



**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-停等协议的设计与实现， 可靠数据传输协议-GBN 协议的设计与实现					
姓名	田雪洋		院系	计算机科学与技术学院		
班级	1937102		学号	1190202110		
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	2021.10.30		
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:
<p>实验 2: 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。</p> <p>实验 3: 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。</p>
实验内容:
<p>概述本次实验的主要内容, 包含的实验项等。</p> <p><b>实验2:</b></p> <ol style="list-style-type: none"> <li>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, 可以不进行差错检测, 可以利用UDP协议差错检测</li> <li>2) 模拟引入数据包的丢失, 验证所设计协议的有效性。</li> <li>3) 改进所设计的停等协议, 支持双向数据传输;</li> <li>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。</li> <li>5) 计时器实现方法: 对于阻塞的socket 可用int setsockopt(int socket,int level, int option_name, const void* option_value, size_t option_len)函数设置套接字发送与接收超时时间; 对于非阻塞socket 可以使用累加sleep时间的方法判断socket 接受数据是否超时(当时间累加量超过一定数值时则认为套接字接受数据超时)。</li> </ol> <p><b>实验3:</b></p> <ol style="list-style-type: none"> <li>1) 基于 UDP 实现的 GBN 协议, 可以不进行差错检测, 可以利用 《计算机网络》实验指导书 20 UDP 协议差错检测;</li> <li>2) 自行设计数据帧的格式, 应至少包含序列号 Seq 和数据两部分;</li> <li>3) 自定义发送端序列号 Seq 比特数 L 以及发送窗口大小 W, 应满足条件 <math>W+1 \leq 2L</math>。</li> <li>4) 一种简单的服务器端计时器的实现办法: 设置套接字为非阻塞方式, 则服务器端在 recvfrom 方法上不会阻塞, 若正确接收到 ACK 消息, 则计时器清零, 若从客户端接收数据长度为-1 (表示没有接收到任何数据), 则计时器+1, 对计时器进行判断, 若其超过阈值, 则判断为超时, 进行超时重传。</li> <li>5) 为了模拟 ACK 丢失, 一种简单的实现办法: 客户端对接收的数据帧进行计数, 然后对总数进行模 N 运算, 若规定求模运算结果为零则 返回 ACK, 则每接收 N 个数据帧才返回 1 个 ACK。当 N 取值大于服务器端的超时阈值时, 则会出现服务器端超时现象。</li> <li>6) 当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。</li> </ol>
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。

由于实验2的停等协议是实验3停等协议的特例，所以下面仅叙述实验3的实验过程

(1) GBN协议数据分组格式，确认分组格式，各个域的作用

序号	数据
----	----

本次实验采取的 GBN 协议的数据分组格式和确认分组格式相同，都采用了序号+数据的分组格式，其中

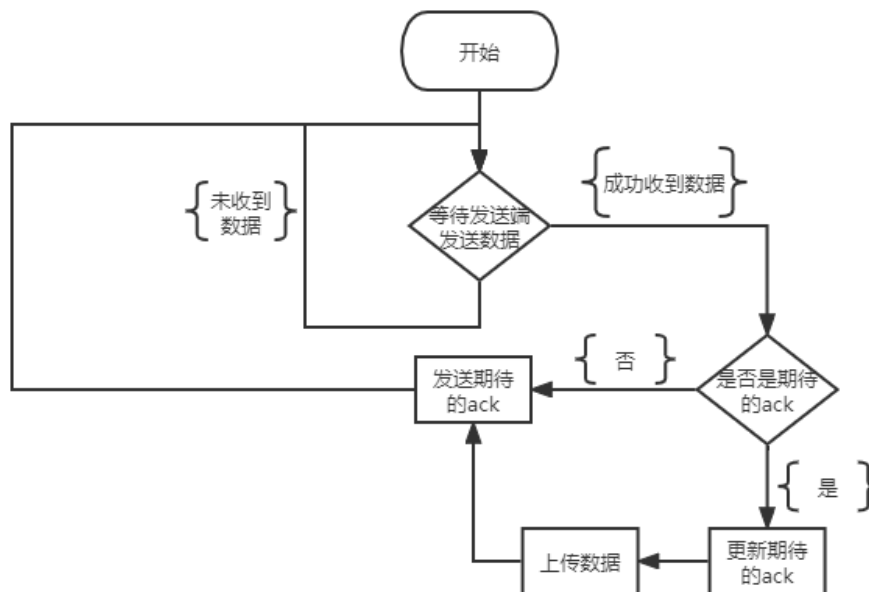
序号表示该数据分组是第几个分组，字段长度为 8 个字节。

数据表示该次传输的数据，字段长度为 0-1024 字节，即最大能够传输 1024 字节。而在确认分组中，数据域的字段长度为 0。

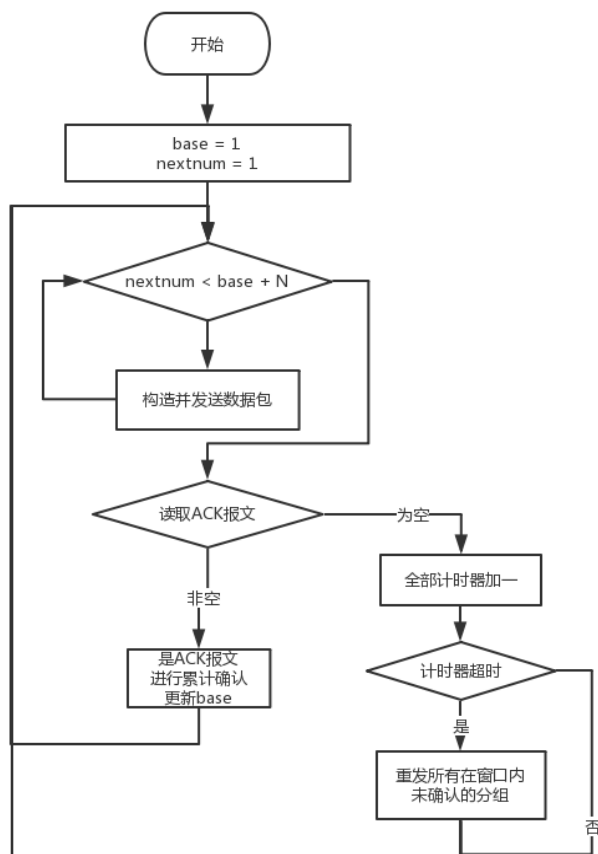
SR 和 GBN 采用相同的数据分组和确认分组，各个域的作用相同。

(2) 协议两端程序流程图

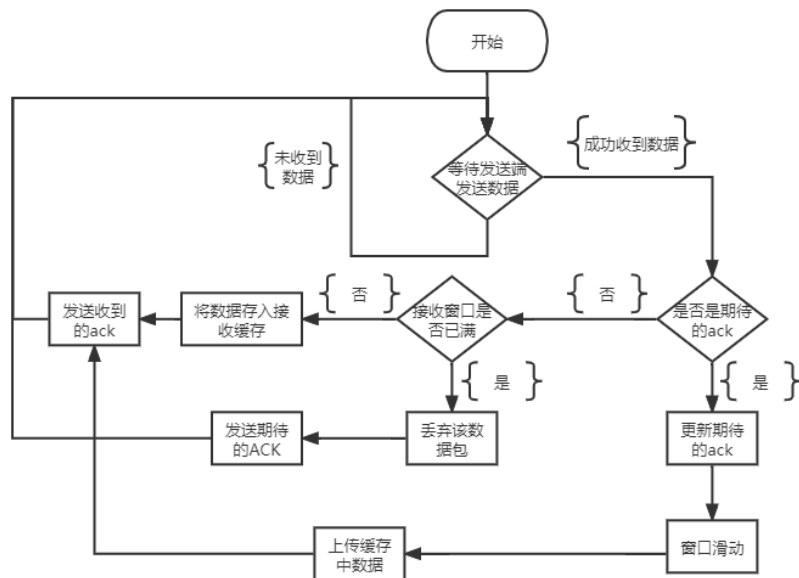
GBN 协议接收端：



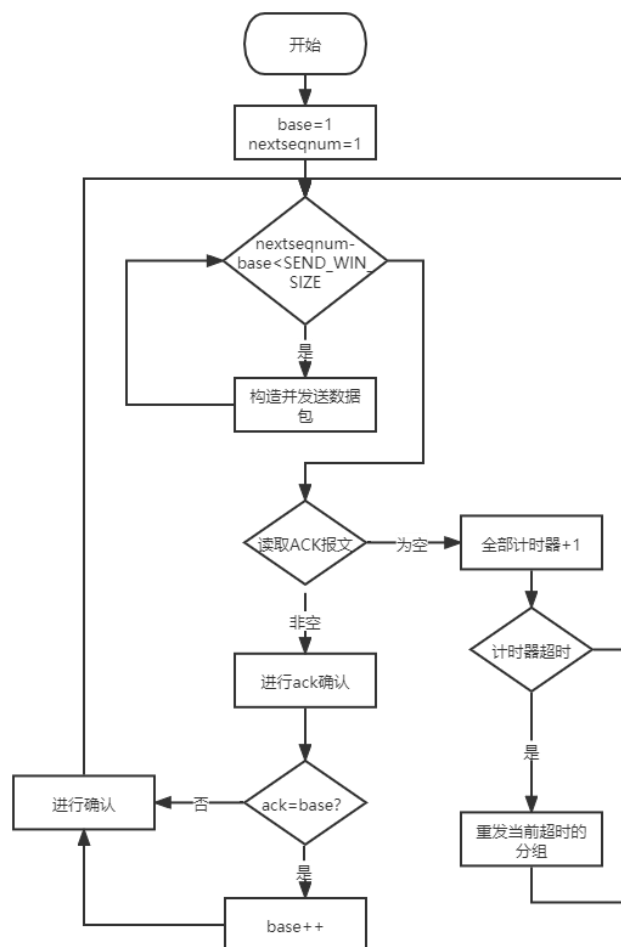
GBN 协议发送端：



SR 协议接收端：



SR 协议发送端：



### (3) 数据分组丢失验证模拟方法

本次实验中采取的模拟数据包丢失的方法,是在接收端中,对于接收到的数据帧,以一定的概率发送ACK报文(在实验中使用的概率为20%), 剩余的情况接收端不发送ACK即表现为ACK报文丢失。

通过这种方法即实验了数据分组的丢失模拟。

### (4) 协议典型交互过程

#### GBN协议:

在发送方端,发送端把需要发送的分组加入发送窗口,然后向接收端发送窗口内的分组。对于接收端来说,若接收到了期望收到的分组,则向发送方返回一个该分组的ACK,否则发送上一个正确接收的分组的ACK。对于发送端来说,发送端收到ACK分组后会将waitSeq移动到收到的ACK序列号+1处,然后进入下一轮循环。这样不断循环便能完成所有的文件发送。

#### SR协议:

SR协议与GBN协议的不同之处在于拥有一个接收窗口,因此需要为每个分组都设置一个计时器,当经过一次循环后,将重发发送窗口中超时的那些分组。对于接收方来说,如果收到乱序但序列号在接收窗口内的分组,则接收方会把它缓存,然后向发送方返回该分组的ACK。当收到按序到达的分组时,接收方会检查接收窗口内是否有能合并的已接收的分组,当这些分组能够从接收窗口的最左端合并时,先将这些分组合并,然后一起提交给上层协议,并滑动接

收窗口到未提交给上层协议的位置处。这样不断循环便能完成所有的文件的发送。

#### (5) 程序实现的主要类（或函数）及其主要作用

由于 SR 是在 GBN 基础上改进而来的，因此，他们主体结构类似，下面对于 SR 不同于 GBN 的设计，将以红字标出

1.

```
bool seqIsAvailable() {
    int step;
    step = nextseqnum - base;
    step = step >= 0 ? step : step + SEQ_SIZE;
    //序列号是否在当前发送窗口之内
    if (step >= SEND_WIND_SIZE) {
        return false;
    }
    //序列号未被确认
    if (ack[nextseqnum]) {
        return true;
    }
    return false;
}
```

该函数用于判断发送端是否需要进行发送数据包，当序列号在发送窗口中，并且该序列号未被确认，这时，可以进行数据发送。

```
bool seqIsAvailable() {
    int step;
    step = nextseqnum - base;
    step = step >= 0 ? step : step + SEQ_SIZE;
    //序列号是否在当前发送窗口之内
    if (step >= SEND_WIND_SIZE) {
        return false;
    }
    //序列号未被使用或重传
    if (ack[nextseqnum] == -1 || ack[nextseqnum] == 1) {
        return true;
    }
    return false;
}
```

SR 除了未确认时可以进行数据发送，若 ack 数组被标记为重传，也需要进行数据发送

2.

```

void ackHandler(char c) {
    unsigned char index = (unsigned char)c - 1; //序列号减一
    printf("收到的ack是: %d\n", index);
    if (base <= index) {
        for (int i = base; i <= index; ++i) {
            ack[i] = TRUE;
        }
        base = (index + 1) % SEQ_SIZE;
    }
    else {
        //ack 超过了最大值, 回到了 base 的左边
        for (int i = base; i < SEQ_SIZE; ++i) {
            ack[i] = TRUE;
        }
        for (int i = 0; i <= index; ++i) {
            ack[i] = TRUE;
        }
        base = index + 1;
    }
}

```

该函数是用于对收到的 ack 进行解析, 若收到的 ack 为想要收到的 ack, 则滑动窗口右移, 并把相应位的 ack 标记为已确认。

```

void ackHandler(char c) {
    unsigned char index = (unsigned char)c - 1; //序列号减一
    printf("收到的ack是: %d\n", index);
    //若base收到确认, base++
    if (base == index)
    {
        ack[base] = -1;
        base = (base + 1) % SEQ_SIZE;
    }
    else
    {
        ack[index] = -1;
    }
}

```

SR 协议, 若收到的 ack 为想要收到的 ack, 则滑动窗口右移, 并把相应位的 ack 标记为已确认。若收到的 ack 不是想要的, 同样进行确认, 但滑动窗口不移动

```
void getCurTime(string& curtime) {  
    char buffer[128];  
    memset(buffer, 0, sizeof(buffer));  
    time_t c_time;  
    struct tm* p;  
    time(&c_time);  
    p = localtime(&c_time);  
    //将数据格式化, 输出到字符串  
    sprintf_s(buffer, "%d/%d/%d %d:%d:%d",  
        p->tm_year + 1900,  
        p->tm_mon,  
        p->tm_mday,  
        p->tm_hour,  
        p->tm_min,  
        p->tm_sec);  
    buffer[strlen(buffer)] = 0;  
    curtime = string(buffer);  
}
```

该函数的作用是从系统获取时间

4.

```
std::ifstream file;  
file.open("server_send.txt");  
file.seekg(0, ios::end);  
int fileSize = (int)file.tellg();  
file.seekg(0, ios::beg);  
file.read(sendBuffer, fileSize);  
totalPacket = sizeof(sendBuffer) / BUFFER_LENGTH;  
printf("发送文件已加载完成, 文件包含%d个字节, 开始发送\n", fileSize);  
int sendSum = fileSize / 20 + (fileSize % 20 != 0);  
file.close();
```

该部分的作用是从文件中读入要发送的文件

5.



```

if (recvSize < 0) {
    timecount++;
    //20 次等待 ack 则超时重传
    if (timecount > 20)
    {
        printf("超时事件发生. \n");
        int index;
        //从发生超时的开始, 所以ack全部变为可用
        for (int i = 0; i < SEND_WIND_SIZE; ++i) {
            index = (i + base) % SEQ_SIZE;
            ack[index] = TRUE;
        }
        //总确认数减少
        totalSeq -= SEND_WIND_SIZE;
        nextseqnum = base;
        timecount = 0;
    }
}

```

这部分的内容是超时处理, 若发生超时, 从 base 开始, 之后的所有 ack 都变为可用, 并将确认总数减少。

```

int count = 0;
for (int i = 0; i < SEND_WIND_SIZE; ++i)
{
    if (ack[(i+base)%SEQ_SIZE] == 0)
    {
        clocktimer[(i + base) % SEQ_SIZE] += 1;
        if (clocktimer[(i + base) % SEQ_SIZE] > 20)
        {
            cout << "超时错误! \n";
            ack[(i + base) % SEQ_SIZE] = 1;
            totalSeq--;
            clocktimer[(i + base) % SEQ_SIZE] = 0;
        }
    }
}

nextseqnum = base;

```

SR 协议把原来的单个变量 timeout 计时改为使用 clocktimer 数组进行计时, 当该 ack 未被确认时, 遍历 clocktimer 数组, 对所有 ack 未被确认的时间+1, 若发生超时错误, 把接受到的 totalSeq 减少相应数量, 并把 nextseqnum 重新移到 base 处。

```

BOOL lossInLossRatio(float lossRatio) {
    int lossBound = (int)(lossRatio * 100);
    int r = rand() % 101;
    if (r <= lossBound) {
        return TRUE;
    }
    return FALSE;
}

```

该函数的作用是根据输入的丢失率，随机生成一个数字，判断是否丢失，丢失则返回 TRUE

## 6.全双工处理

对于全双工，只需要使客户端和服务端都可以互相向对方发送数据，两边的实现是对称的。即发送端和接收端的身份互换即可。

### (6) 源代码

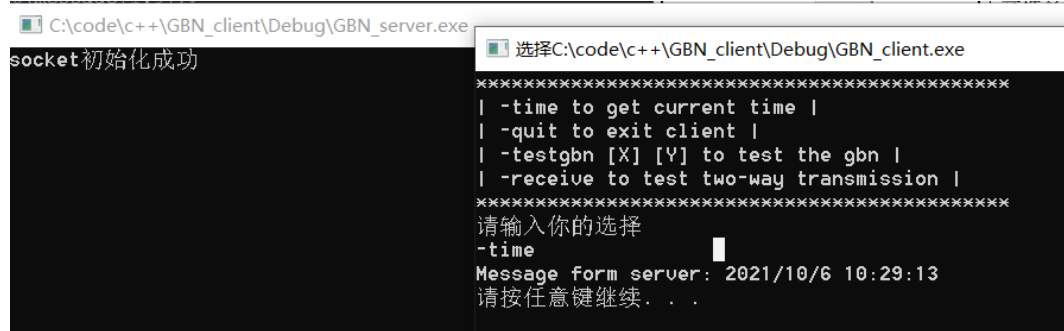
由于本次实验源代码过多，在这里就不粘到报告中，详细源代码见源代码文件夹，下面给出各个cpp文件的含义：

文件名称	含义
GBN_client.cpp	GBN接收端
GBN_server.cpp	GBN发送端
SR_client.cpp	SR接收端
SR_server.cpp	SR发送端
stop_wait_client.cpp	停等协议接收端
stop_wait_server.cpp	停等协议发送端

### 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.-time功能演示：



## 2.GBN功能演示：

```

C:\code\c++\GBN_client  C:\code\c++\GBN_client\Debug\GBN_client.exe
发送的seq为: 6          长度为: 20
发送的seq为: 7          发送的ack为 4
发送的seq为: 8          收到包的seq是 5
发送的seq为: 9          收到的数据为: Rough winds do shake
收到包的ack是: 5        长度为: 20
发送的seq为: 10         发送的ack为 5
收到包的ack是: 5        seq为 6 的包丢失
发送的seq为: 11         收到包的seq是 7
收到包的ack是: 5        发送的ack为 5
发送的seq为: 12         seq为 8 的包丢失
收到包的ack是: 5        收到包的seq是 9
发送的seq为: 13         发送的ack为 5
超时事件发生           收到包的seq是 10
发送的seq为: 6          发送的ack为 5
收到包的ack是: 6        收到包的seq是 11
发送的seq为: 7          发送的ack为 5
收到包的ack是: 7        收到包的seq是 12
发送的seq为: 8          发送的ack为 5
收到包的ack是: 9        收到包的seq是 13
发送的seq为: 10         发送的ack为 5
收到包的ack是: 10       收到包的seq是 6
发送的seq为: 11         收到的数据为: the darling buds of
收到包的ack是: 11       长度为: 20
                        发送的ack为 6
                        seq为 7 的包丢失
    
```

可以看到，当接收端seq为6的数据包丢失后，接收端一直发送他期望收到的ack5，而发送端当收到后，会立刻重传从seq为6的数据包和之后的数据包

## 接收文件和发送文件对比：

接收文件：

```

client_receive.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Shall I compare thee to a Summers day ? Thou art more lovely and more temperate
Rough winds do shake the darling buds of Maie, And summers lease hath all too short a date:
Sometime too hot the eye of heaven shines,
And often is his gold complexion dimm'd,
And every faire from faire some-time declines,
By chance, or natures changing course vntrim'd: But thy eternall Summer shall not fade, Nor loose
possession of that faire thou ow'st,
Nor shall death brag thou wandr'st in his shade,
When in eternall lines to time thou grow'st, So long as men can breath or eyes can see.
file is over.END.
    
```

发送文件：

```

server_send.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Shall I compare thee to a Summers day ? Thou art more lovely and more temperate
Rough winds do shake the darling buds of Maie, And summers lease hath all too short a date:
Sometime too hot the eye of heaven shines,
And often is his gold complexion dimm'd,
And every faire from faire some-time declines,
By chance, or natures changing course vntrim'd: But thy eternall Summer shall not fade, Nor loose
possession of that faire thou ow'st,
Nor shall death brag thou wandr'st in his shade,
When in eternall lines to time thou grow'st, So long as men can breath or eyes can see.
file is over.END.
    
```

可以看到文件实现了准确无误的传输。

### 3.GBN客户端向服务器端发送文件演示：

C:\code\c++\GBN_client\Debug\GBN_server.exe	C:\code\c++\GBN_client\Debug\GBN_client.exe
socket初始化成功 等待用户上传文件..... 收到的数据为: Shall I compare thee 长度为: 20 收到的数据为: to a Summers day ? 长度为: 20 收到的数据为: Thou art more lovely 长度为: 20 收到的数据为: and more temperate 长度为: 20 收到的数据为: Rough winds do shake 长度为: 20 收到的数据为: the darling buds of 长度为: 20 收到的数据为: Maie, And summers l 长度为: 20 收到的数据为: ease hath all too sh 长度为: 20 收到的数据为: Shall I compare thee	*****   -time to get current time     -quit to exit client     -testgbn [X] [Y] to test the gbn     -receive to test two-way transmission   ***** 请输入你的选择 -receive 发送文件已加载完成, 文件包含600个字节, 开始向服务器发送 发送的seq为: 0 收到的ack是: 0 发送的seq为: 1 收到的ack是: 1 发送的seq为: 2 收到的ack是: 2 发送的seq为: 3 收到的ack是: 3 发送的seq为: 4 收到的ack是: 4 发送的seq为: 5
收到的数据为: untrim'd: But thy e 长度为: 20 收到的数据为: ternall Summer shall 长度为: 20 收到的数据为: not fade, Nor loose 长度为: 20 收到的数据为: possession of that 长度为: 20 收到的数据为: faire thou ow'st, No 长度为: 20 收到的数据为: r shall death brag t 长度为: 20 收到的数据为: hou wandr'st in his 长度为: 20 收到的数据为: shade, When in etern 长度为: 20 收到的数据为: all lines to time th 长度为: 20 收到的数据为: ou grow'st, So long 长度为: 20 收到的数据为: as men can breath or 长度为: 20 收到的数据为: eyes can see. file 长度为: 20 收到的数据为: is over.END.烫烫烫烫 长度为: 20 文件传输完毕	收到的ack是: 12 发送的seq为: 13 收到的ack是: 13 发送的seq为: 14 收到的ack是: 14 发送的seq为: 15 收到的ack是: 15 发送的seq为: 0 收到的ack是: 0 发送的seq为: 1 收到的ack是: 1 发送的seq为: 2 收到的ack是: 2 发送的seq为: 3 收到的ack是: 3 发送的seq为: 4 收到的ack是: 4 发送的seq为: 5 收到的ack是: 5 已成功向服务器上传文件, 请重启客户端 请按任意键继续...

### 接收文件和发送文件对比：

服务器端接收文件：

server\_receive.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

Shall I compare thee to a Summers day ? Thou art more lovely and more temperate  
 Rough winds do shake the darling buds of Maie, And summers lease hath all too short a date:  
 Sometime too hot the eye of heaven shines,  
 And often is his gold complexion dimm'd,  
 And every faire from faire some-time declines,  
 By chance, or natures changing course vntrim'd: But thy eternall Summer shall not fade, Nor loose  
 possession of that faire thou ow'st,  
 Nor shall death brag thou wandr'st in his shade,  
 When in eternall lines to time thou grow'st, So long as men can breath or eyes can see.  
 file is over.END.

客户端发送文件:

client\_send.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

Shall I compare thee to a Summers day ? Thou art more lovely and more temperate  
 Rough winds do shake the darling buds of Maie, And summers lease hath all too short a date:  
 Sometime too hot the eye of heaven shines,  
 And often is his gold complexion dimm'd,  
 And every faire from faire some-time declines,  
 By chance, or natures changing course vntrim'd: But thy eternall Summer shall not fade, Nor loose  
 possession of that faire thou ow'st,  
 Nor shall death brag thou wandr'st in his shade,  
 When in eternall lines to time thou grow'st, So long as men can breath or eyes can see.  
 file is over.END.

可以看到文件实现了准确无误的传输。

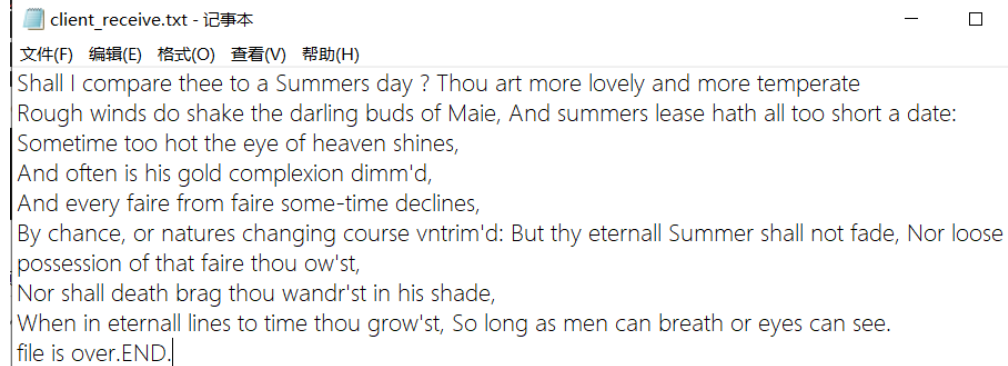
## 2.SR功能演示:

C:\code\c++\SR_client\Debug\SR_	C:\code\c++\SR_client\Debug\SR_client.ex
发送的seq为: 15	收到包的seq是 16
超时错误!	丢失的ACK为: 16
发送的seq为: 15	收到包的seq是 16
收到的ack是: 15	发送的ack为 16
发送的seq为: 0	收到包的seq是 1
超时错误!	丢失的ACK为: 1
发送的seq为: 0	seq为 1 的包丢失
超时错误!	收到包的seq是 1
发送的seq为: 0	发送的ack为 1
收到的ack是: 0	收到包的seq是 2
发送的seq为: 1	丢失的ACK为: 2
超时错误!	收到包的seq是 2
发送的seq为: 1	丢失的ACK为: 2
超时错误!	收到包的seq是 2
发送的seq为: 1	发送的ack为 2
收到的ack是: 1	seq为 3 的包丢失
发送的seq为: 2	收到包的seq是 3
超时错误!	发送的ack为 2

可以看到，当接收端seq为16，1, 2, 3的数据包丢失后，接收端只会发送未确认的数据包

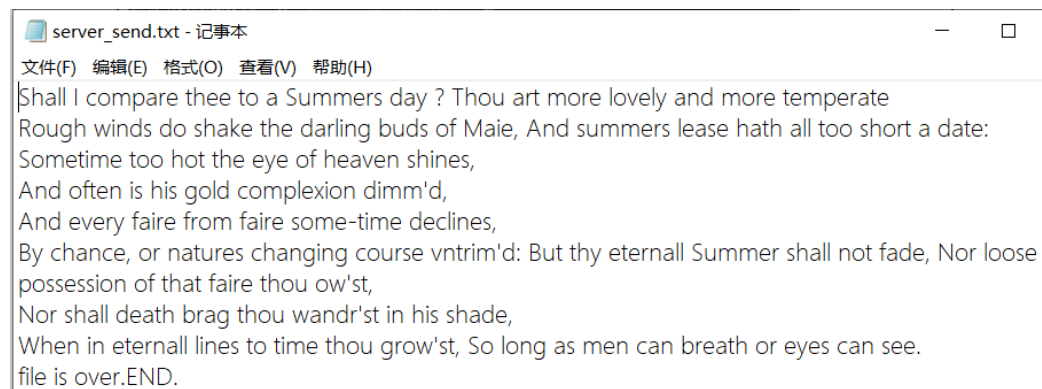
## 接收文件和发送文件对比：

接收文件：



```
client_receive.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Shall I compare thee to a Summers day ? Thou art more lovely and more temperate
Rough winds do shake the darling buds of Maie, And summers lease hath all too short a date:
Sometime too hot the eye of heaven shines,
And often is his gold complexion dimm'd,
And every faire from faire some-time declines,
By chance, or natures changing course vntrim'd: But thy eternall Summer shall not fade, Nor loose
possession of that faire thou ow'st,
Nor shall death brag thou wandr'st in his shade,
When in eternall lines to time thou grow'st, So long as men can breath or eyes can see.
file is over.END.
```

发送文件：



```
server_send.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Shall I compare thee to a Summers day ? Thou art more lovely and more temperate
Rough winds do shake the darling buds of Maie, And summers lease hath all too short a date:
Sometime too hot the eye of heaven shines,
And often is his gold complexion dimm'd,
And every faire from faire some-time declines,
By chance, or natures changing course vntrim'd: But thy eternall Summer shall not fade, Nor loose
possession of that faire thou ow'st,
Nor shall death brag thou wandr'st in his shade,
When in eternall lines to time thou grow'st, So long as men can breath or eyes can see.
file is over.END.
```

可以看到文件实现了准确无误的传输。

## 3.SR客户端向服务器端发送文件演示：



```

C:\code\c++\SR_client\Debug\SR_client.exe
正在从客户端接收数据...
收到包的seq是 1
发送的ack为 1
收到包的seq是 2
发送的ack为 2
收到包的seq是 3
发送的ack为 3
收到包的seq是 4
发送的ack为 4
收到包的seq是 5
发送的ack为 5
收到包的seq是 6
发送的ack为 6
收到包的seq是 7
发送的ack为 7
收到包的seq是 8
发送的ack为 8
收到包的seq是 9
发送的ack为 9
收到包的seq是 10
发送的ack为 10
收到包的seq是 11
发送的ack为 11
收到包的seq是 12
发送的ack为 12
收到包的seq是 13
发送的ack为 13
收到包的seq是 14
发送的ack为 14

*****
| -time to get current time |
| -quit to exit client |
| -tester [X] [V] to test the gbn |
| -receive to test two-way transmission |
*****
请输入你的选择
-receive
正在向服务器发送数据
,发送文件已加载完成, 文件包含600个字节, 开始发送
发送的seq为: 0
收到的ack是: 0
发送的seq为: 1
收到的ack是: 1
发送的seq为: 2
收到的ack是: 2
发送的seq为: 3
收到的ack是: 3
发送的seq为: 4
收到的ack是: 4
发送的seq为: 5
收到的ack是: 5
发送的seq为: 6
收到的ack是: 6
发送的seq为: 7
收到的ack是: 7
发送的seq为: 8
收到的ack是: 8
发送的seq为: 9
收到的ack是: 9

```

```

C:\code\c++\SR_client\Debug\SR_client.exe
发送的ack为 3
收到包的seq是 4
发送的ack为 4
收到包的seq是 5
发送的ack为 5
收到包的seq是 6
发送的ack为 6
收到包的seq是 7
发送的ack为 7
收到包的seq是 8
发送的ack为 8
收到包的seq是 9
发送的ack为 9
收到包的seq是 10
发送的ack为 10
收到包的seq是 11
发送的ack为 11
收到包的seq是 12
发送的ack为 12
收到包的seq是 13
发送的ack为 13
收到包的seq是 14
发送的ack为 14
文件传输完毕

收到的ack是: 2
发送的seq为: 3
收到的ack是: 3
发送的seq为: 4
收到的ack是: 4
发送的seq为: 5
收到的ack是: 5
发送的seq为: 6
收到的ack是: 6
发送的seq为: 7
收到的ack是: 7
发送的seq为: 8
收到的ack是: 8
发送的seq为: 9
收到的ack是: 9
发送的seq为: 10
收到的ack是: 10
发送的seq为: 11
收到的ack是: 11
发送的seq为: 12
收到的ack是: 12
发送的seq为: 13
收到的ack是: 13
请按任意键继续. . .

```

接收文件和发送文件对比:

服务器端接收文件:

server\_receive.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

Shall I compare thee to a Summers day ? Thou art more lovely and more temperate  
 Rough winds do shake the darling buds of Maie, And summers lease hath all too short a date:  
 Sometime too hot the eye of heaven shines,  
 And often is his gold complexion dimm'd,  
 And every faire from faire some-time declines,  
 By chance, or natures changing course vntrim'd: But thy eternall Summer shall not fade, Nor loose  
 possession of that faire thou ow'st,  
 Nor shall death brag thou wandr'st in his shade,  
 When in eternall lines to time thou grow'st, So long as men can breath or eyes can see.  
 file is over.END.

客户端发送文件:

client\_send.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

Shall I compare thee to a Summers day ? Thou art more lovely and more temperate  
 Rough winds do shake the darling buds of Maie, And summers lease hath all too short a date:  
 Sometime too hot the eye of heaven shines,  
 And often is his gold complexion dimm'd,  
 And every faire from faire some-time declines,  
 By chance, or natures changing course vntrim'd: But thy eternall Summer shall not fade, Nor loose  
 possession of that faire thou ow'st,  
 Nor shall death brag thou wandr'st in his shade,  
 When in eternall lines to time thou grow'st, So long as men can breath or eyes can see.  
 file is over.END.

可以看到文件实现了准确无误的传输。

问题讨论:

对实验过程中的思考问题进行讨论或回答。

- 1.开始不知道怎么进行模拟丢包，后来通过查阅相关资料和阅读实验指导书，知道了可以通过引入一个概率因子，然后不处理当前收到的数据包进行模拟。
- 2.对于双向数据传输的实现，其实就是在客户端和服务端同时实现GBN和SR的发送方和接收方。
- 3.SR协议的重传机制相对GBN更加复杂，但却效率更高，因为不需要重传发送窗口的所有文件，SR使用了空间换时间。

心得体会:

结合实验过程和结果给出实验的体会和收获。

同通过本次实验，更加深刻的理解到了GBN和SR协议的全双工通信，对可靠数据通信协议有了更深刻的认识，同时，也意识到，仅仅是理解了原理，是不够的，还会在工程上面临许多困难，比如计时器如何设计，如何循环利用序列号，当程序出现问题后，这种需要双方通信的程序如何debug