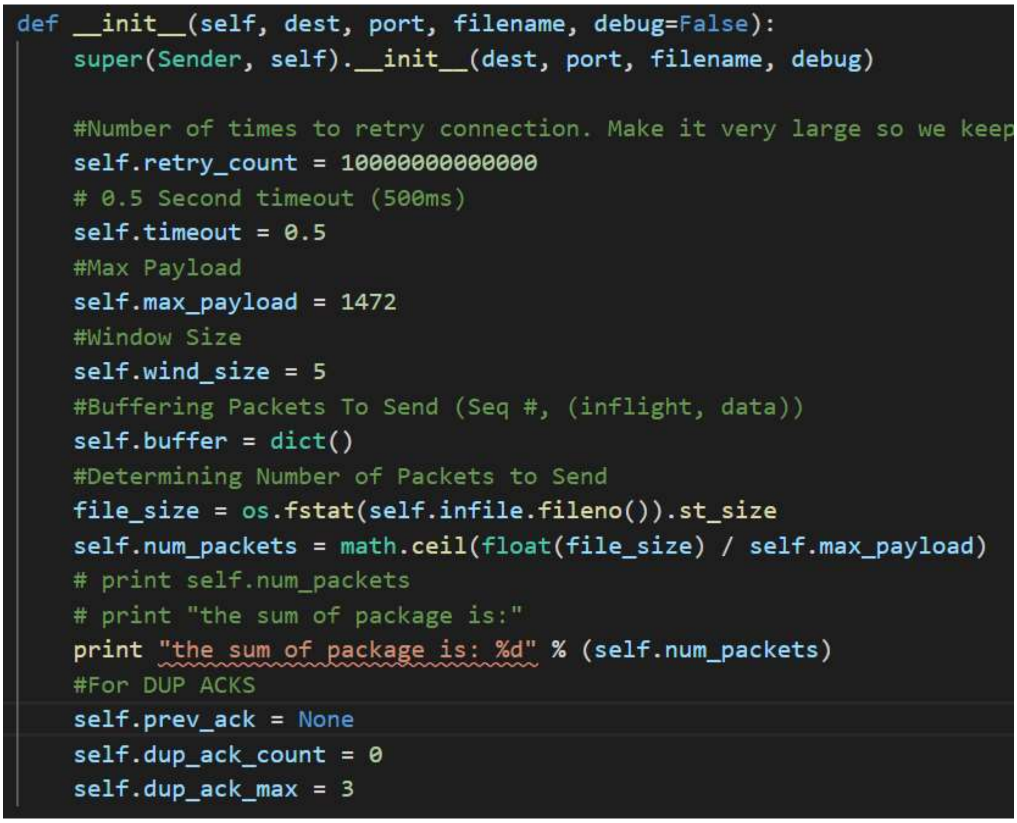
姓名：滕德淋　　　　班级：11J192

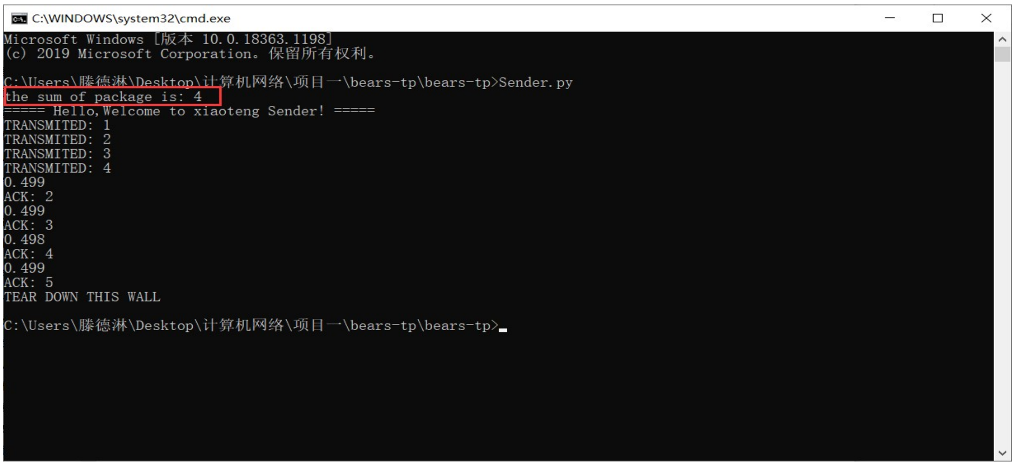
第一步

初始化。做好发送数据准备，规定滑动窗口大小为 5 个 wind、规定 timeout 大小为 0.5s、规定报文段填装数据大小 1472、为滑动窗口分配内存、规定当服务器端无响应的时候重发次数（很大），计算得出一共需要发送几次报文段;



```python
def __init__(self, dest, port, filename, debug=False):
    super(Sender, self).__init__(dest, port, filename, debug)

    #Number of times to retry connection. Make it very large so we keep
    self.retry_count = 10000000000000
    # 0.5 Second timeout (500ms)
    self.timeout = 0.5
    #Max Payload
    self.max_payload = 1472
    #Window Size
    self.wind_size = 5
    #Buffering Packets To Send (Seq #, (inflight, data))
    self.buffer = dict()
    #Determining Number of Packets to Send
    file_size = os.fstat(self.infile.fileno()).st_size
    self.num_packets = math.ceil(float(file_size) / self.max_payload)
    # print self.num_packets
    # print "the sum of package is:"
    print "the sum of package is: %d" % (self.num_packets)
    #For DUP ACKS
    self.prev_ack = None
    self.dup_ack_count = 0
    self.dup_ack_max = 3
```

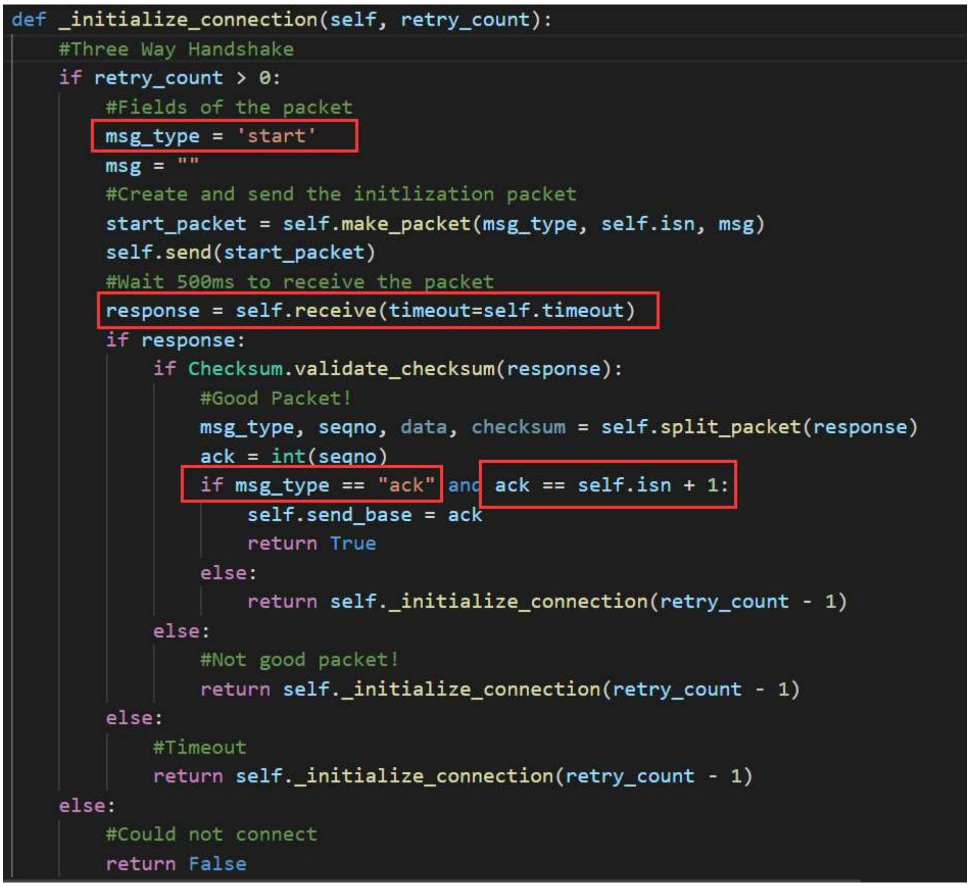
运行结果：



```
Microsoft Windows [版本 10.0.18363.1198]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\滕德淋\Desktop\计算机网络\项目一\bears-tp\bears-tp>Sender.py
the sum of package is: 4
===== Hello,Welcome to xiaoteng Sender! =====
TRANSMITED: 1
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0.499
ACK: 2
0.499
ACK: 3
0.498
ACK: 4
0.499
ACK: 5
TEAR DOWN THIS WALL

C:\Users\滕德淋\Desktop\计算机网络\项目一\bears-tp\bears-tp>
```

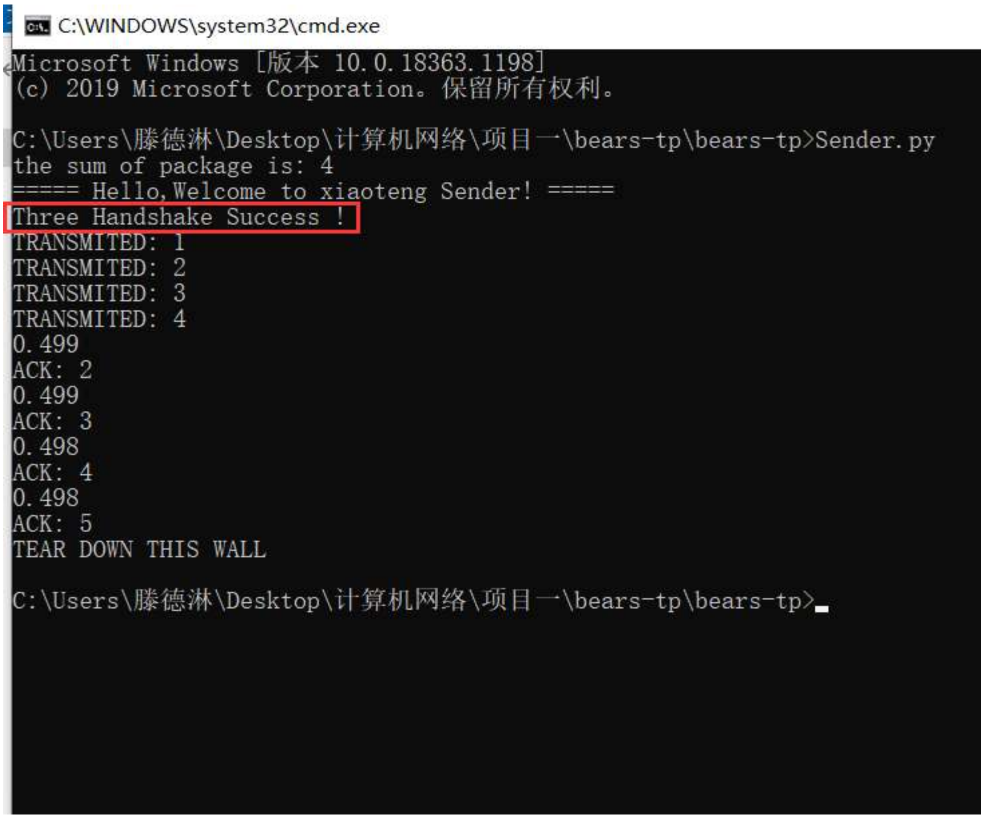
第二步

进行发送数据前准备—三次握手，具体实现是先发送端发送一个“start”信号报文段，经过打包之后发送给服务器端，当服务器端收到开始信号报文段之后回应一个“ack”类型的报文段，并且携带序列号和确认号（由开始报文段序列号+1 得到）；之后 Sender 端经过校验和（通过反码进行核对）确认是一个好包之后再发送一个数据段；三次握手成功，初始准备完成；



```
def _initialize_connection(self, retry_count):
    #Three Way Handshake
    if retry_count > 0:
        #Fields of the packet
        msg_type = 'start'
        msg = ""
        #Create and send the initlization packet
        start_packet = self.make_packet(msg_type, self.isn, msg)
        self.send(start_packet)
        #Wait 500ms to receive the packet
        response = self.receive(timeout=self.timeout)
        if response:
            if Checksum.validate_checksum(response):
                #Good Packet!
                msg_type, seqno, data, checksum = self.split_packet(response)
                ack = int(seqno)
                if msg_type == "ack" and ack == self.isn + 1:
                    self.send_base = ack
                    return True
                else:
                    return self._initialize_connection(retry_count - 1)
            else:
                #Not good packet!
                return self._initialize_connection(retry_count - 1)
        else:
            #Timeout
            return self._initialize_connection(retry_count - 1)
    else:
        #Could not connect
        return False
```

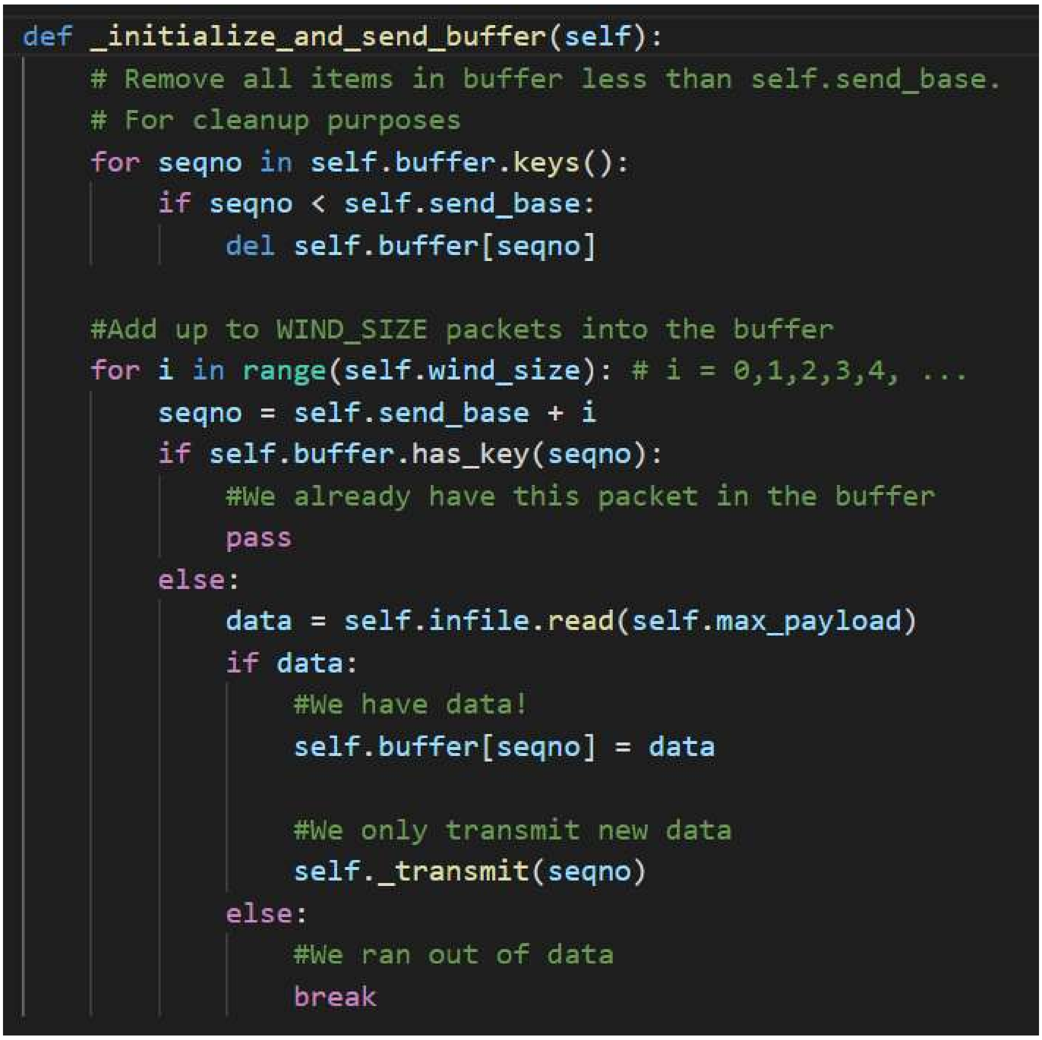
运行结果：



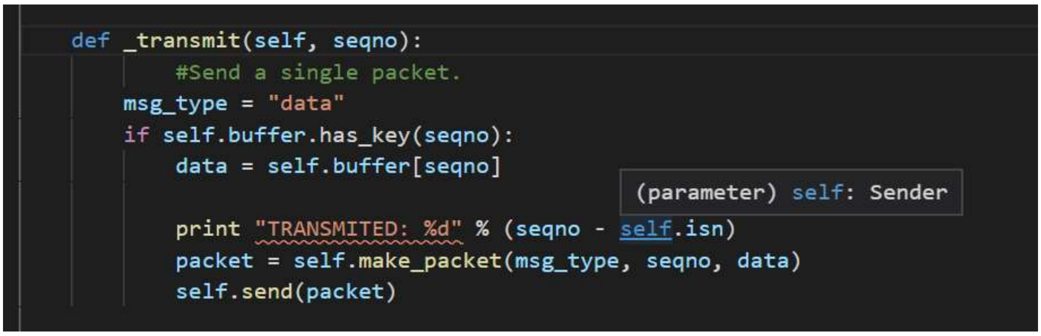
```
Microsoft Windows [版本 10.0.18363.1198]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\滕德淋\Desktop\计算机网络\项目一\bears-tp\bears-tp>Sender.py
the sum of package is: 4
===== Hello,Welcome to xiaoteng Sender! =====
Three Handshake Success !
TRANSMITED: 1
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0.499
ACK: 2
0.499
ACK: 3
0.498
ACK: 4
0.498
ACK: 5
TEAR DOWN THIS WALL

C:\Users\滕德淋\Desktop\计算机网络\项目一\bears-tp\bears-tp>
```

第三步

进行数据发送，先清空 buffer 中左侧的数据，然后将报文读取到的数据进行分包先发送到 buffer 中，然后再发送到接收端，并同时进行 Timeout 的计时；

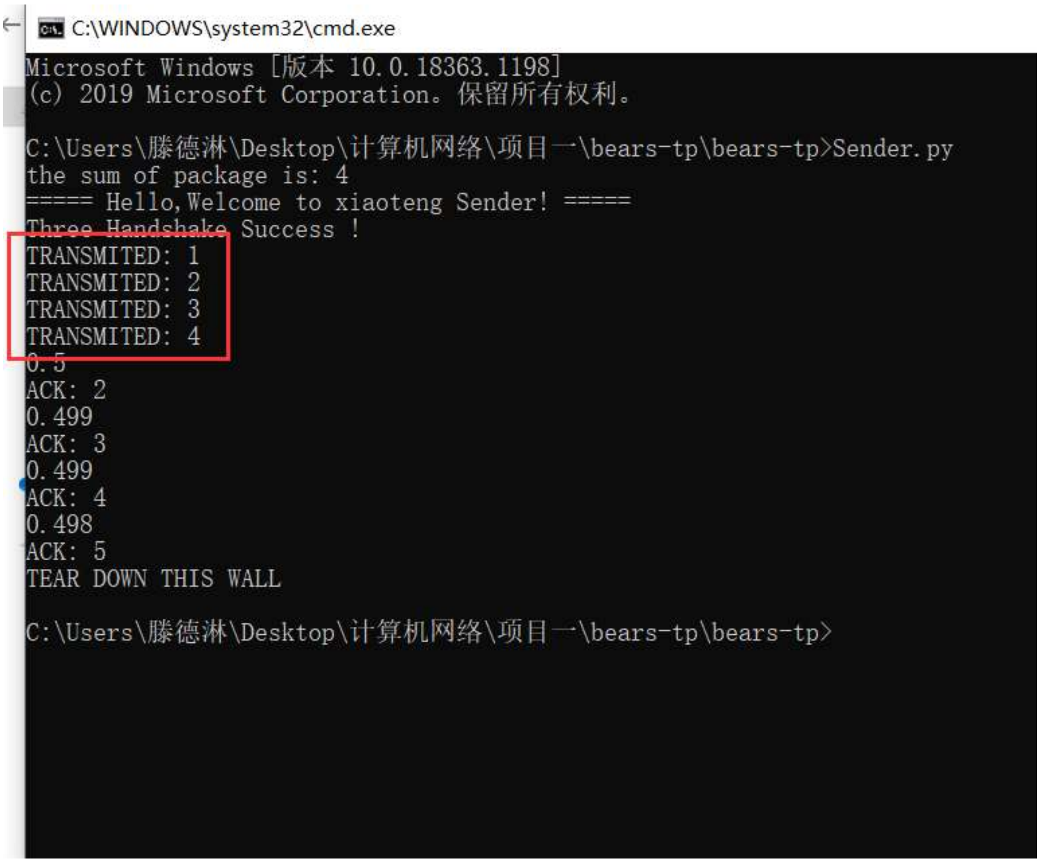


```
def _initialize_and_send_buffer(self):
    # Remove all items in buffer less than self.send_base.
    # For cleanup purposes
    for seqno in self.buffer.keys():
        if seqno < self.send_base:
            del self.buffer[seqno]

    #Add up to WIND_SIZE packets into the buffer
    for i in range(self.wind_size): # i = 0,1,2,3,4, ...
        seqno = self.send_base + i
        if self.buffer.has_key(seqno):
            #We already have this packet in the buffer
            pass
        else:
            data = self.infile.read(self.max_payload)
            if data:
                #We have data!
                self.buffer[seqno] = data

                #We only transmit new data
                self._transmit(seqno)
            else:
                #We ran out of data
                break
```



```
    def _transmit(self, seqno):
            #Send a single packet.
        msg_type = "data"
        if self.buffer.has_key(seqno):
            data = self.buffer[seqno]

            print "TRANSMITED: %d" % (seqno - self.isn)
            packet = self.make_packet(msg_type, seqno, data)
            self.send(packet)
```

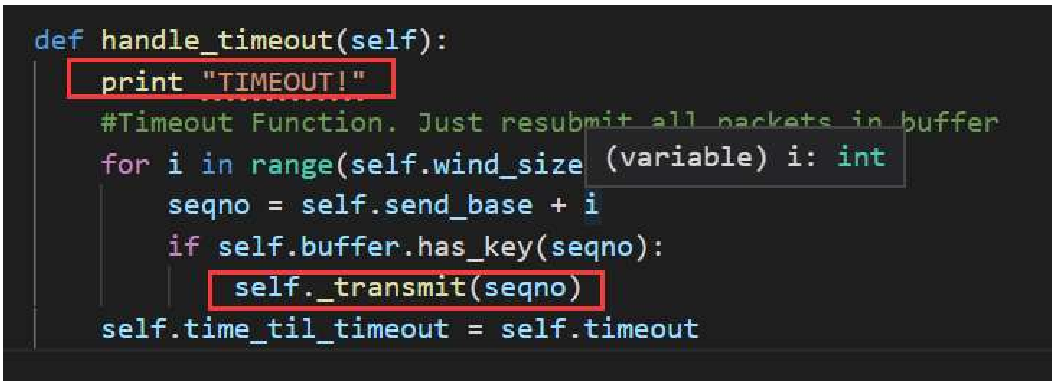
运行结果：



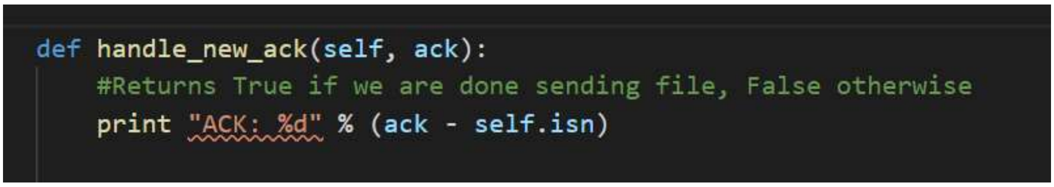
```
Microsoft Windows [版本 10.0.18363.1198]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\滕德淋\Desktop\计算机网络\项目一\bears-tp\bears-tp>Sender.py
the sum of package is: 4
===== Hello,Welcome to xiaoteng Sender! =====
Three Handshake Success !
TRANSMITED: 1
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0.5
ACK: 2
0.499
ACK: 3
0.499
ACK: 4
0.498
ACK: 5
TEAR DOWN THIS WALL

C:\Users\滕德淋\Desktop\计算机网络\项目一\bears-tp\bears-tp>
```

第四步

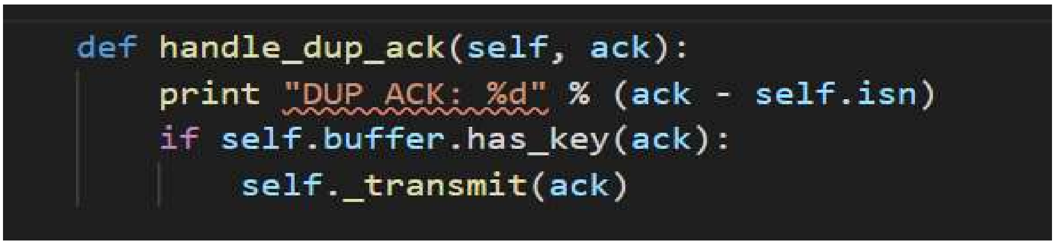
监听“ack”报文段，将发送时间和接收时间差进行相减，计算是否超出 timeout，超出则输出 Timeout 并进行重传，好包则进行拆包；同时进行处理 ack 重复发送的情况。



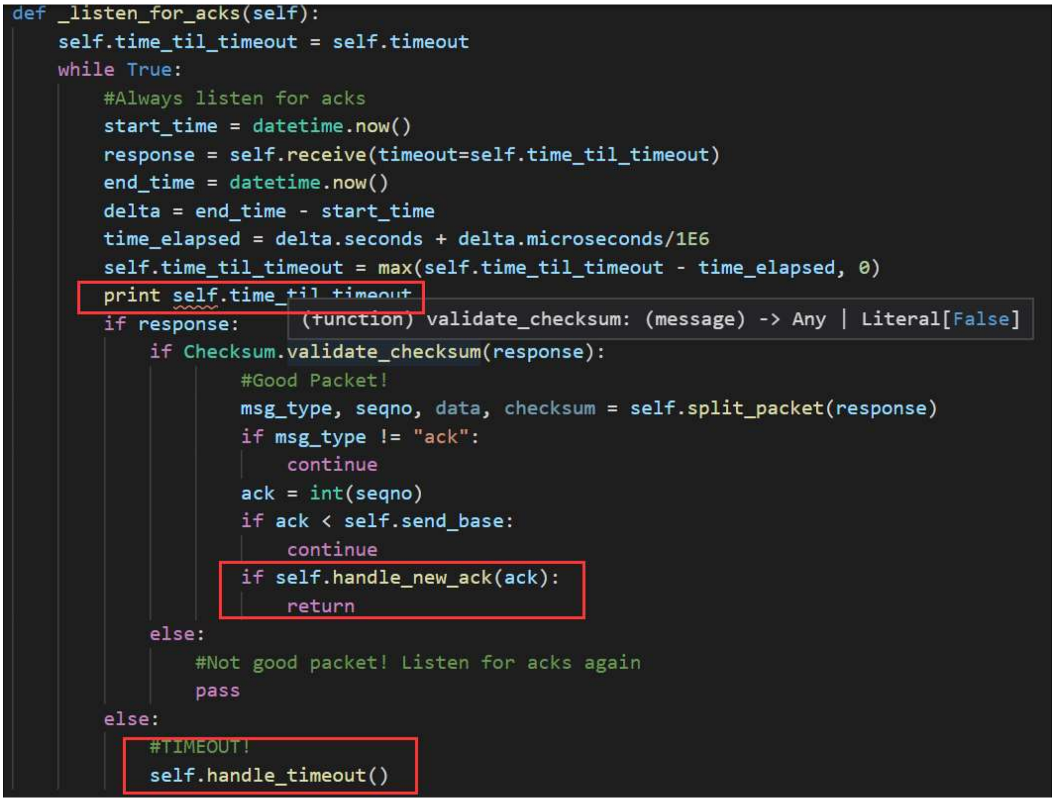
```
def handle_timeout(self):
    print "TIMEOUT!"
    #Timeout Function. Just resubmit all packets in buffer
    for i in range(self.wind_size
        seqno = self.send_base + i
        if self.buffer.has_key(seqno):
            self._transmit(seqno)
    self.time_til_timeout = self.timeout
```



```
def handle_new_ack(self, ack):
    #Returns True if we are done sending file, False otherwise
    print "ACK: %d" % (ack - self.isn)
```

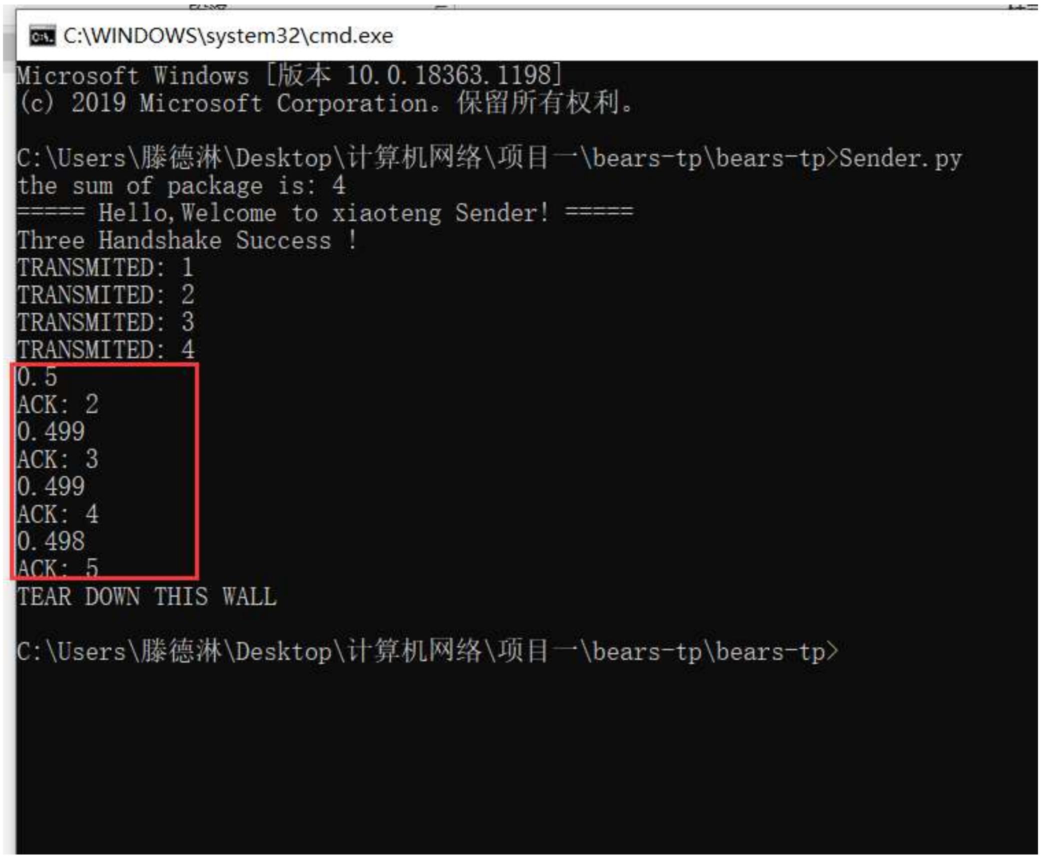


```
def handle_dup_ack(self, ack):
    print "DUP ACK: %d" % (ack - self.isn)
    if self.buffer.has_key(ack):
        self._transmit(ack)
```



```
def _listen_for_acks(self):
    self.time_til_timeout = self.timeout
    while True:
        #Always listen for acks
        start_time = datetime.now()
        response = self.receive(timeout=self.time_til_timeout)
        end_time = datetime.now()
        delta = end_time - start_time
        time_elapsed = delta.seconds + delta.microseconds/1E6
        self.time_til_timeout = max(self.time_til_timeout - time_elapsed, 0)
        print self.time_til_timeout
        if response:    (function) validate_checksum: (message) -> Any | Literal[False]
            if Checksum.validate_checksum(response):
                    #Good Packet!
                    msg_type, seqno, data, checksum = self.split_packet(response)
                    if msg_type != "ack":
                        continue
                    ack = int(seqno)
                    if ack < self.send_base:
                        continue
                    if self.handle_new_ack(ack):
                        return
            else:
                #Not good packet! Listen for acks again
                pass
        else:
            #TIMEOUT!
            self.handle_timeout()
```

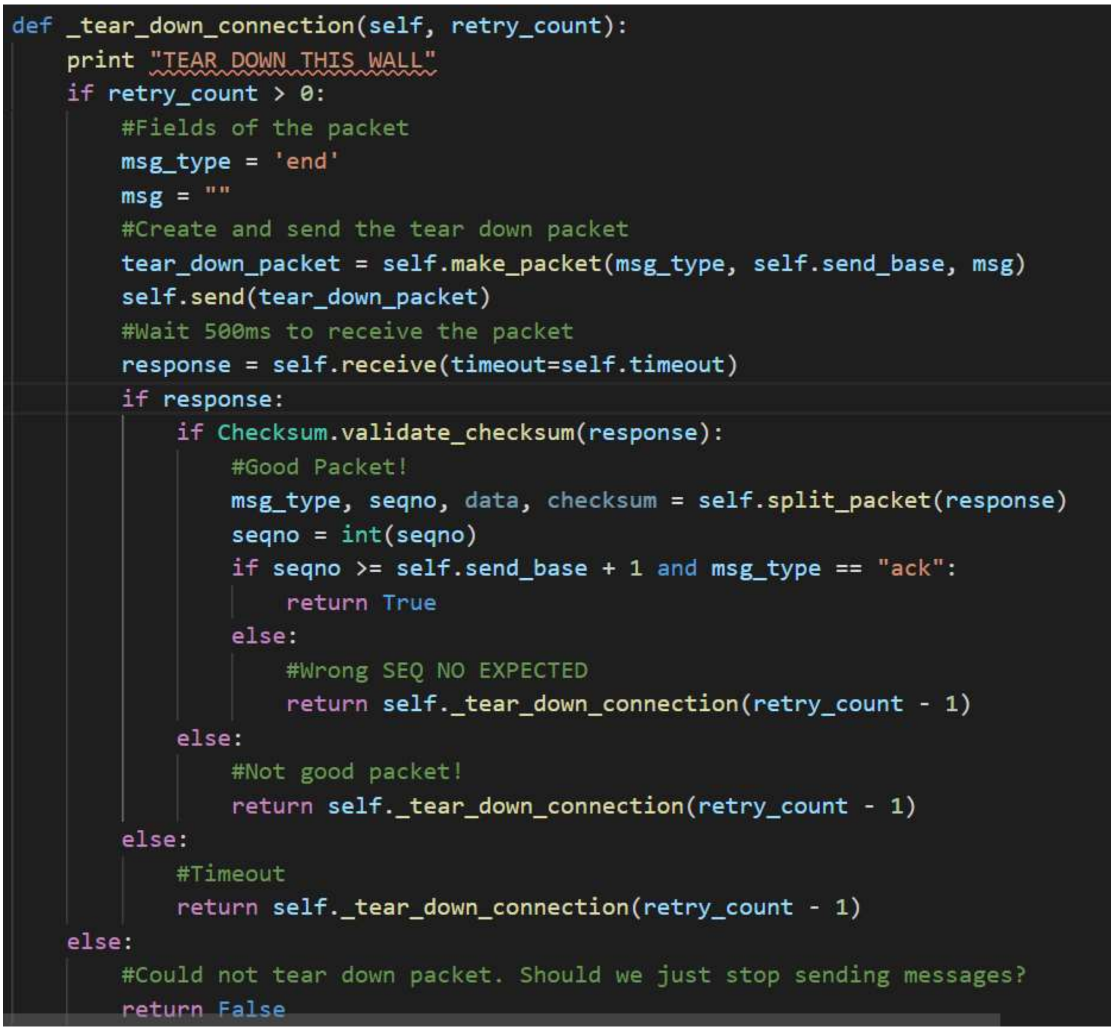
运行结果：



```
Microsoft Windows [版本 10.0.18363.1198]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\滕德淋\Desktop\计算机网络\项目一\bears-tp\bears-tp>Sender.py
the sum of package is: 4
===== Hello,Welcome to xiaoteng Sender! =====
Three Handshake Success !
TRANSMITED: 1
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0.5
ACK: 2
0.499
ACK: 3
0.499
ACK: 4
0.498
ACK: 5
TEAR DOWN THIS WALL

C:\Users\滕德淋\Desktop\计算机网络\项目一\bears-tp\bears-tp>
```

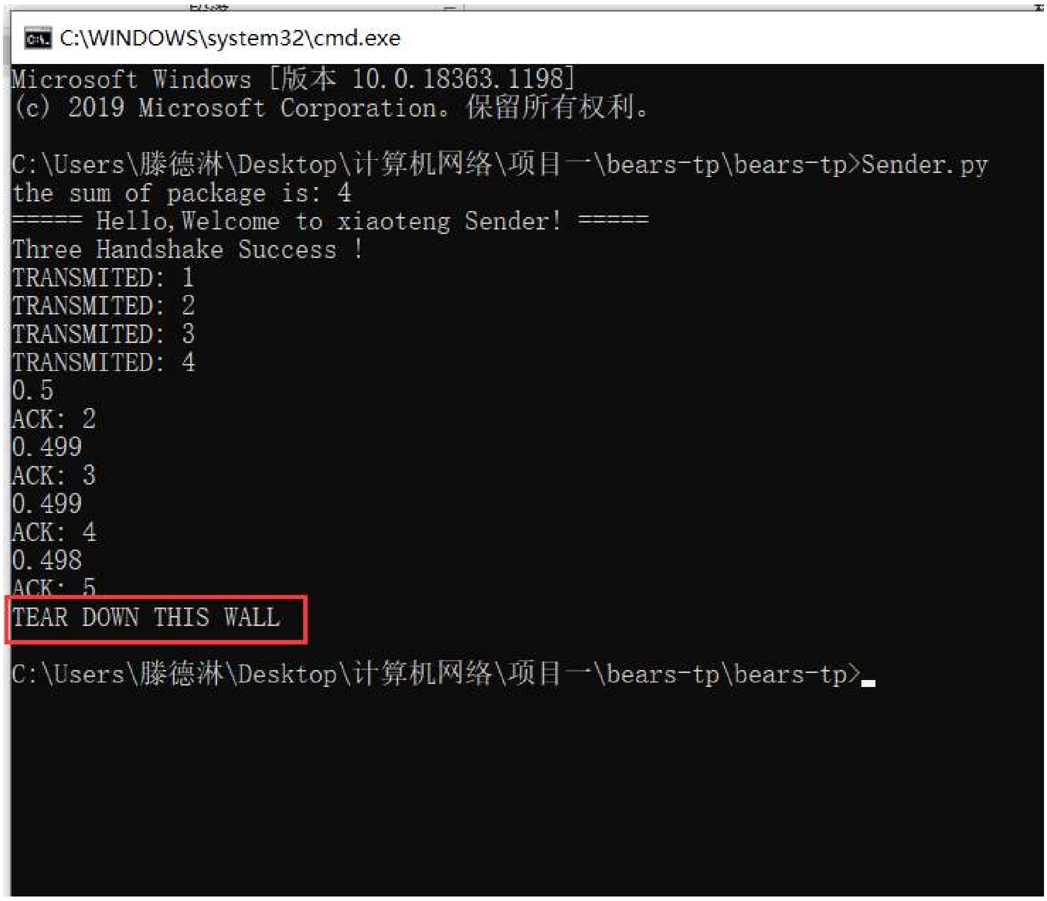
第五步

数据发送完毕，断开连接，首先发送一个 end 类型报文段，接收端返回一个 ack 报文段，并携带序列号和确认号，等 500ms 之后接收端发送一个 end 报文段，发送端接收到之后返回一个 ack，完成四次挥手之后进行连接断开。完成断开。



```
def _tear_down_connection(self, retry_count):
    print "TEAR DOWN THIS WALL"
    if retry_count > 0:
        #Fields of the packet
        msg_type = 'end'
        msg = ""
        #Create and send the tear down packet
        tear_down_packet = self.make_packet(msg_type, self.send_base, msg)
        self.send(tear_down_packet)
        #Wait 500ms to receive the packet
        response = self.receive(timeout=self.timeout)
        if response:
            if Checksum.validate_checksum(response):
                #Good Packet!
                msg_type, seqno, data, checksum = self.split_packet(response)
                seqno = int(seqno)
                if seqno >= self.send_base + 1 and msg_type == "ack":
                    return True
                else:
                    #Wrong SEQ NO EXPECTED
                    return self._tear_down_connection(retry_count - 1)
            else:
                #Not good packet!
                return self._tear_down_connection(retry_count - 1)
        else:
            #Timeout
            return self._tear_down_connection(retry_count - 1)
    else:
        #Could not tear down packet. Should we just stop sending messages?
        return False
```

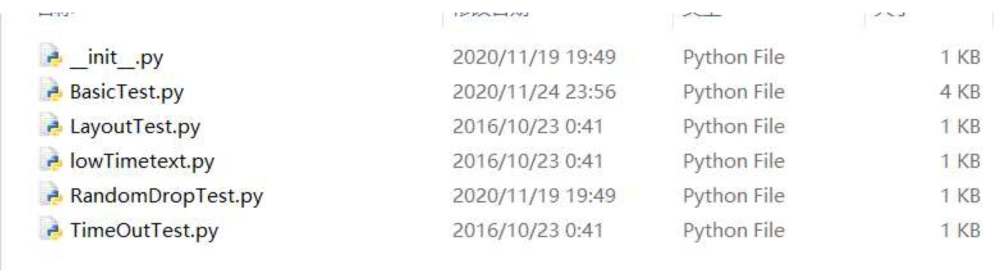
运行结果：



```
Microsoft Windows [版本 10.0.18363.1198]
(c) 2019 Microsoft Corporation。保留所有权利。

C:\Users\滕德淋\Desktop\计算机网络\项目一\bears-tp\bears-tp>Sender.py
the sum of package is: 4
===== Hello,Welcome to xiaoteng Sender! =====
Three Handshake Success !
TRANSMITED: 1
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0.5
ACK: 2
0.499
ACK: 3
0.499
ACK: 4
0.498
ACK: 5
TEAR DOWN THIS WALL

C:\Users\滕德淋\Desktop\计算机网络\项目一\bears-tp\bears-tp>
```

第六步

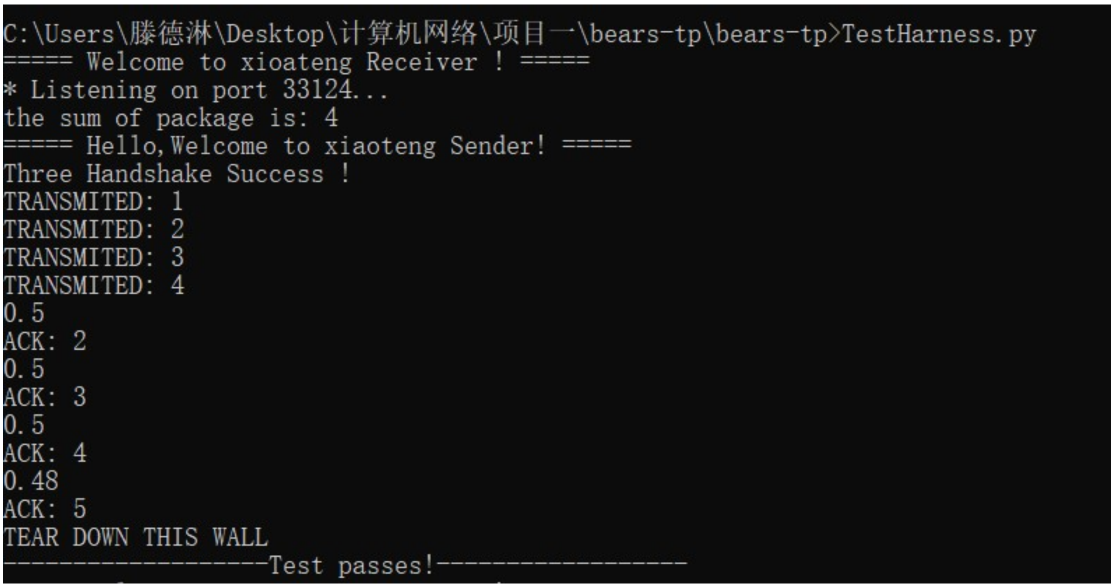
编写测试案例，分别是原有的基础测试案例 BasicTest 和 RandomDropTest，自己新编写的三个测试案例分别是 lowTimetext 和 TimeoutTest 和 LayoutTest（未完成）



__init__.py	2020/11/19 19:49	Python File	1 KB
BasicTest.py	2020/11/24 23:56	Python File	4 KB
LayoutTest.py	2016/10/23 0:41	Python File	1 KB
lowTimetext.py	2016/10/23 0:41	Python File	1 KB
RandomDropTest.py	2020/11/19 19:49	Python File	1 KB
TimeOutTest.py	2016/10/23 0:41	Python File	1 KB

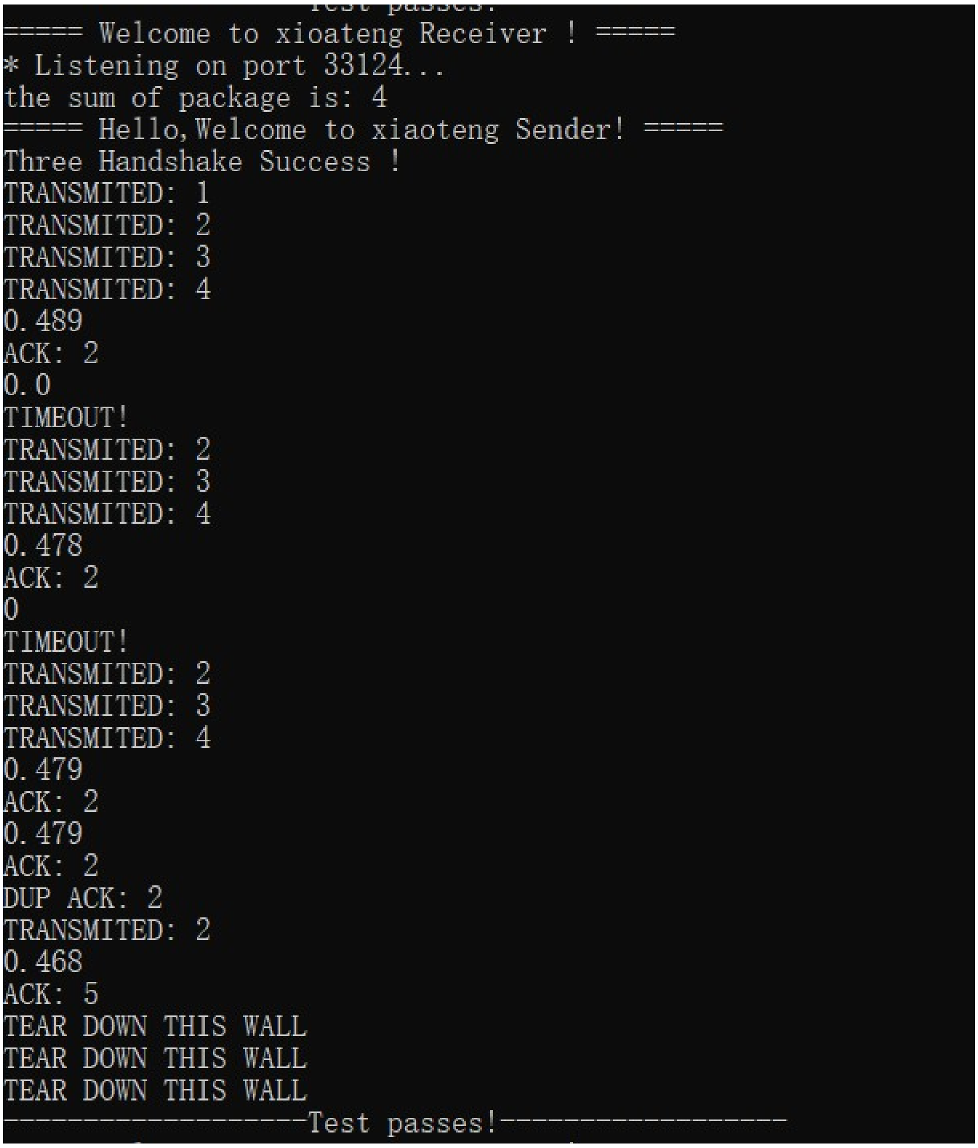
进行测试：

BasicTest



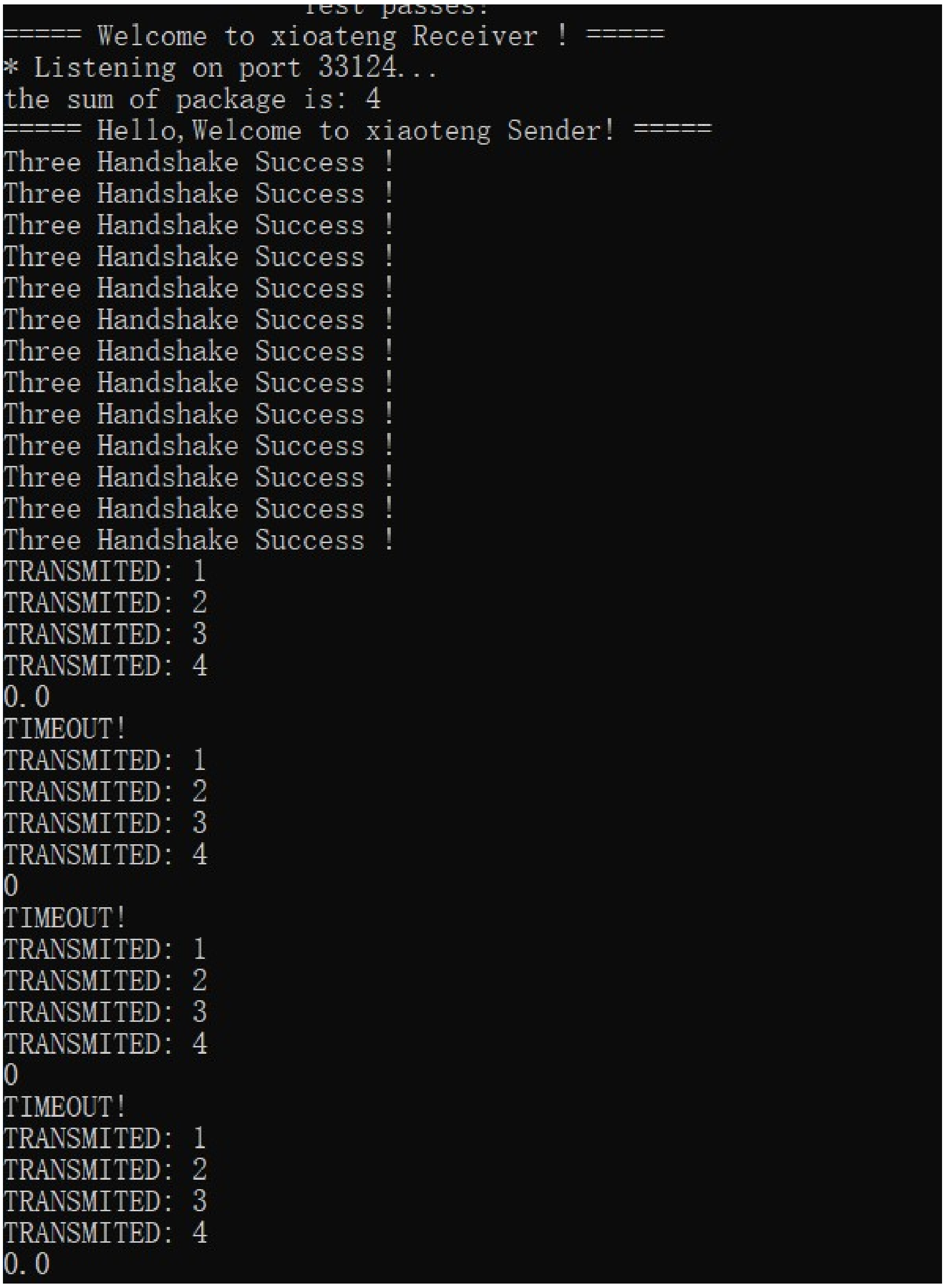
```
C:\Users\滕德淋\Desktop\计算机网络\项目一\bears-tp\bears-tp>TestHarness.py
===== Welcome to xioateng Receiver ! =====
* Listening on port 33124...
the sum of package is: 4
===== Hello,Welcome to xiaoteng Sender! =====
Three Handshake Success !
TRANSMITED: 1
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0.5
ACK: 2
0.5
ACK: 3
0.5
ACK: 4
0.48
ACK: 5
TEAR DOWN THIS WALL
-------------------Test passes!------------------
```

RandomDropTest：

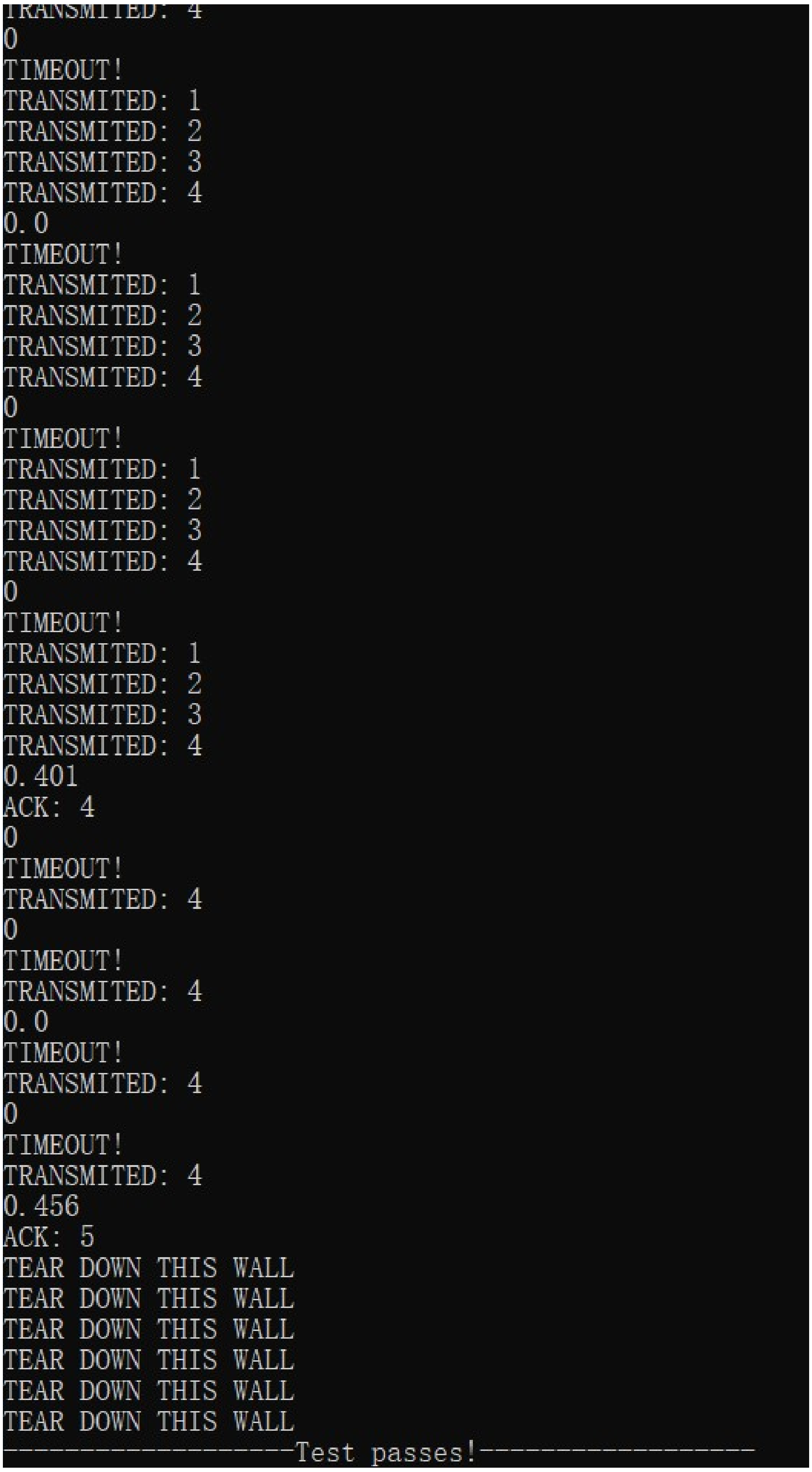


```
                    Test passes!
===== Welcome to xioateng Receiver ! =====
* Listening on port 33124...
the sum of package is: 4
===== Hello,Welcome to xiaoteng Sender! =====
Three Handshake Success !
TRANSMITED: 1
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0.489
ACK: 2
0.0
TIMEOUT!
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0.478
ACK: 2
0
TIMEOUT!
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0.479
ACK: 2
0.479
ACK: 2
DUP ACK: 2
TRANSMITED: 2
0.468
ACK: 5
TEAR DOWN THIS WALL
TEAR DOWN THIS WALL
TEAR DOWN THIS WALL
-------------------Test passes!------------------
```

TimeoutTest：

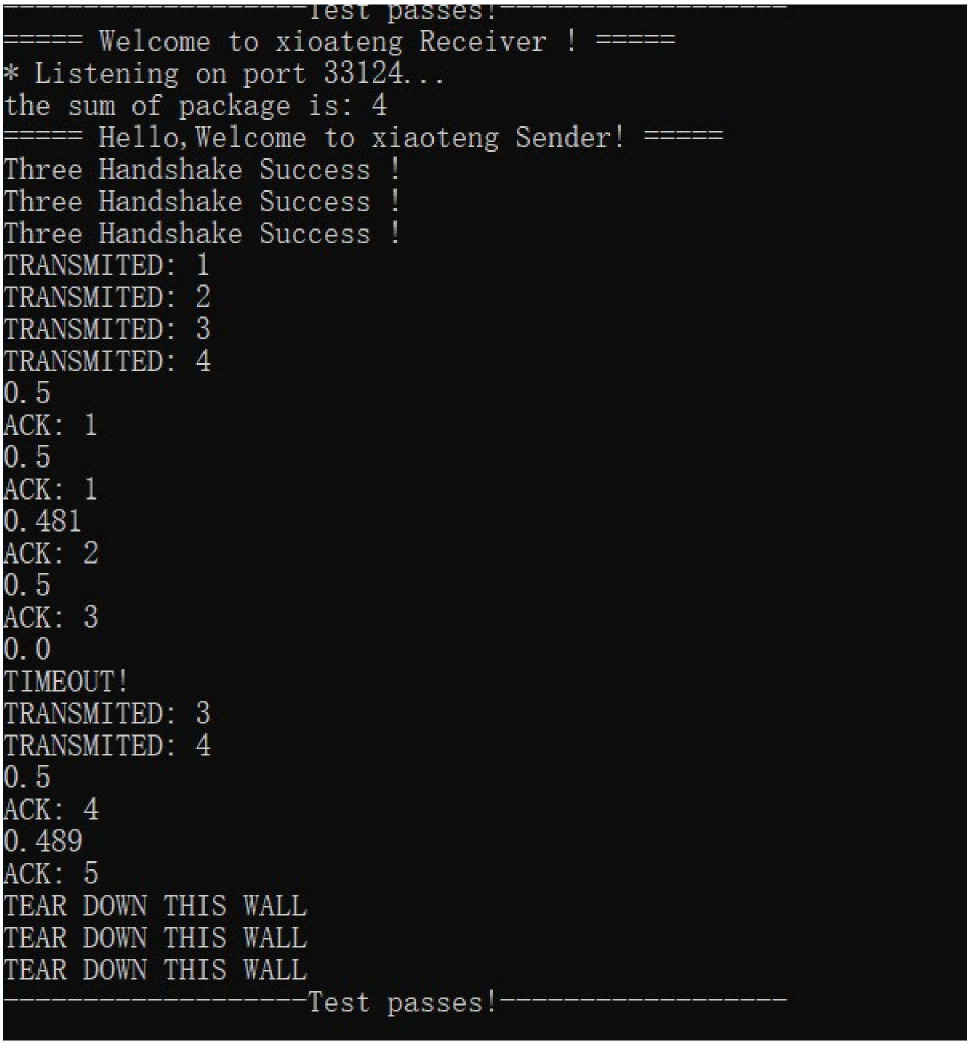


```
                   Test passes!
===== Welcome to xioateng Receiver ! =====
* Listening on port 33124...
the sum of package is: 4
===== Hello,Welcome to xiaoteng Sender! =====
Three Handshake Success !
Three Handshake Success !
Three Handshake Success !
Three Handshake Success !
Three Handshake Success !
Three Handshake Success !
Three Handshake Success !
Three Handshake Success !
Three Handshake Success !
Three Handshake Success !
Three Handshake Success !
Three Handshake Success !
Three Handshake Success !
TRANSMITED: 1
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0.0
TIMEOUT!
TRANSMITED: 1
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0
TIMEOUT!
TRANSMITED: 1
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0
TIMEOUT!
TRANSMITED: 1
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0.0
```



```
TRANSMITED: 4
0
TIMEOUT!
TRANSMITED: 1
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0.0
TIMEOUT!
TRANSMITED: 1
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0
TIMEOUT!
TRANSMITED: 1
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0
TIMEOUT!
TRANSMITED: 1
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0.401
ACK: 4
0
TIMEOUT!
TRANSMITED: 4
0
TIMEOUT!
TRANSMITED: 4
0.0
TIMEOUT!
TRANSMITED: 4
0
TIMEOUT!
TRANSMITED: 4
0.456
ACK: 5
TEAR DOWN THIS WALL
TEAR DOWN THIS WALL
TEAR DOWN THIS WALL
TEAR DOWN THIS WALL
TEAR DOWN THIS WALL
TEAR DOWN THIS WALL
-------------------Test passes!------------------
```

lowTimetext：



```
-------------------Test passes!------------------
===== Welcome to xioateng Receiver ! =====
* Listening on port 33124...
the sum of package is: 4
===== Hello,Welcome to xiaoteng Sender! =====
Three Handshake Success !
Three Handshake Success !
Three Handshake Success !
TRANSMITED: 1
TRANSMITED: 2
TRANSMITED: 3
TRANSMITED: 4
0.5
ACK: 1
0.5
ACK: 1
0.481
ACK: 2
0.5
ACK: 3
0.0
TIMEOUT!
TRANSMITED: 3
TRANSMITED: 4
0.5
ACK: 4
0.489
ACK: 5
TEAR DOWN THIS WALL
TEAR DOWN THIS WALL
TEAR DOWN THIS WALL
-------------------Test passes!------------------
```

第七步

滑动窗口实现动态，经过查阅书籍，了解到动态窗口需要结合触发式ack进行处理，我做不完啦~~~;