CpSc 418 **Homework 2** **Due 29 October 2002**

Four problems, total value 180 points.

1. **Trends:** (40 points)
   Consider a current hard disk, in particular IBM's *Ultrastar 146Z10*, (see
   http://www.storage.ibm.com/hdd/ultra/ul146z10.htm
).
   (a) (10 points) Compare this disk with the trends predicted in the paper by Ruemmler and Wilkes, "An Introduction to Disk Drive Modeling," in terms of track and bit density, rotational speed, capacity, and diameter. Which predictions held, and which did not? Why?
   (b) (10 points) Compare the IBM disk with the "First Law in Disk Density" quoted in the paper by Patterson, Gibson and Kats, "A Case for Redundant Arrays of Inexpensive Disks (RAID)". By how many percent does the actual density differ from the prediction? According to the "First Law," in what year should the current density have been achieved. By how much was it off?
   (c) (20 points) The RAID paper also makes predictions about the number of transistors on a chip (Moore's Law), the performance of microprocessors (Joy's Law), and the ratio of megabytes of main memory to processor MIPS (*alpha*). Find data for current values for these measures and compare with the predictions. Cite your source for the data you use and any assumptions that you need to make. By how many percent were the predictions off? By how many years were the predictions off?

2. **Network Switches:** (40 points)
   Consider a network switch with N input ports and N output ports using cross-bar interconnect. This problem examines the maximum throughput for highly non-uniform traffic when iSLIP scheduling is used. Assume that when multiple input ports offer traffic to an output port, the output selects an input uniformally at random. Likewise, assume that when an input receives multiple return offers from outputs, the input card makes the final selection uniformally at random.

   Assume that in each time slot, each input card receives a packet with probability α. This problem explores the maximum value that α can have before a switch with arbitrarily large buffers will end up dropping packets. Packets arriving at input card i are destined for output ports i, $i \oplus 1$, $i \oplus 2$, $\ldots i \oplus (k-1)$ for some integer k, and where $\oplus$ indicates addition modulo N. Packets at input card i are destined for the k possible outputs uniformally at random.

   (a) (20 points) Write a program to experimentally estimate the maximum value of α before the buffers overflow for $k \in \{0, 1, 2, 3, 4, 8\}$.
   (b) (20 points) For $k \in \{1, 2, 8\}$, and α ranging from 0 to $0.95\alpha_{\max}(k)$, determine the average time that a packet waits in the switch. Determine the time by which 50%, 80%, and 90% and 99% of all packets are routed through the switch.

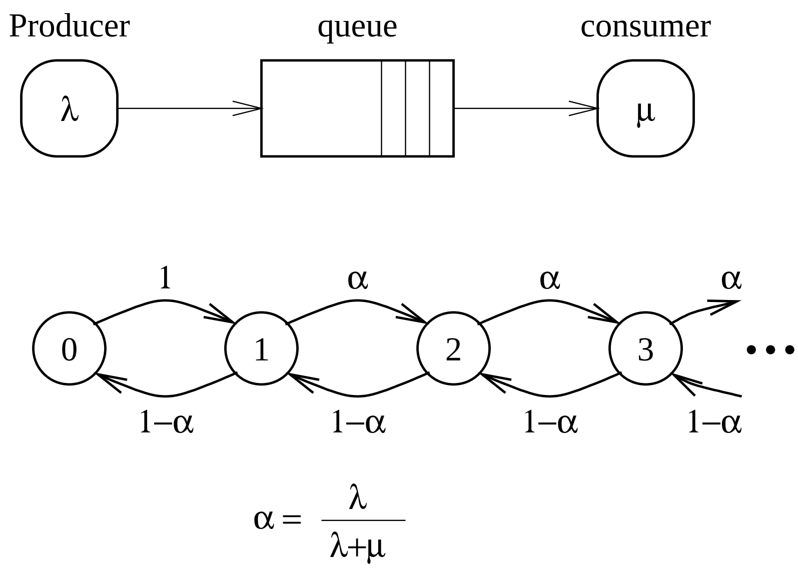

Figure 1: A Simple Queue

3. **Analysing Queues:** (30 points)
   Consider the simple queue shown in figure **??**. The producer generates items that we will say *arrive* at the queue. The times between arrivals are exponentially distributed with mean $1/\lambda$; in other words, λ is the *arrival rate*. When the queue is non-empty, the consumer takes some amount of time to service the item and then removes it from the queue. The service time is exponentially distributed with mean $1/\mu$; in other words, μ is the *service rate*.

   As an example, consider a grocery store with a single check-out stand. The queue is the line-up at the check-out stand. Arrivals correspond to customers getting into the check-out line. Service time is the amount of time it takes the clerk to tally up the groceries, receive payment, and make change. We are interested in the length of the line of customers.

   The lower half of figure **??** shows a state transition diagram for the queue. Each state is labeled with the number of items in the queue for that state. Arrows show the transitions between the state. An arrow to the right corresponds to an arrival, and an arrow to the left corresponds to a departure (i.e. a completion of service).

   From state 0, only arrivals can occur. Therefore, the successor to state 0 is state 1 with probability 1.

   For $i > 0$, the successor to state i can be state $i + 1$ if the next event is an arrival, and state $i - 1$ if the next event is a departure. It is straightforward to show that the time spent in state i is exponentially distributed with mean $1/(\lambda + \mu)$; in other words, it is a process with rate $\lambda + \mu$. Furthermore, the probability that the transition out of state i is an arrival is $\lambda/(\lambda + \mu)$, and the probability that the transition is a departure is $\mu/(\lambda + \mu)$.

   (a) (20 points) Let p_i be the probability that the queue is in state i immediately after a transition. For this problem, you will calculate p_i.

Here are some hints. First, in steady state:

$$p_i = \alpha p_{i-1} + (1-\alpha)p_{i+1} \quad \text{, for } i \geq 2 \tag{1}$$

where $\alpha = \lambda/(\lambda+\mu)$. Show that for $i \geq 1$, there is some β such that $p_{i+1} = \beta p_i$. Solve for β. Include your derivation of β in your solution.

Now, you have

$$p_i = \beta^{i-1} p_1 \quad \text{, for } i \geq 2 \tag{2}$$

The only transition into state 0 is from state 1, and this transition occurs with probability $(1-\alpha)$. Therefore

$$p_0 = (1-\alpha)p_1 \tag{3}$$

You now have equations for all of the p_i in terms of p_1. The queue is in some state with probability 1; this yields:

$$\sum_{i=0}^{infty} p_i = 1 \tag{4}$$

Use your formulas for p_0 and p_i and solve for p_1. You now have probabilities for all of the states.

(b) (10 points) In part **??**, you derived the probabilities immediately after a transition. Different states have different lifetimes. In particular, state 0 has an mean lifetime of $1/\lambda$, and all other states have a lifetime of $1/(\lambda+\mu)$. State 0 has a longer lifetime because it can only be exited by an arrival whereas the other states can be exited by either an arrival or a departure. The time-average length of the queue is given by

$$E(\text{queue length}) = \sum_{i=0}^{infty} i * t_i * p_i \tag{5}$$

where t_i is the mean lifetime of state i. Derive the time-average length of the queue as a function of λ/μ. Plot this function for $0 \leq \lambda/\mu < 1$.

4. **RAID:** (60 points)

By the end of the week, I will post Java code to the course website for a simple disk-array simulator. I will post to the newsgroup ubc.courses.cpsc.418 when the code is available.

(a) (40 points)

The code will include an abstract class called AbstractRAID. This class has three abstract methods: read, write, and recover. Write a class that extends AbstractRAID and implements these methods. Use the static method test in class RAID_Tester to test your implementation.

(b) (30 points)

Add methods largeRead, and largeWrite to your class. Plot S from table VI of the RAID paper as a function of the number of disks, D in the RAID array, for D in $[2, 25]$.