0580024

UNIVERSITY OF YORK

MSc Examinations 2004
MATHEMATICS
Networks

Time Allowed: 2 hours.

Answer all four questions.
Standard calculators will be provided but should be unnecessary.

1. (i) Draw a graph showing the arrivals and departures against time τ for a *FIFO* queue assuming that the system starts empty. Mark on your graph:

- The arrival time of the first three customers, t_1, t_2 and t_3.
- The queuing time of the first three customers, $T(1)$, $T(2)$ and $T(3)$.
- The curve $\alpha(\tau)$ showing arrivals to the system in $[0, \tau]$ and the curve $\beta(\tau)$ showing departures from the system in $[0, \tau]$.
- The number of customers in the system $N(s)$ for some chosen time s.

(5 marks)

(ii) Prove, Little's theorem $N = \lambda T$ where N is the average number of customers in system, λ is the average arrival rate per unit time and T is the average number of time units a customer spends in the system. You may assume that the system is initially empty $N(0) = 0$, that the system is FIFO and that for any time τ there is always some time $t > \tau$ such that $N(t) = 0$. State clearly any other assumptions you make in your proof.

(15 marks)

(iii) Assume that a person browsing the web site looks at a page for an average of P seconds and then gets a new page which takes R seconds on average to display (from the time the user makes the request to the time they can start looking at the page). If N people are browsing the web then use Little's Theorem to calculate λ the number of pages per second served. If the web server can serve up to λ_m pages per second then calculate N_m the maximum number of people who can can browse the site.

(5 marks)

2. (i) Name and briefly (one or two sentences) describe the layers of the TCP/IP reference model. **(4 marks)**

(ii) Consider a queuing system for a computer which processes files in discrete time periods. If the computer is idle at the beginning of a time cycle then there is a probability p_a it will begin to process a new file. If te computer is busy processing at the beginning of a time period then there is a probability p_b that it will complete processing the file at the end of the time period (this is independent of how many time periods of processing the computer has been working for already). Draw a two state Markov process representing the process.

(5 marks)

(iii) If $p_a = 2/5$ and $p_b = 2/3$ then the transition matrix of the chain is given by

$$\mathbf{P} = \begin{bmatrix} 3/5 & 2/5 \\ 2/3 & 1/3 \end{bmatrix}.$$

Find the eigenvalues of the matrix λ_1 and λ_2. **(5 marks)**

(iv) Assuming that the states of the chain are numbered 0 (representing the idle state) and 1 (representing the busy state) then find an equation for the n-step transition probability from state 0 to itself. $p_{00}^{(n)}$ using the eigenvalues previously calculated. What does this imply about the long term probability of finding the system empty?

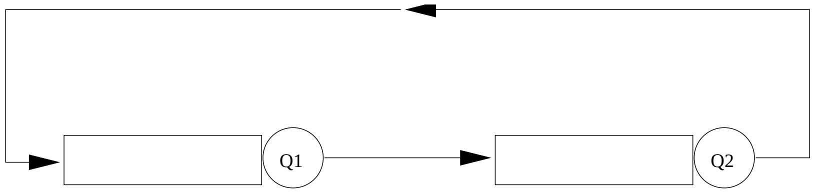
(7 marks)

(v) Write down the balance equations for the chain and, from these, calculate π_0 and π_1 (the equilibrium probabilities of the chain) from these.

(4 marks)

continued on next page

3. Consider the network shown below where the packets leaving the output queue of one computer $Q1$ feed directly into the input queue of a second $Q2$. Both queues are served as Poisson processes with rates μ_1 and μ_2 respectively. There are a total of K packets in the system. Assume the packets can travel between the output of one queue and the input of another instantaneously.



(i) Draw a Markov chain where the states 0 to K represent the number of packets at $Q1$ as a birth-death process.
(4 marks)

(ii) The transition matrix for the chain in the previous question is given by $\mathbf{P} =$

$$\begin{bmatrix} 1-\mu_2 & \mu_2 & 0 & \cdots & 0 & 0 \\ \mu_1 & 1-\mu_2-\mu_1 & \mu_2 & \cdots & 0 & 0 \\ 0 & \mu_1 & 1-\mu_2-\mu_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1-\mu_2-\mu_1 & \mu_2 \\ 0 & 0 & 0 & \cdots & \mu_1 & 1-\mu_1 \end{bmatrix}.$$

In standard queuing theory notation, what type of queue is $Q1$ equivalent to?
(1 mark)

(iii) Write down the balance equations for states π_0, π_K and π_i where $0 < i < K$.
(3 marks)

(iv) Hence, or otherwise, show that

$$\pi_k = \frac{(\mu_2/\mu_1)^k}{\sum_{i=0}^{K}(\mu_2/\mu_1)^i},$$

for $0 \le k \le K$.
Hint: It may help to prove by recursion that $\pi_{i+1} = (\mu_2/\mu_1)\pi_i$ for $0 \le i < K$.
(10 marks)

(v) If $\mu_1 = \mu_2 = \mu$ then find the value of N, the expected value of the queue size at the first computer. Hence, or otherwise, write down the expected queue size at the second computer. Comment briefly on your answer.
(5 marks)

(vi) Consider the case when there is only one packet in the whole system ($K = 1$). According to the answer to the previous part, the expected queue at the first computer is $1/2$. However, there

4. (i) Describe Dijkstra's algorithm for finding shortest paths in a weighted digraph $G = (\mathcal{N}, \mathcal{A})$ with arc weights w_{ij} for each arc $(i, j) \in \mathcal{A}$ and with the notational convenience that $w_{ij} = \infty$ if $(i, j) \notin \mathcal{A}$.
(10 marks)

(ii) Prove that Dijkstra's algorithm does find a shortest path.
(10 marks)

(iii) Consider Dijkstra's algorithm for the path from O to D in the network pictured below. Complete the following table where the figure in brackets is the cost to the node.

Permanent Nodes	Temporary Nodes
O (0)	2 (3), 5(5)
O(0), 2(3)	5(5),1(11),3(17)

Table 1: Nodes for Dijkstra's Algorithm

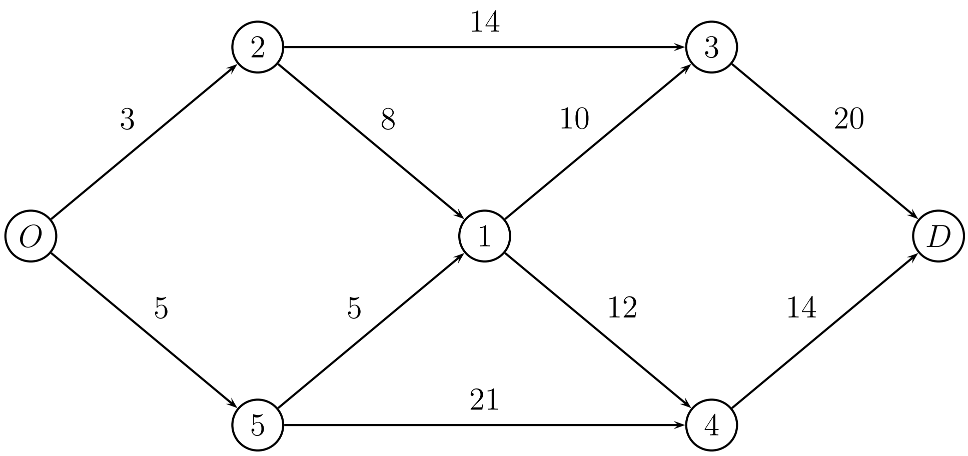


Figure 1: Weighted graph for Dijkstra's algorithm

1. This question is mostly standard book work and should present no problem for students who have revised well. The final part is new but trivial.

(i) (One mark each for the four items in the question and one point for the diagram itself).

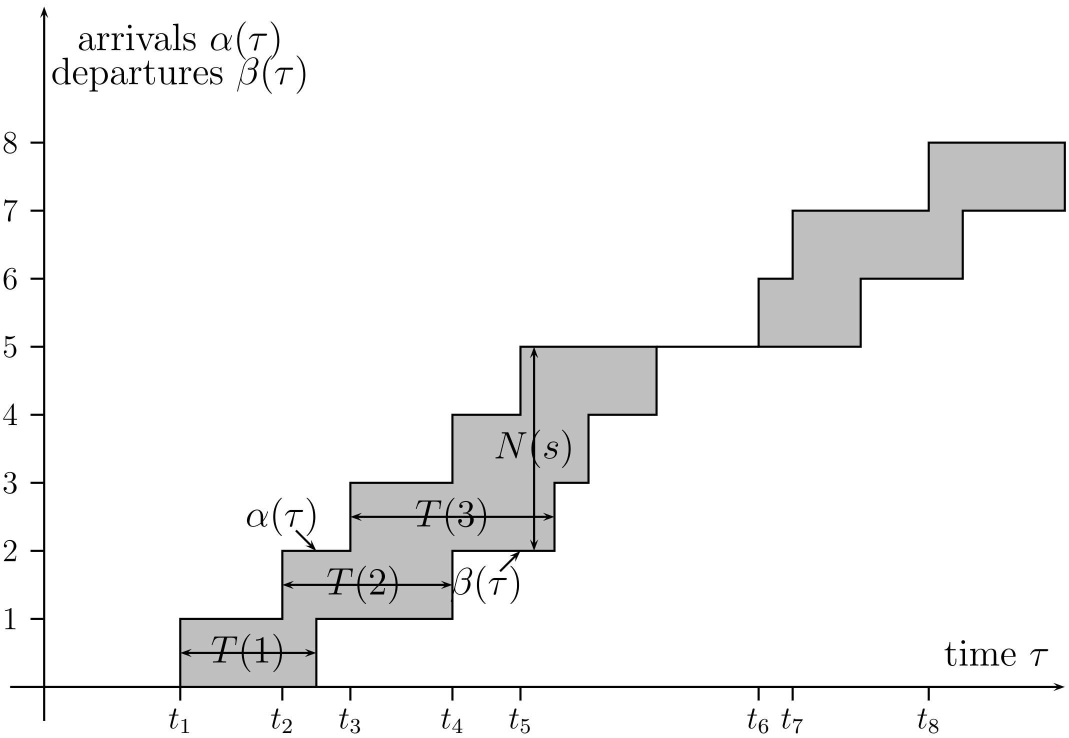


Figure 2: Little's Theorem in a FIFO System

(ii) Choose t is some time such that $N(t) = 0$.
Let N_t be the mean number of customers in the system in the period $[0, t]$. Therefore

$$N_t = \int_0^t \frac{1}{t} N(\tau) d\tau.$$

Let λ_t be the mean number of arrivals in the system in the period $[0, t]$. Therefore,

$$\lambda_t = \alpha(t)/t.$$

Let T_t be the mean time in the system experienced by all the customers who have arrived up to time t (since $N(t) = 0$ these customers have also left). Therefore,

$$T_t = \sum_{i=1}^{\alpha(t)} \frac{T(i)}{\alpha(t)}.$$

Note that this is only defined when at least one customer has entered the system.

Assumptions are:

(a) $N(0) = 0$. (stated in question)

(b) $\forall \tau, \exists t > \tau : N(t) = 0$. (stated in question)

(c) $\lim_{t\to\infty} N_t = N$ exists.

(d) $\lim_{t\to\infty} \lambda_t = \lambda$ exists.

(e) $\lim_{t\to\infty} T_t = T$ exists.

Let $A(t)$ be the area between the curves $\alpha(\tau)$ and $\beta(\tau)$ up to time t.

$$A(t) = \int_0^t \alpha(\tau) - \beta(\tau) d\tau = \int_0^t N(\tau) d\tau.$$

Also

$$A(t) = \sum_{i=1}^{\alpha(t)} T(i).$$

Dividing these by $1/t$ and setting them equal gives,

$$\int_0^t \frac{1}{t} N(\tau) d\tau = \sum_{i=1}^{\alpha(t)} \frac{T(i)}{t} = \sum_{i=1}^{\alpha(t)} \frac{T(i)}{\alpha(t)} \frac{\alpha(t)}{t}.$$

Therefore

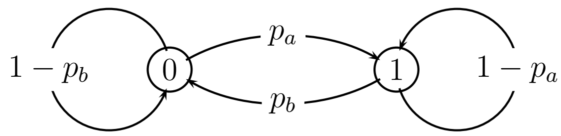
$$N_t = T_t \lambda_t.$$

Taking the limit as $t \to \infty$ (which exists for all quantities by hypothesis) gives the theorem.

(iii) Consider the whole system (users and web site as a queue). There are N people in the queue and it takes each of them $R + P$ seconds to be served (at which point they rejoin the queue). Therefore from Little's Theorem, $\lambda = N/(R+P)$ and, $N_m = \lambda_m(R + P)$.

2. (i) The four layers are: Application (software running the internet), Transport (layer provides reliabily and mediates end-to-end connections), Internet (layer provides basic ability to get packets from a source to a destination) and Host-to-network (layer provides basic connectivity between logically connected computers).

(ii) The situation can be represented by the Markov chain below.



(iii) The standard method of finding eigenvalues gives $|P-I\lambda| = 0$. Therefore, $(3/5-\lambda)(1/3-\lambda)-4/15 = 0$ and, in turn, the two solutions are $\lambda_1 = 1$ and $\lambda_2 = -1/15$.

(iv) The general equation $p_{00}^{(n)} = A+B(-1/15)^n$. From inspection, $p_{00}^{(0)} = 1$ and $p_{00}^{(1)} = 3/5$. Substituting gives $A+B = 1$ and $A - 1/15B = 3/5$. Solving the simultaneous equations gives $A = 5/8$ and $B = 3/8$. Thefore, $p_{00}^{(n)} = 5/8 + 3/8(-1/15)^n$.

This implies that as $n \to \infty$ the probability of the system being empty is $5/8$.

(v) The two balance equations are

$$\pi_0 = 3/5\pi_0 + 2/3\pi_1,$$

and

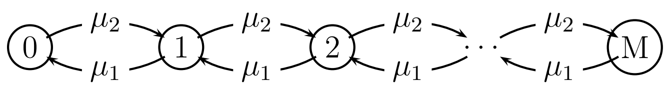
$$\pi_1 = 2/5\pi_0 + 1/3\pi_1,$$

plus the probabilities must sum to one,

$$\pi_0 + \pi_1 = 1.$$

Of course (as expected) the first two are dependent. Solving these in the usual way gives $\pi_0 = 5/8$ and $\pi_1 = 3/8$.

3. (i) The process is represented by the Markov chain shown below (note that the transition from a node to itself is omitted in this diagram though the students may draw this in as well). Note that students who have any intelligence whatsoever can just "copy" this from the transition matrix given in the next part even if they have no insight into the system.



(ii) The queue is equivalent to an M/M/1/K queue.

(iii) The balance equations are

$$\begin{aligned}\pi_0 &= (1-\mu_2)\pi_0 + \mu_1\pi_1,\\ \pi_i &= \mu_2\pi_{i-1} + (1-\mu_1-\mu_2)\pi_i + \mu_1\pi_{i+1},\\ \pi_K &= \mu_2\pi_{K-1} + (1-\mu_1)\pi_K.\end{aligned}$$

Also $\sum_{i=0}^{K}\pi_i = 1$.

(iv) First show by recursion that $\pi_{i+1} = (\mu_2/\mu_1)\pi_i$ for $0 \le i < K$. This is trivially true for $i = 0$ from the first balance equation. Assuming that $\pi_i = (\mu_2/\mu_1)\pi_{i-1}$ then from the second balance equation,

$$\begin{aligned}\pi_i &= \mu_2\pi_{i-1} + (1-\mu_1-\mu_2)\pi_i + \mu_1\pi_{i+1},\\ \pi_i &= \mu_2(\mu_1/\mu_2)\pi_i + (1-\mu_1-\mu_2)\pi_i + \mu_1\pi_{i+1},\\ \mu_2\pi_i &= \mu_1\pi_{i+1},\\ \pi_{i+1} &= (\mu_2/\mu_1)\pi_i,\end{aligned}$$

as required. Hence, $\pi_k = \pi_0(\mu_2/\mu_1)^k$ for $0 \le k \le K$. Now, since

$$\sum_{i=0}^{K} pi_i = 1,$$

then

$$
\begin{aligned}
\sum_{i=0}^{K} \pi_0 (\mu_2/\mu_1)^i &= 1 \\
\sum_{i=0}^{K} (\mu_2/\mu_1)^i &= \frac{1}{\pi_0} \\
\pi_0 &= 1/\sum_{i=0}^{K} (\mu_2/\mu_1)^i \\
\pi_k &= \frac{(\mu_2/\mu_1)^k}{\sum_{i=0}^{K} (\mu_2/\mu_1)^i},
\end{aligned}
$$

which is the required answer

(v) If $\mu_1 = \mu_2$ then the previous equation simply becomes

$$\pi_i = 1^i / \sum_{i=0}^{K} 1^i = 1/(K+1).$$

Let Q_1 be the size of the queue. The expected queue size N is therefore given by

$$N = \mathrm{E}[Q_1] = \sum_{i=0}^{K} i \mathbb{P}[Q_1 = i] = \sum_{i=1}^{K} i/(K+1) = K/2.$$

The expected queue size at the second computer is therefore also $K/2$ since there are always K packets queuing. If the service rates are equal then there is no reason why more packets should be being processed by one computer or the other. Therefore, it is no surprise to find that the mean queue size is simply half the packets. Indeed, it would have sufficed to observe this to get all five marks (a quick solution for brighter students).

(vi) The packet travels instantaneously between computers. At the instant the packet is in transit, there is no queue at either computer. However, because this is an instant, it has not contributed to the time average of the queue size. Therefore, the queue size at any given computer is, at any given moment, on average 1/2 even though the packet will never arrive at the computer and see a queue.

4. (i) The algorithm finds the shortest path to all nodes from an origin node, on a graph $G = (\mathcal{N}, \mathcal{A})$. It requires that all arc weights are non-negative $\forall(i,j) \in \mathcal{A} : w_{ij} \geq 0$.

Dijkstra's Algorithm involves the labelling of a set of permanent nodes P and node distances D_j to each node $j \in \mathcal{N}$. Assume that we wish to find the shortest paths from node 1 to all nodes. Then we begin with: $P = \{1\}$ and $D_1 = 0$ and $D_j = w_1 j$ where $j \neq 1$. Dijkstra's algorithm then consists of following the procedure:

(a) Find the next closest node. Find $i \notin P$ such that:

$$D_i = \min_{j \notin P} D_j$$

(b) Update our set of permanently labelled nodes and our nodes distances:
$P := P \cup \{i\}$.

(c) If all nodes in $\mathcal{N}$ are also in P then we have finished so stop here.

(d) Update the temporary (distance) labels for the new node i. For all $j \notin P$

$$D_j := \min[D_j, w_{ij} + D_i]$$

(e) Go to the beginning of the algorithm.

(ii) At the beginning of each iteration of Dijkstra's algorithm then:

(a) $D_i \leq D_j$ for all $i \in P$ and $j \notin P$.

(b) D_j is, for all j, the shortest distance from 1 to j using paths with all nodes (except, possibly j) in P.

If this proposition can be proved then we can see that, when P contains every node in $\mathcal{N}$ then all the D_j are shortest paths by the second part of this proposition. Therefore proving the above proposition is equivalent to proving that Dijkstra's algorithm finds shortest paths.

The proposition is trivially true at the first step since P consists only of the origin point (node 1) and D_j is 0 for $j = 1$, is $w_{ij} \geq 0$ for nodes reachable directly from node 1 and ∞ otherwise.

The first condition is simply shown to be satisfied since it is preserved by the formula:

$$D_j := \min[D_j, w_{ij} + D_i]$$

which is applied to all $j \notin P$ when node i is added to the set P.

We show the second condition by induction. We have established already that it is true at the very start of the algorithm. Let us assume it is true for the beginning of some iteration of the algorithm and show that it must then be true at the beginning of the next iteration.

Let node i be the node we are adding to our set P and let D_k be the label of each node k at the beginning of the step. The second condition must, therefore, hold for node $j = i$ (the new node we have added) by our induction hypothesis. It must also hold for all nodes $j \in P$ by part one of the proposition which is already proven. It remains to prove that the second condition of the proposition is met for $j \notin P \cup \{i\}$.

Consider a path from 1 to j which is shortest amongst those with all nodes except j in $P \cup \{i\}$ and let D'_j be the corresponding shortest distance. Such a path must contain a path from 1 to some node $r \in P \cup \{i\}$ and an arc(r, j). We have already established that the length of the path from 1 to r must be D_r and therefore we have:

$$D'_j = \min_{r \in P \cup \{i\}} [D_r + w_{rj}] = \min[\min_{r \in P} [D_r + w_{rj}], D_i + w_{ij}]$$

However, by our hypothesis $D_j = \min_{r \in P}[D_r + w_{rj}]$ therefore, $D'_j = min[D_j, D_i + w_{ij}]$ which is exactly what is set by the fourth step of the algorithm. Thus, after any iteration of the algorithm, the second part of the propostion is true if it was true at the beginning of the iteration. Thus the proof by induction is complete.

(iii) The table shown below answers the question

The final route is $O \to 5 \to 1 \to 4 \to D$ with a cost of 36.

Permanent Nodes	Temporary Nodes
O (0)	2 (3), 5(5)
O(0), 2(3)	5(5),1(11),3(17)
O(0), 2(3), 5(5)	1(10),3(17), 4(26)
O(0), 2(3), 5(5), 1(10)	3(17), 4(22)
O(0), 2(3), 5(5), 1(10), 3(17)	4(22), D(37)
O(0), 2(3), 5(5), 1(10), 3(17), 4(22)	D(36)

Table 2: Nodes for Dijkstra's Algorithm