A SEMI-MARKOV MODEL FOR THE PERFORMANCE OF MULTIPLE-BUS SYSTEMS

TREVOR N. MUDGE and HUMOUD B. AL-SADOUN

Reprinted from IEEE Transactions on Computers, Vol. C-34, No. 10, October 1985

Correction

In the equation for $X(k)$ on page 937 (lefthand column) M should be replaced by $M - B + k$, to read:

$$X(k) = \sum_{i=1}^{M-B+k} \frac{\min(k,i)}{i} \binom{M - B + k - 1}{i - 1} p^{i-1}(1 - p)^{(M - B + k - i)}$$

A Semi-Markov Model for the Performance of Multiple-Bus Systems

TREVOR N. MUDGE, SENIOR MEMBER, IEEE, AND HUMOUD B. AL-SADOUN, MEMBER, IEEE

Abstract — **This paper presents a discrete time model of memory interference in multiprocessor systems employing multiple-bus interconnection networks. It differs from earlier models in its ability to model variable connection time and arbitrary interrequest time. The model describes each processing element's behavior by means of a semi-Markov process. It takes as input the number of processing elements, the number of memory modules, the number of buses, the mean think time of the processing elements, and the first and second moments of the connection time between processing elements and memories. The model produces as output the memory bandwidth, processing element utilization, memory module utilization, average queue length at a memory, and average waiting time experienced by a processing element while waiting to access a memory. Using the model, it is possible to analyze the interaction of the input parameters on the system performance. This modeling capability is attained without having to employ a complex Markov chain. In fact, a four-state semi-Markov process is sufficient regardless of the think and connection time distributions. The accuracy and capability of the model is illustrated.**

Index Terms — **Markov chains, memory bandwidth, memory interference, multiple-bus system, multiprocessors, performance evaluation, semi-Markov processes.**

I. INTRODUCTION

THERE has been an interesting variety of proposals for interconnecting processors and memories in multiprocessors [14]. This paper presents a discrete time model for one of these, the multiple-bus interconnection network. The model allows the user to quantify the effects of changing various system design parameters on such performance measures as memory bandwidth, processor utilization, memory queue length, and waiting time. A number of discrete time models for multiple-bus systems have been presented in [7], [18], [17], [3], [10], [13], [5]. The model in this paper is the first to include variable connection time and arbitrary interrequest time. The introduction of semi-Markov processes to model the processor behavior allows this increased generality without model complexity. A four-state semi-Markov process is sufficient for the model. The use of semi-Markov processes also simplifies the derivation of the memory queue length and waiting time.

Fig. 1 shows a typical multiprocessor system in which B buses are used to interconnect N processing elements with M memory modules (where $B \leq \min(N, M)$). The multiprocessor of Fig. 1 will be referred to as an $N \times M \times B$ system. The multiple-bus network has a number of desirable features. First, it is compatible with the prevailing "bus-centered" philosophy implicit in most microprocessor families of components. Second, the multiple-bus system is modular, allowing easy incremental increase in the number of processing elements, the number of memory modules, and the number of buses. Finally, the multiple-bus system is fault tolerant. For instance, if a bus fails the system still works (provided $B > 0$), albeit with degraded performance.

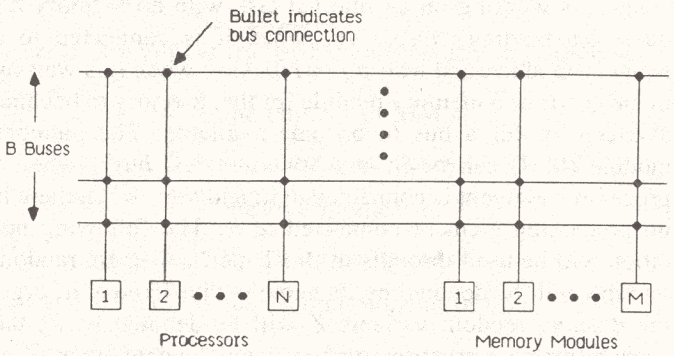


Fig. 1. A multiple-bus system.

The model presented in this paper is an extension of the technique developed in [11] where semi-Markov processes were first used to model memory interference. We adopt the terminology of [11] and refer to our model as a semi-Markov interference (SMI) model. In an $N \times M \times B$ system three types of memory interference, or memory conflict, can occur. A type 1 conflict arises when several processing elements attempt to access an idle memory module simultaneously. A type 2 conflict arises when a processing element attempts to access a busy memory module. A type 3 conflict arises when one or more processing elements attempt to access an idle memory module when no buses are available. We follow the two-stage arbitration scheme proposed by Lang *et al.* [7] to resolve access conflicts. In the first stage, the conflict over the memory modules is resolved by M arbiters of the N-users, 1-server type. Each of these arbiters selects equiprobably one of the processing elements which have outstanding requests to the arbiter's associated memory module. In the second stage, the conflict over the buses is resolved by one arbiter of the M-users, B-servers type. This arbiter assigns the memory requests selected in the first stage to the available buses. The arbiter makes the assignment in a cyclic fashion, i.e., on a round-robin basis.

The paper is organized as follows. Section II describes the assumptions that characterize the operation of the multiprocessor system; Section III develops the SMI model under

Manuscript received February 1, 1985; revised May 30, 1985. This work was supported in part by the National Science Foundation under Grant MCS-8009315. A preliminary version of this paper was presented at the IEEE 1985 International Conference on Parallel Processing, St. Charles, IL, Aug. 1985.

T. N. Mudge is with the Computing Research Lab, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109.

H. B. Al-Sadoun is with the Department of Electrical Engineering, Kuwait University, P.O. Box 5969, Kuwait.

0018-9340/85/1000-0934$01.00

the assumptions of Section II; and Section IV concludes the paper by evaluating the SMI model against simulations.

II. The System Operation Assumptions

The multiprocessor system of Fig. 1 is assumed to be synchronous with a basic time unit of a bus cycle. A processing element (PE) may be in any of three states: *thinking*, when it is working on an internal task with no memory request outstanding; *accessing*, when it is connected to a memory module; and *waiting* or *blocked*, when it is waiting in the queue of a memory module for that memory to become available or for a bus to become available. The memory module (MM) can be in any of two states: *busy*, when a processing element is connected to it; and *idle*, when there is no processing element connected to it. The following notation will be used throughout this paper: a discrete random variable will be denoted by its name with a ~ above it, e.g., the discrete random variable Z will be denoted by $\tilde{Z}$; the probability mass function (pmf) of $\tilde{Z}$ will be denoted by $z(x)$, i.e., $z(x) = \Pr[\tilde{Z} = x]$; the mean value of $\tilde{Z}$ will be denoted by $\overline{Z}$; and the nth moment of $\tilde{Z}$ will be denoted by $\overline{Z^n}$.

System operation will be characterized by the following assumptions.

1) The behavior of the PE's can be modeled as identical stochastic processes.

2) The PE's think for an integer number of bus cycles. The thinking period of any PE is characterized by a discrete independent random variable $\tilde{T}$.

3) Each PE will submit a memory request after its thinking period, i.e., the thinking time is the interrequest time. Requests originating from the same PE are independent of each other, provided they are not resubmitted requests [see 4) and 5)]. The destination of nonresubmitted requests originating from any PE will be uniformly distributed between the M memory modules.

4) The system uses a two-stage arbitration scheme following that described in [7]. In the first stage the conflict over the MM's (first conflict type) will be resolved by M arbiters of the N-users, 1-server type. In the second stage the conflict over the buses (third conflict type) will be resolved by 1 arbiter of the M-users, B-servers type. The blocked PE's will try again to access the same module in the next cycle.

5) When the second type of memory conflict occurs, i.e., the MM is busy when requested by a PE, the blocked PE waits until the connection is completed, and then it resubmits its request to the same module.

6) The connection time between a PE and any MM is characterized by a discrete independent random variable $\tilde{C}$, measured in units of bus cycles.

Empirical evidence reported in [1], [2], [6] supports the assumptions in the case where $\tilde{C}$ is a deterministic random variable with a value of 1. Further work reported in [8] supports the assumptions in the more general case where $\tilde{C}$ is a random variable with arbitrary distribution.

In order to obtain numerical information from the SMI model developed later, the values of $M, N, \overline{T}, \overline{C}$, and $\overline{C^2}$ must be obtained through measurements or by hypothesis. These quantities can be regarded as input parameters of the SMI model; knowledge of the full distributions of $\tilde{T}$ and $\tilde{C}$ is not necessary for solving the SMI model. A number of performance measures can be derived from the analytical model. These are: memory bandwidth (BW); processing element utilization (PU); memory module utilization (MU); utilization of a bus (BU); average queue length at an MM ($\overline{L}$); and average waiting time experienced by a PE ($\overline{W}$).

III. The Semi-Markov Memory Interference (SMI) Model

A Markov chain which models a multiprocessor system according to the assumptions outlined in Section II has an unmanageably large state space; see [1] and [16]. To simplify this we first adopt a technique presented in [9]. In that work separate identical Markov chains are used to describe the behavior of each PE, and the coupling between the N chains appears in the transition probabilities between the states in each chain. Solving the model requires only one of the chains to be considered, which dramatically reduces the solution complexity. Moreover, because the chains are coupled, independence of PE's does not have to be assumed, resulting in a more realistic model (assumption 1) does not imply independence). The number of states in the model of [9] can still grow large, in some cases, because it depends on the number of discrete values $\tilde{T}$ and $\tilde{C}$ can take on. This can be avoided, resulting in a further simplification, by replacing the Markov chains by semi-Markov processes. These have only four states, regardless of the distributions for $\tilde{T}$ and $\tilde{C}$. In addition, the semi-Markov processes simplify the computation of the average queue length at each MM and the average waiting time experienced by a PE.

A detailed discussion of semi-Markov processes can be found in [15]. Briefly, a semi-Markov process (SMP) is a stochastic process which can be in any one of K states $1, 2, \cdots, K$. Each time it enters state i it remains there for a random amount of time (the sojourn time) having mean η_i and then makes a transition into state j with probability p_{ij}. As a special case, a discrete time Markov chain is an SMP with a deterministic sojourn time of value one. If the SMP has an irreducible embedded Markov chain that consists of ergodic states, then the limiting probability of being in state i, denoted by P_i, can be expressed as

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

where π_i is the limiting probability of state i in the embedded Markov chain. All the SMP's that appear in this paper have irreducible embedded Markov chains with ergodic states, and therefore, (1) will always be applicable. The rate of leaving state i, λ_i, is defined as the reciprocal of the average time elapsed between two consecutive departures from state i. The rate can be obtained using the following equation:

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

Since the average sojourn time in any one of the states of the SMP's that appear in this paper is at least one system cycle, then λ_i falls in the range [0, 1] and has the same numerical value as the probability of leaving state i at the beginning of a bus cycle. The term λ_i, determined by context, will be used for both the rate of leaving state i and the probability of leaving state i.

The SMI model uses an SMP to approximate the behavior of a PE which functions according to the system operation assumptions given in Section II. Hence, N identical SMP's will approximate the behavior of the multiprocessor system. The SMP in this case is depicted in Fig. 2. The states of the SMP denote the different states of any PE. The first state is the thinking state 0. The process enters state 0 and remains there for a duration of time equivalent to the thinking time of the PE. The mean sojourn time in this state is η_0. A memory request is modeled by the SMP leaving state 0. The destination state depends on the state of the requested MM and also on whether the memory request is passed by the two levels of the arbitration logic. The second state is the accessing state 1. The process enters state 1 if the memory module is idle and the memory request is successfully passed by the two levels of the arbitration. The process remains in state 1 for a duration equivalent to the connection time between the PE and any MM. The mean value of the sojourn time in state 1 is η_1. From state 1 the process returns to state 0, i.e., the PE resumes thinking after it has completed its memory access. The third state is the full waiting state 2. The process enters state 2 when the PE requests an idle MM simultaneously with at least one other request, and one of these requests obtains access to the MM by passing the two levels of arbitration. In this case the PE has to wait for the full duration of the connection time between the MM and the selected PE. The full connection time has a mean value of η_2. A blocked PE will try again to access the MM at which it is blocked as soon as that MM is released. If it succeeds, the process enters state 1; if other PE's requested the same idle memory module simultaneously and one of these PE's obtains the connection with the MM, the process reenters state 2; otherwise, the process enters state 3. The fourth state is the residual waiting state 3. The process enters state 3 when the PE requests a busy MM, or when, due to bus contention, access is blocked to an MM even though it is idle. The PE has to wait for the residual connection time before retrying to access that particular MM. The mean value for the sojourn time in state 3 is η_3. The process then enters state 1 if the PE succeeds in accessing the MM; or it enters state 2 if the PE requests an idle MM simultaneously with other PE's and one of these PE's manages to obtain the connection with that MM; otherwise, it reenters state 3. Clearly, the SMP 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 behavior because of the symmetry in this case. The underlying approximation of the SMI model is in describing any PE behavior independently from the other PE's while compensating for the coupling between the PE's behaviors in the transition probabilities between the states of the SMP (the coupling results from the PE's sharing the MM's). The effects of this approximation are explored in detail in [13] for the unit connection time case.

In order to derive numerical information from the SMI model, the values of N, M, B, the first moment of $\tilde{T}$, and the first two moments of $\tilde{C}$ must be obtained. These quantities can be regarded as the input parameters to the model. They are defined as follows:

$N \triangleq$ the number of PE's,

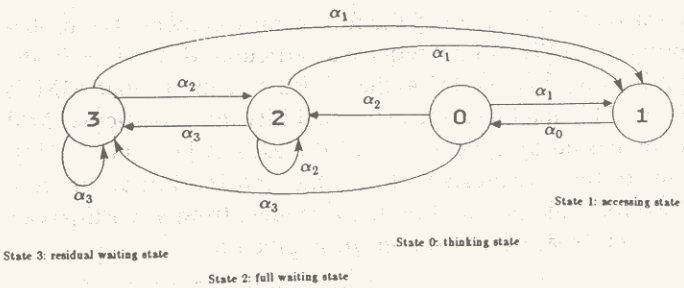


Fig. 2. The SMP that describes PE behavior in the multiple-bus system.

$M \triangleq$ the number of MM's,

$B \triangleq$ the number of buses,

$\overline{T} \triangleq$ the first moment of $\tilde{T}$,

$\overline{C} \triangleq$ the first moment of $\tilde{C}$,

$\overline{C^2} \triangleq$ the second moment of $\tilde{C}$.

The average sojourn times, of the different states of the SMP, can be obtained from the parameters of the model as follows:

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

The average sojourn times in the states 0, 1, and 2 arise directly from the definition of these states. The expression for the average sojourn time in state 3, i.e., the average residual waiting time, is taken from [11].

It is convenient to introduce some terms that will be used in formulating the model. These terms are: R, BUSY, WIN1, and WIN2. The term R is defined as the probability that a PE makes a request to access a particular MM at the beginning of a bus cycle. From our earlier definitions this is the probability of leaving states 0, 2, or 3 to access a particular MM. Therefore, R is given by

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

The term BUSY is defined as the probability that a PE finds a particular MM busy at the beginning of a bus cycle (type 2 conflict). In other words, one of the other $(N - 1)$ PE's is accessing that MM and is not on the point of releasing it. Hence, BUSY is the probability that one of $(N - 1)$ PE's is accessing a particular MM and the accessing PE is not on the point of releasing the MM. By definition, the probability that a PE is accessing an MM is P_1. Thus, the probability that it is accessing and will not leave state 1 (release the MM) in the next bus cycle is $P_1 - \lambda_1$. Therefore, BUSY can be expressed as

$$\text{BUSY} = \frac{N - 1}{M}(P_1 - \lambda_1) = \frac{N - 1}{M}(\overline{C} - 1)\lambda_1. \tag{5}$$

The term WIN1 is the probability that the memory request initiated by a PE passed the first level of arbitration, i.e., one of the M arbiters of the N-users, 1-server type selected it. The

term WIN1 is derived by the following argument. The probability that a PE will not request a particular MM is $1 - R$; the probability that none of the N PE's requests that MM is $(1 - R)^N$, and therefore the probability that a particular MM is requested by at least one of the PE's is $[1 - (1 - R)^N]$. One of these requests will pass the first level of arbitration; therefore, the probability that a request from any PE passes the first level of arbitration, p, is given by

$$p = 1 - (1 - R)^N.$$

The expected number of PE's which requested that MM at the beginning of a bus cycle is NR. Therefore, WIN1 can be defined as follows:

$$\text{WIN1} = \frac{p}{NR}. \tag{6}$$

Finally, the term WIN2 is the probability that the memory request initiated by a PE will pass the second level of arbitration given that it passed the first level of arbitration, i.e., the arbiter of the M-users, B-server type selected the PE after it had been passed by one of the M arbiters of the N-users, 1-server type. The term WIN2 is calculated by conditioning on the number of free buses. We need to calculate two quantities. The first, $X(k)$, is the probability that the request will pass the second level of arbitration given there are k free buses and that the request has already passed the first level of arbitration. The quantity $X(k)$ can be expressed as

$$X(k) = \sum_{i=1}^{M} \frac{\min(k, i)}{i} \binom{M-1}{i-1} p^{i-1}(1-p)^{M-i}.$$

The factor $\min(k, i)/i$ is the probability that if i requests pass the first level then $\min(i, k)$ will obtain buses (pass the second level). The factor $\binom{M-1}{i-1}p^{i-1}(1-p)^{M-i}$ is the probability that $(i-1)$ additional requests pass the first level given that one request has passed with certainty. The second quantity, $Y(k)$, is the probability that there are k free buses. The quantity $Y(k)$ can be derived as follows. The probability that k out of B buses are free is $\binom{B}{k}q^{B-k}(1-q)^k$ where q is the probability that a bus is busy. The term q can be found through an argument similar to that used to derive the term BUSY and can be expressed as $(N-1)((P_1 - \lambda_1)/B)$. The term WIN2 can now be obtained from

$$\text{WIN2} = \sum_{k=1}^{B} X(k)Y(k). \tag{7}$$

The transition probabilities between the states of the SMP can be expressed as the following functions of BUSY, WIN1, and WIN2:

$$\alpha_j = \begin{cases} 1 & j = 0 \\ (1 - \text{BUSY})\ \text{WIN1}\ \text{WIN2} & j = 1 \\ (1 - \text{BUSY})(1 - \text{WIN1})\ \text{WIN2} & j = 2 \\ \text{BUSY} + (1 - \text{BUSY})(1 - \text{WIN2}) & j = 3. \end{cases} \tag{8}$$

Their derivation proceeds as follows. When the process, shown in Fig. 2, leaves any of the three states 0, 2, or 3 it enters the accessing state (state 1) with probability α_1 if the requested MM is idle and the PE's request successfully passes both levels of arbitration; or the process enters the full waiting state (state 2) with probability α_2 if the requested MM is idle and the PE's request fails to be selected in the first level of arbitration and another request for the same MM passed the second level of arbitration; or the process enters the residual waiting state (state 3) with probability α_3 if the requested MM is busy or the requested MM is idle but none of the buses are free. The process always enters the thinking state after it leaves the accessing state ($\alpha_0 = 1$).

The embedded Markov chain can be solved, and the π's can be represented as functions of the transition probabilities, i.e., of BUSY, WIN1, and WIN2. The SMP limiting probabilities can be derived by substituting the limiting probabilities of the embedded Markov chain (π's) into (1); then, using (4), the SMP limiting probabilities can be expressed as functions of R and the transition probabilities as follows:

$$P_j = \begin{cases} \eta_0\alpha_1 MR & j = 0 \\ \eta_1\alpha_1 MR & j = 1 \\ \eta_2\alpha_2 MR & j = 2 \\ \eta_3\alpha_3 MR & j = 3. \end{cases} \tag{9}$$

It can be seen from the above equations that we have to solve a set of simultaneous nonlinear equations to solve the SMP of Fig. 2. The nonlinearity is introduced because the transition probabilities are defined as functions of the SMP's limiting probabilities, while the SMP's limiting probabilities are defined as functions of the transition probabilities. An iterative algorithm can be used to solve for R and λ_1, from which the performance measures discussed earlier can be derived. The algorithm breaks down as follows.

1) Calculate the average sojourn times of the states using (3).

2) Choose an initial value for R in the range $0 < R < 1$ (we used $R = 1/M$) and an initial value for λ_1 (we used $\lambda_1 = 0$).

3) Calculate the terms BUSY, WIN1, and WIN2 using (5)–(7), respectively.

4) Calculate the transition probabilities using (8).

5) Calculate an improved estimate for R by first summing the four equations of (9) to 1 and then calculating R from

$$R = \frac{1}{M(\eta_0\alpha_1 + \eta_1\alpha_1 + \eta_2\alpha_2 + \eta_3\alpha_3)}.$$

6) Calculate an improved estimate for λ_1 from

$$\lambda_1 = \alpha_1 MR.$$

7) Repeat steps 3) through 6) until R and λ_1 have the desired accuracy.[1]

The solution for R and λ_1 may be used to calculate the limiting probabilities of the states using (9). These can in turn be used to calculate the performance measures from the following equations:

$$\text{BW} = NP_1$$

$$\text{PU} = P_0 + P_1$$

[1]This is a fixed-point iteration scheme. The Steffensen iteration algorithm was used to accelerate the convergence, see [4]. No more than four iterations were needed in our experiments.

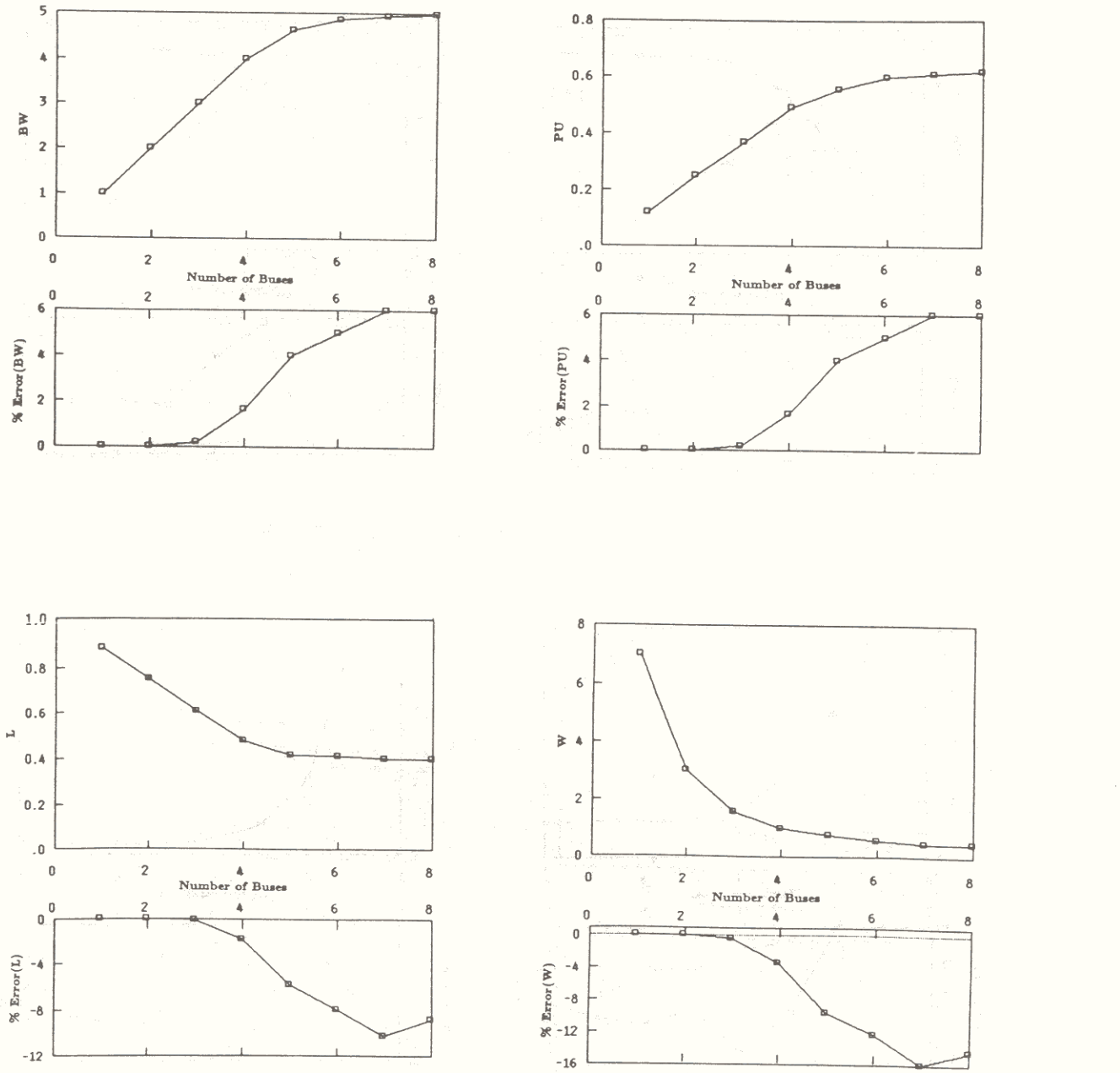


Fig. 3. Simulation results for case 1.

$$\text{MU} = \frac{N}{M} P_1$$

$$\text{BU} = \frac{N}{B} P_1$$

$$\overline{L} = \frac{N}{M} (P_2 + P_3)$$

$$\overline{W} = \frac{\eta_2 \alpha_2 + \eta_3 \alpha_3}{\alpha_1}. \tag{10}$$

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 $\tilde{W}$ in the usual way from the pmf of $\tilde{W}$. The pmf of $\tilde{W}$ can be expressed as follows:

$$\Pr[\tilde{W} = (i - j)\eta_2 + j\eta_3] = \binom{i}{j} \alpha_1 \alpha_2^{i-j} \alpha_3^j.$$

The derivation of the above equations proceeds as follows. The probability that the process moves from state 0 to state 1 after making $(i - j)$ consecutive visits to state 2 followed by j consecutive visits to state 3 is $\alpha_2^{i-j}\alpha_3^j\alpha_1$. Since there are $\binom{i}{j}$ combinations of these i visits to states 2 and 3 (not necessarily consecutive visits), then the probability that the process moves from state 0 to state 1 after making $(i - j)$ visits to state 2 and j visits to state 3 is $\binom{i}{j}\alpha_1\alpha_2^{i-j}\alpha_3^j$. The average value of the waiting time in the queue, $\overline{W}$, is calculated from the pmf of the waiting time described above. Therefore, $\overline{W}$ can be expressed as follows:

$$\overline{W} = \sum_{i=0}^{\infty} \sum_{j=0}^{i} \binom{i}{j} \alpha_1 \alpha_2^{i-j} \alpha_3^j [(i - j)\eta_2 + j\eta_3]$$

$$= \frac{\eta_2 \alpha_2 + \eta_3 \alpha_3}{\alpha_1}.$$

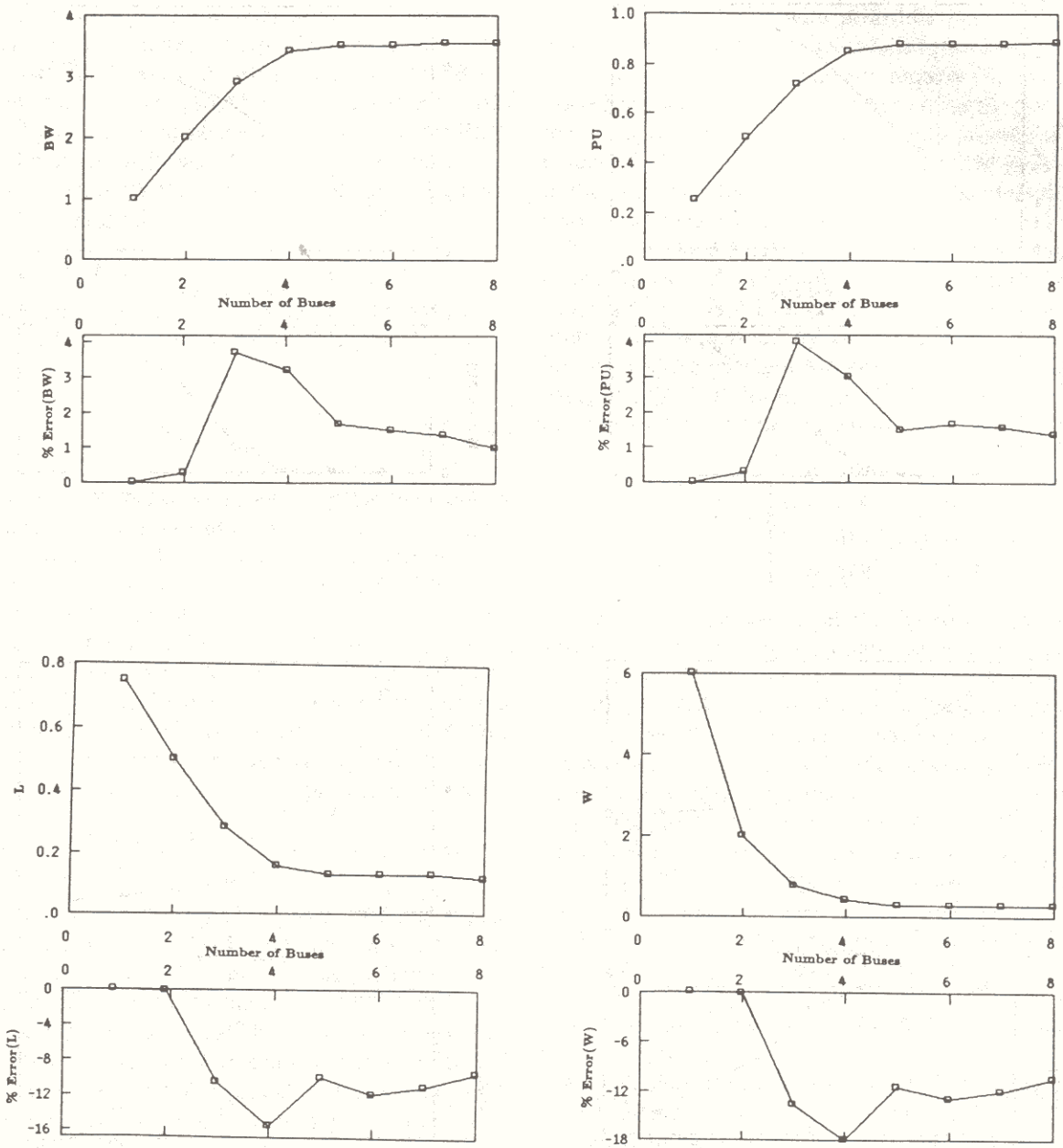


Fig. 4. Simulation results for case 2.

At this point it is possible to show how the SMI model relates to the models presented in [10], [13]. These earlier models only considered the "unit connection time" case where $\tilde{C}$ is a deterministic random variable of one bus cycle. Thus, $\overline{C} = 1$, and from (5), BUSY $= 0$. In other words, the SMI model predicts an absence of type 2 conflicts. This agrees with the system operation found in [7], [10], [13]: in a synchronous system memory accesses cannot start in the midst of a single bus cycle, but rather can start only at the beginning or end of the cycle. From (9), (10), and the relation $\eta_1 = \overline{C} = 1$, the memory bandwidth can be written

$$\text{BW} = N\alpha_1 MR\,.$$

From (6) and (8),

$$\text{BW} = pM\ \text{WIN2}\,, \tag{11}$$

but

$$\text{WIN2} = \sum_{k=1}^{B} X(k)Y(k)\,.$$

Since BUSY $= 0$, the quantity q, appearing in $Y(k)$, is 0 [see the derivation of (7)]. Thus,

$$Y(k) = \begin{cases} 1 & k = B \\ 0 & \text{otherwise}\,, \end{cases}$$

and therefore,

$$\text{WIN2} = X(B) = \frac{1}{pM}\sum_{i=1}^{M} \min(B,i)\binom{M}{i}p^i(1-p)^{M-i}.$$

Finally, from (11),

$$\text{BW} = \sum_{i=1}^{M} \min(B,i)\binom{M}{i}p^i(1-p)^{M-i}.$$

This agrees with the expression for BW first derived in [10]

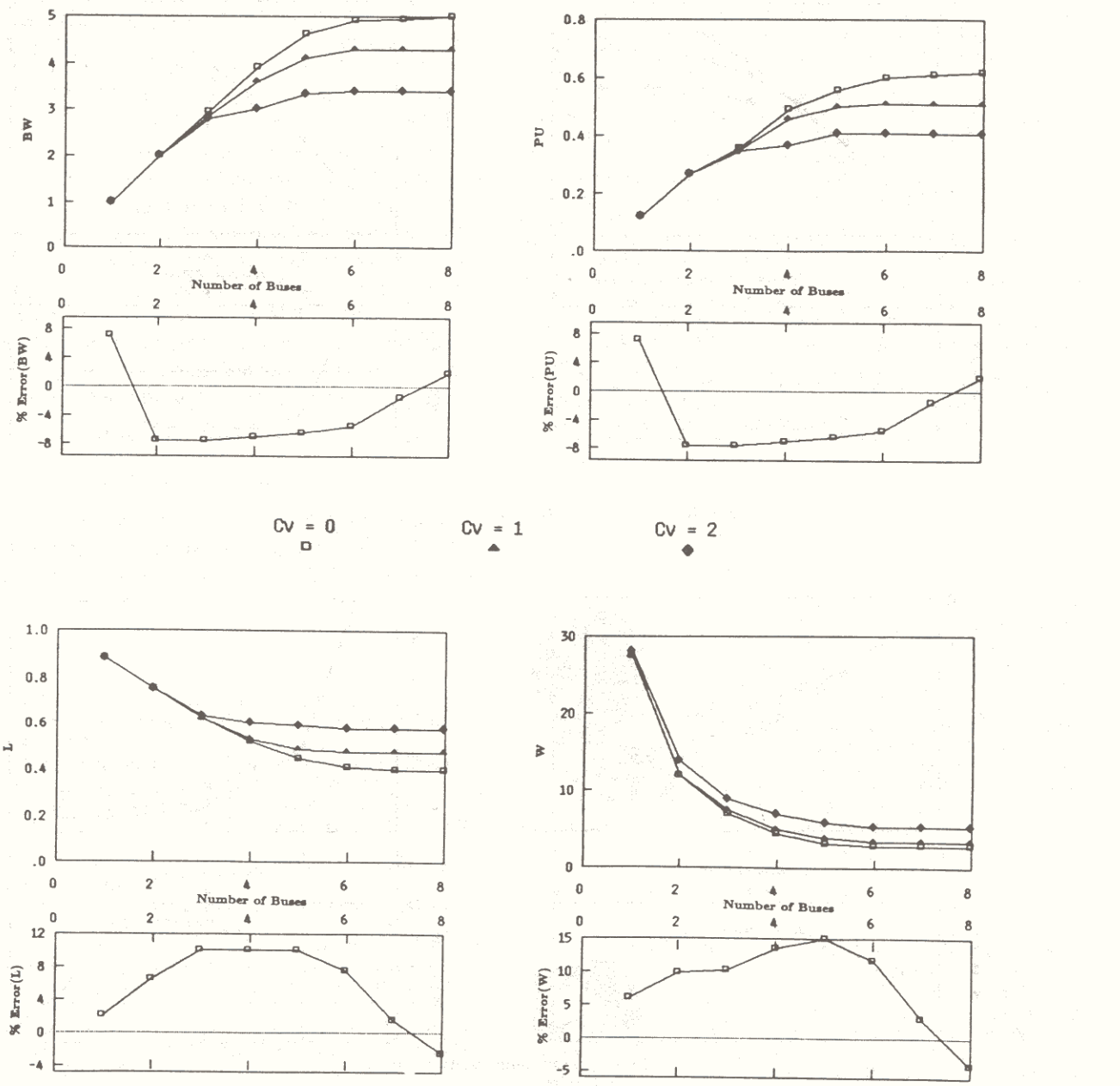


Fig. 5. Simulation results for case 3.

for the unit connection time case. The SMI model can thus be regarded as an extension of the rate adjusted model of [10].

IV. Evaluation of the SMI Model

The empirical evidence reported in [1], [2], [6], [8] led to the assumptions of Section II as a phenomenological basis for the behavior of a large class of multiprocessors. However, in Section III an approximation was introduced to allow the construction of a manageable model. This approximation was the assumption that the SMP's for each of the PE's are independent. In this section we examine the consequences of this approximation by comparing the SMI model to simulations based on the assumptions of Section II. The results are all for multiprocessor systems with eight PE's and eight MM's, and the effects of varying the other input parameters ($\overline{T}$, $\overline{C}$, $\overline{C^2}$, and B) on the performance measures (BW, PU, $\overline{L}$, and $\overline{W}$) are shown. The dependence on $\overline{C^2}$ is shown indirectly through dependence on the coefficient of variation C_v where $C_v = \sqrt{(\overline{C^2}/(\overline{C})^2) - 1}$.

In the first case the connection time between a PE and an MM lasts for one cycle, and the average think time for a PE is zero cycles. The graphs of Fig. 3 show the simulation results of BW, PU, $\overline{L}$, and $\overline{W}$ as functions of B. Below each graph for BW, PU, $\overline{L}$, and $\overline{W}$ is a graph showing the relative percentage error (%Error) between the simulation's results and the model's results. The term %Error is defined as follows:

$$\%\text{Error} = \frac{\text{Model results} - \text{Simultation results}}{\text{Simultation results}} \times 100.$$

Fig. 3 shows close agreement between the SMI model's results and the simulation's results. The utilization measures (BW and PU) were within 6 percent of the simulation. The

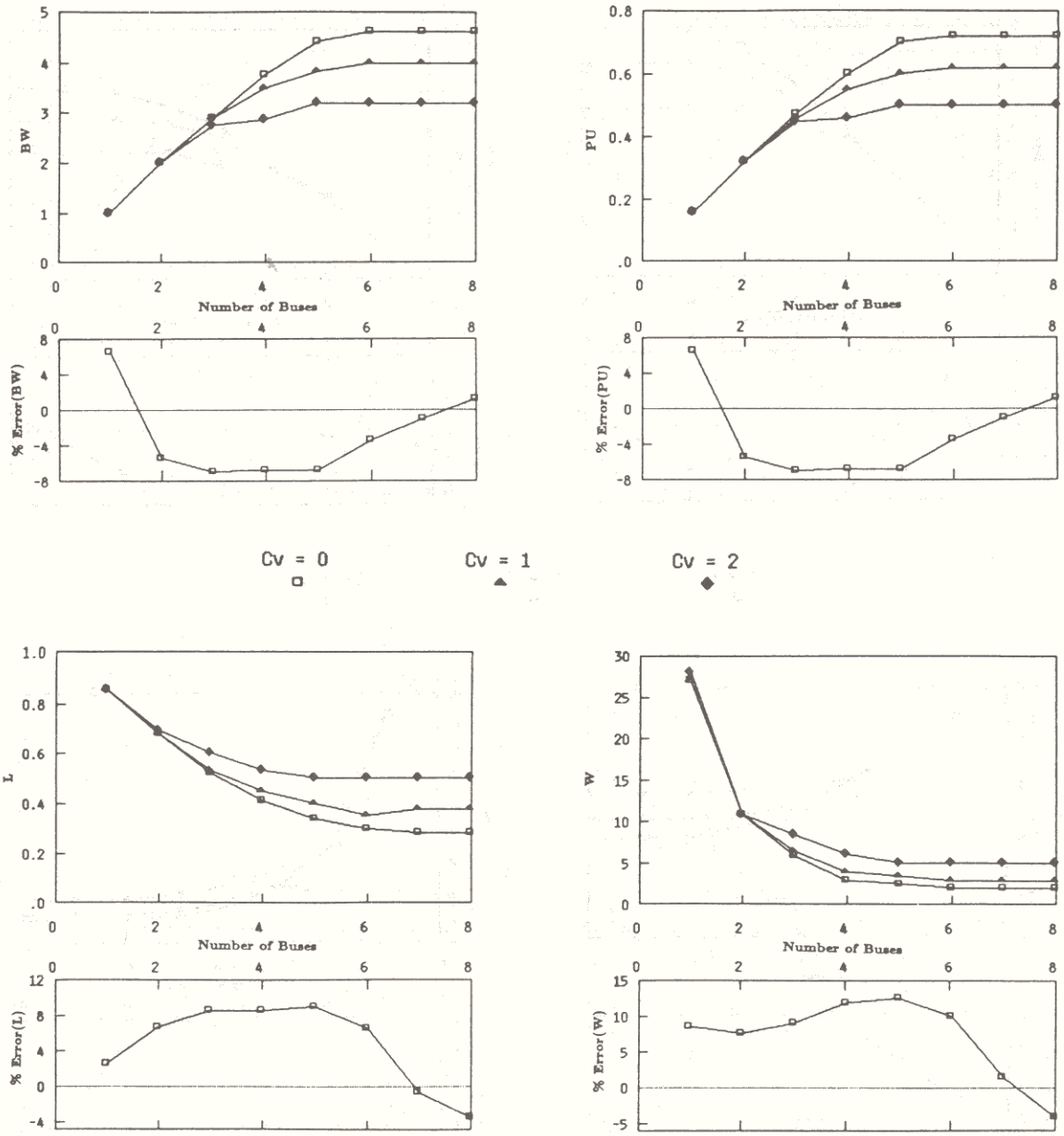


Fig. 6. Simulation results for case 4.

queue measures ($\overline{L}$ and $\overline{W}$) were within 16 percent of the simulation.

In the second case the connection time between a PE and an MM lasts for one cycle, and the average think time for a PE is one cycle. The graphs of Fig. 4 show the results. Again, there is close agreement between the SMI model's results and the simulation's results. In this case the model shows similar accuracy to the previous case. By comparing Figs. 3 and 4 we can deduce the effect of the average think time $\overline{T}$ on the system's performance. The BW decreased as $\overline{T}$ increased. The PU increased as $\overline{T}$ increased. The average queue length and the average queueing time decreased as $\overline{T}$ increased. This agrees with what one would expect, particularly if the PE's have a cache and $\overline{T}$ is the mean time between faults. Both cases 1 and 2 are the same as cases examined in the unit connection time models ($\overline{C} = 1, C_v = 0$) of [7], [10], [13]. They yield exactly the results predicted by the discussion at the end of the previous section. Both cases also point out the observation that for systems with $B << M$ the performance is bus limited, and for systems with $B > M/2$ the performance is memory limited.

In the third case the average connection time between a PE and an MM is four cycles, and the average think time for a PE is zero cycles. The graphs of Fig. 5 show the results. Three plots for each of BW, PU, $\overline{L}$, and $\overline{W}$ are shown for $C_v = 0$, 1, and 2. The %Error plots show the worst case for any of the three values of C_v at a particular value of B. Results presented in [12] separate the %Error for each value of C_v. Again there is close agreement between the SMI model's results and the simulation's results. The utilization measures (BW and PU) were within 8 percent of the simulation. The queue measures ($\overline{L}$ and $\overline{W}$) were within 15 percent of the simulation. This case demonstrates the effect of the variation in the connection time on the system performance. The system performance declines as the variation in the connection time, C_v, increases. The memory bandwidth BW and the processor utili-

zation PU decrease as C_v increases, while the average queue length $\overline{L}$ and the average queueing time $\overline{W}$ increase as C_v increases. This can be explained from the SMP of Fig. 2. Increasing C_v will increase the average sojourn time in state 3, and therefore, P_3 will increase.

In the fourth case the average connection time between a PE and an MM is four cycles, and the average think time for a PE is one cycle. The graphs of Fig. 6 show the results.

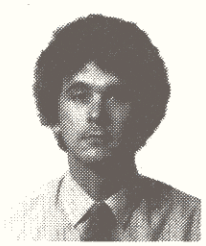
The last two cases highlight the importance of keeping C_v low. Reducing C_v from 2 to 0 can increase the BW by about 65 percent (Fig. 5 top left-hand side) and more than halve $\overline{W}$. In systems where some PE's may be DMA channels that can perform block transfers and other PE's may be simply transferring cache lines, it may be advantageous to break up the block transfers and/or increase the cache line size so that C_v is reduced.

Acknowledgment

The authors would like to thank the reviewers for suggesting ways to improve the paper. The authors would also like to thank O. A. Olukotun for plotting the graphs used in the figures.

References

[1] F. Baskett and A. J. Smith, "Interference in multiprocessor computer systems with interleaved memory," *Commun. ACM*, vol. 19, no. 6, pp. 327–334, June 1976.
[2] D. P. Bhandarkar, "Analysis of memory interference in multiprocessors," *IEEE Trans. Comput.*, vol. C-24, pp. 897–908, Sept. 1975.
[3] L. N. Bhuyan, "A combinatorial analysis of multibus multiprocessors," in *Proc. IEEE 1984 Int. Conf. Parallel Processing*, Aug. 1984, pp. 225–227.
[4] S. D. Conte and C. de Boor, *Elementary Numerical Analysis*. New York: McGraw-Hill, 1980.
[5] A. Goyal and T. Agerwala, "Performance analysis of future shared storage systems," *IBM J. Res. Develop.*, vol. 28, no. 1, pp. 95–108, Jan. 1984.
[6] C. H. Hoogendoorn, "A general model for memory interference in multiprocessors," *IEEE Trans. Comput.*, vol. C-26, pp. 998–1005, Oct. 1977.
[7] T. Lang, M. Valero, and I. Alegre, "Bandwidth of crossbar and multiple-bus connections for multiprocessors," *IEEE Trans. Comput.*, vol. C-31, pp. 1227–1233, Dec. 1982.
[8] B. A. Makrucki, "A stochastic model of multiprocessing," Ph.D. dissertation, Univ. Michigan, Ann Arbor, 1984.
[9] T. N. Mudge and H. B. Al-Sadoun, "Memory interference models with variable connection time," *IEEE Trans. Comput.*, vol. C-33, pp. 1033–1038, Nov. 1984.
[10] T. N. Mudge, J. P. Hayes, G. D. Buzzard, and D. C. Winsor, "Analysis of multiple-bus interconnection networks," in *Proc. IEEE 1984 Int. Conf. Parallel Processing*, Aug. 1984, pp. 228–232.
[11] T. N. Mudge, H. B. Al-Sadoun, and B. A. Makrucki, "A semi-Markov model for memory interference in multiprocessors," Comput. Res. Lab., Dep. Elec. Eng. Comput. Sci., Univ. Michigan, Ann Arbor, Rep. CRL-TR-44-84, Nov. 1984.
[12] T. N. Mudge and H. B. Al-Sadoun, "A semi-Markov model for the performance of multiple-bus systems," in *Proc. IEEE 1985 Int. Conf. Parallel Processing*, St. Charles, IL, Aug. 1985.
[13] T. N. Mudge, J. P. Hayes, G. D. Buzzard, and D. C. Winsor, "Analysis of multiple-bus interconnection networks," in review.
[14] Special Issue on Interconnection Networks, *IEEE Comput.*, vol. 14, Dec. 1981.
[15] S. M. Ross, *Applied Probability Models with Optimization Applications*. San Francisco, CA: Holden-Day, 1970.
[16] C. E. Skinner and J. R. Asher, "Effects of storage contention on system performance," *IBM Syst. J.*, vol. 8, no. 4, pp. 319–333, 1969.
[17] D. Towsley, "An approximate analysis of multiprocessor systems," in *Proc. ACM SIGMETRICS Conf. Measurement Modeling Comput. Syst.*, Aug. 1983, pp. 207–213.
[18] M. Valero, J. Llaberia, J. Labarta, E. Sanvicente, and T. Lang, "A performance evaluation of the multiple-bus network for multiprocessor systems," in *Proc. ACM Conf. Performance Evaluation*, 1983, pp. 200–206.



Trevor N. Mudge (S'74–M'77–SM'84) received the B.Sc. degree in cybernetics from the University of Reading, England, in 1969 and the M.S. and Ph.D. degrees in computer science from the University of Illinois, Urbana, in 1973 and 1977, respectively.

He has been with the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor, since 1977 and currently holds the rank of Associate Professor. His research interests include computer architecture, programming languages, VLSI design, and computer vision.

Humoud B. Al-Sadoun (M'85) was born in Kuwait. He received the B.S.E.E. and M.S.E.E. degrees from the University of Wisconsin, Madison, in May 1977 and December 1977, respectively, the M.S.E. degree in industrial and operational engineering from the University of Michigan, Ann Arbor, in 1982, and the Ph.D. degree in electrical and computer engineering from the University of Michigan in 1985.

From 1978 to 1979 he worked as an Engineer with the Kuwait National Petroleum Company in the area of electronics and instrumentation. He is currently an Assistant Professor in the Department of Electrical and Computer Engineering, Kuwait University.

Dr. Al-Sadoun is a member of the Association for Computing Machinery, the IEEE Computer Society, and ORSA.