

CN Assignment 2 Part II and III

Manav Mittal
2021538
CSAI

Reassembler

I have used vector for the buffer implementation where I store the bytes in the order of their sequence number. And the zeroth index of the vector is mapped to the `ack_index`. When the bytes are pushed in the byte stream, the remaining segments are moved to the front of the vector. The bytes in the byte stream are written according to the remaining capacity in the byte stream and the longest continuous segment available in the buffer. For handling the EOF if the buffer is empty and I have got the EOF, then I ended the input of the byte stream. If I already got EOF and the segment received has index greater than the EOF segment then I ignored it and ended the input to the byte stream.

```
Test project /mnt/ManavMittal/CN_Assignments/A2/assignment2/build
Start 16: fsm_stream_reassembler_cap
1/8 Test #16: fsm_stream_reassembler_cap ..... Passed    0.06 sec
Start 17: fsm_stream_reassembler_single
2/8 Test #17: fsm_stream_reassembler_single ..... Passed    0.01 sec
Start 18: fsm_stream_reassembler_seq
3/8 Test #18: fsm_stream_reassembler_seq ..... Passed    0.24 sec
Start 19: fsm_stream_reassembler_dup
4/8 Test #19: fsm_stream_reassembler_dup ..... Passed    0.01 sec
Start 20: fsm_stream_reassembler_holes
5/8 Test #20: fsm_stream_reassembler_holes ..... Passed    0.01 sec
Start 21: fsm_stream_reassembler_many
6/8 Test #21: fsm_stream_reassembler_many ..... Passed    1.39 sec
Start 22: fsm_stream_reassembler_overlapping
7/8 Test #22: fsm_stream_reassembler_overlapping ... Passed    0.00 sec
Start 23: fsm_stream_reassembler_win
8/8 Test #23: fsm_stream_reassembler_win ..... Passed    1.20 sec

100% tests passed, 0 tests failed out of 8

Total Test time (real) = 2.93 sec
```

Wrapping Integer

To unwrap 32 bit number to 64 bit I have first wrapped the checkpoint with the help of `isn` and the provided `wrap` function. After that I have calculated the difference between the wrapped checkpoint and the provided wrapped `seq_no`. it give by how much margin the 64bit sequence number is greater than the check point. If the margin comes out to be negative then we have to add 2^{32} to the final answer after adding margin to the checkpoint because it means that now due to wrap around our 32 bit `seq_no` has become smaller than the wrapped checkpoint. And to ensure the

current seq_no is greater we have to add 2^{32} since we are taking the modulo by 2^{32} .

```
total test time (real) = 2.99 sec
0 /mnt/ManavMittal/CN_Assignments/A2/assignment2/build ctest -R '^wrapping_integer' --output-on-failure
Test project /mnt/ManavMittal/CN_Assignments/A2/assignment2/build
  Start 1: wrapping_integers_cmp
1/4 Test #1: wrapping_integers_cmp ..... Passed    0.00 sec
  Start 2: wrapping_integers_unwrap
2/4 Test #2: wrapping_integers_unwrap ..... Passed    0.00 sec
  Start 3: wrapping_integers_wrap
3/4 Test #3: wrapping_integers_wrap ..... Passed    0.00 sec
  Start 4: wrapping_integers_roundtrip
4/4 Test #4: wrapping_integers_roundtrip ..... Passed    0.23 sec

100% tests passed, 0 tests failed out of 4

Total Test time (real) = 0.24 sec
```

TCP Reciever

Hints were already given to get the checkpoint, abs_seq_no of the current segment and since we also have syn packets, then we have to shift the sequence number by 1. Now if the syn is received, then I have to store the seq_no of the syn packet as isn. If the syn packet is received after a syn packet has already been received, then the duplicate syn packets are ignored. After the syn is received, the payload is pushed in the resassembler, and if the fin flag is also set in the packet, then the string is pushed with EOF. Then the expected_ack is generated, and the seq number is shifted according to whether the fin and syn are received or not. Then this expected ack is wrapped to 32 bits using the wrap function this is the ack we have to send to the sender.

```
total test time (real) = 4.21 sec
0 /mnt/ManavMittal/CN_Assignments/A2/assignment2/build ctest -R '^recv' --output-on-failure
Test project /mnt/ManavMittal/CN_Assignments/A2/assignment2/build
  Start 10: recv_connect
1/6 Test #10: recv_connect ..... Passed    0.00 sec
  Start 11: recv_transmit
2/6 Test #11: recv_transmit ..... Passed    0.63 sec
  Start 12: recv_window
3/6 Test #12: recv_window ..... Passed    0.00 sec
  Start 13: recv_reorder
4/6 Test #13: recv_reorder ..... Passed    0.00 sec
  Start 14: recv_close
5/6 Test #14: recv_close ..... Passed    0.00 sec
  Start 15: recv_special
6/6 Test #15: recv_special ..... Passed    0.00 sec

100% tests passed, 0 tests failed out of 6

Total Test time (real) = 0.64 sec
```