哈爾濱工業大學
Harbin Institute of Technology

计算机网络
课程实验报告

<table>
<tr><td>实验名称</td><td colspan="6">可靠数据传输协议- 停等协议与 GBN 的设计与实现</td></tr>
<tr><td>姓名</td><td colspan="2">李旻翀</td><td>院系</td><td colspan="3">计算机科学与技术</td></tr>
<tr><td>班级</td><td colspan="2">1903103</td><td>学号</td><td colspan="3">1190200208</td></tr>
<tr><td>任课教师</td><td colspan="2">刘亚维</td><td>指导教师</td><td colspan="3">刘亚维</td></tr>
<tr><td>实验地点</td><td colspan="2">格物 207</td><td>实验时间</td><td colspan="3">2021.11.07</td></tr>
<tr><td rowspan="2">实验课表现</td><td>出勤、表现得分(10)</td><td></td><td rowspan="2">实验报告得分(40)</td><td rowspan="2"></td><td rowspan="2">实验总分</td><td rowspan="2"></td></tr>
<tr><td>操作结果得分(50)</td><td></td></tr>
<tr><td colspan="7">教师评语</td></tr>
<tr><td colspan="7"></td></tr>
</table>

计算机科学与技术学院 SINCE 1956...
School of Computer Science and Technology

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；（选作内容，加分项目，可以当堂完成或课下完成）
4）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。（选作内容，加分项目，可以当堂完成或课下完成）
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；（选作内容，加分项目，可以当堂完成或课下完成）
8）将所设计的 GBN 协议改进为 SR 协议。（选作内容，加分项目，可以当堂完成或课下完成）

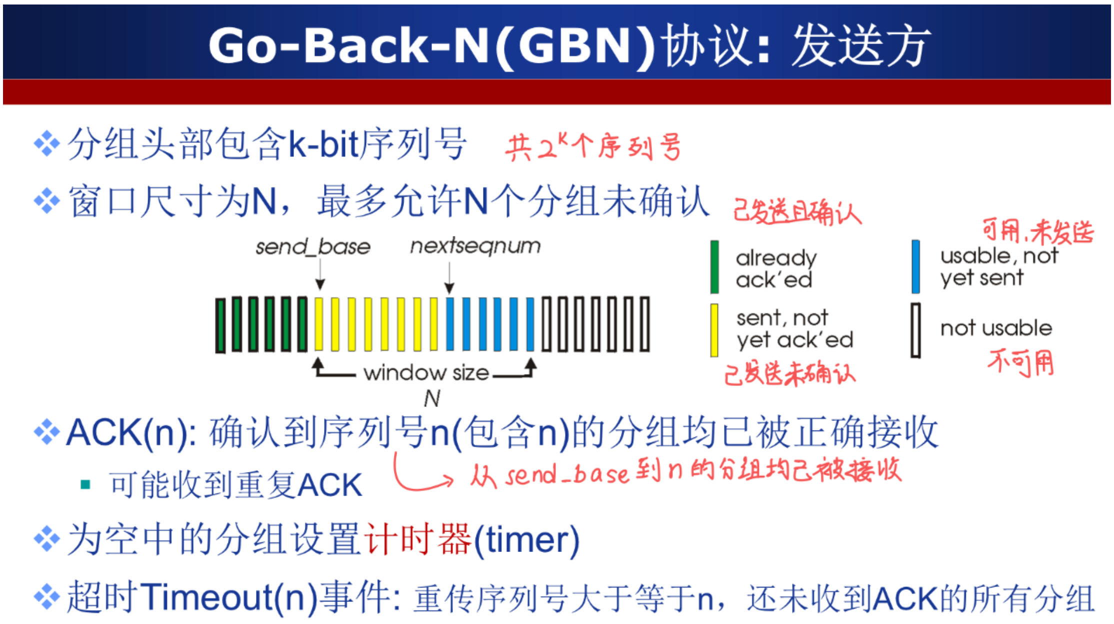
实验过程：

1. 了解实验相关基础知识

1) GBN协议

GBN协议，即回退N步协议(go back N),如果某个报文段没有被正确的接收，那么从这个报文段到后面的报文段都要重新发送，在GBN协议中，返回某个ACK，则代表该ACK序号及其之前的报文段均已经收到。

GBN协议的细节如下：



Go-Back-N(GBN)协议：发送方

❖分组头部包含k-bit序列号 共2^k个序列号

❖窗口尺寸为N，最多允许N个分组未确认

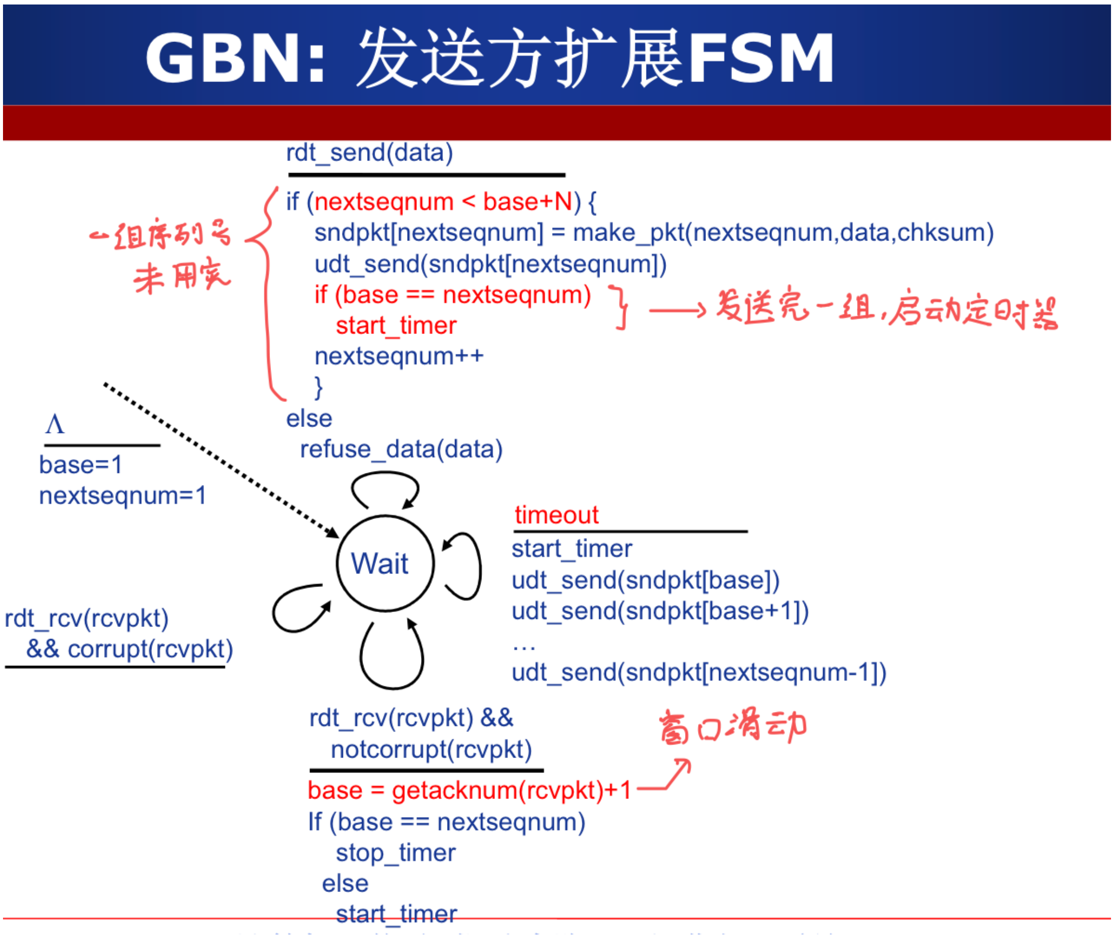
❖ACK(n): 确认到序列号n(包含n)的分组均已被正确接收

- 可能收到重复ACK 从 send_base 到 n 的分组均已被接收

❖为空中的分组设置计时器(timer)

❖超时Timeout(n)事件: 重传序列号大于等于n，还未收到ACK的所有分组

GBN协议中，发送方与接收方的FSM自动机如下：



GBN: 发送方扩展FSM

```
rdt_send(data)
_______________
if (nextseqnum < base+N) {
    sndpkt[nextseqnum] = make_pkt(nextseqnum,data,chksum)
    udt_send(sndpkt[nextseqnum])
    if (base == nextseqnum)
      start_timer
    nextseqnum++
    }
else
  refuse_data(data)
```

一组序列号未用完

发送完一组，启动定时器

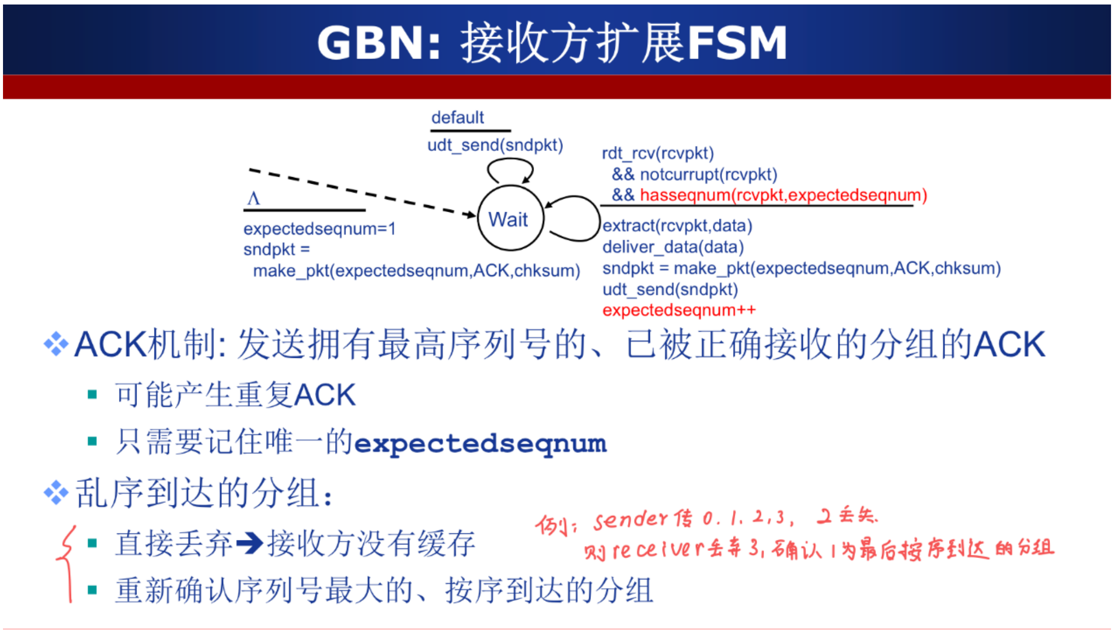
```
Λ
_______________
base=1
nextseqnum=1
```

```
timeout
_______________
start_timer
udt_send(sndpkt[base])
udt_send(sndpkt[base+1])
…
udt_send(sndpkt[nextseqnum-1])
```

```
rdt_rcv(rcvpkt)
  && corrupt(rcvpkt)
_______________
```

```
rdt_rcv(rcvpkt) &&
   notcorrupt(rcvpkt)
_______________
base = getacknum(rcvpkt)+1
If (base == nextseqnum)
   stop_timer
  else
   start_timer
```

窗口滑动



GBN: 接收方扩展FSM

```
default
_______________
udt_send(sndpkt)
```

```
Λ
_______________
expectedseqnum=1
sndpkt =
  make_pkt(expectedseqnum,ACK,chksum)
```

```
rdt_rcv(rcvpkt)
  && notcurrupt(rcvpkt)
  && hasseqnum(rcvpkt,expectedseqnum)
_______________
extract(rcvpkt,data)
deliver_data(data)
sndpkt = make_pkt(expectedseqnum,ACK,chksum)
udt_send(sndpkt)
expectedseqnum++
```

- ACK机制: 发送拥有最高序列号的、已被正确接收的分组的ACK
  - 可能产生重复ACK
  - 只需要记住唯一的**expectedseqnum**
- 乱序到达的分组：
  - 直接丢弃→接收方没有缓存
  - 重新确认序列号最大的、按序到达的分组

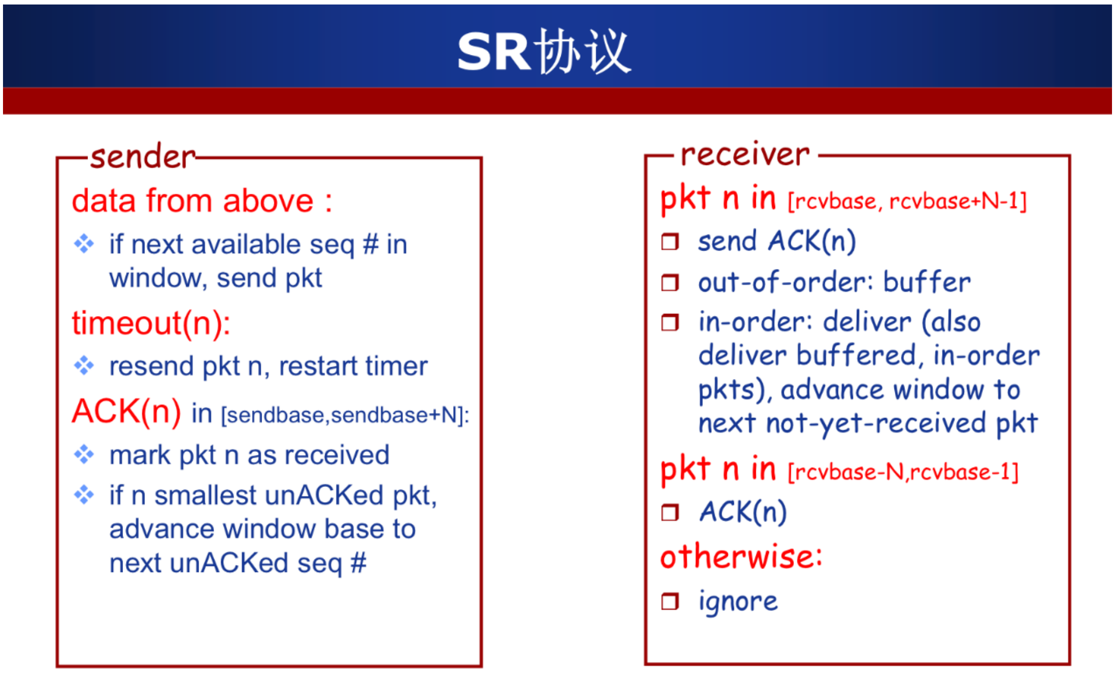
例：sender 传 0.1.2,3， 2 丢失.
则 receiver 丢弃 3，确认 1 为最后按序到达的分组

2) SR协议

在SR协议中，接收方设置缓冲区，且为每个报文段单独设置计时器，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，那么就只需要重发这一个报文段，之前收到的乱序报文段存在缓存中，在接收方确认整理排序之后就收到了之前的

所有报文段。在SR协议中，返回的ACK仅为当前接收成功的报文段序号。

SR协议中，发送方与接收方的任务如下：



SR协议

sender

data from above :

- if next available seq # in window, send pkt

timeout(n):

- resend pkt n, restart timer

ACK(n) in [sendbase,sendbase+N]:

- mark pkt n as received
- if n smallest unACKed pkt, advance window base to next unACKed seq #

receiver

pkt n in [rcvbase, rcvbase+N-1]

- ❐ send ACK(n)
- ❐ out-of-order: buffer
- ❐ in-order: deliver (also deliver buffered, in-order pkts), advance window to next not-yet-received pkt

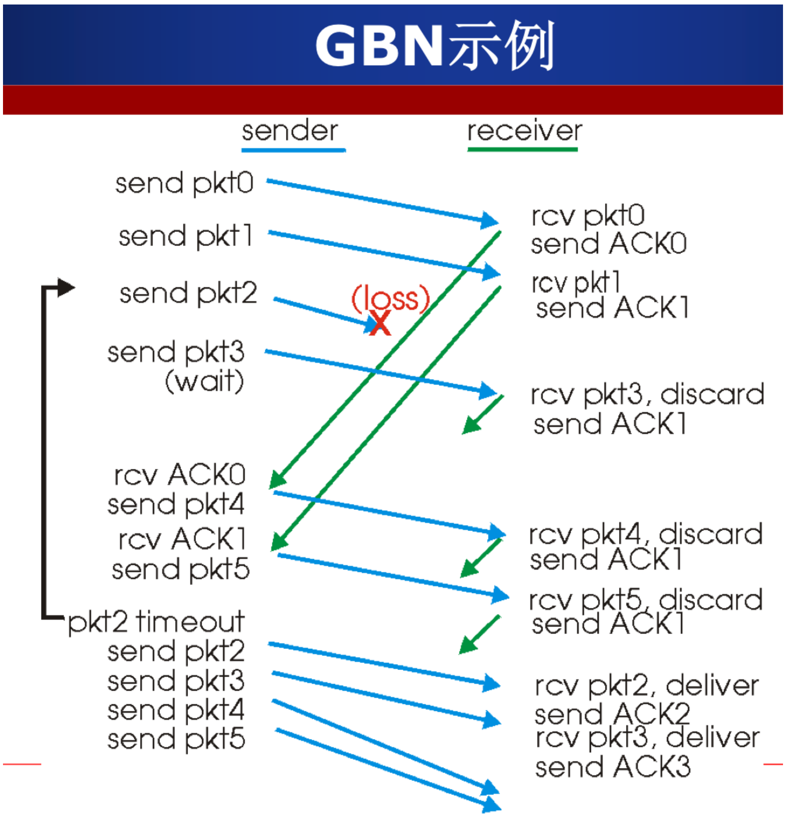
pkt n in [rcvbase-N,rcvbase-1]

- ❐ ACK(n)

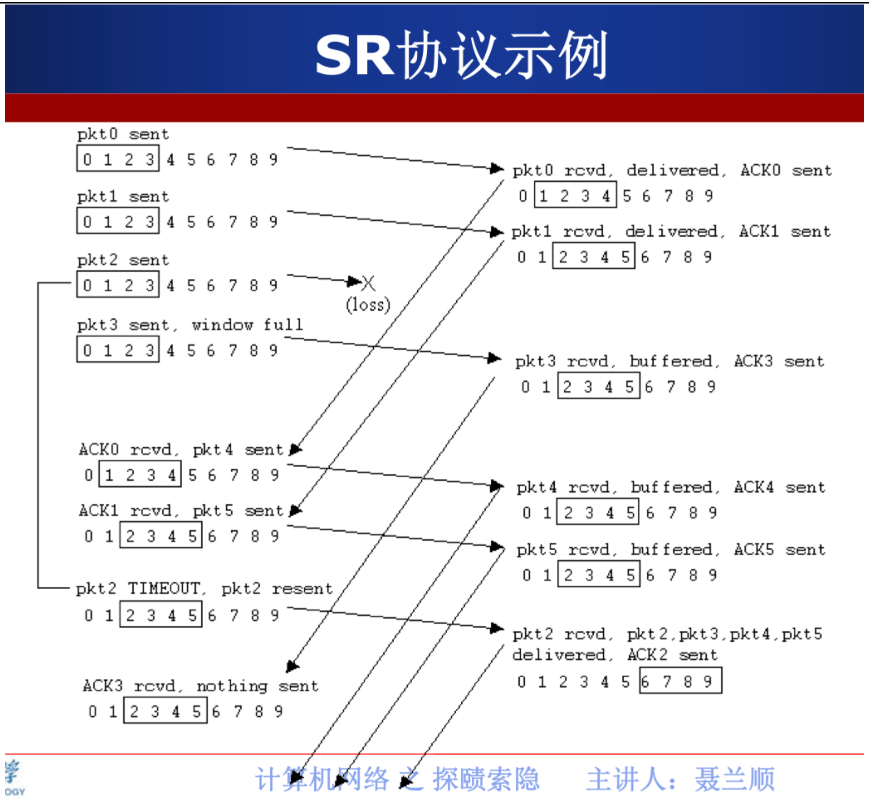
otherwise:

- ❐ ignore

3) GBN与SR协议的交互过程



GBN示例



SR协议示例

pkt0 sent
0 1 2 3 4 5 6 7 8 9

pkt0 rcvd, delivered, ACK0 sent
0 1 2 3 4 5 6 7 8 9

pkt1 sent
0 1 2 3 4 5 6 7 8 9

pkt1 rcvd, delivered, ACK1 sent
0 1 2 3 4 5 6 7 8 9

pkt2 sent
0 1 2 3 4 5 6 7 8 9

(loss)

pkt3 sent, window full
0 1 2 3 4 5 6 7 8 9

pkt3 rcvd, buffered, ACK3 sent
0 1 2 3 4 5 6 7 8 9

ACK0 rcvd, pkt4 sent
0 1 2 3 4 5 6 7 8 9

pkt4 rcvd, buffered, ACK4 sent
0 1 2 3 4 5 6 7 8 9

ACK1 rcvd, pkt5 sent
0 1 2 3 4 5 6 7 8 9

pkt5 rcvd, buffered, ACK5 sent
0 1 2 3 4 5 6 7 8 9

pkt2 TIMEOUT, pkt2 resent
0 1 2 3 4 5 6 7 8 9

pkt2 rcvd, pkt2,pkt3,pkt4,pkt5
delivered, ACK2 sent
0 1 2 3 4 5 6 7 8 9

ACK3 rcvd, nothing sent
0 1 2 3 4 5 6 7 8 9

计算机网络 之 探赜索隐　　主讲人：聂兰顺

2. 分析程序设计思路

1) GBN协议的设计思路

单向通信的GBN协议由客户端发送请求，服务器端收到确认后向客户端发送数据。

对于服务器端，在本实验中，使用 UDP 协议传输数据（在具体实现时，读取我们准备的测试文件进行传输），首先，服务器端等待客户端的请求，接收来自客户端的消息当服务器端接收到客户端发来的数据传输请求时，服务器端将数据分割成一个个数据报进行发送，并在服务器端的命令行界面中打印发送出的数据报相关信息。在发送完成后，客户端即启动计时器，开始等待客户端的 ACK 信息。当收到客户端回复的 ACK 时，服务器端确认 ACK 及其之前的数据报均已收到，服务器端的发送窗口可以滑动，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，则全部重传窗口内最大 ACK 之后的所有已发送的数据报。

对于客户端，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。在实际的程序中，必须模拟 ACK 丢失，然后由服务器端检测到超时重传的情况。

在实际的程序中，首先创建一个套接字，并将其绑定在指定的端口（12340）上。客户端解析命令行中的请求信息。根据不同的字符串，采取不同的应对策略。

当执行数据传输指令（-testgbn）时，客户端首先向服务器端发送请求信息，然后服务器端解析请求，进行握手，由服务器向客户端发送一个 205 状态码（示例程序中的定义），在客户端收到服务器端发来的 205 状态码后，回复一个 200 大小的状态码，表示客户端可以接收数据；在完成握手（服务器收到 200 状态码）之后，服务器端便可以读取本地文件，将其切分，封装到数据报中，通过 GBN 协议的规则（ACK采取累积确认机制，若丢失则重传之后的所有数据报等等）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。

在本实验中，需要在发送端（服务器端）设置分组丢失率和 ACK 丢失率（默认均设为0.2）。若要改造协议实现双向传输的功能，只需要在服务器端和客户端加上对方对应的部分即可。由“客户端”向“服务器端”发送数据的功能通过“-testgbn2”命令调用实现。

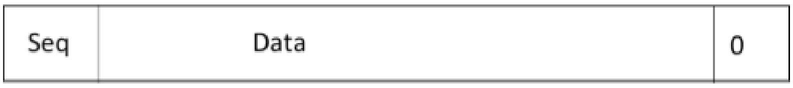
2) SR协议的设计思路

SR协议与GBN协议在握手阶段相同，都是根据 205 状态码和 200 状态码来判断是否能够正常建立连接。它们的不同之处主要是在数据重传方面。对于SR协议，当超时事件发生时，服务器端并不重传窗口内所有已发送的数据报，而是只发送窗口内那些没有收到 ACK 的数据报；对于客户端，其也有一个窗口。它接受服务端发来的数据报并返回确认信息 ACK，但并未采用累积确认机制。此时的客户端收到哪个序号的数据包就返回哪个序号的 ACK。即使前面有还没有收到的分组，也会将该组缓存下来，待全部收到按顺序的一组数据报时，将其交付给上层协议。

3. GBN与SR协议的数据格式

1) GBN协议与SR协议的数据报格式

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：



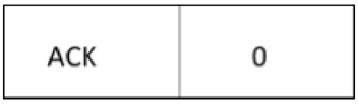
Seq	Data	0

Seq 为 1 个字节，取值为 0~255，（故序列号最多为 256 个）；

Data≤1024 个字节，为传输的数据；

最后一个字节放入 EOF0，表示结尾。

2) GBN协议与SR协议的 ACK 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；

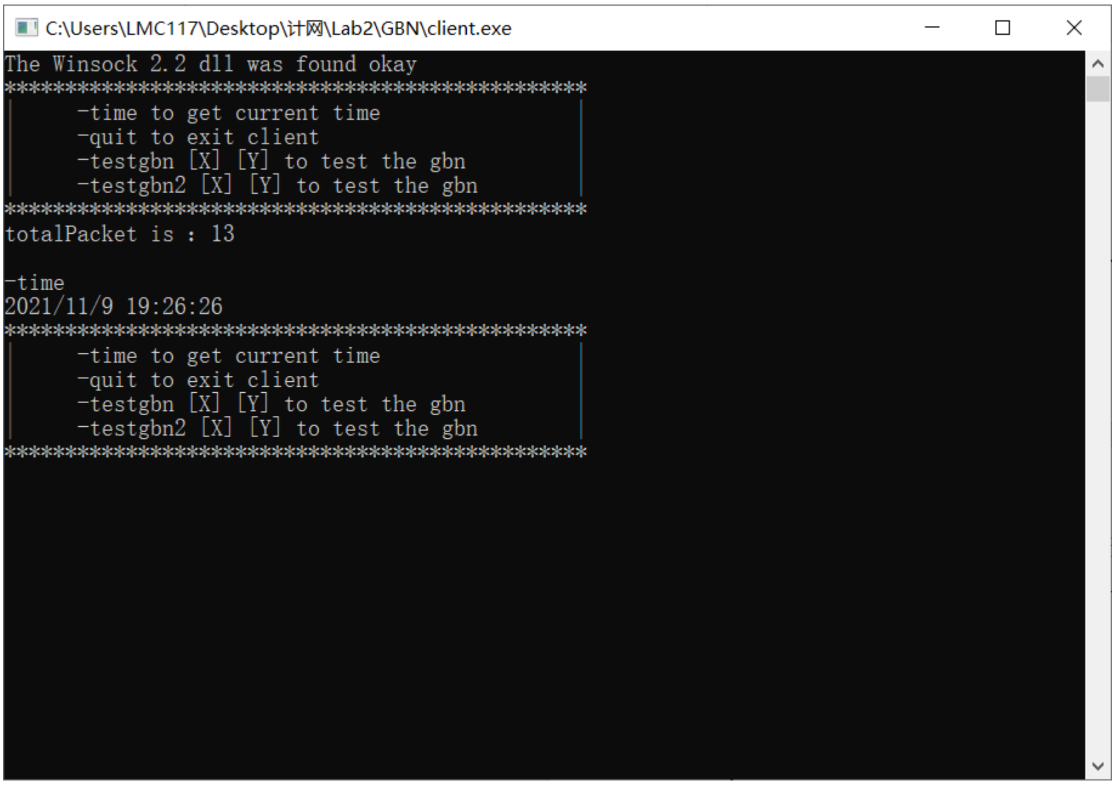
末尾放入 0，表示数据结束。

实验结果：

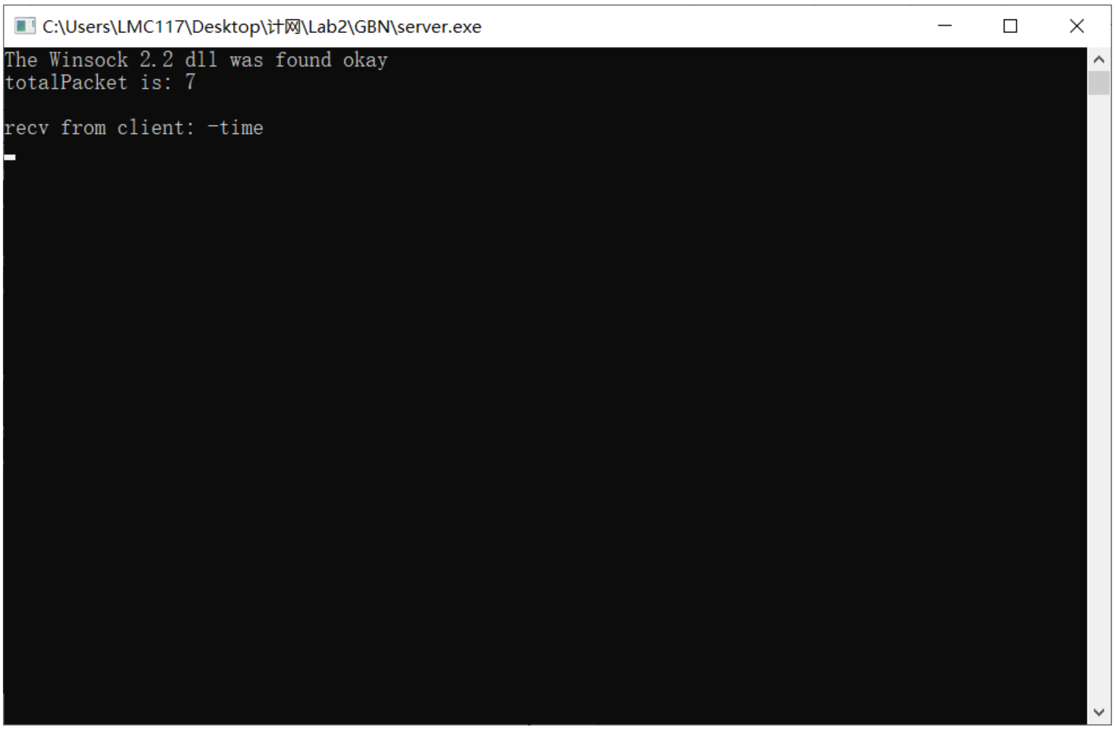
采用演示截图、文字说明等方式，给出本次实验的实验结果。

1. GBN单向传输（基础功能）

首先输入-time进行时间测试：



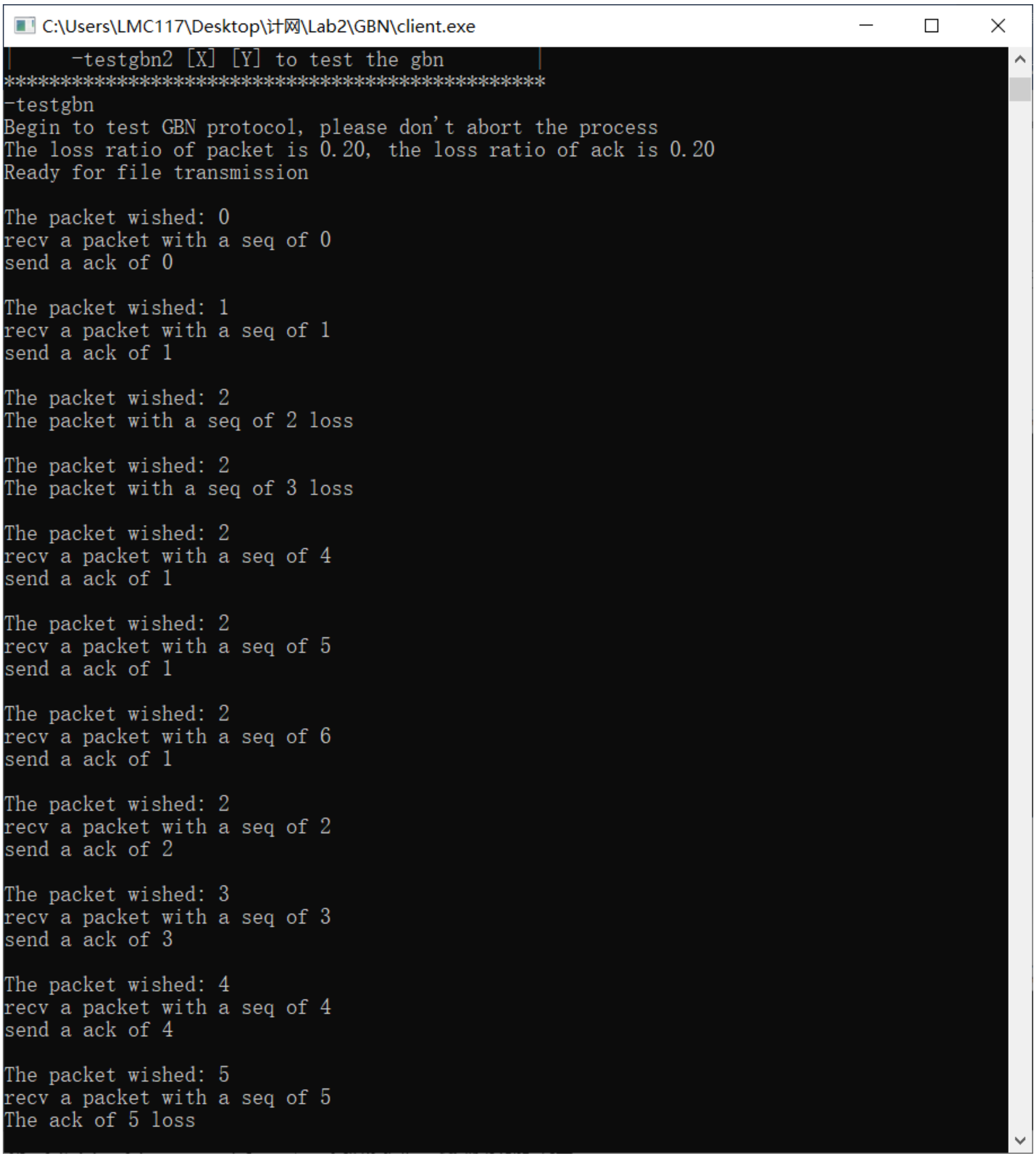
```
The Winsock 2.2 dll was found okay
************************************************
|     -time to get current time                |
|     -quit to exit client                     |
|     -testgbn [X] [Y] to test the gbn         |
|     -testgbn2 [X] [Y] to test the gbn        |
************************************************
totalPacket is : 13

-time
2021/11/9 19:26:26
************************************************
|     -time to get current time                |
|     -quit to exit client                     |
|     -testgbn [X] [Y] to test the gbn         |
|     -testgbn2 [X] [Y] to test the gbn        |
************************************************
```



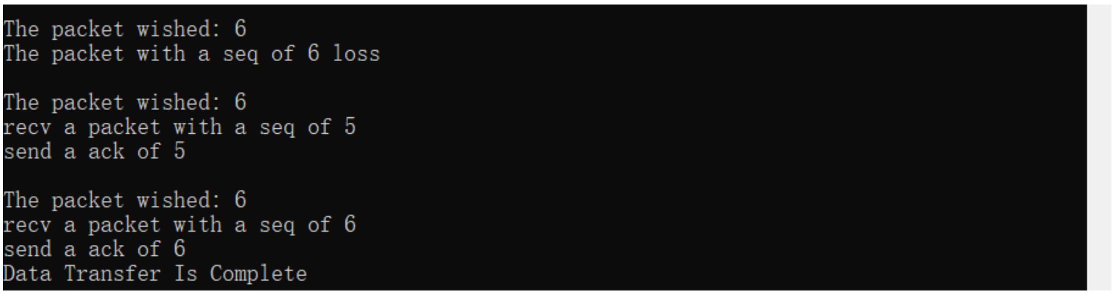
```
The Winsock 2.2 dll was found okay
totalPacket is: 7

recv from client: -time
```

然后输入-testgbn，进行单向传输的实验：

客户端：

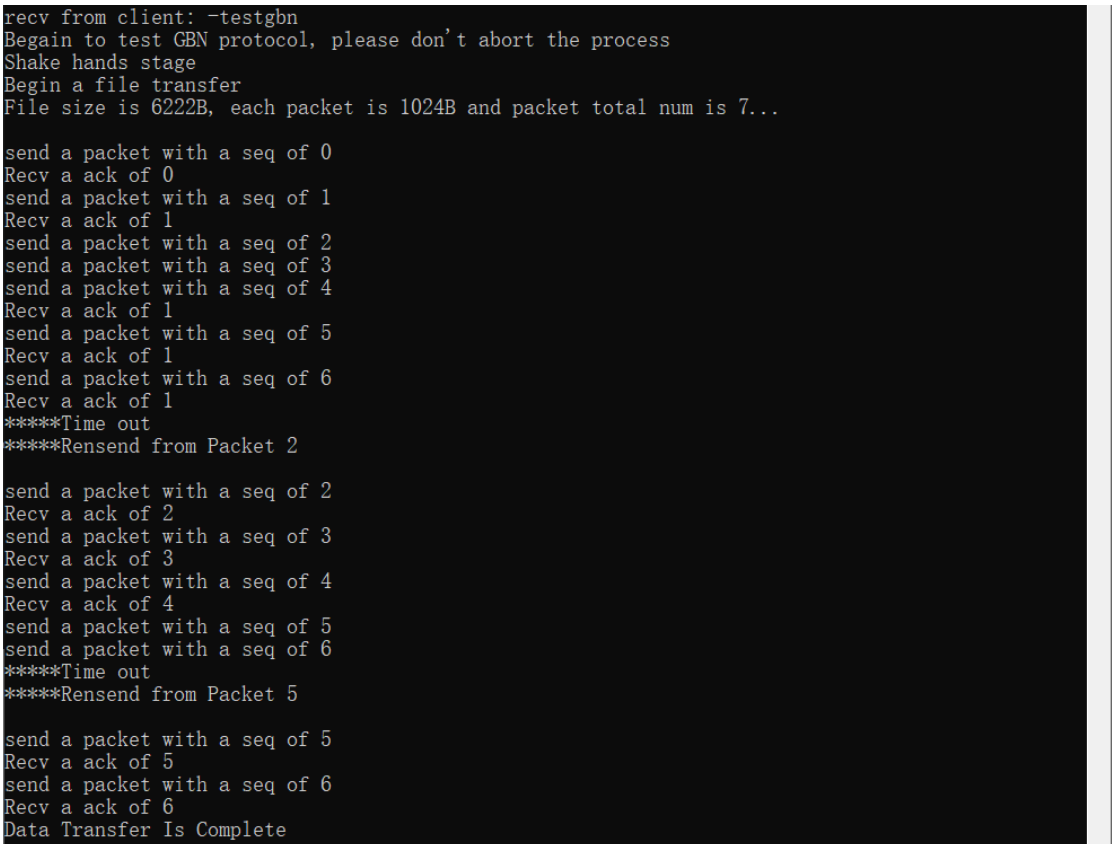


```
|     -testgbn2 [X] [Y] to test the gbn     |
************************************************
-testgbn
Begin to test GBN protocol, please don't abort the process
The loss ratio of packet is 0.20, the loss ratio of ack is 0.20
Ready for file transmission

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

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

The packet wished: 2
The packet with a seq of 2 loss

The packet wished: 2
The packet with a seq of 3 loss

The packet wished: 2
recv a packet with a seq of 4
send a ack of 1

The packet wished: 2
recv a packet with a seq of 5
send a ack of 1

The packet wished: 2
recv a packet with a seq of 6
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
The ack of 5 loss
```



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

The packet wished: 6
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
```

服务器端：



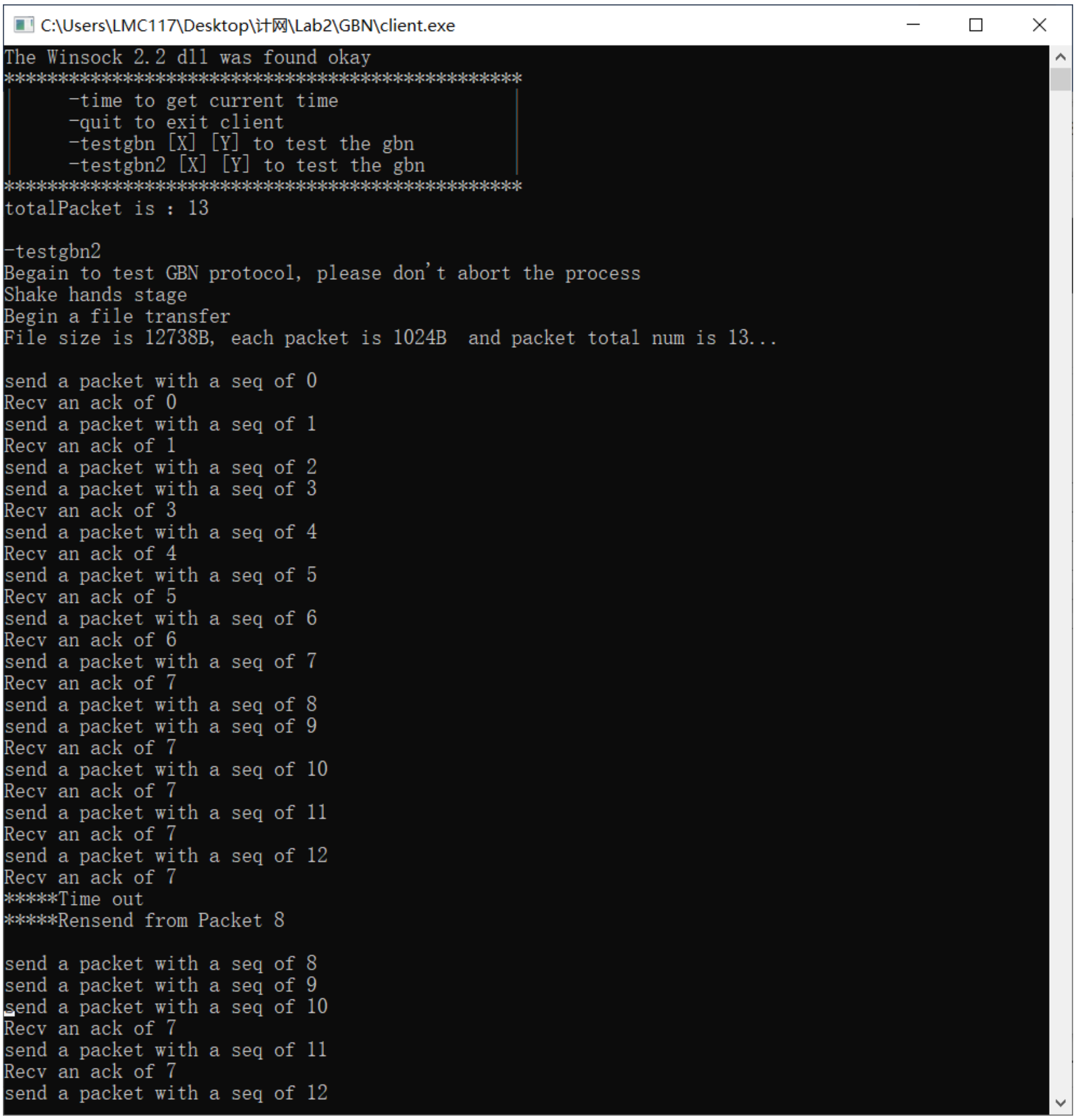
```
recv from client: -testgbn
Begain to test GBN protocol, please don't abort the process
Shake hands stage
Begin a file transfer
File size is 6222B, each packet is 1024B and packet total num is 7...

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
send a packet with a seq of 3
send a packet with a seq of 4
Recv a ack of 1
send a packet with a seq of 5
Recv a ack of 1
send a packet with a seq of 6
Recv a ack of 1
*****Time out
*****Rensend from Packet 2

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
send a packet with a seq of 6
*****Time out
*****Rensend from Packet 5

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
```

可以看出，第二个包在传输过程中丢失，由于服务器端迟迟没有收到 ACK2，在超时后，服务器便对第二个包进行了重传，(对第五个包的分析情况相同)，在完成全部数据报的传输后，打印提示字符，结束传输。

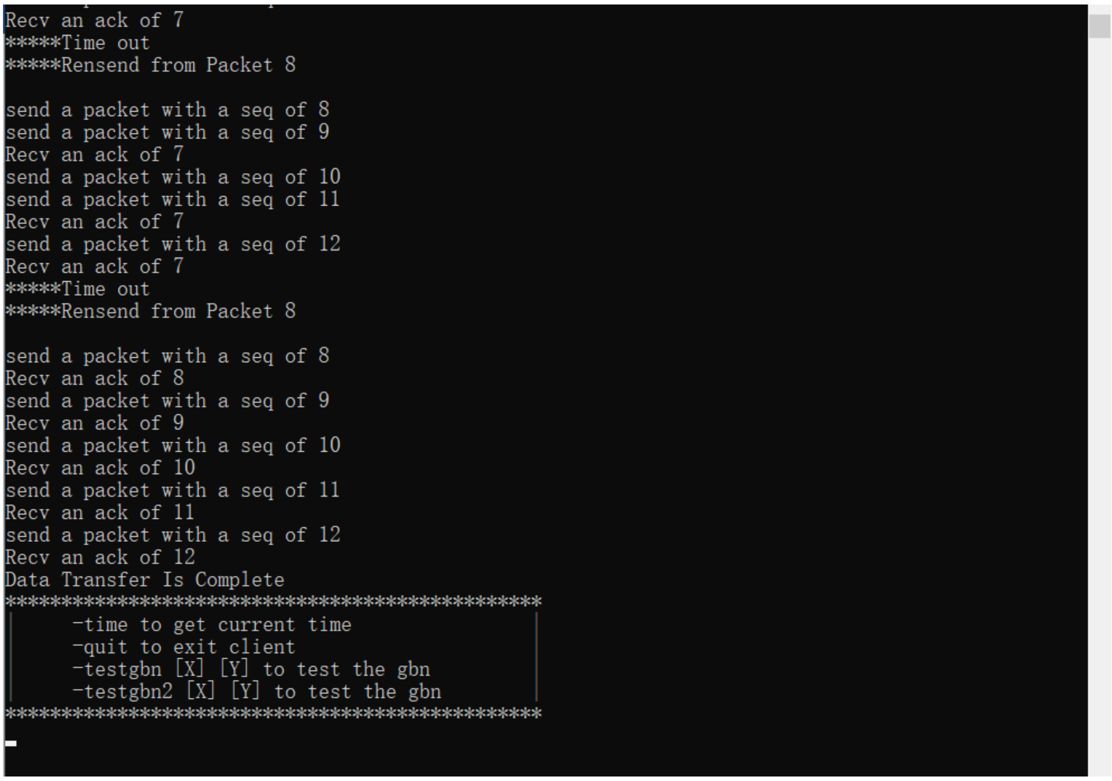
2. GBN双向数据传输功能

在命令行中输入“-testgbn2”，测试反向传输（由客户端向服务器发送数据）功能。结果如下：

客户端（此时发送数据，作“服务器”)：

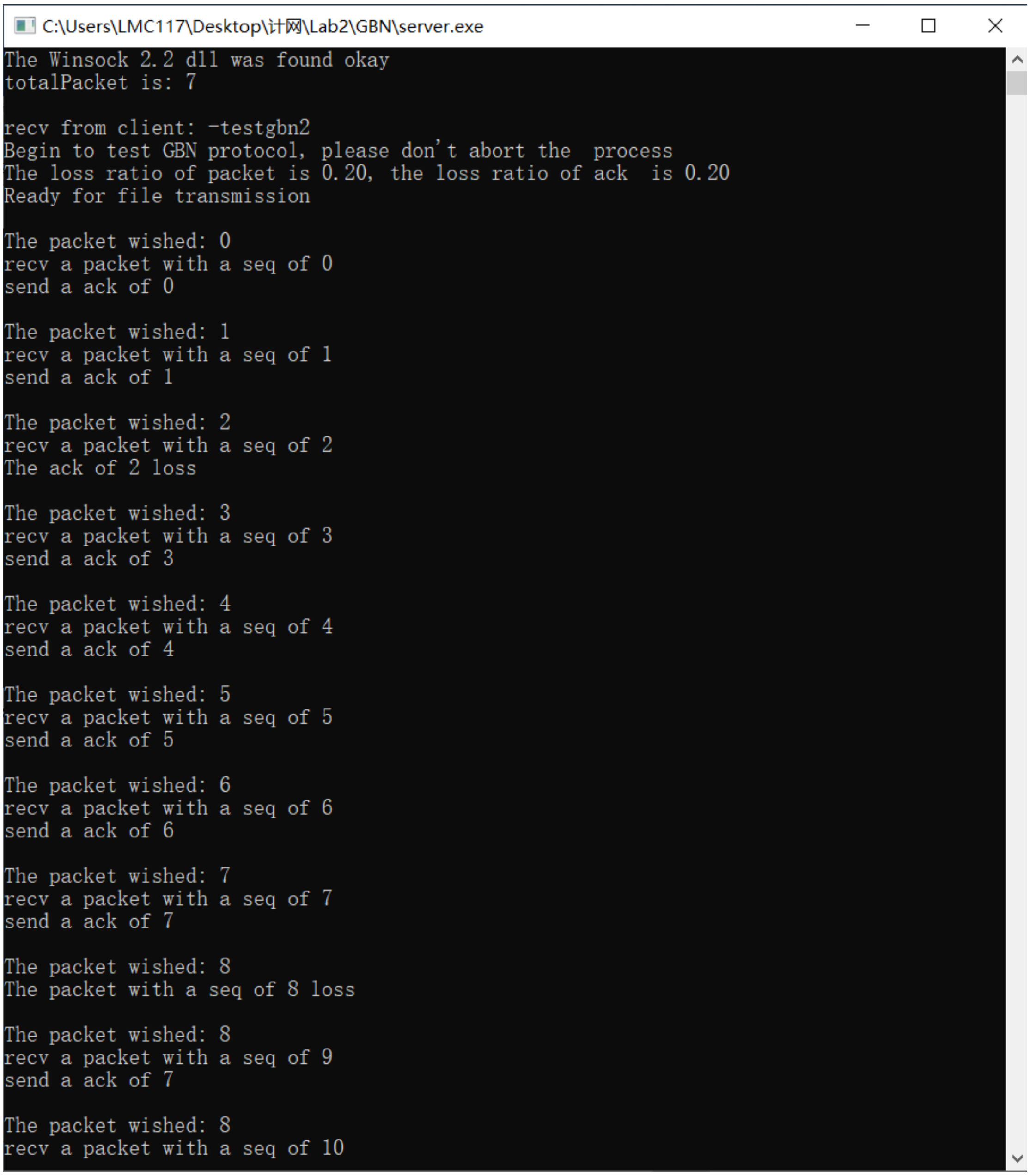


```
The Winsock 2.2 dll was found okay
************************************************
|     -time to get current time                |
|     -quit to exit client                     |
|     -testgbn [X] [Y] to test the gbn         |
|     -testgbn2 [X] [Y] to test the gbn        |
************************************************
totalPacket is : 13

-testgbn2
Begain to test GBN protocol, please don't abort the process
Shake hands stage
Begin a file transfer
File size is 12738B, each packet is 1024B  and packet total num is 13...

send a packet with a seq of 0
Recv an ack of 0
send a packet with a seq of 1
Recv an ack of 1
send a packet with a seq 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
Recv an ack of 6
send a packet with a seq of 7
Recv an ack of 7
send a packet with a seq of 8
send a packet with a seq of 9
Recv an ack of 7
send a packet with a seq of 10
Recv an ack of 7
send a packet with a seq of 11
Recv an ack of 7
send a packet with a seq of 12
Recv an ack of 7
*****Time out
*****Rensend from Packet 8

send a packet with a seq of 8
send a packet with a seq of 9
send a packet with a seq of 10
Recv an ack of 7
send a packet with a seq of 11
Recv an ack of 7
send a packet with a seq of 12
```

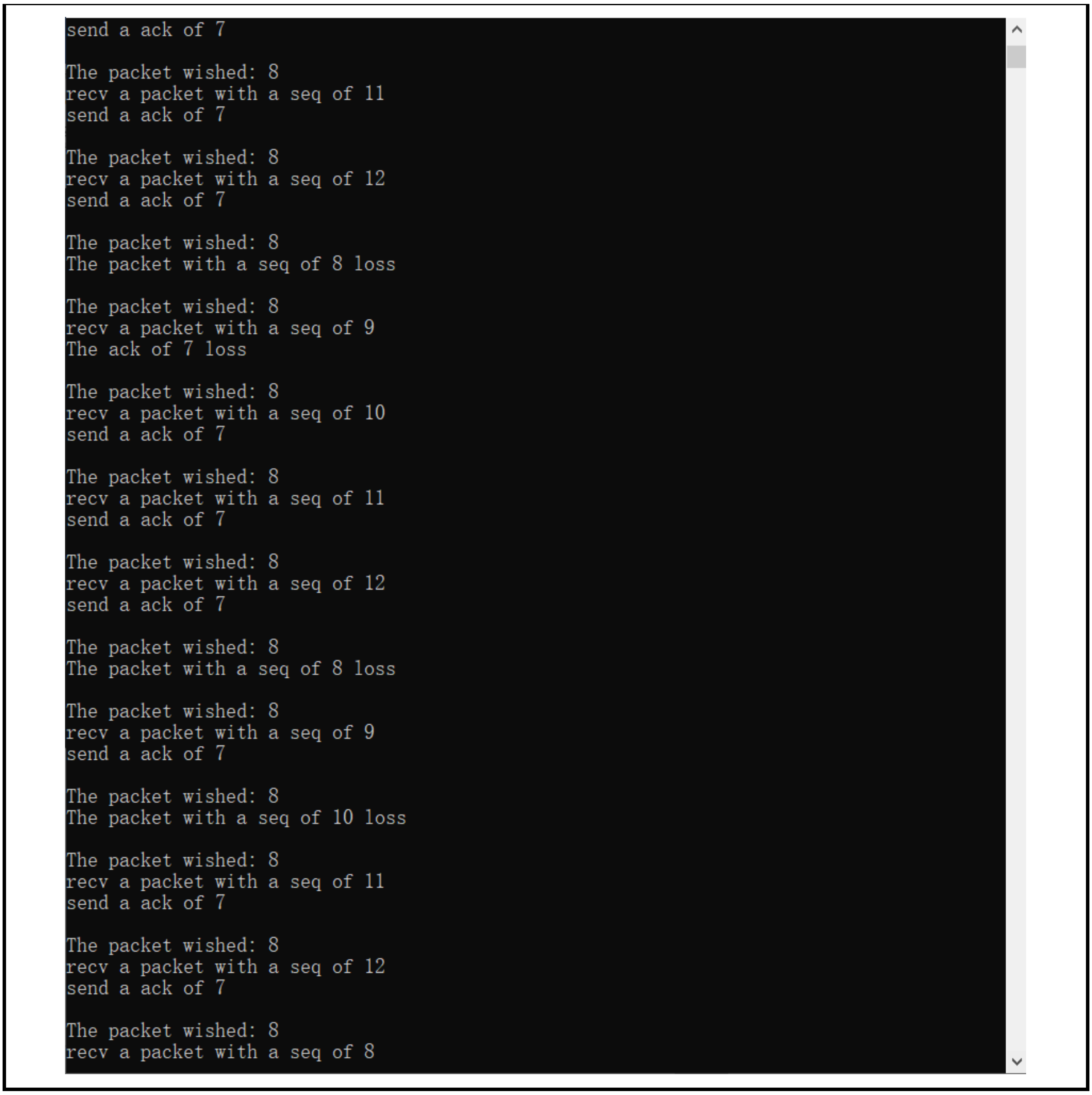


```
Recv an ack of 7
*****Time out
*****Rensend from Packet 8

send a packet with a seq of 8
send a packet with a seq of 9
Recv an ack of 7
send a packet with a seq of 10
send a packet with a seq of 11
Recv an ack of 7
send a packet with a seq of 12
Recv an ack of 7
*****Time out
*****Rensend from Packet 8

send a packet with a seq of 8
Recv an ack of 8
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
send a packet with a seq of 11
Recv an ack of 11
send a packet with a seq of 12
Recv an ack of 12
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        |
************************************************
```

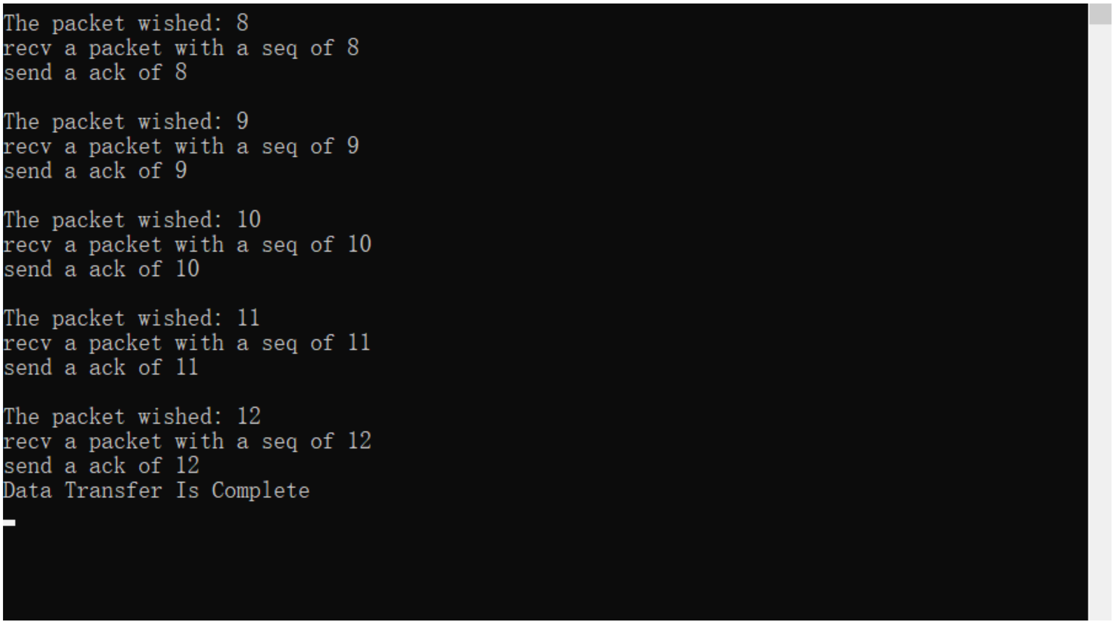
服务器端（此时接收数据，作“客户端”）：



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

recv from client: -testgbn2
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
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
The ack of 2 loss

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

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

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

The packet wished: 8
recv a packet with a seq of 9
send a ack of 7

The packet wished: 8
recv a packet with a seq of 10
```



```
send a ack of 7

The packet wished: 8
recv a packet with a seq of 11
send a ack of 7

The packet wished: 8
recv a packet with a seq of 12
send a ack of 7

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

The packet wished: 8
recv a packet with a seq of 9
The ack of 7 loss

The packet wished: 8
recv a packet with a seq of 10
send a ack of 7

The packet wished: 8
recv a packet with a seq of 11
send a ack of 7

The packet wished: 8
recv a packet with a seq of 12
send a ack of 7

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

The packet wished: 8
recv a packet with a seq of 9
send a ack of 7

The packet wished: 8
The packet with a seq of 10 loss

The packet wished: 8
recv a packet with a seq of 11
send a ack of 7

The packet wished: 8
recv a packet with a seq of 12
send a ack of 7

The packet wished: 8
recv a packet with a seq of 8
```



```
The packet wished: 8
recv a packet with a seq of 8
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

The packet wished: 11
recv a packet with a seq of 11
send a ack of 11

The packet wished: 12
recv a packet with a seq of 12
send a ack of 12
Data Transfer Is Complete
```

在此次传输过程中，数据报8反复丢失，于是按照GBN协议的规则进行超时后重传。

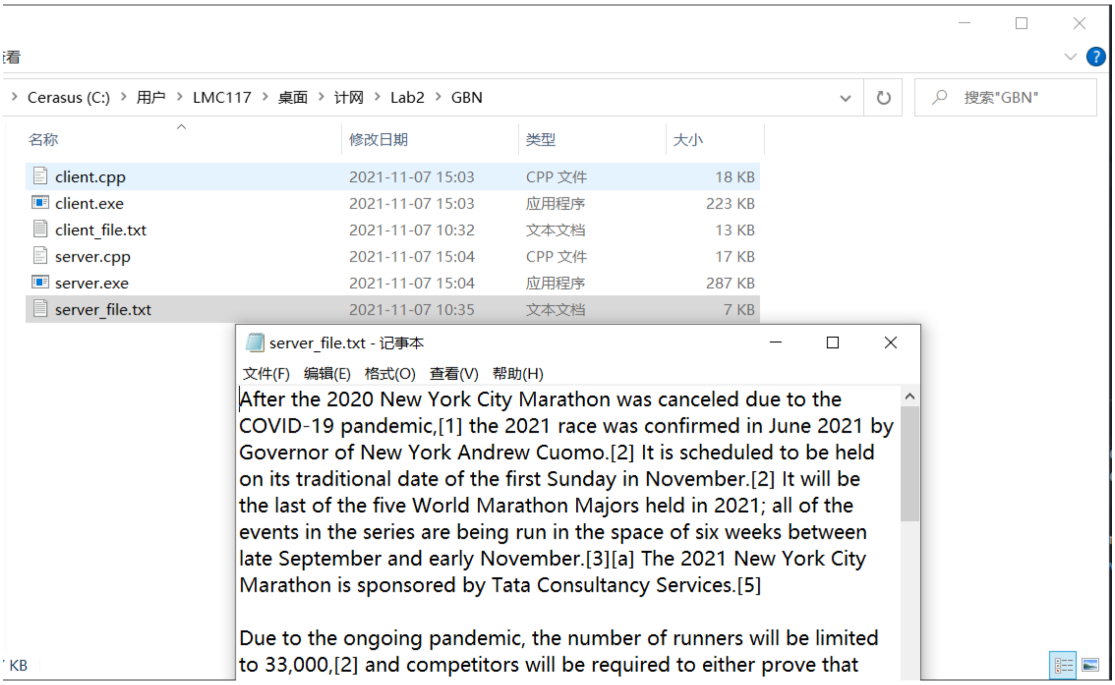
3. C/S架构文件传输功能

1) GBN协议的文件传输

在GBN协议中，我设计了一个比较简单的文件传输机制：在传输数据时，读取文件，并将其封装到数据报中，在命令行中为方便查看GBN传输的过程，仅在传输开始时显示数据报的数量以及总大小。所以，GBN协议传输成功，即代表着实现文件传输功能。

用于测试服务器端向客户端传递数据和客户端向服务器端传递数据的文件分别为server_file.txt与client_file.txt，其中内容均来自英文版维基百科。

由服务器端向客户端传输的文件如下：

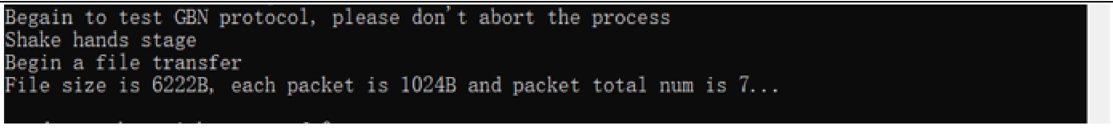

看
Cerasus (C:) › 用户 › LMC117 › 桌面 › 计网 › Lab2 › GBN
搜索"GBN"
KB
server_file.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

名称	修改日期	类型	大小
client.cpp	2021-11-07 15:03	CPP 文件	18 KB
client.exe	2021-11-07 15:03	应用程序	223 KB
client_file.txt	2021-11-07 10:32	文本文档	13 KB
server.cpp	2021-11-07 15:04	CPP 文件	17 KB
server.exe	2021-11-07 15:04	应用程序	287 KB
server_file.txt	2021-11-07 10:35	文本文档	7 KB

After the 2020 New York City Marathon was canceled due to the COVID-19 pandemic,[1] the 2021 race was confirmed in June 2021 by Governor of New York Andrew Cuomo.[2] It is scheduled to be held on its traditional date of the first Sunday in November.[2] It will be the last of the five World Marathon Majors held in 2021; all of the events in the series are being run in the space of six weeks between late September and early November.[3][a] The 2021 New York City Marathon is sponsored by Tata Consultancy Services.[5]

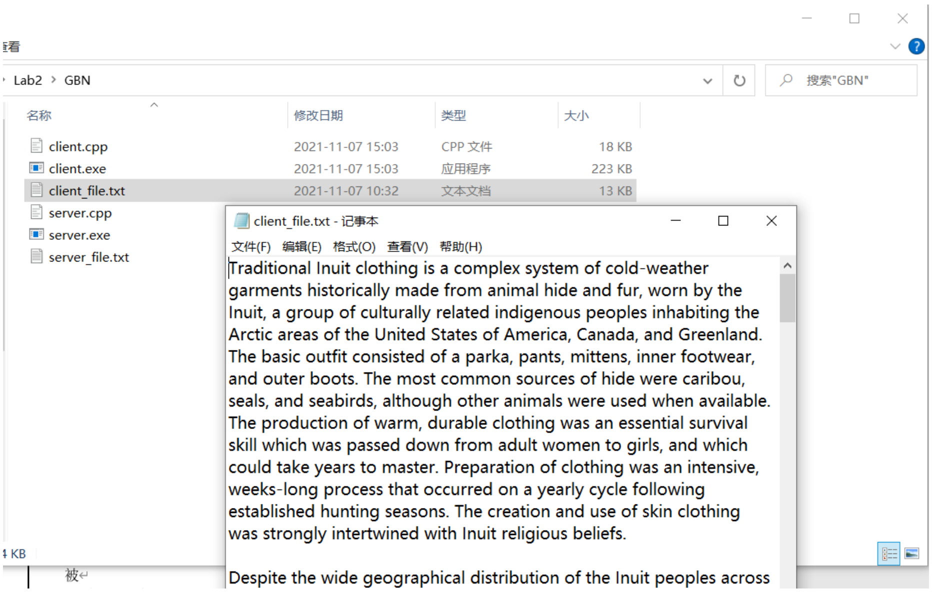
Due to the ongoing pandemic, the number of runners will be limited to 33,000,[2] and competitors will be required to either prove that

其大小为7KB，对应着传输时的7个数据报：



```
Begain to test GBN protocol, please don't abort the process
Shake hands stage
Begin a file transfer
File size is 6222B, each packet is 1024B and packet total num is 7...
```

由客户端向服务器端反向传输的文件如下：

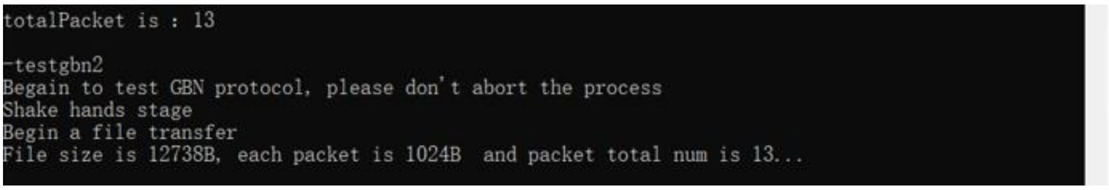


名称	修改日期	类型	大小
client.cpp	2021-11-07 15:03	CPP 文件	18 KB
client.exe	2021-11-07 15:03	应用程序	223 KB
client_file.txt	2021-11-07 10:32	文本文档	13 KB
server.cpp			
server.exe			
server_file.txt			

Traditional Inuit clothing is a complex system of cold-weather garments historically made from animal hide and fur, worn by the Inuit, a group of culturally related indigenous peoples inhabiting the Arctic areas of the United States of America, Canada, and Greenland. The basic outfit consisted of a parka, pants, mittens, inner footwear, and outer boots. The most common sources of hide were caribou, seals, and seabirds, although other animals were used when available. The production of warm, durable clothing was an essential survival skill which was passed down from adult women to girls, and which could take years to master. Preparation of clothing was an intensive, weeks-long process that occurred on a yearly cycle following established hunting seasons. The creation and use of skin clothing was strongly intertwined with Inuit religious beliefs.

Despite the wide geographical distribution of the Inuit peoples across

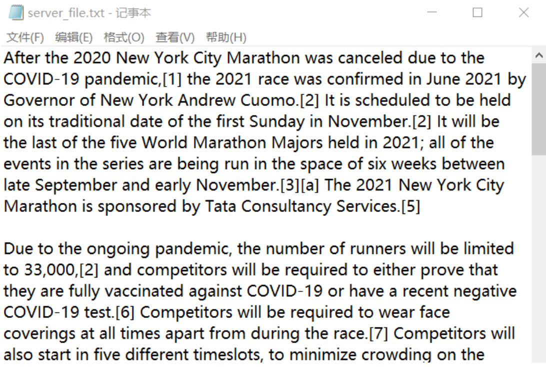
其大小为13KB，对应着传输时的13个数据报：



```
totalPacket is : 13

-testgbn2
Begain to test GBN protocol, please don't abort the process
Shake hands stage
Begin a file transfer
File size is 12738B, each packet is 1024B  and packet total num is 13...
```

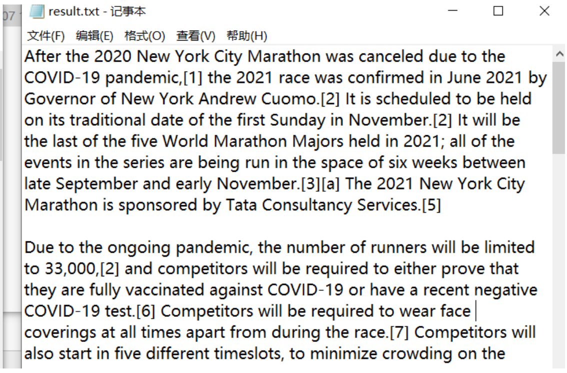
2) SR协议的文件传输

在SR协议中，我采用了另一种方式验证文件传输的正确性：由服务器端向客户端发送数据，客户端接收到数据后，将其保存在result.txt文件中。在传输完成后，查看生成的result.txt文件，发现内容与server_file.txt相同，故从另一种角度验证了文件传输的正确性。



After the 2020 New York City Marathon was canceled due to the COVID-19 pandemic,[1] the 2021 race was confirmed in June 2021 by Governor of New York Andrew Cuomo.[2] It is scheduled to be held on its traditional date of the first Sunday in November.[2] It will be the last of the five World Marathon Majors held in 2021; all of the events in the series are being run in the space of six weeks between late September and early November.[3][a] The 2021 New York City Marathon is sponsored by Tata Consultancy Services.[5]

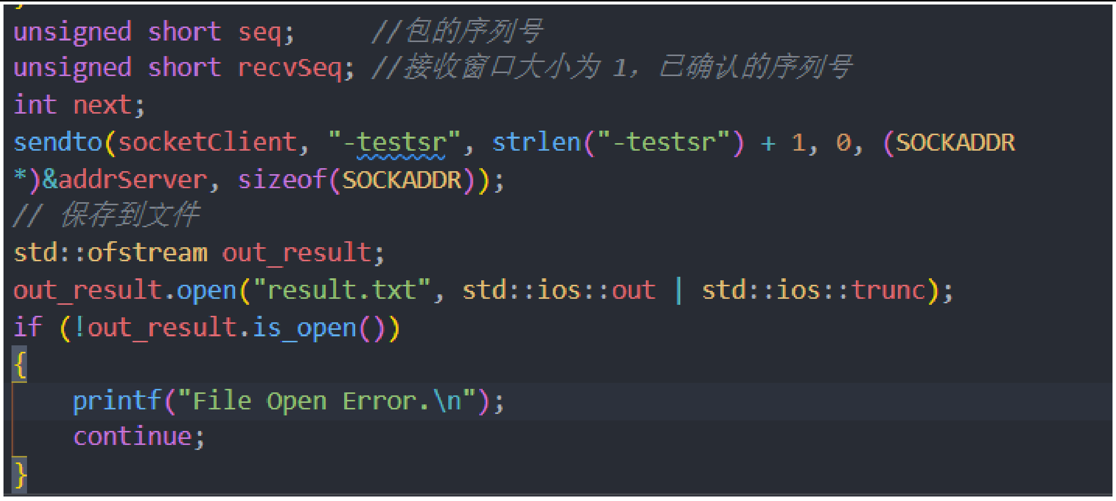
Due to the ongoing pandemic, the number of runners will be limited to 33,000,[2] and competitors will be required to either prove that they are fully vaccinated against COVID-19 or have a recent negative COVID-19 test.[6] Competitors will be required to wear face coverings at all times apart from during the race.[7] Competitors will also start in five different timeslots, to minimize crowding on the



After the 2020 New York City Marathon was canceled due to the COVID-19 pandemic,[1] the 2021 race was confirmed in June 2021 by Governor of New York Andrew Cuomo.[2] It is scheduled to be held on its traditional date of the first Sunday in November.[2] It will be the last of the five World Marathon Majors held in 2021; all of the events in the series are being run in the space of six weeks between late September and early November.[3][a] The 2021 New York City Marathon is sponsored by Tata Consultancy Services.[5]

Due to the ongoing pandemic, the number of runners will be limited to 33,000,[2] and competitors will be required to either prove that they are fully vaccinated against COVID-19 or have a recent negative COVID-19 test.[6] Competitors will be required to wear face coverings at all times apart from during the race.[7] Competitors will also start in five different timeslots, to minimize crowding on the

相关部分代码如下：

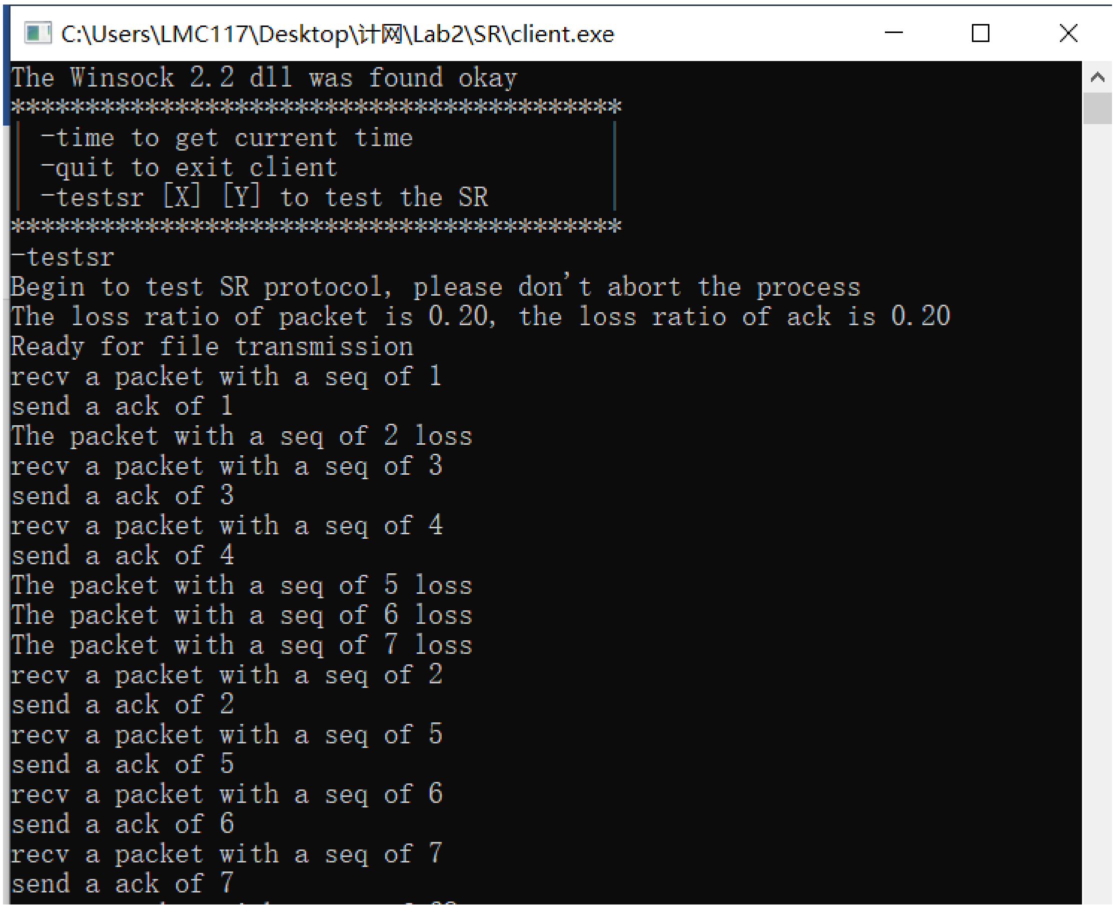


```
unsigned short seq;     //包的序列号
unsigned short recvSeq; //接收窗口大小为 1，已确认的序列号
int next;
sendto(socketClient, "-testsr", strlen("-testsr") + 1, 0, (SOCKADDR
*)&addrServer, sizeof(SOCKADDR));
// 保存到文件
std::ofstream out_result;
out_result.open("result.txt", std::ios::out | std::ios::trunc);
if (!out_result.is_open())
{
    printf("File Open Error.\n");
    continue;
}
```

4. GBN改进为SR协议

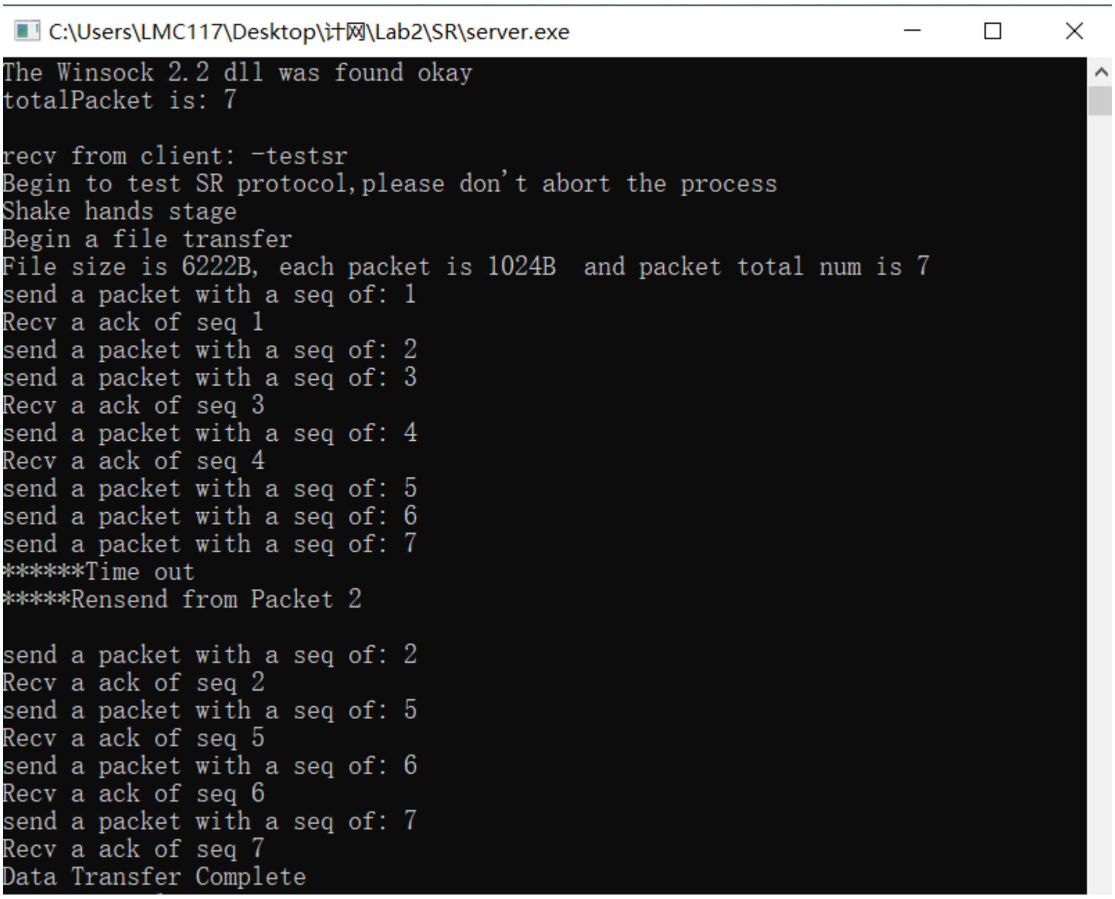
改进GBN协议为SR协议，结果如下：

客户端：



```
The Winsock 2.2 dll was found okay
*****************************************
|  -time to get current time            |
|  -quit to exit client                 |
|  -testsr [X] [Y] to test the SR       |
*****************************************
-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
send a ack of 1
The packet with a seq of 2 loss
recv a packet with a seq of 3
send a ack of 3
recv a packet with a seq of 4
send a ack of 4
The packet with a seq of 5 loss
The packet with a seq of 6 loss
The packet with a seq of 7 loss
recv a packet with a seq of 2
send a ack of 2
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
recv a packet with a seq of 7
send a ack of 7
```

服务器端：



```
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 6222B, each packet is 1024B  and packet total num is 7
send a packet with a seq of: 1
Recv a ack of seq 1
send a packet with a seq of: 2
send a packet with a seq of: 3
Recv a ack of seq 3
send a packet with a seq of: 4
Recv a ack of seq 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
******Time out
*****Rensend from Packet 2

send a packet with a seq of: 2
Recv a ack of seq 2
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
send a packet with a seq of: 7
Recv a ack of seq 7
Data Transfer Complete
```

可以看出，在数据报2丢失的情况下，服务器端超时仍未收到ACK2，便将数据报2重发一次。与GBN协议不同的是，SR协议中只重发了数据报2，并未重发序号2之后的所有数据报。这也说明我们设计的SR协议的合理性。

心得体会：

结合实验过程和结果给出实验的体会和收获。

- 对UDP协议的相关特性有了更为深刻的了解
- 掌握了停等协议、滑动窗口协议的协议内容，学习了全双工传输模式的实现技巧。
- 更加熟悉Socket编程的相关方法

实验源码：

见附件