ECE 158A, Data Networks, Winter Quarter, 1997

Socket Programming Assignments

- **Problem 0.** Read the BSD socket programming tutorial, which can be printed out from the file ipctut.ps or ipctut.pdf available at

  `http://www-cwc.ucsd.edu/~cruz/ece.158/BSDsockets/`

  Also included in this directory are the C language source files of the examples discussed in the tutorial. Compile, run, and experiment with these files until you understand them well. (See the README file which contains information on linking to the BSD compatibility library if you are running under UNIX system V.). You may skip the sections (and example code) for UNIX domain datagrams and UNIX domain streams if you wish.

- **Problem 1.** Due in one week, i.e. 1/14/97. Port number 7 (decimal) is a "well known port" for the UDP protocol that provides an "echo" service. Many unix hosts throughout the world support this service, which upon receipt of a datagram, simply sends exactly the same datagram back to the source. Write a simple program that allows you to send a UDP datagram to a host at port 7, and read back potential replies. Find a host, preferably outside the U.S., which supports this service (send it a datagram at port 7 and see if anything comes back). In addition run the command `traceroute <HOSTNAME>` to find out the route used to send your datagram. (For some reason, on some systems you may need to omit the "e" in the traceroute command, i.e `tracroute <HOSTNAME>` .)

- **Problem 2.** Due in two weeks, i.e. 1/21/97. Write a simple "talk" application program that allows two users on separate machines to engage in a interactive text-based conversation over the Internet. Your program need not be "fancy," but it should be functional (i.e. don't worry about split screens for input and output, and you may assume that port numbers can be agreed upon beforehand). Write two different implementations, one that uses UDP (datagrams), and another that uses TCP/IP (streams). If you want to get fancy, see if you can write a rudimentary program that supports a conversation between more than two parties (i.e. a "chat room" application).

For each problem, write up a short (e.g. 1 page) description of your program(s), as well as a discussion of your results. Include a copy of your source code, and in addition, e-mail a copy of your source code to the TA (vtraore@cts4.ucsd.edu) with the subject "ECE 158A Problem n", for problem n, so that your results may be independently verified.

Use the Internet responsibly!. Malicious behavior is a crime, and is grounds for expulsion. Do not send large amounts of data, especially during peak hours, to cause network congestion just for fun. Do not send data to random ports to see what happens. Remember that you are accountable for your actions, which can be traced by authorities.

A word about working together: You may discuss your experiences and approaches with your colleagues, but each person must write their own unique programs. You may wish to experiment to see if you can get your programs to talk to one another (different approaches need not necessarily be compatible).