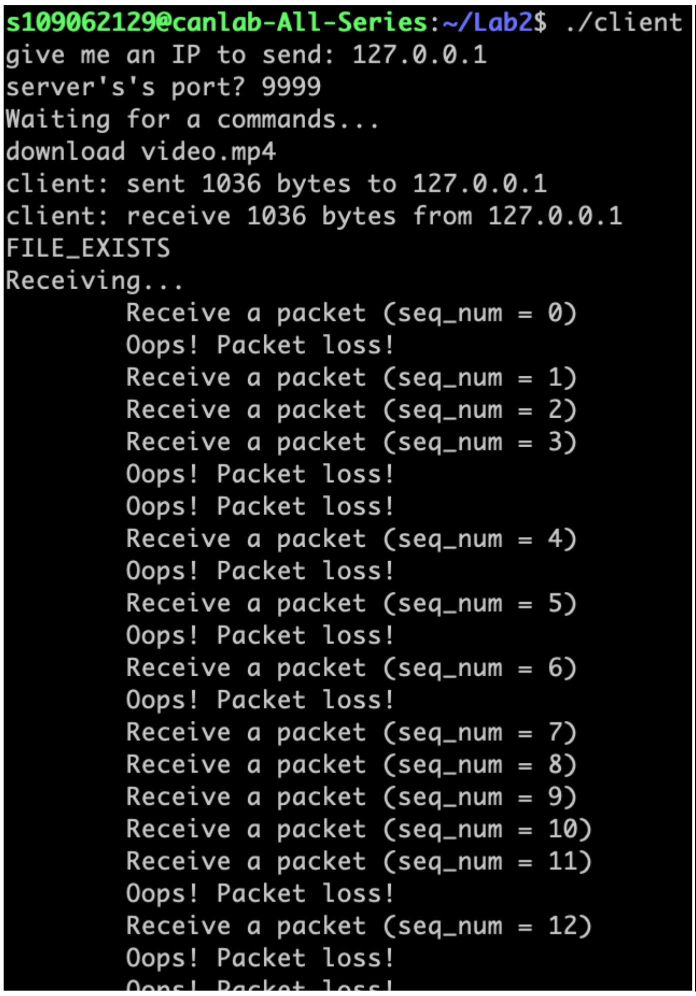
Lab 2

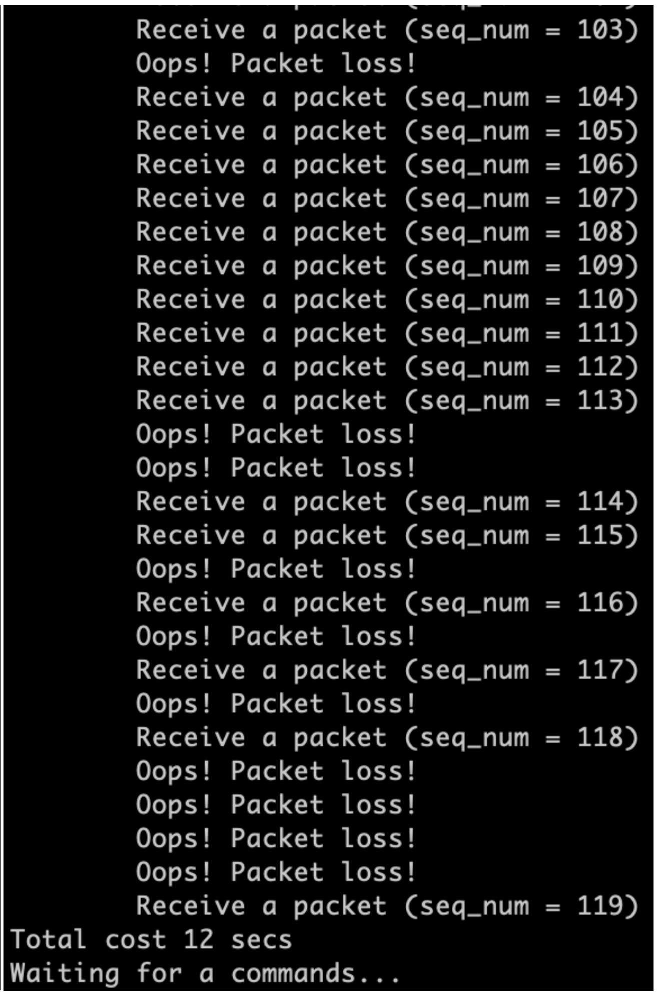
學號: 109062129　　　　　　　　　　姓名: 林奕廷

1. Description
   This socket programming C program can be compiled by gcc, run under Linux machine, and it meets all the requirements. The client can download a video file from the server using stop-and-wait mechanism through a UDP socket.

2. Example
   client.c

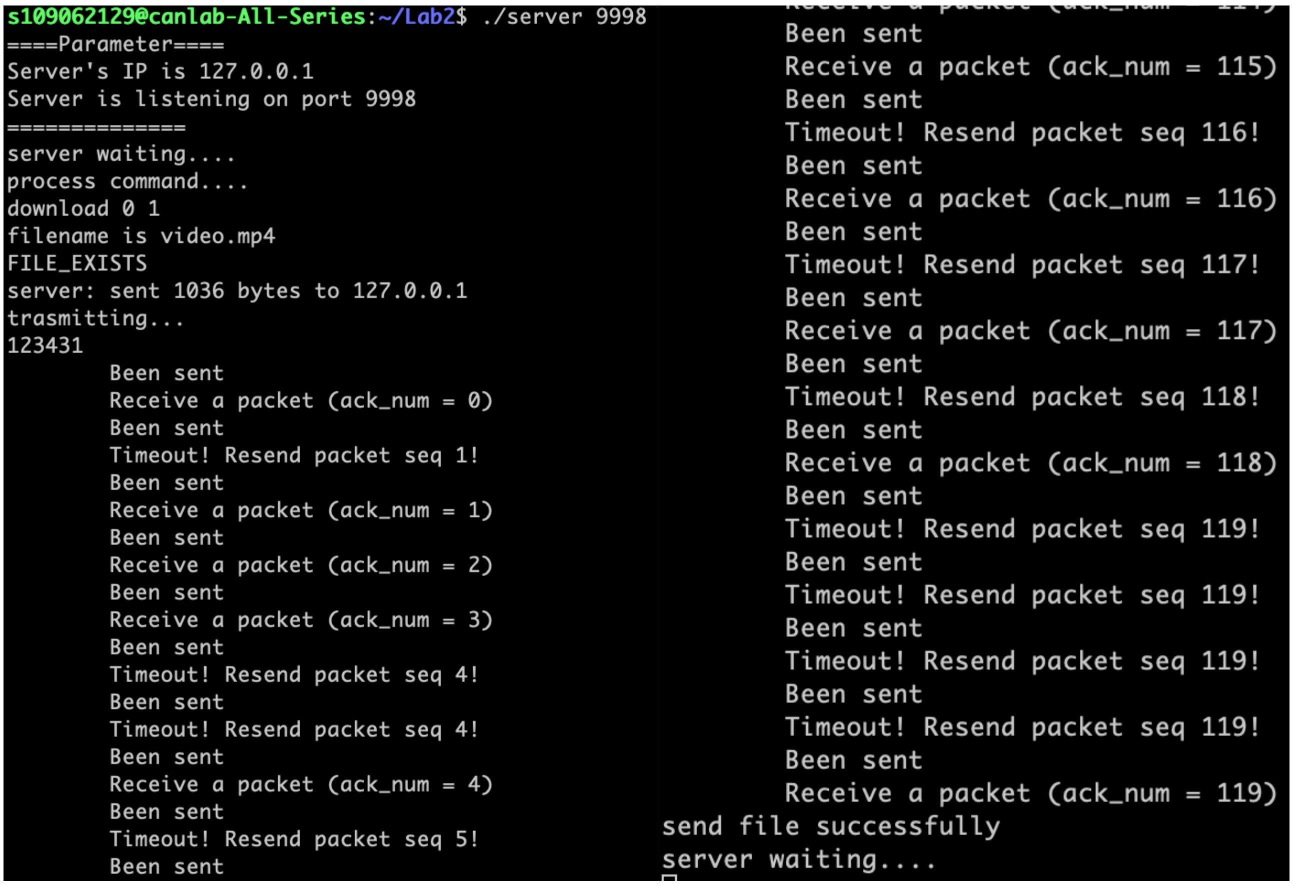


```
s109062129@canlab-All-Series:~/Lab2$ ./client
give me an IP to send: 127.0.0.1
server's's port? 9999
Waiting for a commands...
download video.mp4
client: sent 1036 bytes to 127.0.0.1
client: receive 1036 bytes from 127.0.0.1
FILE_EXISTS
Receiving...
        Receive a packet (seq_num = 0)
        Oops! Packet loss!
        Receive a packet (seq_num = 1)
        Receive a packet (seq_num = 2)
        Receive a packet (seq_num = 3)
        Oops! Packet loss!
        Oops! Packet loss!
        Receive a packet (seq_num = 4)
        Oops! Packet loss!
        Receive a packet (seq_num = 5)
        Oops! Packet loss!
        Receive a packet (seq_num = 6)
        Oops! Packet loss!
        Receive a packet (seq_num = 7)
        Receive a packet (seq_num = 8)
        Receive a packet (seq_num = 9)
        Receive a packet (seq_num = 10)
        Receive a packet (seq_num = 11)
        Oops! Packet loss!
        Receive a packet (seq_num = 12)
        Oops! Packet loss!
        Oops! Packet loss!
```



```
        Receive a packet (seq_num = 103)
        Oops! Packet loss!
        Receive a packet (seq_num = 104)
        Receive a packet (seq_num = 105)
        Receive a packet (seq_num = 106)
        Receive a packet (seq_num = 107)
        Receive a packet (seq_num = 108)
        Receive a packet (seq_num = 109)
        Receive a packet (seq_num = 110)
        Receive a packet (seq_num = 111)
        Receive a packet (seq_num = 112)
        Receive a packet (seq_num = 113)
        Oops! Packet loss!
        Oops! Packet loss!
        Receive a packet (seq_num = 114)
        Receive a packet (seq_num = 115)
        Oops! Packet loss!
        Receive a packet (seq_num = 116)
        Oops! Packet loss!
        Receive a packet (seq_num = 117)
        Oops! Packet loss!
        Receive a packet (seq_num = 118)
        Oops! Packet loss!
        Oops! Packet loss!
        Oops! Packet loss!
        Oops! Packet loss!
        Receive a packet (seq_num = 119)
Total cost 12 secs
Waiting for a commands...
```

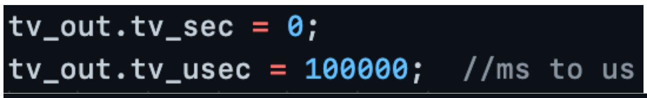
server.c



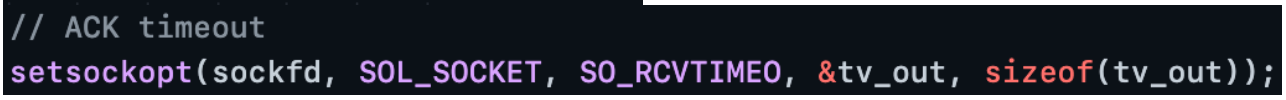
```
s109062129@canlab-All-Series:~/Lab2$ ./server 9998
====Parameter====
Server's IP is 127.0.0.1
Server is listening on port 9998
==============
server waiting....
process command....
download 0 1
filename is video.mp4
FILE_EXISTS
server: sent 1036 bytes to 127.0.0.1
trasmitting...
123431
        Been sent
        Receive a packet (ack_num = 0)
        Been sent
        Timeout! Resend packet seq 1!
        Been sent
        Receive a packet (ack_num = 1)
        Been sent
        Receive a packet (ack_num = 2)
        Been sent
        Receive a packet (ack_num = 3)
        Been sent
        Timeout! Resend packet seq 4!
        Been sent
        Timeout! Resend packet seq 4!
        Been sent
        Receive a packet (ack_num = 4)
        Been sent
        Timeout! Resend packet seq 5!
        Been sent
```

```
        Been sent
        Receive a packet (ack_num = 115)
        Been sent
        Timeout! Resend packet seq 116!
        Been sent
        Receive a packet (ack_num = 116)
        Been sent
        Timeout! Resend packet seq 117!
        Been sent
        Receive a packet (ack_num = 117)
        Been sent
        Timeout! Resend packet seq 118!
        Been sent
        Receive a packet (ack_num = 118)
        Been sent
        Timeout! Resend packet seq 119!
        Been sent
        Timeout! Resend packet seq 119!
        Been sent
        Timeout! Resend packet seq 119!
        Been sent
        Timeout! Resend packet seq 119!
        Been sent
        Receive a packet (ack_num = 119)
send file successfully
server waiting....
```

3. Code snippets
   - Server.c

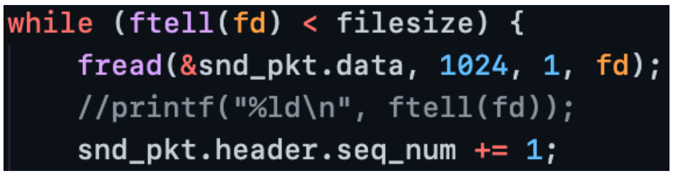


```c
tv_out.tv_sec = 0;
tv_out.tv_usec = 100000;  //ms to us
```



```c
// ACK timeout
setsockopt(sockfd, SOL_SOCKET, SO_RCVTIMEO, &tv_out, sizeof(tv_out));
```

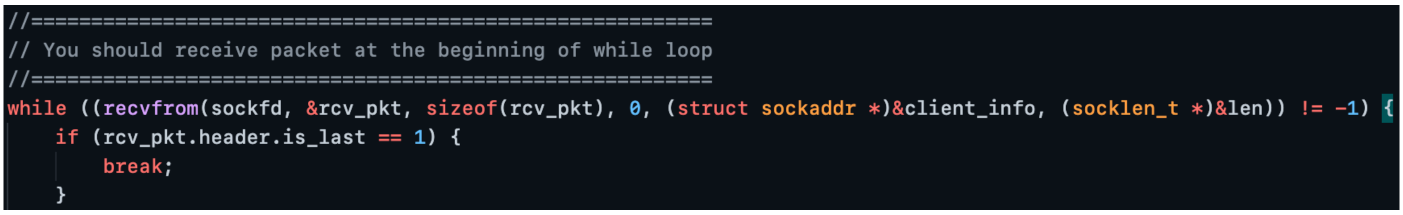
Use this built-in function to implement timeout for the server. The recvfrom function below it will return int < 0 once timeout is detected.



```c
while (ftell(fd) < filesize) {
    fread(&snd_pkt.data, 1024, 1, fd);
    //printf("%ld\n", ftell(fd));
    snd_pkt.header.seq_num += 1;
```

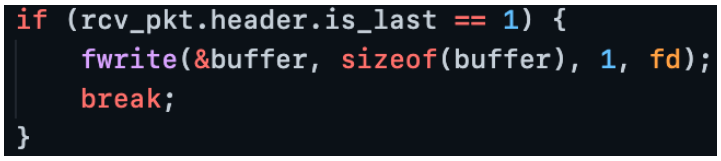
I use a while-loop to move the file pointer by 1024 before each sendto action. A packet is sent in each round, until the pointer reaches the end of the file.

   - Clinet.c



```c
//=======================================================
// You should receive packet at the beginning of while loop
//=======================================================
while ((recvfrom(sockfd, &rcv_pkt, sizeof(rcv_pkt), 0, (struct sockaddr *)&client_info, (socklen_t *)&len)) != -1) {
    if (rcv_pkt.header.is_last == 1) {
        break;
    }
```

I use a while-loop at the beginning of the while(1) loop provided in the template, to keep receiving packet. If the packet is lost(simulated), break out of this loop. If not, I use memcpy to write received data into buffer char. After that, send a ACK message with sequence number back to the server.



```c
if (rcv_pkt.header.is_last == 1) {
    fwrite(&buffer, sizeof(buffer), 1, fd);
    break;
}
```

After receiving packet with is_last flag set to 1, the program will break the while(recvfrom()...) loop, and write buffer into the file, and break the while(1) loop afterwards. Then the receiving process is done.