JOURNAL OF PARALLEL AND DISTRIBUTED COMPUTING 3, 328–343 (1986)

Analysis of Multiple-Bus Interconnection Networks*

T. N. MUDGE, J. P. HAYES, G. D. BUZZARD, AND D. C. WINSOR

Advanced Computer Architecture Laboratory, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, Michigan 48109-1109

Received February 4, 1985

The performance of multiple-bus interconnection networks for multiprocessor systems is analyzed, taking into account conflict arising from memory and bus interference. A discrete stochastic model of bandwidth is presented for systems in which each memory is connected either to all the buses or to a subset of the available buses. The effects of the assumptions made concerning independence among requests for different memories (spatial independence) and resubmission of blocked requests (temporal independence) are investigated systematically. The basic bandwidth model is extended to account for spatial dependence, and compared to previously proposed models. Finally, the various analytic models are shown to be in close agreement with simulation results. © 1986 Academic Press, Inc.

I. INTRODUCTION

A great deal of attention has been paid to the design and analysis of interconnection networks for multiprocessor systems. Most of the previous research has dealt with crossbar networks or multistage networks [1]. While these networks are attractive for applications where high bandwidth is required, their high cost and special implementation requirements have prevented them from being used for the full range of multiprocessor applications. Most commercial systems containing more than one processor employ a single bus; consider, for example, the design philosophy advocated for the iAPX 86 family in which the Multibus (IEEE 796 standard bus) provides all the intrasystem communication [2]. Single-bus systems are inexpensive and easy to implement but have limited bandwidth and lack fault tolerance. A natural extension is to employ several shared buses to increase bandwidth and fault tolerance at moderate cost. Figure 1 shows typical systems in which B

* This work was supported by the National Science Foundation under Grants ECS-8214709 and MCS-8009315.

0743-7315/86 $3.00

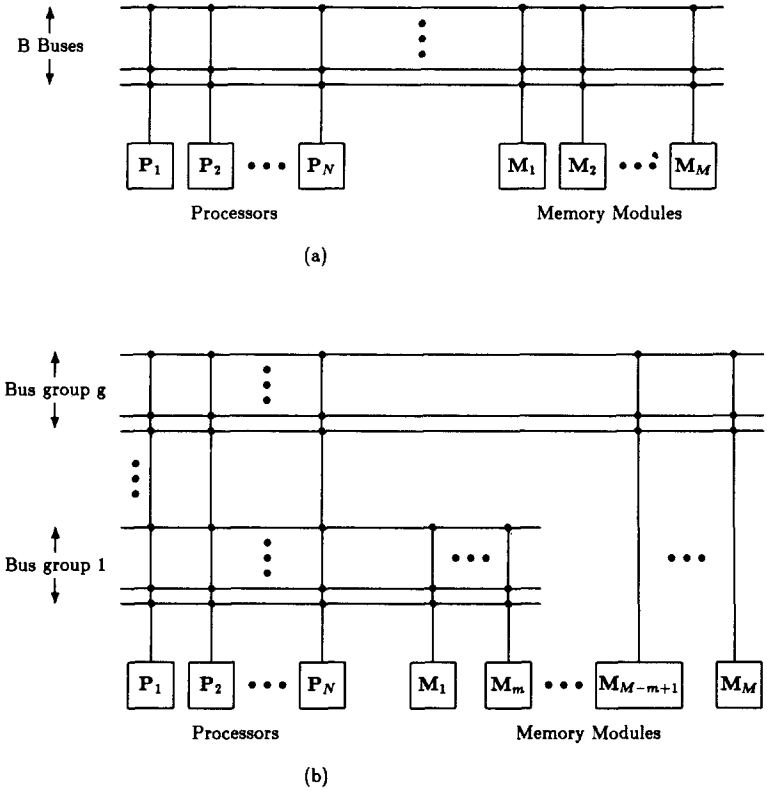

FIG. 1. Two multiprocessor systems with multiple-bus interconnection networks. (a) Complete; (b) partial.

buses are used to interconnect N processors to M memory modules ($B \leq N$). Unlike a crossbar or multistage network, a multiple-bus interconnection scheme allows easy incremental expansion of the number of processors and memories in the system. Furthermore, the buses can be configured in various ways to provide a wide range of trade-offs between bandwidth, connection cost, and reliability.

Lang *et al.* [3, 4] were among the first to investigate the performance of multiple-bus systems of the kind depicted in Fig. 1. Using simulation they determined the bandwidth characteristics of two representative bus configurations, complete and partial. In the *complete* case, which is illustrated in Fig. 1a, every processor and memory module is connected to every bus; in the *partial* case, which is illustrated in Fig. 1b, each memory need only be connected to a subset of the buses. In particular, Lang *et al.* [3] showed that a complete multiple-bus configuration with $B \approx N/2$ has almost the same bandwidth as an $N \times M$ crossbar, as well as higher fault tolerance. Partial bus configurations can achieve the same bandwidth at lower cost and lower fault

tolerance. Recently, others have derived analytical models for the performance of multiple-bus systems under a variety of assumptions [5–9].

In this paper, we develop a discrete stochastic model for the bandwidth of both complete and partial multiple-bus systems following the approach in [7]. A similar result for the complete case is presented in [6]. These papers, and those of most other authors, employ the following *temporal independence* assumption: successive memory requests by a processor are independent, i.e., blocked requests are, in effect, discarded. The basic model of [6, 7] also implicitly employs the following simplifying assumption: the event E_j that there is at least one request for a particular memory $\mathbf{M}_j$ is independent of the E_k's for $k \neq j$. We will refer to this as the *spatial independence* assumption. A Markov chain model that avoids both of these assumptions is postulated by Valero *et al.* in [8]; however, it is intractable for systems with more than about four processors. The same authors derive a somewhat simpler model assuming only temporal independence, which they term the memoryless property. A more elegant formulation based on the same assumptions is presented by Bhuyan [5], which employs Stirling numbers. Nevertheless, the bandwidth formulas of [5] are computationally complex compared to those of [6, 7], and do not extend readily to the partial bus case. A more general model that does not assume temporal independence but does assume spatial independence is presented in [9]. It extends the models of [5–8] by allowing variable-length memory accesses.

In Section II the basic model for the bandwidth *BW* of complete and partial multiple-bus systems is derived. A bandwidth model that accounts for spatial dependence is derived in Section III, and shown to be equivalent to, but simpler than, those of [5, 8]. It is also demonstrated that spatial dependence considerations can be ignored when $B \geq M$, generalizing an earlier result of [10]. Section IV develops an iterative scheme to reduce the error caused by the assumption of temporal independence, and also gives some simple asymptotic approximations to *BW*. Finally, some simulation data are presented in Section V and compared to the analytic results. It is concluded that concern about the spatial dependence of memory requests is usually not warranted, as inaccuracies caused by the spatial independence assumption appear to be masked by those due to temporal independence.

II. Basic Model

The bus systems under consideration (Fig. 1) are assumed to be synchronous, and processor–memory transactions are assumed to occur during discrete time intervals termed bus cycles. (Continuous time analogs of such systems are discussed in [11, 12].) For the purposes of this paper, bandwidth will be defined as the expected number of buses in use during a bus cycle.

Apart from the dimensions of the system, i.e., the values of B, M, N, and the bus grouping used, the most important factors affecting bandwidth are the rate at which memory requests are made by processors and the degree of conflict that those requests experience.

There are two sources of conflict due to memory requests in a multiple-bus system. First, more than one request can be made to the same memory module, resulting in memory interference. Second, there may be an insufficient number of buses available to accommodate all the memory requests, resulting in bus interference. In [3] and later papers, a two-stage arbitration scheme is used to resolve these conflicts. In the first stage, memory interference is resolved by M 1-out-of-N arbiters each of which selects at most one outstanding request per memory module. In the second stage, bus interference is resolved by a B-out-of-M arbiter which assigns the buses to the memory requests selected in the first stage. The assignment is done on a round-robin basis by each bus arbiter. In a realistic system requests that are blocked by either memory or bus interference are resubmitted during the following bus cycle. This policy for handling rejected requests is implemented in the simulation model of [3]. Analytic models that capture this temporal dependence feature appear to be intractable except in those cases where B, M, and N are very small [8].

The basic assumptions underlying our model follow those of Lang *et al.* [3]. Each processor is assumed to generate independent requests (Bernoulli trials) for memory with probability p at the start of each bus cycle. This value of p will be referred to as the request rate. Modeling the memory access process as a Bernoulli process has been validated empirically in [13–15], and is widely used as a basis for memory interference models. The memory requests are assumed to be uniformly distributed across all the memories with probability $1/M$; this is a reasonable assumption when address interleaving based on the low-order address bits is used. Hence, the probability that processor $\mathbf{P}_i$ requests memory $\mathbf{M}_j$ is p/M for all i and j. Note that the foregoing assumptions imply temporal independence, so that the rejected requests are in effect discarded.

The analysis can be treated in two parts corresponding to memory interference and bus interference. Our development follows that presented in [7].

Memory Interference Analysis. As noted above, the probability that processor $\mathbf{P}_i$ requests memory $\mathbf{M}_j$ is p/M. It follows that the probability that $\mathbf{P}_i$ does not request $\mathbf{M}_j$ is $(1 - p/M)$, and further, that the probability that none of $\mathbf{P}_i$ $(i = 1, \ldots, N)$ requests $\mathbf{M}_j$ is $(1 - p/M)^N$. This last expression can also be interpreted as the probability that the 1-out-of-N arbiter associated with $\mathbf{M}_j$ has no input requests from which to choose. Conversely, if E_j is the event that there is at least one request for $\mathbf{M}_j$, then the probability of E_j is

$$\Pr[E_j] = q = 1 - (1 - p/M)^N. \tag{1}$$

From the behavior of the arbiters, we can conclude that the probability that one request gains access to $\mathbf{M}_j$ is q for any j.

Bus Interference Analysis. Only the requests from at most B of the M 1-out-of-N memory request arbiters can be handled during any bus cycle, since there are only B buses. If the events E_j are assumed to be independent, (the spatial independence assumption), then the probability that exactly i of the M memory request arbiters output a memory request is given by

$$f(i) = \binom{M}{i} q^i (1 - q)^{M-i}. \tag{2}$$

The effects of this independence assumption are examined in Section III. The probability that B or more of the M memory request arbiters output a memory request can be written as

$$\mathbf{F}(B) = \sum_{i=B}^{M} f(i). \tag{3}$$

This is the probability that all B buses are in use, i.e., the interconnection network is saturated. Since the bandwidth BW is defined as the expected number of buses in use during a bus cycle, (2) and (3) yield the following expression for BW of a complete multiple-bus system:

$$BW = B\mathbf{F}(B) + \sum_{i=1}^{B-1} if(i). \tag{4}$$

As is shown in Section IV, this expression for BW is in close agreement with simulation results.

We now generalize (4) for the case of partial buses (Fig. 1b). The memory interference analysis is the same as before, since it is independent of the bus configuration, i.e., Eq. (1) continues to apply. However, the bus interference analysis needs modification. Let the B buses be divided into g equal groups, assuming g is a factor of B and M. With $m = M/g$ and $b = B/g$, Eqs. (2) and (3) become

$$f_g(i) = \binom{m}{i} q^i (1 - q)^{m-i}$$

$$\mathbf{F}_g(B) = \sum_{i=b}^{m} f_g(i).$$

Consequently, the bandwidth can be written as

$$BW_g = g\left[b\mathbf{F}_g(B) + \sum_{i=1}^{b-1} if_g(i)\right],$$

which is simply g times the bandwidth of any one of the g subsystems formed from N processors, b buses, and m memories. In the case where $g = 1$, the subscript is omitted as in (2), (3), and (4).

III. Model with Spatial Dependence

In deriving Eq. (2) for $f(i)$, the events E_j were assumed to be independent. Strictly speaking this is not so, as can be seen if one considers the case where $M > N$; according to (2), $f(M) \neq 0$, which is clearly impossible as there are only N possible sources of memory requests. The dependence between the events E_j can be formulated explicitly as follows.

$$\begin{aligned}\Pr[E_j \mid E_k] &= \sum_{i=1}^{N} \Pr[E_j \mid E_k \text{ results from } i \text{ requests}] \times \\ &\qquad \Pr[E_k \text{ results from } i \text{ requests}] \\ &= \sum_{i=1}^{N} \left[1 - \left(1 - \frac{p}{M-1}\right)^{N-i}\right]\binom{N}{i}\left(\frac{p}{M}\right)^i\left(1 - \frac{p}{M}\right)^{N-i} \\ &= \Pr[E_j] - \sum_{i=1}^{N} \binom{N}{i}\left(\frac{p}{M}\right)^i\left(1 - \frac{p}{M}\right)^{N-i}\left(1 - \frac{p}{M-1}\right)^{N-i}.\end{aligned}$$

Therefore, $\Pr[E_j \mid E_k] \neq \Pr[E_j]$, so the events E_j and E_k are dependent.

The effects of spatial dependence can be taken into account by replacing $f(i)$ with a new function $h(i)$ denoting the exact probability that i of the M memory request arbiters output a memory request. An expression for $h(i)$ can be developed as follows. If Q is the event that a processor sends at most one request to i particular memories, then

$$\Pr[Q] = 1 - p + \frac{ip}{M}.$$

If R is the event that each of the N processors sends at most one request to i particular memories, then

$$\Pr[R] = \left(1 - p + \frac{ip}{M}\right)^N.$$

The quantity R can also be viewed as the event that no more than $\min(i, N)$ of the i particular memories are busy. Thus, if S_i is the event that no more than i of the M memories are busy, then

$$\Pr[S_i] = \binom{M}{i}\left(1 - p + \frac{ip}{M}\right)^N.$$

Note that $\binom{M}{i} = 0$ if $i > M$ and M is a positive integer. If s_{ij} is the event that exactly j of any subset of i of the M memories are busy, then

$$\Pr[s_{ij}] = \binom{M - j}{i - j}h(j).$$

Furthermore, if s_i is the event that exactly i of the memories are busy, then

$$s_i = S_i - \bigcup_{j=0}^{i-1} s_{ij}, \qquad i > 0.$$

From the definition of s_{ij} it follows that $s_{ij} \cap s_{ik} \neq \emptyset$ iff $j = k$. Therefore,

$$h(i) = \Pr[s_i] = \Pr[S_i] - \sum_{j=0}^{i-1} \Pr[s_{ij}],$$

that is,

$$h(i) = \binom{M}{i}\left(1 - p + \frac{ip}{M}\right)^N - \sum_{j=0}^{i-1} \binom{M - j}{i - j}h(j). \tag{5}$$

Two alternative expressions for $h(i)$ were developed in [5, 8]. They can both be expressed in the form

$$h(i) = \sum_{k=i}^{N} \binom{N}{k}p^k(1 - p)^{N-k}\binom{M}{i}M^{-k}\theta. \tag{6}$$

In [8], the term θ of (6) is written as

$$\theta = \sum \frac{k!}{n_1! \ldots n_i!}, \tag{7}$$

where the summation is carried out over all $n_1, \ldots, n_i > 0$ such that $n_1 + \cdots + n_i = k$. In [5], on the other hand, we find the closed-form expression

$$\theta = i!\left\{{k \atop i}\right\}, \tag{8}$$

where $\{{k \atop i}\}$ denotes a Stirling number of the second kind. $\{{k \atop i}\}$ is defined as the number of ways to partition a set X_k of k elements into i nonempty disjoint subsets [18]. Thus (8) denotes the number of ways to partition X_k into i subsets, taking their ordering into account; this is precisely the intended meaning of (7). It can be shown that (5) is identical to (6) for both interpretations of θ. Our formulation has the advantage of expressing explicitly the recurrence implicitly required to evaluate (7) and (8), which simplifies the computation of $h(i)$.

Equations (3) and (4) can now be rewritten as

$$\mathbf{H}(B) = \sum_{i=B}^{M} h(i)$$

and

$$BW = B\mathbf{H}(B) + \sum_{i=1}^{B-1} i h(i). \tag{9}$$

To distinguish between the expressions for bandwidth given by (9) and (4), the symbols BW^h and BW^f, respectively, will be used in the remainder of this paper. The superscript will be omitted if the bandwidth can be represented by either expression. Unlike BW^f, the formula for BW^h does not appear to extend easily to the partial bus case.

The spatial dependence among memory requests implies that $BW^h \neq BW^f$. However, in the bus-sufficient case where $B \geq M$ we have $BW^h = BW^f$, as was proved in [10] for the case $p = 1$. We now briefly present the proof for the bus-sufficient case with arbitrary p. Define an indicator random variable $\tilde{I}_j$ as follows:

$$\tilde{I}_j = \begin{cases} 1 & \text{if } \mathbf{M}_j \text{ is busy, i.e., if } E_j \text{ occurs} \\ 0 & \text{otherwise.} \end{cases}$$

From the earlier discussion of spatial dependence, the different $\tilde{I}_j$ variables are dependent. Using these indicator variables, we can express the bandwidth BW^h as follows:

$$BW^h = E\left[\sum_{j=1}^{M} \tilde{I}_j\right].$$

Since the expected value of the sum of random variables is equal to the sum of their expected values, even if the variables are not independent, the ex-

pected value operator can be moved inside the summation to yield the expression

$$BW^h = \sum_{j=1}^{M} E[\tilde{I}_j],$$

in which $E[\tilde{I}_j]$ can be replaced by $\Pr[E_j]$ thus:

$$BW^h = \sum_{j=1}^{M} \Pr[E_j]. \tag{10}$$

To develop an expression for $\Pr[E_j]$ we can follow an argument similar to that presented in the memory interference analysis of Section II. The probability that no processor requests a particular memory is $(1 - p/M)^N$. In the bus-sufficient case this is the same as the probability that a particular memory $\mathbf{M}_j$ does not receive a request. Thus, in the bus-sufficient case, the probability that $\mathbf{M}_j$ receives a request is

$$\Pr[E_j] = 1 - (1 - p/M)^N;$$

hence from (10),

$$BW^h = M[1 - (1 - p/M)^N].$$

Rewriting (4) for $B \geq M$ yields

$$\begin{aligned} BW^f &= Mf(M) + \sum_{i=1}^{M-1} if(i) \\ &= M[1 - (1 - p/M)^N], \end{aligned}$$

which shows that $BW^f = BW^h$. This simple analysis no longer applies in the bus-deficient case ($B < M$), as the probability that $\mathbf{M}_j$ does not receive a request may be greater than $(1 - p/M)^N$ due to lack of available buses. The distribution $h(i)$ must then be determined explicitly in order to compute BW^h.

IV. Other Models

A major source of error arises from the assumption that blocked requests are discarded. In reality, and also in the simulations of [3] and later papers, blocked requests are repeatedly resubmitted or queued until the memory they request allows them access. The BW expressions derived in the preceding sections can be refined by taking this into account in the manner described below.

The probability that a memory request is accepted in the bus cycle in which it is made is given by

$$P_a = \frac{BW}{Np}. \tag{11}$$

The numerator of (11), i.e., the bandwidth, measures the number of requests that obtain memory access during a bus cycle. The denominator of (11) measures the total number of requests made by all the processors during a bus cycle. It is convenient, following [15], to define an "adjusted" request rate α, that accounts for resubmission of rejected requests, where $0 \leq p \leq \alpha \leq 1$. By assumption, each memory request is a Bernoulli trial with success probability p or, in the case of the adjusted rate, α. It follows that the mean number of bus cycles before a request (trial) is $1/p - 1$, or in the case of the adjusted rate, $1/\alpha - 1$ [16]. Thus, the ratio of the number of successful requests to the total number of requests, i.e, P_a, is given by

$$P_a = \frac{1/\alpha - 1}{1/p - 1}. \tag{12}$$

Equations (11) and (12) can be used in an iteration scheme to get an improved estimate for BW due to the adjusted rate α, as follows:

$$\alpha_{k+1}^{-1} = 1 + \frac{BW(\alpha_k)}{Np^2}(1 - p). \tag{13}$$

Here $BW(\alpha_k)$ is defined by (4) or (9) with α replacing p in the equations for $f(i)$ and $h(i)$, respectively. Solution of (13) for α_{k+1} yields an improved value $BW(\alpha_{k+1})$ for the bandwidth. Any remaining deviations from the simulated bandwidth occur because α does not take into account the fact that resubmissions are all directed to the same memory. This technique is an adaptation of a method first proposed by Hoogendoorn [15] (for details see [17]) and will be referred to as the *iterative improvement* method. For systems with large values of M or N, a higher-order iterative scheme may be used in place of Eq. (13) to reduce the number of steps to solution. Notice from (13), that in the limiting case of $p = 1.0$, iteration is unnecessary as $\alpha = 1.0$ also.

Finally, we consider some asymptotic approximations to the bandwidth of multiple-bus systems. From (4) we see that BW^f is bounded above by $\sum_{i=0}^{M} Bf(i)$, that is, $BW^f \leq B$. We obtain another upper bound by replacing the first term on the right-hand side of (4) by $\sum_{i=B}^{M} if(i)$, to yield $BW^f \leq \sum_{i=0}^{M} if(i)$, that is, $BW^f \leq Mq$. It follows that B and Mq define asymptotic bounds on BW, as illustrated in Fig. 2. The equation

$$BW = \min(B, Mq)$$

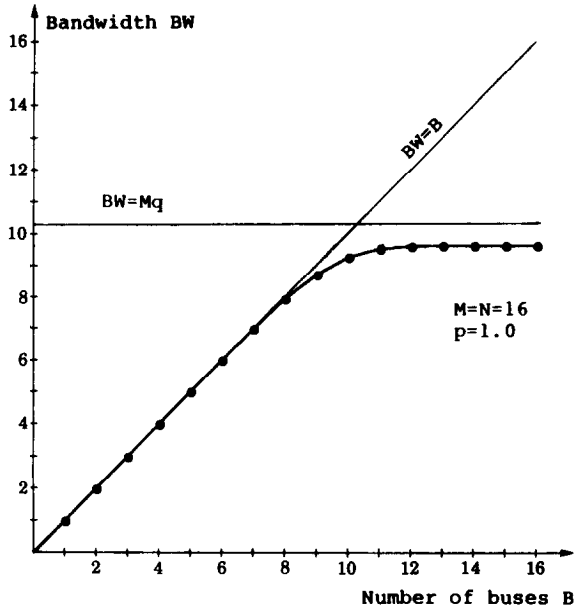

FIG. 2. Asymptotic behavior of the bandwidth *BW*.

is a simple, but useful, bandwidth approximation with the following intuitive interpretation. When the number of buses B is less than the total bandwidth Mq demanded of the memory, the buses become the limiting factor, i.e., $BW = B$. When the number of buses available exceeds Mq, the buses are no longer a scarce resource, and the bandwidth can achieve its maximum value Mq, which we refer to as the *bus-sufficient* bandwidth. This suggests that making B greater than the bus-sufficient bandwidth Mq will have little influence on bandwidth.

To measure the effect ΔBW on bandwidth of removing or losing a bus, we note that

$$\Delta BW = BW[B] - BW[B - 1],$$

where $BW[B]$ is the bandwidth with B buses. From (4) it follows that

$$\Delta BW = \mathbf{F}(B).$$

Equation (3) indicates that $\mathbf{F}(B)$ is the sum of the last $M - B + 1$ terms of a binomial series. This can be approximated by the tail of a normal distribution using the de Moivre–Laplace limit theorem [16], which states that

$$\lim_{n\to\infty} \sum_{r=z_1}^{z_2} \binom{n}{r} t^r(1-t)^{n-r} = N(\alpha) - N(\beta),$$

where $0 \leq t \leq 1$, $\alpha = (z_2 - nt)/\sqrt{nt(1-t)}$, $\beta = (z_1 - nt)/\sqrt{nt(1-t)}$, and $N(x)$ is the area under the normal or Gaussian distribution function from $-\infty$ to x, i.e., $N(x) = \int_{-\infty}^{x} e^{-y^2/2}\, dy$. Applying this theorem to (3) yields

$$\Delta BW \approx N(\alpha) - N(\beta),$$

where, in this case, $\alpha = (M - Mq)/\sqrt{Mq(1-q)}$ and $\beta = (B - Mq)/\sqrt{Mq(1-q)}$. Now $N(2) \approx 0.98$; therefore $N(\alpha) \approx 1$, and $N(\beta) \approx 1$, i.e., $\Delta BW \approx 0$ if the inequality

$$B > Mq + 2\sqrt{Mq(1-q)}$$

holds, assuming $M > B$. For example, when $M = N = 16$ and $p = 0.5$, a value of $B > 10$ yields a bandwidth that changes by no more than 2% if a bus is removed.

V. Evaluation of Results

In this section we compare the results obtained from our analytic models with the simulation data of Lang *et al.* [3]. Following [3], only $N \times N$ multiprocessor configurations ($N = M$) are considered for the complete case, and the partial case with two groups of buses. We present our results in the form of graphs showing the percentage difference or error ε between the simulated BW and the predicted BW, with and without iterative improvement. The actual values of BW for the basic model of Section II and the corresponding simulated values can be found in [7] for both complete and partial bus configurations.

Figure 3 compares simulation results with the basic model (BW^f) for the complete bus configuration with request rate $p = 1.0$. The value of ε is plotted against the number of buses B for four representative values of N ($= M$). It can be seen that for these four cases the maximum error is less than 7%. Since $p = 1.0$, identical results are obtained using the iterative improvement scheme of Section IV. The data for $p = 0.5$ are shown in Fig. 4. Here we have two distinct sets of four plots: the basic model, and the basic model with iterative improvement. Iteration reduces the maximum value of ε from about 13% to about 4%. Figures 5 and 6 show the analogous results for the case when spatial dependence is taken into account, i.e., for BW^h instead of BW^f. Again, iteration on α reduces the error. However, replacing BW^f by BW^h does not necessarily yield greater accuracy. In fact, the data presented in Figs. 3–6 do not show one to be consistently better than the other. Apparently the errors associated with temporal dependence can nullify the spatial dependence correction. Since BW^h is more complicated to compute than BW^f, the usual assumption of spatial independence seems justified.

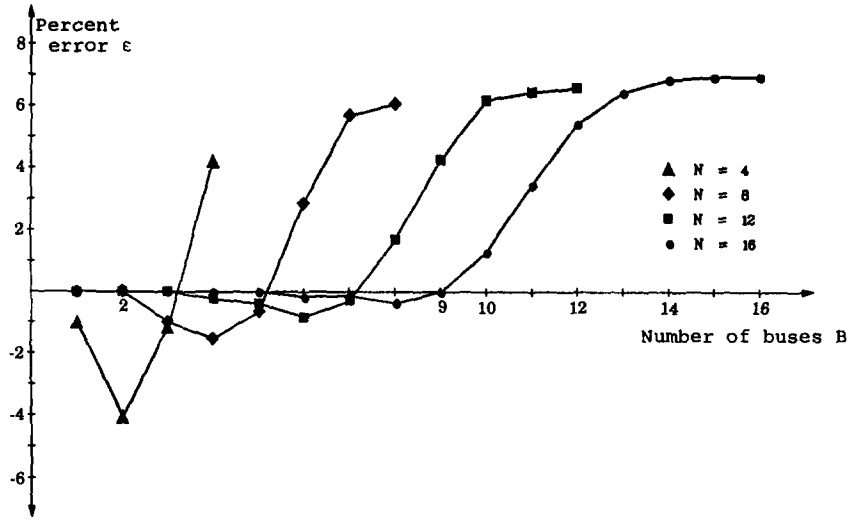

FIG. 3. Simulation vs BW^f for the complete bus configurations with $p = 1.0$.

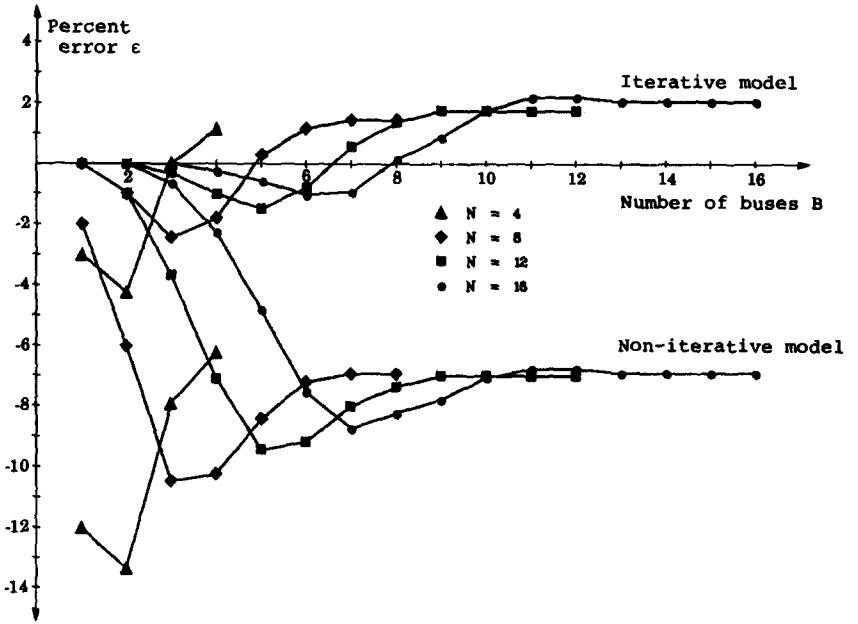

FIG. 4. Simulation vs BW^f for the complete bus configurations with $p = 0.5$.

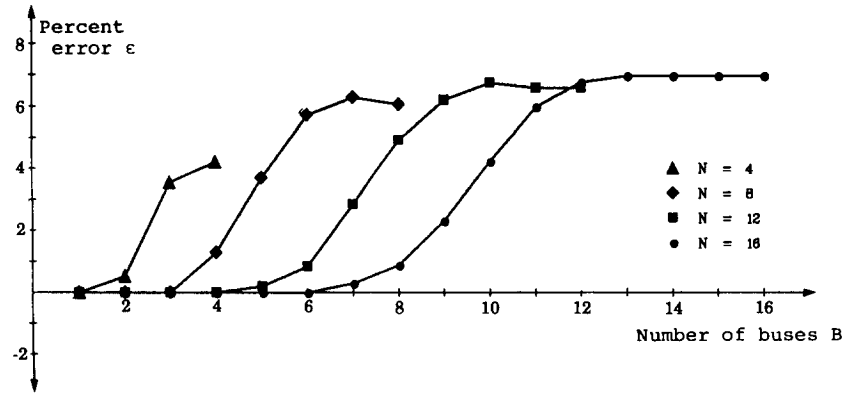

FIG. 5. Simulation vs BW^h for the complete bus configurations with $p = 1.0$.

In conclusion, note that we have focused on one particular performance measure for multiple-bus interconnection networks, namely their bandwidth. Several other related measures exist, and may be useful in some situations, in particular: the probability P_a of a request being accepted, which is defined by (11); the average utilization U of a processor; and the expected waiting

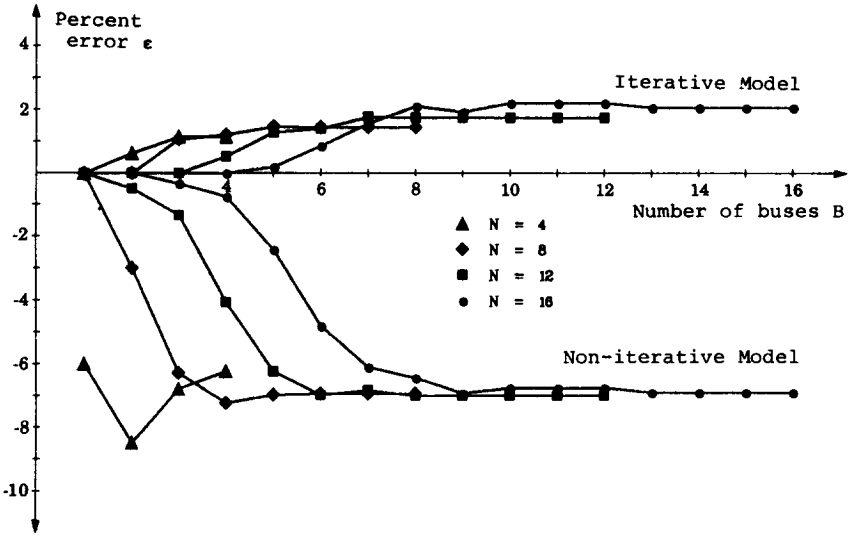

FIG. 6. Simulation vs BW^h for the complete bus configurations with $p = 0.5$.

time W, before a request is allowed memory access. The equations for U and W are given in [19]:

$$U = 1 - p(1 - P_a)$$

$$W = \frac{1}{P_a} - 1.$$

Equation (11) for P_a and these equations for U and W have accuracy similar to that of the BW models reported here.

REFERENCES

1. Siegel, H. J. *Interconnection Networks for Large-Scale Parallel Processing*. Lexington Books, Lexington, Mass., 1985.
2. Intel Corp. *iAPX 86, 88 User's Manual*. Santa Clara, Calif., 1981.
3. Lang, T., Valero, M., and Alegre, I. Bandwidth of crossbar and multiple-bus connections for multiprocessors. *IEEE Trans. Comput.* **C-31** (Dec. 1982), 1227–1233.
4. Lang, T., Valero, M., and Fiol, M. A. Reduction of connections for multibus organization. *IEEE Trans. Comput.* **C-32** (Aug. 1983), 707–716.
5. Bhuyan, L. N. A combinatorial analysis of multibus multiprocessors. *Proc. 1984 Int'l. Conf. on Parallel Processing*, Aug. 1984, pp. 225–227.
6. Goyal, A., and Agerwala, T. Performance analysis of future shared storage systems. *IBM J. Res. Develop.* **28** (Jan. 1984), 95–108.
7. Mudge, T. N. *et al.* Analysis of multiple bus interconnection networks. *Proc. 1984 Int'l. Conf. on Parallel Processing*, Aug. 1984, pp. 228–232.
8. Valero, M., *et al.* A performance evaluation of the multiple bus network for multiprocessor systems. *Proc. ACM Conf. on Performance Evaluation*, 1983, pp. 200–206.
9. Mudge, T. N., and Al-Sadoun, H. B. A semi-Markov model for the performance of multiple-bus systems. *IEEE Trans. Comput.* **C-34** (Oct. 1985), 934–942.
10. Chang, D. Y., *et al.* On the effective bandwidth of parallel memories. *IEEE Trans. Comput.* **C-26** (May 1977), 480–489.
11. Marsan, M. A., and Gerla, M. Markov models for multiple bus multiprocessor systems. *IEEE Trans. Comput.* **C-31** (Mar. 1982), 239–248.
12. Önyüksel, I. H., and Irani, K. B. A Markovian queueing network model for performance evaluation of bus-deficient multiprocessor systems. *Proc. 1983 Int'l Conf. on Parallel Processing*, Aug. 1983, pp. 437–439.
13. Bhandarkar, D. P. Analysis of memory interference in multiprocessors. *IEEE Trans. Comput.* **C-24** (Sept. 1975), 897–908.
14. Baskett, F., and Smith, A. J. Interference in multiprocessor computer systems with interleaved memory. *Comm. ACM* **19** (June 1976), 327–334.
15. Hoogendoorn, C. H. A general model for memory interference in multiprocessors. *IEEE Trans. Comput.* **C-26** (Oct. 1977), 998–1005.
16. Feller, W. *An Introduction to Probability Theory and Its Applications*, Vol. 1, 3rd ed. Wiley, New York, 1968.

17. Yen, D. W. L., Patel, J. H., and Davidson, E. S. Memory interference in synchronous multiprocessor systems. *IEEE Trans. Comput.* **C-31** (Nov. 1982), 1116–1121.

18. Knuth, D. E. *The Art of Computer Programming*, Vol. 1, *Fundamental Algorithms*. Addison–Wesley, Reading, Mass., 1968.

19. Mudge, T. N., Al-Sadoun, H. B., and Makrucki, B. A. A Semi-Markov Model for Memory Interference in Multiprocessors. University of Michigan, Computing Research Lab. Tech. Rep. CRL-TR-44-84, Nov. 1984.