Introduction to Computer Networks

Assignment 3

1. Working environment

   1) Windows environment

      Windows 10 Home (version 1709)

   2) Programming Language

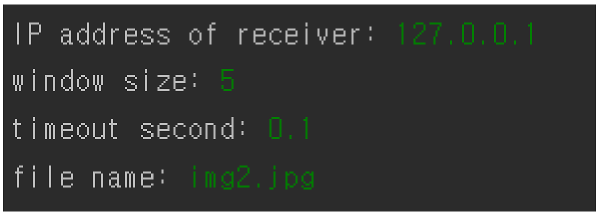
      Python3

   3) compilers/interpreter versions

      PyCharm 2018.1 (32bit)

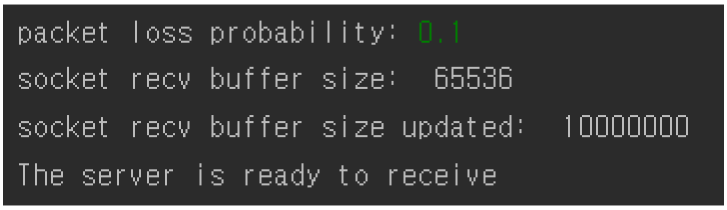
2. How to run the program

   Server 파일을 실행하고 Receiver의 IP주소와 윈도우 사이즈, timeout(초), 파일이름을 입력한다.



```
IP address of receiver: 127.0.0.1
window size: 5
timeout second: 0.1
file name: img2.jpg
```

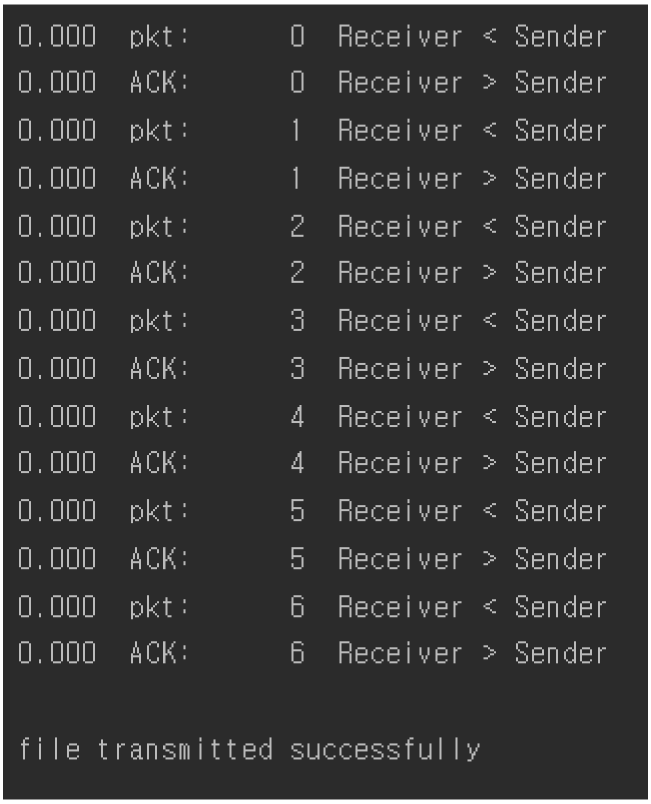
   Receiver 파일을 실행하고 드롭 비율을 입력한다.



```
packet loss probability: 0.1
socket recv buffer size:  65536
socket recv buffer size updated:  10000000
The server is ready to receive
```

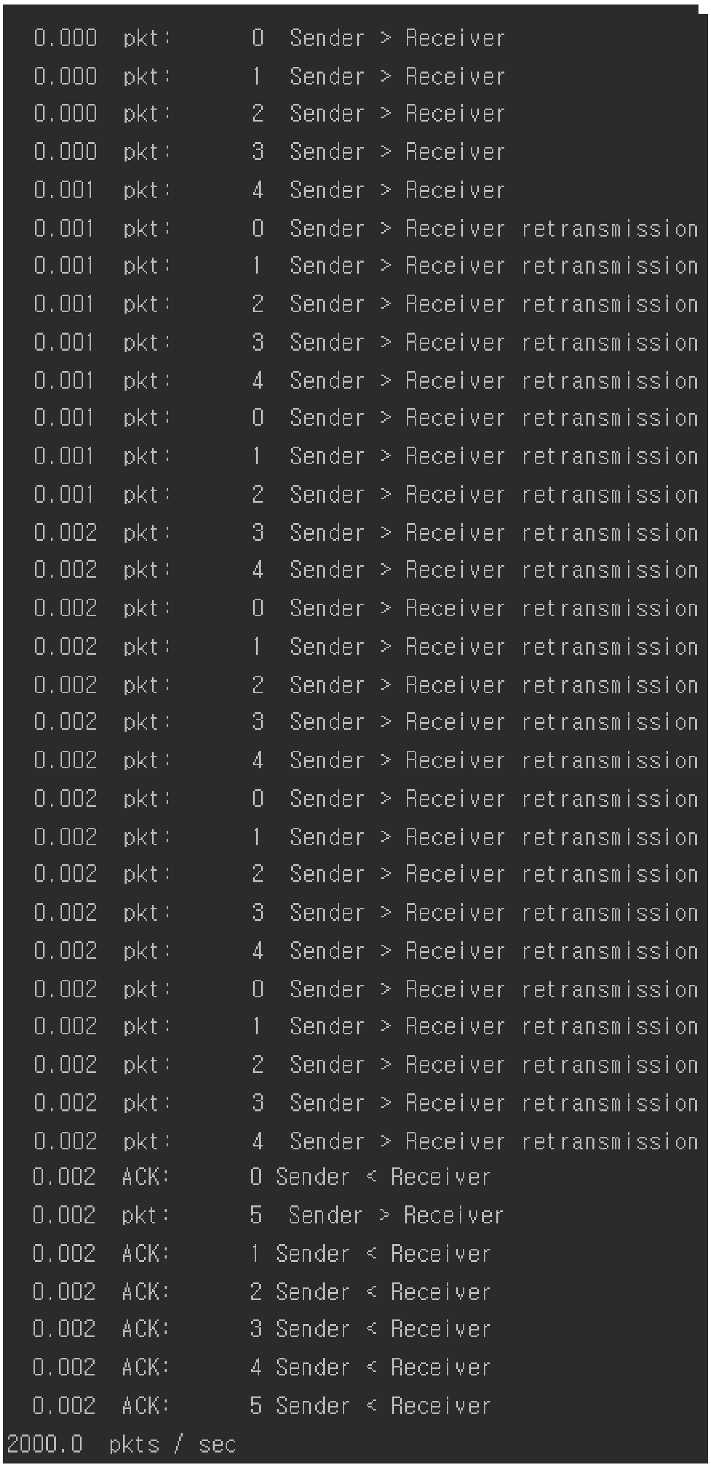
   다음은 실행 결과이다.

   Receiver 측



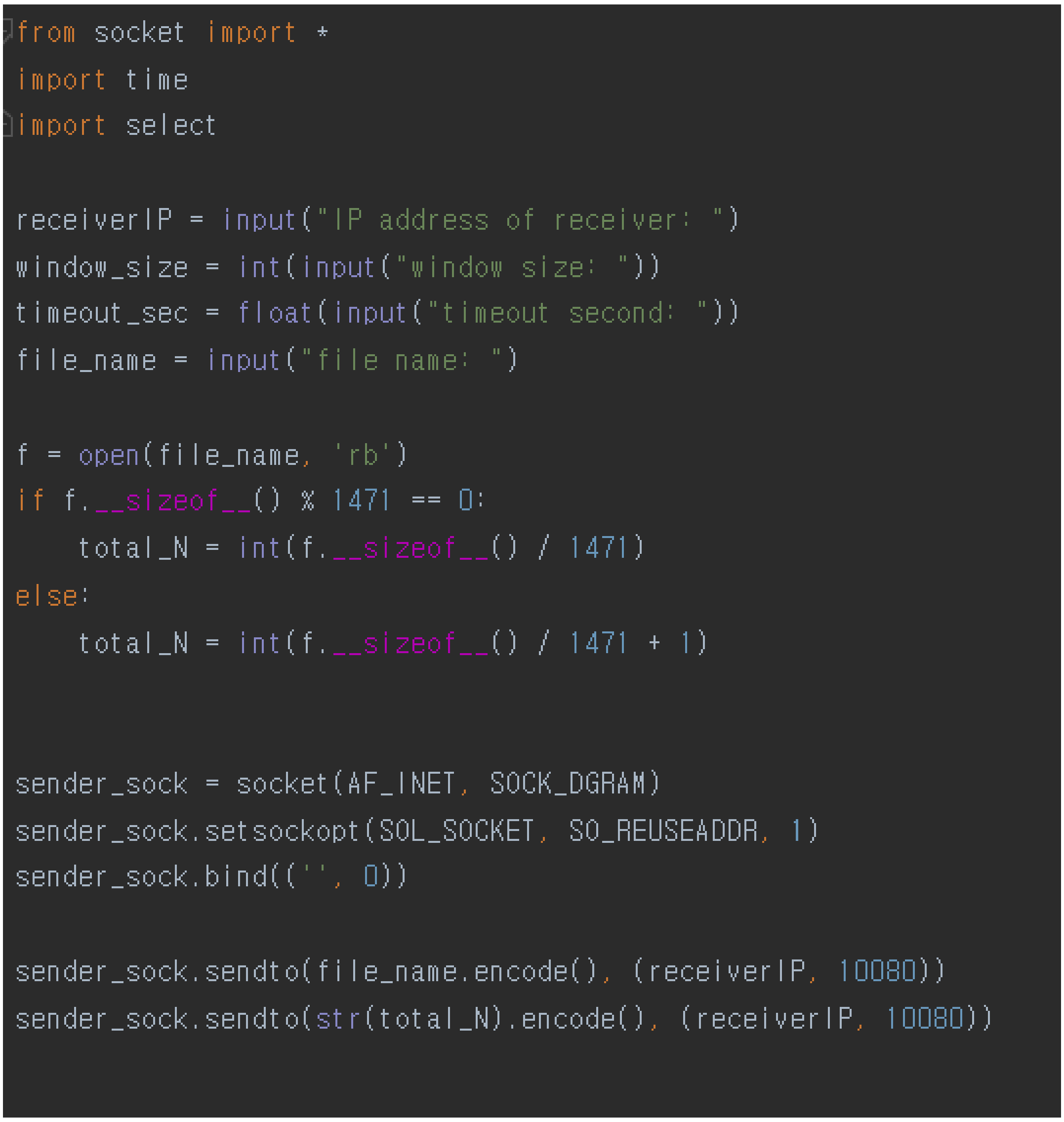
```
0.000  pkt:        0  Receiver < Sender
0.000  ACK:        0  Receiver > Sender
0.000  pkt:        1  Receiver < Sender
0.000  ACK:        1  Receiver > Sender
0.000  pkt:        2  Receiver < Sender
0.000  ACK:        2  Receiver > Sender
0.000  pkt:        3  Receiver < Sender
0.000  ACK:        3  Receiver > Sender
0.000  pkt:        4  Receiver < Sender
0.000  ACK:        4  Receiver > Sender
0.000  pkt:        5  Receiver < Sender
0.000  ACK:        5  Receiver > Sender
0.000  pkt:        6  Receiver < Sender
0.000  ACK:        6  Receiver > Sender

file transmitted successfully
```

Sender 측



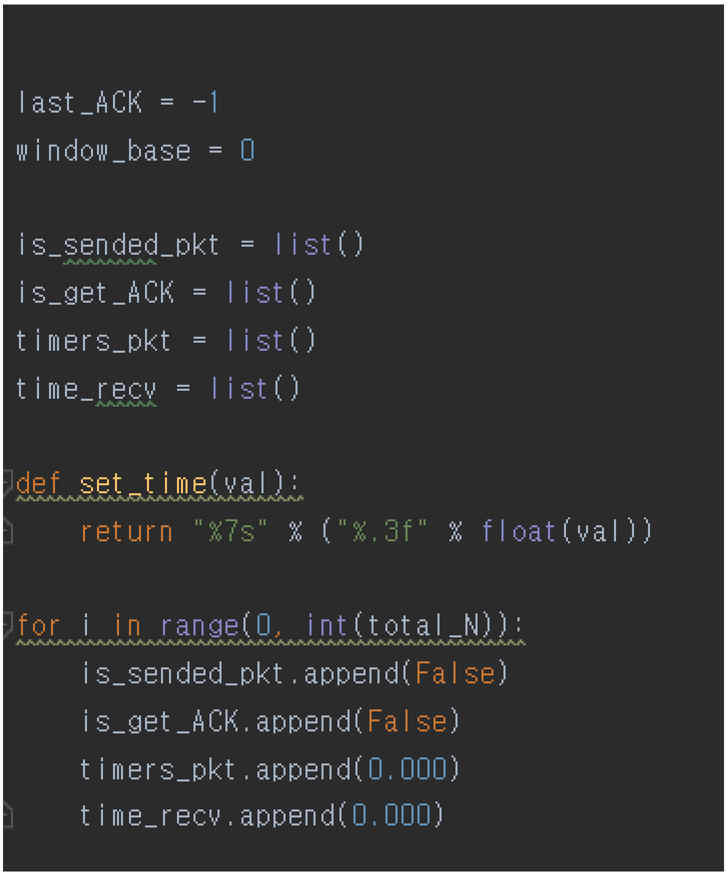
```
  0.000  pkt:      0  Sender > Receiver
  0.000  pkt:      1  Sender > Receiver
  0.000  pkt:      2  Sender > Receiver
  0.000  pkt:      3  Sender > Receiver
  0.001  pkt:      4  Sender > Receiver
  0.001  pkt:      0  Sender > Receiver retransmission
  0.001  pkt:      1  Sender > Receiver retransmission
  0.001  pkt:      2  Sender > Receiver retransmission
  0.001  pkt:      3  Sender > Receiver retransmission
  0.001  pkt:      4  Sender > Receiver retransmission
  0.001  pkt:      0  Sender > Receiver retransmission
  0.001  pkt:      1  Sender > Receiver retransmission
  0.001  pkt:      2  Sender > Receiver retransmission
  0.002  pkt:      3  Sender > Receiver retransmission
  0.002  pkt:      4  Sender > Receiver retransmission
  0.002  pkt:      0  Sender > Receiver retransmission
  0.002  pkt:      1  Sender > Receiver retransmission
  0.002  pkt:      2  Sender > Receiver retransmission
  0.002  pkt:      3  Sender > Receiver retransmission
  0.002  pkt:      4  Sender > Receiver retransmission
  0.002  pkt:      0  Sender > Receiver retransmission
  0.002  pkt:      1  Sender > Receiver retransmission
  0.002  pkt:      2  Sender > Receiver retransmission
  0.002  pkt:      3  Sender > Receiver retransmission
  0.002  pkt:      4  Sender > Receiver retransmission
  0.002  pkt:      0  Sender > Receiver retransmission
  0.002  pkt:      1  Sender > Receiver retransmission
  0.002  pkt:      2  Sender > Receiver retransmission
  0.002  pkt:      3  Sender > Receiver retransmission
  0.002  pkt:      4  Sender > Receiver retransmission
  0.002  ACK:      0 Sender < Receiver
  0.002  pkt:      5  Sender > Receiver
  0.002  ACK:      1 Sender < Receiver
  0.002  ACK:      2 Sender < Receiver
  0.002  ACK:      3 Sender < Receiver
  0.002  ACK:      4 Sender < Receiver
  0.002  ACK:      5 Sender < Receiver
2000.0  pkts / sec
```

3. Algorithm design



```
from socket import *
import time
import select

receiverIP = input("IP address of receiver: ")
window_size = int(input("window size: "))
timeout_sec = float(input("timeout second: "))
file_name = input("file name: ")

f = open(file_name, 'rb')
if f.__sizeof__() % 1471 == 0:
    total_N = int(f.__sizeof__() / 1471)
else:
    total_N = int(f.__sizeof__() / 1471 + 1)


sender_sock = socket(AF_INET, SOCK_DGRAM)
sender_sock.setsockopt(SOL_SOCKET, SO_REUSEADDR, 1)
sender_sock.bind(('', 0))

sender_sock.sendto(file_name.encode(), (receiverIP, 10080))
sender_sock.sendto(str(total_N).encode(), (receiverIP, 10080))
```

먼저, 입력 값을 받고, 파일을 연다. 몇 번에 나누어 보낼 것인지 알기 위해서 total_N을 구한다. 그 다음, 소켓을 연다. 포트번호는 10080이다.



```
last_ACK = -1
window_base = 0

is_sended_pkt = list()
is_get_ACK = list()
timers_pkt = list()
time_recv = list()

def set_time(val):
    return "%7s" % ("%.3f" % float(val))

for i in range(0, int(total_N)):
    is_sended_pkt.append(False)
    is_get_ACK.append(False)
    timers_pkt.append(0.000)
    time_recv.append(0.000)
```

마지막으로 받은 ACK을 저장하는 last_ACK은 -1부터 시작하고, 보내기에 성공할 시에 +1씩 증가한다.

각 패킷에 대하여 보낸 적이 있는지, ACK을 받았는지, 패킷을 보낸 시간, ACK을 받은 시간을 저장하는 리스트를 선언하고 초기화한다.