HW2-Report

B05902050 黃子源

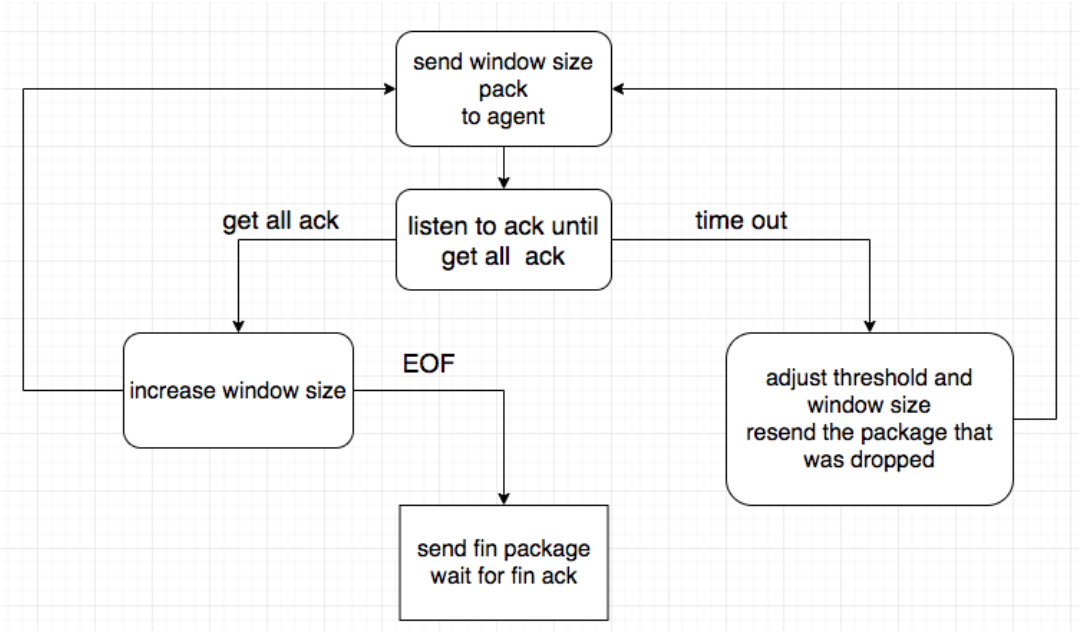
I use python to done the work.

For sender.py, there is one argument filename, which represent the file that wanted to be sent to receiver. So just simply run the ‘python sender.py [filename]’ command and we can run the sender. In addition, I set sender’s address to ‘127.0.0.1’ and port 8887, default threshold 16, timeout 1.

For receiver.py, there is also one argument filename, which represent the file that wanted to be written. Run the command ‘python receiver.py [filename]’ to run the receiver, and I set the buffer size as 32 segments.

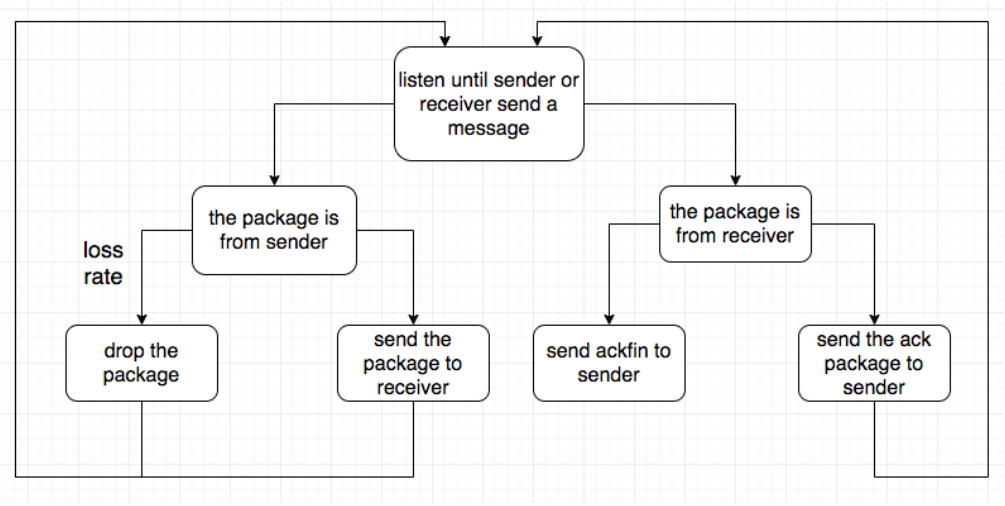
Program structure:

1. Sender:

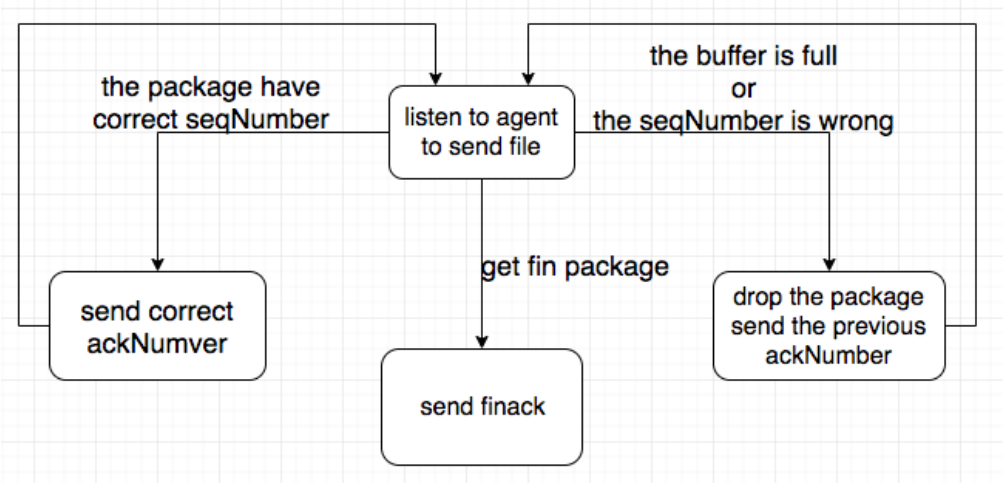


I read 1000 characters as data and sent number of window size packages at a time. Then I use select to listen to the socket, if we get all ack package, we increase the window size and send more data to receiver; on the other hand, if the transport time is larger than 1 second, then I decrease the window size and retransfer the package from the sequence number that didn’t get ack until all the packages were get by receiver.

2. Agent:



3. Receiver:



The receiver will wait for the package and send ack package back if the sequence number is correct, and it will drop the package if the number is wrong or its buffer is full. Receiver will continuously receiving packages until it get the fin package.

Difficulties and Solutions:

用 python 實作的最大問題就是 python 對變量類型的約束較小，但這個抽象的方式使得 socket 要控制變量長度變得很困難。所以我用 struct.pack()函式將不同變數打包成字節串，一起傳出後再用 receiver 的 struct.unpack()來得到裡面的資訊。如果只是單純 encode 成 utf-8 字串不知道為什麼 agent 會無法順利讀取 header 真正的值。

至於 timeout 的部分我用上次的 select 函式來實作，相較於用 signal 等方式簡便很多。