或接收的数据报的状态，每个数据报都是独立处理的。
- 不保证可靠性：UDP 协议不保证数据报的到达，尽力而为。
- 数据报结构简单。
- 支持一对一、一对多、多对多、多对一的交互通信。

2.终端打印中文乱码

使用命令chcp 65001进行设置

3.带有详细注释的源代码

见附件

心得体会：

- 加深了对UDP传输、停等协议、GBN协议、SR协议的理解
- 熟悉了socket编程方法
- 巩固了c++代码调试方法