

**PROBLEM 1**

Read “Measured capacity of an Ethernet” by D. Boggs, J. Mogul, and C. Kent. *Proc. ACM SIGCOMM '88*, pages 222-234, 1988. Give a one-page summary and critique of the paper. Do you agree with their conclusions?

**PROBLEM 2**

Extend the client/server application of Assignment IV such that command-line arguments are supported. Note that redirection (`<`, `>`) and pipes (`|`) are shell features, and need not be implemented. Clearly indicate your new request format. Test your program by first running the server application in the background, then executing four copies of the client in the background “simultaneously” with arguments `finger your-hostname`, `/usr/ucb/ps -a -l`, and three more commands of your choice with options.

If you haven't done so already, add error checking capabilities to your server such that “Command not found” and other errors can be reported. Test the error checking capability by invoking the dummy command `dummy-command` and `/usr/ucb/ps -z` where “-z” is an illegal option.

Use the UNIX `script` facility to record the run-time session. Preserve your source code and binaries as you may be asked to provide them to the TAs for run-time checking. Turn in the output of the script as well as the client and server code. You must provide adequate documentation in your code. (*This hand-in format is the default requirement unless otherwise specified.*)

**PROBLEM 3**

(a) As a continuation of Problem 2, if you had to implement redirection (`<`, `>`) and pipes (`|`) as part of the server capabilities, how would you go about doing so? That is, what are the technical problems to be solved and what are possible solutions? Be precise in your descriptions so that coding is only a stone throw away.

(b) What are the barriers to implementing the above server as an *iterative* server? If you were captured by aliens and your life depended on reimplementing the above server as an iterative server and you were given a year to do so, how would you go about accomplishing the task?

What if you were given only 5 minutes to do so? Is there still hope assuming the aliens are not experts—but “knowledgable”—in UNIX system programming? (*Hint: The goal here would be to deceive the aliens into thinking that your reimplementation is an iterative server when, in fact, it is not.*)

**PROBLEM 4**

Assume our goal is to turn a CSMA/CD Ethernet into a token ring. That is, while preserving Ethernet's existing MAC, we want to build a system—in software—that emulates a token ring-based access mechanism with its associated deterministic properties. How would you go about doing that? Give a detailed sketch of a design and explain how it is able to function as a token ring.

Assuming that you are able to keep your software overhead to a negligible level, under what circumstances, if any, would your system perform better than the pure CSMA/CD Ethernet with respect to throughput?