Report

1. Development Environment information

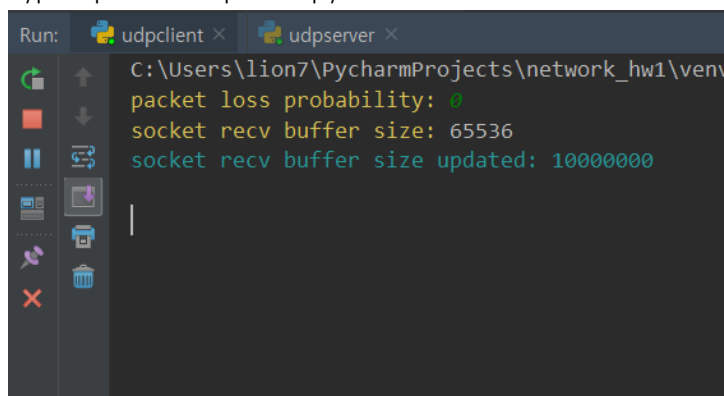Version of operating system: window10

Languages: python

Compiler/interpreter version: python 3.6
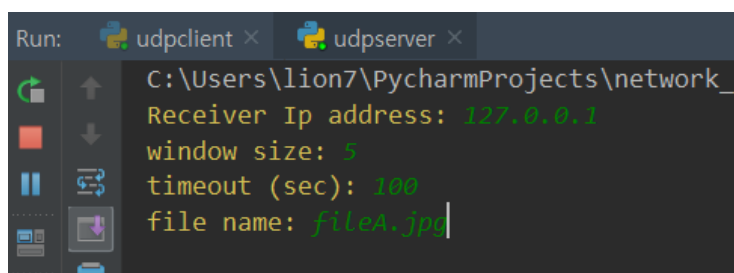
2. How to run sender and receiver programs.

<u>Please follow the order to run safely</u>

1. Run "udpclient.py" in directory 'PA3/client', first!!

2. Type input for "udpclient.py"

3. Next, run "udpserver.py" in directory 'PA3'

4. Type input for "udpserver.py"

It is recommended that the timeout value is set to 0.05 or higher.

5. See file transmission

[left: sender side, right: receiver side]

6. If you want to send another file after the file transmission, run "udpserver.py" again.

[left: receiver side(continue), right: sender side(new)]

3. Screen shot for several scenarios

1. receiver: dropped

2. sender: 3 duplicated acks

3. sender: time out

4. sender: throughput

4. How to design program

1. udpserver.py – sender

- get input from user

- divide object file and store as packet in *packets* (list)

- send packet until receive all acks

- when sending packet, mark some information
  (e.g. whether the packet was sent, when the packet was sent)

- Following the information, catch 3 duplicated acks or timeout event

- If you catch one of those above events, retransmission occurs

2. **udpclient.py – receive**

   ack number는 마지막으로 제대로 받은 packet number로 구성했습니다

   아무것도 제대로 받지 못했을 때 ack를 보낸다면, -1을 보내는 것으로 구성했습니다.

   - get input from user

   - update receive buffer size, if needed

   - receive file information
     (e.g. file name, number of total packets)

   - receive file as packet

   - if the received packet is in order, send received packet number
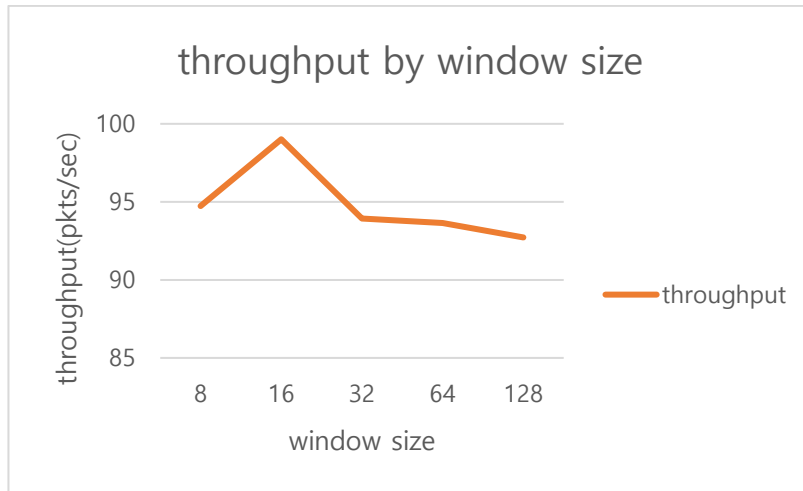
5. Show two graph for the experimentation results

experimented with fileC.mp4(2.23MB)

1. a goodput graph with different probabilities of packet loss
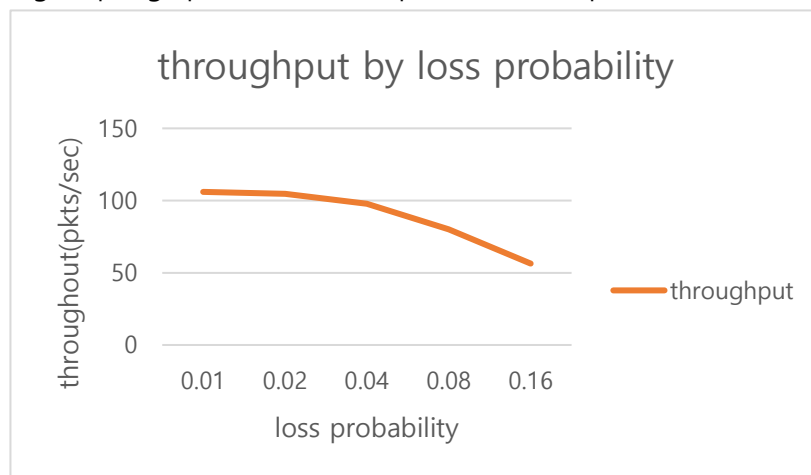
Throughput is decreasing as loss probability grows

2. a goodput graph with different window sizes

The best window size would be 16 in above conditions.