IERG 3310 Project Report

Name: Wong Deshun

SID: 1155190833

Step 1: Run Robot.exe

Step 2: I will be creating a new TCP socket on the Student side and then make a connection thereafter. After Student connect to the Robot, it will send the 10 digit Student ID to the Robot via the send() method.

Text

Description automatically generated

Step 3: Robot will be sending the 5 char string ddddd to student and the student will receive the message by calling recv()

Text

Description automatically generated

Thereafter, the string that is received can be used to create a new TCP socket s2

Text

Description automatically generated

Step 4: Robot will send 12 char string ‘fffff,eeeee”, Similar to step 3, student will receive the message by calling recv(). As the data come in 2 parts, I will split the data into iUDPPortRobot and iUDPPortStudent respectively.

**Text

Description automatically generated**

Using the random number library in python, I will generate a random number and send it to the Robot. Thereafter, I will create a new UDP socket s3 with port number eeeee to receive num \* 10 string from Robot.

Text

Description automatically generated

Step 5: Calling sendto() to transmit the string for 5 times to the Robot

Text

Description automatically generated

Step 6: Running Robot and Student on two different machines (Student6.py on the left, Robot6.py on the right,)

Text

Description automatically generatedText

Description automatically generated

Step 7:

We obtain Student’s receiver buffer size and send it to the Robot

Text

Description automatically generated

To extract the receiver buffer size on the robot’s end, we call recv()

Text

Description automatically generated

Afterward in the next 30 second, I use a while loop to transmit a large number of messages to Student

Text

Description automatically generated

Similarly, the Student will use a while loop to receive and count the message

Text

Description automatically generated

Screenshot of the print out by robot.py and student.py respectively

Text

Description automatically generated

Text

Description automatically generated with low confidence

Step 8:

Similar to Step 7, but I will vary the buffer size into [10, 50, 100, 500, 2000, 5000, 10000] bytes accordingly. In addition, we need to reset the socket so that the address and port number can be reused

Text

Description automatically generated

|  |  |  |
| --- | --- | --- |
| buffer size | total bytes | system throughput |
| 10 | 108589868 | 3619662.267 |
| 50 | 151318038 | 5043934.6 |
| 100 | 150120378 | 5004012.6 |
| 500 | 144604472 | 4820149 |
| 2000 | 138256476 | 4608549.2 |
| 5000 | 86268044 | 2875601.5 |
| 10000 | 63250410 | 2108347 |
| 20000 | 61309670 | 2043655.7 |

As the buffer size increase, the system throughput also increases. If the message size is changed, the optimal receiver buffer size is changed as well. 50 is the optimal buffer size in this situation.

What is the limitation of the experiment set-up since we do it in IE Common Lab? (Hint: What if the network is a low-speed one? )

Some packet may be lost. It makes the student received less packets and affect the throughput. Hence under a slow network, the throughput results are affected as well. Hence, we cannot find the optimal receiver buffer size.