lab3_report

姓名	学号	邮箱	完成时间
陈嘉昀	211220137	jiayunchen@smail.nju.edu.cn	2023年11月9日

1 代码

1.1代码设计

整个发送逻辑就是画好状态图然后对着做就好，有这么几个edge case需要注意：

- syn包一定在local的_syn还在是为false的时候就发送，并且把_syn设置为true
- fin包有可能跟随数据包发送（如果接收方的窗口可以同时容纳“数据”和"fin bit"），也有可能单独发送（发送最后一个数据包的时，接收方的窗口放不下"fin bit"）
- 新的ack到达以后，对与timer的"清零"处理，必须在新的ack对outstanding_seg有影响的情况下进行
- 当tick发生的时候，如果要对timer进行“倍增处理”，必须在当前的window_size不为0的情况下进行

1.2 规约调整

在这次写的代码中，触发了ByteStream的一个defensive programming：

```
  s.payload() = Buffer(this->_stream.read(seg_sz));
  // current window can hold the "fin" bit
```

```
string ByteStream::peek_output(const size_t len) const {
    assert(len <= _curr_size && "read too much bytes");
    return this->buffer.substr(0, len);
}
```

这实际上是我设计的byte stream的一个undefined behavior，现在需要增加相应的specification：在使用read函数时，必须先通过buffer_size的检验

```
// keep send segment
uint16_t seg_sz = min(TCPConfig::MAX_PAYLOAD_SIZE, win_sz + _latest_ack - _next_seqno);
seg_sz = min(seg_sz, static_cast<uint16_t>(this->_stream.buffer_size()));
s.payload() = Buffer(this->_stream.read(seg_sz));
```

2 运行结果

```
100% tests passed, 0 tests failed out of 33
```