**#411**

In the original approach, to track the which packet have received ACK, I add a data member called “tmp\_ack” into the pkc struct. Obviously, it failed since the web submission system has its own emulator.

**#412**

1. I delete all emulator files which are not needed for submission and unnecessary print comment in the code
2. From the test result of #411, I only resend the buffer.[windowfirst] packet if timeout (A\_timeinterrupt)
3. To remember which packet has received ACK, my approach is changing the seq. no. to -1 if the packet has received ACK.
4. I add a global variable ‘reserve\_count’ to remember the number of packets (except the windowfirst packet) which have received ACK. In #411 version, I forget to add ‘break’ in the for loop incrementing ‘reserve\_count’ when meeting -1 in buffer, so the incremented number is false, now I add back the break

**#413**

In SR, in receive side it only send the correctly received data to the application once. So I created an int array ‘rev\_buffer’ to remember data have sent to application already

**#414**

#413 failed. Because the array was initialized as all 0, which will conflict with packet 0, so I created a variable ‘first\_round’ to mark the start of program, when the program starts, the rev\_buffer will be initialized with all ‘-1’. And sizeof(rev\_buffer) can’t get the correct array length, so I used SEQSPACE instead.

#**415**

From the test result, the timer shall stop and restart when the first awaiting packet received packet.

**#416**

#415 failed since it should only restart when windowcount > 0

**#417**

#416 failed since the stop timer should not be included in the if condition ‘windowcount > 0’. So I nested ‘windowcount > 0’ inside ‘packet.no == seqfirst’ condition to run in a correct logic

**#419**

Despite correction in #414, I still cannot pass test 3. I print a comment “SEND TO APP” in each sent to application to figure out where is the bug.

#**420**

As change made in #415, I created a variable ‘data\_existed’ to check if a packet is already sent to application. And I will add first received packet no. to rev\_buffer to update the record. I didn’t include this update to ‘data\_existed == 0’(means the data is first received) condition, so I failed in #419. Now, I put this update back to ‘data\_existed == 0’ condition.

For test 8, I failed to the bug in of code about sliding window. I only increment reserve\_ackcount by 1 if the packet after windowfirst received ACK, and only when reserve\_ackcount is 1 I increment it by 1 if meets each ‘-1’. It is wrong. I can do it simpler and correct by checking all ‘-1’ after receiving first packet to decide how to slide the window.

**#421**

1. Delete “SEND TO APP” comment
2. I created a variable called ‘last\_ACK’ to remember the last received packet, so the program can determine whether the packet is duplicate and print the message ‘A:duplicate ACK received, do nothing!’.