

计算机网络实验 3 任务三报告

1811527

朱韵

一、实验内容

在任务 3-2 的基础上，选择实现一种拥塞控制算法，也可以是改进的算法，完成给定测试文件的传输。

二、实验要求

1. 实现单向传输。
2. 对于每一个任务要求给出详细的协议设计。
3. 给出实现的拥塞控制算法的原理说明。
4. 完成给定测试文件的传输，显示传输时间和平均吞吐率。
5. 性能测试指标包括吞吐率和时延，给出图形结果并进行分析。
6. 完成详细的实验报告（每个任务完成一份）。
7. 编写的程序应结构清晰，具有较好的可读性。
8. 提交程序源码和实验报告。

三、协议设计

1. 数据报格式（存储在字符数组 send_buf 中）

- (1) 序列号：占 16 个字节，表示为 16 位二进制，位于 send_buf[0:15]
- (2) 源端口号：占 16 个字节，表示为 16 位二进制，位于 send_buf[16:31]
- (3) 目的端口号：占 16 个字节，表示为 16 位二进制，位于 send_buf[32:47]
- (4) 长度：占 16 个字节，表示为 16 位二进制，位于 send_buf[48:63]
- (5) 校验和：占 16 个字节，表示为 16 位二进制，位于 send_buf[64:79]
- (6) 最后一个数据包标志位：如果是最后一个数据包，则标志位设为 0；否则，标志位设为 1。位于 send_buf[80]。
- (7) 数据：位于 send_buf[81:]

0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
序列号																源端口号															
目的端口号																长度															
校验和																f															
数据																															

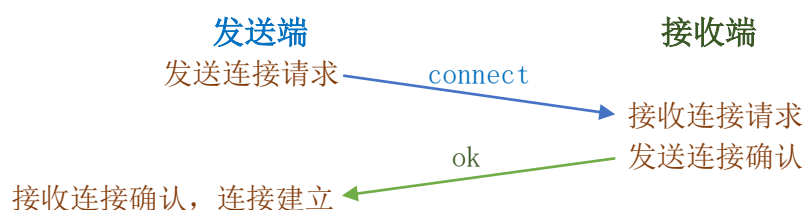
数据报（长度单位为字节），f 代表最后一个数据包标志位

2. 差错检测——计算校验和

- (1) **发送端**：将数据报按 16 位 unsigned short 序列，采用 16 位二进制反码求和运算，计算结果取反写入校验和域段
- (2) **接收端**：将数据报按 16 位 unsigned short 序列，采用 16 位二进制反码求和运算，如果结果为全 0，则代表没有检测到错误；否则，说明数据报存在差错。

3. 建立连接

发送端向接收端发送“connect”，接收端回复“ok”，连接建立成功。



4. 累积确认

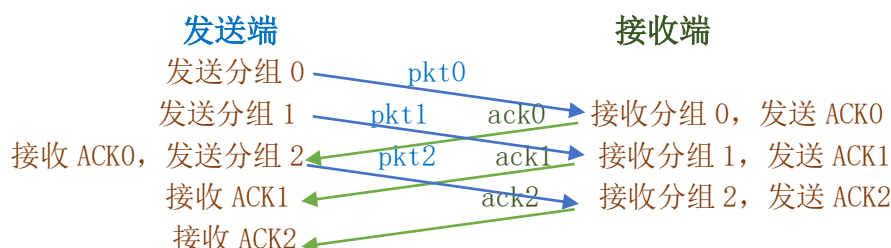
(1) 发送端:

持续接收 ACK。如果收到接收端发来的新的 ACK，代表在当前 ACK 的数据包之前的数据包均已发送成功，将窗口向右滑动，即基序号改为当前 ACK 序号的下一个序号。

(2) 接收端:

接收发送端发来的一个分组，如果分组校验和为全 0，且分组序号为期待的序号（收到数据包的序列号==expectnum），则向发送端发送 ACK+最后一个分组标志位+当前这一分组的序列号（序列号占 16 个字节，表示为 16 位二进制。ack 位于字符数组 send_ack[0:2]，当前 ack 的是否是最后一个分组标志位（0 为最后一个，1 代表不是最后一个）位于 send_ack[3]，序列号位于 send_ack[4:19]），同时 expectnum 加 1（如果 expectnum 已等于最大序列号，则令 expectnum=0）。

对于失序的分组（收到的序列号不等于 expectnum），当收到的序号小于期待的序号时，直接丢弃；当收到的序号大于期待的序号时，发送 ACK+最后一个分组标志位+（期待的序号-1）。



5. 序列号循环使用

序列号的范围为 0-500，循环使用（即序列号 500 的下一个序列号为 0）。

四、拥塞控制

1. 原理：基于窗口的方法，通过拥塞窗口的增大或减小来控制发送速率。

2. 第一阶段：慢启动阶段

初始阈值设为 20，初始拥塞控制窗口 cwnd 设为 1（可以发送一个数据包）。每收到一个新的 ACK，cwnd 加 1，并让记录重复 ACK 的变量为 0。慢启动阶段拥

塞窗口呈指数级增长。具体实现如下：

```
//慢启动阶段
else {
    cwnd = cwnd + 1;
    dupackcount = 0;
}
```

3. 第二阶段：拥塞避免阶段

当拥塞窗口到达阈值时，慢启动阶段结束，进入拥塞避免阶段。每收到一个新的 ACK，cwnd 增加 $(1/cwnd)$ ，并让记录重复 ACK 的变量为 0。拥塞避免阶段拥塞窗口呈线性增长。具体实现如下：

```
//拥塞避免阶段
if (cwnd >= ssthresh) {
    cwnd = cwnd + (1 / cwnd);
    dupackcount = 0;
}
```

4. 丢失检测

(1) 超时重传

发送端设置超时线程，判断当前未确认分组确认时间是否超时，如果没有在规定时间内（具体实现中设为 0.5 秒）收到相应的 ACK，则重传从基序号到当前序号的所有分组。此时阈值设为 $cwnd/2$ ，cwnd 设为 1，重新进入慢启动状态。如果阈值变化后为 0，则将其设为 1。

之前数据包丢失的原因很有可能是传得太快，所以超时之后要降低阈值，从而减缓传输速率。部分代码如下：

```
//如果当前时间减去开始时间大于等于0.5秒，则重传从base到当前seq-1的所有分组
if ((time_now - time_start) / CLOCKS_PER_SEC >= 0.5 && timer_flag == 1) {
    cout << "超时了" << endl;
    ssthresh = cwnd / 2;
    //阈值如果变为0，则将其置1
    if (ssthresh <= 0) {
        ssthresh = 1;
    }
    cwnd = 1; //进入慢启动阶段
    dupackcount = 0;
    //重传
```

(2) 通过三次重复 ACK 检测丢失

如果出现丢包现象，接收端会收到不符合期待的分组。如果接收到的分组序号小于期待的序号，则直接丢弃；如果接收到的分组序号大于期待的序号，则发送 ACK+最后一个分组标志位+（期待的序号-1）。

发送端接收到对大于当前基序号的 ACK 时，基序号会修改为当前 ACK 序号的下一个。如果接收端持续收到分组序号大于期待的序号，则会重复发送对期待的序号-1 的 ACK，发送端同时会收到多次比当前基序号小 1 的 ACK。如果收到三次比当前基序号小 1 的 ACK，则触发快速重传机制。发送端将阈值修改为 $cwnd/2$ ，

cwnd 修改为 ssthresh+3，进入线性增长，即拥塞避免阶段。因为此时仍能接收到 3 次重复 ACK，说明网络拥塞不是很严重，因此不用进入慢启动阶段。如果已处于快速重传状态，但仍收到重复 ACK，则将 cwnd+1，之后继续接收 ACK。当收到新的 ACK 时，退出快速重传状态。部分代码如下：

```
//收到重复ack
if (ack_seq == temp) {
    cout << "收到重复ack" << ack_seq << endl;
    //本来就在快速重传状态
    if (dupsend_flag == 1) {
        cwnd = cwnd + 1;
        cout << "本来就在快速重传状态" << endl;
        continue;
    }
    else {
        dupackcount++;
        //如果重复ack计数为3，则启动快速重传
        if (dupackcount == 3) {
            //进入快速重传状态
            dupsend_flag = 1;
            ssthresh = cwnd / 2;
            if (ssthresh <= 0) {
                ssthresh = 1;
            }
            cwnd = ssthresh + 3;
            //关闭上一阶段计时
            stop_timer();
            //开启快速重传线程
            HANDLE hthread;
            hthread = CreateThread(NULL, 0, quick_resend, NULL, 0, NULL);
            //关闭线程句柄
            CloseHandle(hthread);
        }
    }
}
```

四、代码思路

1. 发送端

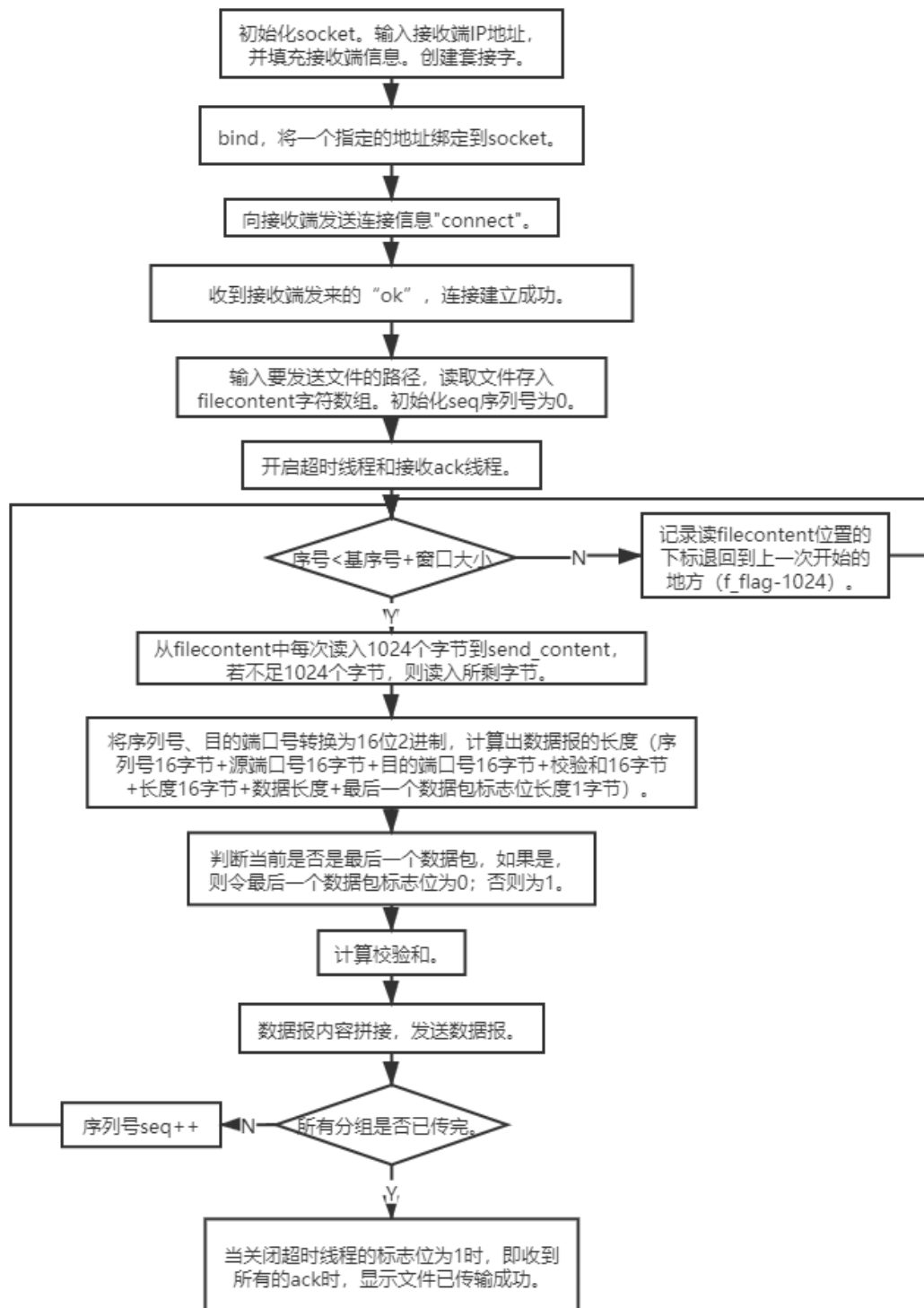
(1) 几个标志位

dupsend_flag: 快速重传状态标志位，1 为处于快速重传状态，0 为其他状态。

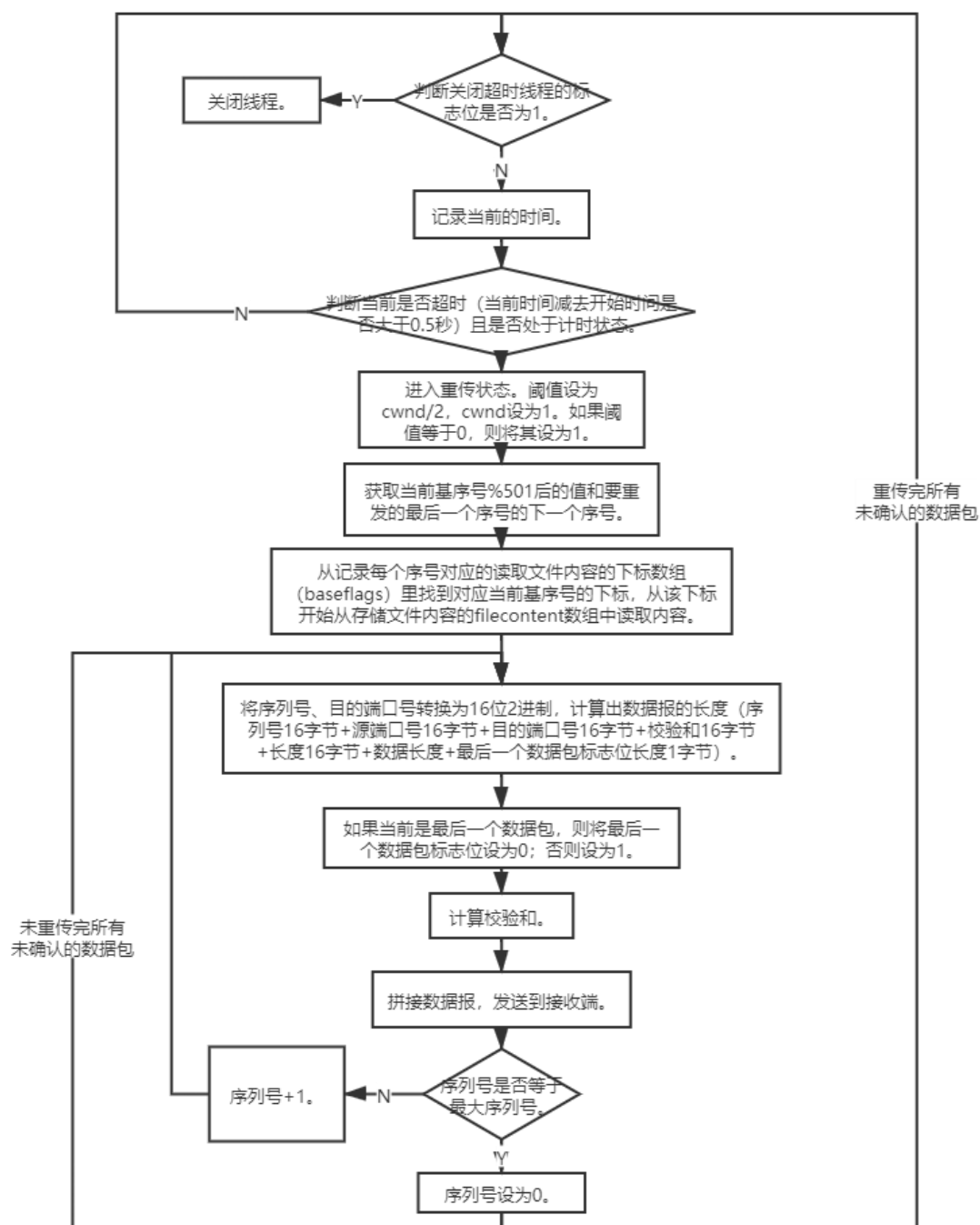
resend_flag: 重传分组状态，1 为当前在重传分组，这时主线程不发送分组，等待重传完后才能发送，0 为当前没有重传分组。

close_resend: 关闭超时和接收 ACK 线程的标志位，1 为当前已收到所有 ACK，可以关闭超时和接收 ACK 线程，0 为当前没有收完所有分组。

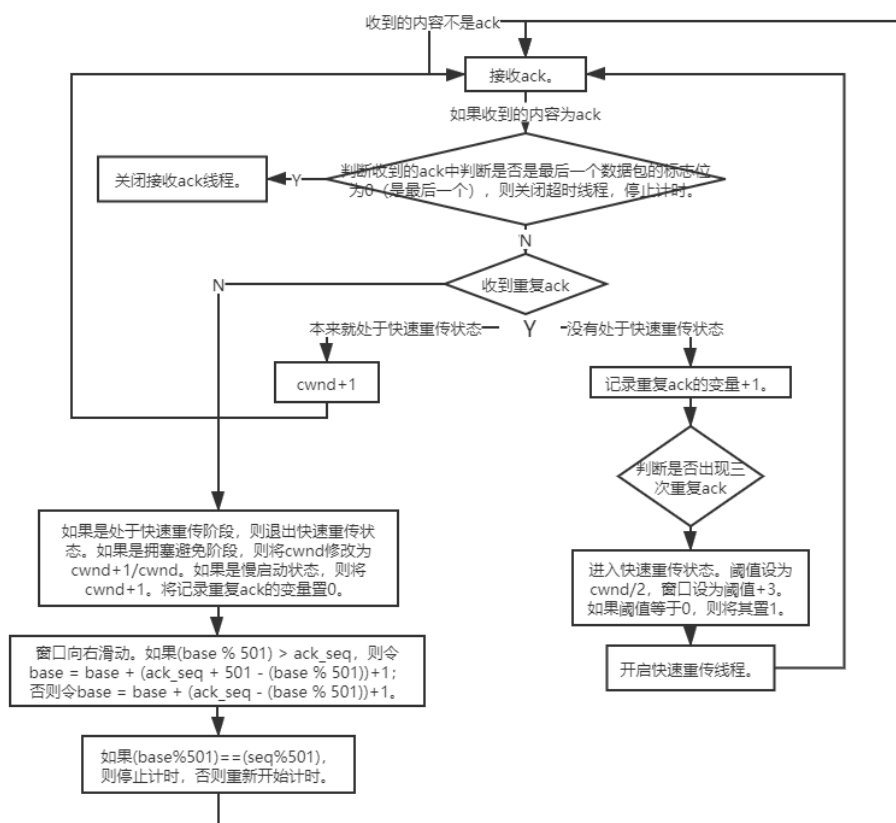
(2) 主线程



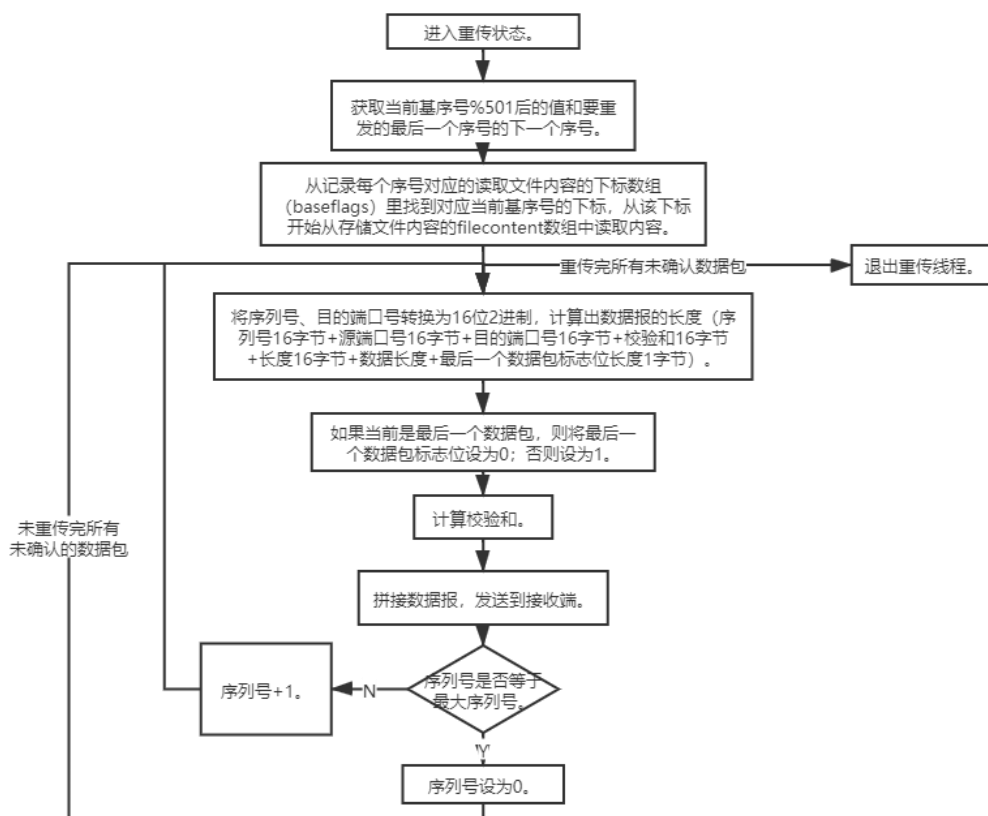
(3) 超时线程



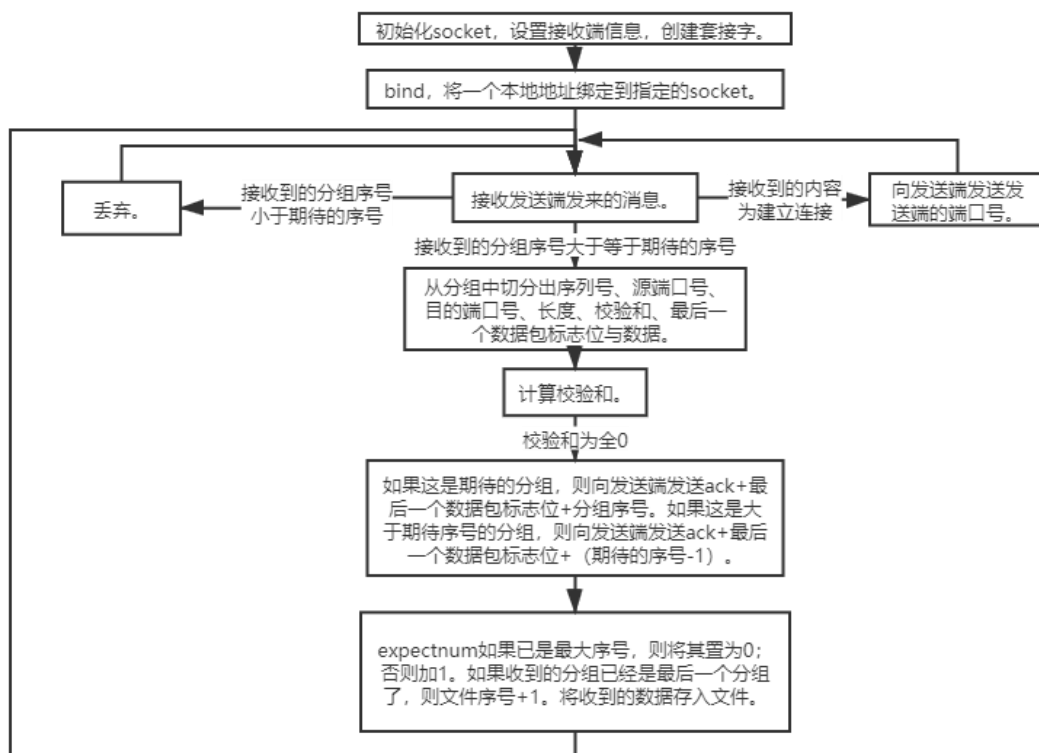
(4) 接收 ack 线程



(5) 快速重传线程



2. 接收端

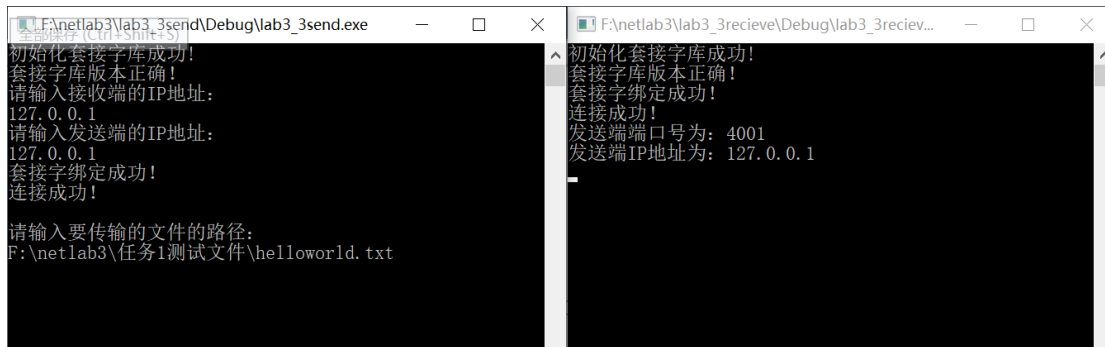


五、结果

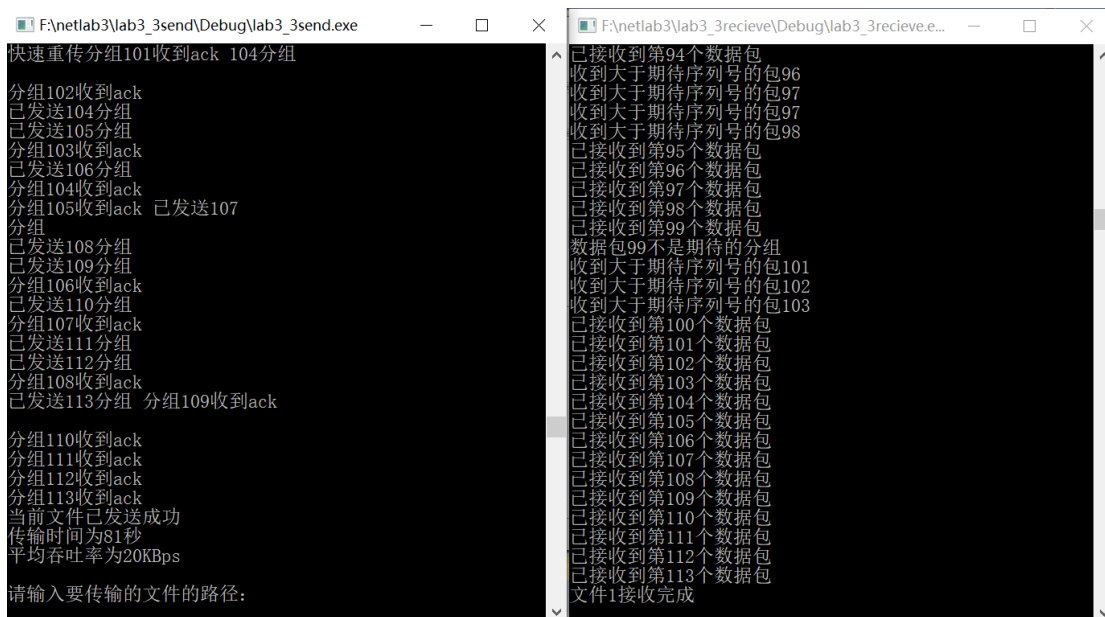
将路由器设为 10%的丢包率。



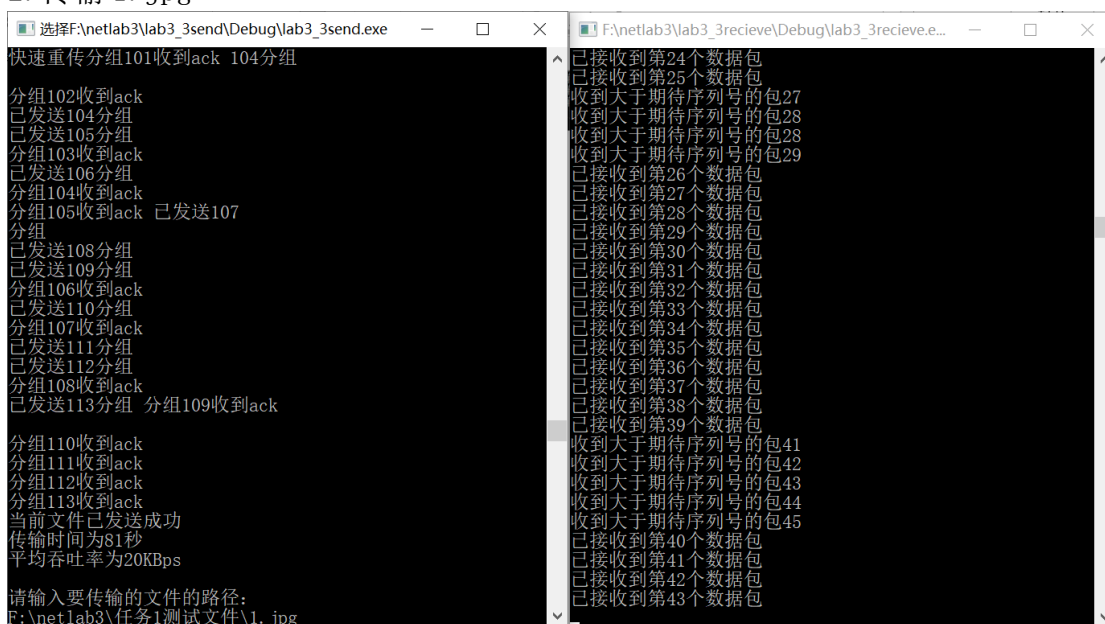
1. 连接建立，并传输 helloworld.txt



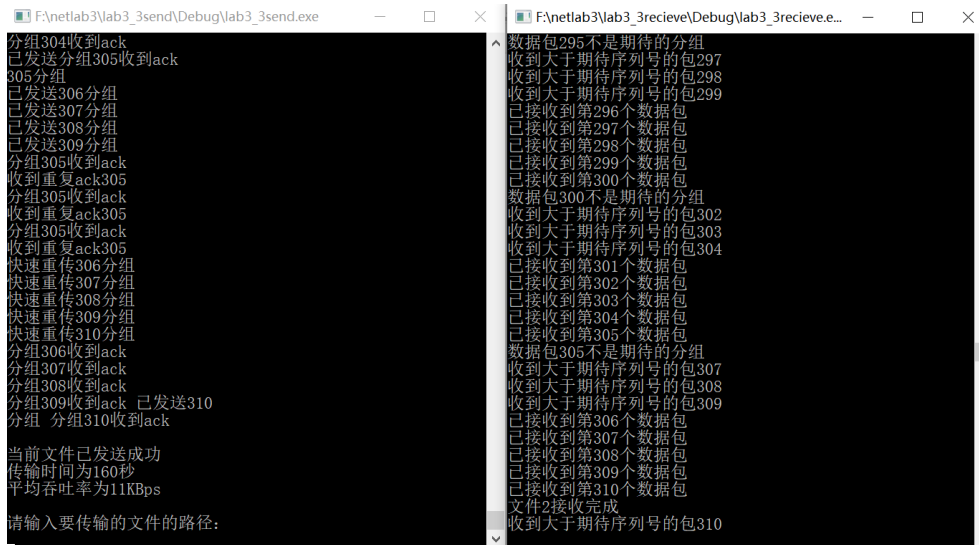
传输结果为:



2. 传输 1.jpg



传输结果为:



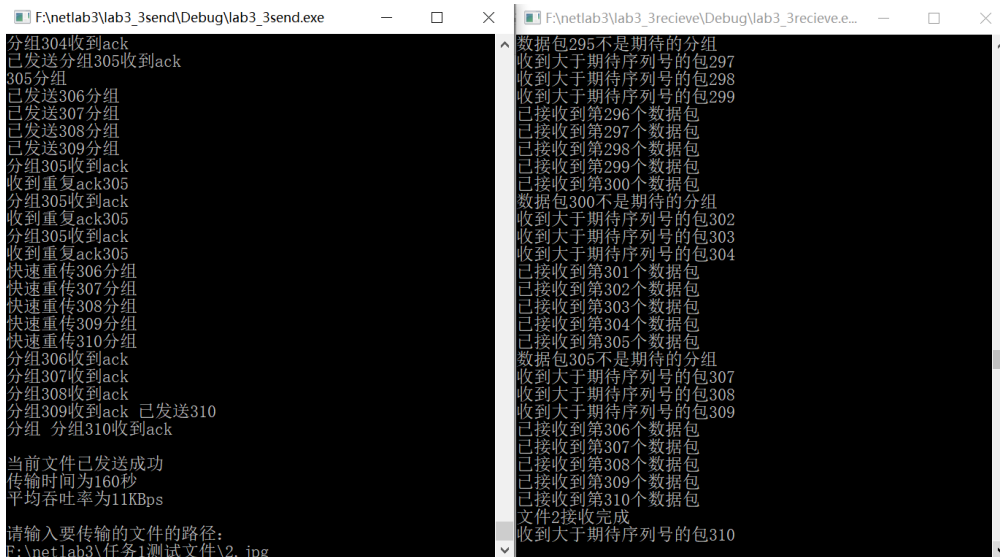
```
F:\netlab3\lab3_3send\Debug\lab3_3send.exe
分组304收到ack
已发送分组305收到ack
305分组
已发送306分组
已发送307分组
已发送308分组
已发送309分组
分组305收到ack
收到重复ack305
分组305收到ack
收到重复ack305
分组305收到ack
收到重复ack305
快速重传306分组
快速重传307分组
快速重传308分组
快速重传309分组
快速重传310分组
分组306收到ack
分组307收到ack
分组308收到ack
分组309收到ack 已发送310
分组 分组310收到ack

当前文件已发送成功
传输时间为160秒
平均吞吐率为11KBps

请输入要传输的文件的路径:
```

```
F:\netlab3\lab3_3receive\Debug\lab3_3receive.exe
数据包295不是期待的分组
收到大于期待序列号的包297
收到大于期待序列号的包298
收到大于期待序列号的包299
已接收到第296个数据包
已接收到第297个数据包
已接收到第298个数据包
已接收到第299个数据包
已接收到第300个数据包
数据包300不是期待的分组
收到大于期待序列号的包302
收到大于期待序列号的包303
收到大于期待序列号的包304
已接收到第301个数据包
已接收到第302个数据包
已接收到第303个数据包
已接收到第304个数据包
已接收到第305个数据包
数据包305不是期待的分组
收到大于期待序列号的包307
收到大于期待序列号的包308
收到大于期待序列号的包309
已接收到第306个数据包
已接收到第307个数据包
已接收到第308个数据包
已接收到第309个数据包
已接收到第310个数据包
文件2接收完成
收到大于期待序列号的包310
```

3. 传输 2. jpg



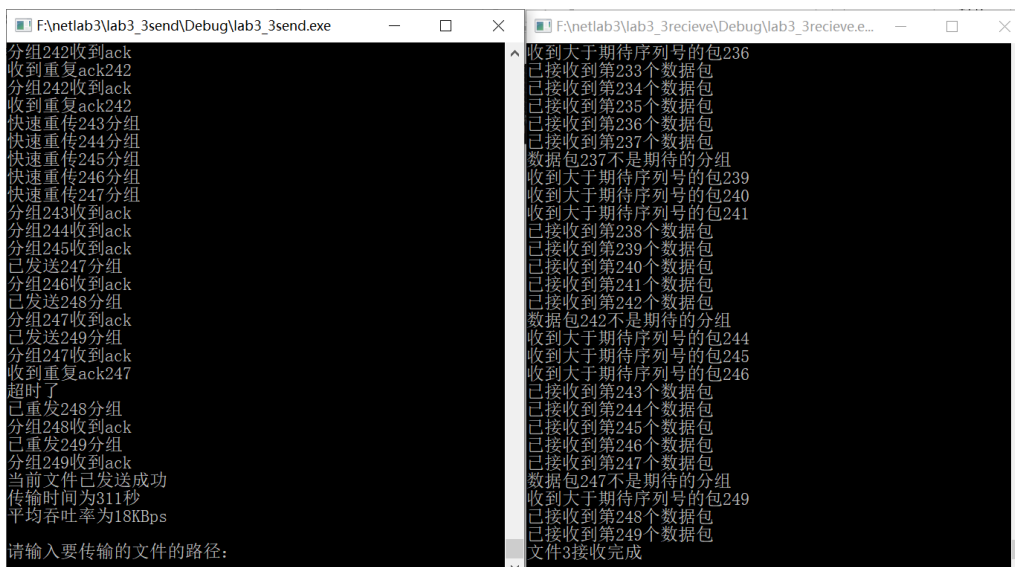
```
F:\netlab3\lab3_3send\Debug\lab3_3send.exe
分组304收到ack
已发送分组305收到ack
305分组
已发送306分组
已发送307分组
已发送308分组
已发送309分组
分组305收到ack
收到重复ack305
分组305收到ack
收到重复ack305
分组305收到ack
收到重复ack305
快速重传306分组
快速重传307分组
快速重传308分组
快速重传309分组
快速重传310分组
分组306收到ack
分组307收到ack
分组308收到ack
分组309收到ack 已发送310
分组 分组310收到ack

当前文件已发送成功
传输时间为160秒
平均吞吐率为11KBps

请输入要传输的文件的路径:
F:\netlab3\任务1测试文件\2.jpg
```

```
F:\netlab3\lab3_3receive\Debug\lab3_3receive.exe
数据包295不是期待的分组
收到大于期待序列号的包297
收到大于期待序列号的包298
收到大于期待序列号的包299
已接收到第296个数据包
已接收到第297个数据包
已接收到第298个数据包
已接收到第299个数据包
已接收到第300个数据包
数据包300不是期待的分组
收到大于期待序列号的包302
收到大于期待序列号的包303
收到大于期待序列号的包304
已接收到第301个数据包
已接收到第302个数据包
已接收到第303个数据包
已接收到第304个数据包
已接收到第305个数据包
数据包305不是期待的分组
收到大于期待序列号的包307
收到大于期待序列号的包308
收到大于期待序列号的包309
已接收到第306个数据包
已接收到第307个数据包
已接收到第308个数据包
已接收到第309个数据包
已接收到第310个数据包
文件2接收完成
收到大于期待序列号的包310
```

传输结果为:



```
F:\netlab3\lab3_3send\Debug\lab3_3send.exe
分组242收到ack
收到重复ack242
分组242收到ack
收到重复ack242
快速重传243分组
快速重传244分组
快速重传245分组
快速重传246分组
快速重传247分组
分组243收到ack
分组244收到ack
分组245收到ack
已发送247分组
分组246收到ack
已发送248分组
分组247收到ack
已发送249分组
分组247收到ack
收到重复ack247
超时了
已重发248分组
分组248收到ack
已重发249分组
分组249收到ack
当前文件已发送成功
传输时间为311秒
平均吞吐率为18KBps

请输入要传输的文件的路径:
```

```
F:\netlab3\lab3_3receive\Debug\lab3_3receive.exe
收到大于期待序列号的包236
已接收到第233个数据包
已接收到第234个数据包
已接收到第235个数据包
已接收到第236个数据包
已接收到第237个数据包
数据包237不是期待的分组
收到大于期待序列号的包239
收到大于期待序列号的包240
收到大于期待序列号的包241
已接收到第238个数据包
已接收到第239个数据包
已接收到第240个数据包
已接收到第241个数据包
已接收到第242个数据包
数据包242不是期待的分组
收到大于期待序列号的包244
收到大于期待序列号的包245
收到大于期待序列号的包246
已接收到第243个数据包
已接收到第244个数据包
已接收到第245个数据包
已接收到第246个数据包
已接收到第247个数据包
数据包247不是期待的分组
收到大于期待序列号的包249
已接收到第248个数据包
已接收到第249个数据包
文件3接收完成
```

4. 传输 3. jpg

```

F:\netlab3\lab3_3send\Debug\lab3_3send.exe
分组242收到ack
收到重复ack242
分组242收到ack
收到重复ack242
快速重传243分组
快速重传244分组
快速重传245分组
快速重传246分组
快速重传247分组
分组243收到ack
分组244收到ack
分组245收到ack
已发送247分组
分组246收到ack
已发送248分组
分组247收到ack
已发送249分组
分组247收到ack
收到重复ack247
超时了
已重发248分组
分组248收到ack
已重发249分组
分组249收到ack
当前文件已发送成功
传输时间为311秒
平均吞吐率为18KBps

请输入要传输的文件的路径:
F:\netlab3\任务1测试文件\3. jpg

F:\netlab3\lab3_3receive\Debug\lab3_3receive.exe
收到大于期待序列号的包236
已接收到第233个数据包
已接收到第234个数据包
已接收到第235个数据包
已接收到第236个数据包
已接收到第237个数据包
数据包237不是期待的分组
收到大于期待序列号的包239
收到大于期待序列号的包240
收到大于期待序列号的包241
已接收到第238个数据包
已接收到第239个数据包
已接收到第240个数据包
已接收到第241个数据包
已接收到第242个数据包
数据包242不是期待的分组
收到大于期待序列号的包244
收到大于期待序列号的包245
收到大于期待序列号的包246
已接收到第243个数据包
已接收到第244个数据包
已接收到第245个数据包
已接收到第246个数据包
已接收到第247个数据包
数据包247不是期待的分组
收到大于期待序列号的包249
已接收到第248个数据包
已接收到第249个数据包
文件3接收完成
  
```

传输结果为:

```

F:\netlab3\lab3_3send\Debug\lab3_3send.exe
分组158收到ack
收到重复ack158
分组158收到ack
收到重复ack158
快速重传159分组 分组159收到ack

快速重传160分组
160收到ack 快速重传161分组

快速重传162分组
快速重传163分组 分组161收到ack

已发送163分组
已发送164分组
分组162收到ack
已发送165分组
分组163收到ack
分组163收到ack
收到重复ack163
超时了
已重发164分组
分组164收到ack
已重发165分组
分组165收到ack
当前文件已发送成功
传输时间为969秒
平均吞吐率为12KBps

请输入要传输的文件的路径:

F:\netlab3\lab3_3receive\Debug\lab3_3receive.exe
收到大于期待序列号的包152
已接收到第149个数据包
已接收到第150个数据包
已接收到第151个数据包
已接收到第152个数据包
已接收到第153个数据包
数据包153不是期待的分组
收到大于期待序列号的包155
收到大于期待序列号的包156
收到大于期待序列号的包157
已接收到第154个数据包
已接收到第155个数据包
已接收到第156个数据包
已接收到第157个数据包
已接收到第158个数据包
数据包158不是期待的分组
收到大于期待序列号的包160
收到大于期待序列号的包161
收到大于期待序列号的包162
已接收到第159个数据包
已接收到第160个数据包
已接收到第161个数据包
已接收到第162个数据包
已接收到第163个数据包
数据包163不是期待的分组
收到大于期待序列号的包165
已接收到第164个数据包
已接收到第165个数据包
文件4接收完成
  
```

5. 存入文件夹中的结果

