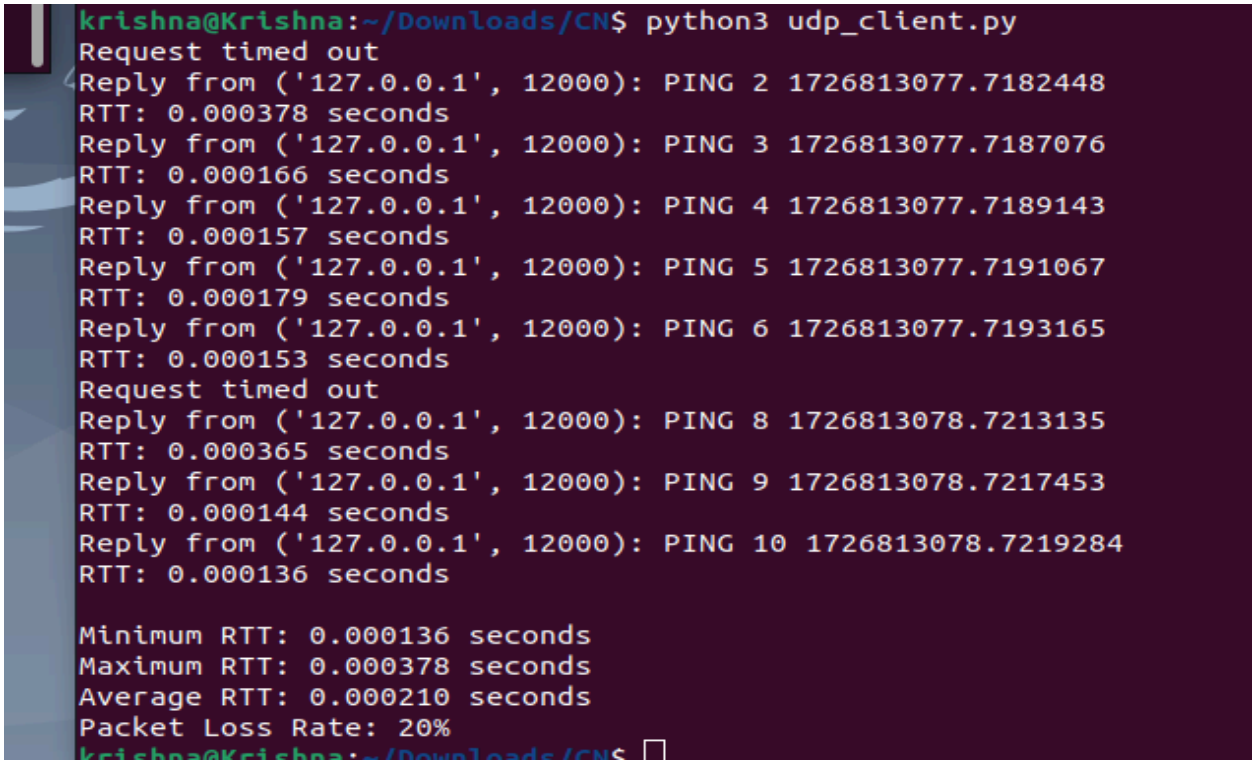
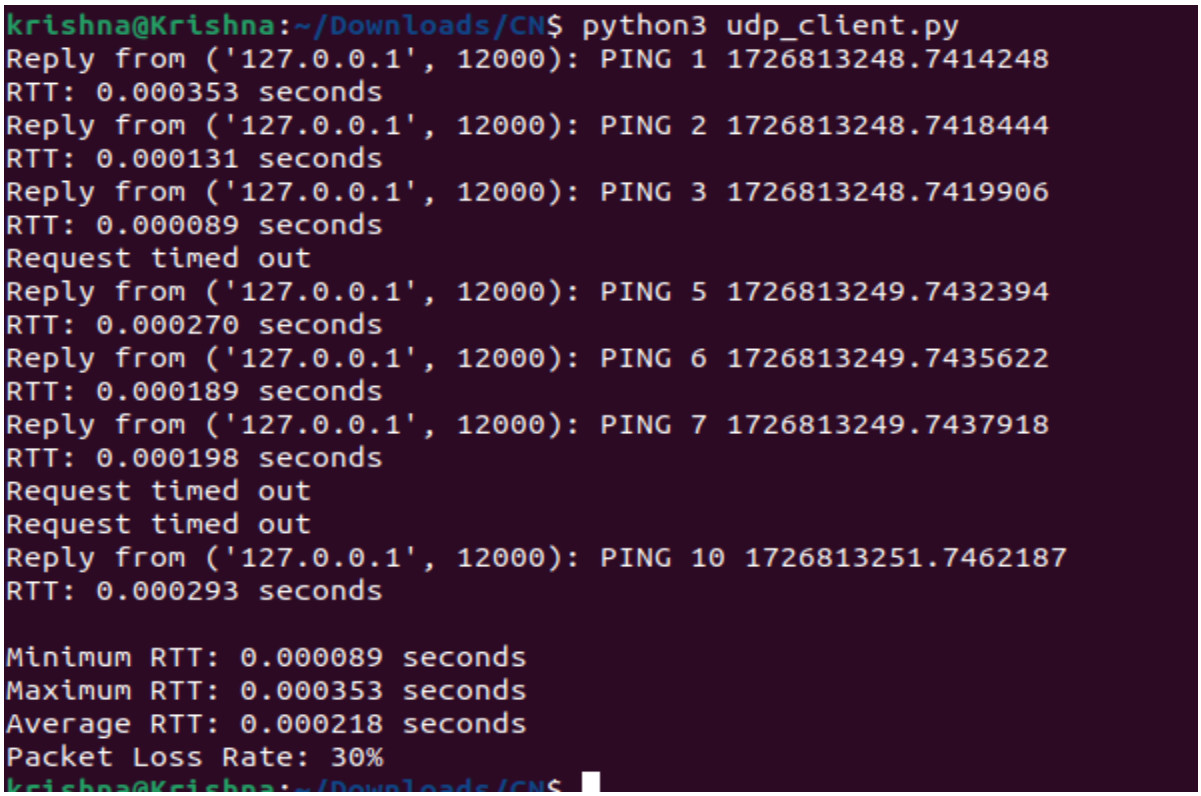


Client side response (1)



Client side response (2)

UDP Cliient code Code Explanation

Thw is a UDP ping client, which sends a series of 10 ping messages to a server running on localhost at port 12000. It uses the socket module for instantiating the UDP socket and the time module, providing management over timestamps.

During every iteration of the loop, client constructs a ping message, combining sequence number and current timestamp, and sends it to the server. It records the time immediately after that and waits for response. It calculates round-trip time if the response is acquired. If the response times out after 1 second, the client increases its loss counter and prints the timeout message.

Then gives statistics: minimum, maximum, and average RTT with a packet loss rate. The program catches exceptions when no responses were received to avoid making arithmetic errors in these statistics calculations. This is an easy way of understanding measurements of the latency of the network with UDP pings.

Question 2>

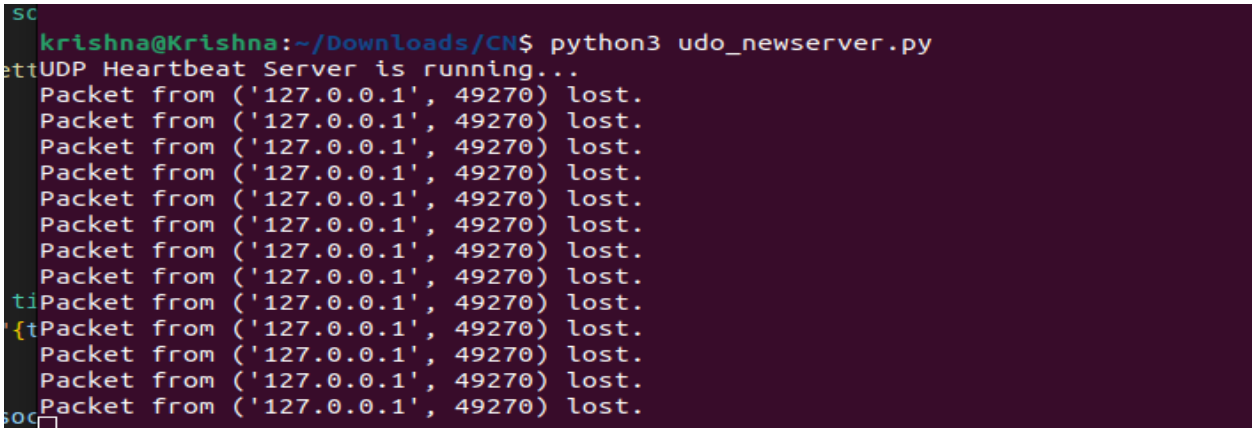
```
krishna@Krishna:~/Downloads/CN$ python3 udo_newserver.py
UDP Heartbeat Server is running...
Packet from ('127.0.0.1', 56143) lost.
Packet from ('127.0.0.1', 56143) lost.
Packet from ('127.0.0.1', 56143) lost.
Packet from ('127.0.0.1', 56143) lost.
Packet from ('127.0.0.1', 56143) lost.
Packet from ('127.0.0.1', 56143) lost.
```

UDP Server Heartbeat response

```
krishna@Krishna:~/Downloads/CN$ python3 udp_newclient.py
Reply from ('127.0.0.1', 12000): Time difference: 0.000273 seconds, Sequence: 1
Request timed out
Reply from ('127.0.0.1', 12000): Time difference: 0.000314 seconds, Sequence: 3
Reply from ('127.0.0.1', 12000): Time difference: 0.000358 seconds, Sequence: 4
Reply from ('127.0.0.1', 12000): Time difference: 0.000348 seconds, Sequence: 5
Reply from ('127.0.0.1', 12000): Time difference: 0.000347 seconds, Sequence: 6
Reply from ('127.0.0.1', 12000): Time difference: 0.000465 seconds, Sequence: 7
Request timed out
Reply from ('127.0.0.1', 12000): Time difference: 0.000346 seconds, Sequence: 9
Reply from ('127.0.0.1', 12000): Time difference: 0.000315 seconds, Sequence: 10
Reply from ('127.0.0.1', 12000): Time difference: 0.000359 seconds, Sequence: 11
Reply from ('127.0.0.1', 12000): Time difference: 0.000360 seconds, Sequence: 12
Request timed out
Reply from ('127.0.0.1', 12000): Time difference: 0.000427 seconds, Sequence: 14
Reply from ('127.0.0.1', 12000): Time difference: 0.000231 seconds, Sequence: 15
Reply from ('127.0.0.1', 12000): Time difference: 0.000398 seconds, Sequence: 16
Reply from ('127.0.0.1', 12000): Time difference: 0.000391 seconds, Sequence: 17
Reply from ('127.0.0.1', 12000): Time difference: 0.000376 seconds, Sequence: 18
Request timed out
Request timed out
Request timed out
Server is down after 21 heartbeats sent.

--- Statistics ---
Total Heartbeats Sent: 21
Total Responses Received: 15
Total Timeouts: 3
Percentage of Failed Responses: 28.57%
Average RTT for Successful Responses: 0.000168 seconds
Minimum RTT for Successful Responses: 0.000096 seconds
Maximum RTT for Successful Responses: 0.000211 seconds
```

UDP Client Heartbeat response



UDP Client Side response (2)

```
krishna@Krishna:~/Downloads/CN$ python3 udp_newclient.py
Reply from ('127.0.0.1', 12000): Time difference: 0.000298 seconds, Sequence: 1
Request timed out
Reply from ('127.0.0.1', 12000): Time difference: 0.000298 seconds, Sequence: 3
Reply from ('127.0.0.1', 12000): Time difference: 0.000344 seconds, Sequence: 4
Reply from ('127.0.0.1', 12000): Time difference: 0.000378 seconds, Sequence: 5
Request timed out
Reply from ('127.0.0.1', 12000): Time difference: 0.000350 seconds, Sequence: 7
Reply from ('127.0.0.1', 12000): Time difference: 0.000303 seconds, Sequence: 8
Reply from ('127.0.0.1', 12000): Time difference: 0.000244 seconds, Sequence: 9
Request timed out
Reply from ('127.0.0.1', 12000): Time difference: 0.000298 seconds, Sequence: 11
Request timed out
Request timed out
Reply from ('127.0.0.1', 12000): Time difference: 0.000372 seconds, Sequence: 14
Reply from ('127.0.0.1', 12000): Time difference: 0.000263 seconds, Sequence: 15
Request timed out
Request timed out
Reply from ('127.0.0.1', 12000): Time difference: 0.000418 seconds, Sequence: 18
Reply from ('127.0.0.1', 12000): Time difference: 0.000268 seconds, Sequence: 19
Reply from ('127.0.0.1', 12000): Time difference: 0.000327 seconds, Sequence: 20
Reply from ('127.0.0.1', 12000): Time difference: 0.000273 seconds, Sequence: 21
Request timed out
Reply from ('127.0.0.1', 12000): Time difference: 0.000311 seconds, Sequence: 23
Reply from ('127.0.0.1', 12000): Time difference: 0.000310 seconds, Sequence: 24
Reply from ('127.0.0.1', 12000): Time difference: 0.000388 seconds, Sequence: 25
Reply from ('127.0.0.1', 12000): Time difference: 0.000300 seconds, Sequence: 26
Reply from ('127.0.0.1', 12000): Time difference: 0.000255 seconds, Sequence: 27
Request timed out
Reply from ('127.0.0.1', 12000): Time difference: 0.000494 seconds, Sequence: 29
Reply from ('127.0.0.1', 12000): Time difference: 0.000270 seconds, Sequence: 30
Request timed out
Reply from ('127.0.0.1', 12000): Time difference: 0.000294 seconds, Sequence: 32
Reply from ('127.0.0.1', 12000): Time difference: 0.000175 seconds, Sequence: 33
Reply from ('127.0.0.1', 12000): Time difference: 0.000308 seconds, Sequence: 34
Request timed out
Request timed out
Request timed out
Server is down after 37 heartbeats sent.

--- Statistics ---
Total Heartbeats Sent: 37
Total Responses Received: 24
Total Timeouts: 3
Percentage of Failed Responses: 35.14%
Average RTT for Successful Responses: 0.000176 seconds
Minimum RTT for Successful Responses: 0.000088 seconds
Maximum RTT for Successful Responses: 0.000294 seconds
krishna@Krishna:~/Downloads/CN$
```

UDP Client side response(2)

UDP Client Code explanation

The UDP Heartbeat client creates a UDP socket designed to track the presence of a server through periodic heartbeat messages. It declares a UDP socket that communicates with a server running on localhost on port 12000, with a timeout of 1 second for responses. The loop is in an infinite loop: it sends heartbeat messages containing the timestamp and a sequence number. It does find the round-trip time for each of the responses received by storing all good responses in a list. If no response is received within the time-out, it increments the counter for missed responses and prints a message about the time out. In case of three consecutive time outs, it concludes that the server is down and leaves the loop. At the end of the test, it also calculates and prints the following statistics: total heartbeats sent, responses received, timeouts, packet loss percent, and RTT in averages, min, and max metrics. Thus, the results tell what's going on with the server health and how things go in the network.

UDP Server Code Explanation

The code will set up a UDP Heartbeat server to listen on port 12000 for incoming heartbeat messages. On receiving of the message it simulates some packet loss at random from 30%. It has some random number so if that is less than 3 the server assumes the packet lost and doesn't send an acknowledgement. Assuming the packet is received, the server processes the heartbeat message with the timestamp along with the sequence number. It also calculates how much time has passed from the time it arrived to the current time by computing a difference between the time elapsed and the received timestamp, and it will return the information to the client along with the sequence number. The server keeps running with multiple heartbeat messages.

How many times should you send the UDP Heartbeat packet before you see 3 consecutive responses missing?

Calculation

1. **Packet Loss Rate**: With a 30% packet loss rate, there is a 70% chance that any single packet will be received.
2. **Consecutive Losses**: To see 3 consecutive missing responses, you need to experience three failures in a row.
3. **Probability of 3 Consecutive Losses**:
    - The probability of missing a single heartbeat is 0.3 (30%).
    - The probability of missing three consecutive heartbeats is (0.3)^3=0.027 (2.7%).

Expected Trials

To find out how many heartbeats you need to send to expect 3 consecutive losses:

1. **Average Trials Until 3 Consecutive Losses**:
    - As this is a Geometic Random Variable, The average number of trials needed can be calculated as 1/P, where P is the probability of the event happening. In this case, it would be:
2. Expected trials=1/0.027 =37 trial

This means you might expect to send around 37 heartbeats before observing 3 consecutive lost packets.

Conclusion

- **Sending 10 Heartbeat Packets**: Unlikely to see 3 consecutive losses (probability too low) appx(25%)
- **Sending 100 Heartbeat Packets**: More likely to see 3 consecutive losses appx (75%)
- **Sending 1000 Heartbeat Packets**: Almost certain to see 3 consecutive losses. appx(99%)

Based on the calculations, sending **around 37 to 100 packets** could lead you to experience 3 consecutive missing responses, depending on the random occurrences of packet loss.