Homework 8: The Network

cs341
Spring 2002

Allen B. Downey
Computer Science Department

Due: Wednesday 8 May

The purpose of this assignment is to try out some network tools and implement a part of a web browser

Explore the network

1. Read the man page for the `arp` command and then run `/sbin/arp`. You should see a list of machines and their hardware addresses. These are the machines on the network that are currently in your ARP cache.

2. Read the man page for `ping` and then ping one of the machine in the Linux lab that is *not* in your ARP cache.

3. Run `arp` again. You should see that the machine you pinged has been added to the ARP cache. What is the hardware address for `cheers`?

4. Read the man page for `host` and then run `host -v rocky`. See what you can make of the results. You can also use `host` to look up an IP address and get the hostname. I use that feature occasionally when I see something funny in my Web log. For example, this entry

   ```
   64.107.200.2 - - [24/Apr/2002:20:01:50 -0400] "GET
   /downey/firstclass.html HTTP/1.0" 304 - "-"
   "Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"
   ```

   shows that someone running Windows 98 downloaded my rant about FirstClass at 8pm on April 24. Use `host` to see what else we can find out about this person.

5. When you hit Control-C to stop `ping` it prints summary statistics for the packets it sent. Notice that it provides the min/avg/max times and a measure of variation (I think it's the maximum deviation from the mean). What an excellent way to summarize a list of numbers! What is the average round trip time to another machine in the cluster?

6. A ping packet is usually 64 bytes, but you can change that with the `-s` option. Ping the same machine again with packet size 1492. What is the round trip time now?

7. For a range of packet sizes, plot the round trip time versus the packet size. Fit a line to this curve and compute its slope and intercept. The slope of the line is the marginal additional time per additional byte, which is the inverse of bandwidth. Compute the inverse of the slope and compare it to the bandwidth you expected to get from the network.

8. Ping a machine in Massachusetts, a machine in California and a machine in Australia. Compare the round trip times. How strong is the relationship between round trip time and geographical distance? Try to find two machines where network "distances" don't reflect geographical distances.

9. Read the man page for `traceroute` and then run `/usr/sbin/traceroute www.mit.edu`. How many hops are there between here and MIT? By looking at the names of the intermediate routers, can you tell where the packets went, geographically? Try a few other hosts that you know are nearby, like `www.babson.edu`. In some cases you will see that the packets take a surprisingly circuitous route. What is the most ridiculous route you can find?

Browse?

The nice people at Sun[1] have a tutorial explaining how to use sockets:

`http://java.sun.com/docs/books/tutorial/networking/sockets/`

Look over this material and the documentation of the Socket class.

1. Click on the link "Reading from and Writing to a Socket" and make a copy of their EchoClient class.

2. Change the program so that it connects to `rocky.wellesley.edu`, Port 80.

3. Compile and run the program. It should allow you to type an HTTP request, and it will print the first line of the reply from the server.

4. Read the documentation of the InetAddress class. Invoke `getInetAddress` on the Socket and print the name and IP address of the server. Use `getPort` and `getLocalPort` to see which ports are being used for this connection.

5. Read the documentation of `setTrafficClass` and then invoke `getTrafficClass` and interpret the result.

6. Print whatever other socket parameters you are interested in, like the send and receive buffer sizes, and the timeout value.

7. Modify the program so that it sends a legal HTTP request for the file named `index.html`, and then prints the entire reply from the server. You should get the HTML version of my homepage.

   The definition of the HTTP protocol is at

   `http://www.w3.org/Protocols/HTTP/HTTP2.html`

   The World Wide Web Consortium (W3C) is the organization that defines, maintains and disseminates the open standards that define the Web.

[1] Yes, the ones that keep telling me to stop using "Java" as a noun.