

In this assignment, you are to implement ping network utility running over UDP protocol.

Ping application allows a client machine to send a packet of data to a remote machine (server), and have the remote machine (server) echo the data back to the client. In addition, ping application provides useful statistics about the network communication, such as segment round trip time, number of received and lost segments, and loss ratio. Try *ping latech.edu* in command line (console) for a sample run.

You are to submit code for two programs: ping application on client side (e.g. UDPPingClient.py) and ping application on server side (e.g. UDPPingServer.py).

Requirements for server side code:

1. Listens for UDP segments on port 12000.
2. Receives message from client.
3. Sends message back to client with no changes.
4. Simulates random loss of messages (e.g. 4 out of 10 received messages are not sent back to client).

Requirements for client side code:

1. Sets socket timeout as 1 second. *".settimeout()"* socket function is your friend.
2. Sends multiple (*n*) UDP segments with ping message to server with specified IP address and port number. Ping message format is provided below*.
3. Receives message back from server and calculates message round trip time in milliseconds.
4. Displays server response if segment was received. Response format is provided below**.
5. Displays "Request time out" string if segment was lost. Segment is considered lost if the timeout is up.
6. Displays ping statistics information (number of sent, received, lost segments, and loss percentage).
7. Displays approximate round trip times in milliseconds (minimum, maximum, and average RTT).
8. Gets server IP address, server port number and number (*n*) of ping packets to send as a command line arguments.

*Ping message format: **"Ping 1 Fri Jan 10 09:00:00 2020"**.

Server response format: **"Reply from 192.168.1.6: Ping 1 Fri Jan 10 09:00:00 2020 time=2.0ms TTL=1".

Please, note that parts in bold are hardcoded.

Sample command to run client side code: *python UDPPingClient.py 192.168.1.6 12000 5*

Sample outputs of client code:

```
at@at:~$ python UDPPingClient.py 192.168.1.6 12000 5
Pinging 192.168.1.6:
Reply from 192.168.1.6: Ping 1 Fri Jan 10 09:01:01 2020 time=3.0ms TTL=1
Reply from 192.168.1.6: Ping 2 Fri Jan 10 09:01:01 2020 time=0.0ms TTL=1
Reply from 192.168.1.6: Ping 3 Fri Jan 10 09:01:01 2020 time=1.0ms TTL=1
Reply from 192.168.1.6: Ping 4 Fri Jan 10 09:01:01 2020 time=1.0ms TTL=1
Request timed out.

Ping statistics for 192.168.1.6:
    Segments: Sent: 5, Received: 4, Lost: 1 (20.0% Loss)
Approximate round trip times in ms:
    Minimum = 0.0ms, Maximum = 3.0ms, Average = 1.25ms
```

Server code does not require any outputs.