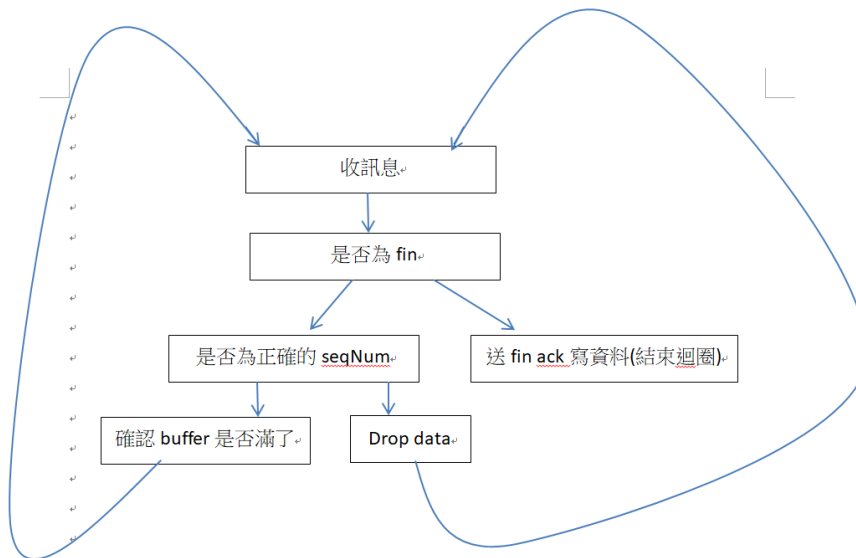
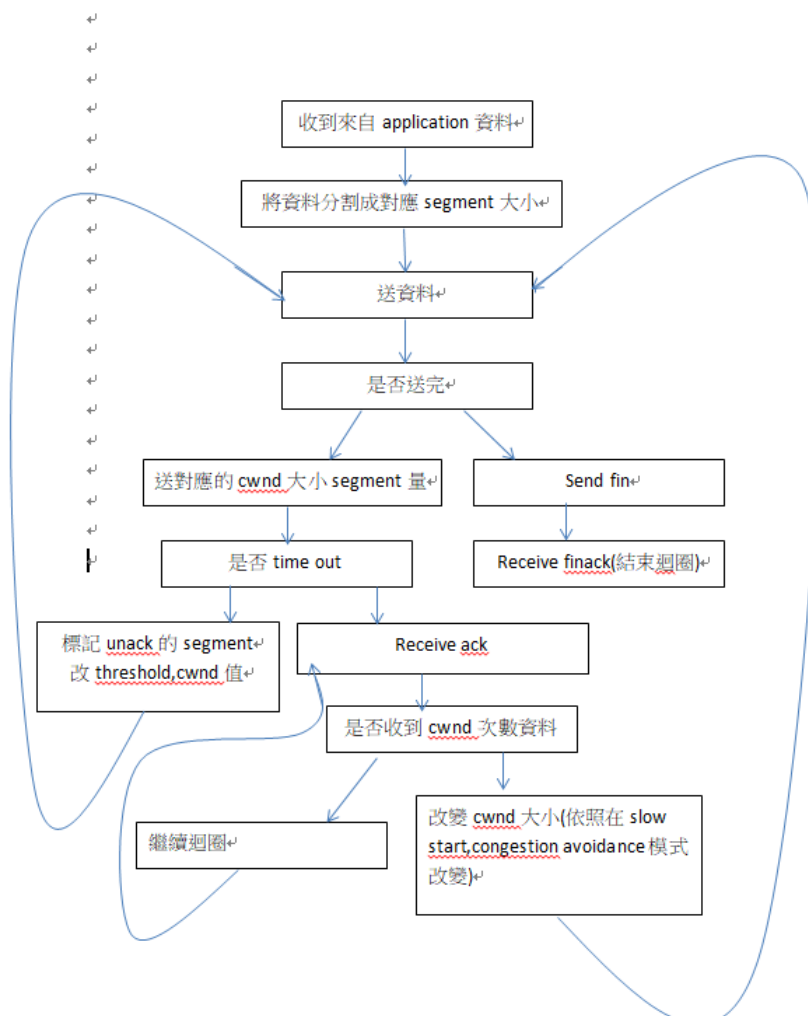


Flow Chart:

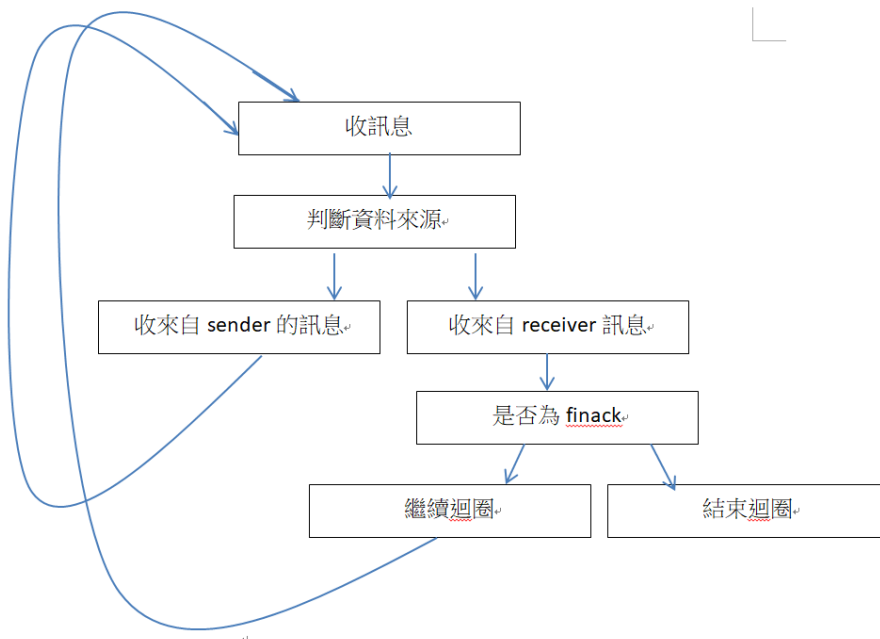
For receiver:



For sender:



For agent:



遇到的問題:

1. Sender:

- Timeout 後的處理，一開始邏輯有點寫錯會進入無限迴圈->利用變數記錄要重傳的號碼
- Timeout 的監控，一開始 function 有錯所以會不斷 timeout->用 setsockopt 進行監控
- 讀 file，用 istream 讀檔要用 binary 才能讀
- 要考量 fin 有可能會遺失的問題->在 while 迴圈中進行執行

2. Receiver:

- 要考慮收到錯誤的 seq 後要做的處理->利用變數記錄應該要收到的 seq
- 寫入檔案，要用 seek 找到上次寫完結束的位置後才能再繼續寫，不然會複寫
- 要考慮 drop data 後，也要傳回 ack

心得:

這次利用 udp 來模擬 TCP congestion control 和 reliable data transfer，一開始因為沒有頭緒所以上網找了一下資料，但是寫了之後發現還是用自己習慣的思考邏輯寫會比較順，也比較快完成，這次最大的問題是因為沒有熟悉 socket 的程式設定，所以很多語法都要查，這次學會使用傳送檔案，寫入檔案，並且完善的寫好一個系統，而在 debug 的過程中覺得最需要的還是把 pseudo code 好好地寫一次，把自己的邏輯對應跑一次，這樣比較快可以找到錯誤，系統最複雜的部份是 sender，因為 sender 是主要控制整個系統傳送的關鍵，也是這次花最久時間所寫的，這次收穫很大，很久沒有寫 c++，這次的作業順便複習以前 c++ 的語法，也更了解

網路傳送的方式!