

Introduction to Computer Network

Lab2

110030018 林芷儀

實作過程:

Server:

從助教提供的模板和 hint 逐漸加入如下功能:

1. ACK timer 和 Timeout retransmission:
 - 用 clock_t 的變數 t_begin, duration 來記錄初始時間和經過時間
 - 每次 while(1)的迴圈中都檢查一次 timeout
 - 如果 timeout 就重傳
 - 如果沒有 timeout，確認「成功送出封包」並確認「收到正確的 ack number 的封包」，都正確的話就繼續送下一份封包
2. Seq number 和影片資料讀取
 - 每次讀取 1024bute 大小的資料，放入封包的 data 區
 - 用變數 count / remain_len 計算當前剩餘的資料大小
 - 如果讀到最後一個封包，只讀入剛好大小的 data 到封包中，送完該封包後就跳出迴圈，表示傳送完成
3. 修改 udp packet 的 isLast 型態為 unsigned int。
 - 主要目的是為了讓 server sequence 為 0 的封包，把整個 video file 的檔案大小資訊傳給 client，方便 client 端計算剩餘的封包大小。
 - 雖然 isLast 本身不是用來這麼做的，但為了盡量不改動原本的架構，我覺得這是比較好的方式。

Client:

從助教提供的模板和 hint 逐漸加入如下功能:

1. 修改 udp packet 的 isLast 型態為 unsigned int，理由同上
2. 用變數 rcv 確認收到的封包是正常的
3. 收到之後，用 isLoss 來模擬 packet loss
4. 如果沒有封包遺失，用變數 receive_packet 來確認收到的是正確的 seq number 的封包。如果正確就寫入 file 中，並回傳 Ack 給 server 端。
5. 封包傳送過程中不斷更新變數 getFilesize，並依此決定讀取資料的大小。
6. 若從 server 端拿來的封包發現是最後一個(is_last flag==1)，只讀入剛好大小的資料(藉由 getFilesize 變數)，並跳出迴圈，代表完成傳輸。

編譯及執行方式:

用 makefile: make

實作結果:

<pre>daphne61221@daphne61221-VirtualBox:~/桌面/ICNlab2\$./server 9999 ====Parameter==== Server's IP is 127.0.0.1 Server is listening on port 9999 ===== server waiting.... process command.... filename is video.mp4 FILE_EXISTS server: sent 1036 bytes to 127.0.0.1 transmitting... Send 1036 byte Receive a packet ack_num = 0 Send 1036 byte</pre>	<pre>server's port? ^[AZ [1]+ 已停止 ./client daphne61221@daphne61221-VirtualBox:~/桌面/ICNlab2\$./client give me an IP to send: 127.0.0.1 server'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</pre>
<pre>Timeout! Resend packet! Send 1036 byte Timeout! Resend packet! Send 1036 byte Timeout! Resend packet! Send 1036 byte Timeout! Resend packet! Send 1036 byte Receive a packet ack_num = 119 Send 563 byte Timeout! Resend packet! Send 563 byte Receive a packet ack_num = 120 send file successfully server waiting.... []</pre>	<pre>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! Oops! Packet loss! Receive a packet seq_num = 119 Oops! Packet loss! Receive a packet seq_num = 120 client received finished Total cost 11 secs Waiting for a commands... []</pre>