

Computer Networks (Section A)

Assignment -2

Ritisha Singh 2021089 (CSE)

Question 2, 3, 4:

- On running the **make** command in my wsl:

```
leif@leif:/mnt/c/Users/leifo/Downloads/Assignment/CN_Assignment2_2021089/CN_Assignment2_2021089/assignment2/assignment2/build$ make
[ 23%] Built target tcp_reciever
[ 26%] Built target tcp_reciever_checks
[ 30%] Built target wrapping_integers_cmp
[ 33%] Built target wrapping_integers_unwrap
[ 36%] Built target wrapping_integers_wrap
[ 39%] Built target wrapping_integers_roundtrip
[ 42%] Built target byte_stream_construction
[ 46%] Built target byte_stream_one_write
[ 49%] Built target byte_stream_two_writes
[ 52%] Built target byte_stream_capacity
[ 55%] Built target byte_stream_many_writes
[ 58%] Built target recv_connect
[ 61%] Built target recv_transmit
[ 65%] Built target recv_window
[ 68%] Built target recv_reorder
[ 71%] Built target recv_close
[ 74%] Built target recv_special
[ 77%] Built target fsm_stream_reassembler_cap
[ 80%] Built target fsm_stream_reassembler_single
[ 84%] Built target fsm_stream_reassembler_seq
[ 87%] Built target fsm_stream_reassembler_dup
[ 90%] Built target fsm_stream_reassembler_holes
[ 93%] Built target fsm_stream_reassembler_many
[ 96%] Built target fsm_stream_reassembler_overlapping
[100%] Built target fsm_stream_reassembler_win
```

- To check my code against all the test cases present, I ran the command:
ctest

```

leif@leif:/mnt/c/Users/leifo/Downloads/Assignment/CN_Assignment2_2021089/CN_Assignment2_2021089/assignment2/assignment2/build$ ctest
Test project /mnt/c/Users/leifo/Downloads/Assignment/CN_Assignment2_2021089/CN_Assignment2_2021089/assignment2/assignment2/build
  Start 1: wrapping_integers_cmp ..... Passed    0.02 sec
1/23 Test #1: wrapping_integers_cmp ..... Passed    0.02 sec
  Start 2: wrapping_integers_unwrap ..... Passed    0.01 sec
2/23 Test #2: wrapping_integers_unwrap ..... Passed    0.01 sec
  Start 3: wrapping_integers_wrap ..... Passed    0.01 sec
3/23 Test #3: wrapping_integers_wrap ..... Passed    0.01 sec
  Start 4: wrapping_integers_roundtrip ..... Passed    0.25 sec
4/23 Test #4: wrapping_integers_roundtrip ..... Passed    0.25 sec
  Start 5: byte_stream_construction ..... Passed    0.01 sec
5/23 Test #5: byte_stream_construction ..... Passed    0.01 sec
  Start 6: byte_stream_one_write ..... Passed    0.02 sec
6/23 Test #6: byte_stream_one_write ..... Passed    0.02 sec
  Start 7: byte_stream_two_writes ..... Passed    0.01 sec
7/23 Test #7: byte_stream_two_writes ..... Passed    0.01 sec
  Start 8: byte_stream_capacity ..... Passed    0.62 sec
8/23 Test #8: byte_stream_capacity ..... Passed    0.62 sec
  Start 9: byte_stream_many_writes ..... Passed    0.01 sec
9/23 Test #9: byte_stream_many_writes ..... Passed    0.01 sec
  Start 10: recv_connect ..... Passed    0.02 sec
10/23 Test #10: recv_connect ..... Passed    0.02 sec
  Start 11: recv_transmit ..... Passed    0.12 sec
11/23 Test #11: recv_transmit ..... Passed    0.12 sec
  Start 12: recv_window ..... Passed    0.02 sec
12/23 Test #12: recv_window ..... Passed    0.02 sec
  Start 13: recv_reorder ..... Passed    0.02 sec
13/23 Test #13: recv_reorder ..... Passed    0.02 sec
  Start 14: recv_close ..... Passed    0.03 sec
14/23 Test #14: recv_close ..... Passed    0.03 sec
  Start 15: recv_special ..... Passed    0.01 sec
15/23 Test #15: recv_special ..... Passed    0.01 sec
  Start 16: fsm_stream_reassembler_cap ..... Passed    0.12 sec
16/23 Test #16: fsm_stream_reassembler_cap ..... Passed    0.12 sec
  Start 17: fsm_stream_reassembler_single ..... Passed    0.02 sec
17/23 Test #17: fsm_stream_reassembler_single ..... Passed    0.02 sec
  Start 18: fsm_stream_reassembler_seq ..... Passed    0.01 sec
18/23 Test #18: fsm_stream_reassembler_seq ..... Passed    0.01 sec
  Start 19: fsm_stream_reassembler_dup ..... Passed    0.02 sec
19/23 Test #19: fsm_stream_reassembler_dup ..... Passed    0.02 sec
  Start 20: fsm_stream_reassembler_holes ..... Passed    0.01 sec
20/23 Test #20: fsm_stream_reassembler_holes ..... Passed    0.01 sec
  Start 21: fsm_stream_reassembler_many ..... Passed    8.30 sec
21/23 Test #21: fsm_stream_reassembler_many ..... Passed    8.30 sec
  Start 22: fsm_stream_reassembler_overlapping ..... Passed    0.02 sec
22/23 Test #22: fsm_stream_reassembler_overlapping ..... Passed    0.02 sec
  Start 23: fsm_stream_reassembler_win ..... Passed    8.12 sec
23/23 Test #23: fsm_stream_reassembler_win ..... Passed    8.12 sec

100% tests passed, 0 tests failed out of 23

Total Test time (real) = 18.23 sec
leif@leif:/mnt/c/Users/leifo/Downloads/Assignment/CN_Assignment2_2021089/CN_Assignment2_2021089/assignment2/assignment2/build$

```

Edge cases or Assumption:

1. For part 2:

- Whenever the data packets value got beyond window, on either of the sides, i kept only the values lying within the window.
- If a data packet has already been written we will drop it.
- If a data packet arrives as an empty string, instead of increasing the `ack_index_` with the data's size, we will increment it with 1.
- Even if the expected `ack_index_` arrives we won't store it in buffer or write it to the stream, unless we have space left for it, i.e. `remaining capacity > 0`.

