

Comp3331 assignment1 report (python3)

Program design:

1. Data structure

As we are sending the STP segment which contains "type", "seqno", and "data". In both sender and receiver file, we create encode and decode function to store STP segment data structure to transform between bytes and string.

2. Sender

In sender file we start with *sender_handshake(self, rot)* function. The sender will send the segment that contains SYN to the sender to start the first way handshake. The sender will wait for receiver to reply a SYN ACK, when receive it then the connection between sender and receiver creates. If the sender didn't receive the SYN ACK within rot it will retransmit the segment at most 3 times. If still not receiving the SYN ACK, the sender will send out the segment contains RESET then exit the program(close connection in sender side).

The second stage we sent DATA and FIN which implements in *ptp_send(self)*. We mainly focus on sending message (including retransmission, timestamp). The sliding window is also implemented in this stage, we use the current receiving ack sequence number and the current packet sequence number that we are sending out to determine the remain window size. We also determine whether we have received data ACK and FIN ACK from the sender in the main thread.

3. Receiver

The receiver starts with handshake stage. As receiver will receive segments with different type which includes the SYN ACK for building handshake, we define a function *receive(self)* to deal with it. While we receive the segment, in *receive* function we will first deal with the possibility rate of losing packets (using uniform distribution). We check the type of the segment, then send out the ack by *sent* function. The sent function will also deal with the possibility rate of losing ACK.

Design trade-offs:

The previous version of using multi-threading cause many problems. For instance, the program is easily stuck when both thread wants to get the same data. And when closing the main thread but the sub-thread is still active even setting sub-thread inactive in the main thread code. As a result, we are only using main thread to control the whole sender program. Second, as what mentioned above, since receiver receives different types of segment from Sender, it's easier to implementing all receiving process in one function, and send the ACK right through receiving the message.