Lab 3 Writeup

My name: 王聪颖

My student ID: 211220180

This lab took me about 8 hours to do. I did attend the lab session.

- **Program Structure and Design of the TCPSender**: 新增Timer类定义如下：

```
class Timer {
  private:
    uint32_t _accumulation_time = {0};
    uint32_t _timeout = {0};
    bool _flag_running = {false};

  public:
    Timer(){};
    bool expired() const { return _flag_running && _accumulation_time >= _timeout; }
    bool running() const { return _flag_running; }
    void elapse(uint32_t interval) {
        if (_flag_running)
            _accumulation_time += interval;
    }
    void set_timeout(uint32_t timeout) { _timeout = timeout; }
    void start(uint32_t timeout) {
        _flag_running = true;
        _accumulation_time = 0;
        set_timeout(timeout);
    }
    void restart(uint32_t timeout) { start(timeout); }
    void stop() { _flag_running = false; }
};
```

TCPSender类定义中，新增的 成员变量/函数 如下： 在本实现中，采取了使_swnd最小值为1的trick，即_swnd>=1; 而rwnd=0的情况，采用_flag_rwnd_zero进行区分； 如此实现的好处是: 无论rwnd=0 or rwnd>0，TCPSender均能保持发送segment（如果swnd有空间的话，且segment.length_in_sequence_space>=1）；

```
// begin new added private member
    std::queue<TCPSegment> _segments_outstanding{};
    Timer _timer{};
    unsigned int _RTO;
    unsigned int _consecutive_retransmissions{0};
    uint16_t _swnd{1};  // sender window
    size_t _bytes_in_flight{0};
    bool _flag_SYN{false};        // whether SYN has been sent
    bool _flag_FIN{false};        // whether FIN has been sent
    bool _flag_rwnd_zero{false};  // whether rwnd == 0
    // send a segment that has been loaded(seqno,[syn],[fin],payload)
    void send_segment(TCPSegment &seg);
    size_t swnd_remain_space();
// end new added private member
```

封装：发送一个装载完毕（seqno,[syn],[fin],payload) 的send_segment函数，减少代码重复，提高可读性

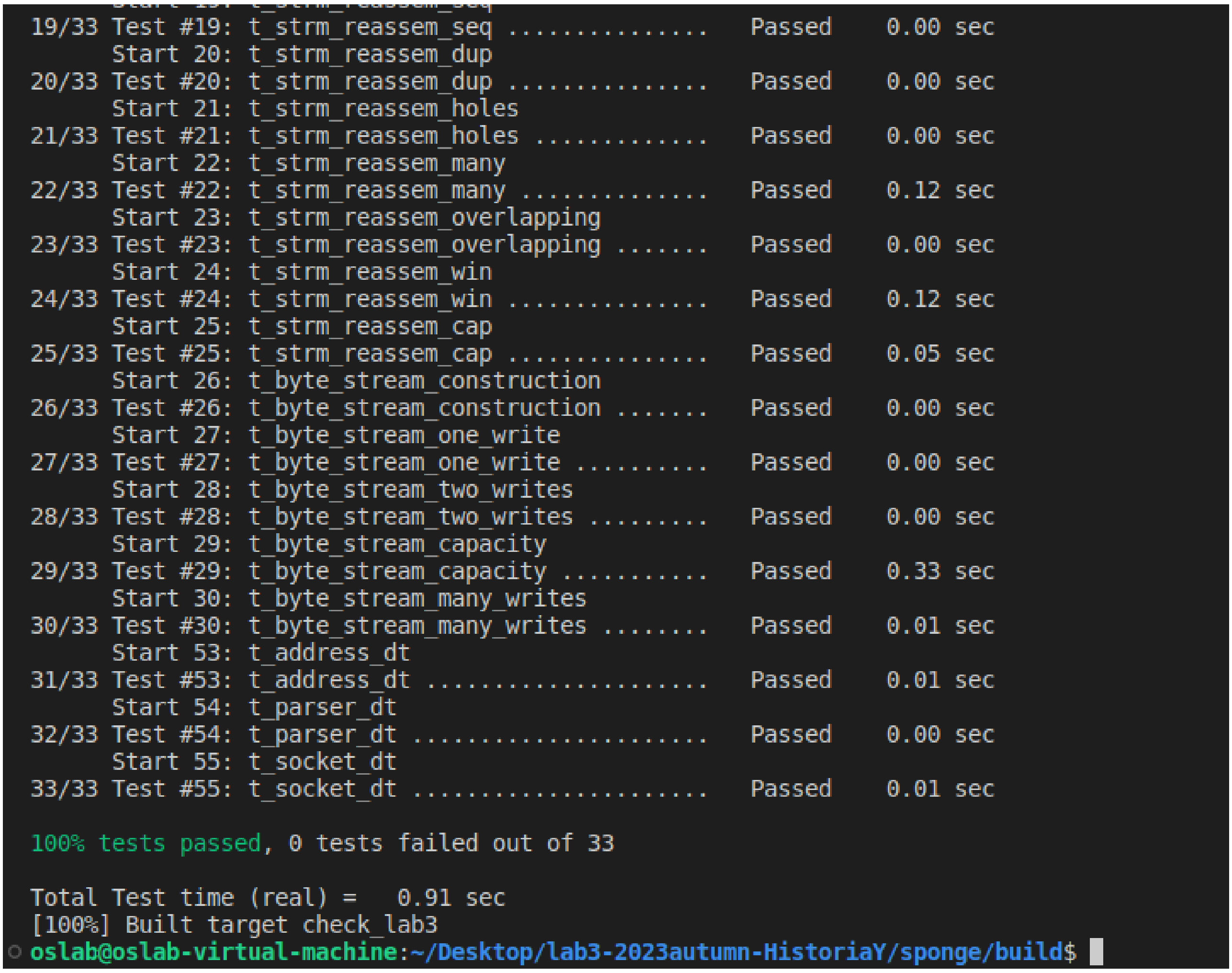
```
void TCPSender::send_segment(TCPSegment &seg) {
    _segments_out.push(seg);
    _segments_outstanding.push(seg);
    _next_seqno += seg.length_in_sequence_space();
    _bytes_in_flight += seg.length_in_sequence_space();
    if (!_timer.running())
        _timer.start(_RTO);
}
```

- **Implementation Challenges:** 实验过程中，遇到并解决过两个bug（如下，均由注释指出）：

```
void TCPSender::ack_received(const WrappingInt32 ackno, const uint16_t window_size) {
    // what if window_size<new(_bytes_in_flight)??????
    uint64_t abs_ackno = unwrap(ackno, _isn, _next_seqno);
    // there was once a bug
    // TCPSenderTestHarness test{"Impossible ackno (beyond next seqno) is ignored", cfg};
    if (abs_ackno > _next_seqno)
        return;
    ......
}
```

```
// there was once a bug, in case: _swnd < _bytes_in_flight
size_t TCPSender::swnd_remain_space() { return (_swnd >= _bytes_in_flight) ? _swnd - _bytes_in_flight : 0; }
```

- **Remaining Bugs**: 本次实验测试样例全部通过，无遗留bug



```
        Start 19: t_strm_reassem_seq
 19/33 Test #19: t_strm_reassem_seq ...............   Passed    0.00 sec
       Start 20: t_strm_reassem_dup
 20/33 Test #20: t_strm_reassem_dup ...............   Passed    0.00 sec
       Start 21: t_strm_reassem_holes
 21/33 Test #21: t_strm_reassem_holes .............   Passed    0.00 sec
       Start 22: t_strm_reassem_many
 22/33 Test #22: t_strm_reassem_many ..............   Passed    0.12 sec
       Start 23: t_strm_reassem_overlapping
 23/33 Test #23: t_strm_reassem_overlapping .......   Passed    0.00 sec
       Start 24: t_strm_reassem_win
 24/33 Test #24: t_strm_reassem_win ...............   Passed    0.12 sec
       Start 25: t_strm_reassem_cap
 25/33 Test #25: t_strm_reassem_cap ...............   Passed    0.05 sec
       Start 26: t_byte_stream_construction
 26/33 Test #26: t_byte_stream_construction .......   Passed    0.00 sec
       Start 27: t_byte_stream_one_write
 27/33 Test #27: t_byte_stream_one_write ..........   Passed    0.00 sec
       Start 28: t_byte_stream_two_writes
 28/33 Test #28: t_byte_stream_two_writes .........   Passed    0.00 sec
       Start 29: t_byte_stream_capacity
 29/33 Test #29: t_byte_stream_capacity ...........   Passed    0.33 sec
       Start 30: t_byte_stream_many_writes
 30/33 Test #30: t_byte_stream_many_writes ........   Passed    0.01 sec
       Start 53: t_address_dt
 31/33 Test #53: t_address_dt .....................   Passed    0.01 sec
       Start 54: t_parser_dt
 32/33 Test #54: t_parser_dt ......................   Passed    0.00 sec
       Start 55: t_socket_dt
 33/33 Test #55: t_socket_dt ......................   Passed    0.01 sec

 100% tests passed, 0 tests failed out of 33

 Total Test time (real) =   0.91 sec
 [100%] Built target check_lab3
○ oslab@oslab-virtual-machine:~/Desktop/lab3-2023autumn-HistoriaY/sponge/build$ 
```