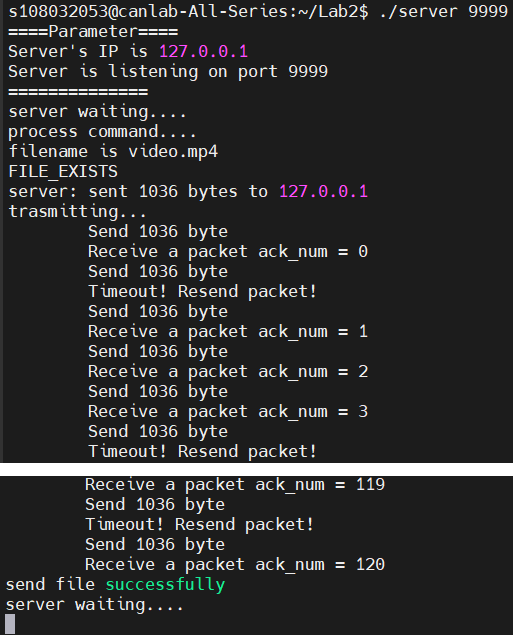
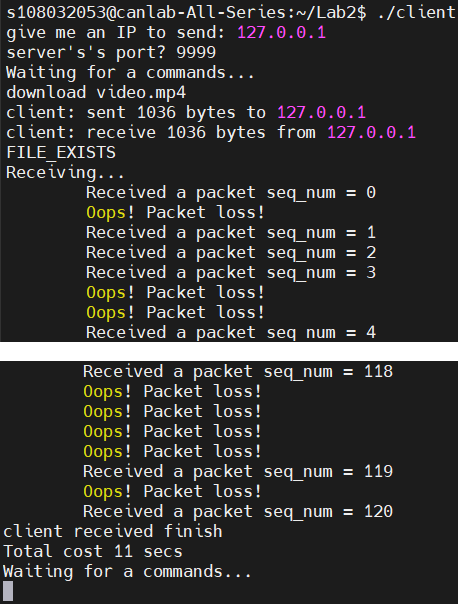
Conputer Network Lab2

學號：108032053 姓名：陳凱揚

1. 執行結果

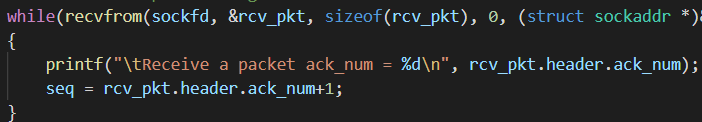
下圖1為在Linux machine上編譯、執行後的輸入、輸出結果。首先先執行server，並輸入port為9999，接著執行client，輸入IP為127.0.0.1、port為9999，就可以輸入指令 ”download video.mp4” 來下載檔案。下圖可以看到stop&wait、packet loss、timeout等功能，並在最後成功傳遞檔案，檔案也能順利開啟。

▲ 圖1

1. 程式功能
2. server.c

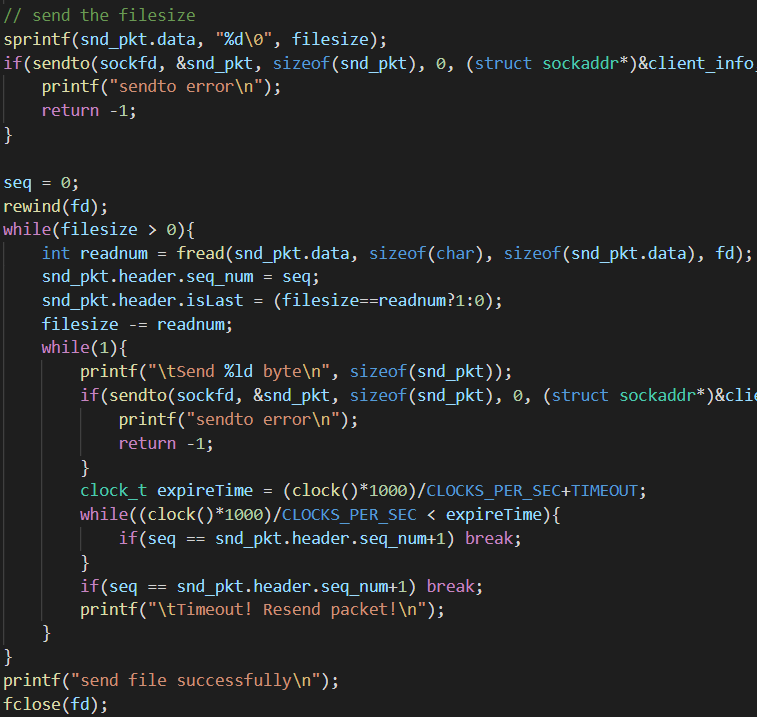
在receive\_thread()中會有個無限迴圈不斷接收封包（下圖2），並更新seq值為ack\_num+1，其中seq為gloabal variable，代表下一次要傳的封包編號。



▲ 圖2

在sendFile()中（下圖3），我首先先傳遞一個封包，儲存檔案大小，讓client先知道檔案的大小，以便決定buffer的大小。接著將seq設為0、檔案位置移回最前面，就開始一個個傳送封包。

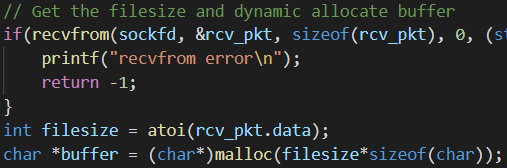
當剩餘檔案大小大於0時，會以fread()讀取資料至snd\_pkt.data、設好header，並更新剩餘檔案大小。接著以無限迴圈的方式傳送封包，每次傳送後會在100ms內檢查是否收到ack，若未收到表示發生timeout，重新傳送封包，直到成功收到ack。當所有資料傳完後，跳出迴圈且關閉檔案。



▲ 圖3

1. client.c

對應到server一開始就先傳送檔案大小的資料，在recvFile()中一開始會先接收這個封包，並動態分配對應大小的buffer（下圖4）。



▲ 圖4

首先index代表buffer存到哪裡了，接著便以無限迴圈不斷接受封包，每次接到後就模擬封包是否遺失，遺失的話直接continue重新接收封包，成功的話就將收到的資料存進buffer，而在這邊會確保不會寫超過buffer的大小，且index最大只到filesize。最後回傳ack，再檢查是不是最後一個封包，是的話就將buffer的資料以fwrite()寫進檔案裡，並結束迴圈且關閉檔案（下圖5）。



▲ 圖5