

Computer Network HW2 Report

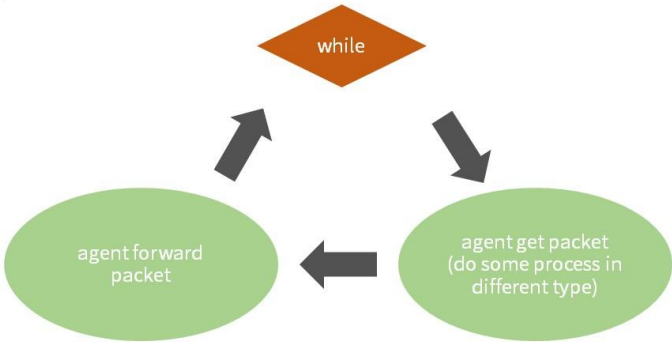
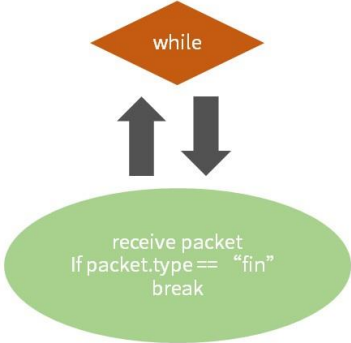
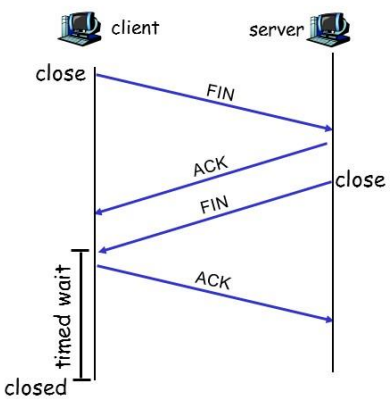
B04705003 資工三 林子雋

1. How to execute your program

建議先開啟 receiver.py，再開啟 agent.py，最後開啟 sender.py	
sender.py	python3 sender.py <file_path> [--IP IP] [--send_port SEND_PORT] [--recv_port RECV_PORT] [--threshold THRESHOLD]
agent.py	python3 agent.py [--IP IP] [--data_port DATA_PORT] [--ack_port ACK_PORT] [--sender_port SENDER_PORT] [--receiver_port RECEIVER_PORT] [--loss_prob LOSS_PROB]
receiver.py	python3 receiver.py <file_path> [--IP IP] [--recv_port RECV_PORT] [--agent_port AGENT_PORT]

2. Explain your program structure

Program structure visualization	<p>Program Structure</p> <pre>class Sender: self.packets self.sock_send class Agent: # counter self.data_counter = 0 self.drop_counter = 0 self.sock_data_from self.sock_data_forward self.sock_ack_forward self.sock_ack_from class Receiver: # buffer and data self.buf_size = 32 self.buf_list = [] self.file = b'' self.sock_recv self.sock_send</pre> <p>Go-Back-N: <code>self.base = 0</code>, <code>self.nextseqnum = 0</code>, <code>self.congestion_window = 1</code></p> <p>Loss probability: <code>self.loss_prob = args.loss_prob</code></p>
sender.py flowchart	<p>Sender Flowchart</p> <pre>graph TD W1{while} --> B1[if there is packet in the socket, receive it else break] B1 --> W1 W1 --> W2{while} W2 --> B2[if next sequence number < base + congestion_window: Send packet else break] B2 --> W2 B2 --> C[check_timeout() If time out, reset congestion window] C --> W1 C --> W2</pre>

agent.py flowchart	<h2 style="text-align: center;">Agent Flowchart</h2>  <pre> graph TD while{while} --> get_packet([agent get packet (do some process in different type)]) get_packet --> forward_packet([agent forward packet]) forward_packet --> while </pre>
receiver.py flowchart	<h2 style="text-align: center;">Receiver Flowchart</h2>  <pre> graph TD while{while} --> receive_packet([receive packet If packet.type == "fin" break]) receive_packet --> while </pre>
fin, finack control flow :	<h3 style="text-align: center; color: blue;">TCP Connection Management (cont.)</h3> <p><u>Closing a connection:</u></p> <p>client closes socket: <code>clientSocket.close()</code> ;</p> <p>Step 1: client end system sends TCP FIN control segment to server</p> <p>Step 2: server receives FIN, replies with ACK. Closes connection, sends FIN.</p>  <pre> sequenceDiagram participant client participant server Note over client: close client->>server: FIN server-->>client: ACK Note over server: close server->>client: FIN client-->>server: ACK Note over client: closed Note over server: closed </pre> <p style="text-align: right;">Transport Layer 3-78</p>

3. Difficulties and Solutions

1. 在 sender.py 中，Go-Back-N 加上 congestion window 的實作比較麻煩一些，所以花了不少時間在看 sender 的輸出，才把小錯誤除掉
2. 架構各個 py 檔 class 花了一些時間，不過這樣讓版面更簡潔