Name: Chim Ka Long

SSID: 1155094482

**For steps 1~6, I will descript my student code briefly.**

Firstly, initialize variable, assuming we have known the IP address and the port number of destination server. In this step, we assume the server is located on local host with port 3310.

We create TCP socket s1 and connect to server, send SID to server and receive 5 char string(distport2) which is the port number of our server in s2 TCP connection.

Then we create TCP connection as server side and bind (localhost, int(distport2)), remember that the first parameter is string, but port number is int.

We listen to the socket and accept 1 connection, after receiving the first connection, we close the listen socket, since there should be no more connection.

We receive the 12 chars string, including UDP receiving port number (data2) and sending port (data1). We do some string method to separate the two port and change them to int type.

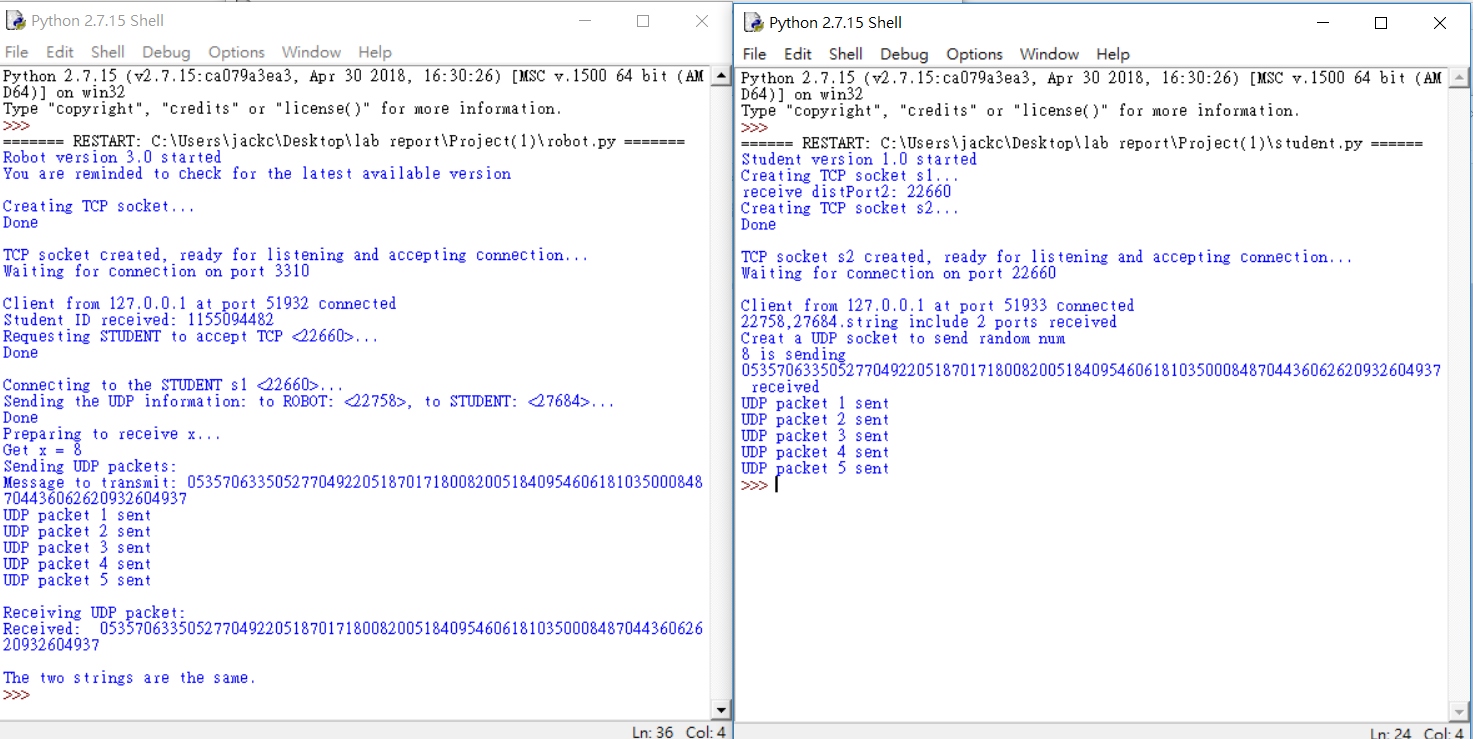
We generate a random number (rnd) between 6 and 9. Then start to create UDP socket s3 and bind data2. Wait for 3 seconds and send random number out. If we send out the number before the robot binding to relative port before, the packet will loss, so that it is safe for wait 3 seconds.

Then, the robot will send a string with rnd\*10 chars to us five times. We need to receive them with a while loop. And send the string received back to robot with also five times to prevent package loss.

If the robot receives our string same as the string it generates before, the code is successful and end.

After that, I tried to connect to my friend’s computer with smartphone as Mobile personal hotspot WI-FI network. Firstly, close the firewall. Then I act as student can change the destination IP to his IP address, and he acts as robot. The code is also fine.

**The below graph is the output of robot(left) and student(right).**



**For steps 7:**

Small buffer size will decrease performance seriously (we use throughput to determine performance here), but over large buff size will also decrease performance.

If buffer size is full of data and more packets come in, the computer cannot store them and causes packet loss which is the worst case. To avoid such cases, there is flow control that the receiver can tell transmitter the buffer size is almost full and do not send more data come in. Small buffer size is easy to full. However, occurrence of flow control means the CPU is not fully utilized, since the transmitter can send more data within same times if the buffer size is larger enough. As a result, the buffer size should be larger enough to avoid flow control to interrupt the normal data transfer operation.

However, if the buffer size is too large, the performance will also decrease in smaller degree due to increase in paging. The data in buffer will be processed until the amount of data is accumulated to certain level which is proportional to buffer size, or time out occur. If buffer size is too large, the certain level is also increase, and performance further decrease.

As a result, we should determine a suitable buffer size to avoid flow control and paging problem. The default buffer size (56636 bytes) is efficient for most cases.

**Step 8: (we test throughput with buffer size form 10 bytes to 15000bytes)**

From the above graph, while buffer size is located between 0 and 10000, increase in buffer size dramatically increase throughput. We deduce that the throughput is low since there is lot of flow control interrupt the data transfer operation. The maximum throughput achieved when buffer size is default (65536 bytes), but there is no huge difference with buffer size among 20000 and 100000. After maximum throughput, when buffer size continues to increase, the throughput decrease since there is paging increase.

**Additional question:**

In a low-speed network environment, the CPU in both sides cannot be utilized, since the network is too slow and CPU cannot keep sending data due to network speed. As a result, even the buffer size is not too large, the throughput is still low. For example, there is dramatically positive relationship when buffer size is among 10 bytes to 1000 bytes. After that, the line in graph is similar to horizontal line.