

# CN LAB REPORT

**NAME :** G.V.Prashanth Reddy

**Roll No :** SE20UARI052

## UDP Server

1. Class with PingServer.
2. A UDP server needs to be aware of the client port on which to send messages to the client so a private static final field is used.
3. A socket is needed to be created for receiving and sending UDP packets through the port. To generate the messages that need to be sent to clients are stringed by the loop.
4. Since we use UDP, datagram packets are used for communication so all these messages need to be wrapped in a datagram packet
5. Datagram packet needs to know the message that will be sent (the message will be in byte type) the length of the message, the IP address of the client (hostname), and the port where the client can receive it.

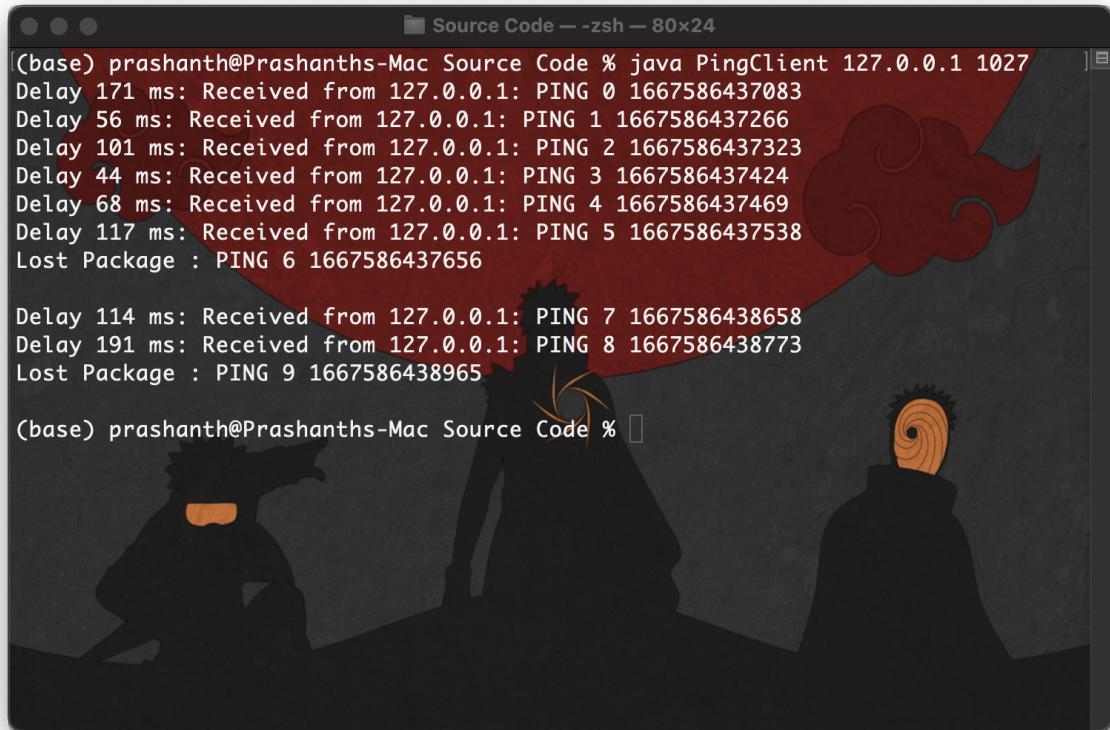
## UDP Client

1. Create a class with PingClient.
2. Define the port on which the client will receive the message.
3. Create a socket and give a host IP address and a port number which is the same as the server port number, then set up a connection socket so that we can send and receive UDP packets.
4. A loop is set up to send out 10 UDP packets the packets will contain the host address, the number of the packets and the time in milliseconds to receive the packet.
5. If the pack is not received in one second or a thousand milliseconds then the packet is assumed lost.
6. The bytes are then wrapped into an array stream so they can be read.

## Server : PingServer.java contains the source code

```
(base) prashanth@Prashanths-Mac Source Code % java PingServer 1027
Received from 127.0.0.1: PING 0 1667586437083
Reply sent.
Received from 127.0.0.1: PING 1 1667586437266
Reply sent.
Received from 127.0.0.1: PING 2 1667586437323
Reply sent.
Received from 127.0.0.1: PING 3 1667586437424
Reply sent.
Received from 127.0.0.1: PING 4 1667586437469
Reply sent.
Received from 127.0.0.1: PING 5 1667586437538
Reply sent.
Received from 127.0.0.1: PING 6 1667586437656
Reply not sent.
Received from 127.0.0.1: PING 7 1667586438658
Reply sent.
Received from 127.0.0.1: PING 8 1667586438773
Reply sent.
Received from 127.0.0.1: PING 9 1667586438965
Reply not sent.
Received from 127.0.0.1: PING 0 1667586496894
Reply sent.
Received from 127.0.0.1: PING 1 1667586497096
Reply sent.
Received from 127.0.0.1: PING 2 1667586497248
Reply sent.
Received from 127.0.0.1: PING 3 1667586497251
Reply sent.
Received from 127.0.0.1: PING 4 1667586497382
Reply sent.
Received from 127.0.0.1: PING 5 1667586497520
Reply sent.
Received from 127.0.0.1: PING 6 1667586497642
Reply sent.
Received from 127.0.0.1: PING 7 1667586497679
Reply sent.
Received from 127.0.0.1: PING 8 1667586497698
Reply not sent.
Received from 127.0.0.1: PING 9 1667586498700
Reply sent.
Received from 127.0.0.1: PING 0 1667586544679
Reply not sent.
Received from 127.0.0.1: PING 1 1667586545693
Reply sent.
Received from 127.0.0.1: PING 2 1667586546694
Reply not sent.
Received from 127.0.0.1: PING 3 1667586547695
Reply sent.
Received from 127.0.0.1: PING 4 1667586548699
Reply sent.
Received from 127.0.0.1: PING 5 1667586549704
Reply not sent.
Received from 127.0.0.1: PING 6 1667586550705
Reply sent.
Received from 127.0.0.1: PING 7 1667586551706
Reply not sent.
Received from 127.0.0.1: PING 8 1667586552707
```

Client : PingClient.java contains the source code.



```
Source Code -- zsh -- 80x24
(base) prashanth@Prashanths-Mac Source Code % java PingClient 127.0.0.1 1027
Delay 171 ms: Received from 127.0.0.1: PING 0 1667586437083
Delay 56 ms: Received from 127.0.0.1: PING 1 1667586437266
Delay 101 ms: Received from 127.0.0.1: PING 2 1667586437323
Delay 44 ms: Received from 127.0.0.1: PING 3 1667586437424
Delay 68 ms: Received from 127.0.0.1: PING 4 1667586437469
Delay 117 ms: Received from 127.0.0.1: PING 5 1667586437538
Lost Package : PING 6 1667586437656

Delay 114 ms: Received from 127.0.0.1: PING 7 1667586438658
Delay 191 ms: Received from 127.0.0.1: PING 8 1667586438773
Lost Package : PING 9 1667586438965

(base) prashanth@Prashanths-Mac Source Code %
```

- 1) Currently the program calculates the round-trip time for each packet and prints them out individually. Modify this to correspond to the way the standard ping program works. You will need to report the minimum, maximum, and average RTTs. (easy)

```
(base) prashanth@Prashanths-Mac Source Code % java clientQ1 127.0.0.1 1027
Delay 191 ms: Received from 127.0.0.1: PING 0 1667586496894
Delay 152 ms: Received from 127.0.0.1: PING 1 1667586497096
Delay 2 ms: Received from 127.0.0.1: PING 2 1667586497248
Delay 130 ms: Received from 127.0.0.1: PING 3 1667586497251
Delay 138 ms: Received from 127.0.0.1: PING 4 1667586497382
Delay 121 ms: Received from 127.0.0.1: PING 5 1667586497520
Delay 36 ms: Received from 127.0.0.1: PING 6 1667586497642
Delay 19 ms: Received from 127.0.0.1: PING 7 1667586497679
Pacote perdido: PING 8 1667586497698
Delay 146 ms: Received from 127.0.0.1: PING 9 1667586498700
RTT: minDelay: 2ms / maxDelay: 1000ms / averageDelay: 193ms
(base) prashanth@Prashanths-Mac Source Code %
```

clientQ1.java contains the source code

2) The basic program sends a new ping immediately when it receives a reply. Modify the program so that it sends exactly 1 ping per second, similar to how the standard ping program works. Hint: Use the Timer and TimerTask classes in java.util. (difficult)

```
(base) prashanth@Prashanths-Mac Source Code % java clientQ2 127.0.0.1 1027
Pacote perdido: PING 0 1667586544679
Delay 82 ms: Received from 127.0.0.1: PING 1 1667586545693
Pacote perdido: PING 2 1667586546694
Delay 197 ms: Received from 127.0.0.1: PING 3 1667586547695
Delay 179 ms: Received from 127.0.0.1: PING 4 1667586548699
Pacote perdido: PING 5 1667586549704
Delay 133 ms: Received from 127.0.0.1: PING 6 1667586550705
Pacote perdido: PING 7 1667586551706
Pacote perdido: PING 8 1667586552707
Delay 101 ms: Received from 127.0.0.1: PING 9 1667586553707
(base) prashanth@Prashanths-Mac Source Code %
```

`clientQ2.java` contains the source code