Comp9331

2017s2 Assignment1

kai fang

z5137591

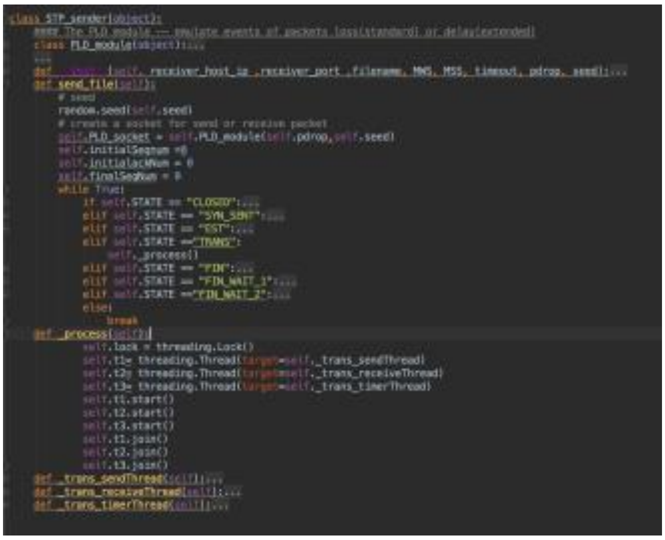
1. Design and Implementation

Language and platform: Python 3.6.0

List of features:

1. three-way handshake
2. four-segment connection termination
3. timeout operation with single-timer
4. fast retransmit
5. cumulative acknowledge and buffer for out-of-order packets
6. flow control using MWS

sender.py

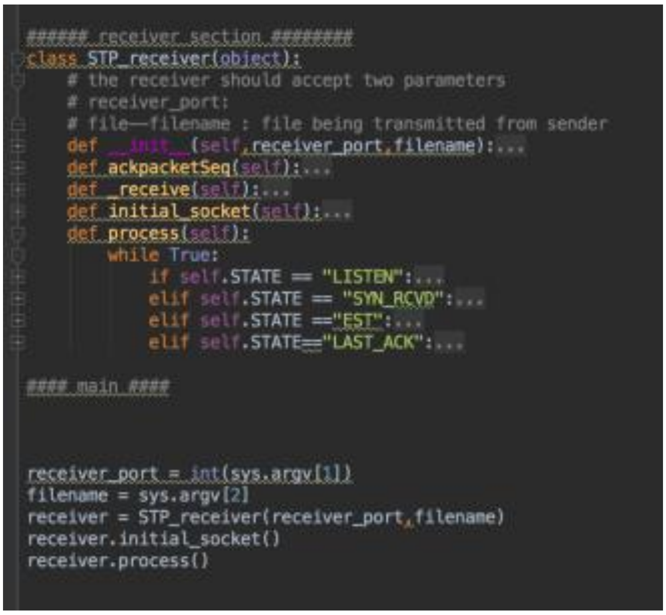


```
class STP_sender(object):
    #### The PLD module --- emulate events of packets loss(standard) or delay(extended)
    class PLD_module(object):...
    ...
    def __init__(self, receiver_host_ip ,receiver_port ,filename, MWS, MSS, timeout, pdrop, seed):...
    def send_file(self):
        # seed
        random.seed(self.seed)
        # create a socket for send or receive packet
        self.PLD_socket = self.PLD_module(self.pdrop,self.seed)
        self.initialSeqnum =0
        self.initialackNum = 0
        self.finalSeqNum = 0
        while True:
            if self.STATE == "CLOSED":...
            elif self.STATE == "SYN_SENT":...
            elif self.STATE == "EST":...
            elif self.STATE =="TRANS":
                self._process()
            elif self.STATE == "FIN":...
            elif self.STATE == "FIN_WAIT_1":...
            elif self.STATE =="FIN_WAIT_2":...
            else:
                break
    def _process(self):
            self.lock = threading.Lock()
            self.t1= threading.Thread(target=self._trans_sendThread)
            self.t2= threading.Thread(target=self._trans_receiveThread)
            self.t3= threading.Thread(target=self._trans_timerThread)
            self.t1.start()
            self.t2.start()
            self.t3.start()
            self.t1.join()
            self.t2.join()
            self.t3.join()
    def _trans_sendThread(self):...
    def _trans_receiveThread(self):...
    def _trans_timerThread(self):...
```

For sending file, we use while loop to model an infinite state machine from three ways handshake state to transmit data state, final with termination state.

In transmission process, we use multi process to do send, receive, timeout separately.

Receiver.py



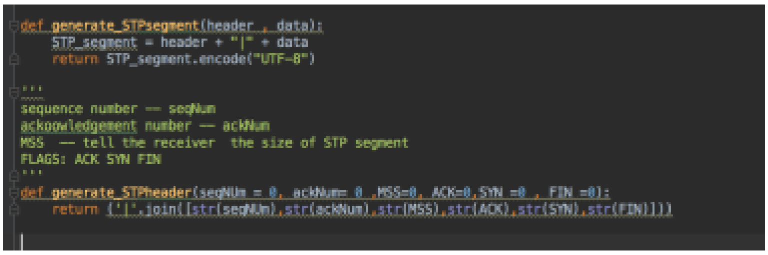
```
###### receiver section ########
class STP_receiver(object):
    # the receiver should accept two parameters
    # receiver_port:
    # file--filename : file being transmitted from sender
    def __init__(self,receiver_port,filename):...
    def ackpacketSeq(self):...
    def _receive(self):...
    def initial_socket(self):...
    def process(self):
        while True:
            if self.STATE == "LISTEN":...
            elif self.STATE == "SYN_RCVD":...
            elif self.STATE =="EST":...
            elif self.STATE=="LAST_ACK":...

#### main ####


receiver_port = int(sys.argv[1])
filename = sys.argv[2]
receiver = STP_receiver(receiver_port,filename)
receiver.initial_socket()
receiver.process()
```

same as sender.py, a simple infinite machine to transfer from each state.

2.Diagram of STP header and explanation



```
def generate_STPsegment(header , data):
    STP_segment = header + "|" + data
    return STP_segment.encode("UTF-8")

'''
sequence number -- seqNum
ackoowledgement number -- ackNum
MSS  -- tell the receiver  the size of STP segment
FLAGS: ACK SYN FIN
'''
def generate_STPheader(seqNUm = 0, ackNum= 0 ,MSS=0, ACK=0,SYN =0 , FIN =0):
    return ('|'.join([str(seqNUm),str(ackNum),str(MSS),str(ACK),str(SYN),str(FIN)]))
```

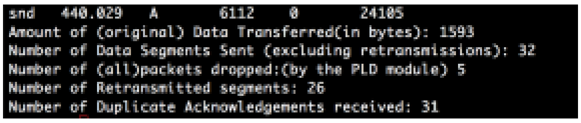
STP header

#seq_num	#ack_num
ACK	SYN
FIN	MSS
Data	

3. Question

(a)

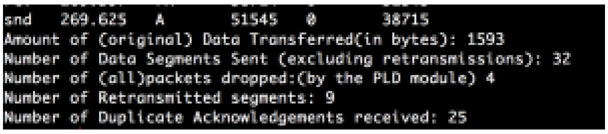
set timeout as 10MS



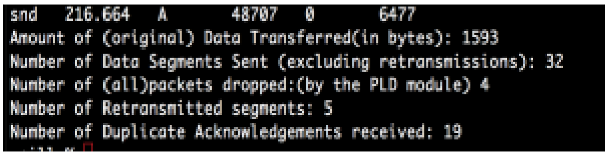
```
snd   440.029   A       6112    0       24185
Amount of (original) Data Transferred(in bytes): 1593
Number of Data Segments Sent (excluding retransmissions): 32
Number of (all)packets dropped:(by the PLD module) 5
Number of Retransmitted segments: 26
Number of Duplicate Acknowledgements received: 31
```

Due to lower timeout, the STP protocol will easily trigger the timeout module to cause a mass of unnecessary Retransmissions.

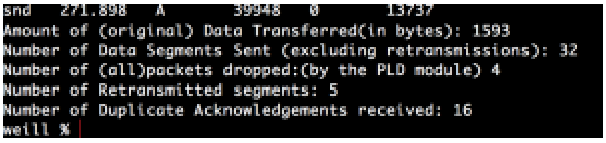
Set time out as 30MS, 50MS, 100ms



```
snd   269.625   A       51545   0       38715
Amount of (original) Data Transferred(in bytes): 1593
Number of Data Segments Sent (excluding retransmissions): 32
Number of (all)packets dropped:(by the PLD module) 4
Number of Retransmitted segments: 9
Number of Duplicate Acknowledgements received: 25
```



```
snd   216.664   A       48707   0       6477
Amount of (original) Data Transferred(in bytes): 1593
Number of Data Segments Sent (excluding retransmissions): 32
Number of (all)packets dropped:(by the PLD module) 4
Number of Retransmitted segments: 5
Number of Duplicate Acknowledgements received: 19
```

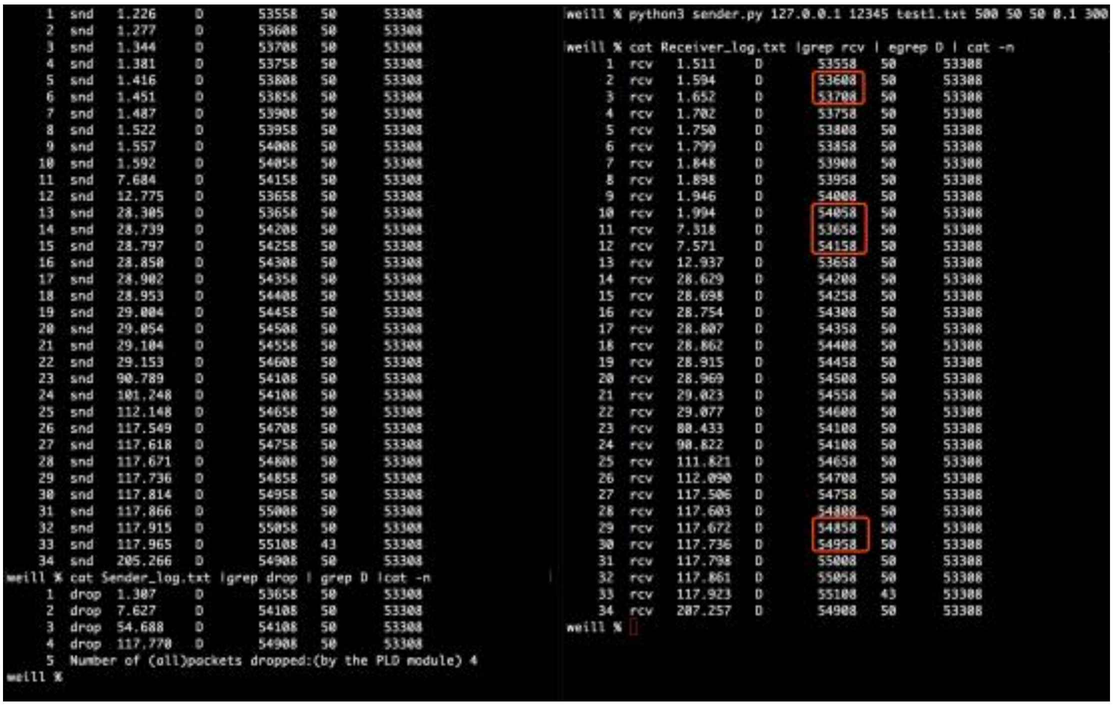


```
snd   271.898   A       39948   0       13737
Amount of (original) Data Transferred(in bytes): 1593
Number of Data Segments Sent (excluding retransmissions): 32
Number of (all)packets dropped:(by the PLD module) 4
Number of Retransmitted segments: 5
Number of Duplicate Acknowledgements received: 16
weill %
```

When the value of timeout increased as 50MS, the number of Retransmission turns reasonable. So we set timeout as 50MS for test.

With the timeout set as 50 MS:

Pdrop = 0.1:



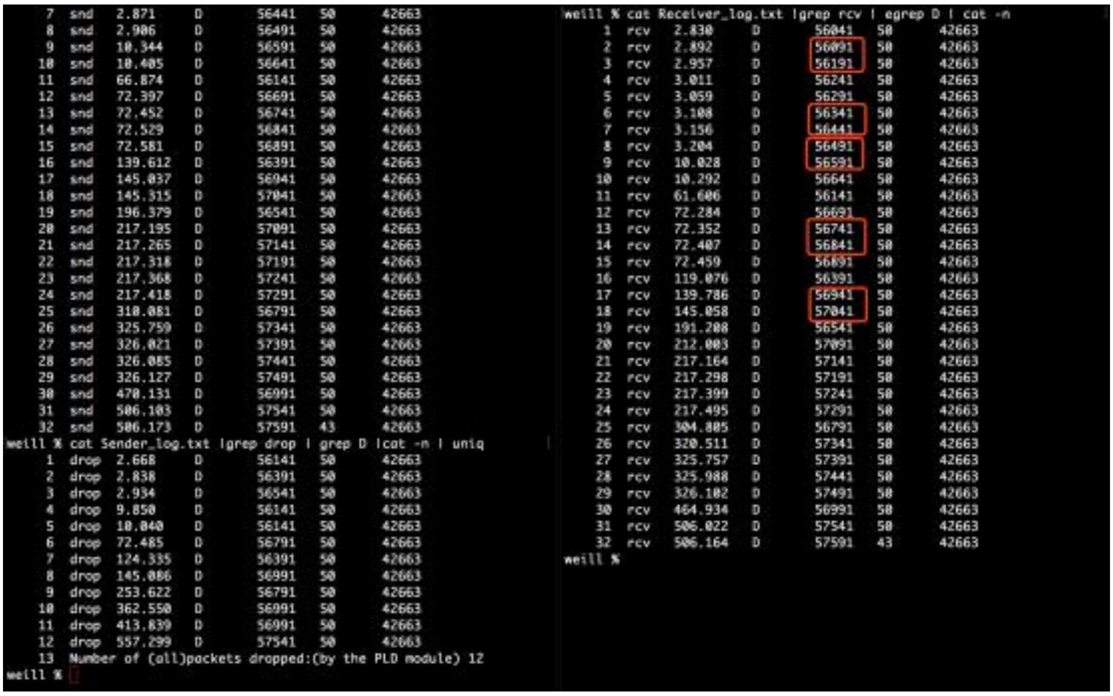
```
     1  snd   1.226     D       53558   50      53308
     2  snd   1.277     D       53608   50      53308
     3  snd   1.344     D       53708   50      53308
     4  snd   1.381     D       53758   50      53308
     5  snd   1.416     D       53808   50      53308
     6  snd   1.451     D       53858   50      53308
     7  snd   1.487     D       53908   50      53308
     8  snd   1.522     D       53958   50      53308
     9  snd   1.557     D       54008   50      53308
    10  snd   1.592     D       54058   50      53308
    11  snd   7.684     D       54158   50      53308
    12  snd   12.775    D       53658   50      53308
    13  snd   28.305    D       53658   50      53308
    14  snd   28.739    D       54208   50      53308
    15  snd   28.797    D       54258   50      53308
    16  snd   28.850    D       54308   50      53308
    17  snd   28.902    D       54358   50      53308
    18  snd   28.953    D       54408   50      53308
    19  snd   29.004    D       54458   50      53308
    20  snd   29.054    D       54508   50      53308
    21  snd   29.104    D       54558   50      53308
    22  snd   29.153    D       54608   50      53308
    23  snd   90.789    D       54108   50      53308
    24  snd   101.248   D       54108   50      53308
    25  snd   112.148   D       54658   50      53308
    26  snd   117.549   D       54708   50      53308
    27  snd   117.618   D       54758   50      53308
    28  snd   117.671   D       54808   50      53308
    29  snd   117.736   D       54858   50      53308
    30  snd   117.814   D       54958   50      53308
    31  snd   117.866   D       55008   50      53308
    32  snd   117.915   D       55058   50      53308
    33  snd   117.965   D       55108   43      53308
    34  snd   205.266   D       54908   50      53308
weill % cat Sender_log.txt |grep drop | grep D |cat -n
     1  drop  1.307     D       53658   50      53308
     2  drop  7.627     D       54108   50      53308
     3  drop  54.688    D       54108   50      53308
     4  drop  117.770   D       54908   50      53308
     5  Number of (all)packets dropped:(by the PLD module) 4
weill %
```

```
weill % python3 sender.py 127.0.0.1 12345 test1.txt 500 50 50 0.1 300

weill % cat Receiver_log.txt |grep rcv | egrep D | cat -n
     1  rcv   1.511     D       53558   50      53308
     2  rcv   1.594     D       53608   50      53308
     3  rcv   1.652     D       53708   50      53308
     4  rcv   1.702     D       53758   50      53308
     5  rcv   1.750     D       53808   50      53308
     6  rcv   1.799     D       53858   50      53308
     7  rcv   1.848     D       53908   50      53308
     8  rcv   1.898     D       53958   50      53308
     9  rcv   1.946     D       54008   50      53308
    10  rcv   1.994     D       54058   50      53308
    11  rcv   7.318     D       53658   50      53308
    12  rcv   7.571     D       54158   50      53308
    13  rcv   12.937    D       53658   50      53308
    14  rcv   28.629    D       54208   50      53308
    15  rcv   28.698    D       54258   50      53308
    16  rcv   28.754    D       54308   50      53308
    17  rcv   28.807    D       54358   50      53308
    18  rcv   28.862    D       54408   50      53308
    19  rcv   28.915    D       54458   50      53308
    20  rcv   28.969    D       54508   50      53308
    21  rcv   29.023    D       54558   50      53308
    22  rcv   29.077    D       54608   50      53308
    23  rcv   80.433    D       54108   50      53308
    24  rcv   90.822    D       54108   50      53308
    25  rcv   111.821   D       54658   50      53308
    26  rcv   112.090   D       54708   50      53308
    27  rcv   117.506   D       54758   50      53308
    28  rcv   117.603   D       54808   50      53308
    29  rcv   117.672   D       54858   50      53308
    30  rcv   117.736   D       54958   50      53308
    31  rcv   117.798   D       55008   50      53308
    32  rcv   117.861   D       55058   50      53308
    33  rcv   117.923   D       55108   43      53308
    34  rcv   207.257   D       54908   50      53308
weill %
```

pdrop = 0.3

#53658 – 53708, $ 54108 – 54158, # 54908 – 54958 dropped



```
     7  snd   2.871     D       56441   50      42663
     8  snd   2.906     D       56491   50      42663
     9  snd   10.344    D       56591   50      42663
    10  snd   10.405    D       56641   50      42663
    11  snd   66.874    D       56141   50      42663
    12  snd   72.397    D       56691   50      42663
    13  snd   72.452    D       56741   50      42663
    14  snd   72.529    D       56841   50      42663
    15  snd   72.581    D       56891   50      42663
    16  snd   139.612   D       56391   50      42663
    17  snd   145.037   D       56941   50      42663
    18  snd   145.315   D       57041   50      42663
    19  snd   196.379   D       56541   50      42663
    20  snd   217.195   D       57091   50      42663
    21  snd   217.265   D       57141   50      42663
    22  snd   217.318   D       57191   50      42663
    23  snd   217.368   D       57241   50      42663
    24  snd   217.418   D       57291   50      42663
    25  snd   318.081   D       56791   50      42663
    26  snd   325.759   D       57341   50      42663
    27  snd   326.021   D       57391   50      42663
    28  snd   326.085   D       57441   50      42663
    29  snd   326.127   D       57491   50      42663
    30  snd   470.131   D       56991   50      42663
    31  snd   506.103   D       57541   50      42663
    32  snd   506.173   D       57591   43      42663
weill % cat Sender_log.txt |grep drop | grep D |cat -n | uniq
     1  drop  2.668     D       56141   50      42663
     2  drop  2.838     D       56391   50      42663
     3  drop  2.934     D       56541   50      42663
     4  drop  9.850     D       56141   50      42663
     5  drop  10.040    D       56141   50      42663
     6  drop  72.485    D       56791   50      42663
     7  drop  124.335   D       56391   50      42663
     8  drop  145.086   D       56991   50      42663
     9  drop  253.622   D       56791   50      42663
    10  drop  362.550   D       56991   50      42663
    11  drop  413.839   D       56991   50      42663
    12  drop  557.299   D       57541   50      42663
    13  Number of (all)packets dropped:(by the PLD module) 12
weill %
```

```
weill % cat Receiver_log.txt |grep rcv | egrep D | cat -n
     1  rcv   2.830     D       56041   50      42663
     2  rcv   2.892     D       56091   50      42663
     3  rcv   2.957     D       56191   50      42663
     4  rcv   3.011     D       56241   50      42663
     5  rcv   3.059     D       56291   50      42663
     6  rcv   3.108     D       56341   50      42663
     7  rcv   3.156     D       56441   50      42663
     8  rcv   3.204     D       56491   50      42663
     9  rcv   10.028    D       56591   50      42663
    10  rcv   10.292    D       56641   50      42663
    11  rcv   61.606    D       56141   50      42663
    12  rcv   72.284    D       56691   50      42663
    13  rcv   72.352    D       56741   50      42663
    14  rcv   72.407    D       56841   50      42663
    15  rcv   72.459    D       56891   50      42663
    16  rcv   119.076   D       56391   50      42663
    17  rcv   139.786   D       56941   50      42663
    18  rcv   145.058   D       57041   50      42663
    19  rcv   191.208   D       56541   50      42663
    20  rcv   212.003   D       57091   50      42663
    21  rcv   217.164   D       57141   50      42663
    22  rcv   217.298   D       57191   50      42663
    23  rcv   217.399   D       57241   50      42663
    24  rcv   217.495   D       57291   50      42663
    25  rcv   304.805   D       56791   50      42663
    26  rcv   320.511   D       57341   50      42663
    27  rcv   325.757   D       57391   50      42663
    28  rcv   325.988   D       57441   50      42663
    29  rcv   326.102   D       57491   50      42663
    30  rcv   464.934   D       56991   50      42663
    31  rcv   506.022   D       57541   50      42663
    32  rcv   506.164   D       57591   43      42663
weill %
```

# 56141 -56191 # 56391 -56441 # 56551 -56591 # 56791 – 56841 # 56991-57041 dropped

(b)

	50ms	200ms	12ms
NumOfTransmitted packets	46	45	60
Time cost	214.308	136.768	250.625

When timeout set as 12ms, it causes lots of Retransmission works. The reason is that: the program could easily trigger the single timer and result in timeout. When the timeout increased from 12 ms to 200 ms, the retransmission work could not mainly produce by timeout instead of fast retransmission. So most of dropped packets will be retransmitted by fast transmission function.