CST 428/528

Instructor: Anand Seetharam

Programming Assignment 1: UDP Pinger

In this programming assignment, you will write a client ping program in Python. Your client will send a simple ping message to a server, receive a corresponding pong message back from the server, and determine the delay between when the client sent the ping message and received the pong message. This delay is called the Round Trip Time (RTT). The functionality provided by the client and server is similar to the functionality provided by standard ping program available in modern operating systems. However, standard ping programs use the Internet Control Message Protocol (ICMP) (which we will study in Chapter 4). Here we will create a nonstandard (but simple!) UDP-based ping program.

Your ping program is to send 10 ping messages to the target server over UDP. For each message, your client is to determine and print the RTT when the corresponding pong message is returned. Because UDP is an unreliable protocol, a packet sent by the client or server 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 from the server; if no reply is received, the client should assume that the packet was lost and print a message accordingly.

In this assignment, you will be given the complete code for the server (available in myCourses). Your job is to write the client code, which will be very similar to the server code. It is recommended that you first study carefully the server code. You can then write your client code, liberally cutting and pasting lines from the server code.

If you choose to write code in any other language other than Python, use have to first understand the python server code and make it work in the language of your choice.

What programming language to choose?

You can code in any language of your choice – C, C++, Java and Python. If you are using any other programming language please talk to me first. Here are a few tips for coding in any of these languages.

Programming in Java

Java encapsulates the concept of a client-side connection-oriented (TCP) socket with the class, Socket. Java encapsulates the concept of a server-side connection-oriented socket with the class, ServerSocket. You'll need to use the accept() and close(0) methods of this class.

If you're interested in a quick Java tutorial targeted specifically at socket programming, check out "Socket Programming in Java: a tutorial," by Q. Mahmoud, Javaworld, Dec. 1996, http://www.javaworld.com/article/2077322/core-java/sockets-programming-in-java-a-tutorial.html .

Programming in C

If you choose to code in C, you will need to learn the use of system calls such as socket(), bind(), listen(), accept(), close(). Here is a url for socket programming http://www.lowtek.com/sockets/

Programming in Python

The socket programming code in your book is written in python. An online Python socket tutorial is https://docs.python.org/2/howto/sockets.html

What to hand in?

1. Submit code with in-line documentation

2. Run your code on your local machine as well as on 'remote'. A design document outlining the design decisions you made and also sample outputs (screenshots), which show that you program works correctly. Please include screenshots that show that your code runs correctly on your machine as well as on remote. If you had to make any changes to run your code on remote, please mention that in your design document.

3. A brief description of cases where you code might fail and possible ways of improving your program.

Grading Criteria

In-line documentation - 20 points

Code compiles and executes correctly – 60 points

Design document – 20 points