Memory-interference model for multiprocessors based on semi-Markov processes

T.N. Mudge
H.B. Al-Sadoun
B.A. Makrucki

Indexing terms: Digital computers and computation, Information and communication theory, Modelling

Abstract: The interference that results from processors attempting to simultaneously access the same main memory in a multiprocessor can be reduced by constructing the memory from separate modules accessible through a crossbar network. The effectiveness of this solution depends on the number of processors and the number of memory modules, and on the parameters of the computation being executed, such as the think time of the processors, the frequency of their access to main memory, and the length of time these accesses are connected to memory. This paper presents a memory-interference model that allows one to evaluate the performance of crossbar-based multiprocessors. The model is a discrete-time model that explicitly describes each processing element's behaviour by means of a semi-Markov process. The chief advantage of the semi-Markov model is its conciseness and its capability of accounting for variance in model parameters. The model is first developed for the case in which memory accesses are directed to each memory module equiprobably. Central to the model is a theorem that gives the residual waiting time experienced by a processor when accessing a busy memory. Comparisons are made with earlier models. These comparisons show the semi-Markov model to be more accurate, particularly in those cases where there is a high degree of variance in the connection time. Finally, the model is generalised to deal with cases where accesses to each memory module are not equiprobable.

1 Introduction

To reduce the bottleneck to main memory in a multiprocessor, the main memory can be organised as a number of memory modules that can be accessed simultaneously by processors through a crossbar interconnection network. An example is depicted in Fig. 1. It has N processing elements (PEs) and M memory modules (MMs).

Paper 5442E(C2), first received 26th August 1986 and in revised form 8th April 1987

T.N. Mudge is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA
H.B. Al-Sadoun is with the Department of Electrical Engineering, Kuwait University, Kuwait
B.A. Makrucki is with AT & T Bell Laboratories, Holmdel, New Jersey, NJ 07733, USA

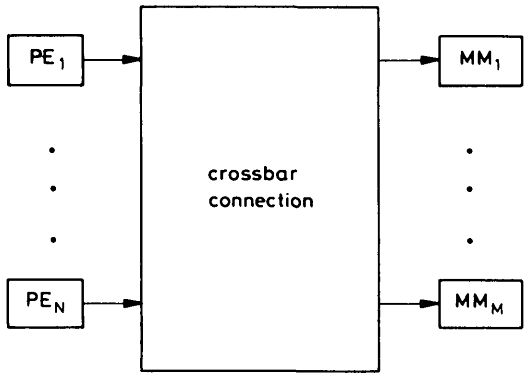

Fig. 1. *Block diagram of a multiprocessor system*

Such systems have been discussed in the literature for some years [1]; recently they have become a commercial reality [2]. Although the crossbar reduces the bottleneck to main memory, two types of memory conflict, or interference, can occur that prevent the complete elimination of the bottleneck. Type 1 conflicts arise when several PEs attempt to access an idle MM simultaneously. In this situation one of the PEs is selected, according to a predefined selection strategy, to access the MM while the other PEs wait until the selected PE is done. Type 2 conflicts arise when one or more PEs attempt to access a busy MM. In this situation the PEs wait until the MM becomes idle before they again attempt an access. Both types of conflict have a negative effect on the overall performance of the multiprocessor system by, among other things, reducing the memory bandwidth and increasing the average queueing time for memory.

This paper introduces a discrete-time model for crossbar-based multiprocessors predicated on a widely accepted set of assumptions that characterise multiprocessor behaviour as a stochastic process [1, 3–13]. The model describes the behaviour of each PE with a semi-Markov process and can be used to determine memory-interference effects on the processor's performance. Semi-Markov processes have been used successfully to model multiple-bus systems [14], where it was demonstrated that they allow complex system interactions to be modelled in a concise way. This paper complements Reference 14 by dealing with a different kind of multiprocessor system. More importantly, a theorem that is central to the semi-Markov model is proved for the first time, and the model is generalised to deal with cases where accesses to each memory module are not equiprobable.

The literature contains a number of discrete-time memory-interference models for crossbar-based systems.

In most of these models, system operation is characterised as follows. At the beginning of the system cycle a PE, which has no pending request, makes a request to access a MM with probability r. The MM is chosen at random with probability $1/M$. If a type 1 conflict occurs at the MM, the memory selects, with equal probability, one of the conflicting PEs to access it. The PEs that are not selected attempt again to access the same memory in the next system cycle. This retry will generally occur in the presence of new requests for access. The connection between the PE and the MM lasts for one system cycle; at the end of this cycle the PE releases the MM. Only type 1 conflicts take place under these assumptions. A PE will have at most one pending request waiting for a MM. The behaviour of the PEs are considered to be independent and statistically identical.

Markov chains have been used to describe the above process [3, 6]. The drawback of using Markov chains is the unmanageably large state space. This obstacle led to the development of models in which further simplifying assumptions were adopted [1, 4–10, 12]. One of the main themes of these is the reduction of the state space while maintaining reasonable agreement with simulation results.

Some of the approximate models have been extended to the case in which memory requests are not sent equiprobably to every MM [1, 7, 10, 12]. Another extension is reported in References 6 and 7 where the description of PE behaviour includes a think time, i.e., the time elapsed between releasing a MM and requesting the next MM. Think times may not be geometrically distributed as is implicitly assumed in the system operation described above. The study in Reference 6 concluded that the memory bandwidth of the multiprocessor system will be affected primarily by the first moment of the think time but will be relatively insensitive to higher moments. A third extension of system operation is introduced in References 11 and 13 where the connection time between a PE and a MM can last for more than the single system cycle. In Reference 11 it is assumed that the connection is a fixed number of cycles. This extension was motivated by a cache model where the cache line is a fixed-size block. The study in Reference 13 allowed variable connection times and showed the effect of variance in the connection times, an important consideration if cache-coherency checks are to be modelled. In Reference 13 a Markov-chain model was used to describe the behaviour of the PE.

The semi-Markov model consolidates the modelling capabilities of all the above models into one model. It further extends these capabilities by providing expressions for the average length of memory request queues and the average waiting time by a PE attempting to access a memory. Thus, it is possible to analyse the interaction of variable connection time, arbitrary think-time distribution and the distribution of the destination of the memory requests on the system performance. This additional modelling capability is attained without having to employ a complex Markov chain as in Reference 13. Indeed, the number of states in the semi-Markov process describing a PE is dependent only on the probability mass function describing the destination of the memory requests. For instance, in the simplest and most common case, that is when requests are directed to each MM equiprobably, a four-state semi-Markov process is sufficient regardless of the think and connection-time distributions.

The next section describes the assumptions that characterise the operation of crossbar-based multiprocessor systems in the uniform case. This is defined as the case when requests are directed to each MM equiprobably, and the think time of each PE and the connection time between PEs and MMs are identically distributed. Section 3 defines the performance measures that can be obtained from the model. Section 4 proves the theorem that gives the residual waiting time experienced by a processor when accessing a busy memory. This section also develops a semi-Markov model for the uniform case, illustrating it with two examples. Section 5 generalises the assumptions and presents a general semi-Markov model. Section 6 concludes the paper.

2 System-operation assumptions

The system of Fig. 1 is synchronised with a system clock whose period is referred to as the 'system cycle', or, where context allows, simply the 'cycle'. A PE in the system may be in one of three states: accessing a MM, waiting at a MM for it to become available, or thinking as when it is working on an internal task with no outstanding memory request. A MM can be either busy, when a PE is accessing it, or idle, when no PE is connected to it. The ith processing element will be denoted by PE_i and the jth memory module by MM_j. Discrete random variables will be denoted with a tilde, e.g. $\tilde{Z}$. The mean value of $\tilde{Z}$ will be denoted by $\bar{Z}$ and the second moment of $\tilde{Z}$ will be denoted $\overline{Z^2}$. The coefficient of variation of Z will be denoted by Z_v and is given by

$$Z_v = \frac{\text{standard deviation of } \tilde{Z}}{\text{expected value of } \tilde{Z}} = \sqrt{\frac{\overline{Z^2}}{(\bar{Z})^2} - 1}$$

It combines the mean and second moment to give a measure of randomness.

System operation for the uniform case will be characterised by the following assumptions:

(i) The behaviour of the PEs can be modelled as identical stochastic processes.

(ii) The PEs think for an integer number of system cycles. The thinking period of any PE is characterised by a discrete, independent random variable $\tilde{T}$.

(iii) Each PE will submit a memory request after its thinking period; requests originating from the same PE are independent of each other provided they are not resubmitted requests (see (iv) and (v)). The destination MM of the nonresubmitted requests originating from any PE will be determined by a discrete, independendent random variable $\tilde{D}$ which is uniformly distributed between 1 and M.

(iv) When the first type of memory conflict occurs, the MM selects, equiprobably, one of the conflicting PEs to gain access. The blocked PE(s) wait until the connection is completed and then they resubmit their requests to the same MM.

(v) When the second type of memory conflict occurs, the blocked PE(s) wait until the connection is completed and then they resubmit their requests to the same MM.

(vi) The connection time between any PE and any MM is characterised by a discrete independent random variable $\tilde{C}$.

Empirical evidence reported in References 5–7 supports the assumptions in the case where $\tilde{C}$ is a deterministic random variable with a value of one. Further work reported in Reference 15 supports the uniform-case assumptions where $\tilde{C}$ is a discrete random variable with arbitrary distribution.

3 Performance measures

A number of performance measures can be derived from the semi-Markov model. These measures are: BW, the memory bandwidth; PU_i, the utilisation of PE_i; MU_j, the utilisation of MM_j; $\bar{L}_j$, the average queue length for MM_j; and $\bar{W}_{ij}$, the average waiting time experienced by PE_i in the queue for MM_j.

(*a*) BW is defined as the average number of busy MMs when the multiprocessor system reaches steady state. This is the same as the average number of accessing PEs when the system reaches steady state. Hence, BW can be expressed as follows:

$$BW = \lim_{t \to \infty} \left[\sum_{i=1}^{N} Pr(PE_i \text{ is accessing at time } t) \right]$$

(*b*) PU_i is the probability that PE_i is thinking or accessing a MM when the multiprocessor system reaches steady state. Hence, PU_i can be expressed as follows:

$$PU_i = \lim_{t \to \infty} Pr(PE_i \text{ is thinking or accessing at time } t)$$
$$= 1 - \lim_{t \to \infty} Pr(PE_i \text{ is waiting at time } t)$$

Some of the memory-interference models [11, 16] which were motivated by cache studies of multiprocessor systems, define PU_i as the probability that PE_i is thinking when the system reaches steady state. In this study, we consider that memory accessing contributes to the progress of the computation and is therefore counted as useful work. The alternative quantity can be readily derived from the model as will be shown in Example 1.

(*c*) MU_j is the probability that MM_j is busy when the system reaches steady state. This is the same as the probability that any PE is accessing MM_j when the multiprocessor system reaches steady state. Hence, MU_j can be expressed as follows:

$$MU_j = \lim_{t \to \infty} \sum_{i=1}^{N} Pr(PE_i \text{ is accessing } MM_j \text{ at time } t)$$

The memory bandwidth BW can be expressed in terms of the memory utilisations as follows:

$$BW = \sum_{j=1}^{M} MU_j$$

(*d*) $\bar{L}_j$ can be defined as the expected number of PEs waiting to access MM_j. Hence, $\bar{L}_j$ can be expressed as follows:

$$\bar{L}_j = \sum_{i=1}^{N} \lim_{t \to \infty} Pr(PE_i \text{ is waiting to access } MM_j)$$

4 Memory-interference model

As noted in the Introduction, the memory-interference model uses the notion of a semi-Markov process. Semi-Markov processes are discussed in detail in References 17 and 18. For our purposes, a semi-Markov process can be viewed as a Markov chain in which the stochastic process may remain in a state for a random amount of time called the sojourn time. In a Markov chain, the sojourn time is deterministic with value one. If a semi-Markov process has K states, a mean sojourn time in state i of η_i and a limiting probability for state i in the embedded Markov chain of π_i, then the limiting probability of the semi-Markov process being in state i is given by

$$P_i = \frac{\pi_i \eta_i}{\sum_{j=1}^{K} \pi_j \eta_j} \tag{1}$$

(Strictly speaking, for eqn. 1 to be true, the embedded Markov chain should be irreducible with ergodic states. This will always be the case in the following discussion.) The rate of leaving state i, λ_i, is the reciprocal of the average time elapsed between two consecutive departures from state i, and is given by

$$\lambda_i = \frac{P_i}{\eta_i} = \frac{\pi_i}{\sum_{j=1}^{K} \pi_j \eta_j} \tag{2}$$

The average sojourn time in any one of the states of the semi-Markov processes in our discussion is at least one system cycle. Therefore, λ_i is in the range 0–1 and has the same numerical value as the probability of leaving state i at the beginning of a cycle. The term λ_i will, depending on the context, be used for both the rate of leaving state i and the probability of leaving state i.

The model uses a semi-Markov process to approximate the behaviour of each PE. Therefore, N semi-Markov processes will approximate the behaviour of the multiprocessor system. In the uniform case these N processes are identical. The states of the semi-Markov process denote the different states of a PE and can be partitioned into four disjoint subsets. In the uniform case these subsets are singletons. In the next section, where the general model is presented, this will no longer be true; however, for the remainder of this section we will consider only the uniform case. The first subset is the thinking subset, $S^{th} = \{0\}$. The process enters state 0 and remains there for a period of time with mean value η_0, equivalent to the thinking time of the PE (see Fig. 2). A

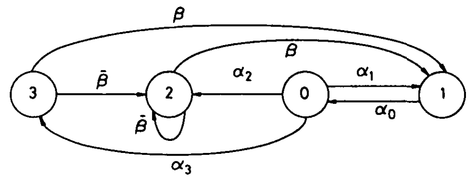

Fig. 2 *Semi-Markov process that describes PE behaviour in the uniform case*

memory request is modelled by the semi-Markov process leaving state 0. The destination state depends on the state of the requested MM. The second subset is the accessing subset, $S^{ac} = \{1\}$. The process enters state 1 and remains there for a period of time with mean value η_1, equivalent to the connection time between the PE and any MM. From state 1, the process returns to state 0, i.e. the PE resumes thinking after it has completed its memory access. The third subset is the full waiting subset, $S^{fw} = \{2\}$. The process enters state 2 when the PE requests an idle MM simultaneously with at least one other request, and the PE fails to be selected by the module, i.e. a type 1 conflict occurs and another PE is selected to have access to the MM. In this case the PE has to wait for the full period of the connection time between the MM and the selected PE; this period has a mean value of η_2. The original PE will retry to access the same MM when the selected PE releases the module. If it succeeds, the process enters state 1, otherwise the process reenters state 2. The fourth subset is the residual waiting subset, $S^{rw} = \{3\}$. The process enters state 3 when the PE requests a busy MM, i.e. when a type 2 conflict occurs. The PE has to wait for the remaining (residual) connection time before retrying to access the MM; the mean value for the residual time is η_3. The process then enters state 1 if the PE succeeds in accessing the MM, or it enters state 2 if it fails to obtain a connection. Clearly, the semi-Markov description does not include which module the PE is accessing or which module the PE is waiting to access. This does not represent an approximation of the PE's behaviour because of the symmetry in the uniform case. In nonsymmetric cases, as will be shown in the next section, the semi-Markov process has to represent this information. The underlying approximation of the model is in describing any PE behaviour independently from the other PEs while compensating for the coupling between the PE's behaviours in the transition probabilities between the states of the semi-Markov process. The coupling results from the PEs sharing the MMs.

The inputs to the semi-Markov model are the values of N, M, $\bar{T}$, $\bar{C}$ and $\overline{C^2}$. The average sojourn times of the states can be obtained from these inputs as follows:

$$\eta_j = \begin{cases} \bar{T} & j = 0 \\ \bar{C} & j = 1 \\ \bar{C} & j = 2 \\ \dfrac{\overline{C^2} - \bar{C}}{2(\bar{C} - 1)} & j = 3, \bar{C} > 1 \\ 0 & j = 3, \bar{C} = 1 \end{cases} \tag{3}$$

The average sojourn times in states 0, 1 and 2 follow from the definition of these states. The average sojourn time in state 3 is obtained from the following theorem:

Theorem 1: If requests for a busy memory are equally likely to occur at any point in time while the MM is busy, then η_3, the average residual waiting time of a busy memory seen by a requesting PE at the beginning of a system cycle, is given by $(\overline{C^2} - \bar{C})/2(\bar{C} - 1)$ when $\bar{C} > 1$ and by 0 when $\bar{C} = 1$.

Proof: Two events, A_i and B, will be used in this proof. They are defined as follows: A_i is the event that a requesting PE will see a residual accessing time of i cycles; B is the event that a requesting PE will find a busy MM. Therefore, the average residual accessing time of a busy memory seen by a requesting PE can be expressed as

$$\eta_3 = \sum_{i=0}^{S-1} i Pr[A_i \mid B]$$

where S is the maximum connection time between a PE and any MM. The summation extends only to $S - 1$ because residual time does not include full waiting. In the case where $\bar{C} = 1$, $\tilde{C}$ must be a deterministic random variable of length 1. Thus $S = 1$, and, therefore, from the previous equation $\eta_3 = 0$. Since A_i and B are dependent events, Bayes' rule can be used to obtain the following equation:

$$Pr[A_i \mid B] = \frac{Pr[A_i \cap B]}{Pr[B]}$$

Let $Pr[\tilde{C} = i]$, the probability mass function of $\tilde{C}$, be written as $c(i)$, then the term $Pr[A_i \cap B]$ can be determined as follows:

$$Pr[A_i \cap B] = \sum_{j=1}^{S-i} (\text{the accessing PE submitted, } j \text{ cycles ago, an accessing request of } i + j \text{ cycles})$$

If requests for a busy memory are equally likely to occur at any point in time while the MM is busy, then the equation can be expressed as

$$Pr[A_i \cap B] = \sum_{j=1}^{S-i} c(i+j) = \sum_{j=i+1}^{S} c(j)$$

At the same time, the term $Pr\,[B]$ can be determined as follows:

$$Pr[B] = \sum_{j=1}^{S-1} Pr\begin{pmatrix}\text{the accessing PE obtained the} \\ \text{connection, } j \text{ cycles ago, for} \\ \text{at least } j+1 \text{ cycles}\end{pmatrix}$$

$$= \sum_{j=1}^{S-1} \sum_{k=j+1}^{S} c(k) = \sum_{j=1}^{S} (j-1)c(j)$$

Therefore, for $\bar{C} > 1$, η_3 can be expressed as follows:

$$\eta_3 = \sum_{i=0}^{S-1} i \frac{\sum_{j=i+1}^{S} c(j)}{\sum_{j=1}^{S} (j-1)c(j)} = \frac{\sum_{i=1}^{S-1} i \sum_{j=i+1}^{S} c(j)}{\sum_{j=1}^{S} (j-1)c(j)}$$

$$= \frac{\sum_{j=1}^{S} \frac{j(j-1)}{2} c(j)}{\sum_{j=1}^{S} (j-1)c(j)} = \frac{\overline{C^2} - \bar{C}}{2(\bar{C} - 1)}$$

This formula for η_3 is very similar to the usual one for mean residual life, i.e. $\overline{C^2}/2\bar{C}$. The difference arises because we do not consider the full waiting event to be part of the residual time. The behaviour of the formula is consistent with intuition, as the following three examples illustrate. When $\bar{C} = 1$, then $\eta_3 = 0$, in other words, only full waiting can occur. When $\bar{C} = 2$ and $\overline{C^2} = 4$ (a fixed-length connection of 2 cycles), then $\eta_3 = 1$, in other words, the wait is usually half the fixed-length connection time. Finally, when $\tilde{C}$ is geometrically distributed with parameter p, such that $\bar{C} = 1/p = \eta_2$ and $\overline{C^2} = (2 - p)/p^2$, then $\eta_3 = 1/p = \eta_2$, in other words, there is an absence of memory in the system. It should be noted that although, as the above examples show, the formula for η_3 behaves in a desirable manner, it is based on an assumption that requests for busy memories occur at any time with equal probability. This is not proved within the assumptions; however, it is supported by the experimental results below.

The following terms are useful in formulating the model: R, WIN and $BUSY$. The first, R, is the probability that a PE makes a request to access a particular MM at the beginning of a system cycle. This is the probability that one of two events occurs. The first event is that the PE will direct a new request to that MM, i.e. the processor leaves state 0. And the second event is that the PE will resubmit a previously blocked request to that particular MM, i.e. it leaves state 2 or 3. Thus, R is given by

$$R = \frac{1}{M}(\lambda_0 + \lambda_2 + \lambda_3) \tag{4}$$

The term WIN is the probability that an idle MM selects a particular PE's request over other requests, if any, at the beginning of a system cycle. The probability that no PE requests a particular MM is $(1 - R)^N$, and thus the probability that a particular MM is requested by at least one PE is $[1 - (1 - R)^N]$. Since the expected number of PEs which requested that MM is NR, WIN can be expressed as follows:

$$WIN = \frac{1}{NR}[1 - (1 - R)^N] \tag{5}$$

Finally, $BUSY$ is the probability that a PE finds a particular MM busy servicing a request from another PE which is not ready to release it. In other words, the requesting PE experienced a type 2 memory conflict. The probability that a PE is accessing a MM and is not ready to release it is given by $(P_1 - \lambda_1)$, i.e., the probability that the PE is in the accessing state (P_1) and will not leave it next cycle (λ_1). Any of $(N - 1)$ PEs may be the one being serviced and there are M possible MMs, therefore $BUSY$ is given by

$$BUSY = \frac{N-1}{M}(P_1 - \lambda_1) = \frac{N-1}{M}(\bar{C} - 1)\lambda_1 \tag{6}$$

The last step follows from eqn. 2.

The transition probabilities between the states of the semi-Markov process can be written as follows:

$$\alpha_j = \begin{cases} 1 & j = 0 \\ (1 - BUSY)\, WIN & j = 1 \\ (1 - BUSY)(1 - WIN) & j = 2 \\ BUSY & j = 3 \end{cases} \tag{7}$$

$$\beta = (1 - BUSY)WIN$$
$$\bar{\beta} = 1 - \beta$$

They can be obtained as follows. When the process of Fig. 2 leaves the thinking state it can enter any of the other states: it enters the accessing state with probability α_1 if the MM is idle and the PE's request is selected; or it enters the full waiting state with probability α_2 if the MM is idle and the PE's request fails to be selected; or it enters the residual waiting state with probability α_3 if the MM is busy. The process always returns to the thinking state after it leaves the accessing state, thus $\alpha_0 = 1$. The process leaves the residual waiting state or the full waiting state to enter the accessing state with probability β if the requested MM is idle and the PE's request is selected; otherwise it will enter the full state with probability $\bar{\beta}$. (Although $\alpha_1 = \beta$, we distinguish them in preparation for the general case discussed in the next section.)

The limiting probabilities of the embedded Markov chain (πs) can be solved in terms of $BUSY$ and WIN from eqn. 7. Then, from eqns. 2 and 4 we have

$$\sum_{j=0}^{3} \pi_j \eta_j = \frac{\pi_1}{M\beta R} \tag{8}$$

therefore, $\lambda_1 = M\beta R = M(1 - BUSY)WIN\ R$. Substituting this in eqn. 6 yields

$$BUSY = \frac{(N-1)(\bar{C} - 1)WIN\ R}{1 + (N-1)(\bar{C} - 1)WIN\ R} \tag{9}$$

The semi-Markov limiting probabilities (Ps) can be derived from the πs, eqns. 1 and 8, and can be expressed as functions of R and the transition probabilities as follows:

$$P_j = \begin{cases} \eta_0 M\beta R & j = 0 \\ \eta_1 M\beta R & j = 1 \\ \left(\alpha_2 + \dfrac{(\bar{\beta})^2}{\beta}\right)\eta_2 M\beta R & j = 2 \\ \alpha_3 \eta_3 M\beta R & j = 3 \end{cases} \tag{10}$$

These four equations sum to 1, yielding

$$R = \frac{1}{\left(\eta_0 + \left(1 + \alpha_2 + \dfrac{(\bar{\beta})^2}{\beta}\right)\eta_1 + \alpha_3\eta_3\right)M\beta} \tag{11}$$

(Note that $\eta_1 = \eta_2$).

At this point we have a set of nonlinear equations. The nonlinearity occurs because the transition probabilities are defined as functions of R by way of WIN (eqn. 5) and $BUSY$ (eqn. 9); on the other hand, the R is defined as functions of the transition probabilities (eqn. 11). An iterative algorithm can be used to solve these equations. The algorithm will iterate on the value of R and then the performance measures of the system can be derived. The algorithm runs as follows:

(i) calculate the ηs from eqn. 3, and choose an initial value for R (we used $R = 0.5$)
(ii) calculate WIN using eqn. 5
(iii) calculate $BUSY$ using eqn. 9
(iv) calculate the transition probabilities using eqn. 7
(v) calculate new value for R from eqn. 11
(vi) repeat steps (ii)–(v) until R has the desired accuracy*.

The solution for R may be used to calculate the Ps from eqn. 10. These can then be used to calculate the performance measures of Section 3, as follows:

$$BW = NP_1$$

$$PU_i = P_0 + P_1 \qquad \forall i$$

$$MU_j = \frac{N}{M}P_1 \qquad \forall j$$

$$\bar{L}_j = \frac{N}{M}(P_2 + P_3) \qquad \forall j$$

$$\bar{W}_{ij} = \eta_2\left[\frac{1}{\beta} - (1 + BUSY)\right] + \eta_3 BUSY \qquad \forall i, j$$

The last equation is the only one that does not follow directly from the definition of the states of Fig. 2. It can be derived by calculating the expected value of $\tilde{W}_{ij}$ in the usual way from the following probability mass function of $\tilde{W}_{ij}$:

$$Pr[\tilde{W}_{ij} = 0] = \alpha_1$$

$$Pr[\tilde{W}_{ij} = k\eta_2 \text{ without visiting state 3}]$$
$$= \alpha_2(\bar{\beta})^{k-1}\beta \qquad k \geq 1$$

$$Pr[\tilde{W}_{ij} = (\eta_3 + k\eta_2) \text{ with exactly one visit to state 3}]$$
$$= \alpha_3(\bar{\beta})^k\beta \qquad k \geq 0$$

The derivation of the above equations proceeds as follows. The probability that the process moves from state 0 to state 1 without waiting is α_1. The probability that the process moves from state 0 to state 1 and makes k visits to state 2 without visiting state 3 is $\alpha_2\ (\bar{\beta})^{k-1}\beta$,

* This is a simple fixed-point iteration scheme. Higher-order iteration schemes could be used but were found unnecessary in our experiments. Four iterations were usually sufficient.

where $k \geq 1$. The probability the process moves from state 0 to state 1 and makes one visit to state 3 and k visits to state 2 is $\alpha_3(\bar{\beta})^k\beta$, where $k \geq 0$. These three cases exhaust all the possible values for $\tilde{W}_{ij}$.

As a final note, it can be shown that, when $\bar{C} = \overline{C^2} = 1.0$, the semi-Markov model simplifies to the rate-adjusted model of Reference 7.

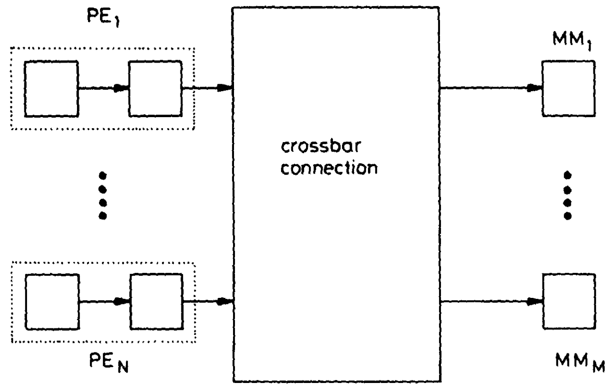

Fig. 3 *Multiprocessor system with private cache memories*

4.1 *Example 1*

In this example, we will use the model to study a multiprocessor system with private cache memories. The system is shown in Fig. 3. Each PE in the system consists of a processor and a private cache memory. The MMs form the shared memory. The system operation is characterised as follows. At the end of a system cycle a processor causes a cache fault in its private cache with probability m. The average think time $\bar{T}$ is related to m by the equation $\bar{T} = (1/m - 1) + 1 = 1/m$, i.e. the think time between consecutive faults plus the cycle during which the fault occurs. The PE which has a cache fault chooses, with equal probability, one of the MMs with which to transfer a line. When the connection is established between the PE and the MM it lasts for a variable number of cycles given by a discrete random variable, $\tilde{C}$. The variability arises because cache coherency requires reads to be performed in some transfers and writes before reads in others. In the case of a memory conflict, of either type, the rejected PEs will resubmit their requests to the same MM when that MM becomes idle. This retry will generally occur in the presence of new requests for access.

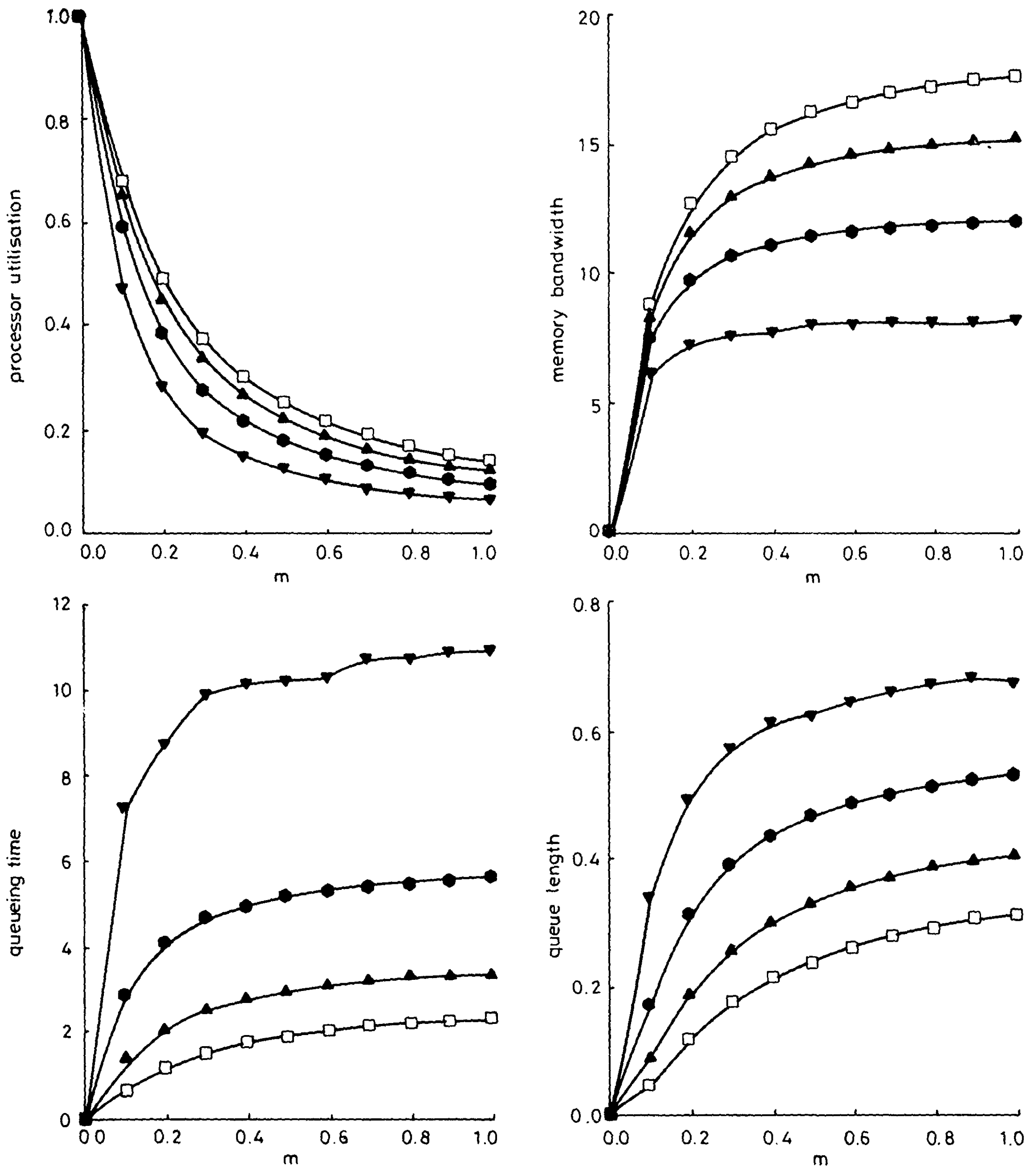

Fig. 4 *Simulation results for a 32 × 32 system*
□ $C_v = 0.0$ ▲ $C_v = 1.0$ ● $C_v = 2.0$ ▼ $C_v = 4.0$

A study, reported in Reference 11, models a similar multiprocessor system in which the random variable $\tilde{C}$ is deterministic and cache coherence is ignored. A later study, reported in Reference 16, develops a more realistic model which includes the effects of coherence checks. However, this model assumes a parallel-pipelined organisation of memories rather than the parallel organisation of Fig. 3. Nevertheless, as a result of considering cache coherency the work of Reference 16 demonstrates the need to model the connection time as a nondeterministic random variable rather than a deterministic random variable, although it does not develop a model suitable for connection times having a high degree of nondeterminism (i.e. a large coefficient of variation C_v). Our results below confirm the need to consider nonzero values for C_v.

In this example, we compare the results of our model and the results of the simple cache model reported in Reference 11 with simulations. Furthermore, we demonstrate the effect of different connection distributions on the results of our model and the simple cache model.

Fig. 4 shows the simulation results for a 32 × 32 system for different connection distributions (all have $\bar{C} = 4.0$). It can be seen that the variation in C_v can dramatically effect BW, PU_i, $\bar{W}_{ij}$ and $\bar{L}_j$. This effect is not captured in the simple cache model because it uses only the first moment of the connection time. Note that PU_i is defined in accordance with Reference 11, that is to say, as the probability that PE_i is thinking when the multiprocessor system reaches steady state (see Section 3). The PU_i measure was less sensitive to variations in C_v for this reason. One clear observation from the results of Fig. 4 is that C_v should be kept to a minimum. This may have some impact on how cache faults are handled.

Fig. 5 shows the error between simulation and the semi-Markov model as a percentage, where

$$\text{Error} \triangleq \frac{\text{model results} - \text{simulation results}}{\text{simulation results}} \times 100$$

The errors are within 10% for utilisation measures (PU and BW) and within 20% for queueing measures ($\bar{W}$ and

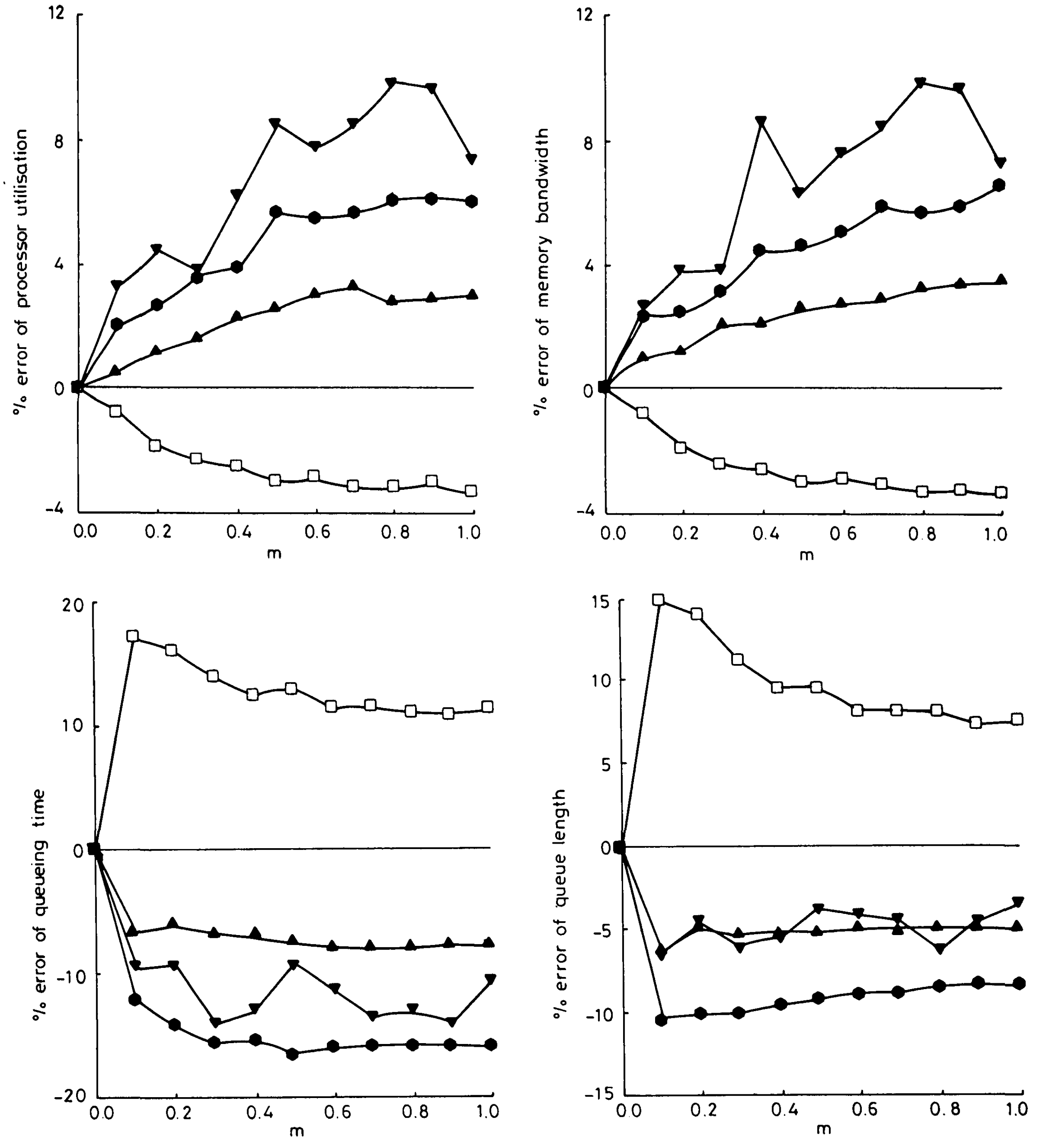

Fig. 5 *Error for the semi-Markov model for a 32 × 32 system*
□ $C_v = 0.0$ ▲ $C_v = 1.0$ ● $C_v = 2.0$ ▼ $C_v = 4.0$

$\bar{L}$). Possible reasons for the relative lack of robustness of the queueing measures has been discussed in Reference 15. The error appears to increase with C_v.

Fig. 6 shows the error between the simulation results and the results obtained from the simple cache model. For the most part, it can be seen that the semi-Markov model produces results much closer to simulation.

4.2 Example 2

The experiment of Example 1 was repeated for a 32×8 system. Fig. 7 and 8 show the results. Again, variation in C_v is seen to dramatically effect the performance measures. In this case the errors have increased to within 20% for utilisation measures and 25% for queueing measures. This reduction in accuracy is common for many probabilistic memory-interference models when there is significantly more processors than memories. The model should be modified if accuracy is to be retained. When there are just one or two memories, this modification can take the form of coupling the semi-Markov processes for individual processors into a single process.

5 General case

In the general case, $\tilde{D}$ may have any distribution and the N semi-Markov processes associated with each PE need not be identical. The multiprocessor operation assumptions now become:

(i) The behaviour of the PEs can be modelled as stochastic processes.

(ii) The PEs think for an integer number of system cycles. The thinking period of PE_i is characterised by a discrete, independent random variable $\tilde{T}_i$ where $1 < i < N$.

(iii) Each PE will submit a memory request after its thinking period; requests originating from the same PE are independent of each other provided they are not resubmitted requests (see (iv) and (v)). The destination MM of the nonresubmitted requests originating from PE_i will be determined by a discrete, independent random variable $\tilde{D}_i$ where $1 \leq i \leq N$.

(iv) When the first type of memory conflict occurs, the MM selects, equiprobably, one of the conflicting PEs to

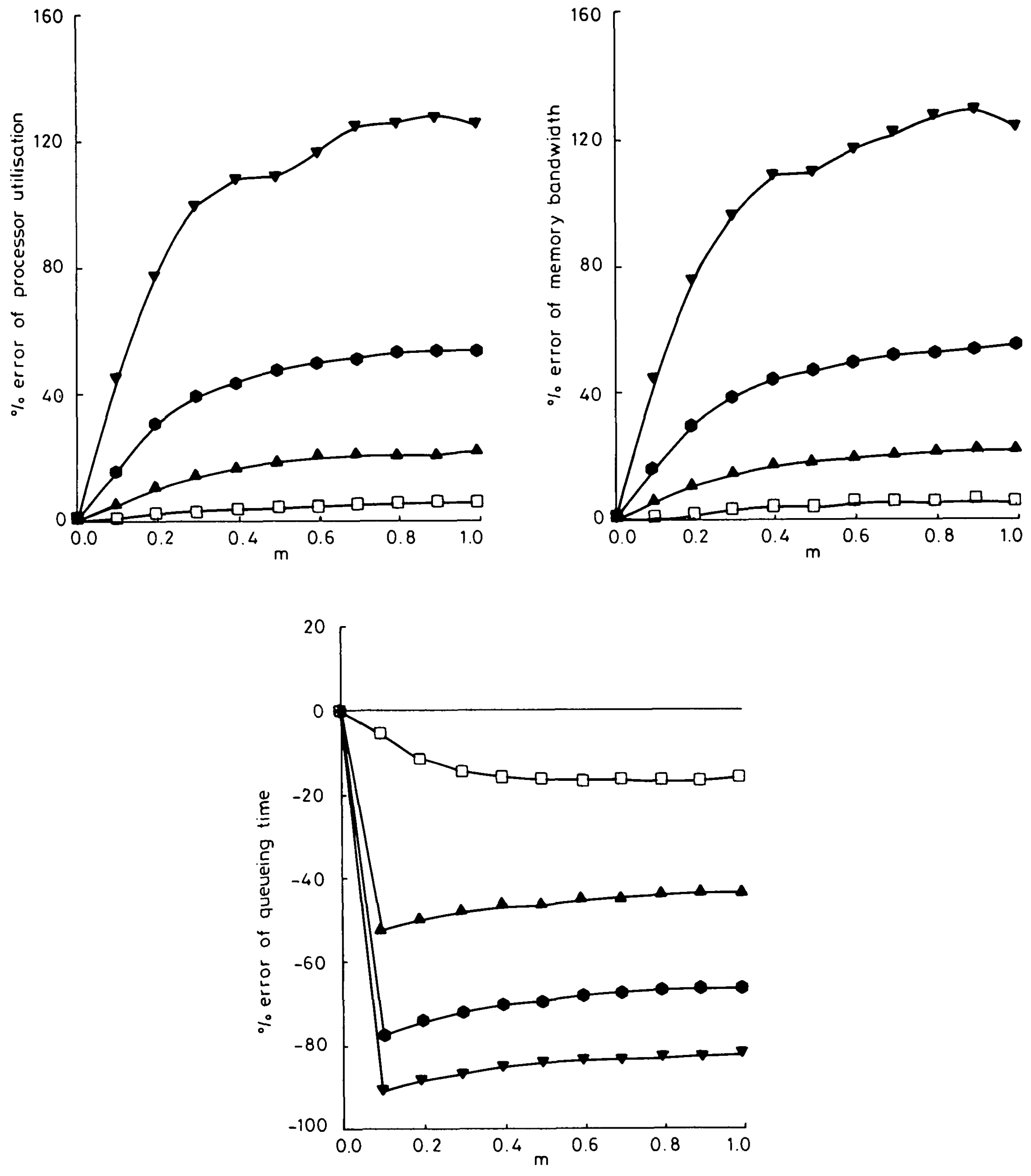

Fig. 6 *Error for the simple cache model*
□ $C_v = 0.0$ ▲ $C_v = 1.0$ ● $C_v = 2.0$ ▼ $C_v = 4.0$

gain access. The blocked PE(s) wait until the connection is completed and then they resubmit their requests to the same MM.

(v) When the second type of memory conflict occurs, the blocked PE(s) wait until the connection is completed and then they resubmit their requests to the same MM.

(vi) The connection time between PE_i and MM_j is characterised by a discrete independent random variable $\tilde{C}_{ij}$ where $1 \leqslant i \leqslant N$ and $1 \leqslant j \leqslant M$.

The input to the semi-Markov model are the values of $\bar{T}_i$, $\bar{C}_{ij}$, $\overline{C^2}_{ij}$ and $a_{ij} \triangleq Pr[\tilde{D}_i = j]$.

The semi-Markov process of Fig. 9 describes the behaviour of PE_i. The general case requires N different semi-Markov processes to describe the system behaviour. The state space of each of the processes can be divided to four disjoint subsets: $S_i^{th} = \{0\}$, $S_i^{ac} = \{1, \ldots, M\}$, $S_i^{fw} = \{M+1, \ldots, 2M\}$ and $S_i^{rw} = \{2M+1, \ldots, 3M\}$. It may be seen that this semi-Markov process collapses to the uniform case process if $a_{ij} = 1/M$ for all i and j. The solution in the general case follows the lines of earlier cases, therefore only the outline of the solution will be presented. The equation numbers indicate correspondence with counterparts in the uniform case.

The average sojourn times in the states are as follows:

$$\eta_{ij} = \begin{cases} \bar{T}_i & j \in S_i^{th} \\ \bar{C}_{ij} & j \in S_i^{ac} \\ \displaystyle\sum_{\substack{k=1 \\ k \neq i}}^{N} \frac{R_{k,\,j-M}}{\displaystyle\sum_{\substack{l=1 \\ l \neq i}}^{N} R_{l,\,j-M}} \bar{C}_{k,\,j-M} & j \in S_i^{jw} \\ \displaystyle\sum_{\substack{k=1 \\ k \neq i}}^{N} \frac{(P_{k,\,j-2M} - \lambda_{k,\,j-2M})}{\displaystyle\sum_{\substack{l=1 \\ l \neq 1}}^{N} (P_{l,\,j-2M} - \lambda_{l,\,j-2M})} & \\ \dfrac{\overline{C^2}_{k,\,j-2M} - \bar{C}_{k,\,j-2M}}{2(\bar{C}_{k,\,j-2M} - 1)} & j \in S_i^{rw} \end{cases} \tag{3'}$$

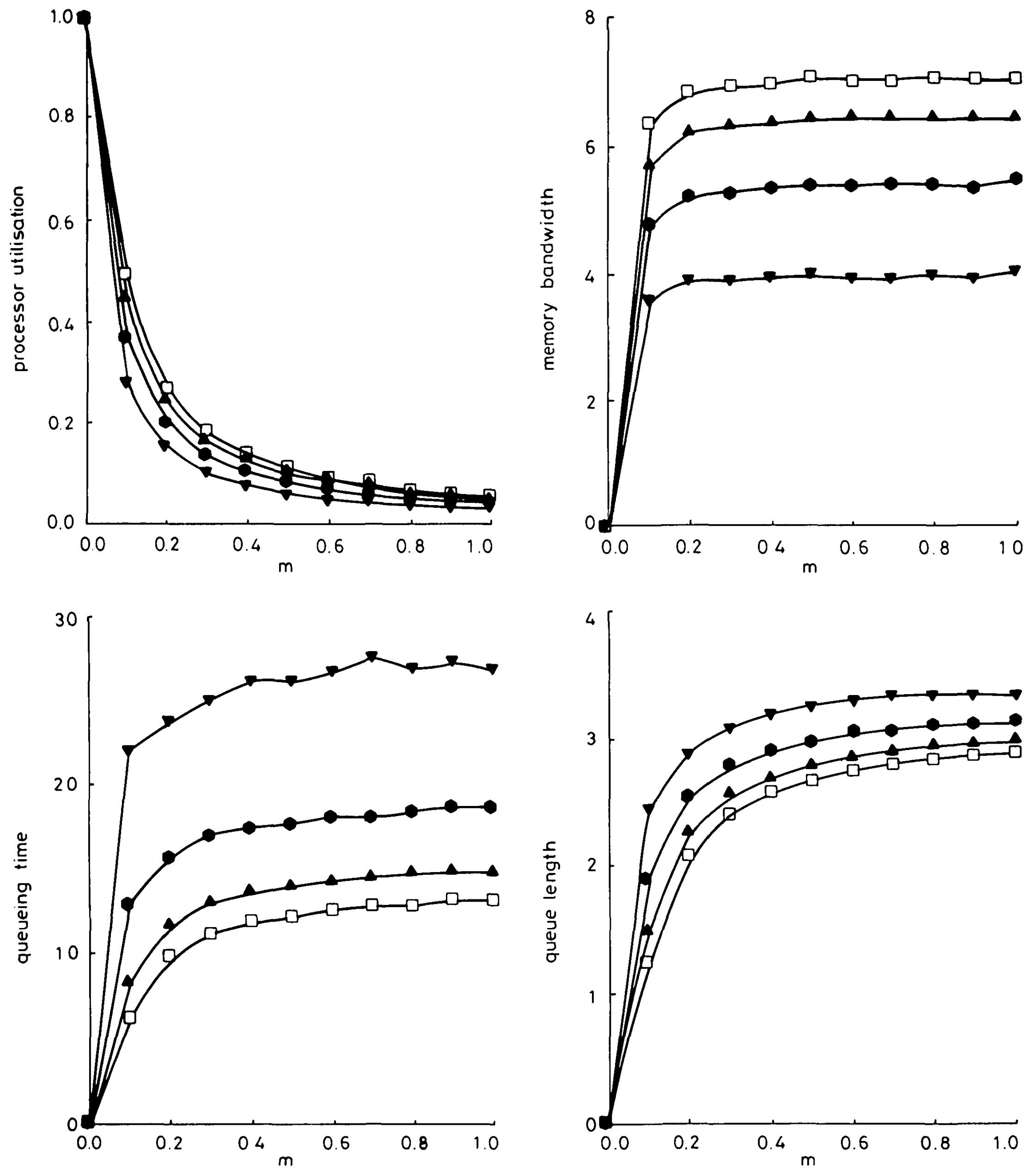

Fig. 7 *Simulation results for a 32 × 8 system*
□ $C_v = 0.0$ ▲ $C_v = 1.0$ ● $C_v = 2.0$ ▼ $C_v = 4.0$

where

$$R_{ij} = a_{ij}\lambda_{i,o} + \lambda_{i,j+M} + \lambda_{i,j+2M} \qquad (4')$$

Analogously to the case for η_3 when $\bar{C}_{k,j-2M} = 1$ the corresponding terms in the summations for the residual waiting times are 0. The first subscript indicates the PE and the second indicates the state. The terms WIN and $BUSY$ also require subscripts: the first indicates the PE and the second indicates the MM. These terms can be expressed as follows:

$$WIN_{ij} = \sum_{k=1}^{N} \frac{1}{k} \Psi_{ijk} \qquad (5')$$

where,

$$\Psi_{ijk} = \sum_{l=1}^{\binom{N-1}{k-1}} \prod_{\substack{h=1 \\ h \neq i}} \omega_{ijk}(h)$$

and,

$$\omega_{ijk}(h) = \begin{cases} R_{hj} & \text{if } PE_i,\ PE_h \text{ and } k-2 \text{ other PEs request } MM_j \text{ in the } l\text{th case} \\ 1 - R_{hj} & \text{otherwise} \end{cases}$$

Furthermore

$$BUSY_{ij} = \sum_{\substack{k=1 \\ k \neq i}}^{N} (P_{kj} - \lambda_{kj}) = \sum_{\substack{k=1 \\ k \neq i}}^{N} (\bar{C}_{kj} - 1)\lambda_{kj} \qquad (6')$$

The transition probabilities between the states of the process can be defined as follows:

$$\alpha_{ij} = \begin{cases} 1 & j \in S_i^{th} \\ a_{ij}(1 - BUSY_{ij})\, WIN_{ij} & j \in S_i^{ac} \\ a_{i,j-M}(1 - BUSY_{i,j-M})(1 - WIN_{i,j-M}) & j \in S_i^{fw} \\ a_{i,j-2M}\, BUSY_{i,j-M} & j \in S_i^{rw} \end{cases}$$

$$\beta_{ij} = WIN_{ij}(1 - BUSY_{ij}) \qquad j \in S_i^{ac}$$

$$\bar{\beta}_{ij} = 1 - \beta_{ij} \qquad j \in S_i^{ac}$$

$$(7')$$

The embedded Markov chain can be solved and the πs can be represented as functions of transition probabil-

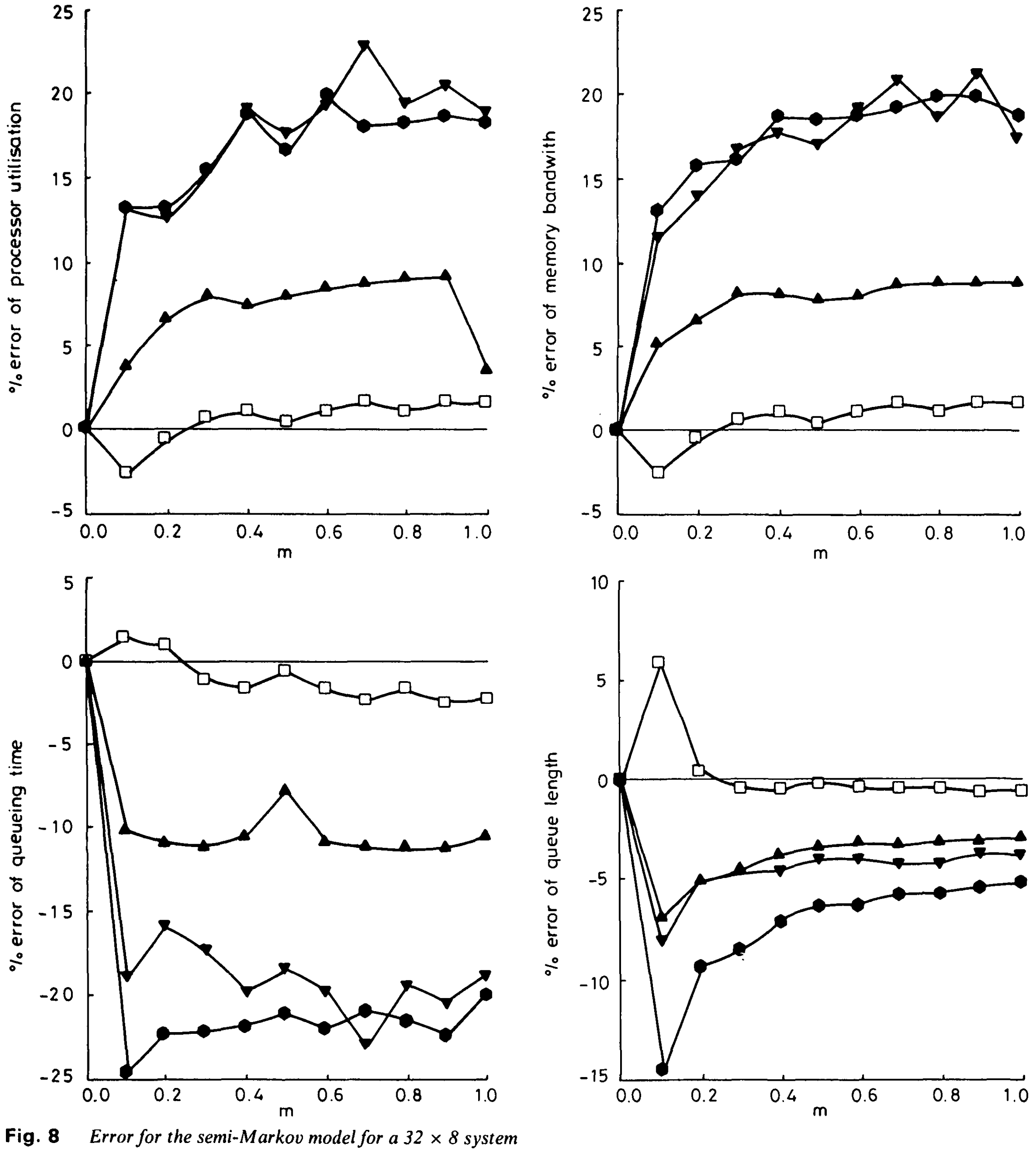

Fig. 8 *Error for the semi-Markov model for a 32 × 8 system*
□ $C_v = 0.0$ ▲ $C_v = 1.0$ ● $C_v = 2.0$ ▼ $C_v = 4.0$

ities. Then, from eqn. 2, λ_{kj} may be defined as a function of R and the transition probabilities. Therefore, using

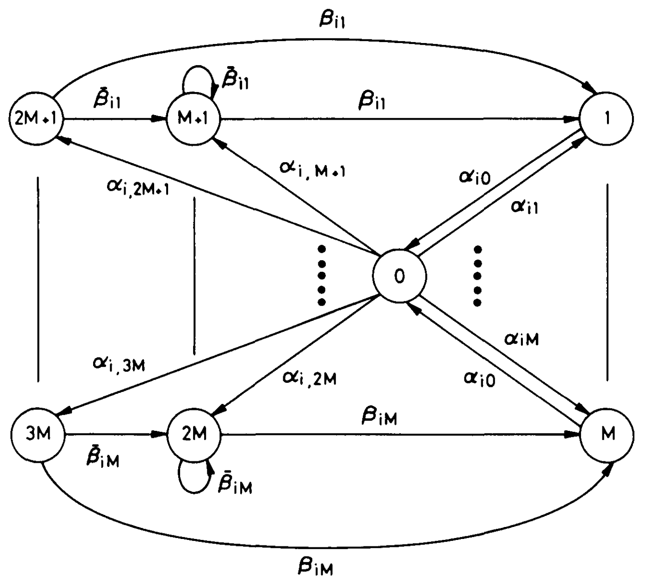

Fig. 9 *Semi-Markov process that describes PE behaviour in the general case*

eqn. 6′, the term $BUSY_{ij}$ can be expressed as follows:

$$BUSY_{ij} = \sum_{\substack{k=1 \\ k \neq i}}^{N} (\bar{C}_{kj} - 1)\beta_{kj} R_{kj} \tag{9'}$$

Furthermore, the semi-Markov process limiting probabilities can be expressed as functions of R and the transition probabilities as shown below:

$$P_{ij} = \begin{cases} \eta_{ij} \sum_{k=1}^{M} \beta_{ik} R_{ik} & j \in S_i^{th} \\ \eta_{ij} \beta_{ij} R_{ij} & j \in S_i^{ac} \\ \left[\alpha_{ij} + a_{i,j-M} \dfrac{(\bar{\beta}_{i,j-M})^2}{\beta_{i,j-M}}\right] \dfrac{\eta_{ij}\beta_{i,j-M}}{a_{i,j-M}} R_{i,j-M} & j \in S_i^{fw} \\ \dfrac{\alpha_{ij}\eta_{ij}}{a_{i,j-2M}} \beta_{i,j-2M} R_{i,j-2M} & j \in S_i^{rw} \end{cases} \tag{10'}$$

To solve the general case an iterative algorithm can be used similar to that proposed for the uniform case. In the general case the algorithm iterates on the set of variables R_{ij}, where $1 \leq i \leq N$ and $1 \leq j \leq M$. The performance measures can be derived from the Ps as follows:

$$BW = \sum_{i=1}^{N} \sum_{j=1}^{M} P_{ij}$$

$$PU_i = 1 - \sum_{j=M+1}^{3M} P_{ij}$$

$$MU_j = \sum_{i=1}^{N} P_{ij}$$

$$\bar{L}_j = \sum_{i=1}^{N} (P_{i,j+M} + P_{i,j+2M})$$

$$\bar{W}_{ij} = \eta_{i,M+j}\left[\frac{1}{\beta_{ij}} - (1 + BUSY_{ij})\right] + \eta_{i,2M+j} BUSY_{ij}$$

Examples illustrating the general case can be found in Reference 19. A similar degree of accuracy was obtained in the general case results as in those for the uniform case of Section 4.

6 Conclusion

This paper has presented a discrete-time model of memory interference for crossbar-based multiprocessor systems. The model characterises such systems by describing the behaviour of each PE as an independent semi-Markov process. By viewing events from the perspective of the PEs, it is possible to model memory-interference effects not explicitly modelled previously, such as queueing time and queue length. In addition, the effects of the coefficient of variation of the connection time can also be readily modelled. Its importance was shown in the examples given.

The complexity of solution for the model was shown to be directly related to the number of distinct values that the discrete random variable $\tilde{D}$ can take on. For the uniform case, the model requires only four states independent of the number of discrete values that N, M, $\tilde{T}$ and $\tilde{C}$ can have. One insight that emerged from developing the model was that the simplifying assumption in the rate-adjusted models, typified by the model in Reference 7, is the decoupling between the PEs rather than the resubmission policy as suggested in Reference 10.

Examples were presented to illustrate how the model can be used to predict the performance of multiprocessor systems. Simulations were included for comparison, and, in all cases except queueing time and queue length, the model differed by about 20% (the queueing time and queue length differed by less than 25%). In most cases the error was much smaller. The model performed well relative to a more typical model when C_v was high. Further examples can be found in Reference 19. The error in the semi-Markov model can be attributed to the decoupling between the PE's behaviours. Because of this decoupling, certain events are permissable in the model that cannot occur in the real system. For example, the event that all PEs are waiting has a nonzero probability. In reality, at least one PE would be accessing.

7 Acknowledgment

This work was supported in part by the US National Science Foundation under grant MCS-8009315

8 References

1 MUDGE, T.N., and MAKRUCKI, B.A.: 'Probabilistic analysis of a crossbar switch.' in Proceedings of the IEEE 9th Annual Symposium on Computer Architecture, Austin, Texas, USA, April 1982, pp. 311–319
2 CHANG, R.: 'Parallel-processing computer overcomes memory contention,' *Comput. Des.*, 15th September 1985
3 SKINNER, C.E., and ASHER, J.R.: 'Effects of storage contention on system performance,' *IBM Syst.* 1969, **8**, (4), pp. 319–333
4 STRECKER, W.D.: 'Analysis of the instruction execution rate in certain computer structures'. Ph.D. Thesis, Carnegie-Mellon University 1970
5 BHANDARKER, D.P.: 'Analysis of memory interference in multiprocessors,' *IEEE Trans.* 1975, **C-24**, (9), pp. 897–908
6 BASKETT, F., and SMITH, A.J.: 'Interference in multiprocessor computer systems with interleaved memory,' *Commun. ACM*, 1976, **19**, (6), pp. 327–334
7 HOOGENDOORN, C.H.: 'A general model for memory interference in multiprocessors,' *IEEE Trans.* 1977, **C-26**, (10), pp. 998–1005
8 RAU, B.R.: 'Interleaved memory bandwidth in a model of a multiprocessors,' *ibid.*, 1979, **C-28**, (9), pp. 678–681
9 PATEL, J.H.: 'Processor-memory interconnections for multiprocessors,' *ibid.*, 1981, **C-30**, (10)., pp. 771–780
10 YEN, D.W.L., PATEL, J.H., and DAVIDSON, E.S.: 'Memory interference in synchronous multiprocessor systems,' *ibid.*, 1982, **C-31**, (11), pp. 1116–1121

11 PATEL, J.H.: 'Analysis of multiprocessors with private cache memories,' *ibid.*, 1982, **C-31**, (4), pp. 296–304
12 BHUYAN, L.N., and LEE, C.W.: 'An interference analysis of interconnection networks.' in Proceedings of the International Conference on Parallel Processing, Bellaire, Michigan, USA, August 1983, pp. 2–9
13 MUDGE, T.N., and Al-SADOUN, H.B.: 'Memory interference models with variable connection time,' *IEEE Trans.*, 1984, **C-33**, (11), pp. 1033–1038
14 MUDGE, T.N., and Al-SADOUN, H.B.: 'A semi-Markov model for the performance of multiple-bus systems,' *ibid.*, 1985, **C-34**, (10), pp. pp. 934–942
15 MAKRUCKI, B.A.: 'A stochastic model of multiprocessing'. Ph.D. Thesis, University of Michigan, 1984
16 BRIGGS, F.A., and DUBOIS, M.: 'Effectiveness of private caches in multiprocessor systems with parallel-pipelined memories,' *IEEE Trans.*, 1983, **C-32**, (1) pp. 48–59
17 CINLAR, E.: 'Introduction to stochastic processes'. Prentice-Hall Inc., Englewood Cliffs, N.J., 1975
18 ROSS, S.M.: 'Applied probability models with optimization applications.' Holden-Day, Inc., San Francisco, Calif., 1970
19 Al-SADOUN, H.B.: 'A study in memory interference models.' Ph.D. Thesis, University of Michigan, 1985