

CMPSC 381
Data Communications and Networks
Spring 2016
Bob Roos

<http://cs.allegheeny.edu/sites/rroos/cs381s2016>

Lab 3
11 February 2016
Due via Bitbucket on Thursday, 18 February, 8 a.m.
NOTE THE 8 a.m. DEADLINE!

Summary: Build a ping-like client. (This is Assignment 2 on page 179 of our textbook.)

Details:

1. **[Learn about ping.]**

Ping is a computer network administration software utility used to test the reachability of a host on an Internet Protocol (IP) network and to measure the round-trip time for messages sent from the originating host to a destination computer and back. [Wikipedia entry for “Ping (networking utility)”]

See the full entry at [https://en.wikipedia.org/wiki/Ping_\(networking_utility\)](https://en.wikipedia.org/wiki/Ping_(networking_utility)). Also read the manual page by typing “man ping” in a terminal window.

For security reasons, the ping command is often disabled to prevent denial-of-service attacks from outside. However, it works on our internal network. Try pinging one of the machines in the lab (aldenv25, ..., aldenv43), e.g.,

```
ping -c 5 aldenv25
```

The “-c 5” tells it to send 5 packets; typing “ping aldenv25” causes packets to be sent until you hit CTRL-C.

2. **[Learn more about Python socket programming.]** There are many, many Python socket tutorials online. Here is one specifically about UDP:

<http://www.binarytides.com/programming-udp-sockets-in-python/>

If you find another tutorial you like, please post the link in the Slack group so everyone can see it.

A quick note about importing: in Python, you can import in two ways. If you just import the name of the package:

```
import socket
mysocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
... etc. ...
```

then every element in that package (methods, constants, etc.) must be preceded by the package name (thus, “`socket.socket(...)`”, “`socket.SOCK_DGRAM`”, etc.)

Or you can use:

```
from socket import *
mysocket = socket(AF_INET,SOCK_DGRAM)
... etc. ...
```

Here, all the method names, constants, etc. have been imported and you don’t need the “`socket`” qualifier. In large projects the first method is preferred in order to avoid naming conflicts and provide better documentation.

3. [**Download the server file.**] On the course website, on the “Lab Handouts” page, find the file “`pingserver.py`” and save it in the `lab3` directory of your bitbucket repository. This is set to imitate a `ping` server and to simulate packet loss by randomly ignoring a certain percentage of all packets received. (If we didn’t simulate it, we probably would not see any packet loss in our little lab network.) The server sits in an infinite loop listening for incoming UDP packets. When a packet comes in and if a randomized integer is greater than or equal to 4, the server simply capitalizes the encapsulated data and sends it back to the client.
4. [**Create a ping client.**] Create a Python client program named `yourlastname-lab3.py`. [From the Assignment 2 handout in the textbook:] “The client should send 10 pings to the server. Because UDP is an unreliable protocol, a packet sent from the client to the server may be lost in the network, or vice versa. For this reason, the client cannot wait indefinitely for a reply to a ping message. You should make the client wait up to one second for a reply; if no reply is received within one second, your client program should assume that the packet was lost during transmission across the network. You will need to look up the Python documentation to find out how to set the timeout value on a datagram socket. Specifically, your client program should
 - (a) send the ping message using UDP (Note: Unlike TCP, you do not need to establish a connection first, since UDP is a connectionless protocol.)
 - (b) print the response message from server, if any
 - (c) calculate and print the round trip time (RTT), in seconds, of each packet, if server responds
 - (d) otherwise, print “Request timed out”

During development, you should run `pingserver.py` on your machine, and test your client by sending packets to port 12345 of your local machine. After you have fully debugged your code, you should see how your application communicates across the network with the ping server and ping client running on different machines.”

5. [**One more thing.**] After your program has sent the last message, have it print out the number of successful pings, the number of unsuccessful pings, and the average round-trip time of the successful pings.

The ping message should contain some text and the number of the ping (between 1 and 10). See sample output below. I have provided some skeleton code for the client on the website.

The `pingserver.py` file illustrates the use of the `sendto` and `recvfrom` methods.

You should run the ping server on your local machine, but you may test your client by `ssh`-ing into a different lab machine and then pinging your local machine.

In addition to uploading your ping client, save a screenshot of a successful run of your client, clearly showing all the messages and responses. (The SHIFT-PrtScn key lets you select an area of the screen to capture.) Make sure your screenshot is legible! (You might consider editing your terminal profile preferences to make the font larger and/or change the colors to black on white before taking your screenshot.)

You will find the following Python packages and functions useful: the `time` package; the `settimeout` method of the `socket` package.

SAMPLE OUTPUT:

```
$ python rooslab3.py
Sending: Ping 1
Reply from 127.0.0.1: PING 1
RTT: 0.000362873077393
Sending: Ping 2
Request timed out.
... etc. ...
Sending: Ping 10
Reply from 127.0.0.1: PING 10
RTT: 0.000219106674194

6 successes, 4 failures
average RTT = 0.000182191530863
```

[**Submit your work.**] Make sure you have a webserver named “*yourlastname-lab3.py*” in your `lab3` folder, as well as a screenshot of a sample run of your client. Upload this folder to your Bitbucket repository by the lab deadline.

Make sure your name, and the honor code pledge appear in your header comments. Make sure you have commented all of your added code and deleted obsolete comments.

[**CLASSIC GEEK HUMOR.**] There is a famous children’s book by Marjorie Flack called *The Story about Ping*. First, read the Wikipedia entry for it to see what the book is about:

https://en.wikipedia.org/wiki/The_Story_about_Ping

Then read the first Customer Review of the book on `amazon.com`:

<http://www.amazon.com/Story-about-Ping-Marjorie-Flack/dp/0448421658>

Be sure to click the “Read more” button if the full review doesn’t display.