CS 536 (Park) **Assignment I** **Due: Sep. 5 (Fri.), 1997**

PROBLEM 1

Read Chapt. 1 from Peterson & Davie (henceforth, P & D) and "The Design Philosophy of the DARPA Internet Protocols" by David Clark (in *Proc. SIGCOMM '88*, pp. 106–114, 1988. Give a 1-page summary of Clark's paper, stressing the parts that you find interesting and important. Include your own viewpoints and critique when writing your commentary. *Remark: Your write-up should be typed, not greater than 12pt font size, and not smaller than 10pt. Do not double space your writing.*

PROBLEM 2

(a) We mentioned in class that it is common to have three levels of addressing in computer networks—to identify the LAN in a WAN, the host in a LAN, and the process on a host. Assuming you need to facilitate communication between hosts in an internetwork, design a formatting solution to how the header of a packet or message should look like. Assume that there will never be more than X number of LANs, Y number of hosts on a *single* LAN, and Z number of processes on a single host.

A *router*, for present purposes, is a host that is attached to two different LANs. As a packet travels from source to destination hopping from router to router, what information in the header does a router need to examine? What additional information, if any, does the *destination router* (the one attached to the destination LAN) need to examine? What additional information, if any, does the destination host need to examine?

(b) Assume a host on the network can be *mobile*; i.e., like a notebook, it can be easily moved about and attached to different LANs pending on location. If your friends are to reach you by e-mail whereever you may roam, design the simplest scheme that you can think of that enables this form of "location transparency." Try to change as little of the set-up in **(a)**.

In part **(a)**, we have glanced over the fact that in operating systems such as UNIX, processes get assigned unique (nonnegative) integer IDs upon birth which get relinquished upon termination for use by other processes. Since the real end-points of a communication are processes residing on separate hosts running their own operating systems, how is one process to know how to contact its counterpart on the destination host? Design two different solutions to the problem, one based on the notion of *global* process IDs.

PROBLEM 3

Recall the expression of message delay given by $d(M) = D_P + D_T(M) + D_Q(M, \mathcal{S})$ where D_P is the propagation time, D_T is the transmission time, and D_Q is the queueing delay. Assume you need to send a message M of size S bytes to a place X miles away. Assume bits travel (encoded as physical signals) at 90% of the speed-of-light (SOL) c. Assume there are n routers along the way and each link has bandwidth B; throughput, however, is 80% of the bandwidth. The queueing delay D_Q, at *each* router, is between 500 μs and 2 ms.

Find the expressions for the most optimistic estimate of $d(M)$ (i.e., lower bound) and the most pessimistic estimate (i.e., upper bound). Assuming you are sending a file of size 1 MB from the east coast to the west coast (say $X = 3000$ miles) where the links are T3 links and there are 10 hops along the way, what are the lower and upper bounds on the time to complete sending the file?

Why is even the worst-case estimate, in real networks under heavy background traffic conditions (remember, a network is almost always shared with other traffic streams), an optimistic estimate? That is, what other important factor have we ignored here? Sketch an approach by which this ignored factor could be approximately incorporated into a formulae for $d(M)$. (*Hint: First, try to quantify this ignored factor by representing it as a variable, and then consider what actions—not necessarily unique—your protocol would take to respond to such "bad" events and how this would influence* $d(M)$.)