Please type your answers and submit the hardcopy output by 5pm in CS 164 on the due date. Submit the soft copies—programs, output, and written answers—using the turnin utility by the same due date & time. The course page (HW link) contains instructions on how to use turnin. For CS 536Q students, please use turnin for all your submissions as with on-campus students. The only difference is that hardcopy submission is not required.

PROBLEM 1 (20 pts)
Give a one-page summary and critique. Do you agree with their conclusions?

PROBLEM 2 (30 = 10 + 20 pts)
(a) What are the drawbacks of the current 32-bit IP address structure? What optimizations/refinements have been carried out on the 32 bits to make better use of the total address space? Assuming there are 80 million hosts on the Internet needing, say, one IP address each (some are, of course, multihomed), is the 32-bit address space, in principle, sufficient? What if you consider the class A, B, and C classification?

(b) Assume you have 128 bits to use to identify or assign names to hosts (i.e., interfaces) on the IP internetwork. With a suitable routing mechanism, every host should be efficiently found and without ambiguity. Propose your own design, how it is to function (i.e., routing mechanism at hosts connected to two or more LANs), and justify why you think it is a “good” design. First, it has to be correct design, and second, it should be efficient and “nice” in other senses that you will explain.

PROBLEM 3 (50 pts)
As a continuation of Problem 3(b), Assignment II, extend the client/server application such that client and server processes are allowed to run on different hosts and communication is achieved via socket calls using UDP/IP. Thus the FIFO IPC structure needs to be replaced by the appropriate networking calls. Note that UDP is a connectionless (i.e., stateless), unreliable transport protocol. Test your application using the two hosts where the server and client processes are run, respectively. Assume the server listens on a well-known port, say 10101. The client should accept three command-line arguments

% client.bin IP-address timeout command

where IP-address is in dotted decimal form.

On your host machine, open two windows, one for each separate host, running the server on one machine and the client on the other. Test the system by executing the client with the requested commands date, who am i, ps, ps -l, ps -l -a, ls, ls -a, ls -a -l -i, terse (e.g., % client.bin IP-address 500 ps -l). In addition to the code and script files, hand in a screen dump which shows the contents of the two windows.

PROBLEM 4 (50 pts)
Repeat Problem 3, albeit using TCP in place of UDP. Make your code modular so that the transport layer differences between Problems 3 and 4 are isolated to the extent possible. Discuss the pros/cons of the UDP- vs. TCP-based solution. Utilize the benchmark tests and any insights gained when formulating your answer.