UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2003

BEng Honours Degree in Computing Part III
MSc in Computing for Industry
BEng Honours Degree in Information Systems Engineering Part III
MEng Honours Degree in Information Systems Engineering Part III
BSc Honours Degree in Mathematics and Computer Science Part III
MSci Honours Degree in Mathematics and Computer Science Part III
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute
This paper is also taken for the relevant examinations for the Associateship of the Royal College of Science

PAPER C337=I3.18

SIMULATION AND MODELLING

Tuesday 6 May 2003, 10:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators required

1 In a transaction processing system processes submit a sequence of read and write transactions to a database. Access to the database is mediated via a database lock that ensures that each process sees a consistent view of the database. Processes request the lock when they want to perform a read/write transaction and release the lock when the transaction is finished.

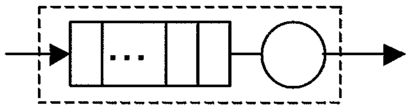
The lock works as follows: If no process is currently accessing the database (i.e. no process has the lock) a process can claim the lock to perform a write. If a process has acquired the lock to perform a write (but has not yet released it) no other process can be granted the lock. If no process has acquired the lock to perform a write then any number of processes can acquire the lock simultaneously to perform a read transaction. If readers and writers are both waiting for a writer to release the lock, priority is given to the readers.

Processes are independent and execute on separate processors, so there is no resource contention among processes apart from the lock. It has been established that the time between the creation ("arrival") instants of new processes has a distribution that can be sampled by the method `nextP()`. The number of transactions generated by each process has a distribution that can be sampled by the method `tcount()`. Within a process, transactions are independent; a transaction is a read transaction with probability r and a write with probability (1-r). Read transactions hold a lock for a time that can be sampled by the method `holdR()`; similarly `holdW()` for write transactions. Processes perform local processing between transactions; the local processing times have a distribution that can be sampled by the method `think()`. Processes that are waiting to acquire the lock for a read are held in a FIFO (First-In-First-Out) 'reader' queue; writers are similarly held in a 'writer' queue. The number of processes that can exist at any time is unbounded.

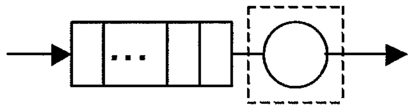
Design a discrete-event simulation of this system that estimates, on a given run, the mean time that a write transaction has to wait before acquiring the lock.

You may use any notation you wish but should summarise *briefly* at the outset any library methods/classes you assume, e.g. for managing queues, events, process interaction etc. Avoid unnecessary detail.

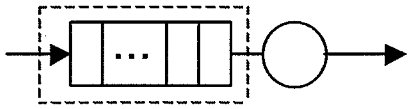
2a Little's Law relates the arrival rate (throughput), λ, mean population, L, and mean waiting time, W, of *any* system in equilibrium thus: $L = \lambda W$. In each diagram below a different "subsystem" of a single-server queue is shown enclosed in the dotted box. In each case the customer at the head of the queue is removed before going into service. For each subsystem, what do L and W correspond to? Explain your answers.



(i)



(ii)



(iii)

Part b continues overleaf

2b A particular processing plant handles two types of jobs: a proportion $100q\%$ of jobs are *simple* jobs which are processed by a machine M_1 each requiring an average service time of μ_1^{-1} hours; the remaining $100(1-q)\%$ of jobs are *complex* jobs, which are like simple jobs except they require additional processing by a machine M_2 which takes an average time of μ_2^{-1} hours. The arrival rate of jobs to the plant is λ jobs per hour and jobs waiting to be processed are queued up in very large storage areas which are well approximated by infinite capacity queues. The current set up is shown in the form of a queueing network in Figure 1 below.

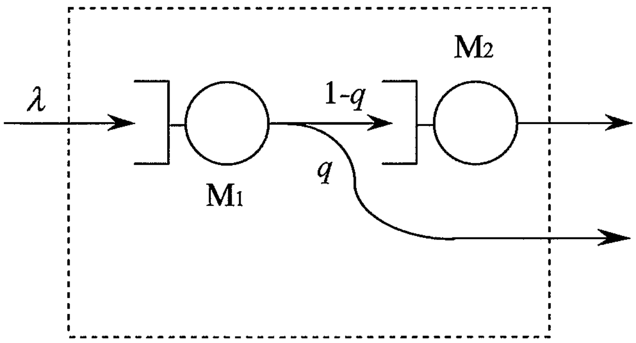


Figure 1

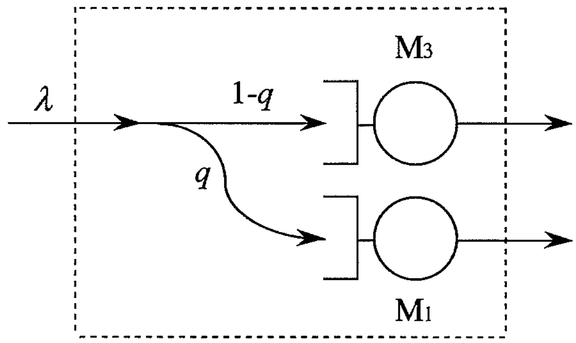


Figure 2

The owner of the plant wants to reduce the average processing time of jobs and is seeking an alternative set up. He has been offered a new machine M_3 which is capable of processing a complex job in the same mean time as the existing plant (i.e. $\mu_1^{-1} + \mu_2^{-1}$ hours) and the idea is to trade in machine M_2 for machine M_3 and use M_1 and M_3 in combination as shown in Figure 2. The various parameters have the following values:

$$\lambda = 5 \qquad \mu_1 = 10 \qquad \mu_2 = 5 \qquad q = 0.5$$

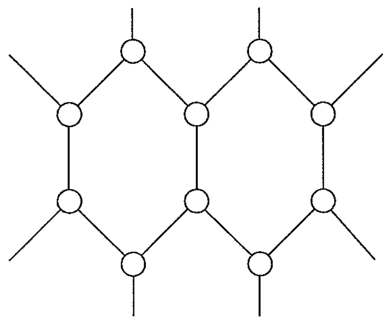
Assuming that the arrival process is Poisson and that the service times of all three machines are exponentially distributed, compute the following equilibrium quantities for both set-ups:

i. The utilisation of each machine
ii. The mean number of jobs in the queue associated with each machine, including any in service
iii. The mean total processing time for each job (i.e. the mean time each job spends inside the boxed region in the respective diagram).

On the basis of these calculations what recommendation would you make to the owner of the plant? Justify your answer.

The two parts carry 30% and 70% (20+20+20+10 for recommendation) of the marks respectively.

3 A packet-switched mobile telecommunications network has been proposed which consists of a set of nodes, each with three I/O ports, interconnected in a mesh:



At certain times, a node (the *transmitter)* will initiate a communication with another node (the *receiver*). This takes the form of a sequence of fixed-sized packets which are routed to the receiving node in accordance with a routing algorithm. The distribution of the number of packets in each message is the same for all transmitters. The routing algorithm tells the transmitter on which of its three output links it should send the packet first, although the details are unimportant here. When a packet arrives on an input link the algorithm determines whether the packet is destined for that node (i.e. if it is the receiver of the packet) and, if it is not, which output link to forward it to in order to reach the destination. *Packets may not arrive at the destination in the same order they left the receiver.* Each output link has a FIFO (first-in-first-out) buffer containing any outstanding packets waiting to use the link.

A stochastic discrete-event simulator has been written to model this system for experimental purposes, but no measurement code has yet been incorporated. You are asked to advise the simulation designer on various aspects of the measurement process.

a Explain *briefly* how the designer might incorporate code to estimate the following quantities (do *not* attempt to detail the required code):

i. The mean number of packets queued at a given output buffer

ii. The distribution (in the form of a histogram) of the time between the first packet of a given message leaving the transmitter and the last packet of the message reaching the receiver.

iii. The distribution (in the form of a histogram) of the number of packets in a specified output buffer in the network.

b It is required to take the above measurements when the system is at equilibrium. Explain what is meant by equilibrium and how you would detect when approximate equilibrium has been reached in the simulation.

c Having incorporated the measurement code, the simulator is executed eight times, with each run being based on a different random number sequence. The mean packet transmission times in milliseconds recorded on each run are as follows:

26.2	29.4	25.5	27.9	24.8	26.1	28.0	25.1

Compute the 90% confidence interval for the mean transmission time.

The three parts carry 45%, 20% and 35% of the marks respectively.

4a Give four reasons why it is preferable to sample a mathematical distribution to generate inputs to a discrete-event simulation, rather than use data obtained by observation of a real-world system.

4b A Poisson random variable X with parameter λ has a probability distribution function:

$$\Pr\{X = x\} = p_x = \frac{e^{-\lambda}\lambda^x}{x!} \quad i = 0,1,2...$$

Show that $p_{x+1} = \dfrac{\lambda}{x+1} p_x$ and use this property to design a method for sampling a Poisson distribution with parameter λ. Efficiency will be taken into account when awarding marks.

4c Prove that if X is a random variable with cumulative distribution function (cdf) $F(x),\ x \geq 0$, then $R = F(X)$ is uniformly distributed on the interval (0,1). Using this result explain the principles of the *inverse transform* method for sampling a distribution. As part of your answer show how the method can be used to sample the Weibull distribution whose cdf is given by:

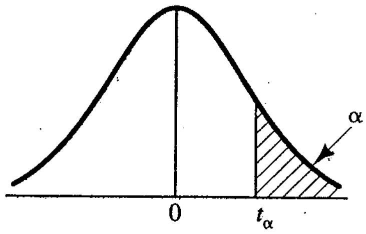
$$F(x) = 1 - \exp(-(x/\alpha)^{\beta}) \qquad x \geq 0$$

for given parameters α and β.

4d. A packet of mixed Petunia seeds contains on average $N = n_{c_1} + ... + n_{c_M}$ seeds where n_{c_j} is the number of seeds which, when grown, will flower with colour $c_j,\ 1 \leq j \leq M$. Outline code that will sample this (empirical) distribution and so model the constitution of a randomly-generated seed packet containing N mixed seeds. In what sense is your sampler exploiting the inverse transform method of part c?

The four parts carry 20%, 25%, 30% and 25% of the marks respectively.

Table A.5. PERCENTAGE POINTS OF THE STUDENTS t DISTRIBUTION WITH v DEGREES OF FREEDOM



v	$t_{0.005}$	$t_{0.01}$	$t_{0.025}$	$t_{0.05}$	$t_{0.10}$
1	63.66	31.82	12.71	6.31	3.08
2	9.92	6.92	4.30	2.92	1.89
3	5.84	4.54	3.18	2.35	1.64
4	4.60	3.75	2.78	2.13	1.53
5	4.03	3.36	2.57	2.02	1.48
6	3.71	3.14	2.45	1.94	1.44
7	3.50	3.00	2.36	1.90	1.42
8	3.36	2.90	2.31	1.86	1.40
9	3.25	2.82	2.26	1.83	1.38
10	3.17	2.76	2.23	1.81	1.37
11	3.11	2.72	2.20	1.80	1.36
12	3.06	2.68	2.18	1.78	1.36
13	3.01	2.65	2.16	1.77	1.35
14	2.98	2.62	2.14	1.76	1.34
15	2.95	2.60	2.13	1.75	1.34
16	2.92	2.58	2.12	1.75	1.34
17	2.90	2.57	2.11	1.74	1.33
18	2.88	2.55	2.10	1.73	1.33
19	2.86	2.54	2.09	1.73	1.33
20	2.84	2.53	2.09	1.72	1.32
21	2.83	2.52	2.08	1.72	1.32
22	2.82	2.51	2.07	1.72	1.32
23	2.81	2.50	2.07	1.71	1.32
24	2.80	2.49	2.06	1.71	1.32
25	2.79	2.48	2.06	1.71	1.32
26	2.78	2.48	2.06	1.71	1.32
27	2.77	2.47	2.05	1.70	1.31
28	2.76	2.47	2.05	1.70	1.31
29	2.76	2.46	2.04	1.70	1.31
30	2.75	2.46	2.04	1.70	1.31
40	2.70	2.42	2.02	1.68	1.30
60	2.66	2.39	2.00	1.67	1.30
120	2.62	2.36	1.98	1.66	1.29
∞	2.58	2.33	1.96	1.645	1.28

Source: Robert E. Shannon, *Systems Simulation: The Art and Science*, ©1975, p. 375. Reprinted by permission of Prentice-Hall, Upper Saddle River, NJ.