

## CS 453: Programming assignment 2

### 100 points

### Due by Sunday April 1 11:59 PM

In this lab, you will study a simple Internet ping server written in the Java language, and implement a corresponding client. The functionality provided by these programs is similar to the standard ping programs available in modern operating systems, except that they use UDP rather than Internet Control Message Protocol (ICMP) to communicate with each other.

The ping protocol allows a client machine to send a packet of data to a remote machine, and have the remote machine return the data back to the client unchanged (an action referred to as echoing). Among other uses, the ping protocol allows hosts to determine round-trip times to other machines.

You are given the complete code for the Ping server below. Your job is to write the Ping client.

### Server Code

The Ping Server code has been uploaded on to e-campus. You need to compile and run this code. You should study this code carefully, as it will help you write your Ping client.

The server sits in an infinite loop listening for incoming UDP packets. When a packet comes in, the server simply sends the encapsulated data back to the client.

### Packet Loss

UDP provides applications with an unreliable transport service, because messages may get lost in the network due to router queue overflows or other reasons. In contrast, TCP provides applications with a reliable transport service and takes care of any lost packets by retransmitting them until they are successfully received. Applications using UDP for communication must therefore implement any reliability they need separately in the application level (each application can implement a different policy, according to its specific needs).

Because packet loss is rare or even non-existent in typical campus networks, the server in this lab injects artificial loss to simulate the effects of network packet loss. The server has a parameter `LOSS_RATE` that determines which percentage of packets should be lost. The server also has another parameter `AVERAGE_DELAY` that is used to simulate transmission delay from sending a packet across the Internet. You should set `AVERAGE_DELAY` to a positive value when testing your client and server on the same

machine, or when machines are close by on the network. You can set AVERAGE\_DELAY to 0 to find out the true round trip times of your packets.

## Compiling and Running Server

To compile the server, do the following:  
`javac PingServer.java`

To run the server, do the following:  
`java PingServer port`

where port is the port number the server listens on. Remember that you have to pick a port number greater than 1024, because only processes running with root (administrator) privilege can bind to ports less than 1024.

Note: if you get a class not found error when running the above command, then you may need to tell Java to look in the current directory in order to resolve class references. In this case, the commands will be as follows:  
`java -classpath . PingServer port`

## Your Job: The Client

You should 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.

Template PingClient.java has been provided. Fill in the places with “?” to write this code.

You should write the client so that it can be executed with the following command:  
`java PingClient host port`

where host is the name of the computer the server is running on and port is the port number it is listening to. Note that you can run the client and server either on different machines or on the same machine.

The client should send 10 pings to the server. Because UDP is an unreliable protocol, some of the packets sent to the server may be lost, or some of the packets sent from server to client may be lost. For this reason, the client cannot wait indefinitely for a reply to a ping message. You should have the client wait up to one second for a reply; if no reply is received, then the client should assume that the packet was lost during transmission across the network. You will need to research the API for DatagramSocket to find out how to set the timeout value on a datagram socket.

When developing your code, you should run the ping server on your machine, and test your client by sending packets to localhost (or, 127.0.0.1).

## **Message Format**

The ping messages in this lab are formatted in a simple way. Each message contains a sequence of characters terminated by a carriage return character (r) and a line feed character (n). The message contains the following string:

PING sequence\_number time CRLF

where sequence\_number starts at 0 and progresses to 9 for each successive ping message sent by the client, time is the time when the client sent the message, and CRLF represent the carriage return and line feed characters that terminate the line.

After displaying the message for each of the 10 ping messages, display the minimum, maximum and average round trip times.

## **Submission**

Submit PingClient.java along with a text file containing output from PingClient, showing the minimum, maximum and average round trip times.