CN3: EXERCISE 2

Submission deadline: 2019/10/28

(Please submit your report as one PDF file to taizucourse@gmail.com)

Lab questions:

Note:

- *This programming exercise should be done by **a team of two students**, each runs one host. **Please find a partner for yourself**. However, each student should write his own report.*
- *If you want to do it alone, please run client and server on two different terminal windows on your host.*
- *In your answers, you should show the **new/modified code lines.** Also, show the **results** by the screen capture.*
- *Please see Hints at the end of this document*

Please run the client and server (UDP) programs provided in our folder. Each program is run in a terminal window.

1- Please connect these two programs (by assigning IP address and port number). Show the screen capture of the client after it has received the modified message from the server. (15 points)

2- Modify the source code of the server so that it can display the source IP address and source port number of the client on the server's window. (15 points)

3- Open one more terminal window and run the client program in this new window. Show that the server can process messages from both clients (by showing screen captures of clients and server). (10 points)

4- Modify the source code of the client so that it always waits for new text input. (10 points)

5- Modify the client of question 4, so that it adds a sequence number to the beginning of each message. This value is the order (e.g. 1, 2, 3,…, N with N $\leq$9) of each message from the client. Also, modify the server to show the **sequence number** on the **server**'s window. (10 points)

Homework questions (each question = 5points)

Please answer the following questions:

1. How is the source port number of the client program assigned?
2. How is the source port number of the server program assigned?
3. Is it possible to use clientSocket.bind() method in the client program? Please explain.
4. What are multiplexing and demultiplexing?
5. Describe the possible techniques to provide reliable service over UDP.

6. What is a subnet?
7. What are the difference between IPv4 and IPv6?
8. What is the relation of UDP & RTP? Please explain the fields provided by RTP.

Note on programs

When you re-run the server program, you may get an error of ***"socket.error: [Errno 98] Address already in use"***. To fix it, please close the current terminal window and open another terminal window for the server program.

Hints for Lab questions:

Q1's Hint:

Step 1: Identify the IP address of your server
- Open a terminal window
- Type '`ifconfig -a`' into the terminal window and press 'Enter'.
The IP address is found as below.

```
std4dc39:Desktop d8192106[13]$ ifconfig -a
lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL> mtu 8232 index 1
        inet 127.0.0.1 netmask ff000000
net0: flags=100001000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4,PHYSRUNNING> mtu 1500 index 2
        inet 163.143.14.69 netmask ffffff00 broadcast 163.143.14.255
lo0: flags=2002000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv6,VIRTUAL> mtu 8252 index 1
        inet6 ::1/128
net0: flags=120002004841<UP,RUNNING,MULTICAST,DHCP,IPv6,PHYSRUNNING> mtu 1500 index 2
        inet6 fe80::cad3:ffff:feba:a455/10
```

Step 2: Assign IP address and port number
- In the client's source code, assign the IP address of your server to *serverName,* and the port number of your server to *serverPort*.

Step 3: Run the programs by executing the following commands.
- In the server's terminal window:

```
python UDPServer.py
```

- In the client's terminal window:

```
python UDPClient.py
```

Q2's Hint: In the server's source code, *clientAddress[0]* and *clientAddress[1]* store the IP address and the port number of the client.

Q4's Hint: use a *while* loop like the server.

Q5's Hint:
- To add a *seqNum* and a string *message,* you can use

```
addedMessage  = str(seqNum)+message
```

For example, *seqNum* = 2 and *message*= 'abc', then *addedMessage*= '2abc'.

- To extract the *seqNum* from a string *addedMessage,* you can use

```
seqNum   = addedMessage[0]
message = addedMessage[1:]
```