

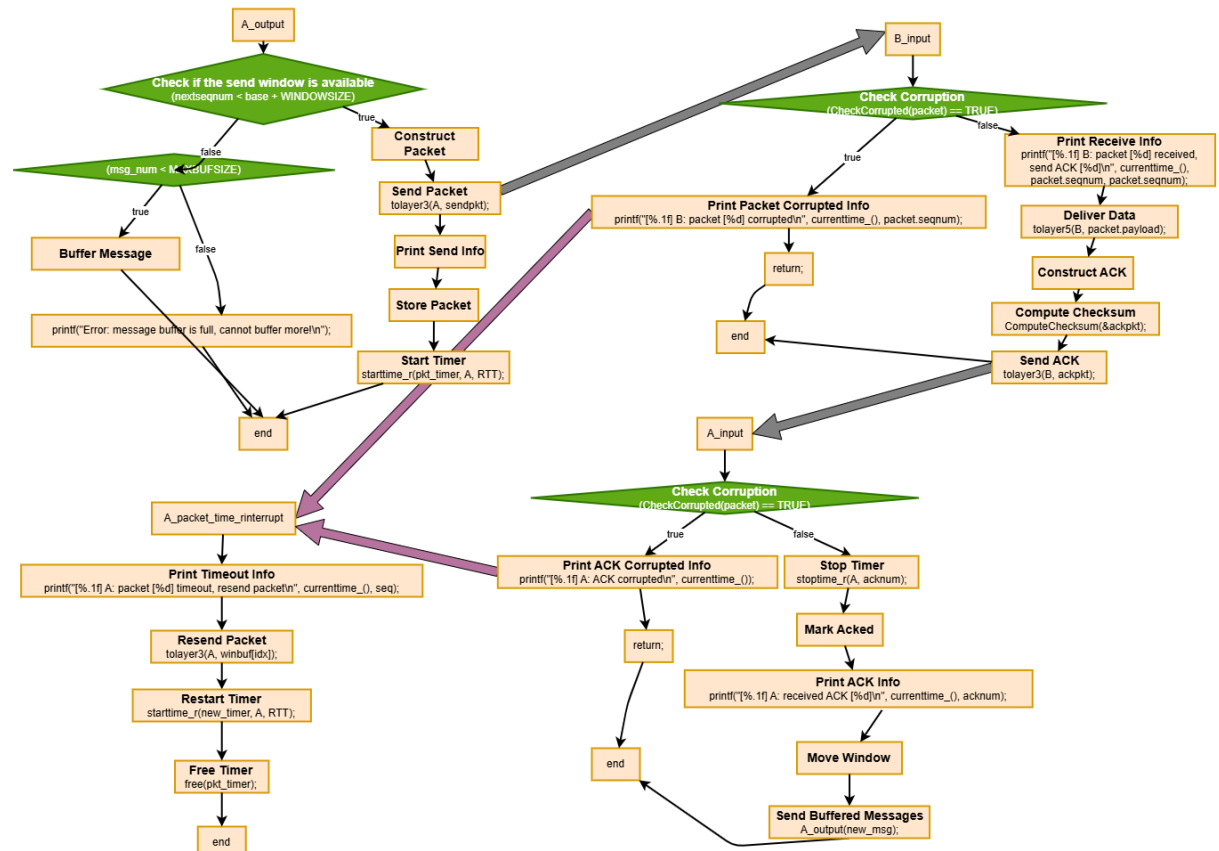
# IERG3310 Lab1 Report

SU Caiyi 1155191405

## I. The environment for compiling my codes

```
PS C:\Users\苏采奕> gcc --version
gcc.exe (MinGW.org GCC Build-2) 9.2.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

## II. Flow chart of *A\_output*, *A\_input*, *B\_input* and *A\_packet\_time\_rinterrupt*



## III. Problems met in the implementation and corresponding solutions

**Problem:** After my code can successfully test the two examples given, I tried to test other Enter channel pattern strings. I found that if I set the Enter channel pattern string input to put x and - at the beginning of the information transmission, such as "xxxx-----oooooooooooo", my output will show that my packet cannot be sent out completely (as shown in the figure below)

```

*****
IERG3310 lab1: Reliable Data Transfer Protocol-SR, implemented by 1155191405
*****
Enter the number of messages to simulate:
9
Enter time_ between messages from sender's layers [ > 0.0]:
2
Enter channel pattern string
xxxx----oooooooooooo
Enter sender's window size
3
*****
***** Packet Transmission Log *****
*****
[2.0] A: send packet [0] base [0]
[4.0] A: send packet [1] base [0]
[6.0] A: send packet [2] base [0]
[8.0] A: buffer packet [3] base [0]
[10.0] A: buffer packet [3] base [0]
[12.0] A: buffer packet [3] base [0]
[14.0] A: buffer packet [3] base [0]
[16.0] A: buffer packet [3] base [0]
[17.0] A: packet [0] time_out, resend packet
[18.0] A: buffer packet [3] base [0]
[19.0] A: packet [1] time_out, resend packet
[21.0] A: packet [2] time_out, resend packet
[25.5] B: packet [999999] corrupted
[27.5] B: packet [999999] corrupted

[32.0] A: packet [0] time_out, resend packet
[34.0] A: packet [1] time_out, resend packet
[36.0] A: packet [2] time_out, resend packet
[38.5] B: packet [999999] corrupted
[40.5] B: packet [999999] corrupted
[42.5] B: packet [2] received, send ACK [2]
[47.0] A: packet [0] time_out, resend packet
[49.0] A: packet [1] time_out, resend packet
[49.0] A: received ACK [2]
[53.5] B: packet [0] received, send ACK [0]
[55.5] B: packet [1] received, send ACK [1]
[60.0] A: received ACK [0]
[60.0] A: send packet [3] base [1]
[62.0] A: received ACK [1]
[62.0] A: send packet [4] base [2]
[66.5] B: packet [3] received, send ACK [3]
[68.5] B: packet [4] received, send ACK [4]
[73.0] A: received ACK [3]
[75.0] A: received ACK [4]
Simulator terminated.
*****
***** Packet Transmission Summary *****
*****
From Sender to Receiver:
total sent pkts: 13
total correct pkts: 5
total resent pkts: 8
total lost pkts: 4
total corrupted pkts: 4
the overall throughput is: 17.067 Kb/s
*****
请按任意键继续

```

**Solution:** After I carefully checked the code and the principle of SR, I found that it was because the A side only base++ when acknum == base (the window only moves one packet at a time), but did not continue to check whether the subsequent sequence number was also ACKed, or did not update the status of the corresponding window cache in the "out-of-order" scenario.

The correct approach should be:

- ◆ When A\_output() sends a new packet, put it into the send window array winbuf[i] and write the status:
  - Sent + Waiting for ACK.
- ◆ When A\_input() receives a certain ACK=acknum:
  - Find the item with seq=acknum in the send window and mark it as "ACKed".
  - If acknum == base, move base forward, but not just base++, but continuously check: Is the next sequence number marked as ACKed? If so, continue ++ until you encounter a seq that has not been ACKed.
  - When the window moves forward, if there are still cached messages (in buffer[]) and there is space in the current window, continue to send new packets.

After my modification, the input of "xxxx----oooooooooooooooo" can be transmitted completely, but when I re-enter the two examples given, I find that the base of the packet at the 32nd second in the output of the second example is different from my output. I checked SR's interactive animation and it was the same as my output. I immediately asked TAs and the professor and got a positive answer.