CNT4704: Analysis of Computer Communication Networks (Fall 2014)

Programming Assignment 3: Port Scanning Tool

(Assigned Nov. 12th; due: Nov. 23rd midnight)

In this programming assignment, you will reinforce your socket programming knowledge and skill to generate a small but useful port scanning tool, which can scan a remote machine for a range of TCP ports in order to find which TCP services have been provided by the remote machine (just like the popular port scanning software Nmap).

You can use either C, or C++, or Python, or Java, to complete this program. The program should be able to run in Eustis2 machine.

Functionality Description:

Suppose your program executable code is called 'netProb'. It is required to have the following command line:

```
czou@eustis2:~$./netProb hostname m-n
```

where 'hostname' is the remote machine's domain name (or IP address). 'm' and 'n' are the port scanning starting port number and ending port number, respectively. For example, if the command line is:

```
czou@eustis2:~$ ./netProb monroe.cs.ucf.edu 1-500
```

Then the program should scan TCP port incrementally one-by-one from port 1 to port 500.

When port x is not open on the remote machine, the program netProb should show nothing. If the port x is open and accepting client connection request, your program netProb should try to obtain the first response packet message from the server and then print it out. For example, when netProb the longwood machine in the above example, when scanning its port 25, the server accepts the connection (because it is an Email server), and then sends back a short message. Your netProb program should print out the following:

```
Port 25 is open. Response from server is:
220 monroe.cs.ucf.edu ESMTP Sendmail 8.13.8+ Sun/8.13.3; Tue, 11 Nov 2014 23:52:52 -0500 (EST)
```

However, some service (such as port 111) does not send back any response upon a client connection; in this case your netProb should print out:

```
Port 111 is open. But the server does not send back any response.
```

What to turn in

Turn in two files: (1). A project report. (2). The source code.

To show me that you did successfully accomplished the assignment, in your project report:

(1) describe in your report your overall program design with explanations for the design choices you might have made;

(2) Run your program in Eustis or Eustis2 machine. Provide the screenshot image showing the running results by using the following command line:

```
netProb longwood.cs.ucf.edu 1-1000
```

Helpful Hints

Setting timeout value for receiving packets: I will use C socket programming as example. If a socket is connected and you use socket function recv() to obtain message sent from the server, it works if the server sends back a message, but it

will have a VERY LONG timeout if the server does not send back anything (for example, it is expecting the client sending request first). This will slow down your port scanning speed.

One way to solve this is to explicitly set the timeout value for recv(). In C programming, you can add the following code after the socket 'sockfd' is created:

```
struct timeval tv;
tv.tv_sec = 2;
tv.tv_usec = 0;
setsockopt(sockfd, SOL_SOCKET, SO_RCVTIMEO, (char *)&tv, sizeof tv);
```

This code will set the recv() to timeout after 2 seconds. When timeout happens, recv() returns -1.