

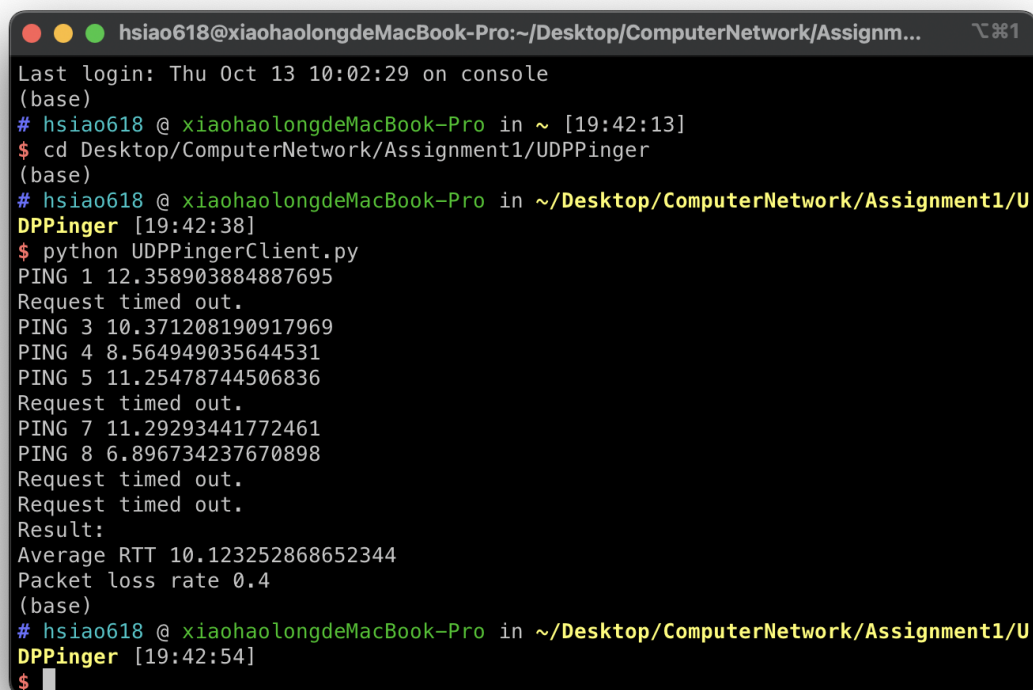
# Report of HW1

## Part 1: UDP Pinger

- Implement

First, I created a UDP socket to send messages to the server and receive the message from the server. I also set the variable to record the number of packets I sent and lost. Then I created a while loop to execute under the condition when the number of packets I sent is less than 10. In the while loop, I record the time, set timeout to 1 second and send the packet to the server, then use a try and except method to receive the reply message. If the socket receives the reply message, then records the time again to calculate the RTT; otherwise, print the timed out message.

- Result

A terminal window on a MacBook Pro showing the execution of a UDP Pinger client. The window title is 'hsiao618@xiaohaolongdeMacBook-Pro:~/Desktop/ComputerNetwork/Assignm...'. The terminal output shows the user running 'python UDPPingerClient.py', which sends 10 ping requests. The first, third, fourth, fifth, seventh, and eighth requests receive replies with RTT values. The second, sixth, and ninth requests time out. The final output shows an average RTT of 10.123252868652344 and a packet loss rate of 0.4.

```
hsiao618@xiaohaolongdeMacBook-Pro:~/Desktop/ComputerNetwork/Assignm...
Last login: Thu Oct 13 10:02:29 on console
(base)
# hsiao618 @ xiaohaolongdeMacBook-Pro in ~ [19:42:13]
$ cd Desktop/ComputerNetwork/Assignment1/UDPPinger
(base)
# hsiao618 @ xiaohaolongdeMacBook-Pro in ~/Desktop/ComputerNetwork/Assignment1/UDPPinger [19:42:38]
$ python UDPPingerClient.py
PING 1 12.358903884887695
Request timed out.
PING 3 10.371208190917969
PING 4 8.564949035644531
PING 5 11.25478744506836
Request timed out.
PING 7 11.29293441772461
PING 8 6.896734237670898
Request timed out.
Request timed out.
Result:
Average RTT 10.123252868652344
Packet loss rate 0.4
(base)
# hsiao618 @ xiaohaolongdeMacBook-Pro in ~/Desktop/ComputerNetwork/Assignment1/UDPPinger [19:42:54]
$
```

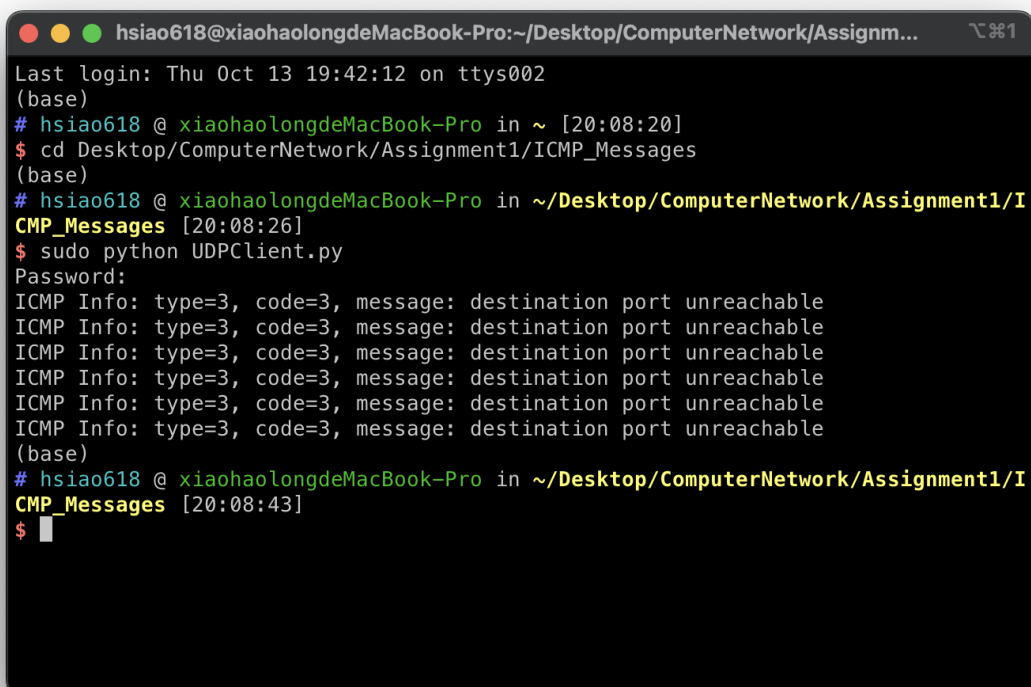
## Part 2: ICMP Messages

- Implement

First, I created a UDP socket to send packets to a closed server port and an additional ICMP socket to receive packets. After setting a timeout on the ICMP socket. I set a variable to count the packets I sent and let the execution of send packets via UDP socket in a while loop which under the condition of the packets I sent less than 10.

Moreover, I used the method in the hint which created an additional thread to receive the messages by using the ICMP socket. I've also imported the package called struct. Once the ICMP socket receives any packets, the *struct.unpack* function will unpack the message in the ICMP packet. After unpacking the message, we will get the message type. Finally, by looking at the ICMP message table which I searched from the Internet, we can print out the meaning of the message of the type.

- Result

A terminal window on a MacBook Pro showing the execution of a Python script. The user is in a directory named ~/Desktop/ComputerNetwork/Assignment1/ICMP\_Messages. They run 'sudo python UDPClient.py' and receive a password prompt. The script outputs six lines of ICMP information, all indicating 'destination port unreachable' with type=3 and code=3. The terminal window has a title bar with the user's name and the current directory path.

```
hsiao618@xiaohaoLongdeMacBook-Pro:~/Desktop/ComputerNetwork/Assignm...
Last login: Thu Oct 13 19:42:12 on ttys002
(base)
# hsiao618 @ xiaohaoLongdeMacBook-Pro in ~ [20:08:20]
$ cd Desktop/ComputerNetwork/Assignment1/ICMP_Messages
(base)
# hsiao618 @ xiaohaoLongdeMacBook-Pro in ~/Desktop/ComputerNetwork/Assignment1/ICMP_Messages [20:08:26]
$ sudo python UDPClient.py
Password:
ICMP Info: type=3, code=3, message: destination port unreachable
ICMP Info: type=3, code=3, message: destination port unreachable
ICMP Info: type=3, code=3, message: destination port unreachable
ICMP Info: type=3, code=3, message: destination port unreachable
ICMP Info: type=3, code=3, message: destination port unreachable
ICMP Info: type=3, code=3, message: destination port unreachable
(base)
# hsiao618 @ xiaohaoLongdeMacBook-Pro in ~/Desktop/ComputerNetwork/Assignment1/ICMP_Messages [20:08:43]
$
```

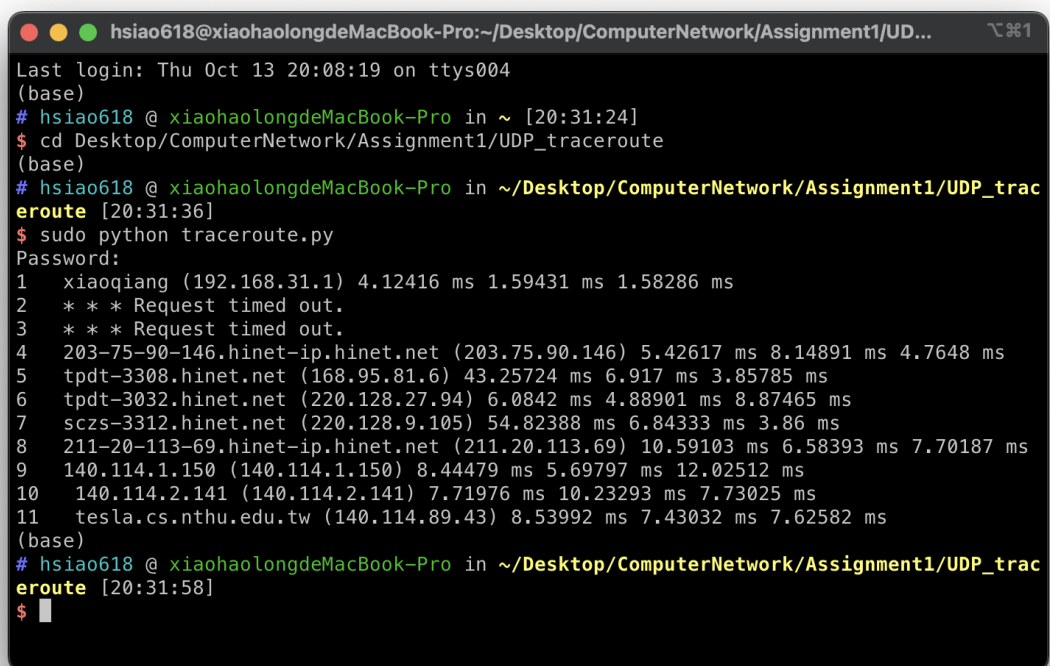
## Part 3: UDP Traceroute

- Implement

First, I set the variable to record the number of hops in order to make sure the number of hops didn't exceed the maximum of hops. I used a while loop to execute under the condition of the number of hops less than *maxHops*, then created an ICMP socket and a UDP socket. After creating the sockets, I send the packet out, set timeout to 1 then wait for the ICMP to receive a reply message. By using a while loop to track three times whether receiving the reply message or not. If the ICMP socket receives any replies, we should get the router's name by the address, if the router didn't have a name, then name it by the address, finally we can calculate the RTT and print it out; otherwise, print out a '\*' which indicates the packet has been lost. Then if we didn't receive any reply from the router after three tries, just print the timed out message.

Once the number of hops reach the maximum or the current address equal to the destination, then break the while loop.

- Result



```
hsiao618@xiaohaolongdeMacBook-Pro:~/Desktop/ComputerNetwork/Assignment1/UD...
Last login: Thu Oct 13 20:08:19 on ttys004
(base)
# hsiao618 @ xiaohaolongdeMacBook-Pro in ~ [20:31:24]
$ cd Desktop/ComputerNetwork/Assignment1/UDP_traceroute
(base)
# hsiao618 @ xiaohaolongdeMacBook-Pro in ~/Desktop/ComputerNetwork/Assignment1/UDP_traceroute [20:31:36]
$ sudo python traceroute.py
Password:
1  xiaoqiang (192.168.31.1) 4.12416 ms 1.59431 ms 1.58286 ms
2  * * * Request timed out.
3  * * * Request timed out.
4  203-75-90-146.hinet-ip.hinet.net (203.75.90.146) 5.42617 ms 8.14891 ms 4.7648 ms
5  tpd-t3308.hinet.net (168.95.81.6) 43.25724 ms 6.917 ms 3.85785 ms
6  tpd-t3032.hinet.net (220.128.27.94) 6.0842 ms 4.88901 ms 8.87465 ms
7  sczs-3312.hinet.net (220.128.9.105) 54.82388 ms 6.84333 ms 3.86 ms
8  211-20-113-69.hinet-ip.hinet.net (211.20.113.69) 10.59103 ms 6.58393 ms 7.70187 ms
9  140.114.1.150 (140.114.1.150) 8.44479 ms 5.69797 ms 12.02512 ms
10 140.114.2.141 (140.114.2.141) 7.71976 ms 10.23293 ms 7.73025 ms
11 tesla.cs.nthu.edu.tw (140.114.89.43) 8.53992 ms 7.43032 ms 7.62582 ms
(base)
# hsiao618 @ xiaohaolongdeMacBook-Pro in ~/Desktop/ComputerNetwork/Assignment1/UDP_traceroute [20:31:58]
$
```

## **The Problem I faced**

Although I've written python several times, I didn't have any experience of implementing any computer network application. So I think the problem I faced in the homework is that I'm unfamiliar with the "socket" package and the detailed procedure of computer networks, so I spent lots of time realizing the usage of each function. After finishing the homework, I think I can use the tool more efficiently and realize the deeper layer of computer networks.