CS 422 (Park) **Assignment II** **Due: Sep. 19 (Wed.), 2007**

Submission instructions: Please type your answers and submit electronic copies using `turnin` *by 5pm on the due date. You may use any number of word processing software (e.g., Framemaker, Word, LaTeX), but the final output must be in pdf or ps format that uses standard fonts (a practical test is to check if the pdf/ps file prints on a CS Department printer without problem). For experiments and programming assignments that involve output to terminal, please use* `script` *to record the output and submit the output file. Use* `gnuplot` *to plot graphs. Use* `ps2gif` *to convert a eps/ps plot to gif format (e.g., for inclusion in Word) if there is a need.*

PROBLEM 1

Read Chapters 3, 5, 6, and 7 from Comer. Solve Problems 5.1, 5.10, 6.10, and 7.7.

PROBLEM 2 (30 pts)

(a) Suppose TDM is used to support 1024 simultaneous users over a shared link (wired or wireless). Each user needs to get 1 Mbps. A sine wave of period T sec (i.e., frequency $f = 1/T$) is used to encode a bit where an amplitude of 20 means bit value "1" and an amplitude of 10 represents bit value "0". What must the value of T be so that all 1024 users achieve their required bandwidth? What is the frequency of the sine wave? How many bits would you allocate per time slot (i.e., packet or frame size)? Does this matter?

(b) Suppose that after a year of operating the TDM system (e.g., for running TCP/IP in cell phones using TDM), competition requires that user bandwidth be increased from 1 Mbps to 4 Mbps. Your EE employees inform you that period T cannot be decreased without investing heavily in new equipment which is too costly. However, manipulating the amplitude so that more levels than 10 and 20 are recognized is feasible by updating firmware. Discuss two alternative solutions to the problem. Which one do you prefer, and why?

PROBLEM 3 (40 pts)

(a) Suppose in a CDMA system, 5 users must be simulaneously supported. Let $(a_1, a_2, a_3, a_4, a_5)$ be the basis vector assigned to user A, $(b_1, b_2, b_3, b_4, b_5)$ to user B, $\ldots$, $(e_1, e_2, e_3, e_4, e_5)$ to user E. Find an orthonormal basis that is different from $(1, 0, 0, 0, 0)$, $(0, 1, 0, 0, 0)$, $\ldots$, $(0, 0, 0, 0, 1)$ (time to make use of linear algebra; negative numbers are fine and are used in practice). Suppose A sends a single bit 1, B sends 0, C sends 0, D sends 1, and E sends 0. As in Problem 2(a), an amplitude (i.e., coefficient or spectrum) of 20 is used to represent bit value "1" and 10 to encode "0". Show what are the vectors transmitted by A, B, $\ldots$, E, respectively. Show what is the combined vector seen by a base station (attached to a cell tower, say, along I65 going to Indy). Show how the base station can disentangle from from the combined vector that it has received what bit value from user D. How would the CDMA work when each user has 8 bits to send?

(b) Suppose an eavesdropper is monitoring the airwaves (maybe out to sniff passwords that have not been encrypted, say, using SSH, a higher layer protocol), and thus seeing the same combined vector that the base station is receiving. If you were the eavesdropper, how would you go about determining what user D sent? Discuss how easy or difficult this task may be. Using encryption at the higher layers in the protocol stack (e.g., HTTPS in place of HTTP), when sensitive information is transmitted, is highly advised. Assuming encryption is not employed at higher layers nor lower layers (including CDMA), how would you operate the basic CDMA system so that eavesdropping success is further decreased?

PROBLEM 4 (30 pts)

You will find a client/server application under /u/u9/park/pub/cs422; the client program is `client.c` and the server program is `server.c`. Compile, run, and check their behavior. The server binary should be executed first (execute the server and client from different windows). Run the client binary multiple times. Use `script` to record the output. Reverse-engineer the code and explain what the server and client are doing. The most important technical aspect of this exercise is to understand and explicate what `dup2` is doing. What is the role of `waitpid` near the end of the server code? What happens if you leave it out? Is it necessary for the proper functioning of the server?

PROBLEM 5 (50 pts)

Rewrite the client/server application (currently running on a single host communicating using FIFO; we will change it later to run on different hosts communicating over TCP/IP sockets) such that the client can request any command (not just `finger`) to be executed by the server and the result returned to the client. To reduce secondary implementation overhead, you may ignore command-line arguments (e.g., "`ls -l`", "`ps gaux`" need not be supported at this time but "`ls`" and "`ps`" must work). The client program, call it, `client.bin`, accepts the command to be executed by the server as a command-line argument. That is,

% `client.bin command`

The client's request "packet" should be backward-compatible (a key requirement in most real-world systems), meaning that when the client is invoked without a command then a request sent to the original server (i.e., the one where `finger` is hard-coded) works as before. When a command is specified, then define an extended request format that builds on the legacy format such that it is correctly parsed by the new server that handles arbitrary command requests. You will also need to consider technical details such as what function in the `exec` family (`execlp` may not be the most convenient call) to invoke. Perform benchmark runs with `ls`, `ps`, and three other commands of your choice. Log the interaction using `script` as before. What happens when you send a client request with a specified command to the original server? Does it remain backward-compatible? What happens when the client sends a request of a command that does not exist (say `terve`)? Is there a need for the client to first check if a command exists?