Experiment- 3
UDP Pinger

Haricharan B
Electrical Engineering
Indian Institute of Technology, Madras
Chennai, India
ep21b015@smail.iitm.ac.in

***Abstract*—This experiment introduces Socket programming for UDP in Python through a Ping application. It includes a server and client components. The server simulates packet loss, while the client sends 10 Ping messages and calculates average, minimum, and maximum round-trip times. The client also has a time-out of one second, and if it takes any longer it is marked as a time-out.**

I. Procedure and Methodology

In the `UDPPingerServer.py` file provided in the question, we first create a server socket for UDP, and bind it to port 12000 (this needs to be used as the destination port for our client). Then, we wait for a message from the client. When a message is received, 30% of the packets are simulated to be lost, while the remaining 70% of the data is capitalized and sent back to the client.

In the file `UDPPingerClient.py` which we created, we again initialize a UDP socket. Here we've declared the source port number to be 8000. Also, a socket timeout of 1 seconds is declared. This is because there might be packet loss at the server side and the packet might not be received at the client side.

10 ping packets are initialized and sent to the server. The messages are of the format `Ping sequence_number time`. If a response is obtained, the message is printed, along with the round trip time (RTT), which is just `received_time` − `send_time`. If a response is not obtained within 1 second, a "Request timed out" message is printed.

As mentioned in the Optional Exercises, the minimum, maximum, average RTTs and the packet loss rates are also printed.

II. Observations

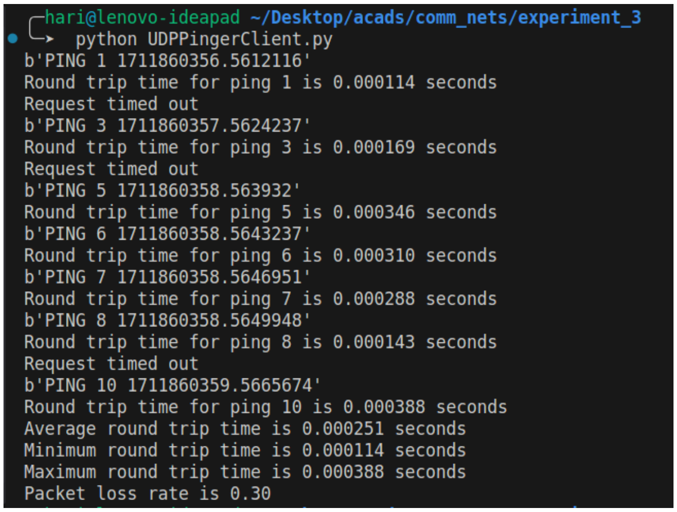
Pinging Local Host

In the 10 packets sent, it is observed that packets 2, 4, 9 have timed out while the other packets have been received successfully. It is also noted that when the response is being printed, the message is capitalized (in the message we sent, `Ping` was in this form, but in the received message it says `PING`).

The probability that any three out of 10 packets have been lost (probability of one packet loss is 0.3) is $\binom{10}{3}(0.7)^7 0.3^3 = 0.26$.

The probability that three particular packets (namely packets 2, 4, 9) out of 10 packets have been lost is $(0.7)^7 0.3^3 = 0.0022$.

It is also noted that the average RTT is around $251\mu s$. This is expected, since there is a small delay in sending messages to local host. The packet loss rate is 0.30.

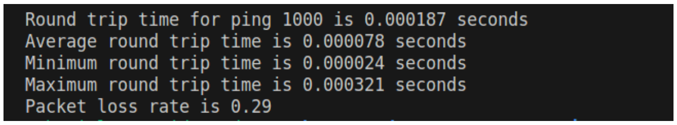


```
 ┌hari@lenovo-ideapad ~/Desktop/acads/comm_nets/experiment_3
● └➤  python UDPPingerClient.py
b'PING 1 1711860356.5612116'
Round trip time for ping 1 is 0.000114 seconds
Request timed out
b'PING 3 1711860357.5624237'
Round trip time for ping 3 is 0.000169 seconds
Request timed out
b'PING 5 1711860358.563932'
Round trip time for ping 5 is 0.000346 seconds
b'PING 6 1711860358.5643237'
Round trip time for ping 6 is 0.000310 seconds
b'PING 7 1711860358.5646951'
Round trip time for ping 7 is 0.000288 seconds
b'PING 8 1711860358.5649948'
Round trip time for ping 8 is 0.000143 seconds
Request timed out
b'PING 10 1711860359.5665674'
Round trip time for ping 10 is 0.000388 seconds
Average round trip time is 0.000251 seconds
Minimum round trip time is 0.000114 seconds
Maximum round trip time is 0.000388 seconds
Packet loss rate is 0.30
```

Fig. 1. Pinging Local Host

Statistics for 1000 packets

When 1000 packets were sent, the following statistics were observed (note that the RTT for packet 1000 is also printed as a demonstration):



```
Round trip time for ping 1000 is 0.000187 seconds
Average round trip time is 0.000078 seconds
Minimum round trip time is 0.000024 seconds
Maximum round trip time is 0.000321 seconds
Packet loss rate is 0.29
```

Fig. 2. Statistics for 1000 packets

Again, a packet rate of approximately 30% is observed.

References

[1] Python Socket Library: https://docs.python.org/3/library/socket.html