CS 422 (Park) **Assignment IV** **Due: Mar. 10 (Fri.), 2006**

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 10 and 11 from Comer.

PROBLEM 2 (30 pts)

(a) Assume there are k hosts competing for a time slot in a baseband network—wireline or wireless—using MA (multiple access). Each host, independently, chooses to send a frame with probability p. If two or more hosts decide to send, there is a collision. If no host decides to send, the time slot goes to waste. In the lectures, we showed that the probability that some host successfully acquires the slot is given by $\eta = kp(1-p)^{k-1}$. Since each host has only one degree of freedom—how to choose the probability p of sending (and all hosts much use the same probability)—we are interesting in finding the value of p that maximizes the probability that the slot is usefully used. Show why the optimum value is $p = 1/k$. (*Hint: View η as a function of p and find its peak.*)

(b) In the bakery problem discussed in class, k customers must independently pick a number between 1, 2, ..., k. If two or more customers pick the same number, they go home hungry. Of those customers who have picked a number that no one else has picked, the customer with the smallest number gets serviced first. Assuming each customer's goal is not to leave empty-handed, how should a customer pick a number such that he/she maximizes the probability of returning home with something to show? Solve the problem by relating it to Problem 2(a).

(c) In Problem 2(a), one might think that as the number of hosts k becomes very large, it must be that with probability close to 1 the slot will incur a collision. When is this true? When is this not true? Why is this not true?

PROBLEM 3 (30 pts)

(a) In the lectures, we showed that the utilization of Ethernet (under simplifying assumptions) can be approximated by $\varrho = E[\text{good}]/(E[\text{good}] + E[\text{bad}])$. We calculated $E[\text{bad}]$ and found it to be $1/\eta$. What is a much simpler way of showing that $E[\text{bad}]$ must be $1/\eta$? (*Hint: Consider coin tosses where heads come up with probability, say, 1/10, which implies that on average one would expect to see heads every 10th throw.*) Based on this reasoning, what must be the expression for $E[\text{good}]$?

(b) When expressing ϱ as a function of bandwidth B, frame size F, and wire length L (the three key parameters of an Ethernet LAN design), why are the exact expressions of $E[\text{good}]$ (and $E[\text{bad}]$) not essential for showing how utilization depends on B, F, and L? Note that in the lectures we referred to $E[\text{good}]$ by γ (a notational change).

(c) Keeping in mind that Ethernet uses CSMA/CD with exponential backoff, what were all the simplifications we made when deriving utilization ϱ? For each simplification we made, what would have been the impact on ϱ of not making the simplication? That is, would it have decreased ϱ or increased it?

PROBLEM 4 (30 pts)

(a) Suppose that you could reprogram your Ethernet NIC card so that the way that it implements CSMA/CD (e.g., backoff) can be modified. Knowing that all other hosts are using the traditional CSMA/CD with exponential backoff, how would you reprogram the NIC (call it "greedy CSMA/CD") so that it allows you to eat up (nearly)

all the bandwidth at the expense of starving others? How would you reprogram the NIC if your goal were only to "disable" the Ethernet LAN by performing a denial-of-service (DoS) attack?

(b) What are possible defenses against such misbehaving NIC cards? Discuss your top 2 defenses with respect to their strengths and weaknesses.

PROBLEM 5 (30 pts)

As a continuation of Problem 4, Assignment III, crash your server by sending a request that will break it. Sending a non-existent command is not "breaking" it since it just responds with *command not found*, as you've already discovered when testing with *terve*. You must break the server that you submitted in Assignment III (the TA will test with the submitted version), not one modified for this assignment with weaknesses such as backdoors. How can you fix the vulnerability that you've discovered? Implement the fix, and show that it works.