



哈尔滨工业大学
Harbin Institute of Technology

计算机网络 课程实验报告

实验名称	可靠数据传输协议-停等协议/GBN 协议的设计与实现					
姓名	傅浩东		院系	软件工程		
班级	1937102		学号	1190202105		
任课教师	李全龙		指导教师	李全龙		
实验地点	格物 207		实验时间	2021.11.6		
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						

实验目的：**本次实验的主要目的：**

1. 理解可靠数据传输的基本原理；
2. 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；
3. 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；
4. 掌握 SR 的工作原理；掌握基于 UDP 设计并实现一个 SR 协议的过程与技术。

实验内容：

概述本次实验的主要内容，包含的实验项等。

1. 基于UDP设计一个简单的GBN协议（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 GBN 协议（停等协议），支持双向数据传输；
4. 将所设计的 GBN 协议改进为 SR 协议；
5. 基于所设计的停等协议，实现一个 C/S 结构的文件传输应用。
6. 值得注意的是，这里如果设置服务器端发送窗口的大小为1时，GBN协议就是停等协议。

实验过程：

以文字描述、实验结果截图等形式阐述实验过程，必要时可附相应的代码截图或以附件形式提交。

一. 单向数据传输的GBN协议，实质上就是实现为一个C/S应用。

服务器端：使用UDP协议传输数据（比如传输一个文件），等待客户端的请求，接收并处理来自客户端的消息（如数据传输请求），当客户端开始请求数据时进入“伪连接”状态（并不是真正的连接，只是一种类似连接的数据发送的状态），将数据打包成数据报发送，然后等待客户端的ACK信息，同时启动计时器。当收到ACK时，窗口滑动，正常发送下一个数据报，计时器重新计时；若在计时器超时前没有收到ACK，则全部重传窗口内的所以已发送的数据报。

客户端：使用UDP协议向服务器端请求数据，接收服务器端发送的数据报并返回确认信息ACK。

1. **服务器命令解析：**当客户端发送请求“-time”来获取当前时间，服务器收到该请求后回复当前时间。当客户端发送请求信息“-testgbn”，然后服务器端解析请求，进行一个握手阶段，首先服务器向客户端发送状态码205，表示服务器可以发送数据；客户端收到205之后回复状态码200，表示客户端可以接收数据；服务器收到200状态码之后，就开始使用GBN协议发送数据，服务器端读取本地文件，放到缓存中，发送给客户端。当客户端发送退出请求“-quit”，服务器回复“Good bye!”

```
case 0: // 发送 205 阶段
    buffer[0] = 205;
    buffer[1] = '\0';
    sendto(sockServer, buffer, strlen(buffer)+1, 0,
(SOCKADDR*)&addrClient, sizeof(SOCKADDR));
    Sleep(100);
    stage = 1;
    break;
case 1: // 等待接收 200 阶段
```

```

        recvSize = recvfrom(sockServer, buffer,
BUFFER_LENGTH, 0, ((SOCKADDR*)&addrClient), &length);
        if(recvSize < 0) {
            waitCount ++;
            if(waitCount > 20) {
                runFlag = false;
                printf("200 Timeout error\n");
                break;
            }
            Sleep(500);
            continue;
        } else {
            // waitCount = 0;
            if((unsigned char)buffer[0] == 200) {
                printf("Begin a file transfer\n");
                printf("File size is %dB, each packet
is 1024B and packet total num is %d...\n\n", sizeof(data),
totalPacket);

                curSeq = 0;
                curAck = 0;
                totalSeq = 0;
                waitCount = 0;
                stage = 2;
            }
        }
        break;

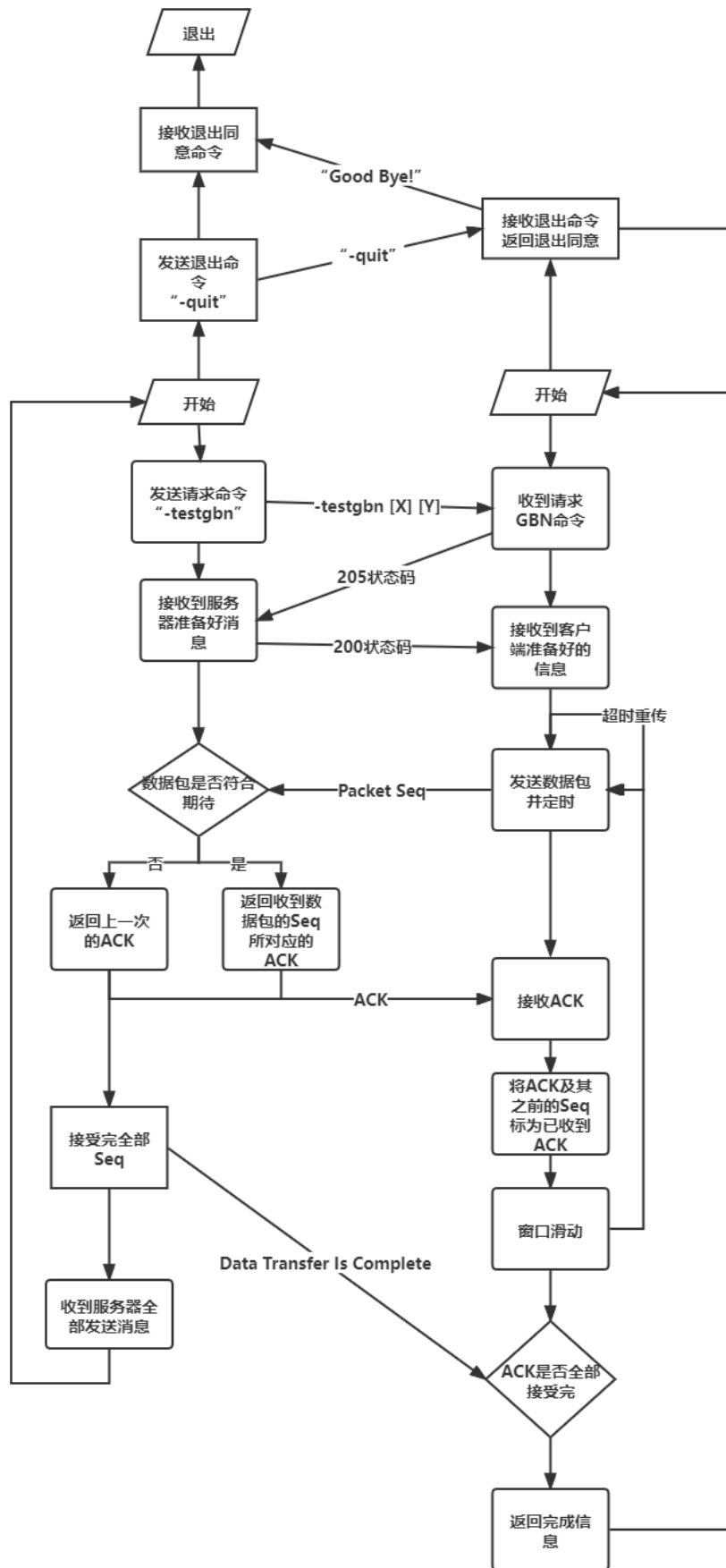
```

2. **客户端命令解析：**客户端的命令和服务端端的解析命令向对应，获取当前用户输入并发送给服务器并等待服务器返回数据。这里注意“-testgbn [X] [Y]”命令，其中[X] [Y]是两个从零到一的小数，表示服务器发送包丢失的概率和客户端ACK丢失概率，当没有输入时默认两个丢失率均为0.2.

```

case 0://等待握手阶段
    u_code = (unsigned char)buffer[0];
    if((unsigned char)buffer[0] == 205) {
        printf("Ready for file transmission\n");
        buffer[0] = 200;
        buffer[1] = '\0';
        sendto(socketClient, buffer, 2, 0,
(SOCKADDR*)&addrServer, sizeof(SOCKADDR));
        stage = 1;
        recvSeq = 0;
        waitSeq = 1;
    }
    break;
case 1://等待接收数据阶段

```



3. **GBN协议的解析：**ACK采用累积确认（取数据帧的第一个字节，为ACK对应分组的序列号），对于服务器来说，当收到一个字段的序列号时，在其之前的所有分组全都确认被收到。当发生超时情况时，发送端重新发送整个窗口中的所有数据分组。对于客户端来说，若来自服务器的数据包丢失，那么就之后的ACK都不会继续发送（在这里处理方法是继续发送上一个序列号）。

4. 数据传输数据帧格式定义

在以太网中，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），为此定义UDP数据报的数据部分格式为：

Seq	Data	0
-----	------	---

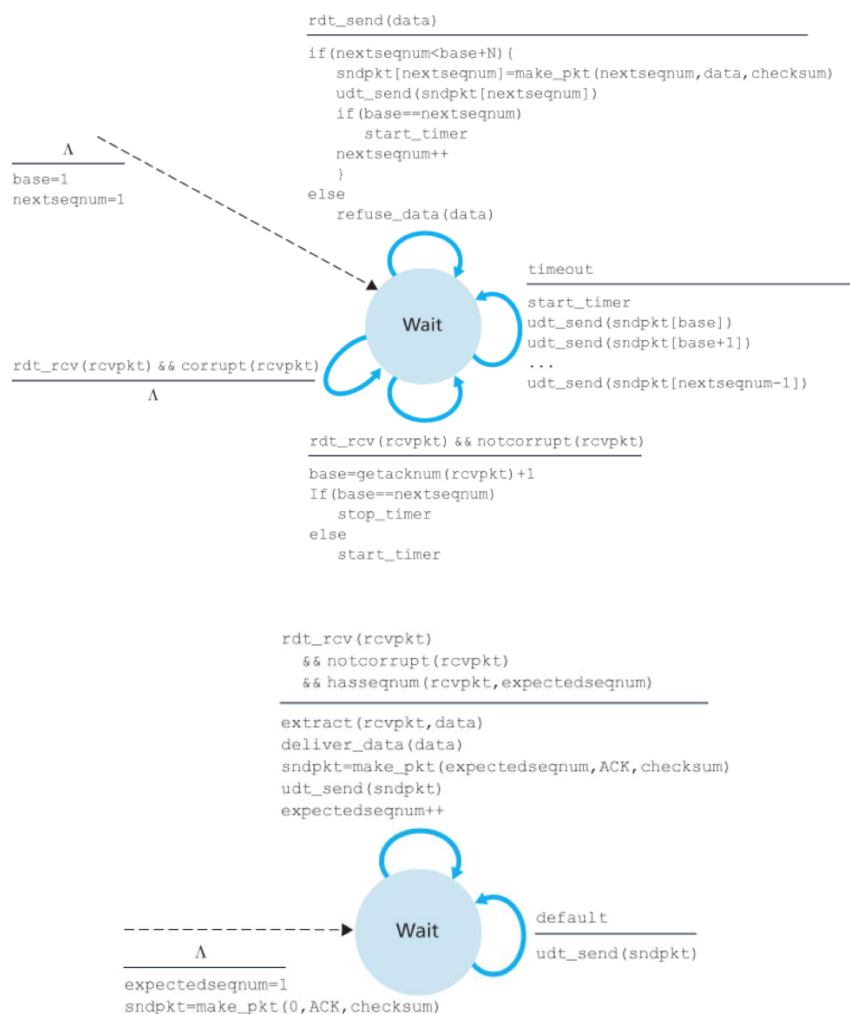
Seq为1个字节，取值为 0~255，（故序列号最多为256个，在实验中设置为20个序列号）；Data≤1024 个字节，为传输的数据； 最后一个字节放入 EOF0，表示结尾。

5. ACK数据帧格式定义

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（序列号）发送给服务器端即可，其中ACK占一个字节，末尾用“\0”结束。

ACK	\0
-----	----

具体有：

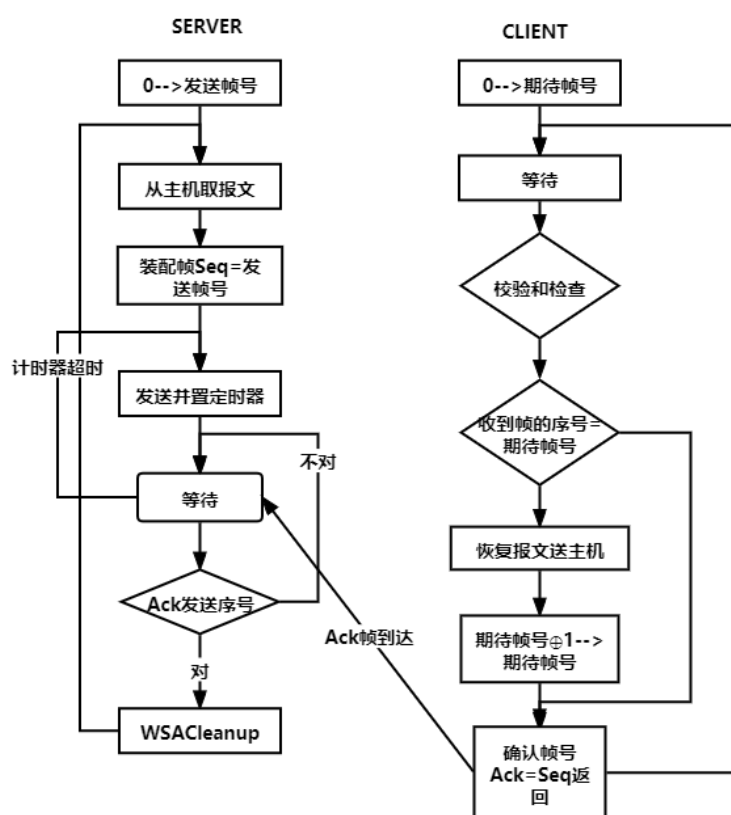


二. GBN 双向传输原理相同，但客户端与服务器同时具有发送和接收功能。

客户端发送“-testgbn_Duplex [X][Y]”命令，表示自己要发送文件，当服务器收到命令后，将socket设置为阻塞模式，由此来接收来自客户端的数据包。也就是说客户端要有服务器发送功能，服务器要有客户端接收功能。

三. 对于停等协议，将GBN协议滑动窗口大小设置为1即可，这里就不再赘述。

当发送窗口和接收窗口的大小固定为1时，滑动窗口协议退化为停等协议（stop-and-wait）。该协议规定发送方每发送一帧后就要停下来，等待接收方已正确接收的确认（acknowledgement）返回后才能继续发送下一帧。由于接收方需要判断接收到的帧是新发的帧还是重新发送的帧，因此发送方要为每一个帧加一个序号。由于停等协议规定只有一帧完全发送成功后才能发送新的帧，因而只用一比特来编号就够了。其发送方和接收方运行的流程图如下所示。



四. SR协议

SR协议（即选择重传协议），顾名思义与GBN协议（回退N步协议）不同点主要在于错误报文的重传机制。SR协议的接收方设置缓冲区，为每个报文段设置计时器，如果某个报文段没有被正确接收但是后面的报文段被正确接收了，那么就只需要服务器重发这一个报文段。由此，自然而然地发送方和接收方数据包的状态就不止两种，其中发送方（服务器）的状态有：已发送且已收到ACK（当滑动窗口首个数据包是该状态时才滑动）、已发送且未收到ACK、可用序号，接收方（客户端）的状态：已接受并已提交、已接收已确认但未提交（滑动窗口前有未确认的数据包）、可接受。

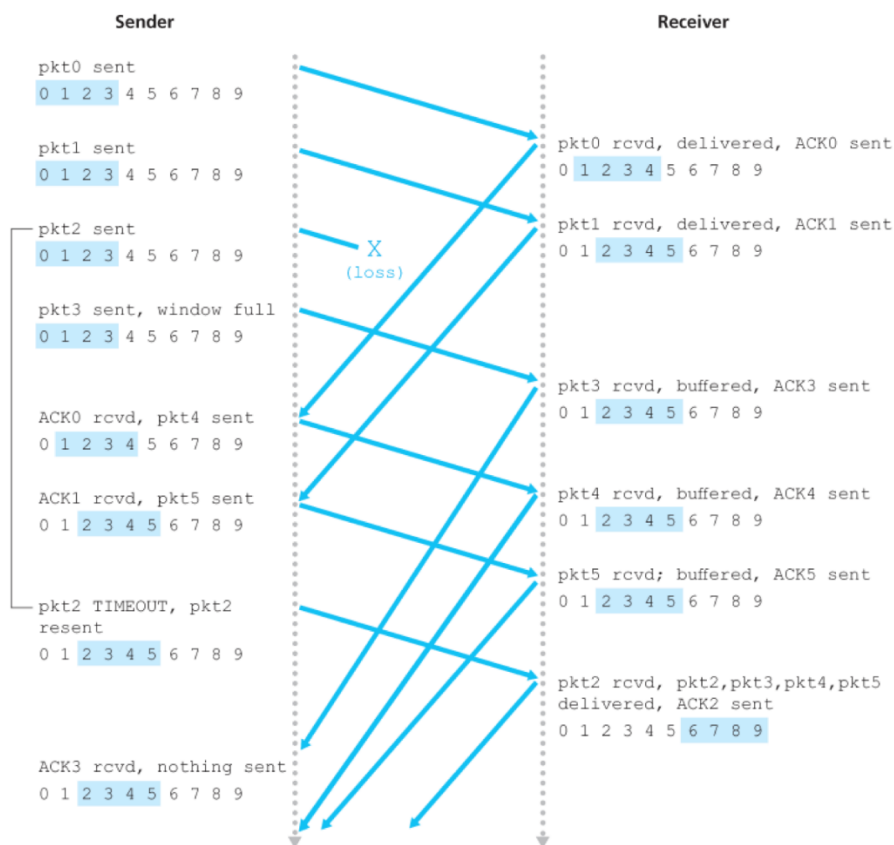
// totalSeq<=(totalPacket-1): 未传到最后一个数据包

if(seqIsAvailable() == 1 && totalSeq <= (totalPacket - 1)) {

//发送给客户端的序列号从 1 开始

```
buffer[0] = curSeq + 1;
ack[curSeq] = FALSE;
//数据发送的过程中应该判断是否传输完成->现在此代码已经实现了 ok
//为简化过程此处并未实现->现在此代码已经实现了 ok
memcpy(&buffer[1], data + 1024 * totalSeq, 1024);
printf("send a packet with a seq of: %d \n", curSeq + 1);
// printf("totalSeq now is: %d\n", totalSeq+1);
sendto(sockServer, buffer, BUFFER_LENGTH, 0,
(SOCKADDR*)&addrClient, sizeof(SOCKADDR));
++curSeq;
curSeq %= SEQ_SIZE;
++totalSeq;
Sleep(500);
}
else if(seqIsAvailable() == 2 && totalSeq <= (totalPacket - 1)) {
++curSeq;
curSeq %= SEQ_SIZE;
++totalSeq;
break;
}
//等待 Ack, 若没有收到, 则返回值为-1, 计数器+1
recvSize = recvfrom(sockServer, buffer, BUFFER_LENGTH, 0,
((SOCKADDR*)&addrClient), &length);
if(recvSize < 0) {
waitCount++;
//20 次等待 ack 则超时重传
if (waitCount > 20) {
timeoutHandler();
waitCount = 0;
}
} else {
//收到 ack
ackHandler(buffer[0]);
waitCount = 0;
if(totalAck == totalPacket) { //数据传输完成
printf("Data Transfer Is Complete!!!!\n");
strcpy(buffer, "Data Transfer Is Complete\n");
runFlag = false;
break;
}
}
Sleep(500);
break;
```

理论上选择重传协议要为每个分组使用一个计时器。当某个计时器超时后，只有相应的分组被重传，换言之，返回N协议将所有的分组当做一个整体对待，而选择重传协议则分别对待每一个分组，但是大多数SR的运输层仅使用了一个计时器。注意只使用一个计时器而做到跟踪所有发出去的分组的情况的做法是：标记发出分组，当ACK=Sf时，将窗口滑过所有连续的已确认的分组，如果还有未确认的分组，则重发所有检测到的未被确认的分组并重启计时器，如果所有分组都被确认了则停止计时器。



实验结果：

采用演示截图、文字说明等方式，给出本次实验的实验结果。

一．GBN时间命令“-time”测试

```
The Winsock 2.2 dll was found okay
*****
| -time to get current time |
| -quit to exit client |
| -testgbn [X] [Y] to test the gbn |
| -testgbn_Duplex [X] [Y] to test the gbn |
*****
totalPacket is : 11
-time

The Winsock 2.2 dll was found okay
totalPacket is: 7
recv from client: -time
```

二．GBN无错误命令“-testgbn 0 0”测试

当客户端发送命令“-testgbn 0 0”，将ACK丢失率和数据包丢失率都设置为0，那么对于服务器每次发送数据包之后都会有收到一个来自客户端的ACK确认信息，而对于客户端每次都会如期待值一样收到来自服务器的数据包（与期待序列相同），同时也会发送一个

确认ACK，这个过程有如下结果表示：

```
*****
| -time to get current time |
| -quit to exit client |
| -testgbn [X] [Y] to test the gbn |
| -testgbn_Duplex [X] [Y] to test the gbn |
*****
-testgbn 0 0
Begin to test GBN protocol, please don't abort the process
The loss ratio of packet is 0.00, the loss ratio of ack is 0.00
Ready for file transmission

recv a packet with a seq of 2
send a ack of 2

The packet wished: 3
recv a packet with a seq of 3
send a ack of 3

The packet wished: 4
recv a packet with a seq of 4
send a ack of 4

The packet wished: 5
recv a packet with a seq of 5
send a ack of 5

The packet wished: 6
recv a packet with a seq of 6
send a ack of 6
Data Transfer Is Complete
```

```
recv from client: -testgbn
Begin to test GBN protocol, please don't abort the process
Shake hands stage
Begin a file transfer
File size is 7034B, each packet is 1024B and packet total num is 7...

send a packet with a seq of 0
Recv a ack of 0
send a packet with a seq of 1
Recv a ack of 1
send a packet with a seq of 2
Recv a ack of 2
send a packet with a seq of 3
Recv a ack of 3
send a packet with a seq of 4
Recv a ack of 4
send a packet with a seq of 5
Recv a ack of 5
send a packet with a seq of 6
Recv a ack of 6
Data Transfer Is Complete
```

三. GBN有错误命令“-testgbn”测试

如下所示，当ACK丢失时，就算接下来接收到了其他Seq的数据包也不再继续发送往后的ACK，超时之后从ACK丢失的地方开始全部重传，数据包丢失同理：

```

The Winsock 2.2 dll was found okay
totalPacket is: 7

recv from client: -testgbn
Begin to test GBN protocol, please don't abort the process
Shake hands stage
Begin a file transfer
File size is 7034B, each packet is 1024B and packet total num is 7...

send a packet with a seq of 0
Recv a ack of 0
send a packet with a seq of 1
send a packet with a seq of 2
Recv a ack of 0
Recv a ack of 0

Timer out error.....
Rensend from 1 Packet.....

send a packet with a seq of 1
Recv a ack of 1
send a packet with a seq of 2
Recv a ack of 2
send a packet with a seq of 3
send a packet with a seq of 4
send a packet with a seq of 5
Recv a ack of 2
send a packet with a seq of 6
Recv a ack of 2

Timer out error.....
Rensend from 3 Packet.....

send a packet with a seq of 3
Recv a ack of 3
send a packet with a seq of 4
send a packet with a seq of 5
send a packet with a seq of 6

Timer out error.....
Rensend from 4 Packet.....

```

```

The Winsock 2.2 dll was found okay
*****
| -time to get current time |
| -quit to exit client |
| -testgbn [X] [Y] to test the gbn |
| -testgbn_Duplex [X] [Y] to test the gbn |
*****
totalPacket is : 11

-testgbn
Begin to test GBN 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

The packet wished: 0
recv a packet with a seq of 0
send a ack of 0

The packet wished: 1
The packet with a seq of 1 loss

The packet wished: 1
recv a packet with a seq of 2
send a ack of 0

The packet wished: 1
recv a packet with a seq of 3
send a ack of 0

The packet wished: 1
recv a packet with a seq of 4
send a ack of 0

The packet wished: 1
recv a packet with a seq of 5
The ack of 0 loss

```

```

The packet wished: 1
recv a packet with a seq of 6
send a ack of 0

The packet wished: 1
recv a packet with a seq of 1
send a ack of 1

The packet wished: 2
recv a packet with a seq of 2
send a ack of 2

The packet wished: 3
The packet with a seq of 3 loss

The packet wished: 3
recv a packet with a seq of 4
The ack of 2 loss

send a ack of 2

The packet wished: 3
recv a packet with a seq of 3
send a ack of 3

The packet wished: 4
The packet with a seq of 4 loss

The packet wished: 4
recv a packet with a seq of 5
The ack of 3 loss

The packet wished: 4
The packet with a seq of 6 loss

The packet wished: 4
recv a packet with a seq of 4
send a ack of 4

The packet wished: 5
recv a packet with a seq of 5
The ack of 5 loss

The packet wished: 6
recv a packet with a seq of 6
send a ack of 6
Data Transfer Is Complete

```

四. 退出命令“-quit”测试

```

*****
| -time to get current time |
| -quit to exit client |
| -testgbn [X] [Y] to test the gbn |
| -testgbn_Duplex [X] [Y] to test the gbn |
*****
-quit
Good bye!
PS D:\Learn\Computer_Network\Lab\Lab2\GBN>
recv from client: -quit

```

五. GBN协议的双向传输“-testgbn_Duplex”命令测试

同理，只不过这里客户端作为发送方，而服务器作为接受方，服务器首先接收到来自客户端的命令，这样与客户端进行三次握手接下来传输数据，结束发送传输结束信号。

```

*****
| -time to get current time |
| -quit to exit client |
| -testgbn [X] [Y] to test the gbn |
| -testgbn Duplex [X] [Y] to test the gbn |
*****
-testgbn_Duplex
Begin to test GBN protocol, please don't abort the process
Shake hands stage
Begin a file transfer
File size is 11049B, each packet is 1024B and packet total num is 11...

send a packet with a seq of 0
send a packet with a seq of 1
Recv an ack of 1
send a packet with a seq of 2
Recv an ack of 2
send a packet with a seq of 3
Recv an ack of 3
send a packet with a seq of 4
Recv an ack of 4
send a packet with a seq of 5
Recv an ack of 5
send a packet with a seq of 6
send a packet with a seq of 7
send a packet with a seq of 8
send a packet with a seq of 9
Recv an ack of 5
send a packet with a seq of 10

```

```

Timer out error.....
Rensend from 6 Packet.....
send a packet with a seq of 6
send a packet with a seq of 7
Recv an ack of 5
send a packet with a seq of 8
Recv an ack of 5
send a packet with a seq of 9
Recv an ack of 5
send a packet with a seq of 10

Timer out error.....
Rensend from 6 Packet.....
send a packet with a seq of 10
Recv an ack of 8

Timer out error.....
Rensend from 9 Packet.....
send a packet with a seq of 9
Recv an ack of 9
send a packet with a seq of 10
Recv an ack of 10
Data Transfer Is Complete

```

```

recv from client: -testgbn_Duplex
Begin to test GBN 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

The packet wished: 0
recv a packet with a seq of 0
The ack of 0 loss

The packet wished: 1
recv a packet with a seq of 1
send a ack of 1

The packet wished: 2
recv a packet with a seq of 2
send a ack of 2

The packet wished: 3
recv a packet with a seq of 3
send a ack of 3

```

```
The packet wished: 4
recv a packet with a seq of 4
send a ack of 4

The packet wished: 5
recv a packet with a seq of 5
send a ack of 5

The packet wished: 6
The packet with a seq of 6 loss

The packet wished: 6
recv a packet with a seq of 7
The ack of 5 loss

The packet wished: 6
The packet with a seq of 8 loss

The packet wished: 6
recv a packet with a seq of 9
send a ack of 5

The packet wished: 6
The packet with a seq of 10 loss

The packet wished: 6
The packet with a seq of 6 loss

The packet wished: 6
recv a packet with a seq of 7
send a ack of 5
```

```
The packet wished: 6
The packet with a seq of 10 loss

The packet wished: 6
recv a packet with a seq of 6
send a ack of 6

The packet wished: 7
recv a packet with a seq of 7

The packet wished: 9
The packet with a seq of 9 loss

The packet wished: 9
recv a packet with a seq of 10
send a ack of 8

The packet wished: 9
recv a packet with a seq of 9
send a ack of 9

The packet wished: 10
recv a packet with a seq of 10
send a ack of 10
Data Transfer Is Complete
```

六. 停等协议就是窗口为1的GBN协议，这里就不再赘述

七. SR协议传输测试

```
*****
-testsr
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
recv a packet with a seq of 1
The ack of 1 loss
recv a packet with a seq of 2
send a ack of 2
recv a packet with a seq of 3
send a ack of 3
The packet with a seq of 4 loss
recv a packet with a seq of 5
The ack of 5 loss
recv a packet with a seq of 6
The ack of 6 loss
recv a packet with a seq of 7
send a ack of 7
recv a packet with a seq of 1
send a ack of 1
recv a packet with a seq of 4
send a ack of 4
recv a packet with a seq of 5
send a ack of 5
recv a packet with a seq of 6
send a ack of 6
Data Transfer Is Complete
```

```
The Winsock 2.2 dll was found okay
totalPacket is: 7

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 7034B, each packet is 1024B and packet total num is 7
send a packet with a seq of: 1
send a packet with a seq of: 2
Recv a ack of seq 2
send a packet with a seq of: 3
Recv a ack of seq 3
send a packet with a seq of: 4
send a packet with a seq of: 5
send a packet with a seq of: 6
send a packet with a seq of: 7
Recv a ack of seq 7

Timer out error.
Rensend from 1 Packet.....

send a packet with a seq of: 1
Recv a ack of seq 1
send a packet with a seq of: 4
Recv a ack of seq 4
send a packet with a seq of: 5
Recv a ack of seq 5
send a packet with a seq of: 6
Recv a ack of seq 6
Data Transfer Is Complete!!!!
recv from client: Data Transfer Is Complete
```

问题讨论:

1. 停等协议与 GBN 协议之间转换

发送窗口大小为 W (实验中设置为 10), GBN 中应满足 $W + 1 \leq N$, 其中 W 为发送窗口大小, N 为序列号个数 (实验中设置为 20)。当接收发送窗口的大小变为 1 时, GBN 协议则退化为停等协议。

2. 发送端与接收端在实现三次握手

- (1) 首先服务器向客户端发送 205 状态码 (客户端进入准备发送状态);
- (2) 客户端等待服务器回复 200 状态码, 如果收到表示客户端准备好了接收数据包, 则开始传输文件, 否则延时等待直至超时;
- (3) 开始使用 GBN 协议发送数据。

3. GBN 协议改变为 SR 协议

设置一个缓冲区记录接收端接收的分组序号, 若有分组超时未到, 则通知发送端分组序号, 让其只重传对应分组的数据帧。当然在实现发送端数据分包的过程中已经使用数组, 对应找到其下标即可。

心得体会:

结合实验过程和结果给出实验的体会和收获。

进一步理解了可靠数据传输的基本原理; 进一步掌握停等协议的工作原理, 并且基于 UDP 设计实现了一个停等协议的过程与技术; 进一步掌握了 GBN 的工作原理, 并且基于 UDP 设计并实现了一个 GBN 协议的过程与技术; 进一步掌握了 SR 的工作原理, 并且基于 UDP 设计并实现了一个 SR 协议的过程与技术; 最后让我对 socket 编程有了进一步的了解。