CN ASSIGNMENT - 01

By -
Ketan Mohan Garg (2022248)
Keshav Bindlish (2022246)

UDP PINGER

Description

Created a UDP client that sends 10 ping messages to a server. The client calculates and displays the round-trip time (RTT) for each message, prints "Request timed out" if no reply is received within one second, and reports the minimum, maximum, average RTTs, and packet loss percentage.

Terminal Output

The output in the terminal would include the round-trip times (RTTs) for each of the 10 pings and the Ping statistics (Min, Max and Average of all the RTT's).

UDP PINGER

Output at the client end:

- The UDP Ping client connects with server using socket at given IP address.
- The UDP client sends 10 ping requests to the server, waiting for replies.
- Client calculates round trip time (RTT) for every packet, storing RTT values.
- After 10 pings, the client calculates max, min, and average RTT values.
- In this output, 5 packets were received successfully and 5 were lost.

```
ketangarg@KETAN-PC:/mnt/c/Users/KETAN GARG/Desktop/CN_Assignment-1$ python3 client.py
Sending ping 1..
Request timed out
Sending ping 2..
Reply from server: PING 2 1726760774.541391 RTT = 0.0003 seconds
Sending ping 3..
Reply from server: PING 3 1726760774.5417883 RTT = 0.0002 seconds
Sending ping 4..
Request timed out
Sending ping 5..
Reply from server: PING 5 1726760775.5435295 RTT = 0.0004 seconds
Sending ping 6..
Reply from server: PING 6 1726760775.5440092 RTT = 0.0004 seconds
Sending ping 7..
Request timed out
Sending ping 8..
Request timed out
Sending ping 9..
Request timed out
Sending ping 10..
Reply from server: PING 10 1726760778.5486956 RTT = 0.0006 seconds

--Ping statistics--
Packets sent: 10 Packets received: 5 Packets lost: 5 Percentage of Packet loss: 50%
RTT min/avg/max = 0.0002/0.0004/0.0006 seconds
```

UDP HEARTBEAT

Description

Implement a UDP Heartbeat application where the client sends a sequence number and timestamp to the server. The server responds with the time difference. If the server fails to respond three consecutive times, the client assumes the server is down. Simulate a 30% packet loss rate to test response failures.

Terminal Ouptput

The terminal output on the client side includes about whether the server replied for each ping packet sent, and at the end prints the total number of packets sent before the 3 consecutive misses. The server side just displays the time difference for each packet.

UDP HEARTBEAT

Output when runs on the same machine:

- Client calculates RTT for successful pings, stores RTT values in an array.
- Server calculates the time difference for each successfully received packet and prints on terminal.
- No. of times the packet were sent before the application closed is 5, and total 8 packets were sent.

```
ketangarg@KETAN-PC:/mnt/c/users/KETAN GARG/Desktop/CN_Assignment
-1$ python3 server2.py
Packet received successfylly
time difference: 0.00046896934509277344
Packet received successfylly
time difference: 0.000537872314453125
Packet received successfylly
time difference: 0.00015401840209960938
Packet received successfylly
time difference: 0.0003521442413330078
Packet received successfylly
time difference: 0.00026416778564453125
Packet received successfylly
time difference: 0.00023984909057617188
Packet received successfylly
time difference: 0.0003261566162109375
Packet received successfylly
time difference: 0.0014979839324951172
```

```
ketangarg@KETAN-PC:/mnt/c/Users/KETAN GARG/Desktop/CN_Assignment
-1$ python3 client2.py
Sending ping 1..
Request timed out
Sending ping 2..
RTT = 0.0010 seconds
Sending ping 3..
Request timed out
Sending ping 4..
RTT = 0.0006 seconds
Sending ping 5..
RTT = 0.0005 seconds
Sending ping 6..
Request timed out
Sending ping 7..
Request timed out
Sending ping 8..
Request timed out
Application stopped
No. of times the packets were sent before applications stopped:
 5
```

UDPHeartbeat Application's Server and Client Run on Different Machine

- The client and server now run on separate machines.
- The server's machine address replaces 127.0.0.1 in the client code.
- The client sends ping requests to the server using its machine address.
- The server receives requests and processes them as before.
- The rest of the ping process remains unchanged.

```
keshav@keshav-Vostro-5320:~/Downloads/UDP_Pinger$ python3 client2.py
Sending ping 1..
RTT = 0.0921 seconds
Sending ping 2..
RTT = 0.0072 seconds
Sending ping 3..
RTT = 0.0080 seconds
Sending ping 4..
Request timed out
Sending ping 5..
Request timed out
Sending ping 6..
Request timed out
Application stopped
No. of times the packets were sent before applications stopped:  3
```

```
voldie@voldie:~/Desktop/CN/CN_Assignment_1$ python3 server2.py
Packet recieved successfully
time difference form client to server: 0.16230225563049316
Packet recieved successfully
time difference form client to server: 0.07850527763366699
Packet recieved successfully
time difference form client to server: 0.07877516746520996
Packet recieved successfully
time difference form client to server: 0.07932925224304199
Packet recieved successfully
time difference form client to server: 0.177140474319458
Packet recieved successfully
time difference form client to server: 0.12564706802368164
```

THANK YOU FOR YOUR ATTENTION