[Lang et al 82] T. Lang, M. Valero, I. Alegre, "Bandwidth of Crossbar and Multi-bus Connections for Multiprocessors", IEEE Transactions on Computers, Vol C-31, December 1982, pp. 1227-1234.

[Mitrani 87] I. Mitrani, "Modelling of Computer and Communication Systems", Cambridge University Press, 1987.

[Patel 81] J. H. Patel, "Performance of Processor-Memory Interconnections for Multiprocessors", *IEEE Transactions on Computers*, **C-30**, pp. 771-780, Oct. 1981.

[Reed and Grunwald 87] D.A. Reed, D.C.Grunwald, "The Performance of Multicomputer Interconnection Networks", *IEEE Computer*, June 1987, pp. 67-73.

time distribution is found to have a *shorter* tail than that of FCFS, which is the opposite of the open case considered here.

It is interesting to note that in a number of multiprocessor systems, particularly those incorporating large numbers of processors, the communication network is usually comprised of an array of crossbar switches arranged in mutiple stages, see for example [Adams et al 87]. Transmission times under FCFS have been calculated for buffered versions of these networks [Kruskal et al 86,Harrison 91] and it will be interesting to see whether the techniques presented here could be similarly applied to multistage network configurations.

References

[Adams et al 87] G. Adams, D.P.Agrawal, H.J. Siegel, "A Survey and Comparison of Multi-stage Interconnection Networks", *IEEE Computer*, June 1987, pp. 14-27.

[Coffman et al 70] E.G. Coffman, R.R. Muntz and H. Trotter. "Waiting Time Distributions for Processor Sharing Systems". *JACM* 17(1), pp. 123-130.

[Dias and Jump 81] D.M. Dias, J.R. Jump, "Analysis and Simulation of Buffered Delta Networks", *IEEE Trans on Computers*, Vol. C-30, No. 4, April 1981, pp. 273-282.

[Dubner and Abate 68] H. Dubner and J. Abate. "Numerical Inversion of Laplace Transforms by Relating them to the Finite Fourier Cosine Transform". *JACM* 15(10), pp. 115-123.

[Field and Cripps 85] A.J. Field and M.D. Cripps. "Self-clocking Networks". *Proc. 1985 International Conference on Parallel Processing*, pp. 384-387.

[Field and Harrison 92] A.J. Field and P.G. Harrison. "Transmission Times in Unbuffered Crossbars with Cyclic Arbitration". *Proc. 1992 International Conference on Parallel Processing*, pp. 132-137 (Vol. 1).

[Harrison and Patel 87] P.G. Harrison, N.M. Patel, "The Representation of Switching Networks in Queueing Models of Parallel Systems". *Proceedings of the 12th International Symposium on Computer Performance*, Editors P.-J. Courtois and G. Latouche, Brussels, December 1987, pp.497-512.

[Harrison 91] P.G. Harrison. "On Non-uniform Packet Switched Delta Networks and the Hot-spot Effect". *IEE Proceedings*, Vol. 138, No. 3, May 1991, pp. 123-130.

[IEEE 85a] IEEE. "IEEE 802-4 Token Bus Access Method". *IEEE*, 1985.

[IEEE 85b] IEEE. "IEEE 802-5 Token Passing Ring Access Method". *IEEE*, 1985.

[Kelly 86] F.P. Kelly, "Blocking Probabilities in Large Circuit-switched Networks", *Advances in Applied Probability*, 18, pp. 473-505 (1986).

[Kruskal et al 86] C.P. Kruskal, M. Snir and A. Weiss. "The Distribution of Waiting Times in Clocked Multistage Interconnection Networks". *Proc. 1986 International Conference on Parallel Processing*, pp. 12-19.

In the case of exponentially distributed service times, with parameter μ, the equation for $M^*(\theta)$ yields $M^*(\theta)$ as the smaller root of the quadratic in x:

$$x = \frac{\mu}{\mu + \theta + \lambda(1-x)}$$

so that $M^*(\theta)^{-1}$ is the larger root of the equation

$$\mu z^2 - (\lambda + \mu + \theta)z + \lambda = 0$$

i.e. $M^*(\theta) = r'^{-1}$. Thus, since in the exponential case, $S^*(\theta) = \frac{\mu}{\mu+\theta} = L_0(\theta)$, we have that $L = T^*$ with L as defined in equation 2. The models are therefore consistent in this case.

6 Summary and Conclusions

In this paper we have derived the Laplace transform of the transmission time density function of a buffered full crossbar communication network operating a cyclic arbitration scheme. This differs from the FCFS scheme firstly in that the (FIFO) message buffers are located at the network inputs, and secondly that the messages contained in the buffers are serviced using a variation of the 'round robin' arbitration scheme, rather than one based on their order of arrival. In our discussion we used a circulating token as an analogy for the way the hardware in these systems actually works, and, indeed, a similar analysis could be used to obtain transmission times in both bus- and ring-based systems operating the token passing access methods [IEEE 85a,IEEE 85b].

The cyclic scheme has the property that arriving messages can sometimes jump the queue and obtain service ahead of messages already queued in the system. We find that in this open model, the transmission time density function for the cyclic scheme is found to have a slightly longer tail than the equivalent FCFS scheme, although the mean transmission time for both is the same.

By validating the model with a simulation model, we find that it is accurate for light to moderate loads, but that the accuracy deteriorates as $\rho \to 1$. This is attributable to our approximating assumption that an arriving message jumps ahead of a given monitored message with a probability p which is independent of the current state of the queue. The longer the queue length on arrival, and the nearer the monitored message is to the head of the queue, so the jump probability *should* decrease in order to preserve an accurate representation of the scheme. A general analysis would therefore benefit from p being state dependent, requiring at least the solution of a differential equation for $G(z)$.

For the fixed p assumption, we noted that an alternative approach is to model the system as a LCFS queue with an arrival rate equal to λp. This enables us to handle general service time distributions at the expense of producing a less flexible solution method: the approach taken under the assumption of exponential service times is more easily extended to state-dependent jump probabilities than the adapted LCFS model. The general service time model has been validated by showing that if the service times are indeed exponential then the solution is the same as that already derived.

A related area of interest is the analysis of unbuffered crossbar networks. In these systems, messages arriving at the network must wait for their requested output to become available before a second message can be accepted from the same transmitter. A model of this system is described in [Field and Harrison 92], under the assumption that the service times are exponentially distributed. Since the message population is fixed, this approach uses a closed queueing model and expresses the tagged customer's delay as the sum of an integral number of identical exponentially distributed service times. The transmission time density function is then found to be a suitably weighted sum of Erlang-k densities. Curiously, in this situation the transmission

Thus,

$$E[\, e^{-\theta Q} \mid R, A\,] \;=\; e^{-\theta R} E[\, e^{-\theta(S_1+\ldots+S_{A-1})} E[\, e^{-\theta(M_0-R)} \mid R\,] \prod_{i=1}^{A-1} E[\, e^{-\theta(M_i-S_i)} \mid S_i\,] \mid R, A\,]$$

since the M_i are mutually independent and independent of A, M_i depends only on S_i $(i > 0)$ and M_0 depends only on R. Thus we obtain, for $i > 0$,

$$\begin{aligned} E[\, e^{-\theta(M_i-S_i)} \mid S_i\,] &= E[\, E[\, e^{-\theta(M_{i1}+\ldots+M_{iZ}} \mid Z, S_i\,] \mid S_i\,] \\ &= E[\, M^*(\theta)^Z \mid S_i\,] \end{aligned}$$

since the M_{ij} are independent of S_i

$$= \; e^{-\lambda S_i(1-M^*(\theta))}$$

since, given S_i, Z is Poisson with parameter λS_i. Similarly, $E[\, e^{-\theta(M_0-S_0)} \mid R\,] = e^{-\lambda R(1-M^*(\theta))}$. Thus we obtain

$$E[\, e^{-\theta Q} \mid R, A\,] \;=\; e^{-(\theta+\lambda(1-M^*(\theta)))R} E\left[\, \prod_{i=1}^{A-1} e^{-(\theta+\lambda(1-M^*(\theta)))S_i} \right]$$

and so, for $A > 0$,

$$E[\, e^{-\theta Q} \mid A\,] \;=\; R^*(\theta+\lambda(1-M^*(\theta)))S^*(\theta+\lambda(1-M^*(\theta)))^{A-1}$$

We obtain $M^*(\theta))$ by deconditioning the equation

$$E[\, e^{-\theta(M-S)} \mid S\,] = e^{-\lambda S(1-M^*(\theta))}$$

after multiplying by $e^{-\theta S}$ to get

$$\begin{aligned} M^*(\theta) = E[\, E[\, e^{-\theta M} \mid S\,]\,] &= E[\, e^{-S(\theta+\lambda(1-M^*(\theta)))} \\ &= S^*(\theta+\lambda(1-M^*(\theta))) \end{aligned}$$

Finally, this yields the result

$$T^*(\theta) \;=\; S^*(\theta)\left\{ P(A=0) + \frac{R^*(\theta+\lambda(1-M^*(\theta)))}{M^*(\theta)} [G_A(M^*(\theta)) - P(A=0)\,] \right\}$$

For the case of exponential service times, we can recover the solution obtained before by noting that then $R^* = S^*$ so that $T^*(\theta) = S^*(\theta) G_A(M^*(\theta))$. The generating function $G_A(z)$ can be determined as follows:

$$\begin{aligned} G_A(z) &= \sum_{a=0}^{\infty} z^a P(A=a) \\ &= (1-\rho)\left\{ 1 + \sum_{a=1}^{\infty} z^a \sum_{q=a}^{\infty} \frac{\rho^q}{q} \right\} \\ &= (1-\rho)\left\{ 1 + \sum_{q=1}^{\infty} \frac{\rho^q}{q} \sum_{a=1}^{q} z^a \right\} \\ &= (1-\rho)\left\{ 1 + \frac{z}{1-z} \sum_{q=1}^{\infty} \left[\frac{\rho^q}{q} - \frac{(\rho z)^q}{q} \right] \right\} \\ &= (1-\rho)\left\{ 1 + \frac{1}{z^{-1}-1} \log_e \frac{z^{-1}-\rho}{z^{-1}(1-\rho)} \right\} \end{aligned}$$

In this section we solve for the response time distribution of the tagged customer in this system for *arbitrary* service time distributions. We define the following random variables:

S, S_i	Service time random variable.
R	Residual service time random variable.
A	Number of customers in the queue immediately before the arrival instant of the tagged customer. For $A > 0$, the customer in service is numbered 0 and the others in first come first served order.
Z	Number of arrivals during a full service period.
Z'	Number of arrivals during a residual service period.
M_i	Sum of the service periods arising from the service period of customer i, i.e. comprising that of customer i, those of the arrivals during customer i's service period, those of the arrivals during their service periods and so on until there are no more arrivals. Thus the M_i are all independent and, for $i > 0$, identically distributed—customer 1 has a residual service time rather than a full one.
M, M_{ij}	Generic random variables distributed as M_i for $i > 0$.
T	Response time, i.e. the time elapsed between the tagged customer's arrival and completion of service.
Q	Queueing time, i.e. the time elapsed between the tagged customer's arrival and beginning of service, so that $T = Q + S$. Thus $Q = M_0 + ... + M_{A-1}$.

We denote the probability generating function of a discrete random variable X by G_X and the Laplace transfom of the distribution of a continuous random variable Y by Y^*. Thus:

$$\begin{aligned} Q^*(\theta) &= E[\, e^{-\theta Q} \,] \\ &= E[\, E[\, e^{-\theta Q} \mid A \,] \,] \end{aligned}$$

where

$$\begin{aligned} E[\, e^{-\theta Q} \mid A \,] &= 1 && \text{for } A = 0 \\ &= E[\, E[\, e^{-\theta Q} \mid R, A \,] \mid A \,] && \text{for } A > 0 \end{aligned}$$

Now, labeling the service time of customer i in the queue at the arrival instant by S_i, we have

$$E[\, e^{-\theta Q} \mid R, A \,] = E[\, E[\, e^{-\theta(M_0+...+M_{A-1})} \mid R, A, S_1, ..., S_{A-1} \,] \mid R, A \,]$$

and

$$\begin{aligned} M_0 &= R + M_{01} + ... + M_{0Z'} \\ M_i &= S_i + M_{i1} + ... + M_{iZ} && \text{for } i > 0 \end{aligned}$$

in some way, possibly requiring additional conditioning parameters to that of n in $F_n(t)$. This is discussed further in Section 6 below.

A further comparison of some relevance is that between the cyclic and FCFS models under light to moderate loading. The original network with FCFS arbitration can be approximated by an open queueing network in which the FIFO output buffers are modelled by unbounded FCFS queues. The decomposition of the model into x independent single-server queues is then the same as for the cyclic case except that the queueing discpline at each queue is FCFS. The Laplace Transform of the transmission time density function in this case is well known, and has been derived earlier from $G(z)$ by setting $p = 0$:-

$$L(s) = \frac{\mu - \lambda}{s + \mu - \lambda}$$

Figure 5 shows the derived density functions of the cyclic and FCFS queuing disciplines for $\lambda = 1$ and $\mu = 2$. The cyclic scheme promotes some messages in the queue at the expense of others when compared to FCFS and for this reason relatively more of the distribution lies in the tail. For example, approximately 0.9% of messages have transmission times longer than 5.0 in the FCFS scheme, compared to approximately 1.4% for the cyclic scheme. The net effect this has on the transmission time density is not so pronounced, however, and we may reasonably conclude that the cyclic scheme represents a viable and 'well behaved' alternative to FCFS in this respect.

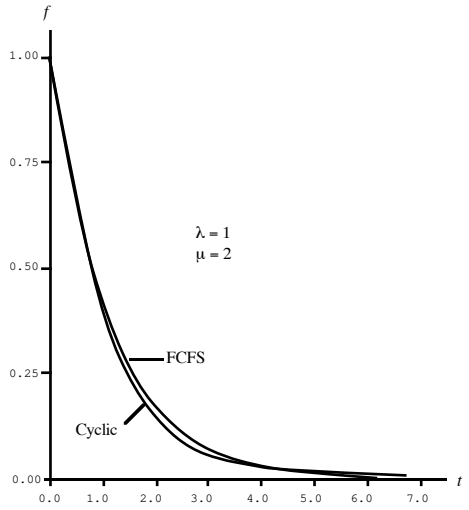

Figure 5: Comparison with FCFS $\lambda = 1, \mu = 2$

5 General Message Lengths

We now consider an alternative formulation of the problem and relax our previous assumption of exponential service times. We observe that under the assumption that the jump probability if fixed, the queueing system we have considered can be regarded in the following more general form. A customer arrives at a queue in steady state at a random position in the queue. Susequent arrivals form two independent Poisson streams with rates $(1 - p)\lambda$ of lower priority which can be ignored since the tagged customer will be served first and with rate $p\lambda$ of higher priority. In this section we write λ instead of $p\lambda$ to simplify the notation. The customers ahead of the tagged customer at its arrival instant also have higher priority, i.e. will be served before the tagged customer.

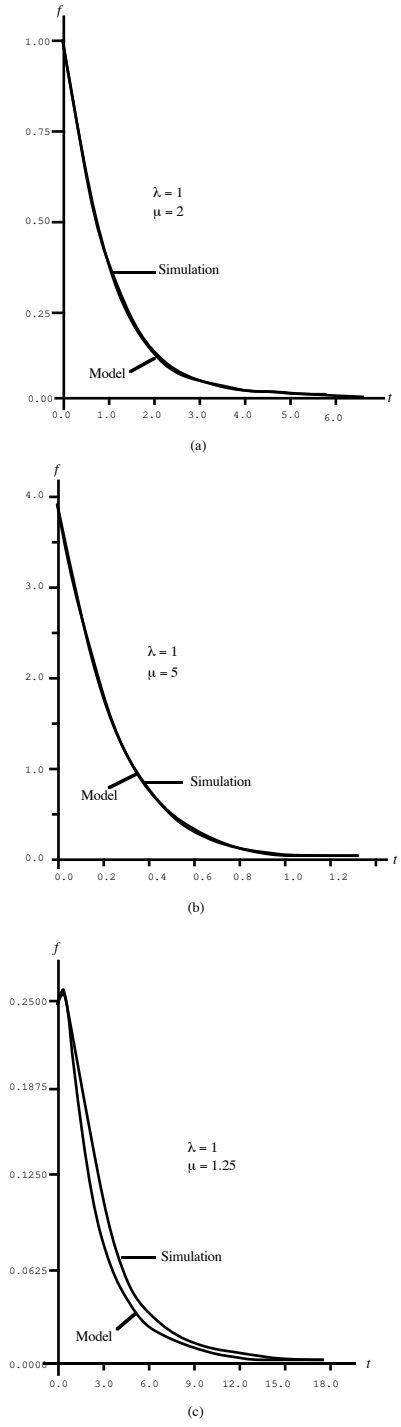

Figure 4: Comparison with Simulation for $\rho = 0.2, 0.5, 0.8$

3.1 On the Calculation of p

In the above analysis, we have assumed that an arriving task is serviced ahead of the monitored task with a fixed probability $p = (x-1)/2x$.

For lightly loaded systems, we would expect the approximation to be reasonably accurate: Consider the situation in which the monitored task, M say, is the only task queued for the (busy) output—a common siuation under light loading. Now consider a new arrival, A. For both M and A, the probability that they originate from input i is $1/x$ for all $1 \leq i \leq x$ by virtue of the random arrival process. Furthermore, if A originates from the same input as M, it will certainly be serviced after M, because the input queues are FCFS. Therefore, summing over all other possible inputs we obtain:

$$\sum_{n=1}^{x-1} \frac{1}{x} \left(\frac{n}{x}\right) = \frac{x-1}{2x}$$

which is the same as the intuitive expression for p we gave earlier.

In a heavily loaded system, the approximation will be good for the early part of the monitored tasks lifetime in the system, since by slicing off equal numbers of customers from each queue (assumed in equilibrium) we have a similar situation to that of the lightly loaded system. However, as the task nears the front of its input queue, and as further arrivals queue up behind it and in other queues, so new arrivals *should* be progressively less likely to be serviced ahead of it, hence the less precise the fixed p assumption becomes. We would therefore expect the model's accuracy to deteriorate as $\lambda/\mu \rightarrow 1$, and indeed this can be seen in the density functions derived below. The results will, however, be pessimistic in that some messages will have measured transmission times which are higher than would be expected in practice. Therefore, if we focus on the tails of the resulting distributions under heavy loads we will at least obtain safe estimates of the relative frequency of long transmission times, which is often a measure of some concern.

4 Analysis

The analytical inversion of the Laplace Transform of a function is possible in some cases, but complex in general, and in the absence of any obvious method for the performing the inversion of $L(s)$ analytically we appeal to a numerical method. To obtain the required transmission time density function we apply the procedure of [Dubner and Abate 68] to $L(s)$.

In order to validate these transmission times and so assess their accuracy under different loads, the density functions derived were compared with those from a simulation model of the network. The results of these comparisons are shown in Figure 4 for $x = 4$. (The value of x was chosen simply to shorten the simulation times; similar curves are easily obtainable for $x > 4$.)

For light to moderate loads (Figures 4(a) and (b)) the derived and simulation-generated curves are almost indistinguishable. As the load increases beyond $\lambda/\mu = 0.5$, the number of arrivals experienced by the monitored task during its lifetime in the system increases accordingly, and this leads to a higher number of incoming tasks jumping the queue ahead of the monitored task. It is under these conditions that the accuracy of the model gradually deteriorates, as is illustrated by the graph of Figure 4(c) which compares the densities for $\rho = 0.8$.

We find that the analytical and simulation models correspond closely for values of $\rho \leq 0.6$ approximately, but that the separation of the curves becomes visibly evident thereafter. Further experimentation with the value of p under heavy load also reveals that the curves become further coincident as p is lowered, as we would expect from the discussion above. However, the discrepency is still apparent even for very small values of p suggesting that even a rate-dependent (but still fixed) value of p is not sufficient to abtain accurate densitites under heavy loads. It would appear that the model can only be improved in these cases by making p state dependent

by the law of total probability, where Q is the equilibrium queue length random variable. A new arrival can arrive on *any* input to the crossbar and so we make the approximation that $P(\, N = n \mid Q = q \,) = 1/q \; (q \leq n)$. For small to moderate values of Q (which occur with higher probability under light to medium loads) the approximation is good since the inputs will be relatively lightly populated with messages for the output in question, so that in the model, a new arrival will be roughly equally likely to jump into any position in the queue. Under heavy loads, the range of N and Q will be higher: although the equal probability assumption is less accurate in this case, $1/q$ will at least be small for large q, so the probability of N being small when q is large will vanish in the limit, as we would want. Hence,

$$\begin{aligned}
L(s) &= \sum_{n=1}^{\infty}\sum_{q=n}^{\infty}(1-\rho)\frac{\rho^q}{q}L_n(s) \; + \; (1-\rho)\, L_0(s) \\
&= (1-\rho)\sum_{n=1}^{\infty}\int_0^{\rho} L_n(s)\sum_{q=n}^{\infty} z^{q-1}dz \; + \; (1-\rho)L_0(s) \\
&= (1-\rho)\int_0^{\rho}\sum_{n=1}^{\infty} L_n(s)\frac{z^{n-1}}{1-z}dz \; + \; (1-\rho)L_0(s) \\
&= (1-\rho)\int_0^{\rho}\frac{G(z)-L_0}{z(1-z)}dz \; + \; (1-\rho)L_0(s)
\end{aligned}$$

(Note that we are able to interchange the summation and integral because the series is absolutely convergent at $|z| < 1$.) We now simplify the integration term:-

$$\begin{aligned}
\frac{G(z)-L_0(s)}{z} &= \frac{p\lambda L_1(s) - \mu z L_0(s)}{\mu(z-r)(z-r')} \\
&= \frac{(r-z)\mu L_0(s)}{\mu(z-r)(z-r')} \\
&= \frac{L_0(s)}{r'-z}
\end{aligned}$$

The above integral is therefore

$$\begin{aligned}
&\frac{L_0(s)}{r'-1}\int_0^{\rho}\left\{\frac{1}{1-z}-\frac{1}{r'-z}\right\}dz \\
&\qquad = \frac{L_0(s)}{r'-1}\left[\log_e\frac{r'-z}{1-z}\right]_0^{\rho}
\end{aligned}$$

from which we obtain

$$L(s) = (1-\rho)L_0(s)\left\{1+\frac{1}{r'-1}log_e\frac{r'-\rho}{r'(1-\rho)}\right\} \tag{2}$$

where

$$\begin{aligned}
L_0(s) &= \frac{\mu}{(s+\mu)} \\
r' &= \frac{x+\sqrt{x^2-4p\mu\lambda}}{2\mu}
\end{aligned}$$

and where $x = s + \mu + p\lambda$.

This implies, as we already know, that $F_0(t) = 1 - e^{-\mu t}$. Now let $f_n(t)$ have Laplace Transform $L_n(s)$. Then, by a simple integration by parts, we can establish that $f_n'(t)$ has Laplace Transform $sL_n(s) + f_n(0)$. Moreover, for $n \geq 1$, the response time is a sum of two or more exponential random variables with parameter μ. Hence its distribution is a mixture of Erlang-k distributions with $k \geq 2$. But, the density function of an Erlang-k random variable vanishes at the origin and so $f_n(0) = 0$ for $n \geq 1$. We therefore have (after differentiating equation 1 above)

$$(s + p\lambda + \mu)L_n(s) = p\lambda L_{n+1}(s) + \mu L_{n-1}(s) \qquad (n \geq 1)$$

with $L_0(s) = \mu/(s + \mu)$. To solve this recurrence, let $G(z) = \sum_{n=0}^{\infty} L_n(s)z^n$. Multiplying the recurrence throughout by z^n and summing over $n \geq 1$ now yields

$$(s + p\lambda + \mu)\,[G(z) - L_0(s)\,] = p\lambda z^{-1}\,[G(z) - L_0(s) - zL_1(s)\,] + \mu z G(z)$$

Thus,

$$G(z) = \frac{(-(s + p\lambda + \mu)z + p\lambda)L_0(s) + p\lambda L_1(s)}{\mu z^2 - (s + p\lambda + \mu)z + p\lambda}$$

Now, we know $L_0(s)$, but must find one more equation in order to determine the value of $L_1(s)$ and hence $G(z)$. We therefore use the analyticity of $G(z)$ inside the unit disc: the denominator above vanishes at $z = r$ and $z = r'$, where r and r' are the roots of the quadratic equation $Q(x) = 0$, with

$$Q(x) = \mu x^2 - (s + p\lambda + \mu)x + p\lambda$$

Without loss of generality, let $r \leq r'$. Now, $Q(x)$ is positive as $x \to \infty$, and for all $x \leq 0$. But $Q(1) = -s$ so that $r \leq 1 \leq r'$ and for $s > 0$, $r < 1 < r'$. A necessary condition for the analyticity of $G(z)$ for $|z| < 1$ is therefore that the numerator also vanishes at $z = r$. Hence, using the quadratic equation,

$$-\mu r^2 L_0(s) + p\lambda r L_1(s) = 0$$

so that $p\lambda L_1(s) = \mu r L_0(s)$. This serves to define $L_1(s)$ and hence $G(z)$.

Note, in the special case that p=0 (instead of $(x-1)/2x$), we obtain, since $L_0(s) = \mu/(s+\mu)$,

$$G(z) = \frac{\mu}{s + \mu - \mu z}$$

Thus, in the steady state, the response time density has Laplace Transform

$$L(s) = \sum_{n=0}^{\infty} p_n L_n(s)$$

where p_n is the stationary probability mass function of the queue length, by the random observer property, see for example [Mitrani 87], i.e. $p_n = (1 - \rho)\rho^n$, where $\rho = \lambda/\mu$. Hence,

$$\begin{aligned} L(s) &= (1 - \rho)G(\rho) \\ &= \frac{\mu - \lambda}{s + \mu - \lambda} \end{aligned}$$

as expected, since the queueing discipline is FCFS at $p = 0$. In the general case we have

$$L(s) = \sum_{n=1}^{\infty} P(\,N = n\,)\,L_n(s) \;+\; P(\,N = 0\,)\,L_0(s)$$

But, since a task in service cannot be preempted, $P(\,N = 0\,) = p_0 = 1 - \rho$, and we have

$$L(s) = \sum_{n=1}^{\infty} \sum_{q=n}^{\infty} P(\,Q = q\,)\,P(\,N = n \mid Q = q\,)\,L_n(s) \;+\; (1 - \rho)\,L_0(s)$$

than longer ones, and that the exponential assumption in these cases is actually quite good [Field and Harrison 92]. The need for a more rigorous analysis of network traffic in these systems in the long term is apparent, however. The second approach is rather less flexible than the first in its assumptions about the queue jumping process, but is capable of handling general service time distributions.

3 Response Times with Exponentially Distributed Message Lengths

In keeping with the above discussion we consider an open single-server queue with a Poisson arrival process with rate λ, and negative-exponentially distributed service times with mean $1/\mu$. This restriction is lifted in Section 5 at the expense of a slightly less flexible overall approach. We further identify a special task whose progression through the queue we shall monitor as the basis of our analysis. The general method is similar to that in [Coffman et al 70].

Let T denote the response time random variable of a task, i.e. the sum of the tasks queueing time and service time, and define

$$F_n(t) = P(T \leq t \mid N = n)$$

where the random variable N is the number of tasks in the queue ahead of the monitored task at its arrival instant. Consider now the intervals $[0, h)$ and $[h, t+h)$ where time 0 is the arrival instant of the monitored task. During the initial interval $[0, h)$, as $h \to 0$, there are two possible events: firstly, an arrival with probability $\lambda h + o(h)$, and secondly a service completion with probability μh. Moreover, if an arrival occurs we make the approximationn that the arriving task is serviced before the monitored task with a fixed probability $p = (x-1)/2x$.

A fuller justification of this approximation is deferred until Section 3.1 but it may be argued intuitively as follows: if the new arrival originates from the same input as the monitored task, M say, then it is certain to be be serviced after M because the input queues are FCFS. If it arrives on any other input (with probability $(x-1)/x$), there will be, on average, a 50% chance of it being served ahead of M and a 50% chance of it being served after M—an overall probability of $(x-1)/2x$. The approximation gets worse as $\lambda/\mu \to 1$, but is found to be accurate under light to moderate loading—see Section 3.1 for a more detailed explanation.

By the law of total probability, and using the fact that the forward recurrence time of an exponential random variable is an identical exponential random variable, we have

$$\begin{aligned} F_n(t+h) &= (1 - (\lambda+\mu)\, h)\, F_n(t) \\ &+ p\lambda h F_{n+1}(t) \\ &+ (1-p)\, \lambda h F_n(t) \\ &+ \mu h F_{n-1}(t) \end{aligned}$$

for $n \geq 1$, i.e.

$$f_n(t) \equiv \frac{dF_n(t)}{dt} = -(p\lambda + \mu) F_n(t) + p\lambda F_{n+1}(t) + \mu F_{n-1}(t) \qquad (1)$$

For n=0, the arriving task inevitably enters service immediately and so

$$F_0(t+h) = (1 - \mu h) F_0(t) + \mu h$$

since a new arrival cannot effect the response time and a service completion certainly results in $T \leq t$. Thus

$$f_0(t) = -\mu F_0(t) + \mu$$

It is this property which complicates any intuitive understanding of the switch's behaviour and the primary one which motivates our desire to measure the distribution of the transmission times.

2.2 Representation as an Open Queueing Model

An exact analysis of this system as a queueing network would be intractable since we would have to represent explicitly the state of every input queue in the system. We therefore choose an approximate analysis in which we model the system as though the queues were located at the outputs only. Unlike the system discussed earlier, however, the output queues will not be FCFS since they must take into account the cyclic nature of the arbitration scheme in the physical system.

To obtain the transmission times we follow the progress of a special 'monitored' task (a message) from its arrival at an output queue to its service (transmission) completion. We now note that once the task is queued, a newly arriving task may jump ahead of the monitored task by virtue of the cyclic arbiter. Our approximating assumption in this respect is that a 'jump' of this sort occurs with a probability, p, that remains fixed throughout the task's lifetime in the system. Note that we do not allow preemption since the start of a message transfer corresponds to the originating input holding the required output token. A discussion of the choice of value for p is given in Section 3.1 below.

Our model of the system is therefore an open queueing network comprising x independent and identical Poisson input streams, each with rate λ, and x single-server queueing centres, each with service rate μ (Figure 3). An arriving task is assumed to select one of the x service centres with equal probability (i.e. $1/x$). Because of the branching properties of the Poisson process, this enables us to decompose the system into x independent and identical single-server queueing systems, each with a Poisson arrival process with rate $x(\lambda/x) = \lambda$ and service rate μ. The analysis can then proceed by examining just one single-server queue, but with the unusual 'jump' queueing discipline.

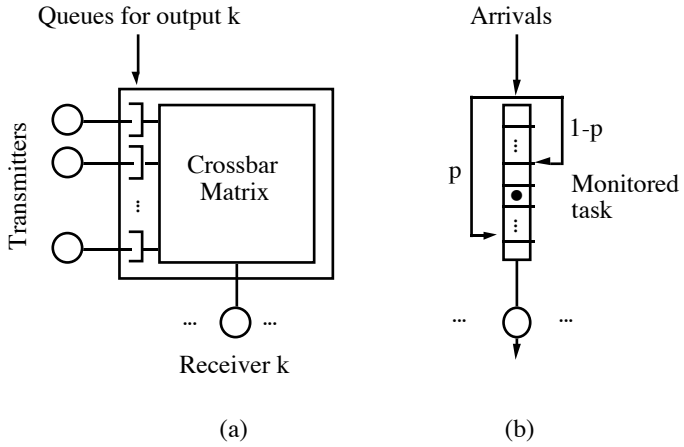

Figure 3: The Queueing Model

In practice, the input buffers will be finite so the open network model (which assumes infinite buffers) is, in itself, an approximation. However, it is well known that for light to moderate loads (which, incidentally we shall be focussing on in due course) moderate sized buffers present a good approximation.

We present two approaches to the anslysis. The first assumes that the message length distribution is (negative) exponential. Although in general this is far from exact, we find that in some architectures at least, short messages do occur with much higher probability

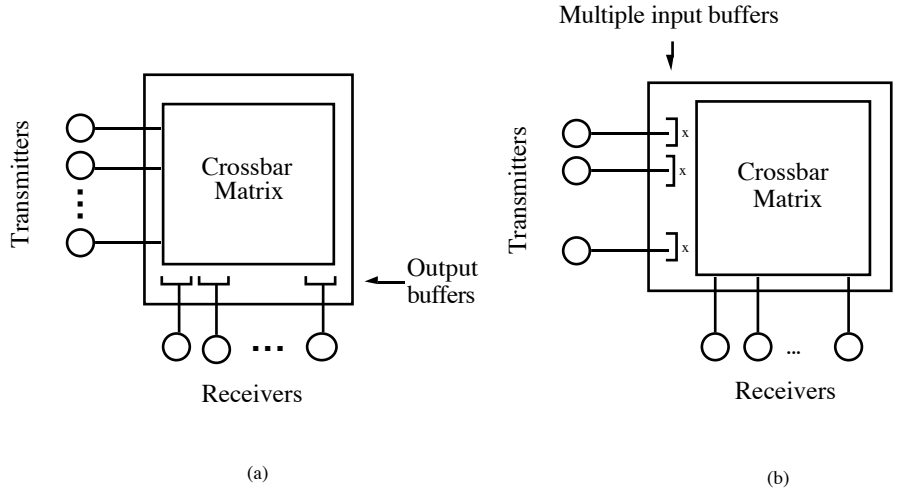

Figure 2: FCFS and Cyclic Buffering

each transmitter) is a token, and that in order to transmit a message to that output, the input must first acquire the token. When there are no messages for an output, the corresponding token circulates around the inputs in the order 1,2,...,x,1,2,... and so on, until such a message arrives. When a message for the output appears at one of the inputs, that input waits for the token to arrive and then holds it for the duration of the transmission. During this time other messages for the same output may be loaded into their respective input buffers. Also, at the same time, messages originating from other inputs and destined for other outputs may be being transmitted.

When the message has been transferred to the receiver the input releases the token which then continues circulating the inputs, even though it may have a second message for that output in its buffer. In theory, an input may hold all x tokens, that is it has the capacity to service all of its input buffers simultaneously. This is in contrast to an unbuffered crossbar where each input can be connected to at most one output at any time.

The 'cyclic' arbitration scheme is starvation free since every transmitter has the opportunity to transmit a message on each cycle of the token. The scheme is also advantageous in that it has a particularly straightforward and efficient hardware implementation using a system of multi-phase clocks, which obviates the need for an explicit token to be circulated. The above description is therefore only an analogy for what actually happens in the hardware. An implementation of the clocking scheme in the context of circuit switched interconnection networks is described in [Field and Cripps 85].

In a practical implementation of this scheme, and under the assumption that the buffer refill times are negligable, the number of message slots in each input buffer can be made small with surplus messages being retained in buffers within the transmitter itself. Essentially the input buffer is then split between the network and the attached device. In the limiting case, each buffer may have the capacity to hold just a single message, with all surplus messages buffered within the transmitter.

The significant property of the arbiter, from the point of view of the performance model is that, unlike a FCFS scheme, it is possible for a message generated by an attached device to be transmitted ahead of an already queued message destined for the same receiver. As an example, suppose input 1 holds the token for output k, $1 \leq k \leq x$, and that input $j, j > 2$ already has a message in its kth input buffer destined for the same output. If another message for output k arrives at the corresponding buffer of an input $i, 1 < i < j$, before the message at input 1 is transferred then that message will be transmitted before the one at input j even though it arrived in the system at a later time.

analysis. Having produced a formula for the Laplace transform of the message transmission time density, we show that in the special case of exponential service times the alternative model produces the same result as the original. The conclusions from the paper are laid out in Section 6.

2 Full Crossbar Communication Networks

The full crossbar communication network is a multi-port communication device allowing a collection of transmitters to communicate with a collection of receivers, potentially simultaneously. Each transmitter has access to every receiver and in the simple, unbuffered, crossbar only one transmitter may be connected to a given receiver at any time. Figure 1 shows the general structure of an unbuffered crossbar matrix with a input and b output connections. The function of each crossover box is to couple a horizontal bus to one of the vertical buses when the corresponding transmitter/receiver connection is required. In the rest of this paper we shall assume that $a = b = x$ for simplicity although the model is easily adapted for $a \neq b$. If the switch is being used for communication among a collection of processing elements (PEs), for example, we will indeed require $a = b$ and each PE will then act as both a transmitter and a receiver.

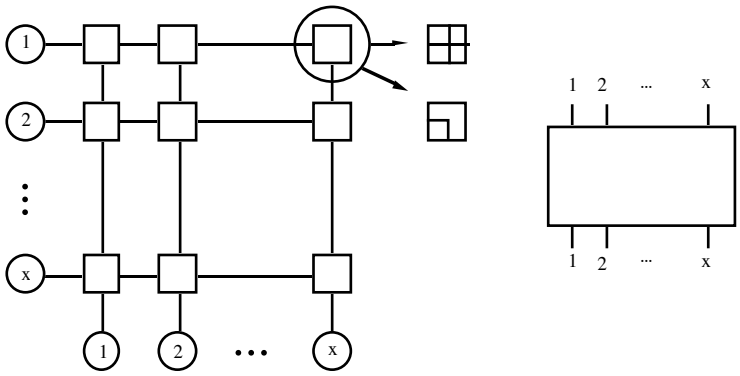

Figure 1: A Full Crossbar Network

2.1 Arbitration

Aside from the $O(x^2)$ complexity of the switching matrix in these networks, much of the complexity of the internal logic of the switch is dedicated to arbitration—that is, determining the order in which messages are transmitted when more than one transmitter wishes to access the same receiver at any time. An important property of the arbiter is that it be starvation free, that is, provided each message is of finite length, then the overall transmission time of the message (including waiting time) must also be finite in the worst case.

A common solution is to use a FCFS scheme in which the incoming messages are routed to FIFO buffers located on the output side of the network, Figure 2(a). The buffers enable the arbiter to serve messages in the order in which they arrive; at the same time they help to smooth the flow of traffic through the network and hence improve the throughput. In a practical implementation which uses finite buffers, however, a second arbitration scheme is required at the network inputs in the event that one of these buffers fills: messages blocked by the full buffer need to be admitted to the output queue fairly when buffer space is eventually released, if starvation is to be avoided.

An alternative approach, which we explore here, uses a single arbiter and locates the message buffers at the *input* side of the network. Each such buffer is divided into x separate FIFO buffers—messages for output $j, 1 \leq j \leq x$, are then queued at the back of buffer j, Figure 2(b).

The operation of the arbiter is now similar to that of the well known token passing access method, see for example [IEEE 85a]: we envisage that associated with each network output (i.e.

1 Introduction

A critical component of a parallel computer system is the interconnection network used for interprocessor or processor/memory communication and a rigorous understanding of the networks performance characteristics is essential to the effective design of parallel machines. Considerable attention has been paid to the analysis of these networks, but with a majority of the effort dedicated to the calculation of resource-based measures such as throughputs, queue lengths, mean response times etc. Whilst these measures provide useful insight into the behaviour of the machine—specifically in locating the performance bottlenecks in the system—task-oriented measures such as the distribution of transmission times are very often of greater importance in understanding the behaviour of the modelled system. Indeed, different design options are often indistinguishable solely on the basis of resourse-based performance measurements.

The objective of the work presented here is to quantify the behaviour of a 'cyclic' arbitration scheme within a full crossbar communication network with buffering. This work follows on naturally from [Field and Harrison 92] which considered a circuit-switched configuration in which transmitters wait until completion of their message transfer is complete. Crossbar switches are commonly used in parallel computers, either as the principal communication system, or as a component of a larger, multi-layer, configuration, but most implementations to date use a FCFS arbiter for conflict resolution. The cyclic arbitration system is conceptually more complicated than that of FCFS but has a particularly straightforward implementation in hardware using a multi-phase clocking arrangement as described in [Field and Cripps 85].

Whilst communication networks of the type we describe have been analysed to obtain both resource-based and task-based performance measures, see for example [Harrison and Patel 87, Kelly 86,Lang et al 82,Dias and Jump 81] and [Kruskal et al 86] respectively, these analyses have assumed FCFS arbitration, either to reflect the physical system structure, or as an approximation to the actual discipline used. As expected, the resource-based measurements are the same in both cases (the corresponding queueing disciplines for each are work-conserving). However, as will be described in more detail in Section 2, the cyclic arbiter allows some tasks to jump the queue and obtain service ahead of other tasks already queueing for service. Because of this, we would expect some variation in the transmission time distribution when compared to that of FCFS.

The response time distribution of the FCFS scheme can be relatively easily calculated, since the system can be modelled as a collection of independent FCFS queues for which the response time distribution is well known. The ability of the cyclic arbiter to allow queue jumping rather complicates the analysis, however, not least because the number of tasks ahead of a given monitored task is no longer strictly decreasing, as it is for FCFS.

The rest of the paper is organised as follows: In Section 2 we describe the architecture of the communication networks in question and give details of the operation of the cyclic arbitration scheme. In Section 2.2 we show how the system can be modelled by an open queueing model in which the network input queues are relocated at the outputs of the crossbar switch. The rearrangement of the queues for the purposes of the model enables the analysis to proceed by examining just a single queue. In Section 3 we derive the Laplace Transform of the transmission time density function for the cyclic arbitration scheme. This is obtained from a conditional Laplace Transform using a generating function together with the marginal equilibrium queue length probabilities. In Section 4 we use a numerical inversion method to obtain the distribution function of the transmission time through the network for varying network loads. These are compared with similar results from a simulation model of the system, and also with an equivalent system with FCFS arbitration. Section 5 shows how the anlysis can be extended to general service time distributions. This model is more flexible in this sense, but is more difficult to enhance than the previous model in the special case of exponential service times because of the inherent assumptions about independent and identically distributed delays in its delay cycle

similar results from a simulation model. This analysis (which assumes exponential service times) is then extended to general service times by modelling each output as a LCFS queue with a suitably reduced arrival rate. In the special case of exponential service times, this model is less flexible than the previous one since it only works in the case where the jump probability is fixed. In this case, however, it is shown to produce the same result as the original model.

UNE IMP

EHE CWI

QMIPS

INR UER

ZAR UTO

Quantitative
Modeling
In
Parallel
Systems

TRANSMISSION TIMES IN BUFFERED FULL-CROSSBAR COMMUNICATION NETWORKS WITH CYCLIC ARBITRATION

A.J. Field P.G.Harrison
Department of Computing
Imperial College
180 Queen's Gate, London SW7 2BZ

D W2.T1.11

22 September 1994

©IMP

Distribution Level: All partners

Approved by: Peter G. Harrison

Abstract In this paper we consider the distribution of message transmission times in full crossbar interconnection networks with cyclic arbitration. The transmitting devices generate messages which are placed in independent FIFO buffers. The arbitration then works by servicing the buffers in a cyclic, or 'round robin' fashion. The system is modelled as an open queueing network in which the queues appear at the network outputs. These queues are no longer FCFS queues since the scheduling discipline must take into account the cyclic nature of the arbitration mechanism: arriving messages sometimes jump the queue. We derive a recurrence equation for the Laplace transform of the density function of the transmission time through the system, conditional on the number of entries ahead of an arriving message, after that message has been placed in the queue. From this we obtain a generating function which yields the unconditional Laplace transform in a steady-state system, using the marginal equilibrium queue length probabilities. The density function is then enumerated by numerical inversion and compared with

The QMIPS Project: The QMIPS project is a collaborative research project supported by the CEC as ESPRIT-BRA project no 7269. It is being carried out by the following organisations: CWI (Amsterdam), EHEI (University of Paris V), Imperial College (London), INRIA (Sophia-Antipolis), University of Erlangen, University of Newcastle, University of Torino and University of Zaragoza.