

THE UNIVERSITY OF MICHIGAN
COMPUTING RESEARCH LABORATORY¹

**MEMORY INTERFERENCE MODELS WITH
VARIABLE CONNECTION TIME**

T.N. Mudge and Humoud B. Al-Sadoun

CRL-TR-16-84

MARCH 1984

**Room 1079, East Engineering Building
Ann Arbor, Michigan 48109
USA
Tel: (313) 763-8000**

¹This work was supported in part by National Science Foundation under Grant MCS-8009315. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the funding agencies.

MEMORY INTERFERENCE MODELS WITH VARIABLE CONNECTION TIME

by

T. N. Mudge and Humoud B. Al-Sadoun
Computing Research Lab
Department of Electrical and Computer Engineering
The University of Michigan
Ann Arbor, MI 48109
313/764-0203

Abstract

This paper develops two discrete time models of the interference that occurs during memory access in multiprocessor systems. These models, the equivalent rate model and the Markov chain model, provide for variable connection times between processors and memories if these times can be characterized by a discrete random variable, X , with a probability mass function $f(i)$. Neither model requires a complete description of $f(i)$. The equivalent rate model, which is the simpler, requires only the first moment, while the Markov chain model requires the first and second moments. The models yield estimates of the bandwidth, BW , and related measures such as the probability that a memory request is accepted, P_a and processor utilization, U_p . A brief summary of earlier discrete time models is included, and it is shown that one of them is a proper special case of the Markov chain model. Comparisons with simulations show that both models give good estimates of BW when the coefficient of variation, C_v , of X is small. When C_v reaches 2.0 the Markov chain model still shows an error of less than 4% while the equivalent rate model exhibits a 50% error that, unlike the Markov chain model, continues to increase with increase in C_v . Finally, it is shown that BW drops significantly with increase in C_v , suggesting that processor-memory transfers should use a fixed block size if memory conflict is to be minimized.

Index terms--Memory Interference, Multiprocessors, Memory Bandwidth, Performance Evaluation, Markov Chains.

1. Introduction

This paper develops two discrete time models of the memory interference that occurs during memory access in a multiprocessor system. These models, termed the equivalent rate (ER) model and the Markov chain (MC) model, are based on a set of assumptions that characterize the memory access behavior of a multiprocessor system as a stochastic process. Figure 1 illustrates the type of system dealt with by the models. It shows a synchronous multiprocessor having N processors and M memory modules. The processors share the memory modules through an $N \times M$ crossbar interconnection network. The whole system is synchronized with a system clock whose period is referred to as "the system cycle." Previous discrete time memory interference models assume that once a connection is established between a processor and a memory module it is maintained for a fixed number of system cycles (usually one). The models developed in this paper consider the case where the connection may be maintained for a variable number of system cycles. In particular, the connection time is represented by a discrete random variable, X , with a probability mass function $f(i)$ ¹. This extension allows the modeling of processor-memory transactions that consist of variable length packets. Those cases where the processor-memory connection time is a fixed number of system cycles will be referred to as fixed connection time (FCT) systems. While, those that allow variations in the number of system cycles will be referred to as variable connection time (VCT) systems.

The literature contains a number of memory interference models for FCT systems (see [1]-[11]). In these studies system operation is approximated by a stochastic process as follows. At the beginning of the system cycle a processor selects a memory module at random and makes a request to access that module with probability r (≤ 1). If more than one request is made to the same memory module, it will choose one at random; the other processors will retry in the next cycle. A processor has at most one pending request waiting for access at any time. The behavior of the processors are considered to be independent but statistically identical. A processor that obtains a connection to a memory module at the beginning of the system cycle will release that

¹ In other words, $f(i) = Pr[X = i]$.

module at the end of the cycle.

A model for the FCT system described above results in a Markov chain having an unmanageably large state space, see [1] and [4]. One of the main themes of the work in [1]-[11] is to develop models which avoid this complexity while maintaining reasonable agreement with simulation results. This is done by further simplifying the assumptions of the system behavior. The models develop equations for the memory bandwidth, BW , the probability that a memory request is accepted, P_a , and in some cases processor utilization, U_p . In [10] a classification has been proposed for these models according to the approach used in their formulation. The classes are probabilistic models, rate-adjusted probabilistic models, queueing system models and steady state flow models. Since we make use of some of these models, and to provide further background, the classification is summarized briefly below.

The first class—probabilistic models—simplifies the analysis by assuming that a memory request from a processor will be discarded if it is denied memory access as a result of memory interference. At the next system cycle a new and unrelated request will be made by the processor

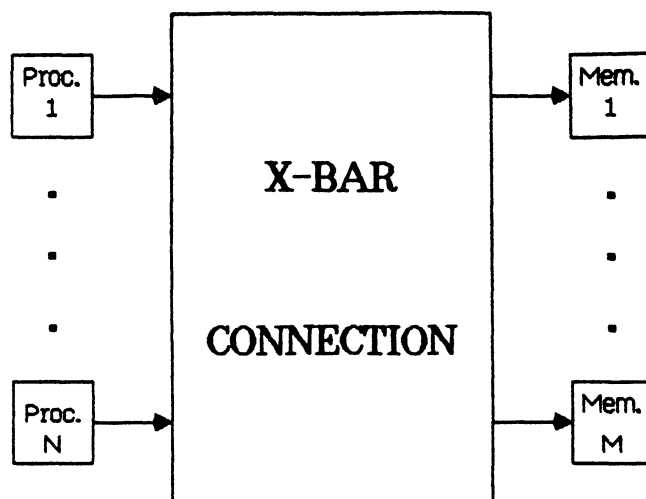


Figure 1. Block diagram of a multiprocessor system.

with probability r . Examples are reported in [2], [3], [7]-[9], [11] and [12]. In this class of models the bandwidth is given by:

$$BW = M \left[1 - \left(1 - r/M \right)^N \right] \quad (1)$$

The second class—rate-adjusted probabilistic models—is proposed in [5] and [6]. This class simplifies the analysis by assuming that a processor which is denied memory access will make a new request, not necessarily to the same memory module, with probability 1.0 in the next cycle; this assumption tends to give an upper bound for the memory bandwidth. The bandwidth is obtained by calculating the adjusted rate, α , from the following two equations using iteration:

$$P_a = \frac{M}{N\alpha} \left[1 - \left(1 - \alpha/M \right)^N \right] \quad (2a)$$

And:

$$\alpha = \frac{1}{1 + P_a \left(1/r - 1 \right)} \quad (2b)$$

The bandwidth is then obtained from:

$$BW = M \left[1 - \left(1 - \alpha/M \right)^N \right] \quad (2c)$$

Notice the "rate," r , in equation (1) is replaced by the "adjusted rate," α . This takes into account the fact that, in general, more than one request is made before the connection is established, i.e., $0 \leq r \leq \alpha \leq 1.0$.

The third class—queueing system models—views the memory modules as service stations and processor requests as customers. In [4] a binomial approximation is used to solve these queueing systems. The assumption in this class is that arrivals to the queue are binomially distributed. The resulting bandwidth is given by the following equation:

$$BW = \frac{M}{2} \left[2 + 2L - \frac{1}{M} - \sqrt{\left(2 + 2L - 1/M \right)^2 - 8L} \right] \quad (3a)$$

Where, L , the approximate mean queue length is given by:

$$L = \frac{-\left(\frac{1}{r} - \frac{N-1}{M}\right) + \sqrt{\left(\frac{1}{r} - \frac{N-1}{M}\right)^2 + 4\left(\frac{N-1}{M}\right)}}{2\left(\frac{N-1}{N}\right)} \quad (3b)$$

The expression for L is obtained by approximating the length of the queue seen by an arriving customer to $\left(1 - \frac{1}{N}\right)L$. This linear approximation usually gives good result for highly utilized system but works poorly for system with low r . An improvement on this approximation is given in [7] for the case $r = 1.0$. The improvement relies on a decomposition approximation suggested in [4].

A fourth class—steady state flow models—is proposed in [10]. The idea behind this proposal is that the flow of active processors' requests to the memory modules will equal the flow of satisfied requests from the memory modules. An active processor is a processor that does not have a pending request. The bandwidth obtained from the flow model is given by the following two equations:

$$BW = N U_p r \quad (4a)$$

And :

$$BW = M \left[1 - \left(1 - \frac{U_p r}{M} \right)^N \left(1 - \frac{1}{M} \left(1 - \left(1 - \frac{1-U_p}{M} \right)^N \right) \right)^M \right] \quad (4b)$$

The above equations can be solved by iteration. In [10] it is shown that the steady state flow model has a smaller maximum error and a smaller average mean-square error than the other models over the whole range of r , i.e., $0 \leq r \leq 1$.

In addition to the above discrete time models, continuous time models have been proposed. These are typified by the models in [13] and [14], in which the the memory-processor connection time is denoted by an exponentially distributed random variable and the processor interrequest time is also denoted by an exponentially distributed random variable. Therefore, unlike the discrete time FCT models, these models can accommodate variable connection times provided the

times can be approximated by an exponentially distributed random variable². This is an important step in modeling VCT systems. However, continuous time models are usually less accurate than discrete time models when discrete time events are being modeled [16], [17]. (The trade-off being that the continuous time models use the memoryless property of the exponential distribution to simplify model development.) Furthermore, the restriction of approximating connection times as exponentially distributed random variables does not allow one to gauge accurately or to gain insight into the effect of situations where this approximation does not hold.

As stated earlier, this paper develops two discrete time VCT models. The connection time is modeled as a discrete random variable, X , having a probability mass function $f(i)$. There are no restrictions on $f(i)$; for example, $f(i)$ need not be a geometric distribution (the discrete time analogue of an exponential distribution). This freedom to specify $f(i)$ is important, because, as we will show, the memory interference behavior is highly dependent on, C_v , the coefficient of variation³ of the random variable X . In particular, it is shown that increasing C_v reduces, BW , P_a and U_p . Thus, for example, caching schemes that employ variable block size transfers will experience greater memory conflict than schemes employing fixed size blocks, all other things being equal.

The paper is organized as follows: Section 2 will describe the assumptions that characterize system operation; Section 3 will define the performance measures that can be obtain from the models; Section 4 will develop the two analytic models, i.e., the ER and MC models, which are used to approximate the performance measures of VCT systems; Section 5 will present the simulation results and compare them to the models' results; Section 6 will present some concluding remarks.

² These models also considered the case, not considered here, of multiple-bus interconnections rather than a crossbar. Until recently no simple discrete time model existed for this [15].

³ $C_v = \frac{\text{standard deviation of } X}{\text{expected value of } X} = \sqrt{\frac{\bar{X}^2}{(\bar{X})^2}} - 1$ where \bar{X} and \bar{X}^2 are the first and second moments of $f(i)$ respectively.

2. The System Operation Assumptions

A processor in the system depicted in Figure 1 can be in any of three states: *thinking*, when it has no outstanding request to memory (it might be performing local processing); *accessing*, when it is connected to a memory module; and *waiting*, when it has a pending memory request waiting to be serviced. The memory module can be in one of two states: *busy*, when it is being accessed by a processor, and *idle*, when it is not being accessed. The following assumptions further characterize the operation of the system:

- I. Processors' requests for memory form independent, statistically identical stochastic processes.
- II. At the beginning of a system cycle a processor in the thinking state or that has just completed a memory access makes a request to access a memory module with probability r .
- III. If more than one processor issues a request to a particular memory module, that memory will choose one at random to get the connection. The other processors will retry to the same memory module in the next cycle.
- IV. The requests originating from the same processor are independent of each others. In each case a memory module will be chosen at random from the M memory modules with equal probability, i.e., with probability $1/M$.
- V. The connection time, in units of system cycles, between a processor and a memory module is determined by a discrete random variable X , which has a probability mass function $f(i)$.

Assumptions I through IV are common to all the discrete FCT models mentioned earlier. Empirical evidence to support their adoption can be found in references [3]-[5] among others. Assumption V is the key difference between the FCT and VCT models; thus, the models developed in this paper depend on the extent to which connection times can be approximated by a random variable, X , having a probability mass function $f(i)$. From assumption V it can be seen that the FCT case is just special case of the VCT case where $X = 1$ with probability one.

In order to derive numerical information from the memory interference models developed later, the values of M , N , r and the first two moments of $f(i)$ must be obtained through measurement or, if it is considered satisfactory, by hypothesis. These quantities can be regarded as the inputs to the models.

3. Performance Measures

The performance measures that will be derived from our VCT models are: memory bandwidth, BW , the probability that a memory request is accepted, P_a , and processor utilization, U_p . These quantities can be regarded as the outputs of the models. The memory bandwidth is defined as the expected number of busy memory modules seen by a random arrival after the system reaches its steady state. Equivalently, it can be defined as the expected number of accessing processors seen by a random arrival after the system reaches its steady state. It can be shown that the Markov chain which describes the behavior of a system defined by assumptions I through V is ergodic. This implies that the limiting probability of a state equals the probability that a random arrival sees the system in that state, see [18] and [19]. Thus, the memory bandwidth can be expressed as follows:

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

Since the processors are independent and statistically identical (assumption I), the above equation can be restated as follows:

$$BW = N \lim_{t \rightarrow \infty} Pr\{ \text{a processor is accessing at time } t \}$$

The probability that a memory request is accepted, P_a , is defined as follows:

$$P_a = Pr\{ \text{a proc. obtains access to a memory} \mid \text{the proc. requests that memory} \}$$

Finally, processor utilization, U_p , can be defined as the fraction of time a processor spends thinking or accessing a memory after the system has reached steady state. However, system ergodicity also implies that the limiting probability of a state equals the fraction of time the system spends in that state when it has reached steady state. Thus, U_p can be expressed as follows:

$$U_p = 1 - \lim_{t \rightarrow \infty} Pr\{ \text{a processor is waiting at time } t \}$$

4. Memory Interference Models

4.1. Equivalent Rate Model (ER Model)

This model is a modification of an approach first presented in [20] to study the behavior of multiprocessor systems in which each processor has a private cache memory. The processor-memory connections were assumed to be a fixed number of system cycles—the time needed to transfer one line. An FCT probabilistic model (equation (1)) was used to compute BW , and an equivalent value for r (the equivalent rate) was derived from the fixed number of system cycles needed to transfer a line and the probability of requesting a line transfer. In our ER model the foregoing approach is retained; the main differences are, firstly, that the equivalent rate is modified to take into account the fact that processor-memory transfers take a variable number of system cycles and, secondly, that the probabilistic model of equation (1) is replaced by the steady state flow model of equation (4) because it produces a smaller error. The ER model can be regarded as a technique for mapping a VCT system into an equivalent FCT model by appropriately defining an equivalent value for r in the FCT model.

The derivation of the equivalent rate, r_{eq} , proceeds as follows. The average connection time in units of system cycles can be expressed by:

$$\bar{X} = \sum_{i=1}^S i f(i)$$

i.e., the first moment of the random variable X (S is the maximum connection time). From assumption II the thinking time is geometrically distributed. Therefore, if \bar{T} is the average thinking time, it follows that:

$$\bar{T} = \frac{1 - r}{r}$$

The equivalent rate can now be defined by:

$$r_{eq} = \frac{\bar{X}}{\bar{X} + \bar{T}}$$

This is the fraction of time a processor in a VCT system is accessing memory assuming no interference. Thus, it can be viewed as the request rate of an equivalent FCT system (the request

rate is obtained from an interference free trace of processor activity [5]). As will be shown, the ER model captures the average behavior of VCT systems only if C_v is zero. It introduces inaccuracies in those cases where there is randomness in the connection time, because it does not take into account the second moment of the connection time. However, it cannot be improved upon if the only information about $f(i)$ is its first moment.

4.2. Markov Chain Model (MC Model)

The Markov chain which models a VCT system according to the assumptions outlined in Section 2 has an unmanageably large state space, see [1] and [4]. The MC model dramatically reduces the size of this state space by making a further simplifying assumption. The Markov chain in the MC model describes the behavior of one processor. Within the framework of the assumptions this is sufficient to describe the complete system behavior because all processors are assumed independent and statistically identical (assumption I). The essence of the simplification is similar to that used in the rate adjusted probabilistic models (see [5] and [6]); when a processor is blocked in an attempt to access a memory module after placing a request, it enters a series of waiting states for the residual service time of that module. The *residual service time* (RST) of a memory module is the time remaining before the currently accessing processor releases the memory module. The model assumes that the processor waits for the RST before placing a new and independent request with a probability of 1.0. The request is directed to any memory module independent of the particular memory it was previously blocked at. This assumption, IIIa, is a relaxation of assumption III which requires that resubmission be to the same memory. It simplifies the Markov chain but usually causes an overestimate of the memory bandwidth of the VCT system being modeled.

The Markov chain, shown in Figure 2, defines a processor's behavior if assumption IIIa is used. When a processor is in state i it is accessing a memory module and it needs i more cycles before releasing the connection. After one cycle in state i the processor always moves to state $i-1$, indicating it needs $i-1$ more cycles. Thus, there is a single transition from state i ($1 < i \leq S$) to state $i-1$ with a probability of 1.0. The set of states $\{i\}$ are accessing states. When a processor is

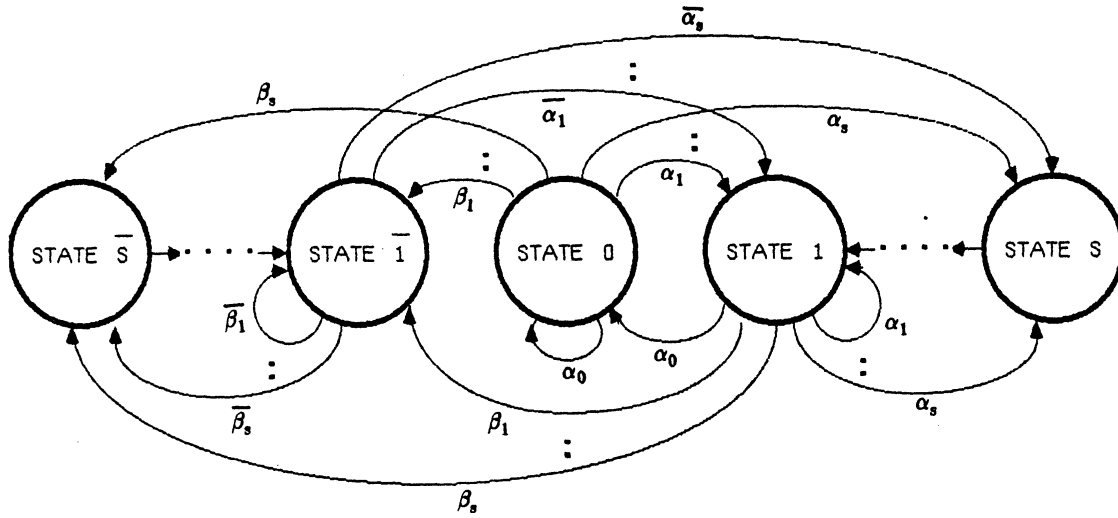


Figure 2. The Markov chain for a VCT processor (assumption IIIa).

in state 0 it is in the thinking state and it may be performing local processing. When a processor is in state \bar{i} it had a memory request blocked and must wait i cycles before it can resubmit the request. After one cycle in state \bar{i} the processor always moves to state $\overline{i-1}$, indicating it needs $i-1$ more cycles. Thus, there is a single transition from state \bar{i} ($1 < i \leq S$) to state $\overline{i-1}$ with a probability of 1.0. The set of states $\{\bar{i}\}$ are waiting states. The transition probabilities α_i , β_i , $\bar{\alpha}_i$ and $\bar{\beta}_i$ are defined by the following equations :

$$\alpha_i = \begin{cases} 1-r & i=0 \\ Pr[\text{a request is made and accepted and needs } i \text{ cycles}] & 1 \leq i \leq S \end{cases}$$

$$\beta_i = Pr[\text{a request is made to a memory that has an RST of } i \text{ cycles}] \quad 1 \leq i \leq S$$

$$\bar{\alpha}_i = Pr[\text{a request is accepted and needs } i \text{ cycles}] = \frac{\alpha_i}{r} \quad 1 \leq i \leq S$$

$$\bar{\beta}_i = Pr[\text{a memory has an RST of } i \text{ cycles} | \text{a request was made to it}] = \frac{\beta_i}{r} \quad 1 \leq i \leq S$$

If the processor is in the thinking state, i.e., state 0, one of three possibilities can occur at the beginning of the next system cycle: the processor continues in the thinking state with probability α_0 ($=1-r$); the processor accesses a memory module for the next i cycles with probability

α_i , i.e., the processor enters state i ; or the processor is blocked for the next i cycles with probability β_i , i.e., the processor enters state \bar{i} . The same three possibilities can occur if the processor is about to end its connection period at the beginning of the next system cycle, i.e., when the processor is in state 1. On the other hand, if the processor is at the end of its waiting period, i.e., it is in state $\bar{1}$, and since assumption IIIa requires the resubmission of blocked requests with a probability of one, only one of two possibilities can occur at the beginning of the next system cycle: the processor gains access to a memory module for the next i cycles with probability $\bar{\alpha}_i$, i.e., the processor enters state i ; or the processor is again blocked and must wait for the next i cycles with probability $\bar{\beta}_i$, i.e., the processor enters state \bar{i} . There is no transition from state $\bar{1}$ to the thinking state, i.e., state 0, because of assumption IIIa.

The following definition will be useful in the remainder of this section:

$$\begin{aligned} R &\triangleq \text{Pr}[\text{a request is made at the beginning of any system cycle}] \\ &= r (P_1 + P_0) + P_{\bar{1}} \end{aligned}$$

where P_i is the limiting probability for state i . One important distinction should be noted: R is the probability that a processor makes a request at the beginning of any system cycle, while r ($\leq R$) is the probability that a processor makes a request at the beginning of any system cycle given that the processor is thinking or has just finished accessing (assumption II).

4.2.1. FCT Models as a Special Case of the MC Model

If the MC model is to have validity it should agree closely an FCT model when X is deterministic with probability mass function defined as:

$$f(i) = \begin{cases} 1 & \text{if } i = 1 \\ 0 & \text{otherwise} \end{cases}$$

In this subsection it is shown that an FCT model, the rate-adjusted probabilistic model, is a proper special case of the MC model.

The Markov chain for the case, where the probability mass function is given above, is shown in Figure 3. In order to develop expression for the transition probabilities it is first necessary to develop an expression for, P_a , the probability that a request is accepted (see definition in Section

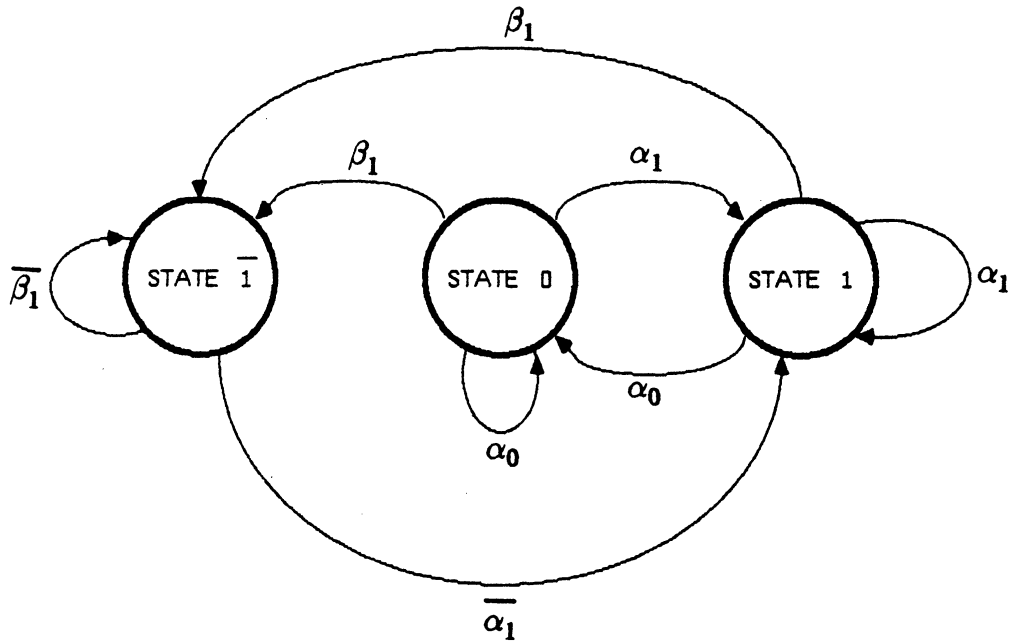


Figure 3. The Markov chain for an FCT processor (assumption IIIa).

3). The development proceeds as follows. The probability that a request is made at the beginning of a system cycle is given by R , and the probability that this request is directed to a specific memory is R/M (assumption IV). It follows that the probability that the request is not directed to the memory is $(1 - R/M)$, and further, that the probability that no request is directed to the memory from any processor is $(1 - R/M)^N$. Thus, the probability that at least one processor makes a request for the memory is given by $1 - (1 - R/M)^N$. The expected number of busy memories is then given by $M[1 - (1 - R/M)^N]$. However, the expected number of request made by all of the processors is NR , therefore the fraction that are accepted, i.e., P_a , is given by:

$$P_a = \frac{M}{NR} [1 - (1 - R/M)^N] \quad (5)$$

The transition probabilities can now be expressed as follows:

$$\begin{aligned}
\alpha_0 &= 1 - r \\
\alpha_1 &= r P_a f(1) = r P_a \\
\beta_1 &= r - \alpha_1 = r(1 - P_a) \\
\bar{\alpha}_1 &= \frac{\alpha_1}{r} = P_a \\
\bar{\beta}_1 &= \frac{\beta_1}{r} = 1 - P_a
\end{aligned}$$

The limiting probabilities for the states of Figure 3 satisfy the following equations:

$$\begin{aligned}
P_1 &= \alpha_1 P_0 + \alpha_1 P_1 + \bar{\alpha}_1 P_{\bar{1}} \\
P_0 &= \alpha_0 P_0 + \alpha_0 P_1 \\
P_{\bar{1}} &= \beta_1 P_0 + \beta_1 P_1 + \bar{\beta}_1 P_{\bar{1}}
\end{aligned}$$

And:

$$P_1 + P_0 + P_{\bar{1}} = 1$$

These simultaneous equations together with the earlier definition for R yield the following equation:

$$R = \frac{1}{1 + P_a (1/r - 1)} \quad (6)$$

Equations (5) and (6) can be solved using iteration. From the definition of BW in Section 3 it can be shown that:

$$BW = N P_1 = M \left[1 - (1 - R/M)^N \right] \quad (7)$$

Similarly, it can be shown that:

$$U_p = 1 - P_{\bar{1}} = 1 - (1 - P_a) R$$

Equations (5), (6) and (7) are isomorphic to equations (2a, b, c) with α replaced by R as the adjusted rate. Thus, the Markov chain shown in Figure 3 is equivalent to the rate-adjusted probabilistic model of [5], and thus the rate-adjusted probabilistic model is a proper special case of the MC model.

Before concluding this subsection, it is worth noting that a Markov chain model, similar to the MC model of Figure 2, can be developed which assumes that the processor discards the unaccepted request altogether and, after waiting for the RST of the memory module, makes a new request with a probability of r . This assumption, IIIb, is another relaxation of assumption III that

also simplifies the model, but is less realistic than IIIa. The Markov chain describing the processor's behavior under assumption IIIb is shown in Figure 4. Unlike the MC model of Figure 2 there is a transition from state $\bar{1}$ to the thinking state. A solution to the Markov chain of Figure 4 can be found in a similar fashion to that presented later for the MC model. The similarity between the two models becomes apparent as $r \rightarrow 1$. They are identical when $r = 1.0$. Assumption IIIb is used by the probabilistic models developed in [2], [3], [7]-[9] and [11]. For the case where the probability mass function is again given by:

$$f(i) = \begin{cases} 1 & \text{if } i = 1 \\ 0 & \text{otherwise} \end{cases}$$

Figure 5 results. Under assumption IIIb the expression for R changes, since resubmission in state $P_{\bar{1}}$ is now made with probability r rather than 1.0. Therefore:

$$R = r(P_1 + P_0 + P_{\bar{1}}) = r$$

And:

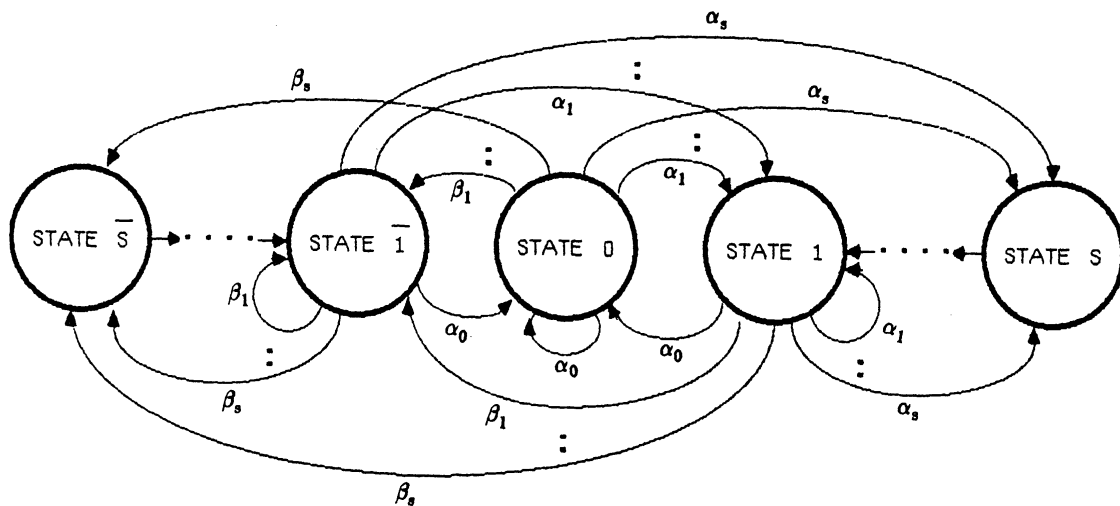


Figure 4. The Markov chain for a VCT processor (assumption IIIb).

$$P_a = \frac{M}{Nr} \left[1 - (1 - r/M)^N \right]$$

The branching probabilities for the Markov chain shown in Figure 5 are as follows:

$$\alpha_0 = 1 - r ; \alpha_1 = r P_a ; \beta_1 = r - \alpha_1$$

The limiting probabilities are:

$$P_1 = \alpha_1 ; P_0 = \alpha_0 ; P_{\bar{1}} = \beta_1$$

And the memory bandwidth for the system can be expressed as :

$$BW = N P_1 = M \left[1 - (1 - r/M)^N \right]$$

This equation agrees with the bandwidth equation of the most simple memory interference model--the probabilistic models of equation (1).

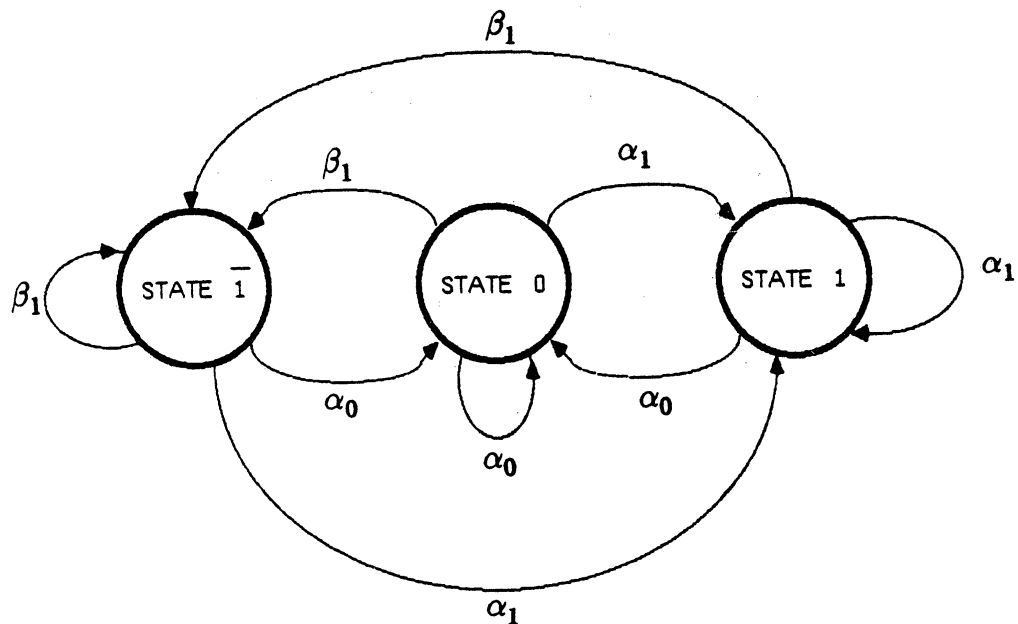


Figure 5. The Markov chain for an FCT processor (assumption IIIb).

4.2.2. General MC Model

In this subsection the equations for the general MC model are developed. The following definitions will be used:

$$\begin{aligned} B &\triangleq \text{Pr}[a \text{ memory is busy at the end of a system cycle}] \\ &= \text{Pr}[a \text{ processor is still accessing at the end of a system cycle}] \\ &= \sum_{i=2}^S P_i \end{aligned}$$

$$\begin{aligned} B_i &\triangleq \text{Pr}[\text{processor } i \text{ requests a memory that is busy}] \\ &= \sum_{\substack{j=1 \\ j \neq i}}^N \text{Pr}[\text{processor } j \text{ is still accessing a memory at the end of a system cycle}] \\ &= \sum_{\substack{j=1 \\ j \neq i}}^N \frac{B}{M} = \frac{(N-1)B}{M} \end{aligned}$$

It follows from assumption I that B_i is independent of i , thus we can define:

$$B' \triangleq B_i \quad 1 \leq i \leq N$$

From these definitions the following terms can be developed. These terms are used to derive the transition probabilities of the MC model. They are presented here by way of explanation:

$$(1 - P_a)f(i) = \text{Pr}[a \text{ memory request is blocked and the proc. which obtains the memory and blocks the request needs } i \text{ cycles}]$$

$$1 - B' = \text{Pr}[a \text{ processor requests a memory that is idle}]$$

$$\frac{P_{i+1}}{B} = \text{Pr}[a \text{ memory seen by a proc. has an RST of } i \text{ cycles} | \text{the memory is busy}]$$

The transition probabilities for the Markov chain of Figure 2 can now be expressed as follows:

$$\begin{aligned} \alpha_0 &= 1 - r \\ \alpha_i &= r P_a (1 - B') f(i) & 1 \leq i \leq S \\ \beta_i &= r \left[B' \frac{P_{i+1}}{B} + (1 - B') (1 - P_a) f(i) \right] & 1 \leq i \leq S-1 \\ \beta_S &= r (1 - B') (1 - P_a) f(S) \\ \bar{\alpha}_i &= \frac{\alpha_i}{r} & 1 \leq i \leq S \\ \bar{\beta}_i &= \frac{\beta_i}{r} & 1 \leq i \leq S \end{aligned} \tag{8}$$

The limiting probabilities for the states can be written as follows:

$$\begin{aligned}
P_i &= \left(\sum_{j=i}^S \alpha_j \right) \frac{R}{r} & 1 \leq i \leq S \\
P_0 &= \frac{\alpha_0}{(1-\alpha_0)} \left(\sum_{i=1}^S \alpha_i \right) \frac{R}{r} \\
P_i &= \left(\sum_{j=i}^S \beta_j \right) \frac{R}{r} & 1 \leq i \leq S
\end{aligned} \tag{9}$$

Since the above $2S+1$ equations add to one, it follows that:

$$R = \frac{r}{\sum_{i=1}^S i (\alpha_i + \beta_i) + \frac{\alpha_0}{1-\alpha_0} \sum_{i=1}^S \alpha_i} \tag{10}$$

From the definitions of B' , B and equation (8), B can be expressed as follows:

$$B = \sum_{i=2}^S P_i = \left(\sum_{i=1}^S (i-1) \alpha_i \right) \frac{R}{r}$$

Substituting in the above equation for α_i from equation (8) allows B to be expressed as the following function of R :

$$B = \frac{(\bar{X} - 1)P_a R}{1 + \frac{N-1}{M} (\bar{X} - 1)P_a R} \tag{11}$$

Substituting equations (8) and (9) into (10) results in the following equation:

$$R = \frac{1}{\left(1 - \frac{N-1}{M} B \right) \left[\bar{X} + (1/r - 1)P_a + \frac{(N-1)P_a R}{M} \sum_{i=1}^S \frac{i(i-1)}{2} f(i) \right]}$$

By defining the second moment of the connection time distribution in the normal way, i.e.,

$\bar{X}^2 = \sum_{i=1}^S i^2 f(i)$, the above equation can be written:

$$R = \frac{1}{\left(1 - \frac{N-1}{M} B \right) \left[\bar{X} + (1/r - 1)P_a + \frac{(N-1)P_a R}{M} \frac{(\bar{X}^2 - \bar{X})}{2} \right]} \tag{12}$$

The equations for P_a (equation (5)), for B (equation (11)) and the above equation form the MC model. They can be solved by iteration. In the experiments reported in the next section a fixed-point iteration on the value of R was used. Solution typically required 4 iterations (with a maximum of 8) when an initial value of $R = r$ was used. Higher order iterations schemes could be

used but were found unnecessary within the scope of our work. The value of P_a falls out directly from the above solution method. The value for memory bandwidth, BW , and processor utilization, U_p , can be calculated from the following:

$$BW = N (P_1 + B)$$

$$U_p = 1 - \sum_{i=1}^S P_i$$

These equations follow from the definitions of Section 3. It can be seen from equation (12) that R depends, among other things, on the inverse of \bar{X}^2 . Therefore, it follows from the above equations, that both BW and U_p depend on the inverse of \bar{X}^2 . Furthermore, it can also be seen from equation (12) that R is independent of S . Thus the underlying Markov chain of the MC model need not be finite.

5. Simulation Results

A simulation, written in SIMSCRIPT II.5, of VCT systems operating according to the assumptions given in Section 2 was run for different probability mass functions, $f(i)$, and different values of r . The results were compared to results calculated from the ER model and the MC model. The comparisons, which are shown in Figures 6, 7 and 8, were made for three systems sized 8×8 , 16×16 and 32×32 respectively. The figures compare BW . The %Error shown in the figures was defined as:

$$\%Error = \frac{Model\ BW - Simulated\ BW}{Simulated\ BW} \times 100$$

Six different distributions for the connection time were used. All the distributions had the same expected value, \bar{X} , but their coefficient of variation, C_v , ranged from 0.0 (i.e. fixed connection time) to 2.0. As can be seen from the figures, both models gave results within 4% of the simulation for small C_v . The MC model remained within this error bound, but the ER model showed a monotonic increase in error with increase in C_v . In the case of $C_v = 2.0$ the error was as large as 50%. Similar results are obtained if P_a or U_p are compared.

The poor performance of the ER model, which continues to worsen as C_v increases beyond $C_v = 2.0$, confirms the importance of using the second moment of the connection time distribution in calculating BW , see equation (12). The error in the MC model is due to assumption IIIa being used in place of assumption III. As noted, the simulation works according to the assumptions of Section 2, in particular assumption III. The key difference between assumption IIIa and III is that in IIIa blocked requests for memory need not be resubmitted to the memory from which they were previously blocked. This is clearly unrealistic, but our experimental evidence indicates that the effect on the quantities BW , P_a and U_p is quite small--less than 4% error in all cases. Furthermore, as mentioned earlier, the relaxation of assumption III to that of IIIa makes possible a manageable model. Finally, the error is comparable with the empirical evidence reported in earlier work [3]-[5] that led to the assumptions of Section 2 as a phenomenological basis for the behavior of a large class of multiprocessors of the type shown in Figure 1.

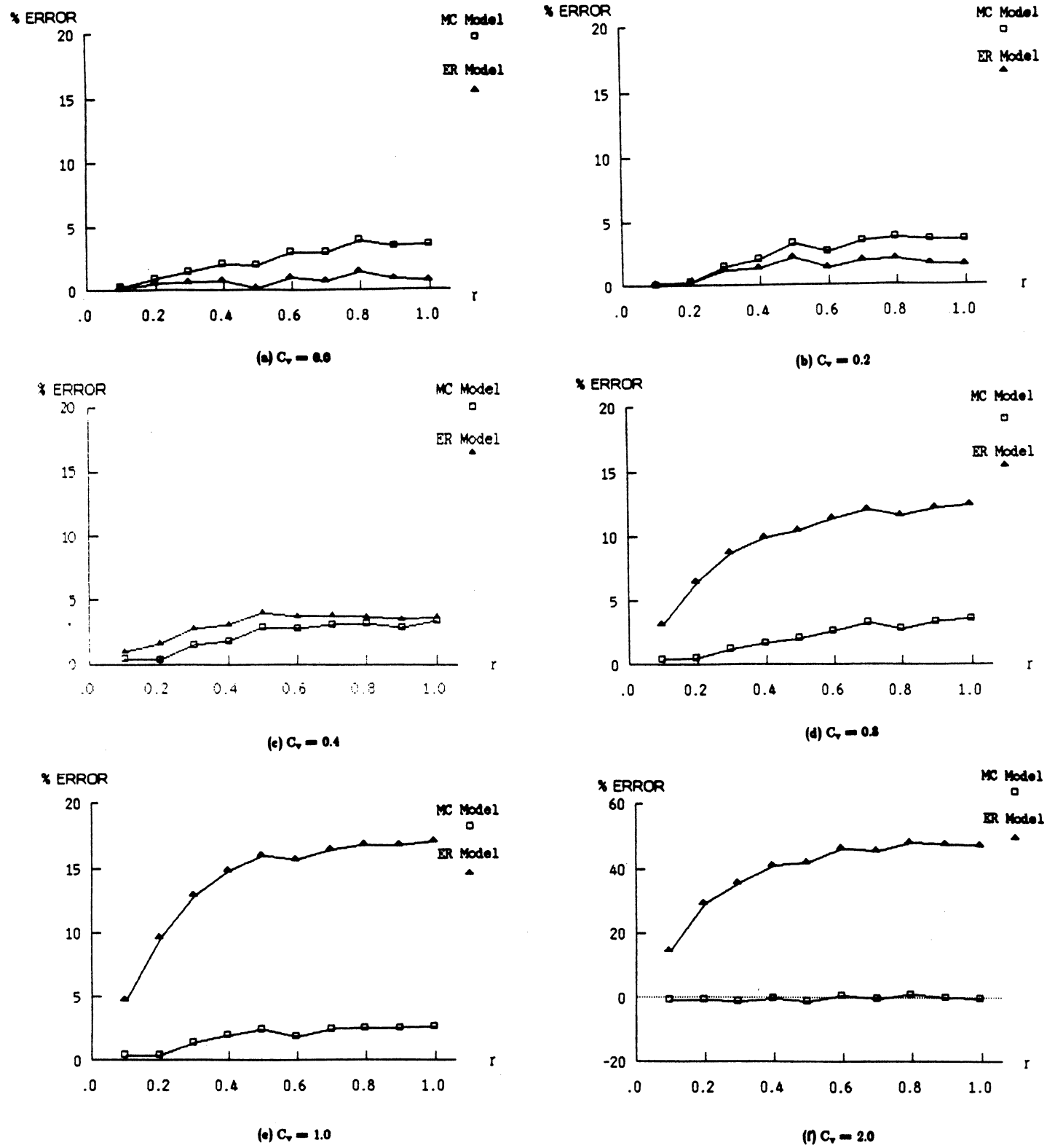
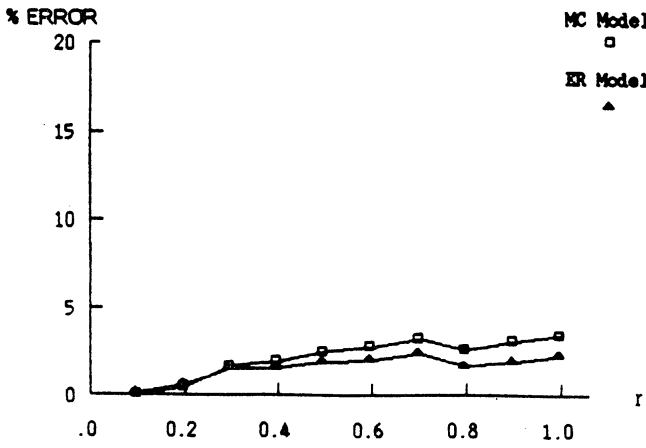
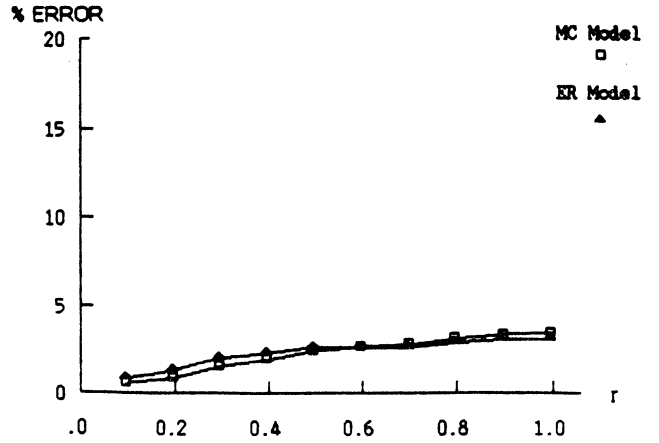


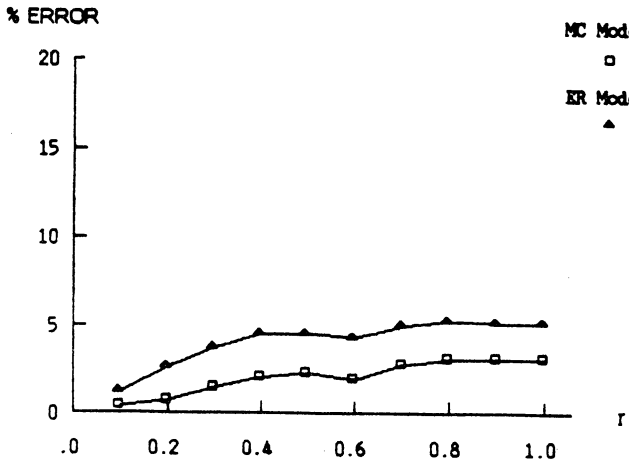
Figure 6. Comparisons with simulation results for an 8x8 system.



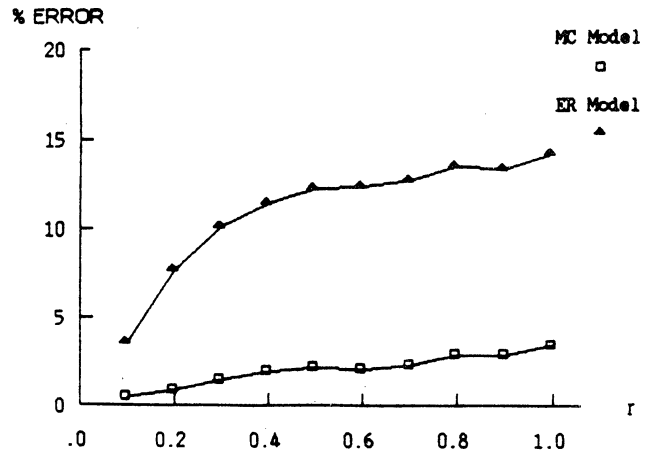
(a) $C_v = 0.0$



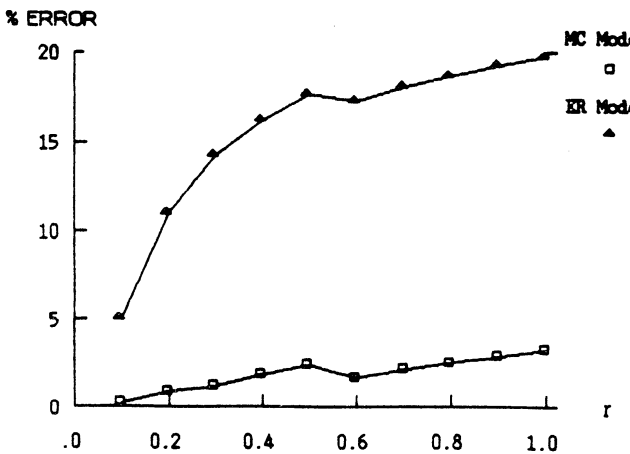
(b) $C_v = 0.2$



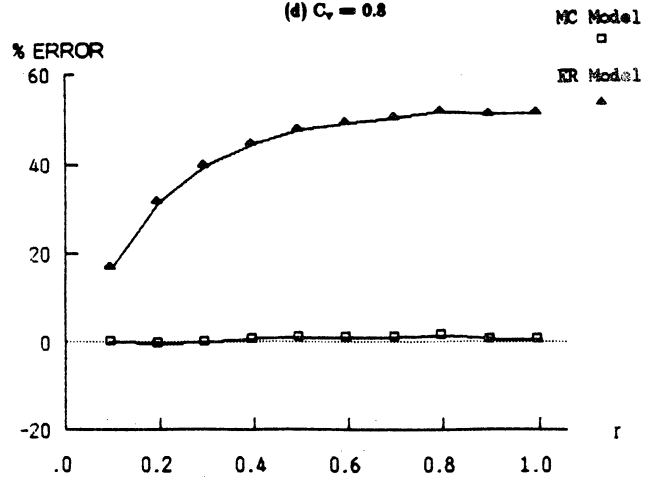
(c) $C_v = 0.4$



(d) $C_v = 0.8$

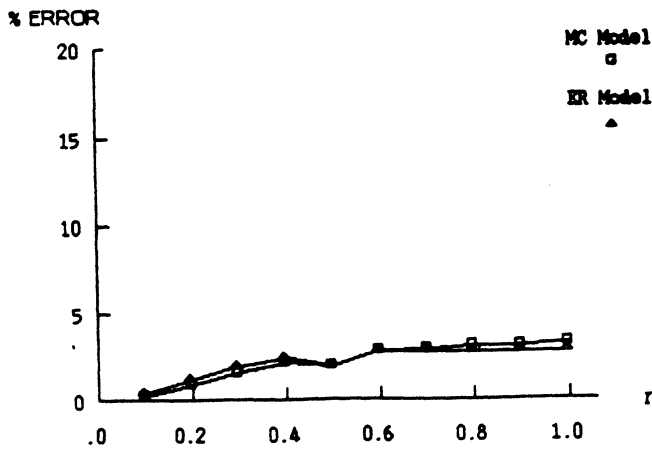


(e) $C_v = 1.0$

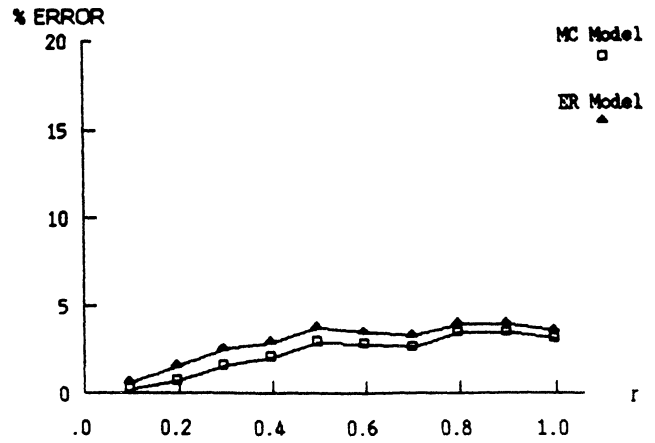


(f) $C_v = 2.0$

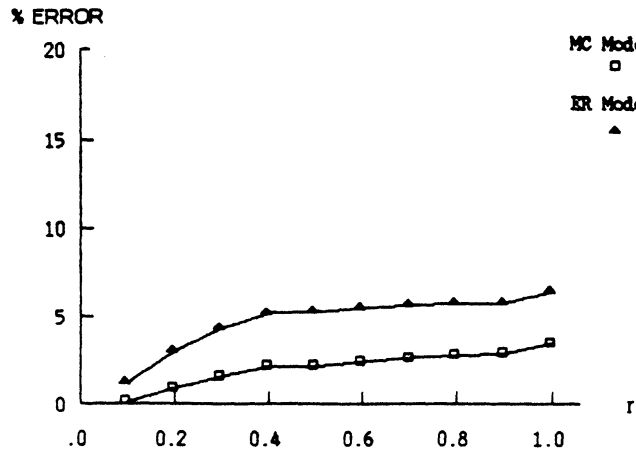
Figure 7. Comparisons with simulation results for a 16x16 system.



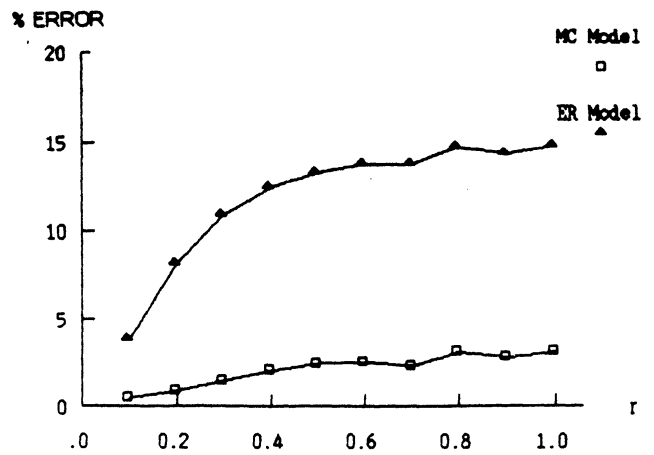
(a) $C_v = 0.0$



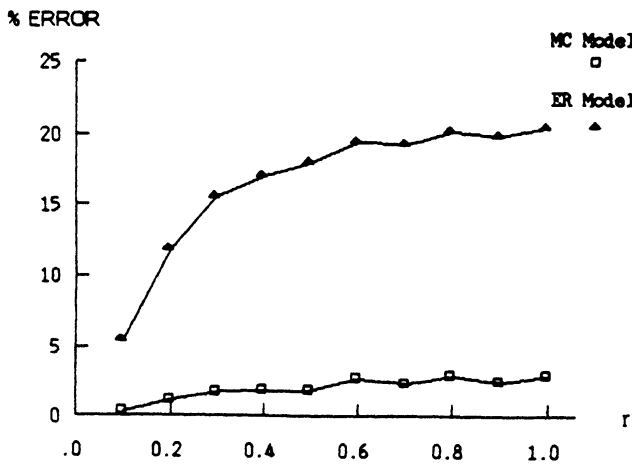
(b) $C_v = 0.2$



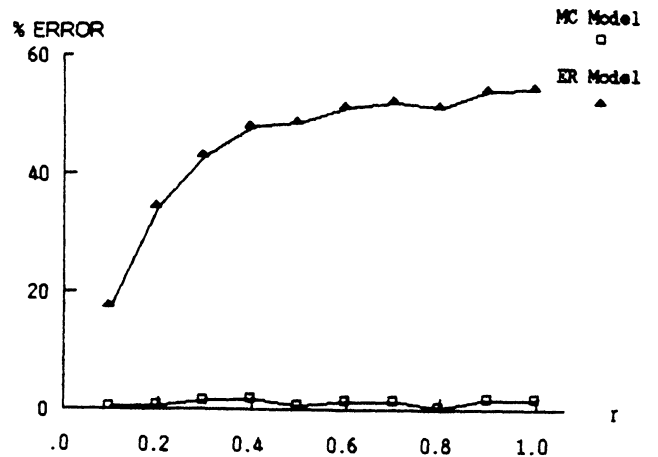
(c) $C_v = 0.4$



(d) $C_v = 0.8$



(e) $C_v = 1.0$



(f) $C_v = 2.0$

Figure 8. Comparisons with simulation results for a 32x32 system.

Figure 9 shows explicitly how BW varies with C_v while \bar{X} is held constant. As can be seen, by just varying C_v from 0 to 2.0 the BW can drop by as much as 40% for high request rates ($r \approx 1$). In fact, in the case of 32×32 system with $C_v = 2.0$ the BW drops to the point where only 13 of the 32 memory modules are busy even where $r = 1$. This agrees with the interpretation, based on equation (12), that was made at the end of the last section where it was concluded that BW would decrease if \bar{X}^2 (or C_v) increased. The most obvious consequence of BW depending on C_v in this way is that transfers between processors and memories should be restricted to fixed blocks, or nearly fixed blocks in which variations in size are rare, if maximum BW is to be achieved.

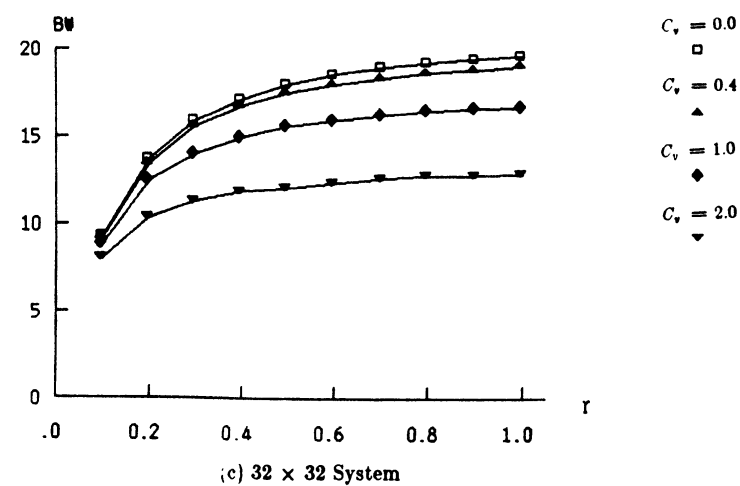
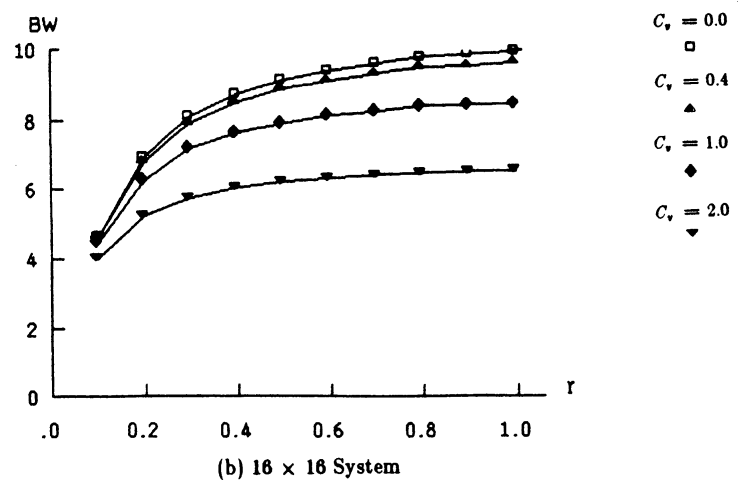
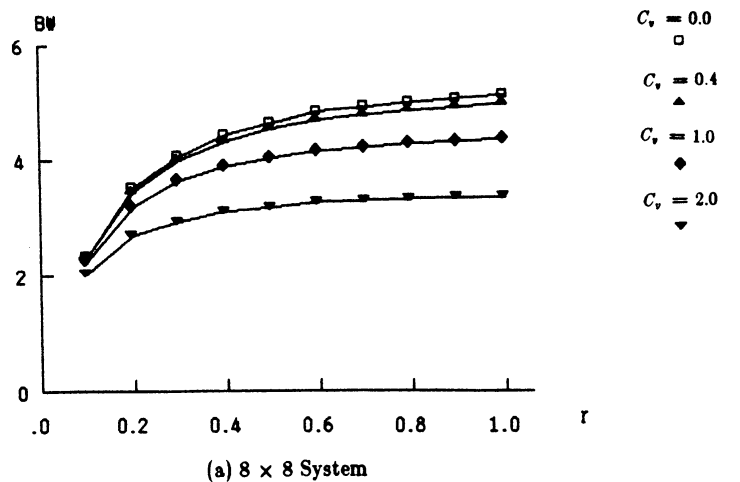


Figure 9. The results of varying C_v on BW.

6. Conclusion

This paper has developed two discrete time models of the memory interference that occurs during memory access in a multiprocessor system when that access can have a variable duration. The first of these models, the ER model, is the simpler model and, according to comparisons with simulations, provides accurate estimates of the values for P_a , BW and U_p , if C_v is small. The second of these, the MC model, is the more complex model but, according to comparisons with simulations, provides accurate estimates of the values for P_a , BW and U_p , for a wide range of C_v . The ER model requires the values of M , N , r and \bar{X} as inputs. The MC model requires, in addition, the value of \bar{X}^2 . The explicit dependence of the MC model on \bar{X}^2 (and hence C_v) can be observed in equation (12). This was confirmed empirically; specifically, it was shown that BW decreases with increase in C_v . The fact that the second moment is an important feature of memory interference should not be completely unexpected as the behavior of similar systems, e.g., networks of queues, also depend on the variance of underlying stochastic processes (in the case of queues, it is the variances of the inter-arrival time and service time).

There are a number of possibilities for future research including the following two fairly straightforward ones. The first would study the effect of relaxing assumption IV so that systems in which each processor has a preferred memory module, or in which a subset of the memory modules are used as common memory (perhaps as mailboxes), can be studied in a manner analogous to earlier FCT studies in [5], [9] and [11]. The second would attempt a synthesis of the techniques presented here with the discrete time models of multiple-bus systems developed in [15]. Analytical comparisons of circuit switched versus packet switched buses would then be possible.

Acknowledgments: The authors gratefully acknowledge comments and suggestions made by B. A. Makrucki.

7. References

- [1] C. E. Skinner and J. R. Asher, "Effects of Storage Contention on System Performance," *IBM Systems Journal*, vol. 8, no. 4, pp. 319-333, 1969.
- [2] W. D. Strecker, in *Analysis of the Instruction Execution Rate in Certain Computer Structures* Ph.D. Thesis, Carnegie-Mellon Univ., 1970.
- [3] D. P. Bhandarkar, "Analysis of Memory Interference in Multiprocessors," *IEEE Trans. on Computers*, vol. C-24, no. 9, pp. 897-908, September 1975.
- [4] F. Baskett and A. J. Smith, "Interference in Multiprocessor Computer Systems with Interleaved Memory," *Comm. of ACM*, vol. 19, no. 6, pp. 327-334, June 1976.
- [5] C. H. Hoogendoorn, "A General Model for Memory Interference in Multiprocessors," *IEEE Trans. on Computers*, vol. C-26, no. 10, pp. 998-1005, October 1977.
- [6] J. S. Emer and E. S. Davidson, "Control Store Organization for Multiple Stream Pipelined Processors," in *Proc. 1978 Int'l Conf. on Parallel Processing*, pp. 43-48, August 1978.
- [7] B. R. Rau, "Interleaved Memory Bandwidth in a Model of a Multiprocessors," *IEEE Trans. on Computers*, vol. C-28, no. 9, pp. 678-681, September 1979.
- [8] J. H. Patel, "Processor-Memory Interconnections for Multiprocessors," *IEEE Trans. on Computers*, vol. C-30, no. 10, pp. 771-780, October 1981.
- [9] T. N. Mudge and B. A. Makrucki, "Probabilistic Analysis of a Crossbar Switch," in *Proc. IEEE 9th Ann. Symp. on Computer Architecture*, pp. 311-319, April 1982.
- [10] D. W. L. Yen, J. H. Patel, and E. S. Davidson, "Memory Interference in Synchronous Multiprocessor Systems," *IEEE Trans. on Computers*, vol. C-31, no. 11, pp. 1116-1121, November 1982.
- [11] L. N. Bhuyan and C. W. Lee, "An Interference Analysis of Interconnection Networks," in *Proc. 1983 Int'l Conf. on Parallel Processing*, pp. 2-9, August 1983.
- [12] F. A. Briggs and E. S. Davidson, "Organization of Semiconductor Memories for Parallel-Pipelined Processors," *IEEE Trans. on Computers*, vol. C-26, no. 2, pp. 162-169, February 1977.
- [13] M. A. Marsan and M. Gerla, "Markov Models for Multiple Bus Multiprocessor Systems," *IEEE Trans. on Computers*, vol. C-31, no. 3, pp. 239-248, March 1982.
- [14] I. H. Onyuksel and K. B. Irani, "A Markov Queueing Network Model for Performance Evaluation of Bus-Deficient Multiprocessor Systems," in *Proc. 1983 Int'l Conf. on Parallel Processing*, pp. 437-439, August 1983.



- [15] T. N. Mudge, J. P. Hayes, G. D. Buzzard, and D. C. Winsor, "Analysis of Multiple-Bus Interconnection Networks," Computing Research Lab Report # CRL-TR-12-84, February 1984.

- [16] S. H. Fuller, "Performance Evaluation," in *Introduction to Computer Architecture*, H. S. Stone, Ed S.R.A., Inc., pp. 474-546, 1975.

- [17] D. P. Bhandarkar and S. H. Fuller, "Markov Chain Models for Analyzing Memory Interference in Multiprocessor Computer Systems," in *Proc. 1st Ann. Symp. Computer Architecture*, pp. 1-6, December 1973.

- [18] D. P. Heyman and M. J. Sobel, in *Stochastic Models in Operations Research*, vol. 1 McGraw-Hill Book Co., 1982.

- [19] E. Cinlar, in *Introduction to Stochastic Processes*. Englewood Cliffs, N.J.: Prentice-Hall Inc., 1975.

- [20] J. H. Patel, "Analysis of Multiprocessors with Private Cache Memories," *IEEE Trans. on Computers*, vol. C-31, no. 4, pp. 296-304, April 1982.