CS 422 (Park) Assignment II Due: Feb. 12 (Mon.), 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 (20 pts)

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

PROBLEM 2 (30 pts)

(a) Suppose FDM with 50 carrier frequencies is being used in the 10–15 GHz range, with the first carrier frequency f_1 at 10 GHz, the second carrier frequency $f_2 = 10.1$ GHz, $f_3 = 10.2$ GHz, …, $f_{50} = 14.9$ GHz. That is, adjacent carrier frequencies are spaced 100 MHz apart (guardband) to reduce mutual interference. Suppose that AM with two levels is used, and a single period of a sine wave is used as the baud to represent a single bit. What is the speed or data rate (in bps) of this FDM communication system? Make sure to show how you arrive at the solution.

(b) Fixing the frequency range, the number of carrier frequencies, and sticking to the AM way of encoding bits (no FM or PM allowed), what are two different ways to increase the bps two-fold? Can one of the two methods be used to increase the data rate 32-fold? In the latter, how long does a single data bit, on average, get delayed at the sender at carrier frequency 10 GHz before it heads out on the link? Do you consider this a large delay? Compare it to the typical seek time of today's commodity hard disks.

(c) Suppose you were to visualize (if you had superman's GHz frequency vision) how many bits were traveling on the $f_1 = 10$ GHz carrier wave over a space of 1 meter (about 3.28 feet). How many bits are packed, in-flight, within this stretch of distance? (Since you don't have superman's GHz vision you'll have to calculate.) Keep in mind that physicists have confirmed over the years that electromagnetic waves travel at the same, constant speed: the speed of light. Suppose the distance from Purdue to a beach near San Diego is 2000 miles. How big a file (MB) do you have to transmit so that when the first bit of the file hits the west coast the last bit of the file is just leaving Purdue?

PROBLEM 3 (25 pts)

Suppose that you are the president of an audio club, *GoodEar*, whose human members (canines are automatically admitted) are folks who can discern audio frequencies in the 20 KHz range. Being blessed (or cursed) with such sensitivity, they feel dissatisfied having to listen to digital radio (FM and AM radio are banned from club premises) broadcast by Sirius and XM. So, you start a new satellite radio company (making money is not the object) that broadcasts 3000 digital radio channels, each channel able to carry audio frequencies up to 40 KHz (a demand placed by canine subscribers). The amplitude of an analog audio signal is quantized into 4294967296 levels. Disliking FM and AM, which both use analog FDM, you decide to employ TDM to broadcast the 3000 channel ultra-super-quality digital audio to your club members in West Lafayette and a few other places on the globe. Describe the technical details of your TDM satellite radio system, showing how you arrive at the final data rate (bps) of the communication system. Assuming that 1 Mbps bandwidth from a tier-1 ISP goes at the market price of \$25/month (minimum 25 Mbps subscription) and your club boasts 55 members, how much should you charge each member to break even assuming you yourself buy satellite bandwidth from a tier-1 ISP? Launching one's own GEO satellites is quite a bit more expensive, and we will ignore that \$25/month are landline bandwidth prices. And, you may give yourself a \$500000 monthly salary. Also, remembering from CS422 that long sequences of 0 (or 1) bits when transmitted using square waves can cause counting errors due to inaccurate clocks, you decide wisely to use 4B/5B encoding. Your system design and bit rate calculations should reflect this additional overhead.

PROBLEM 4 (20 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`. Reverse-engineer the code and explain what the server and client are doing. 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. The most important technical aspect of this exercise is to understand what `dup2` is doing.

PROBLEM 5 (60 pts)

(a) The client/server code, as far as client/server coding is concerned, uses a subtle technique to pipe the *stdout* output of a *legacy application* back to a client. (In Problem 4, you are asked to explain how this technically works.) Use the client/server code as a starting template, but simplify it significantly such that the server, to any contacting client, returns the string "I don't know" upon receiving a client request. Whereas the original server is *concurrent*—it forks a child and let's the child do the brunt of the work including returning a response to the client—make the simplified server *iterative*. That is, there is only a single process, the server, who does all the work. In this instance, sending the "I don't know" response to the client, then parsing the next client request. Test the client/server application by running multiple instances of the client binary as in Problem 4. Use `script` to record the output.

(b) Rewrite (a) such that the server is now concurrent. Also, make the server more lightweight by letting the child process handle the brunt of the parsing chore, including the creation of a FIFO that is used to return the "I don't know" message. The server, however, needs to do some minimal parsing, namely, dequeueing of each client request from the server FIFO (forwinding until the character '#') and handing the client's process ID as a string to a newly forked child. The client converts the string into an integer.

(c) Rewrite the original server in Problem 4 as an iterative server that continues to use the `dup2` trick. What happens when you run the client once? What happens when you run the client a second time? Is there a way to write an iterative server that provides the same service as the concurrent server in Problem 4? Explain your reasoning.