

Report on the Socket Programming Project

Hugo Hue 11/23/2018

Basic Programming Part:

The program has to interact with ROBOT according to the following steps:

1. Start the ROBOT by running “robot.exe” in a command prompt.
2. When the ROBOT is started, a message “ROBOT IS STARTED” will be printed, indicating that the ROBOT is now listening on TCP Port 3310. STUDENT has to connect to the ROBOT TCP Port 3310 and send your 10 char student ID string via the connection established.
3. The ROBOT will then send a 5 char string *dddd* to the STUDENT. STUDENT will need to create a TCP socket *s_2* at port *dddd* to accept a new connection. The ROBOT will initiate the new connection 1 second later after sending *dddd*.

ROBOT will then send a 12 char string “*fffff,eeee*.” to the STUDENT using the new connection. STUDENT needs to decode the message and create a UDP socket *s3* to send a randomly generated variable *num* ($5 < num < 10$) to ROBOT on port *fffff*. Then ROBOT will send a char string *xxx* with length $num * 10$ to STUDENT one second after receiving *num* and STUDENT will receive the string using *s3* on port *eeee* .

5. When the STUDENT received the char string *xxx*, it will send back the string to the ROBOT at UDP port *fffff*. Similar to the ROBOT, the string will be sent 5 times, once every 1 second. The

For the robot side (server)

```
huehugodeMacBook-Pro:3310 hugo$ python robot.py
ROBOT IS STARTED
Robot version 3.0 started
You are reminded to check for the latest available version

Creating TCP socket...
Done

TCP socket created, ready for listening and accepting connection...
Waiting for connection on port 3310

Client from 10.6.57.243 at port 60554 connected
Student ID received: 1155079446
Requesting STUDENT to accept TCP <29342>...
Done

Connecting to the STUDENT s1 <29342>...
Sending the UDP information: to ROBOT: <22945>, to STUDENT: <20744>...
Done
Preparing to receive x...
Get x = 9
Sending UDP packets:
Message to transmit: 026760516004311072200821701624077950215409625021230874400249029580318302791071020164405582
UDP packet 1 sent
UDP packet 2 sent
UDP packet 3 sent
UDP packet 4 sent
UDP packet 5 sent

Receiving UDP packet:
Received: 026760516004311072200821701624077950215409625021230874400249029580318302791071020164405582

The two strings are the same.
```

For the student side (client)

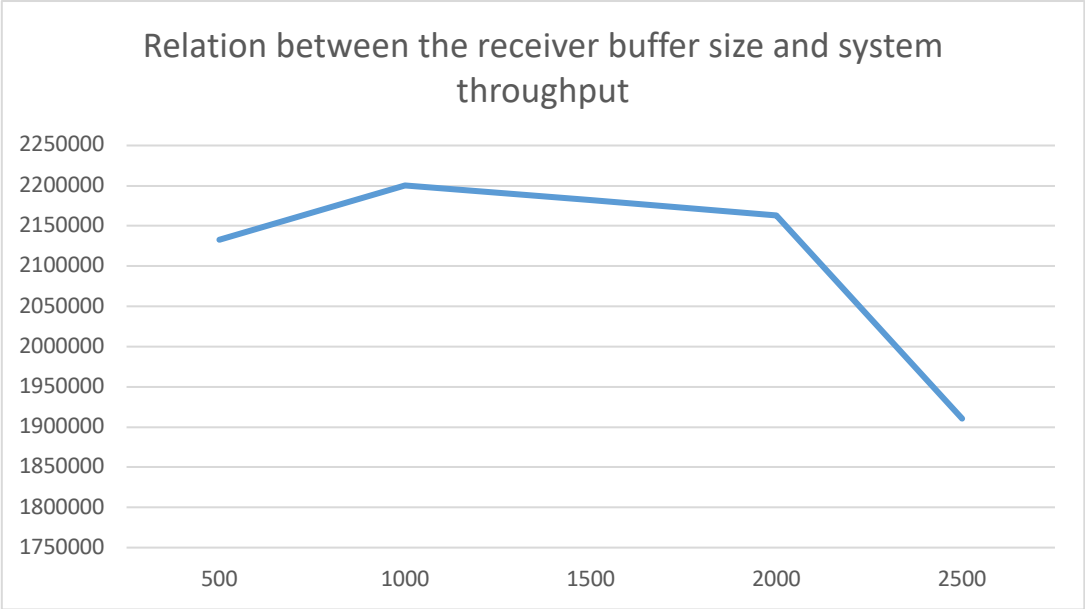
```
huehugodeMacBook-Pro:3310 hugo$ python student.py
Got connection from ('10.6.57.243', 60555)
22945 20744
026760516004311072200821701624077950215409625021230874400249029580318302791071020164405582
```

Two port numbers has been printed out to confirm the correctness of the receiving and decoding.
The packet is also been printed out for double-checked.

Experiment Part:

- Studying how and why the receiver buffer size affects the performance

Graph showing the relation between the **receiver buffer size** and the **system throughput**,



The blue line is the data of the system throughput sampled in a range from 500 to 2500 bytes of receiver buffer sizes. **Remark: System throughput = total received bytes / duration**

From the graph, it shows that the larger receiver buffer size we have, the lower system throughput we get.

Total size: 61786150bytes	Receiver buffer size	Size	Time	System throughput
Total packet: 899274	500	61786150	28.96630979	2133034.91
Total time: 28.9663097858	1000	65981900	29.98746181	2200316.266
	1500	63218970	28.97449589	2181883.345
Total size: 65981900bytes	2000	64799050	29.95285511	2163368.058
Total packet: 934628	2500	57288820	29.98757935	1910418.288
Total time: 29.9874618053				
Total size: 63218970bytes	Receiver buffer size	System throughput		
Total packet: 849710	500	2133034.91		
Total time: 28.9744958878	1000	2200316.266		
	1500	2181883.345		
Total size: 64799050bytes	2000	2163368.058		
Total packet: 879691	2500	1910418.288		
Total time: 29.9528551102				
Total size: 57288820bytes				
Total packet: 821054				
Total time: 29.9875793457				

The raw data collected from the programs.

What is the limitation of the experiment set-up since we do it in IE Common Lab?

If the network is in low-speed, the effects of the buffer size are low to the throughput, and thus it hard to see the relationship between throughput and the buffer size.