Group Members' Names: Nick Murrin and Danny Caballero

Professor: Professor Li

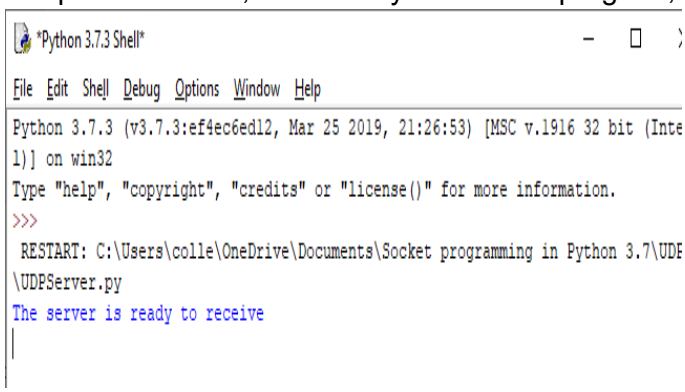
Class: CS 436 Section: MWF 11:30 AM -12:20 PM

Due Date: April 17th, 2019 at 11:55 PM

Project 1 Report:

In order to run the project, first download the folder called Socket Programming in Python 3.7, and then extract it. Danny is working on the UDPServer.py, and Nick is working on the UDPClient.py. After you extract the folder, go onto IDLE, open the file UDPServer.py, and run that program using the Python shell. Once the server program accepts the message and compiles and runs correctly, then run the client program on another computer, and run the client program. This is the backbones of the project. Specifically, we have 11 steps that we need to do.

For the first step, it's self-explanatory, as we ran the server's program first, and most of the code was provided to us, followed by the client's program, in which we deal with in step 2.



In step 2, for the client program, we extracted the MAC address, and since we thought that since an array list does the same thing, and we believed it was more efficient, we went ahead and used an array list to extract the MAC addresses. Since in step 2, it says that when the client sees the DISCOVER message, then we created a sendDiscover() method, which as mentioned, sends the DISCOVER message, as shown below.

```
['OFFER', '881FA11D3E14', '19216813']
['OFFER', '881FA11D3E14', '19216814']
```

In step 3 for the server program, when the server receives a DISCOVER message, it checks its current list whether any IP addresses has already been assigned to the client, for instance, the server checks the list whether the MAC address in the message exists in the list or not. Then, if

a record was found, the client needs to reply, indicating that this certain IP address has already been assigned to the client already. After that, we create a menu selecting these choices, but won't be covered entirely until the next steps. Then, If no record was found for this client, check whether the given pool of IP addresses has not been fully occupied by the current clients. Then, If the whole pool of IP addresses is already occupied by the clients, the server declines the request and replies a DECLINE message to the client. Otherwise, take the next available IP address and send an OFFER message to the client. This message contains the client's MAC address and the offered IP address. Additionally, if the client has received an IP address in the past, and the IP address has not been used for other clients yet, the server will assign that same IP address to the client. Otherwise, it will assign the next available IP address.

```
['DISCOVER', '881FA11D3E14', '']
```

In step 4 of the project, when the client receives an OFFER message, we checked to see if the client's MAC address is the same as the MAC address that the server encoded the client in the message that the server encoded to the client. If it was the same, then that OFFER message would be a REQUEST message. If the MAC address is different, then it would decline it by sending a DECLINE message.

```
['OFFER', '881FA11D3E14', '19216811']
```

Since it was the same, then this message appeared:

```
['REQUEST', '881FA11D3E14', '19216811']
```

In step 5, when the server receives a REQUEST message, we checked to see if the IP address is still there. If not, then the server offers a new IP address to the client. After that, it sends an ACKNOWLEDGE message containing the client's MAC address and the offered IP address. The IP address is still there, so this happened, along with the ACKNOWLEDGE message:

```
['REQUEST', '881FA11D3E14', '19216811']
['ACKNOWLEDGE', '881FA11D3E14', '19216811']
```

In step 6, When the client receives an ACKNOWLEDGE message, we check whether the MAC address in the message matches its own MAC address. If not , we displayed a proper message and terminate the program; Otherwise, we would display a message indicating that it got an IP address.

```
['ACKNOWLEDGE', '881FA11D3E14', '19216811']
```

Then, I made a menu allowing different options.

```
['OFFER', '881FA11D3E14', '19216811']
['ACKNOWLEDGE', '881FA11D3E14', '19216811']
To Release <re>
To Renew <rn>
To Quit <q>
Choice:
```

After that, in step 8, we checked to see if the server receives a RELEASE message, and if it does, it releases the IP address assigned to the client. The client will be removed from the list of current clients on the server. Then the server 4 responds back to the client. Once the client received this reply, it displays the menu for the user again. If the server could not find any

record for the client, it responds back to the client with a message indicating that the IP address has already been released. Then the client will display the menu again.

```
['RELEASE', '881FA11D3E14', '19216811']
```

In step 9, we checked to see if the server receives a RENEW message, and if it does, it first checks whether any IP address has been assigned to the client. If yes, the client replies, indicating that the client already has an IP address and the IP address is whatever the IP address currently is. Otherwise, the server offers an IP address and replies an OFFER message. Then the client should send a REQUEST. Then the server should assign an IP address and send an ACKNOWLEDGE. If the last IP address of the client is still available, assign the same IP address to the client. Otherwise, assign the next available IP address.

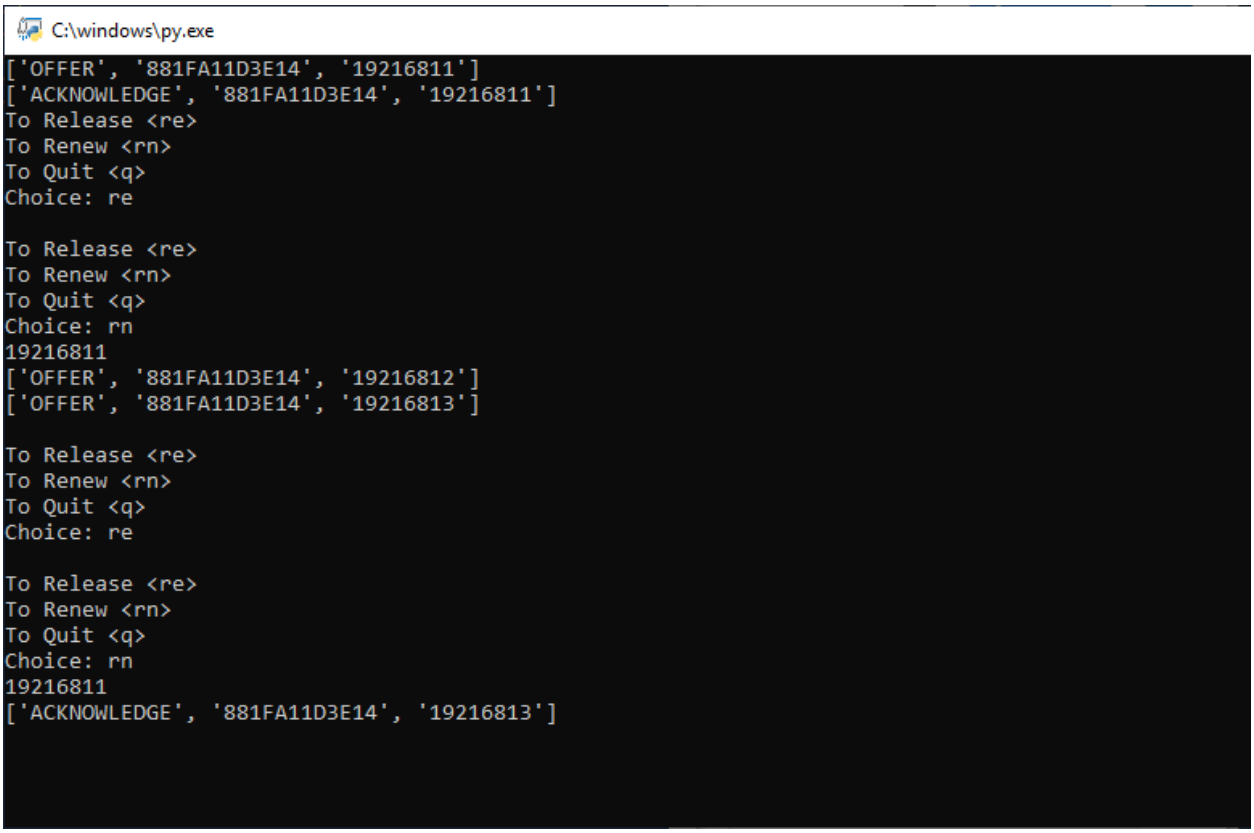
```
['RENEW', '881FA11D3E14', '19216811']
['REQUEST', '881FA11D3E14', '19216812']
['REQUEST', '881FA11D3E14', '19216813']
```

In step 10, we made sure that the server never assigns more than one IP address to a client at the same time or assigns the same IP address to more than one client. Then, we verified that when a client runs release and renew, it receives the same IP address as it used to have.

```
19216811
['ACKNOWLEDGE', '881FA11D3E14', '19216813']
```

Last, we made sure that the clients and the server displayed all feedback messages on the screen to demonstrate each step of the process.

```
The server is ready to receive
['DISCOVER', '881FA11D3E14', '']
['REQUEST', '881FA11D3E14', '19216811']
['DISCOVER', '881FA11D3E14', '']
881FA11D3E14
881FA11D3E14
19216811
OFFER|881FA11D3E14|19216811
['REQUEST', '881FA11D3E14', '19216811']
['REQUEST', '881FA11D3E14', '19216812']
['RELEASE', '881FA11D3E14', '19216811']
['RENEW', '881FA11D3E14', '19216811']
['REQUEST', '881FA11D3E14', '19216812']
['REQUEST', '881FA11D3E14', '19216813']
['RELEASE', '881FA11D3E14', '19216811']
```


C:\windows\py.exe
['OFFER', '881FA11D3E14', '19216811']
['ACKNOWLEDGE', '881FA11D3E14', '19216811']
To Release <re>
To Renew <rn>
To Quit <q>
Choice: re
To Release <re>
To Renew <rn>
To Quit <q>
Choice: rn
19216811
['OFFER', '881FA11D3E14', '19216812']
['OFFER', '881FA11D3E14', '19216813']
To Release <re>
To Renew <rn>
To Quit <q>
Choice: re
To Release <re>
To Renew <rn>
To Quit <q>
Choice: rn
19216811
['ACKNOWLEDGE', '881FA11D3E14', '19216813']