

NSCAP hw5-report

姓名：杜峯

學號：109550096

1. Explain how you implement error control: 5%
 - a. 針對每一個stream的packet都會需要reciever傳送一個ack回sender
 - b. 在接收端每一個stream會有一個ack_dict()用來儲存每一個packet發送出去的時間，同時開啟一個thread來對每一個dict裡面的元素計算是否已經超過設定的timer時間，如果超過，且receiver還沒傳回ack那就重新傳送這個packet
 - c. 如果sender收到ack那就會對stream更新這個封包的狀態為“已收到ack”，在計算time_out的thread就不會去計算這個封包的time_out了
 - d. 在receiver端，會針對每一個stream_id開啟一個thread處理是否已經收完全部的packet，如果收完，那就會對這個buffer裡面的list進行sort，並將排列好的正確data傳送到一個buffer裡面，如果call了recv()，且buffer裡面有完整的数据，那就會回傳這個完整的数据，若是有多筆已經排列好的完整data，那就會取出第一個。
2. Explain how you implement flow control: 5%
 - a. 在receiver端的buffer收取packet的時候會同時計算還剩下多少的空間可以接收新的packet，計算完之後會利用ack的packet一起將flow_window_size送回去給sender，讓sender進行“可以送多少packet”的數量調整
3. Explain how you implement congestion control: 5%
 - a. 在sender端會有一個sliding的滑動窗口，窗口內的packet都是可以傳送的，並不會因為傳送一個packet就block下來等待ack
 - b. window滑動的條件是，窗口內最小編號的packet已經正確的收到ack，這個時候window會這個向後滑動一個packet的單位。
 - c. 如果sender在recv_ack的時候連續收到了三個同樣的ack_next（下一個想要哪一個編號的packet）那就會縮減window_size，若是很順利的一直受到持續前進的ack那就會增加window知道設定的初始window_size
4. If you use two streams to send data simultaneously from the client to the server or in the other direction, what will happen if one packet of a stream gets lost? Is the behavior of QUIC different from that of TCP? Why? 10%
 - a. 當兩個stream同時在QUIC中傳輸數據時，每個stream的packet都使用獨立的傳輸控制來調整傳輸窗口大小。如果一個stream的packet丟失了，QUIC會重傳該packet，並繼續處理另一個流的packet，這樣就能實現stream之間的simultaneous sent。
 - b. 若是TCP，如果一個packet loss了，TCP會將所有未被確認的packet都視為已經loss，並將所有packet進行重傳。這意味著，即使另一個stream的packet已經到達了receiver，TCP也不會將其傳送給應用程序，直到loss的packet被完全重傳並接收到為止。
 - c. QUIC是每一個stream獨立控制傳輸，即使loss，其他stream依然可以繼續work，而TCP則是視為一個整體，一個loss會使得整體被block住。