Computer Networks Lab 6 Assignment

Name: Muvvala N V Sai Gurudeep

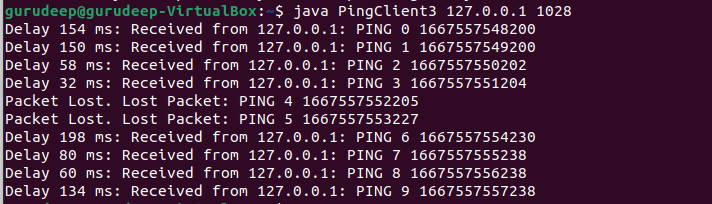
Roll No: SE20UARI103

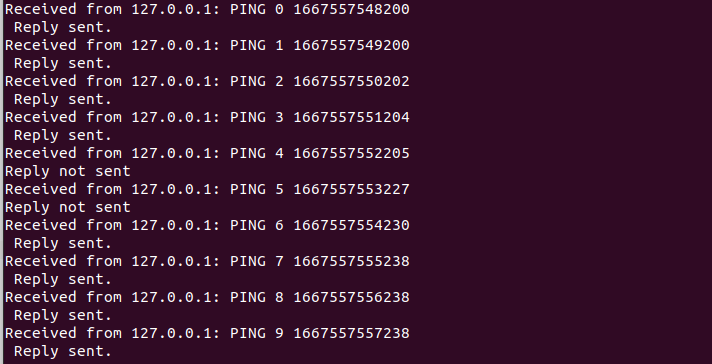
Group: A2

Branch: AI

1. Write the client so that it sends 10 ping requests to the server, separated by approximately one second. Each message contains a payload of data that includes the keyword PING, a sequence number, and a timestamp. After sending each packet, the client waits up to one second to receive a reply. If one second goes by without a reply from the server, then the client assumes that its packet or the server's reply packet has been lost in the network.

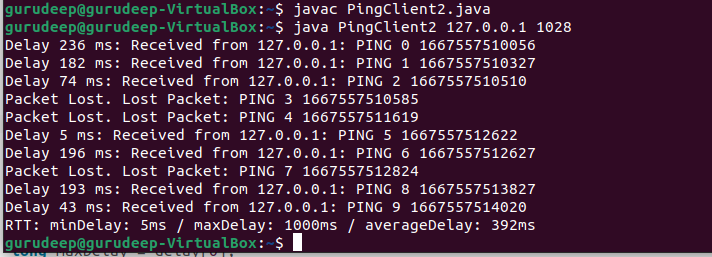
* The PingClient class was used to implement the client. Here we have TIMEOUT(useful if the connection wasn't made on time), MAX\_REQUESTS(tells the number of ping requests the client should be making to the server) and CLIENT\_PORT(having the default client port number).
* Initially we check if host and port are given in the command line as inputs to the program. If not we return back telling the user to provide host and port number.
* After use the server name as 127.0.0.0 and parse the port number of the server and its Ip Address.
* We use a while loop to create a running for maximum response datagram received.
  + Next we create a response datagram socket for receiving and sending the UDP packets through the port.
  + Next we sent the desired timeout to the client socket.
  + We start a timer and make up a message and convert it into bytes which will be sent to the server.
  + Next we create a datagram packet with message, message length, IP address of server and server port number. We send this datagram packet to the server.
  + We get the response from the server from the client and calculate into the time we get the response from the server.
  + On Subtraction of 2 times captured we get the delayed time and print the response.
  + We raise an exception if the packet is lost and display the lost packet.
* We have another function called printData where we get the response and convert it into bytes and later into stream of characters which is displayed later.

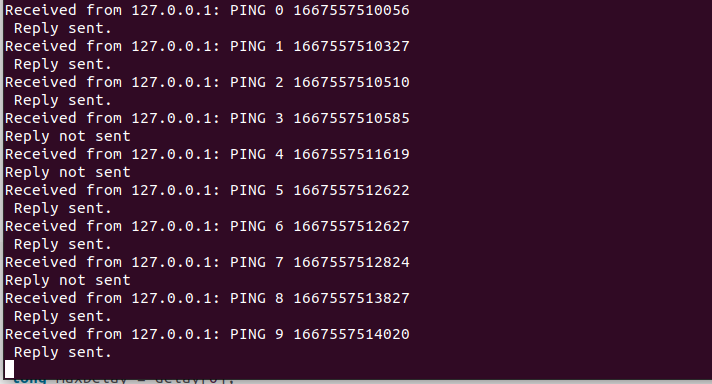




1. Currently the program calculates the round-trip time for each packet and prints them out individually. Modify this to correspond to the way the standard ping program works. You will need to report the minimum, maximum, and average RTTs. (easy)

* The PingClient class was used to implement the client. Here we have TIMEOUT(useful if the connection wasn't made on time), MAX\_REQUESTS(tells the number of ping requests the client should be making to the server) and CLIENT\_PORT(having the default client port number).
* Initially we check if host and port are given in the command line as inputs to the program. If not we return back telling the user to provide host and port number.
* After use the server name as 127.0.0.0 and parse the port number of the server and its Ip Address.
* We will store the delays calculated in each loop.
* We use a while loop to create a running for maximum response datagram received.
  + Next we create a response datagram socket for receiving and sending the UDP packets through the port.
  + Next we sent the desired timeout to the client socket.
  + We start a timer and make up a message and convert it into bytes which will be sent to the server.
  + Next we create a datagram packet with message, message length, IP address of server and server port number. We send this datagram packet to the server.
  + We get the response from the server from the client and calculate into the time we get the response from the server.
  + On Subtraction of 2 times captured we get the delayed time and print the response. We store the delay time in an array each time.
  + We raise an exception if the packet is lost and display the lost packet.
* We calculate minimum delay, maximum delay and average delay which are in milliseconds and print them out.
* We have another function called printData where we get the response and convert it into bytes and later into stream of characters which is displayed later.





1. 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)

* The PingClient2 class was used to implement the client. Here we have TIMEOUT(useful if the connection wasn't made on time), MAX\_REQUESTS(tells the number of ping requests the client should be making to the server) and CLIENT\_PORT(having the default client port number).
* Initially we check if host and port are given in the command line as inputs to the program. If not we return back telling the user to provide host and port number.
* Here as requests need to be performed after one second we will call a class ExecuteTask. Time.schedule tells that execute class should be called every 1 second.
* This ExecuteTask class has a user defined constructor for initializing the values and a run() function which will be executed desired number of times here its 10 to call a function called ping in public class PingClient2.
* After use the server name as 127.0.0.0 and parse the port number of the server and its Ip Address.
* We use a while loop to create a running for maximum response datagram received.
  + Next we create a response datagram socket for receiving and sending the UDP packets through the port.
  + Next we sent the desired timeout to the client socket.
  + We start a timer and make up a message and convert it into bytes which will be sent to the server.
  + Next we create a datagram packet with message, message length, IP address of server and server port number. We send this datagram packet to the server.
  + We get the response from the server from the client and calculate into the time we get the response from the server.
  + On Subtraction of 2 times captured we get the delayed time and print the response.
  + We raise an exception if the packet is lost and display the lost packet..
* We have another function called printData where we get the response and convert it into bytes and later into stream of characters which is displayed later.

