# CSE 310: Computer Networks
# Homework #3

Ricky Lu
ID:112829937

Due Date: December 1, 2021

## Overall Implementation

For this assignment functions that require my implementation was doOnePing and receiveOnePing. For doOnePing all I needed to do was to create the RAW socket called mySocket. It was done by calling the socket constructor with parameter (socket.AF_INET, socket.socket_RAW, socket.IPPROTO_ICMP) to specify that we want a RAW socket with protocol ICMP. That was it for doOnePing, now for receiveOnePing I had to unpack the recPacket from the socket, meaning parsing apart the ICMP header. I know that the first 160 bit are not ICMP but rather IP header, hence I can safely skip 160 bit or jump straight to byte 20 knowing that that will be where the ICMP protocol starts. I then unpacked the ICMP header using the same format that I learned from sendOnePing, "bbHHhd" and calculated rtt by subtracting the time we received this packet versus the time this packet that sent, multiply by 1000 to get it to miliseconds. For rtt_sum we can just add the rtt we just found to it for calculating rtt average later. We increment rtt_cnt by 1 to indicate that we received a packet. For rtt_min and rtt_max we can just determine the running minimum and maximum by comparing it with rtt. I also found the time to live field from the IP header. Then return with all these information packed into a String for the while loop to print. Average rtt can be calculated by dividing rtt_sum by rtt_cnt.

## Part A

Testing sending packets to localhost 127.0.0.1

```
tamarine@pop-os:~/Desktop/CSE-310-Homework/HW3$ sudo python2 sample_pinger.py 127.0.0.1
Pinging 127.0.0.1 using Python:
16 bytes from 127.0.0.1 seq=1 ttl=64 rtt-time=0.101ms
16 bytes from 127.0.0.1 seq=1 ttl=64 rtt-time=0.064ms
16 bytes from 127.0.0.1 seq=1 ttl=64 rtt-time=0.071ms
16 bytes from 127.0.0.1 seq=1 ttl=64 rtt-time=0.067ms
16 bytes from 127.0.0.1 seq=1 ttl=64 rtt-time=0.140ms
^C--- 127.0.0.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max 0.064/0.089/0.140 ms
tamarine@pop-os:~/Desktop/CSE-310-Homework/HW3$
```

Figure 1: Pinging to localhost. Stopped after 5 transmissions by using ctrl + c

# Part B

I wasn't able to ping **stonybrook.edu** nor **cs.stonybrook.edu** while I was working on this assignment from home. I was able to ping **www.cs.stonybrook.edu**. Hence, instead of pinging those two I pinged **www.cs.stonybrook.edu** which I was able to hear back, and hope this is a okay alternative.

```
tamarine@pop-os:~/Desktop/CSE-310-Homework/HW3$ sudo python2 sample_pinger.py www.cs.stonybrook.edu
Pinging 23.185.0.2 using Python:
16 bytes from 23.185.0.2 seq=1 ttl=57 rtt-time=105.595ms
16 bytes from 23.185.0.2 seq=1 ttl=57 rtt-time=19.092ms
16 bytes from 23.185.0.2 seq=1 ttl=57 rtt-time=15.570ms
16 bytes from 23.185.0.2 seq=1 ttl=57 rtt-time=17.922ms
16 bytes from 23.185.0.2 seq=1 ttl=57 rtt-time=10.658ms
^C--- www.cs.stonybrook.edu ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max 10.658/33.767/105.595 ms
tamarine@pop-os:~/Desktop/CSE-310-Homework/HW3$
```

Figure 2: Pinging to www.cs.stonybrook.edu. Stopped after 5 transmissions by using ctrl + c

# Part C

The servers that I pinged comes from the public domain name server list: https://public-dns.info/

1. The first DNS that I pinged is in India in Burla which yielded the following result:

```
tamarine@pop-os:~/Desktop/CSE-310-Homework/HW3$ sudo python2 sample_pinger.py 223.31.121.171
Pinging 223.31.121.171 using Python:
16 bytes from 223.31.121.171 seq=1 ttl=49 rtt-time=278.837ms
16 bytes from 223.31.121.171 seq=1 ttl=49 rtt-time=275.472ms
16 bytes from 223.31.121.171 seq=1 ttl=49 rtt-time=276.545ms
16 bytes from 223.31.121.171 seq=1 ttl=49 rtt-time=341.993ms
16 bytes from 223.31.121.171 seq=1 ttl=49 rtt-time=292.010ms
^C--- 223.31.121.171 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max 275.472/292.971/341.993 ms
tamarine@pop-os:~/Desktop/CSE-310-Homework/HW3$
```

Figure 3: Pinging India, Burla DNS (Asia)

2. The second DNS that I pinged is in Poland in Warsaw which yielded the following result:

```
tamarine@pop-os:~/Desktop/CSE-310-Homework/HW3$ sudo python2 sample_pinger.py 217.17.34.10
Pinging 217.17.34.10 using Python:
16 bytes from 217.17.34.10 seq=1 ttl=52 rtt-time=115.121ms
16 bytes from 217.17.34.10 seq=1 ttl=52 rtt-time=111.413ms
16 bytes from 217.17.34.10 seq=1 ttl=52 rtt-time=117.957ms
16 bytes from 217.17.34.10 seq=1 ttl=52 rtt-time=114.821ms
16 bytes from 217.17.34.10 seq=1 ttl=52 rtt-time=205.725ms
^C--- 217.17.34.10 ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max 111.413/133.007/205.725 ms
tamarine@pop-os:~/Desktop/CSE-310-Homework/HW3$
```

Figure 4: Pinging Poland, Warsaw (Europe)

3. The third DNS that I pinged is in Brazil in Rio das Pedras which yielded the following result:

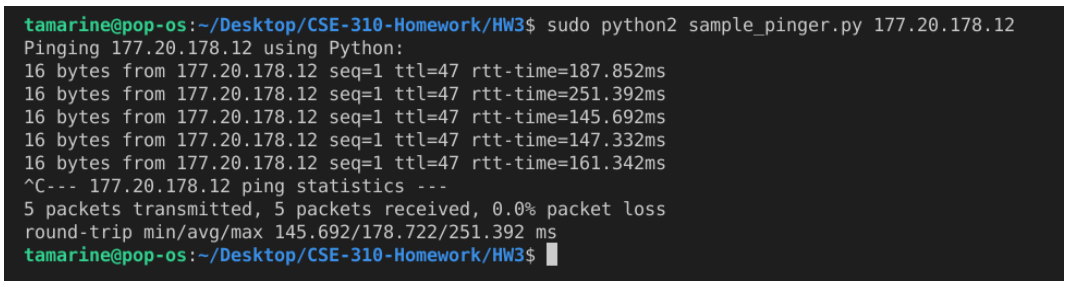


Figure 5: Pinging Brazil, Rio das Pedras (South America)

3. The last DNS that I pinged is in Egypt which yielded the following result:

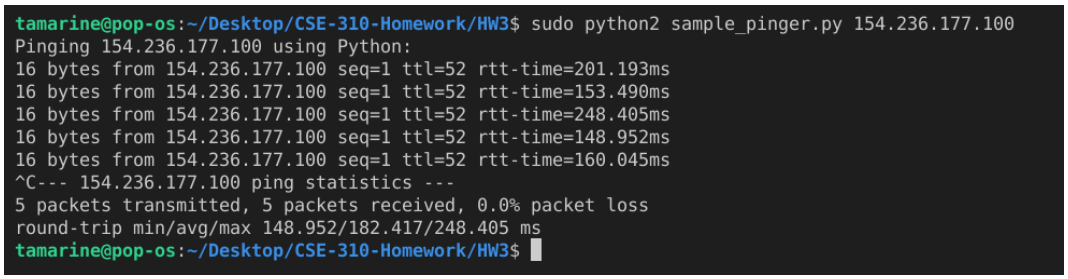


Figure 6: Pinging Egypt (Africa)

# Part D

As we can see the minimum round trip time from each part are as follows: 0.064ms, 10.658ms, and 275.472ms, 111.413ms, 145.692ms, 148.952ms respectively for each part. For part A since I'm pinging my own machine it should have the least amount of delay because it should take minimum time to reach myself and back, hence the low round trip time. For Stony Brook University's server, since I'm still within a reasonable close range to Stony Brook, in Queens of New York, the 110.658ms is a reasonable round trip time for a packet to be sent to Stony Brook server and come back to me, consider the fact that Queens to Stony Brook University is about 50 miles. However, when we take a look at the servers that I pinged which are from other continents, Asia, Europe, South America, and Africa respectively, we can see that the minimum round trip time are significantly greater because of the larger physical distances from New York to each of those servers in different continents.

# Part E and F

Program implemented to report the minimum, average, max round trip time as well as the percentage of packet loss after the user stopped the ping program.

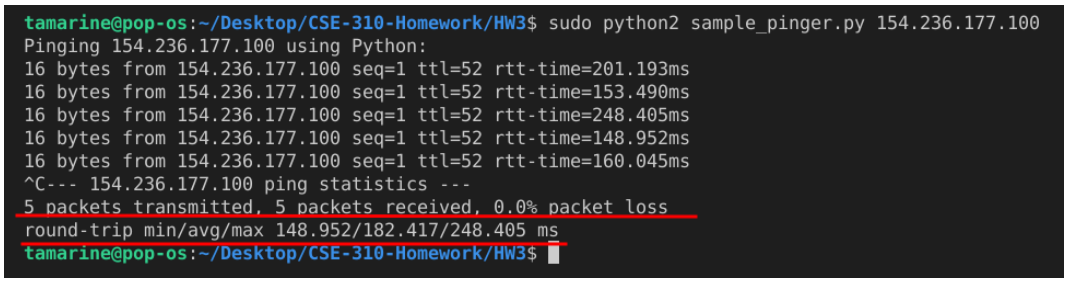


Figure 7: Statistical summary just like the ping program