

计算机网络实验 3 任务一报告

1811527

朱韵

一、实验内容

利用数据报套接字在用户空间实现面向连接的可靠数据传输，功能包括：建立连接、差错检测、确认重传。流量控制采用停等机制，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) 数据：位于 send_buf[80:]

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																数据															
数据																															

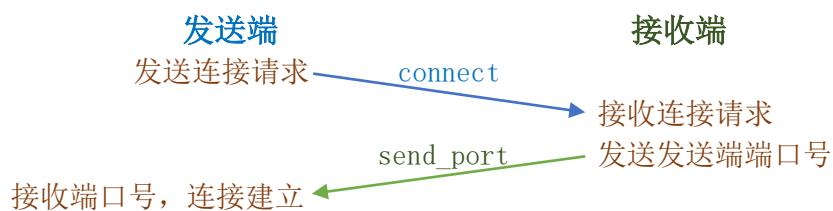
数据报（长度单位为字节）

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

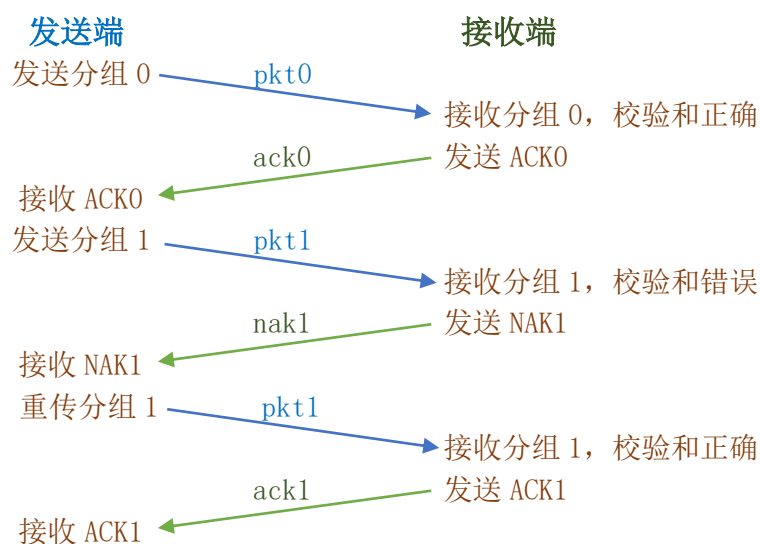
发送端向接收端发送“connect”，接收端回复发送端的端口号（16 个字节，表示为 16 位二进制，1 位占一个字节），连接即建立成功。



4. 确认重传

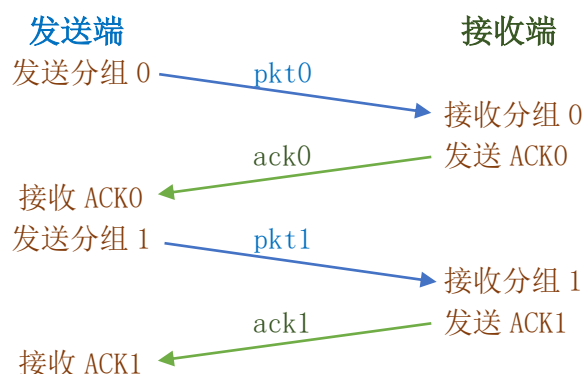
(1) **发送端**：接收 ACK 或者 NAK。如果收到接收端发来的 ACK，则继续发送下一个分组；如果收到接收端发来的 NAK 则重传当前分组。

(2) **接收端**：接收发送端发来的分组，如果分组校验和为全 0，则向发送端发送 ACK（具体内容为 ack+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]，序列号位于 send_ack[3:18]）；如果分组校验和为全 1，则向发送端发送 NAK（具体内容为 nak+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。nak 位于字符数组 send_nak[0:2]，序列号位于 send_nak[3:18]）。



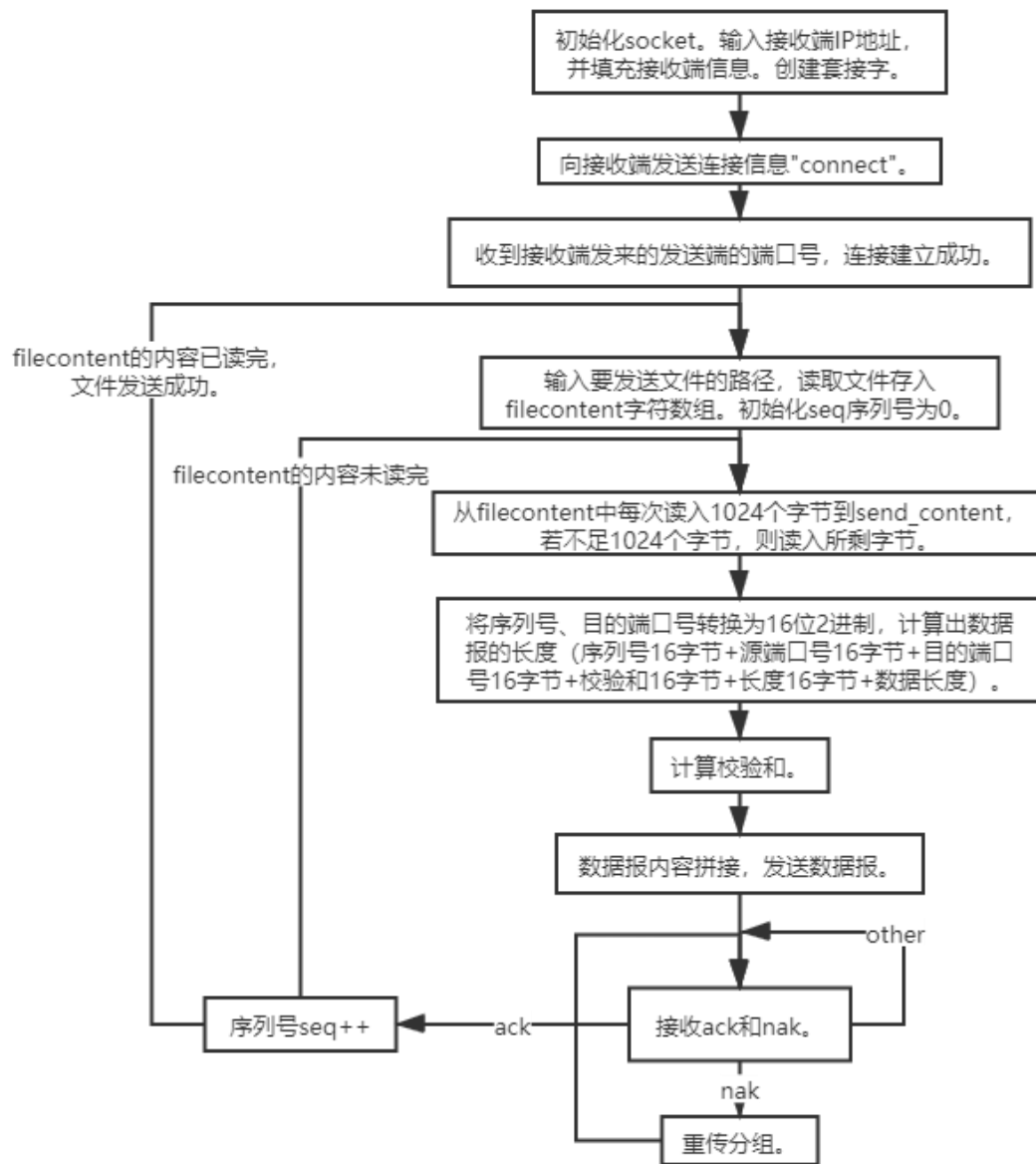
5. 流量控制——停等机制

发送端发送一个分组，接收端接收分组，若校验和正确，则发送 ACK，发送端接收 ACK，再发送下一个分组。

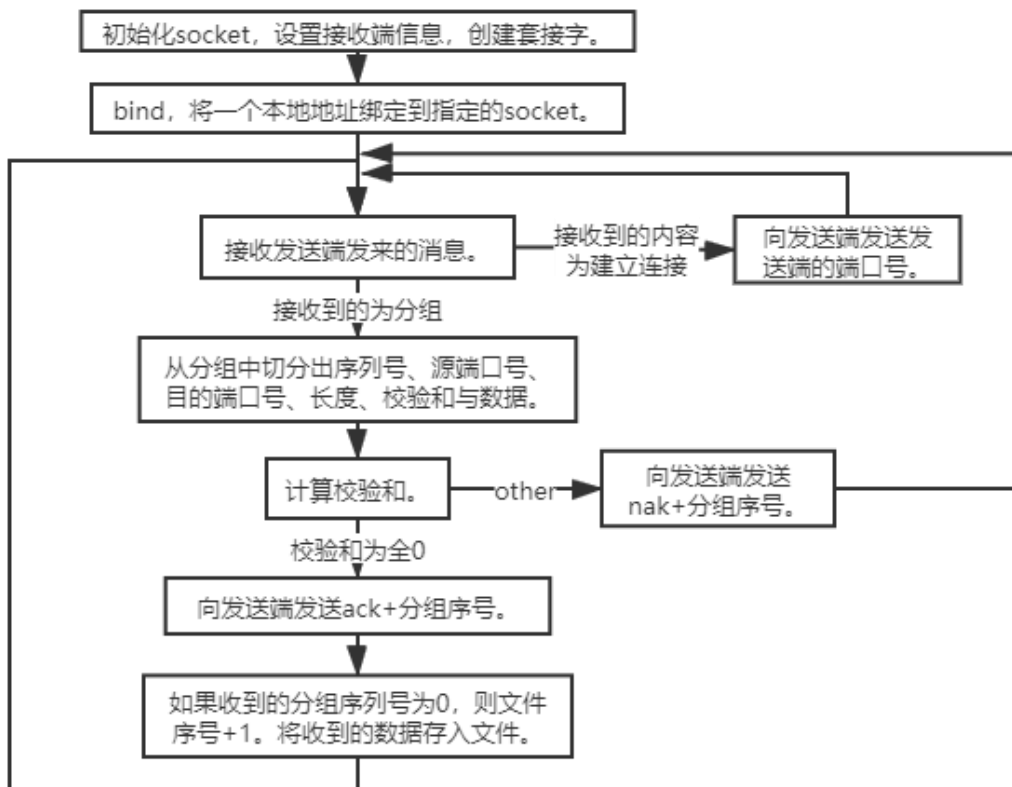


四、代码思路

1. 发送端

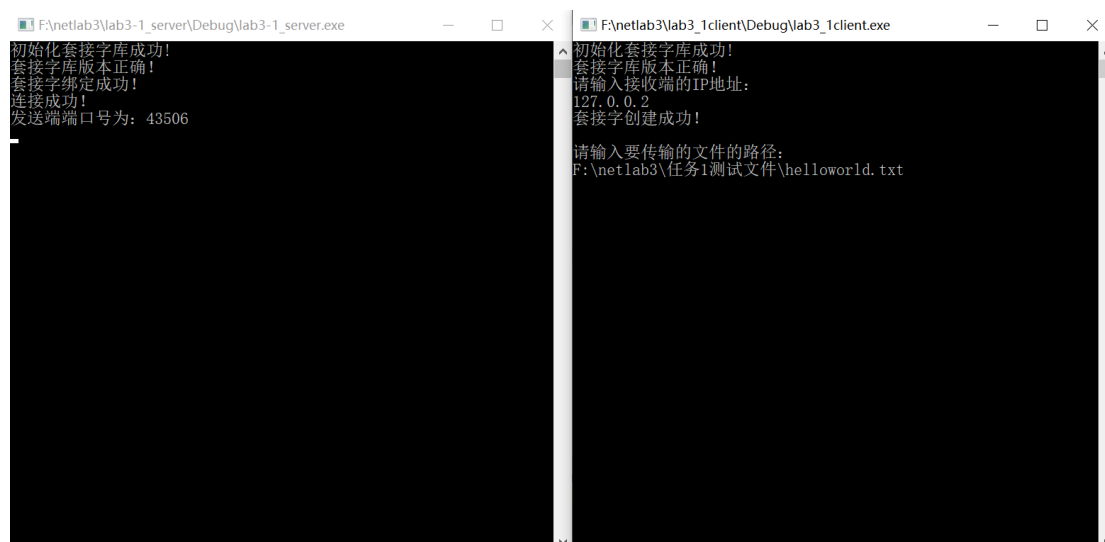


2. 接收端

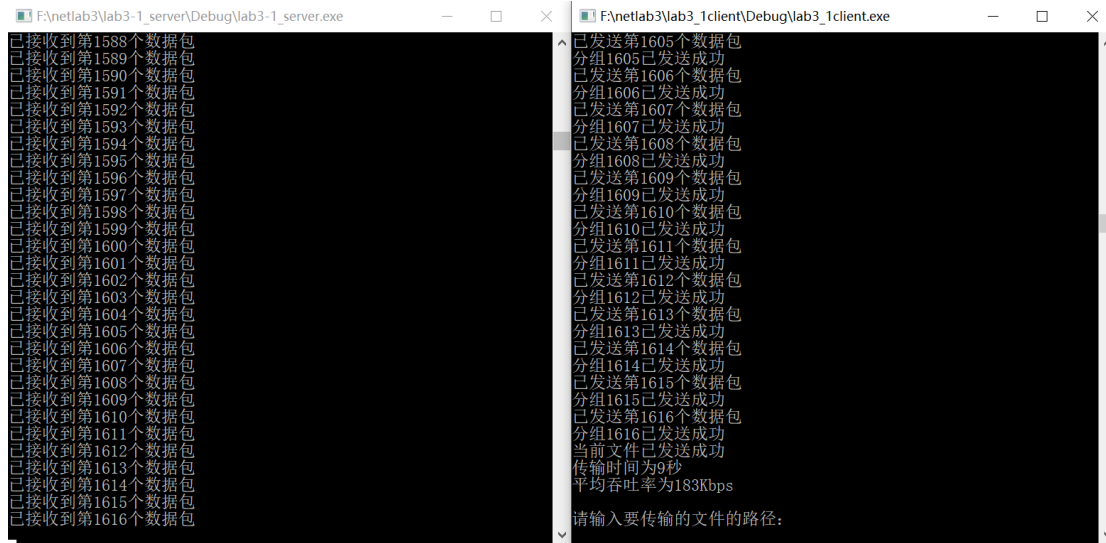


五、结果

1. 连接建立, 并传输 helloworld.txt



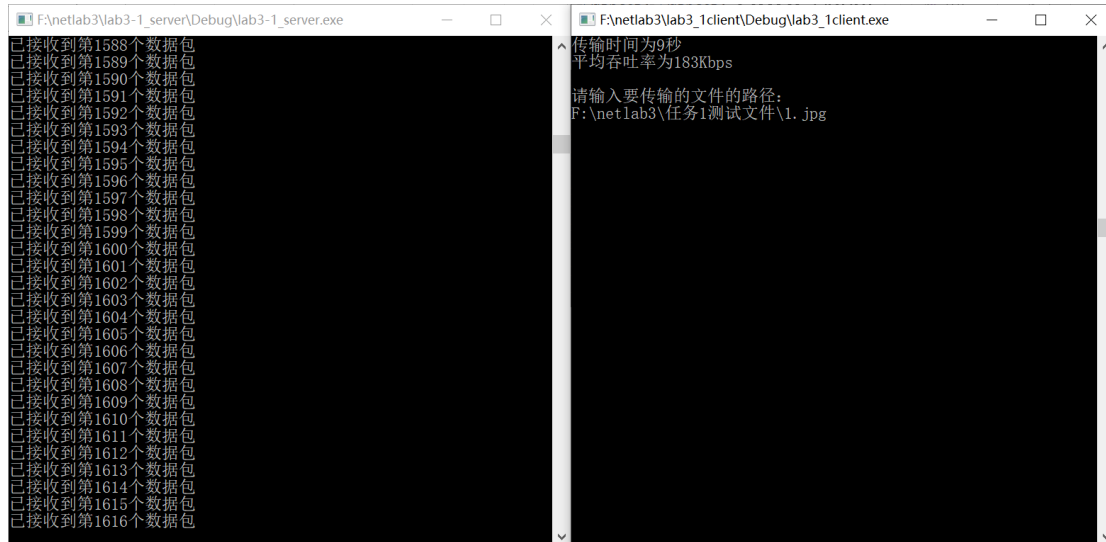
传输结果为：



```
F:\netlab3\lab3-1_server(Debug)\lab3-1_server.exe
已接收到第1588个数据包
已接收到第1589个数据包
已接收到第1590个数据包
已接收到第1591个数据包
已接收到第1592个数据包
已接收到第1593个数据包
已接收到第1594个数据包
已接收到第1595个数据包
已接收到第1596个数据包
已接收到第1597个数据包
已接收到第1598个数据包
已接收到第1599个数据包
已接收到第1600个数据包
已接收到第1601个数据包
已接收到第1602个数据包
已接收到第1603个数据包
已接收到第1604个数据包
已接收到第1605个数据包
已接收到第1606个数据包
已接收到第1607个数据包
已接收到第1608个数据包
已接收到第1609个数据包
已接收到第1610个数据包
已接收到第1611个数据包
已接收到第1612个数据包
已接收到第1613个数据包
已接收到第1614个数据包
已接收到第1615个数据包
已接收到第1616个数据包

F:\netlab3\lab3_1client(Debug)\lab3_1client.exe
已发送第1605个数据包
分组1605已发送成功
已发送第1606个数据包
分组1606已发送成功
已发送第1607个数据包
分组1607已发送成功
已发送第1608个数据包
分组1608已发送成功
已发送第1609个数据包
分组1609已发送成功
已发送第1610个数据包
分组1610已发送成功
已发送第1611个数据包
分组1611已发送成功
已发送第1612个数据包
分组1612已发送成功
已发送第1613个数据包
分组1613已发送成功
已发送第1614个数据包
分组1614已发送成功
已发送第1615个数据包
分组1615已发送成功
已发送第1616个数据包
当前文件已发送成功
传输时间为9秒
平均吞吐率为183Kbps
请输入要传输的文件的路径:
```

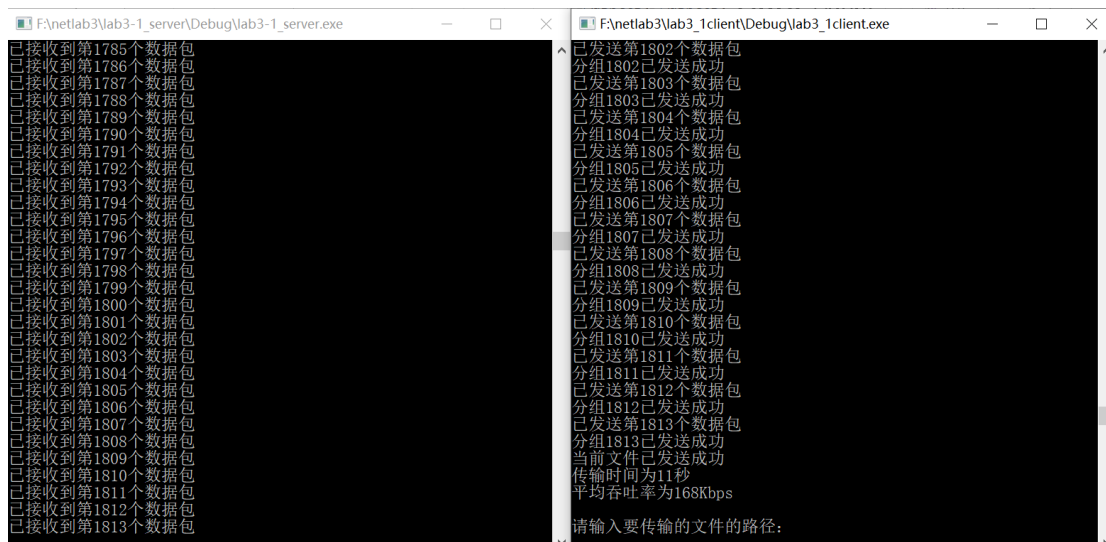
2. 传输 1.jpg



```
F:\netlab3\lab3-1_server(Debug)\lab3-1_server.exe
已接收到第1588个数据包
已接收到第1589个数据包
已接收到第1590个数据包
已接收到第1591个数据包
已接收到第1592个数据包
已接收到第1593个数据包
已接收到第1594个数据包
已接收到第1595个数据包
已接收到第1596个数据包
已接收到第1597个数据包
已接收到第1598个数据包
已接收到第1599个数据包
已接收到第1600个数据包
已接收到第1601个数据包
已接收到第1602个数据包
已接收到第1603个数据包
已接收到第1604个数据包
已接收到第1605个数据包
已接收到第1606个数据包
已接收到第1607个数据包
已接收到第1608个数据包
已接收到第1609个数据包
已接收到第1610个数据包
已接收到第1611个数据包
已接收到第1612个数据包
已接收到第1613个数据包
已接收到第1614个数据包
已接收到第1615个数据包
已接收到第1616个数据包

F:\netlab3\lab3_1client(Debug)\lab3_1client.exe
传输时间为9秒
平均吞吐率为183Kbps
请输入要传输的文件的路径:
F:\netlab3\任务1测试文件\1.jpg
```

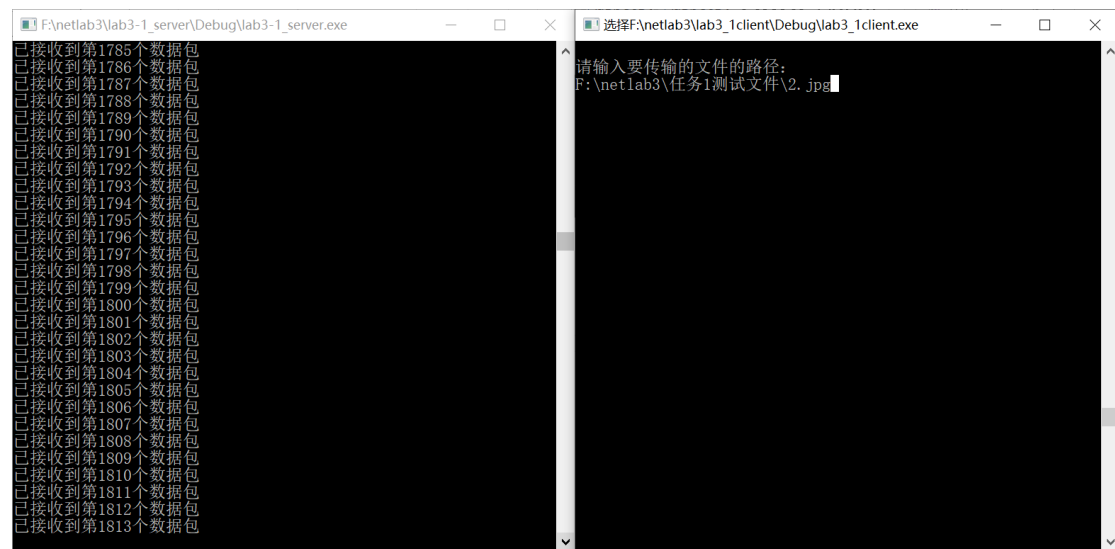
传输结果为：



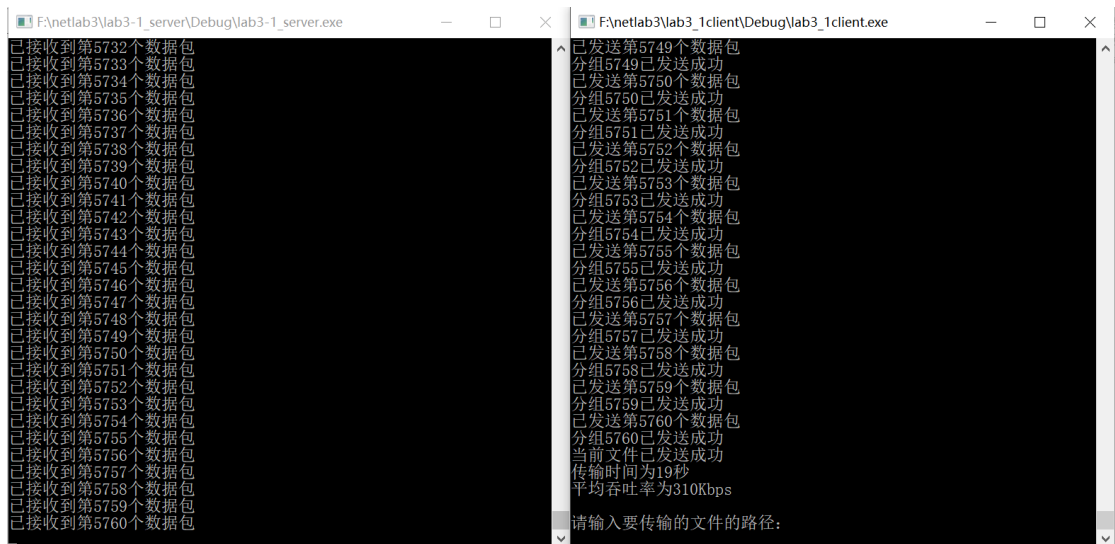
```
F:\netlab3\lab3-1_server(Debug)\lab3-1_server.exe
已接收到第1785个数据包
已接收到第1786个数据包
已接收到第1787个数据包
已接收到第1788个数据包
已接收到第1789个数据包
已接收到第1790个数据包
已接收到第1791个数据包
已接收到第1792个数据包
已接收到第1793个数据包
已接收到第1794个数据包
已接收到第1795个数据包
已接收到第1796个数据包
已接收到第1797个数据包
已接收到第1798个数据包
已接收到第1799个数据包
已接收到第1800个数据包
已接收到第1801个数据包
已接收到第1802个数据包
已接收到第1803个数据包
已接收到第1804个数据包
已接收到第1805个数据包
已接收到第1806个数据包
已接收到第1807个数据包
已接收到第1808个数据包
已接收到第1809个数据包
已接收到第1810个数据包
已接收到第1811个数据包
已接收到第1812个数据包
已接收到第1813个数据包

F:\netlab3\lab3_1client(Debug)\lab3_1client.exe
已发送第1802个数据包
分组1802已发送成功
已发送第1803个数据包
分组1803已发送成功
已发送第1804个数据包
分组1804已发送成功
已发送第1805个数据包
分组1805已发送成功
已发送第1806个数据包
分组1806已发送成功
已发送第1807个数据包
分组1807已发送成功
已发送第1808个数据包
分组1808已发送成功
已发送第1809个数据包
分组1809已发送成功
已发送第1810个数据包
分组1810已发送成功
已发送第1811个数据包
分组1811已发送成功
已发送第1812个数据包
分组1812已发送成功
已发送第1813个数据包
分组1813已发送成功
当前文件已发送成功
传输时间为11秒
平均吞吐率为168Kbps
请输入要传输的文件的路径:
```

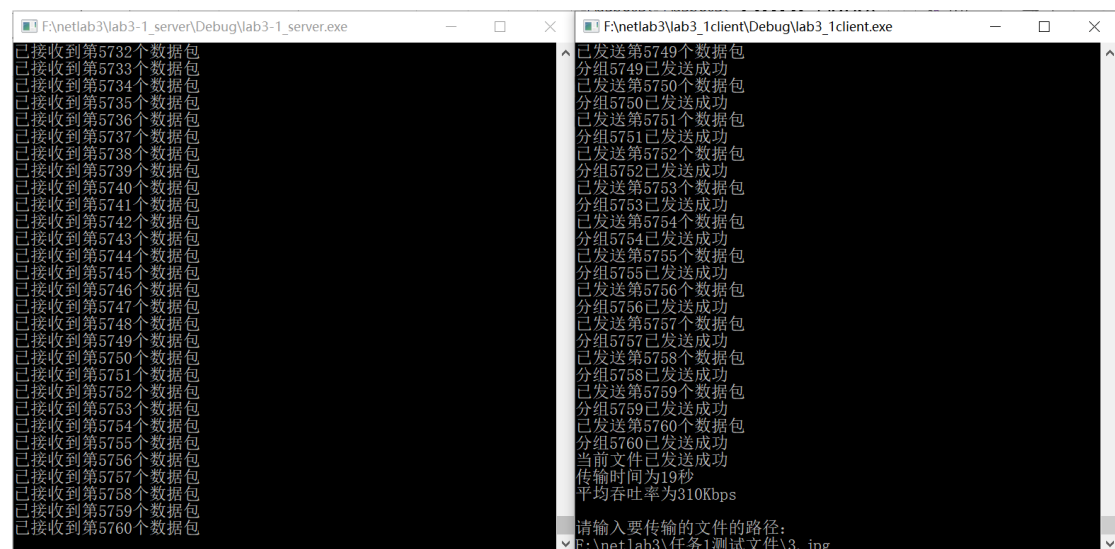
3. 传输 2. jpg



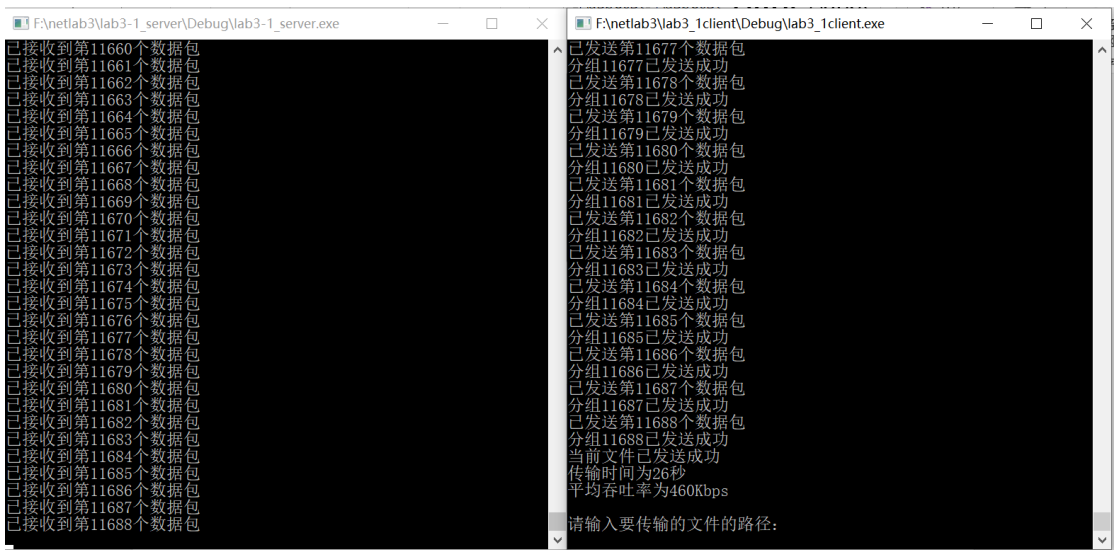
传输结果为:



4. 传输 3. jpg



传输结果为：



5. 存入文件夹中的结果

