Performance Model for a Prioritized Multiple-Bus Multiprocessor System

Lizy Kurian John
Department of Computer Science and Engineering
University of South Florida
Tampa, FL 33620

and

Yu-cheng Liu
Department of Electrical and Computer Engineering
University of Texas at El Paso
El Paso, TX 79968

Abstract

The performance of a shared memory multiprocessor system with a multiple-bus interconnection network is studied in this paper. The effect of bus and memory contention is modeled using a probabilistic model and a closed form solution for the acceptance probability of each processor is presented. It is assumed that each processor in the system has a distinct priority assigned to it and that arbitration is based on priority. Whenever a request from a processor is rejected due to bus or memory conflicts, the request is resubmitted until granted. Based on the model, individual processor acceptance probabilities are first estimated, from which the effective memory bandwidth is computed. The accuracy of the analytical model is verified based on simulation results. Results from the model are compared against other approximate models previously reported in literature. It is observed that the inaccuracy of the model measured in terms of error from simulation results is less than that in previously reported studies.

Keywords: Acceptance Probability, Effective Memory Bandwidth, Interconnection Network, Multiple-Bus, Multiprocessor, Performance Modeling and Evaluation.

1 Introduction

A variety of interconnection networks have been used for interconnecting processors and memories in shared memory multiprocessor systems [8]. This paper presents an analytical model for one of these, the multiple-bus interconnection network. A multiple-bus system consists of N processors $P_0, P_1, P_2, ...P_{N-1}$ (each may have its own private cache, local memory and local I/O), connected to a set of M memory modules $M_0, M_1, ...M_{M-1}$ by B buses $B_0, B_1, ...B_{B-1}$ as in Fig. 1. Memory and bus conflicts in these systems are typically resolved by two-stage arbitration schemes which employ policies of random choice, daisy chaining, round-robin, fixed-priority, or rotating priority. In two-stage arbitration, memory conflicts are resolved first, by M 1-of-N arbiters, one per memory bank. The processor requests selected by the memory arbiters are then allocated a bus by a B-of-M arbiter.

The performance of various multiprocessor configurations have been studied widely in the past using analytical models and simulations. One class of analytical solutions [1] [9] [16] use Markov chains, queueing networks or Petri nets and yields very accurate results. However, when systems become large, these models become prohibitively complex to analyze. Another class of analytical solutions employs probabilistic or combinatorial techniques and derive approximate solutions using simplified assumptions [2] [5] [7] [13] [14] [18] [17] [20] [24]. Most of these solutions [2] [5] [6] [7] [13] [20] assume independent requests in each cycle and ignore resubmission of rejected requests. For instance, Ravi [20], Chang, Kuck and Lawrie [5], Bhuyan [2], Das and Bhuyan [7], Liu and Jou [13] etc. derived simple closed-form equations for calculating the memory bandwidth of crossbar and/or multiple-bus systems, but with the assumption that rejected requests are ignored. The errors in these solutions, approximately 10% in some cases, and the complexity of exact Markov models prompted researchers [14] [15] [24] to develop probabilistic models which consider resubmission of rejected requests. For instance, Yen, Patel and Davidson [24] considered a model for crossbar networks in which the blocked processors resubmit their requests to the same memories in the next cycle. Similarly, Liu and Wang [14] considered the resubmission of rejected requests and obtained an expression for the probability of acceptance of each processor in a prioritized crossbar system. They also derived the effective memory bandwidth from the probability of acceptance and showed that the prioritized model yields similar effective memory bandwidth results as the steady-state flow method [24], for crossbar networks. Yen et. al. [24] and Liu and Wang [14] consider crossbar systems only. More recently, Mahmud [15] used steady-state flow approach, to study multiple-bus systems in which processors and memories are organized as clusters. In this paper, we develop a probabilistic model for multiple-bus systems, considering resubmission of rejected requests. Based on the model, individual processor acceptance probabilities are first estimated, from which the effective memory bandwidth is computed. Our analysis of effective memory bandwidth yields a closed-form solution. By contrast, the solution in [15] uses steady-state flow approach [24], which therefore requires solving a nonlinear equation by iterative improvements.

This paper presents an analytical solution for processor acceptance probabilities and effective memory bandwidths of prioritized multiple-bus multiprocessor systems considering resubmitted requests, and compares the analytical results with simulation data and other analytical solutions previously reported in literature. In Section 2, the prioritized multiple-bus model assumptions are described. In Section 3, an analytical model for acceptance probability of each processor is developed first. Then a preliminary equation to compute effective memory bandwidth based on the acceptance probabilities and the static request rate of the processors is derived. Finally, the increase in request rate due to resubmission is considered, a dynamic request rate is evaluated, and the bandwidth equation is accordingly modified to improve the accuracy. In Section 4, the results from our analytical model is compared with simulation results and analytical models previously reported in literature. Section 5 offers concluding remarks.

2 Assumptions

As in most of the previous studies, our analysis is also based on the homogeneous model which assumes that each processor has the same rate R for generating new requests which are uniformly distributed among all memory modules. The assumptions of the model in this paper are stated below:

1. Each processor in the system is assigned a distinct priority. Processor P_0 has the highest priority and P_{n-1} has the lowest priority. Processor P_0 can never be blocked. Processor P_k can be blocked only by those processors with higher priority viz. $P_0, P_1, \ldots . P_{k-1}$.

2. At the beginning of each memory cycle each processor generates a new request with a probability R. Thus R is also the average number of new requests generated per cycle by each processor. Processors that have been denied service in the previous cycle will not generate a new request, but will resubmit the previous request.

3. The processors are synchronized, i.e., the processors issue requests at the beginning of a cycle.

4. The requests are uniformly distributed over the memory modules.

5. The memory cycle time is a constant. The propagation delays and arbitration times are considered as part of the cycle time and are not modeled explicitly.

6. During a cycle, the requests to each memory module are resolved on the basis of processor priority and the processor with the highest priority is chosen from those currently requesting the module. Then buses are assigned to B of those selected processors which have higher priorities and the other processors requests are ignored.

3 An Analytical Model

Effective memory bandwidth, acceptance probability, blocking probability, and throughput are popular metrics used to indicate the performance of multiprocessor systems. Effective memory bandwidth (EMBW) is defined as the average number of memory modules remaining busy in a cycle. The acceptance probability of a processor is defined as the probability that the request from that processor will be accepted in any cycle. The blocking probability of a processor is the complement of acceptance probability, i.e., the probability that the processor's request is blocked either due to bus or memory contention. In fact, all these measures are interrelated and the one particularly favored in past studies is EMBW. We use EMBW and acceptance probability in our performance evaluation. Acceptance probability is particularly relevant in the study of a prioritized system, because the performance of each individual processor can be studied. EMBW is also chosen because (i) it is a very popular performance metric, (ii) extensive results from previous studies are available for comparison, and (iii) it serves as a yardstick to verify the accuracy of the new models developed in this paper.

In Section 3.1, we describe Stirling numbers [11] [13] [14], which are used in deriving our solution. In Section 3.2, we develop an analytical model for the probability of acceptance of each processor in the system. In Section 3.3, an equation to compute EMBW from the acceptance probabilities is presented.

3.1 Stirling Numbers

As noted in [20], the number of ways of placing n distinct objects into k distinct cells with no cell left empty is equal to

$$\sum_{i=0}^{k}(-1)^i C(k,i)(k-i)^n$$

which is represented as $k!S(n.k)$ where $S(n,k)$ is called the Stirling number of the second kind and is given by

$$S(n,k) = \sum_{i=0}^{k} \frac{(-1)^i C(k,i)(k-i)^n}{k!} \qquad \ldots \qquad (1)$$

It follows that the number of ways of placing r distinct objects into n nondistinct cells with no cell left empty is equal to $S(r,n)$. The Stirling numbers of the second kind satisfy the recurrence relation

$$S(n,k) = kS(n-1,k) + S(n-1,k-1) \qquad for\ 1 < k < n-1 \quad \ldots \qquad (2)$$

3.2 Probability of Acceptance

Let PA(k) represent the acceptance probability of processor P_k where k is the assigned priority in any cycle after the system has reached equilibrium. Probability of acceptance PA(0) is always equal to 1. Let E_i be the event that processor P_i does not block processor P_k for $0 \leq i \leq k$. For $k \geq 1$, PA(k) can be expressed as

$$\begin{aligned}\text{PA(k)} &= \text{Prob}(E_0 \bigcap E_1 \bigcap E_2 \bigcap \ldots\ldots \bigcap E_{k-1}) \\ &= \text{Prob}(E_0) X Prob(E_1 \mid E_0) X Prob(E_2 \mid E_0 \bigcap E_1) X Prob(\text{E}_{k-1} \mid E_0 \bigcap E_1 \bigcap \ldots \bigcap E_{k-2}) \\ &= \text{PA(k-1) Prob } (E_{k-1} \mid E_0 \bigcap E_1 \bigcap \ldots \bigcap E_{k-2})\end{aligned}$$

Let a_i denote the probability that processor P_i makes a new request and blocks processor P_{i+1} due to memory or bus contention, provided that processors with a priority higher than P_i do not block it, i.e., if $P_0, P_1, \ldots, P_{i-1}$ do not block P_{i+1}. Let b_i be the probability that P_i resubmits a rejected request and blocks P_{i+1}. Note that b_0 is always 0, since processor P_0 is never blocked. Thus,

$$\begin{aligned}\text{PA(1)} &= \text{PA(0) Prob}(E_0) \\ &= \text{PA(0) [1 -}a_0]\end{aligned}$$

$$\begin{aligned}\text{PA(2)} &= PA(0) Prob(E_0) X Prob(E_1 \mid E_0) \\ &= PA(1) Prob(E_1 \mid E_0) \\ &= \text{PA(1) [1-Prob(}P_1 \text{ blocks } P_2 \text{ provided that } P_0 \text{ does not block } P_2)] \\ &= \text{PA(1) [1-Prob(} P_1 \text{ issues a new request and blocks } P_2) \text{ -} \\ &\qquad\qquad \text{Prob}(P_1 \text{ resubmits a rejected request and blocks } P_2)] \\ &= \text{PA(1) [1 - } a_1 \text{ - } b_1]\end{aligned}$$

⋮

In general,

$$\begin{aligned}\text{PA(k)} &= \text{PA(k-1) [1-Prob(}P_{k-1} \text{ blocks } P_k \text{ given that} \\ &\qquad\qquad P_0, P_1, \ldots P_{k-2} \text{ do not block } P_k)] \\ &= \text{PA(k-1) [1 -(Prob that } P_{k-1} \text{ issued a new request} \\ &\qquad\qquad \text{and blocked } P_k) \text{ - (Prob that } P_{k-1} \\ &\qquad\qquad \text{resubmitted a rejected request and blocked } P_k)] \\ &= \text{PA(k-1) [1 - } a_{k-1} \text{ - } b_{k-1}]\end{aligned}$$

In the above equation, term a_{k-1} considers the effect of new requests issued by higher priority processors and b_{k-1} represents the effect of rejected requests which are resubmitted. Whenever a request is rejected, it is resubmitted until the request is granted. Until that time, obviously, the processor does not issue any new requests. In any cycle, the processor submits a maximum of one request, either a new request or a resubmitted request. Hence we modify the above equation to account the fact that there are no new requests in the cycles when there is a resubmitted request. This term is properly accounted by the product of a_{k-1} and b_{k-1}, i.e., by the factor $a_{k-1}b_{k-1}$. Hence,

$$PA(k) = PA(k-1)[1 - a_{k-1} - b_{k-1} + a_{k-1}b_{k-1}] \quad \ldots \tag{3}$$

where a_{k-1} and b_{k-1} are derived as follows.

Derivation of a_i

Now we have to determine $a_0, a_1, \ldots a_{k-1}$, where

a_0 = Prob(P_0 blocks P_1)

a_1 = Prob(P_1 issues a new request and blocks P_2 given that P_0 does not block P_2)

$\vdots$

a_{k-1} = Prob(P_{k-1} issues a new request and blocks P_k given that $P_0, P_1, ...P_{k-2}$ do not block P_k.

Let A represent the event that P_0 does not block P_2. And let B denote the event that P_1 does not block P_2 and P_0 does not block P_2. Then, a_1 is represented by the conditional probability Prob($B \mid A$). This conditional probability may be expressed in terms of the joint probability Prob($A \bigcap B$) and the marginal probability Prob(A). Thus we have

$$a_1 = Prob(B \mid A)$$

$$= \frac{Prob(B \bigcap A)}{Prob(A)}$$

$$= \frac{Prob(A) - Prob(B' \bigcap A)}{Prob(A)}$$

$$= 1 - \frac{Prob(B' \bigcap A)}{Prob(A)}$$

$$= 1 - \frac{Prob\ (P_0\ \ and\ P_1\ \ do\ not\ block\ P_2)}{Prob\ (P_0\ \ does\ not\ block\ P_2)}$$

Now redefining A and B as the events

A : P_0 and P_1 do not block P_3 and

B : P_2 blocks P_3

and proceeding in a similar fashion, we obtain

$$a_2 = 1 - \frac{Prob(B' \bigcap A)}{Prob(A)}$$

$$=1 - \frac{Prob\ (P_0,\ P_1\ \ and\ P_2\ \ do\ not\ block\ P_3)}{Prob\ (P_0\ \ and\ P_1\ \ do\ not\ block\ P_3)}$$

In general,

$$a_{k-1} = 1 - \frac{Prob\ (P_0,P_1,P_2...P_{k-1}\ \ do\ not\ block\ P_k)}{Prob\ (P_0,P_1,P_2...P_{k-2}\ \ do\ not\ block\ P_k)}$$

Now we have to determine the probability that $P_0, P_1, P_2, ...P_{k-1}$ do not block P_k. Bus and memory contentions have to be considered in order to determine this probability. ProcessorP_k gets blocked if the higher priority processors requested the same memory module, or they used up all the B buses in the system. As first explained in [20], the probability that j memory modules are requested from (M-1) modules by i processors is

$$\frac{j!S(i,j)\begin{pmatrix} M-1 \\ j \end{pmatrix}}{M^i} \qquad \ldots \qquad (4)$$

where S(i,j) is the Stirling number of the second kind. If a request is made in every cycle, the probability that P_k is not blocked by i processors with higher priorities by using up all the B buses is

$$\frac{\sum_{j=1}^{B-1} j!S(i,j)\begin{pmatrix} M-1 \\ j \end{pmatrix}}{M^i} \qquad \ldots \qquad (5)$$

When the request rate of the processor is less than 1.0, we need to consider the probability of exactly i processors making a request. Hence the probability that P_k is not blocked by i higher priority processors is

$$\begin{pmatrix} k \\ i \end{pmatrix} R^i(1-R)^{k-i}\frac{\sum_{j=1}^{B-1} j!S(i,j)\begin{pmatrix} M-1 \\ j \end{pmatrix}}{M^i} \qquad \ldots \qquad (6)$$

Hence the probability that $P_0, P_1, P_2 \ldots P_{k-1}$ do not block P_k is

$$\sum_{i=0}^{k} \begin{pmatrix} k \\ i \end{pmatrix} R^i(1-R)^{k-i}\frac{\sum_{j=1}^{B-1} j!S(i,j)\begin{pmatrix} M-1 \\ j \end{pmatrix}}{M^i}$$

We can similarly obtain the probability that $P_0, P_1, P_2, P_3 \ldots\ldots P_{k-2}$ do not block P_k as

$$\sum_{i=0}^{k-1} \begin{pmatrix} k-1 \\ i \end{pmatrix} R^i(1-R)^{k-1-i}\ \frac{\sum_{j=1}^{B-1} j!S(i,j)\begin{pmatrix} M-1 \\ j \end{pmatrix}}{M^i}$$

Therefore,

$$a_{k-1} = 1 - \left[\frac{\sum_{i=0}^{k} \binom{k}{i} R^i (1-R)^{k-i} \frac{\sum_{j=1}^{B-1} j! S(i,j) \binom{M-1}{j}}{M^i}}{\sum_{i=0}^{k-1} \binom{k-1}{i} R^i (1-R)^{k-1-i} \frac{\sum_{j=1}^{B-1} j! S(i,j) \binom{M-1}{j}}{M^i}} \right] \quad \ldots \qquad (7)$$

This equation may be mathematically manipulated to obtain a_{k-1} as equal to $\frac{R}{M}$ for the crossbar case, and hence for crossbar if resubmitted requests are not considered,

$$PA(k) = PA(k-1)[1 - \frac{R}{M}]$$

which has also been derived in previous literature [14] using other methods.

<u>Derivation of b_i</u>

The probability that P_1 issues a new request and blocks P_2 is equal to a_1; hence the probability that P_1 was blocked in the previous cycle and hence resubmits a rejected request in the current cycle is

$$b_1 = a_1(1 - PA(1)).$$

Similarly, we know that a_2 is the probability that P_2 issues a new request and blocks P_3. To determine b_2, we need to know the patterns in which P_2 gets blocked. Processor P_2 could be blocked by P_0 or P_1 or both. However, in calculating b_2, which is the probability that P_2 resubmits a rejected request and blocks P_3 given that P_0 and P_1 do not block P_3, we need to include only the cases of P_2 being blocked by either P_0 or by P_1 but not by both. As explained in [14], this is because if P_2 was blocked by P_0 and P_1 in the previous cycle, then P_1 must also resubmit the same request, thus implying that P_1 also blocks P_3 in the current cycle. The probability with which P_2 resubmits a rejected request and blocks P_3, given that P_0 and P_1 do not block P_3 may now be given as

$$b_2 = a_2 \ \ X \ \ [1 - PA(2)] X \frac{\binom{2}{1}(a_2)(1-a_2)}{1-(1-a_2)^2} \qquad \ldots \qquad (8)$$

where $\frac{\binom{2}{1}(a_2)(1-a_2)}{1-(1-a_2)^2}$ represents the probability that P_2 will be blocked only by P_0 or by P_1, given that P_2 is blocked.

Following the above analysis, we may obtain the expression for the general case as follows:

Probability that P_{k-1} resubmits a rejected request and blocks P_k is

$$a_{k-1} \;\; X \;\; [1 - PA(k-1)] X \frac{Prob\ P_{k-1}\ \text{is}\ blocked\ by\ exactly\ one\ processor}{Prob\ P_{k-1}\ \ is\ blocked\ by\ any\ processor}$$

where a_{k-1} is the probability that P_{k-1} issues a new request and blocks P_k.

The probability that P_{k-1} is blocked by exactly one processor is,

$$\binom{k-1}{1} a_{k-1}(1 - a_{k-1})^{k-2}$$

and the probability that P_{k-1} is blocked by any processor is $1 - (1 - a_{k-1})^{k-1}$. Therefore

$$b_{k-1} = a_{k-1} \;\; X \;\; [1 - PA(k-1)] \;\; X \frac{\binom{k-1}{1} a_{k-1}(1 - a_{k-1})^{k-2}}{1 - (1 - a_{k-1})^{k-1}} \qquad \ldots \qquad (9)$$

Note that equation 9 is reduced to $\left(\frac{R}{M}\right)^2$ for crossbar systems. When applying equation 3 to crossbar configurations, the inclusion of the term $a_{k-1}b_{k-1}$ eliminates the major problem found in the solution given in [14] for the special case of an $N \times 1$ crossbar with $N \geq 3$. For this case, solution in [14] gives

$$PA(2) = PA(1)(1 - R - R^2)$$

which becomes an invalid negative acceptance probability when $R \geq 0.62$. By contrast, equation 3 yields

$$PA(2) = PA(1)[1 - R - R^2 + R^3]$$

which reduces to 0 as R approaches to 1, but never becomes negative.

3.3 Effective Memory Bandwidth (EMBW)

EMBW is defined as the expected number of accepted requests from all processors in each cycle. We use acceptance probability to evaluate EMBW. Since PA(k) is the acceptance probability of processor P_k, $R(k)PA(k)$ represents the contribution of the particular processor P_k towards system memory bandwidth. The effective memory bandwidth of the whole system is obtained by summing up the contributions from all processors. Hence

$$EMBW = \sum_{k=0}^{N-1} R(k)PA(k) \qquad \ldots \qquad (10)$$

where $R(k)$ is the request rate of processor P_k.

The request rate R is defined as the average number of requests per cycle, which is also the probability of a processor making a new request in a given cycle. This request rate R does not reflect the resubmission of rejected requests. It has been shown by Hwang and Briggs [10] that an adjusted request rate considering the resubmitted requests also as new requests can be derived based on a Markov graph. This modified request rate is called dynamic request rate and is denoted by R'. This dynamic request rate of a processor with priority number k is represented by $R'(k)$. Following the derivation of dynamic request rate given in [10], it can be shown that

$$R'(k) = \frac{R}{R + PA(k)(1-R)} \qquad \ldots \qquad (11)$$

where $R'(k)$ is the dynamic request rate of P_k,
R is the static request rate, and
PA(k) is its probability of acceptance.

It may be noted that when request rate R is equal to 1, R' also equals 1, since Equation 11 reduces to R/R = 1. This can be explained rather easily; a processor can submit a maximum of 1 request in a cycle, either a new request or a resubmitted request.

In a prioritized system, each processor will have a different dynamic request rate even if all the processors may have the same static request rate. This is because the processors have different priorities and therefore each has an acceptance probability depending on its priority . Lower priority processors become blocked often and hence resubmit their requests frequently. Hence they have very high dynamic request rates. On the other hand, higher priority processors do not get blocked very often and resubmissions are fewer in number. Hence they have lower dynamic request rates relative to the lower priority processors. In fact $R'(k)$ represents the overall request rate if resubmission of a rejected request is also treated as a new independent request. This modification in request rate will be very prominent in systems where there are fewer buses since bus contention is quite high in such systems and it results in a large number of rejections and resubmissions.

Considering the dynamic request rate, effective memory bandwidth becomes

$$EMBW = \sum_{k=0}^{N-1} R'(k)PA(k) \quad \ldots \qquad (12)$$

where $R'(k)$ is the dynamic request rate of processor P_k.

It may be noted that the EMBW for a system of a particular size is independent of whether it is a prioritized or non-prioritized configuration. But the contributions from the individual processors are different in the two systems. In the non-prioritized system, all processors contribute equally whereas in a prioritized system, the contribution depends on the priority.

Effective memory bandwidth can be calculated using Equation 12 with probability of acceptance from Equation 3. Equation 3 slightly overestimates PA(k) values, and hence the EMBW calculated using Equation 12 may exceed the bus number for a configuration with only one or two buses. This problem is corrected by considering the computed value or B whichever is smaller.

4 Performance Comparison

In this section the results obtained from the models in this paper are compared against models from previous researchers and simulation results. A simulator for a prioritized multiple-bus system was developed in SIMSCRIPT II.5 [21]. A system with N processors, M memories and B buses is modeled with each processor assigned a distinct priority. The simulator accurately follows the model in Section 2. The program delivers the probability of acceptance of a request from each processor and the overall system bandwidth. Further details of the simulator including the source code may be found in [11].

Probability of Acceptance

Previous results on acceptance probability are available from Liu and Jou [13] only. Estimated values of acceptance probability PA(k) values from the proposed model and the model in [13] are compared for a few representative configurations. Figure 2 shows the percent errors for an 8 X 8 X 5 and 16 X 16 X 9 system. It may be observed that the proposed model is significantly more accurate than the previous model developed by Liu and Jou [13]. Another observation is that the model inaccuarcy increases with decrease in processor priority.

Another study is based on the PA(k) value of the lowest priority processor in the system, for various bus numbers. The lowest priority processor is chosen because that indicates the worst case performance. Fig. 3 shows the acceptance probability for the lowest priority processor in a 16 X 16 system, for $R = 0.2$ and $R = 0.5$. It is seen that the proposed model is better than the model in [13].

A third comparative study is based on the PA(k)s in a single bus system for various processor priorities. Fig. 4 shows the acceptance probability for 8 X 8 and 16 X 16 systems. It is again seen that the proposed model is better than the model in [13].

EMBW

Analytical results on EMBW are available from Bhuyan [2], Das and Bhuyan [7], Mudge et al. [18], Liu and Jou [13], etc. In Fig 5, we plot the percent error in EMBW for 16 X 16 and 8 X 8 systems for various bus numbers, for the different models. All previous models except Mudge et al's iterative model had higher errors than the proposed model. Even Mudge's iterative model yields low errors for $R = 0.5$, but for request rate $R = 1$, the proposed model is significantly better than that model. In the crossbar case, at R=1.0, the inaccuracy reduces from 6 or 7% to less than 0.5% for both 16 X 16 and 8 X 8 systems. The bandwidth and percentage errors for 8 X 8 X B and 16 X 16 X B systems for various bus numbers, from our model and results from previous researchers are also presented in Table 1, Table 2, Table 3, and Table 4, in Appendix I.

Fig 5 illustrates the results for only request rates 0.5 and 1.0. It would be interesting to study the performance at other request rates as well. Results from [13] are available at all request rates, and hence we perform a comparison. The variation of error in EMBW with respect to request rate is illustrated in Fig 6 for 16 X 16 X 8 and 16 X 16 X 12 systems. It may be observed that the proposed model performs better than the existing model in [13]. It is evident from Appendix I that all models except Mudge's iterative model give similar results as in [13]. Hence the comparison in Fig 6 should hold for those models as well. More detailed comparison for more cases are presented in [11].

5 Summary and Conclusion

We derived an analytical model for multiple-bus multiprocessor systems. In most of the approximate models in the past, for reason of simplicity, rejected requests are ignored. Our model considers resubmission of rejected requests and hence the results are closer to ideal (simulation) results than those reported in previous studies. Since crossbar is a special case of multiple-bus, the model can be used for crossbar systems as well.

References

[1] D. P. Bhandarkar, "Analysis of memory interference in multiprocessors", IEEE Transactions on Computers, Vol. C-24, No.9, pp. 897-908, September 1975.

[2] L. N. Bhuyan, "An analysis of processor memory interconnection networks", IEEE Transactions on Computers, Vol. C-34, No. 3, pp. 279-283, March 1985.

[3] L. N. Bhuyan, Q. Yang, and D. P. Agrawal, "Performance of multiprocessor interconnection networks", Computers, Vol. 22, No.2, pp. 25-37, February 1989.

[4] L. N. Bhuyan and D. P. Agrawal, "Design and performance of generalized interconnection networks", IEEE Transactions on Computers, Vol. C-32, pp. 1081-1090, December 1983.

[5] D. Y. Chang, D. J. Kuck, and D. H. Lawrie, "On the effective bandwidth of parallel memories", IEEE Transactions on Computers, Vol. C-26, No. 5, pp. 480-489, May 1977.

[6] W. T. Chen and J. P. Shen, "Performance analysis of multiple bus interconnection networks with hierarchical requesting model", IEEE Transactions on Computers, Vol. 40, No. 7, pp. 834–842, July 1991.

[7] C. R. Das and L. N. Bhuyan, "Bandwidth availability of multiple bus multiprocessors", IEEE Transactions on Computers, Vol. C-34, No. 10, pp. 918-926, October 1985.

[8] T. Y. Feng, "A Survey of interconnection networks", IEEE Computer, Vol. 14, No. 12, pp. 12-27, December 1981.

[9] M. A. Holliday and M. K. Vernon, "Exact performance estimates for multiprocessor memory and bus interference", IEEE Transactions on Computers, Vol. C-36, No. 1, pp. 76-85, January 1987.

[10] K. Hwang, and F. A. Briggs, Computer Architecture and Parallel Processing, McGraw Hill, New York, 1984.

[11] L. Kurian, "Performance Evaluation of Prioritized Multiple-Bus Multiprocessor Systems", M.S. Thesis, Dept of Electrical Engr, University of Texas at El Paso, December 1989.

[12] T. Lang, M. Valero and I. Alegre, "Bandwidth of crossbar and multiple-bus connections for multiprocessors", IEEE Transactions on Computers, Vol. C-31, No.12, pp. 1227-1234, December 1982.

[13] Y. C. Liu and C. J. Jou, “Effective Memory Bandwidth and Processor Blocking Probability in multiple-bus systems”, IEEE Transactions on Computers, Vol. C-36, No. 6, pp. 761-764, June 1987.

[14] Y. C. Liu and C. C. Wang, “Analysis of prioritized crossbar multiprocessor Systems”, Journal of Parallel and Distributed Computing, 7, pp. 321-334, October 1989.

[15] S. M. Mahmud, “Performance of multilevel bus networks for hierarchical multiprocessors”, IEEE Transactions on Computers, Vol. 43, No. 7, pp. 789 – 805, July 1994.

[16] M. A. Marsan, G. Balbo, G. Conte and F. Gregoretti, “Modeling bus contention and memory interference in a multiprocessor system”, IEEE Transactions on Computers, Vol. C-32, No. 1, pp. 60-72, January 1983.

[17] T. N. Mudge, J. P. Hayes, G. D. Buzzard and D. C. Winsor, “Analysis of multiple-bus interconnection Networks”, Journal of Parallel and Distributed Computing, 3, pp. 328-343, March 1986.

[18] T. N. Mudge, J. P. Hayes, G. D. Buzzard and D. C. Winsor, “Analysis of multiple-bus interconnection Networks”, Proceedings of the 1984 Conference on Parallel Processing, pp. 228-232, August 1984.

[19] T. N. Mudge and H. B. Al-Sadoun, “A Semi-Markov Model for the Performance of Multiple-Bus Systems”, IEEE Transactions on Computers, Vol. C-34, Number 10, pp. 934-942, October 1985.

[20] C. V. Ravi, “On the bandwidth and interference in interleaved memory systems”, IEEE Transactions on Computers, Vol. C-21, No. 8, pp. 899-901, August 1972.

[21] E. C. Russel, “Building Simulation Models with SIMSCRIPT II.5”, CACI Products Company, 3333 North Torrey Pines Ct, La Jolla, California 92037.

[22] D. Towsley, “Approximate models of multiple-bus multiprocessor systems”, IEEE Transactions on Computers, Vol. C-35, No. 3, pp. 220-228, March 1986.

[23] Q. Yang and S. Zaky, “Communication performance in multiple-bus systems”, IEEE Transactions on Computers, Vol. C-37, No. 7, July 1988.

[24] D. W. L. Yen, J. H. Patel, and E. S. Davidson, “Memory Interference in Synchronous Multiprocessor Systems”, IEEE Transactions on Computers, Vol. C-31, No. 11, pp. 1116-1121, November 1982.

Appendix I

	Bus No.	Bhuyan	Das and Bhuyan, Mudge noniterative	Mudge iterative	Liu and Jou	Proposed	Simulation
EMBW	1	1.00	1.00	1.00	1.00	1.0000	1.0000
	2	2.00	2.00	2.00	2.00	2.0000	2.0000
	3	2.99	2.98	3.00	2.99	3.0000	3.0000
	4	3.97	3.91	3.99	3.97	4.0000	4.0000
	5	4.86	4.74	4.95	4.86	5.0000	5.0000
	6	5.57	5.41	5.79	5.57	6.0000	5.8749
	7	6.04	5.87	6.37	6.04	6.6281	6.4514
	8	6.27	6.15	6.71	6.27	6.8444	6.7216
	9	6.35	6.29	6.88	6.35	6.9196	6.8094
	10	6.37	6.35	6.95	6.37	6.9381	6.8286
	11	6.37	6.37	6.98	6.37	6.9411	6.8638
	12	6.37	6.37	6.98	6.37	6.9414	6.8550
	13	6.37	6.37	6.98	6.37	6.9414	6.8427
	14	6.37	6.37	6.98	6.37	6.9414	6.8366
	15	6.37	6.37	6.98	6.37	6.9414	6.8670
	16	6.37	6.37	6.98	6.37	6.9414	6.8329
% Error	1	0.00	0.00	0.00	0.00	0.00	
	2	0.00	0.00	0.00	0.00	0.00	
	3	-0.30	-0.67	0.00	-0.30	0.00	
	4	-0.80	-2.25	-0.25	-0.80	0.00	
	5	-2.40	-4.82	-0.60	-2.40	0.20	
	6	-4.80	-7.52	-1.03	-4.80	2.13	
	7	-6.10	-8.71	-0.93	-6.10	2.74	
	8	-6.40	-8.21	0.15	-6.40	1.83	
	9	-6.90	-7.77	0.88	-6.90	1.62	
	10	-6.70	-7.03	1.76	-6.70	1.60	
	11	-6.70	-6.70	2.20	-6.70	1.13	
	12	-6.70	-6.70	2.20	-6.70	1.26	
	13	-6.87	-6.87	2.05	-6.87	1.44	
	14	-6.87	-6.87	2.05	-6.87	1.53	
	15	-6.87	-6.87	2.05	-6.87	1.08	
	16	-6.87	-6.87	2.05	-6.87	1.59	

Table 1: EMBW from various models and their percent errors (16 X 16 system, R = 0.5)

	Bus No.	Bhuyan	Das and Bhuyan, Mudge noniterative	Mudge iterative	Liu and Jou	Proposed	Simulation
EMBW	1	1.00	1.00	1.00	1.00	1.0000	1.0000
	2	2.00	2.00	2.00	2.00	1.9960	2.0000
	3	3.00	3.00	3.00	3.00	2.9892	3.0000
	4	4.00	4.00	4.00	4.00	3.9708	4.0000
	5	5.00	5.00	5.00	5.00	4.9335	5.0000
	6	6.00	5.99	5.99	6.00	5.8705	5.9999
	7	7.00	6.97	6.97	7.00	6.7773	6.9967
	8	7.99	7.89	7.89	7.99	7.6428	7.9627
	9	8.92	8.72	8.72	8.92	8.4245	8.7748
	10	9.66	9.39	9.39	9.66	9.0317	9.3234
	11	10.10	9.86	9.86	10.10	9.3896	9.5545
	12	10.26	10.13	10.13	10.26	9.5312	9.6250
	13	10.30	10.25	10.25	10.30	9.5629	9.6257
	14	10.30	10.29	10.29	10.30	9.5662	9.6264
	15	10.30	10.30	10.30	10.30	9.5663	9.6023
	16	10.30	10.30	10.30	10.30	9.5663	9.6130
% Error	1	0.00	0.00	0.00	0.00	0.00	
	2	0.00	0.00	0.00	0.00	-0.20	
	3	0.00	0.00	0.00	0.00	-0.36	
	4	0.00	0.00	0.00	0.00	-0.73	
	5	0.00	0.00	0.00	0.00	-1.33	
	6	0.00	-0.17	-0.17	0.00	-2.16	
	7	0.30	-0.14	-0.14	0.30	-3.14	
	8	0.90	-0.38	-0.38	0.90	-4.02	
	9	2.30	0.00	0.00	2.30	-3.99	
	10	4.20	1.29	1.29	4.20	-3.13	
	11	6.00	3.46	3.46	6.00	-1.73	
	12	6.80	5.41	5.41	6.80	-0.97	
	13	7.00	6.44	6.44	7.00	-0.68	
	14	7.00	6.85	6.85	7.00	-0.63	
	15	7.00	7.00	7.00	7.00	-0.37	
	16	7.00	7.00	7.00	7.00	-0.49	

Table 2: EMBW from various models and their percent errors (16 X 16 system, R = 1.0)

	Bus No.	Bhuyan	Das and Bhuyan, Mudge noniterative	Mudge iterative	Liu and Jou	Proposed	Simulation
EMBW	1	1.00	0.98	1.00	1.00	1.0000	1.0000
	2	1.94	1.88	1.98	1.94	2.0000	1.9988
	3	2.69	2.57	2.80	2.69	2.9820	2.8867
	4	3.09	2.99	3.27	3.09	3.3798	3.3346
	5	3.21	3.16	3.46	3.21	3.4889	3.4509
	6	3.23	3.22	3.51	3.23	3.5040	3.4634
	7	3.23	3.23	3.52	3.23	3.5048	3.4647
	8	3.23	3.23	3.52	3.23	3.5048	3.4647
% Error	1	0.00	-2.00	0.00	0.00	0.00	
	2	-2.94	-5.94	-0.94	-2.94	0.06	
	3	-6.81	-10.97	-3.00	-6.81	3.30	
	4	-7.34	-10.33	-1.94	-7.34	1.36	
	5	-6.98	-8.43	0.26	-6.98	1.10	
	6	-6.74	-7.03	1.35	-6.74	1.17	
	7	-6.77	-6.77	1.60	-6.77	1.16	
	8	-6.77	-6.77	1.60	-6.77	1.16	

Table 3: EMBW from various models and their percent errors (8 X 8 system, R = 0.5)

	Bus No.	Bhuyan	Das and Bhuyan, Mudge noniterative	Mudge iterative	Liu and Jou	Proposed	Simulation
EMBW	1	1.00	1.00	1.00	1.00	1.0000	1.0000
	2	2.00	2.00	2.00	2.00	1.9838	2.0000
	3	3.00	2.97	2.97	3.00	2.9435	2.9982
	4	3.98	3.87	3.87	3.98	3.8327	3.9556
	5	4.79	4.59	4.59	4.79	4.5304	4.6507
	6	5.18	5.04	5.04	5.18	4.8703	4.9116
	7	5.25	5.22	5.22	5.25	4.9356	4.9552
	8	5.25	5.25	5.25	5.25	4.9379	4.9553
% Error	1	0.00	0.00	0.00	0.00	0.00	
	2	0.00	0.00	0.00	0.00	-0.81	
	3	0.06	-0.94	-0.94	0.06	-1.82	
	4	0.62	-2.16	-2.16	0.62	-3.11	
	5	3.00	-1.31	-1.31	3.00	-2.59	
	6	5.46	2.61	2.61	5.46	-0.84	
	7	5.95	5.34	5.34	5.95	-0.40	
	8	5.95	5.95	5.95	5.95	-0.35	

Table 4: EMBW from various models and their percent errors (8 X 8 system, R = 1.0)

Figure Captions

Index Terms:
Acceptance Probability, Effective Memory Bandwidth, Interconnection Network, Multiple-Bus, Multiprocessor, Performance Modeling and Evaluation.

Preferred address for return of galley proofs

Dr. Lizy Kurian John
Computer Science and Engineering
4202 E. Fowler Ave, ENB 118
The University of South Florida
Tampa, FL 33620
Phone: (813)-974-2114
Fax: (813)-974-5456
e-mail: kurian@csee.usf.edu

List of footnotes

Lizy Kurian John is with the Department of Computer Science and Engineering at the University of South Florida, Tampa, FL 33620.

Yu-cheng Liu is with the Electrical and Computer Engineering Department at the University of Texas at El Paso, El Paso, TX 79968.

A preliminary version of this paper was presented at the 1994 Symposium on Parallel and Distributed Processing, October 1994.

Lizy Kurian John (S'89-M'93) received the B.Sc Engineering Degree in Electronics and Communication from the University of Kerala, India in 1984, the M.S. degree in Computer Engineering from the University of Texas at El Paso in 1989 and the Ph. D degree in Computer Engineering from the Pennsylvania State University in 1993.

She is currently an Assistant Professor of Computer Science and Engineering at the University of South Florida, Tampa. Prior to joining graduate school in 1988, she worked in the Indian Space Research Organization for 4 years. Her current research interests include high performance processor and memory systems, compiler optimization techniques, dynamic computer architectures using Field Programmable Gate Arrays (FPGAs), rapid prototyping etc.

She is a member of the ACM, ACM SIGARCH, SIGMETRICS and SIGMICRO. She is also a member of Eta Kappa Nu, Tau Beta Pi and Phi Kappa Phi honor societies.

Yu-cheng Liu (M'70) received the B. S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1965, and the M. S. degree in mathematics and the Ph. D. degree in electrical engineering, both from Nothwestern University, Evanston, Illinois, in 1967 and 1970, respectively. He is currently a professor in the Department of Electrical and Computer Engineering at the University of Texas at El Paso, El Paso, Texas. His areas of interest include multiprocessor interconnection networks, computer architecture, and microprocessor-based systems design. He has authored or coauthored three books on microprocessors, all published by Prentice Hall.

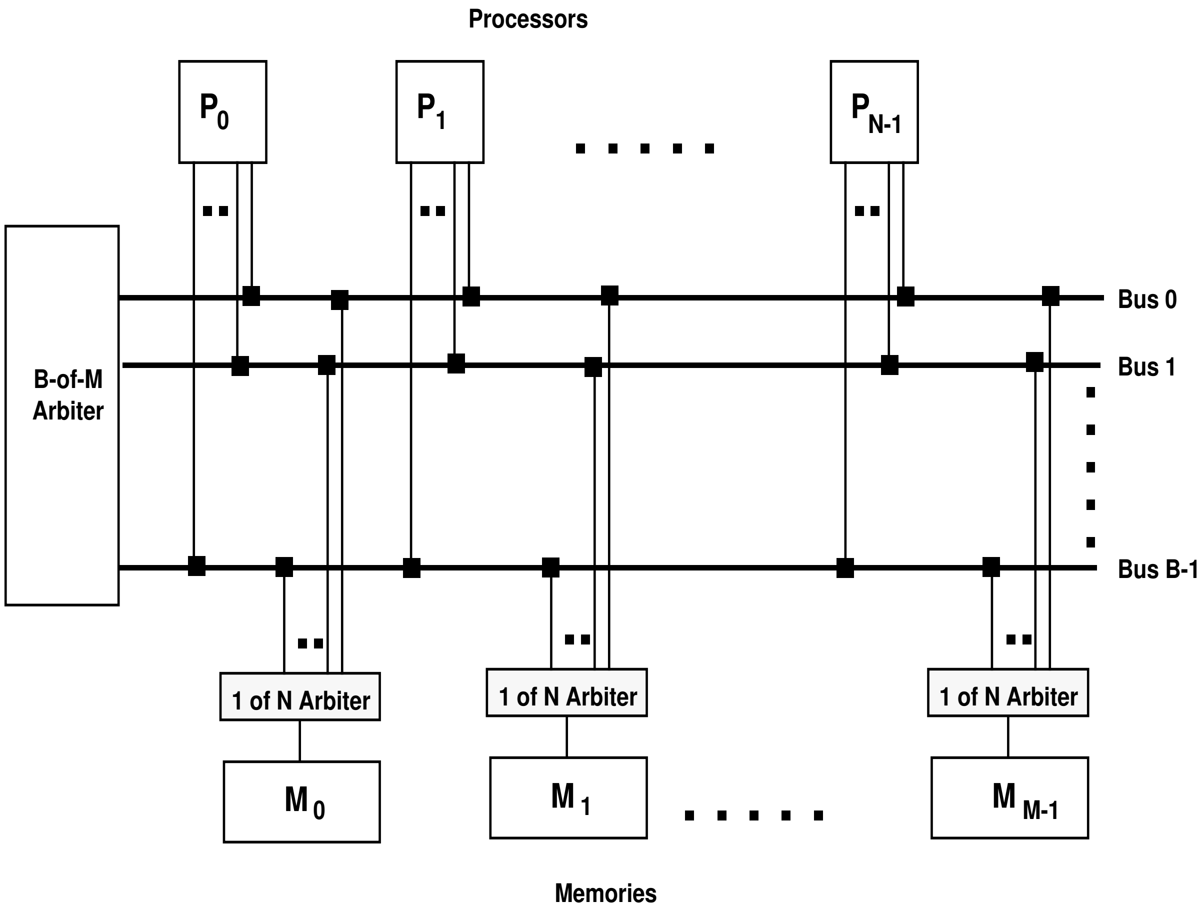


Figure 1: A multiple bus system with N processors, M memory modules and B buses

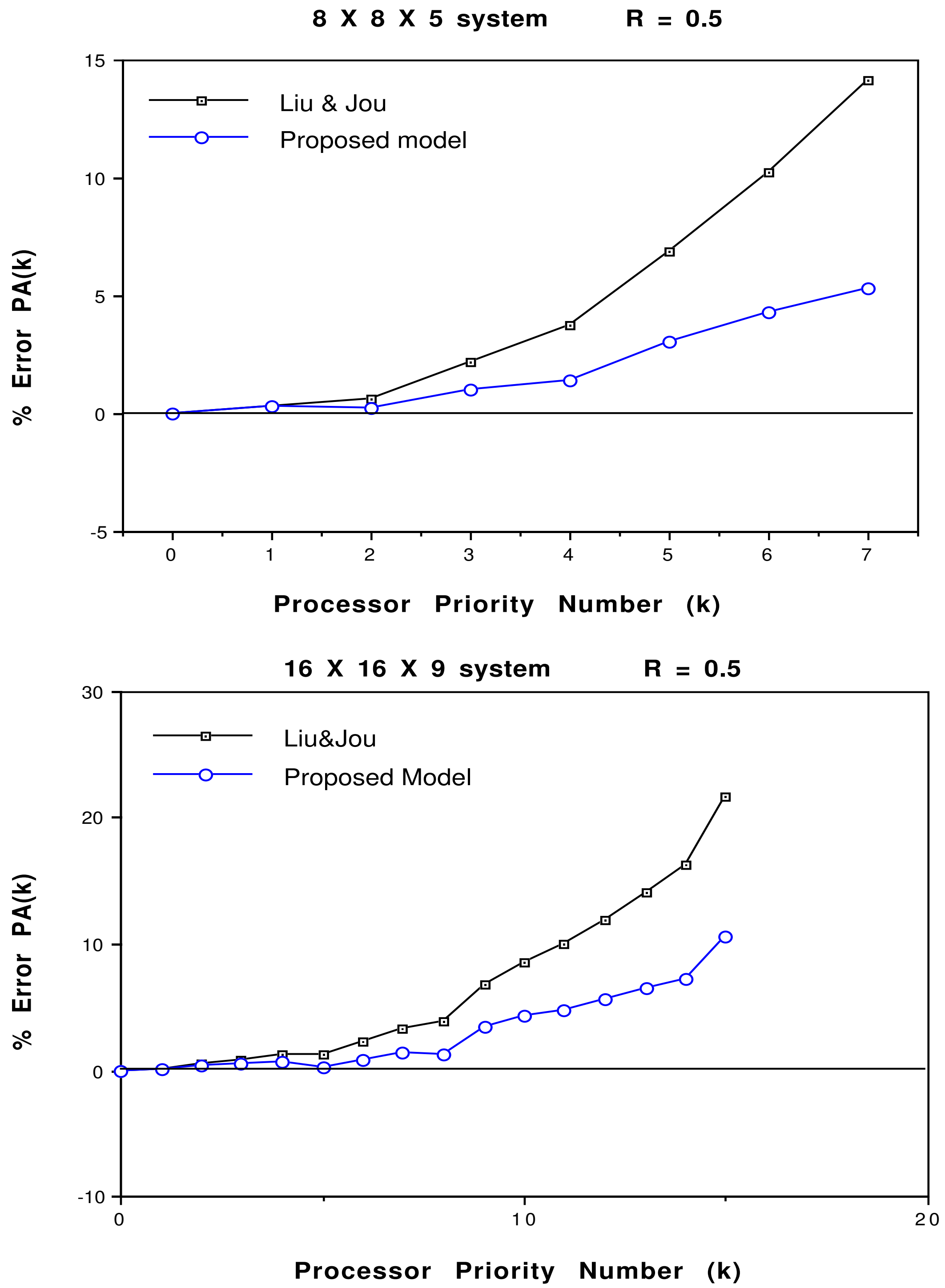


Figure 2: Percentage Error in Acceptance Probability vs Processor Priority

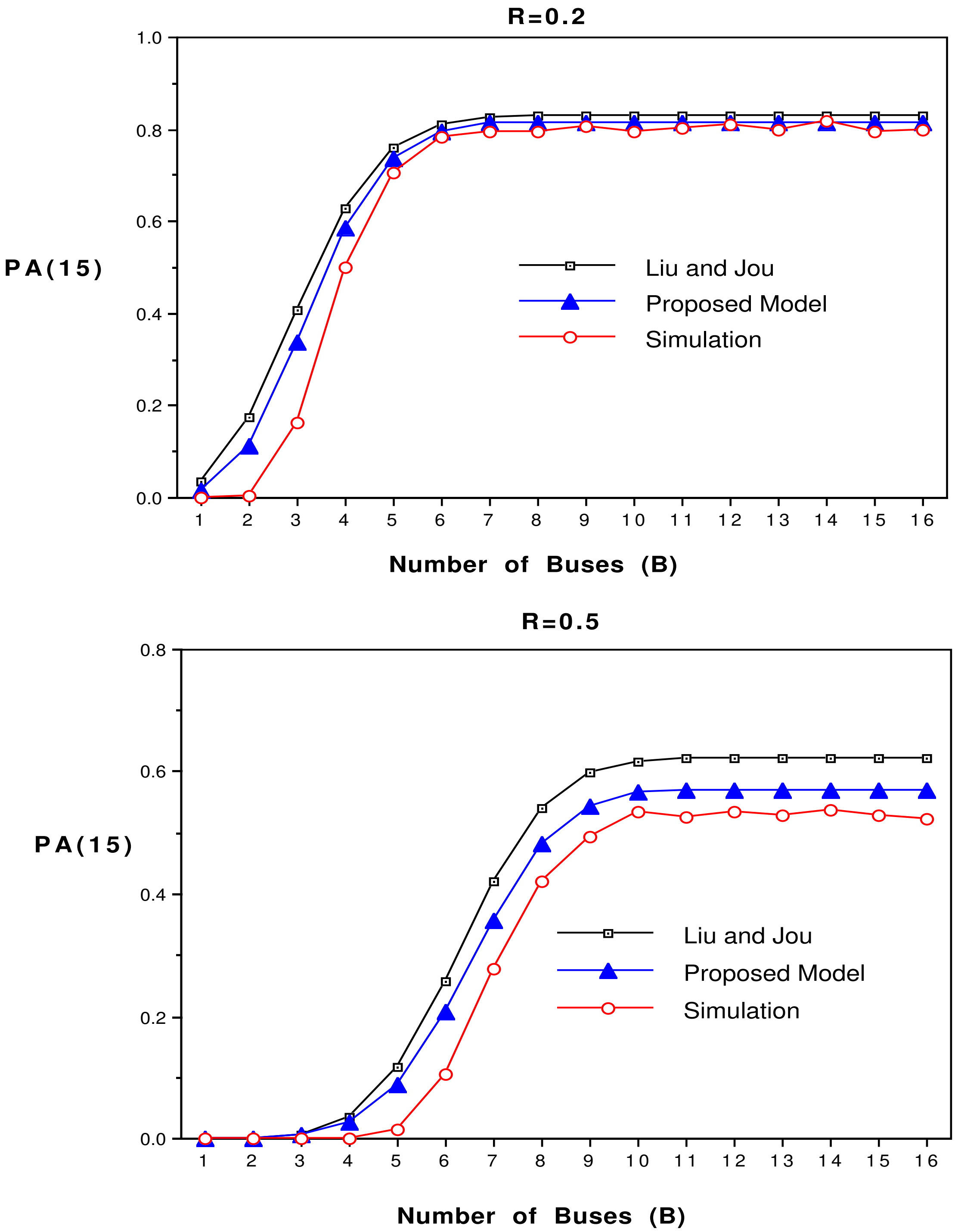
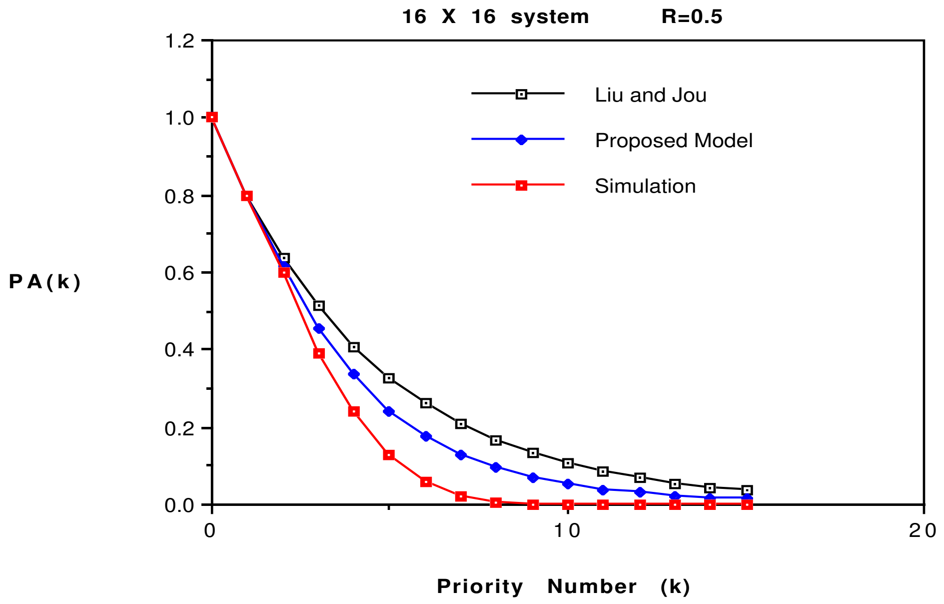


Figure 3: Acceptance Probability variation with Number of Buses



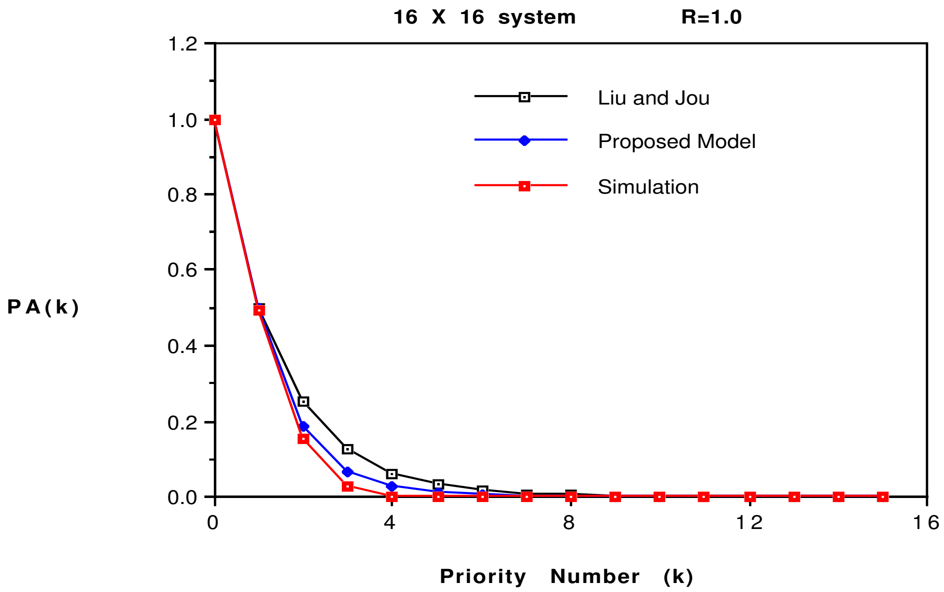


Figure 4: Acceptance Probability vs Processor Priority

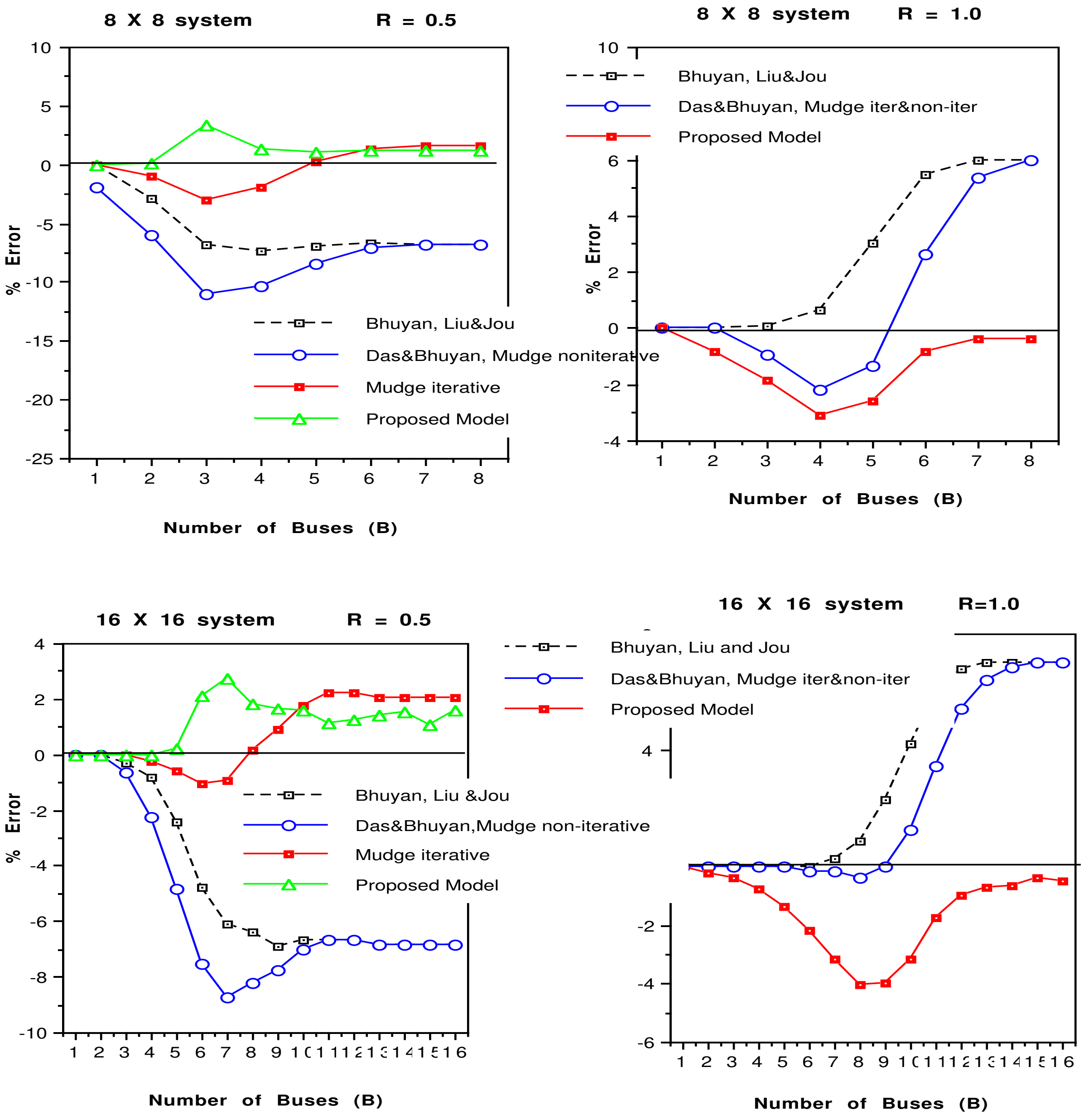


Figure 5: EMBW - Comparison of proposed model with previous studies

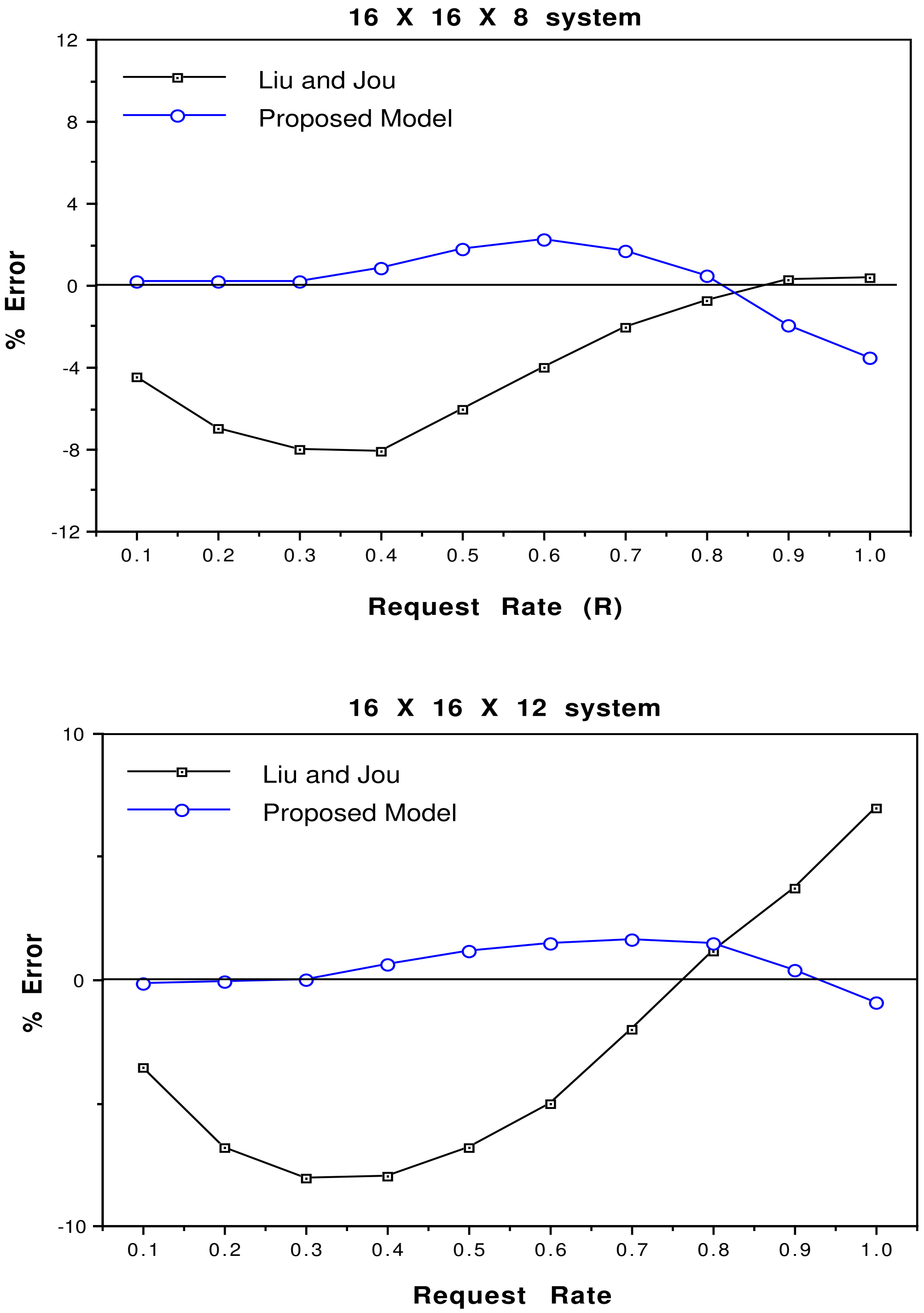


Figure 6: Error in EMBW as a function of Request rate