Performance Analysis of Finite-Buffered Asynchronous Multistage Interconnection Networks*

Prasant Mohapatra
Department of Electrical and Computer Engineering
Iowa State University
Ames, IA 50011

Chita R. Das
Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA 16802

Abstract

In this paper, we present a queueing model for performance analysis of finite-buffered multistage interconnection networks. The proposed model captures network behavior in an asynchronous communication mode and is based on realistic assumptions. A uniform traffic model is developed first and then extended to capture non-uniform traffic in the presence of hot-spot. *Throughput* and *delay* are computed using the proposed model and the results are validated via simulation. The analysis is extended to predict performance of MIN-based multiprocessors where the concept of the maximum number of outstanding memory requests is included. The effects of buffer length, switch size, and the maximum allowable outstanding requests on the system performance are discussed. Various design decisions using this model are drawn with respect to delay, throughput, and system power.

Index terms: Multiprocessor, Multistage Interconnection Network, Finite Buffer, Performance Analysis, Queueing Model.

* This research was supported in part by the National Science Foundation under grant MIP-9104485. A priliminary version of this paper was presented at the International Conference on Parallel Processing, 1993.

1. Introduction

Multistage interconnection networks (MIN's) have been proposed as an efficient interconnection medium for multiprocessors. They have been used in various commercial and experimental systems [1-4]. Behavior of the interconnection network plays an important role in the performance of multiprocessors. For an optimal design, it is necessary to analyze various configurations and constraints of the interconnection network. In this paper, we present a queueing model for performance prediction of MIN and MIN-based multiprocessors.

Earlier research on MIN performance study has focussed on three types of network models: circuit switched [5]; packet switched with infinite buffer [6-9]; packet switched with finite buffer [10-14]. Study of circuit switched MIN's has gradually diminished since various packet switching techniques have become more prevalent. Infinite buffer analysis does not necessarily predict realistic behaviors of MINs under various workloads. For example, it is argued that small buffer lengths (2 or more) behave as infinite buffers [6]. This is true only under light loads or when we restrict one outstanding request per processor in the network. Multiple outstanding requests increase traffic in the system and the buffer length needs to be large in order to mimic the infinite buffer performance. Futhermore, practical designs have finite length buffers in the switches. Recent research effort therefore is directed towards analysis of finite-buffered MINs.

A model for finite buffered MINs should capture the following issues for predicting realistic performance.

- The processors in an MIMD mode operate independently of each other with occasional synchronization. Thus the network model should be based on *asynchronous* message transmission.
- The packets are normally of fixed size. Therefore, the time required for transferring a packet from one stage to the next stage is *deterministic*.
- Messages that can not be transmitted from one stage to the next due to the unavailability of buffer space should be *blocked* rather than rejected. Systems like Cedar use blocking of packets to avoid unnecessary regeneration process [1].

- The model should be general enough to analyze *uniform* as well as *non-uniform* memory reference patterns.

In addition, analysis of an isolated interconnection network does not reveal the behavior in a *multiprocessor environment*. An integrated study of the network and the system level constraints can provide better insight to the performance study.

Prior work on finite-buffered MINs are mainly based on probabilistic models [10-13, 21]. These analyses are valid for synchronous networks where all the input/output operations happen at discrete stage cycles. These models do not capture asynchronous behavior especially when the service time of the SEs is more than one clock cycle. The queueing model for finite-buffered asynchronous MINs developed in [14] assumes non-blocking capability and exponential service time for the switching elements.

None of the above models has considered all the design issues mentioned earlier. In this paper, we present a queueing model for performance analysis of MINs that considers asynchronous packet switching transmission, finite buffers, deterministic switch service time, message blocking, and constraints of a multiprocessor environment. The MIN is first modeled assuming uniform memory references. Next, the methodology for extending the model to analyze non-uniform traffic in the presence of hot-spot is described demonstrating the versatility of the analytical model. The model has been validated via extensive simulation. Average message delay and throughput are used as performance measures to characterize a MIN. Variation of performance with input load and buffer length is discussed. The analysis is extended to predict performance of MIN-based multiprocessors. Results are obtained for the effect of multiple outstanding requests on the multiprocessor performance. A performance metric called *system power* is analyzed which gives a meaningful measure considering the tradeoffs between delay and throughput [19].

The rest of the paper is organized as follows. The network architecture and operations are described in Section 2. In Section 3, a queueing model for MINs is developed for uniform traffic, and the extension to a non-uniform traffic pattern is presented in Section 4. Performance analysis and discussion on various aspects of network behavior are presented in Section 5, followed by the concluding remarks in Section 6.

2. Network Operations

An N-node multiprocessor consists of N processing elements (PEs) and N memory modules (MMs) interconnected by an $(N \times N)$ MIN. An $(N \times N)$ MIN designed using $(a \times a)$ SEs has n stages, where $n = log_a N$. An (8×8) baseline MIN is shown in Figure 1. It consists of (2×2) switching elements (SEs), each of which has buffers of size L at their input ports. Placement of buffers at the input ports of SEs is advantageous and cheaper compared to having buffers at the output ports [13]. The analysis however can be used for MINs that use buffers at the output ports as the effective arrival rate at each stage remains the same irrespective of the location of buffer.

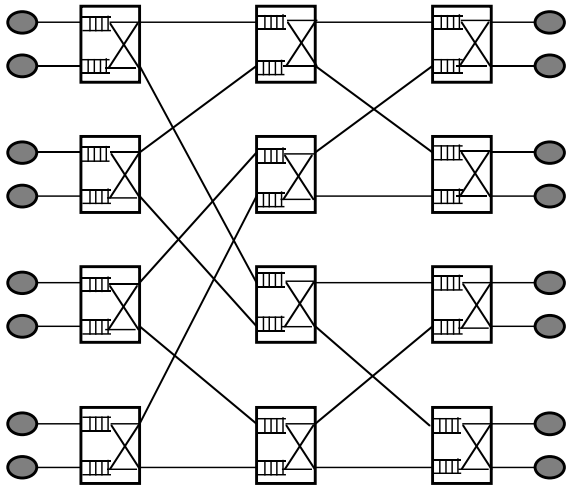

Fig. 1. An (8x8) Buffered MIN.

The message transmission protocol is packet-switched where a packet is forwarded to the next stage as and when there is an availability of buffer space. The model is based on the following assumptions.

(i) Each processor generates fixed-size messages independently at a rate λ and the inter-message times are exponentially distributed.

(ii) A memory request is uniformly distributed among all the MMs.

(iii) The SEs have deterministic service time (d cycles). During this period, the address is decoded, the destination address is checked, and the data is transferred depending upon the availability of buffer space.

(iv) A packet is blocked at a stage if the destination buffer at the next stage is full. Packets arriving at the first stage of the MIN are discarded if the buffer is full.

Almost all performance studies incorporate assumptions (i) and (ii) to ensure mathematical simplicity. Relaxation of the second assumption to non-uniform traffic is possible and the analysis of a single hot spot traffic model is presented in Section 4. Assumption (iii) is based on practical systems like Cedar and BBN Butterfly. Cedar also uses blocking of packets in the MIN and this concept is absorbed in assumption (iv).

A request from a processor is routed to the destined MM through the interconnection network (IN). An acknowledgement/reply from the MM is returned through another layer of MIN in the reverse direction to the PE that originated the request [1]. The "forward network" and the "reverse network" are distinct but are topologically identical. It is thus sufficient to analyze the performance of either network [10]. By using the effective input rate, the analysis presented here can be used for both forward and reverse networks.

3. Queueing Model

The buffers of the SEs of a MIN are of finite length and have deterministic service time. Hence, each of them can be modelled as an $M/D/1/L$ queueing center. The study consists of two parts. First, we present the analysis of an $M/D/1/L$ queue, and then extend the analysis for a network of n queues, where n is the number of stages in the MIN.

3.1. M/D/1/L Queue Analysis

Notations:

λ: packet generation rate of a source (processor).

d: switch service time.

L: length of a buffer in the SEs.

p_k: probability that there are k customers in an $M/D/1$ queueing center at steady state.

$p_k^{(L)}$: probability that there are k customers in an $M/D/1/L$ queueing center at steady state.

ρ: traffic intensity at the server $= \lambda \cdot d$.

The state probabilities of an $M/G/1/L$ queueing system are proportional to the corresponding state probabilities of the $M/G/1$ system in the interval, $0 \leq k \leq L$ [20]. Using this concept, the steady state probabilities of an $M/D/1/L$ queueing center can be derived from an $M/D/1$ queue in the range $0 \leq k \leq L$. The derivation is described in detail in [20]. The probability that there are k customers in an $M/D/1/L$ queueing center is given as

$$p_k^{(L)} = \frac{(1-x)p_k}{\sum_{i=0}^{L} p_i}, \qquad 0 \leq k \leq L, \tag{1}$$

where x denotes the probability that the buffer is full. The buffer becomes full when there are $(L+1)$ packets at the service center; L packets in the queue and one in the server. x can be also termed as the *blocking probability* as it represents the probability that a packet will be blocked at the preceeding stage. From [20],

$$x = p_{L+1}^{(L)} = \frac{p_0 - (1-\rho)\sum_{i=0}^{L} p_i}{p_0 + \rho \sum_{i=0}^{L} p_i}. \tag{2}$$

The values of p_k can be obtained by analyzing the steady state probabilities of an $M/D/1$ queueing center. The results are summarized as [19],

$$p_k = \begin{cases} 1-\rho, & \text{for } k = 0; \\ (1-\rho)(e^{\rho}-1), & \text{for } k = 1; \\ (1-\rho)\sum_{j=0}^{k} \frac{(-1)^{k-j}(j\rho)^{k-j-1}(j\rho+k-j)e^{j\rho}}{(k-j)!}, & \text{for } k \geq 2. \end{cases} \tag{3}$$

Let Q be a random variable that represents the number of jobs at a service center. The average value of Q, denoted as $E[Q]$, is given as

$$E[Q] = \sum_{k=1}^{L+1} k p_k^{(L)}. \tag{4}$$

Using Little's law, the average time, $E[T]$ spent at the center is

$$E[T] = \frac{E[Q]}{\lambda(1-x)}. \tag{5}$$

The denominator captures the effect of blocking by adjusting the arrival rate at a finite-buffered service center.

3.2. MIN Analysis

The notations used in Section 3.1 are also used for the MIN analysis with a few modifications as follows.

λ_i: packet arrival rate at stage i, $1 \leq i \leq n$.

$p_k^{(L)}(i)$: $p_k^{(L)}$ at stage i, $1 \leq i \leq n$.

ρ_i: traffic intensity at the server $= \lambda_i \cdot d$, $1 \leq i \leq n$.

x_i: blocking probability at stage $i = p_{L+1}^{(L)}(i)$, $1 \leq i \leq n$.

The basic model of a (4x4) MIN using (2x2) SEs is shown in Figure 2. The packet arrival and departure rates at each buffer are indicated in the figure. Note that the departure rate from a buffer may not be the same as the arrival rate at the buffer as it is affected by the blocking probability as well as the service time distribution of the server. The uniform memory reference assumption makes all the servers of a particular stage statistically indistinguishable. This can be verified from Figure 2. The departure rate from a first stage buffer is λ_2 which is divided equally among the output ports of the SEs because of the uniform memory reference assumption. Each output port of first stage SEs receive packets at a rate $\lambda_2/2$ from the outputs of two buffers. They add up to make the effective arrival rate at the buffers of the second stage equal to λ_2. All the buffers at a particular stage have the same arrival rate. A packet is transmitted from stage to stage passing through exactly one buffer per stage. It is therefore sufficient to analyze one buffer per stage of the MIN. A packet has to travel through a chain of n buffers in an n-stage MIN. Each buffer is modelled as an $M/D/1/L$ queueing center capturing the deterministic service time and the finite buffer considerartion. A MIN is thus modelled as a chain of n $M/D/1/L$ queueing centers as shown in Figure 3.

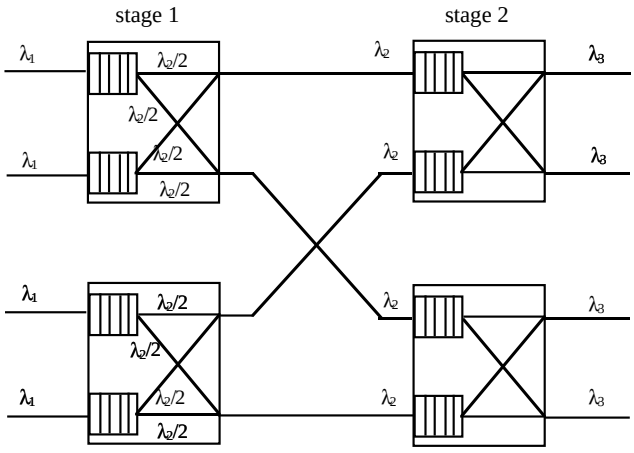

Fig. 2. Model of a (4x4) MIN.

Consider the ith stage of the MIN. Each buffer at this stage sees an average arrival rate λ_i at its input. The outputs of the ith stage buffers act as inputs to the next stage buffers. Characterization of the interdeparture time distribution and hence the departure rate is necessary to analyze the subsequent stage(s) of the MIN.

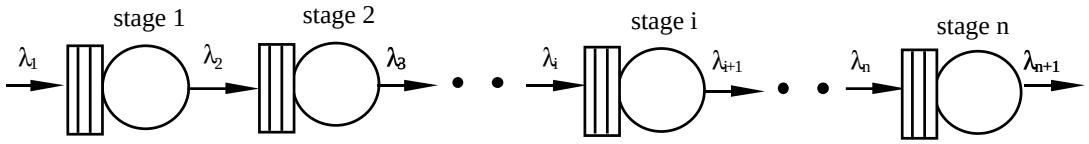

Fig. 3. A Queueing Model of an n-stage MIN.

We therefore analyze the probability density function (pdf) of the interdeparture time of an $M/D/1/L$ queue. Let τ_i be a random variable which represents the time between departures from an $M/D/1/L$ queueing center of ith stage. Let ϕ be the event that the queue is empty after a departure. $f_{\tau_i}(t)$ represents the probability density function of τ_i and $f_{\tau_i|\phi}(t)$ denotes the probability density function of τ_i given that the queue is empty. $f_{\tau_i|\overline{\phi}}(t)$ denotes the probability density of τ_i, given that the queue is not empty. State $p_0^{(L)}(i)$ is the probability that the queue is empty. Instead of using the departure point probability of an empty queue, the asymptotic limit (as $L \to \infty$), i.e., the general time probability, is used. This makes the interdeparture time density approximate but allows bounds on the approximation region [14]. This approximation is validated by comparing the results with those obtained through simulation (the deparature point probabilities are

not approximated in the simulation). Without such an approximation, the analysis would become extremely complex. Thus, the interdeparture probability density function is given by

$$f_{\tau_i}(t) = f_{\tau_i|\phi}(t)p_0^{(L)}(i) + f_{\tau_i|\overline{\phi}}(t)[1 - p_0^{(L)}(i)]. \tag{6}$$

The *pdf* is simply the density of the server when the queue is not empty. As the server has a deterministic service time of d cycles, there will be a departure every d cycles when the queue is not empty. Thus

$$f_{\tau_i|\overline{\phi}}(t) = \delta(t - d), \tag{7}$$

where $\delta(t)$ is an impulse function. When the queue is empty, the *pdf* is the density of the service time plus the arrival time. The service time and the arrival time are independent of each other. The Laplace transform of the sum of two independent density functions is equal to the product of their Laplace transforms. Taking the Laplace transforms,

$$f^*_{\tau_i|\phi}(s) = \left[\frac{\lambda_i}{\lambda_i + s}\right]\left[e^{-sd}\right]. \tag{8}$$

The inverse Laplace transform is

$$f_{\tau_i|\phi}(t) = \lambda_i e^{-\lambda_i(t-d)}U(t - d), \tag{9}$$

where $U(t)$ is an unit step function. Thus,

$$f_{\tau_i}(t) = p_0^{(L)}(i)\lambda_i e^{-\lambda_i(t-d)}U(t - d) + (1 - p_0^{(L)}(i))\delta(t - d). \tag{10}$$

The expected value of the density function of the interdeparture time can be approximated as the mean interarrival time at the next stage buffer. Let $E[\tau_i]$ represent the expected value of the interdeparture time of packets from the queueing center. $E[\tau_i]$ can be obtained from equation (10) as

$$E[\tau_i] = \int_0^\infty t \cdot f_{\tau_i}(t)dt = d + \frac{p_0^{(L)}(i)}{\lambda_i}. \tag{11}$$

It is extremely difficult to accurately characterize the nature of interdeparture processes. In order to keep the model tractable, we can approximate the interdeparture time distribution from one stage to the next as exponential with an average value of $\lambda_{i+1} = 1/E[\tau_i]$ requests/cycle. It will be shown in Section 5 that this assumption does not induce substantial difference between analytical and simulation results. We compute the departure rate, λ_{i+1}, from equation (11) as

$$\lambda_{i+1} = \frac{\lambda_i}{p_0^{(L)}(i) + \lambda_i d}. \tag{12}$$

Based on our approximation, buffers at each stage of the MIN will have a Poisson arrival process and can thus be modelled as $M/D/1/L$ queueing centers. Using equation (12), we get

$$\lambda_i = \begin{cases} \dfrac{\lambda_{i-1}}{p_0^{(L)}(i) + \lambda_{i-1} d}, & \text{for } 2 \leq i \leq n; \\ \lambda, & \text{for } i = 1. \end{cases} \tag{13}$$

The above expression is used to compute λ_i starting from $i = 1$ to n. Using equations (4) and (5), the average time spent at the ith stage is

$$E[T_i] = \frac{\sum_{k=1}^{L+1} k p_k^{(L)}}{\lambda_i}, \qquad for\ 1 \leq i \leq n. \tag{14}$$

The average delay for a packet is obtained by summing up the delays of all the stages. The normalized throughput, X, is determined by the output of a buffer in the last stage of the MIN, and is equal to λ_{n+1}.

4. Non-Uniform Traffic Model

Traffic non-uniformity in parallel systems can occur due to concurrent requests by several processors to a shared memory module (hot MM). This creates tree saturation in the interconnection network and results in *hot spot contention* [15-18]. Performance of a MIN is degraded due to the presence of hot-spots.

In this section, we extend the proposed technique for analyzing a single hot-spot traffic model. A certain fraction of the traffic (hot traffic) from each processor is assumed to be directed towards the hot MM and the remaining traffic (cold traffic) is uniformly distributed over all the MMs. Let h be the fraction of requests directed to the hot MM from a processor. For a MIN that uses $(a \times a)$ SEs, the traffic rate at the input port of an SE at stage i in the fan-in tree of a hot MM, is $(1-h)\lambda_i + a^{i-1}h\lambda_i$, where λ_i denotes the request departure rate of stage $i-1$. This is illustrated in Figure 4 where $a = 2$. The bold path represents the fan-in tree for the hot memory (MM3).

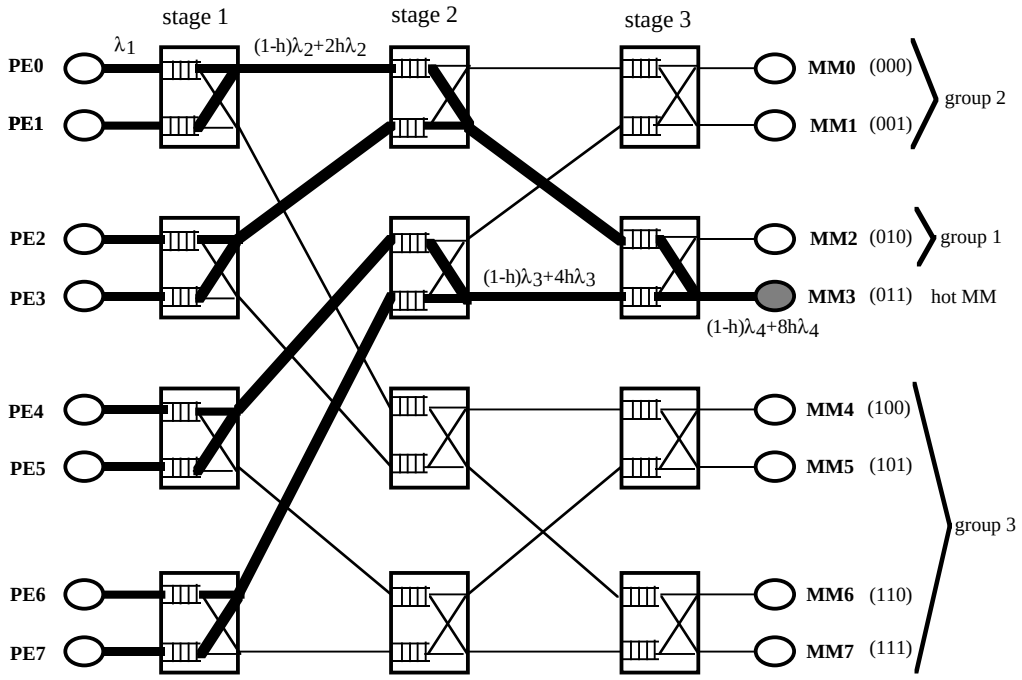

Fig. 4. Hot Spot Traffic in an (8x8) MIN.

The analysis of a single hot-spot traffic model can be explained by considering an example of an (8x8) MIN as shown in Figure 4. The following traffic patterns can be observed from the figure. Any processor accessing the hot memory module (MM3) has to take a route which can be modelled as a series of n queues as shown by the path in Figure 5(a). The cold MMs can be formed into n groups depending upon their location with respect to the hot MM. The traffic interactions in the access path to the MMs of a group are the same for any processor. Note that the route taken by each processor to access the MMs of a group will be different but because of the same traffic interaction, the model

for the path remains the same. Different groups have different access paths. Groupings for the example under consideration is illustrated in Figure 4. A PE accessing MMs of group 1 uses the path shown in Figure 5(b). The path shown in Figure 5(c) is used by a PE accessing MMs of group 2. Similarly, any PE accessing MMs of group 3 uses the path shown in Figure 5(d). The additional traffic at various stages of the MIN are due to the traffic from the other PEs. The cold traffic rate of $(1-h)\lambda_4$ is the output of the last stage at each of the networks excluding the hot traffic path. The output rate of the hot traffic path has an additional rate of $8\lambda_4$ due to the hot traffic from all the processors in the system. Thus, in order to access an MM, a PE needs to take a particular path depending upon the location of the hot MM. There are four different types of paths in this case. In general, there are $(n+1)$ different types of paths for a PE in an n-stage MIN. This is due to the network topology and can be verified by trivial observations.

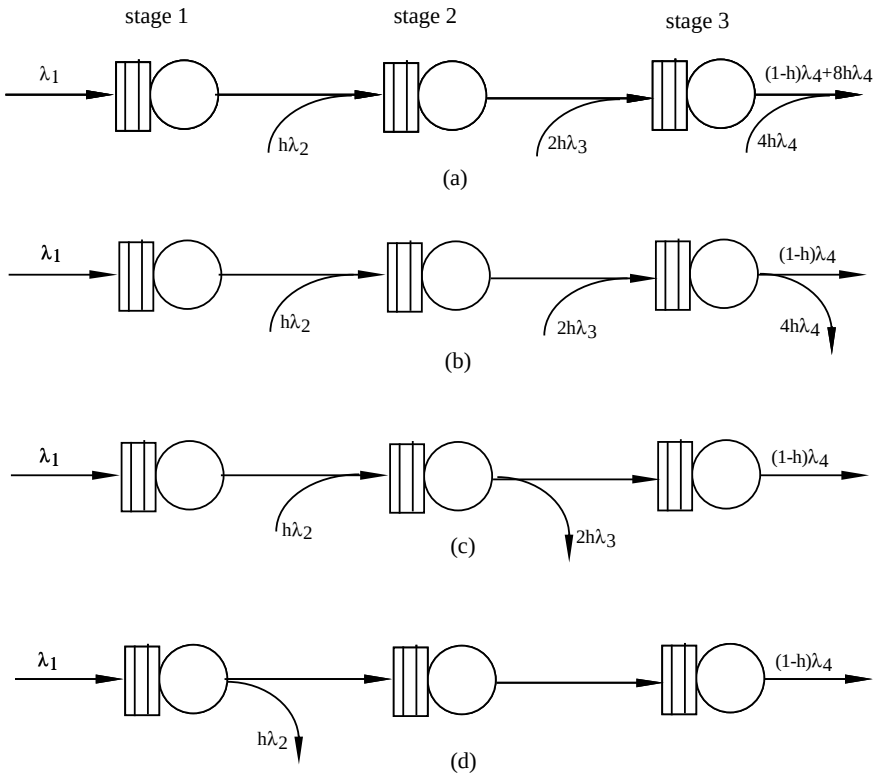

Fig. 5. Queueing Networks for Non-Uniform Traffic in an (8x8) MIN.

The paths for an (NxN) MIN that uses $(a \times a)$ switches can be constructed as follows. The first series of queues represents the path destined for the hot MM. At each stage i,

excluding stage 1, there is an additional input rate of $a^{i-2}h\lambda_i$ due to the hot traffic from other processors. At the first stage the input rate is equal to λ_1. Next, we need to group the cold MMs. For an n-stage MIN, there will be n groups. The grouping of the cold MMs can be done easily by representing the address of the MMs in base-a notation. Figure 4 shows the addresses in binary form (here $a = 2$). The address of an MM in an n-stage MIN is n-bit long. Let the address of the hot MM be represented as $(b_n b_{n-1} \cdots b_1)$. The MMs that have the addresses $(b_n b_{n-1} \cdots b_2 *)$ excluding the hot MM form group 1 ("$*$" denotes *don't care*). Group 2 comprises the cold MMs that have addresses $(b_n b_{n-1} \cdots b_3 * *)$ and are not included in group 1. Similarly, group $(n-1)$ will consist of the cold MMs which are not in any of the previously formed groups (group 1 through $n-2$) and have addresses $(b_n * * \cdots *)$. The nth group will comprise the remaining MMs. It can be derived that the number of MMs in group i is equal to $(a-1)a^{i-1}$. The path for the n groups can be modelled using n queueing networks. The first path for cold traffic represents the route to access MMs of group 1. It is the same as the hot traffic path except for the departure of a traffic rate of $a^{n-1}h\lambda_{n+1}$ instead of the additional arrival rate at the MM. The next queueing network represents the path for accessing MMs of group 2. Thus, the ith queueing network for the cold MMs represents the path for group i MMs. The additional arrival and departure rates follow a regular pattern and can be derived and formalized by observing the queueing networks shown in Figure 5.

The finite-buffered queueing networks shown in Figure 5 can be solved using the methodology described in Section 3. The delay of hot traffic can be obtained by solving the queueing network of Figure 5(a). Similarly, the mean cold traffic delay can be obtained by computing the average of the results obtained by solving the networks shown in Figures 5(b)-(d). For the example under consideration, let d_a, d_b, d_c, and d_d be the delays obtained by solving the four queueing networks shown in Figures 5(a), (b), (c), and (d), respectively. Then, the delay for hot traffic is d_a, and the average delay for cold traffic is equal to $\frac{d_a+d_b+2d_c+4d_d}{8}$. The average hot and cold traffic delays can be obtained for an n-stage MIN by solving the $(n+1)$ queueing networks.

In general, the delay calculation of hot traffic can be derived as follows. In the hot traffic path, traffic at the input port of an SE at stage i, denoted as λ_i^h, is $(1-h)\lambda_i + a^{i-1}h\lambda_i$, where λ_i denotes the request departure rate of stage $i-1$. λ_i can be computed using (13). The average time spent at the ith stage can be computed from (14). Thus the average hot traffic delay, D_{hot}, can be obtained from

$$D_{hot} = \sum_{i=1}^{n} E[T_i], \text{ where } E[T_i] = \frac{\sum_{k=1}^{L+1} k p_k^{(L)}}{\lambda_i^h}. \tag{15}$$

Similarly, the cold traffic delay for the n groups can be computed using equations (13) and (14). The input rates at various stages for each of the groups will be different and can be determined as explained earlier.

Performance degradation due to the presence of hot-spots can be improved by combining requests [15-18]. The model for non-uniform traffic can also be extended to capture the effect of combining. Depending upon the combining technique used, one can derive the probability of combining at various stages. The effective input rate at different stages depends upon the probability of combining and can be obtained using the methodologies described in [17]. These values can be used in equations (12)-(15) to compute the average delay and throughput.

5. Performance Evaluation

In order to validate the proposed analytical model, an (NxN) delta network was simulated. The network uses (2x2) SEs. Packets were generated randomly with an exponential distribution of interarrival time by each processor with an average rate of λ requests per cycle. A uniform random number generator was used to determine the destination memory. Throughput and delay were computed by counting the number of request completions and the average time taken to reach the output port, respectively. The simulation was run for sufficiently long time to obtain results in steady state. The 95% confidence interval was observed to be within 3% of the mean. Comparisons between the analytical and simulation results for (64x64) and (1024x1024) systems using (2x2) SEs are shown in Figures 6 and 7. The difference between the analysis and the simulation results is within 7%. The curves indicate that the analytical results are fairly accurate.

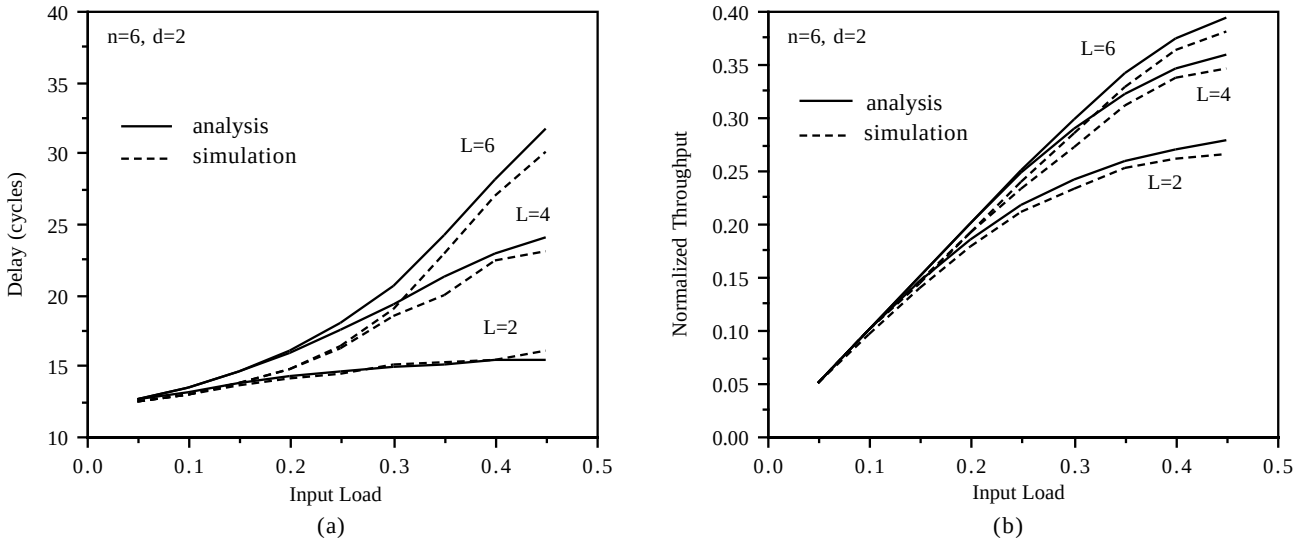

Fig. 6. Delay and Throughput of a (64x64) MIN.

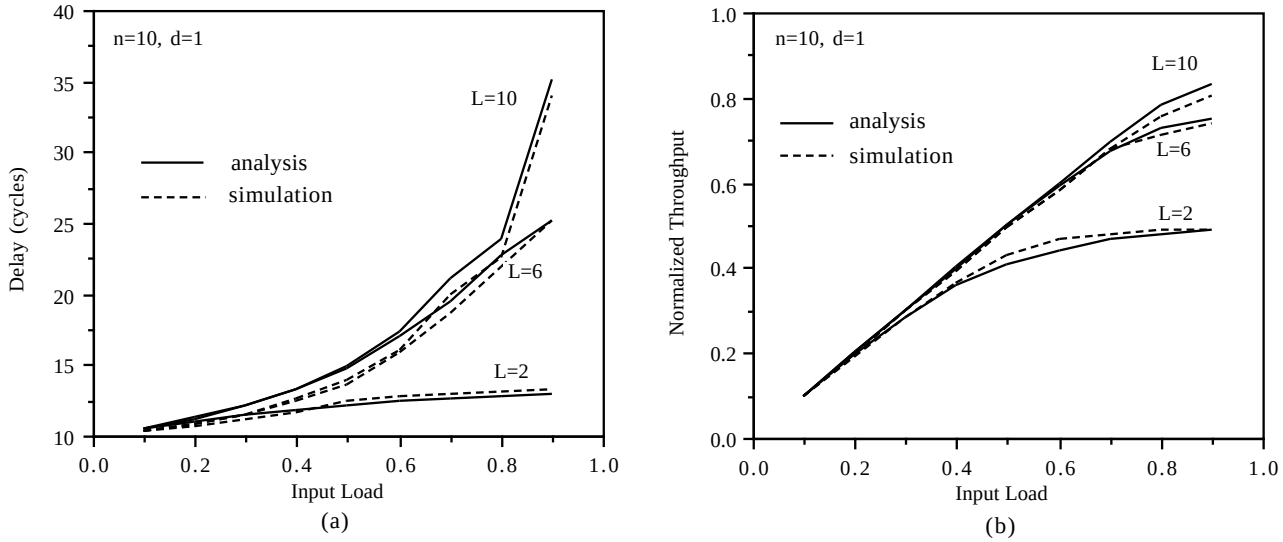

Fig. 7. Delay and Throughput of a (1024x1024) MIN.

The variation of delay for a (64x64) MIN with non-uniform traffic is shown in Figure 8. Results are plotted for hot spot traffic rates of 2%, 4%, 8%, and 16%. Figure 8(a) shows the delay incurred by a request directed to the hot memory module. Average delay for cold traffic is plotted in Figure 8(b). The simulation results are also shown to validate

the analysis. In the presence of a hot-spot, the network saturates much earlier than the uniform memory reference case.

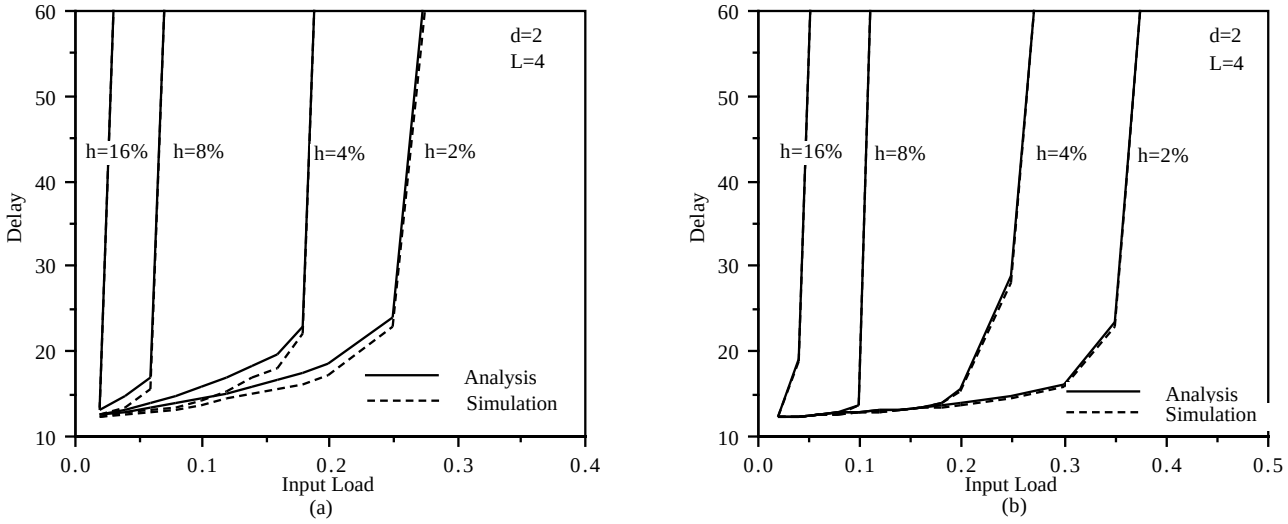

Fig. 8. Delay Variation of Non-Uniform Traffic in a (64x64) MIN.

The effect of buffer length on delay and throughput of a MIN (256×256) is depicted in Figure 9. It is mentioned in [6] that a small buffer length shows performance equivalent to an infinite buffer. It can be inferred from Figures 9(a) and 9(b) that this is true only when the input load is not high. Under light traffic, i.e. for $\lambda < 0.5$, the finite-buffered MINs with $L \geq 4$ mimics the performance of MINs having infinite buffering capacity. The variation of delay and throughput is prominent until the buffer length is considerably high for heavy traffic. The model can be used to determine the minimum buffer length required to get a performance equivalent to the infinite buffer case. For example, the minimum buffer length required to mimic the performance of infinite buffer is approximately 15 at $\lambda = 0.9$.

The effect of switch size, a, on delay and normalized throughput of a (4096x4096) MIN is shown in Figures 10(a) and 10(b), respectively. The results are plotted for five different switch sizes ($a = 2, 4, 8, 16, 64$) and for different buffer lengths. The delay is higher for smaller SEs as expected. Depending upon the priority of performance metric, an optimal SE size and buffer length can be computed. If delay is the primary concern, large SEs

with small buffers should be used to satisfy the performance requirement. Small SEs with large buffers would be more cost effective for throughput oriented systems.

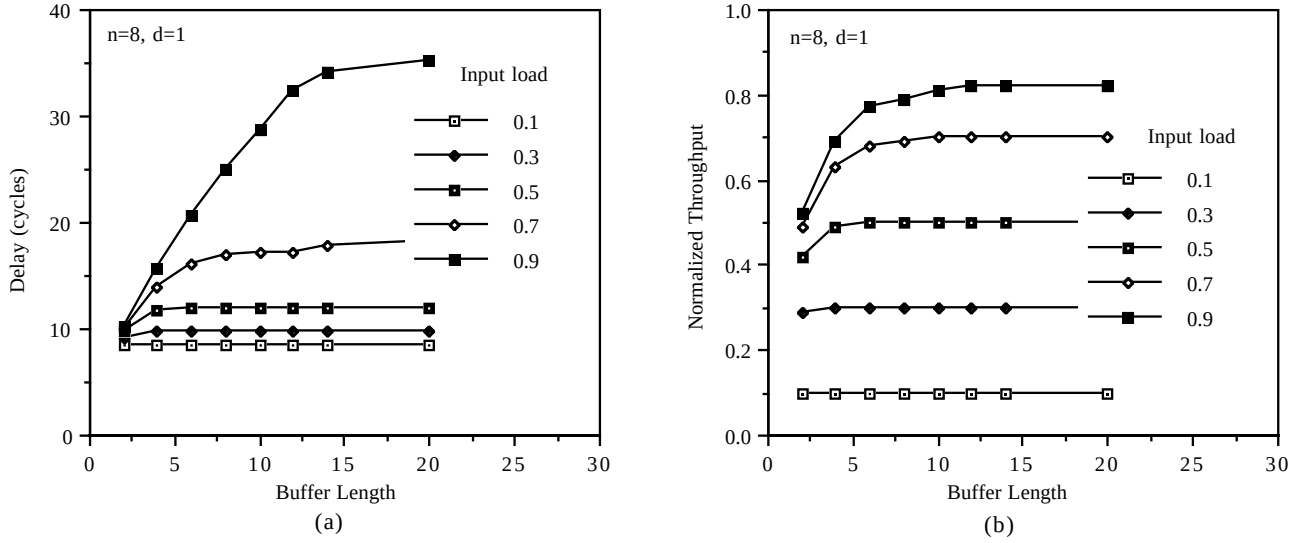

Fig. 9. Effect of Buffer Length on MIN Performance.

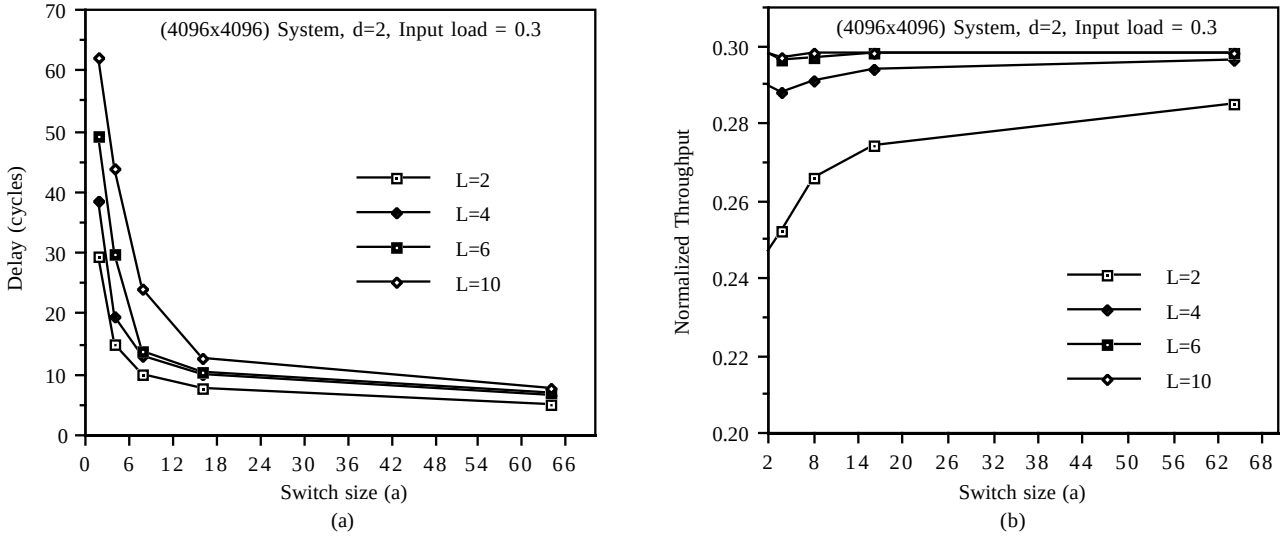

Fig. 10. Effect of SE size on Delay and Throughput.

The analysis proposed here is for a generic MIN model and can be extended to incorporate several system level constraints. In a multiprocessor environment, there is a limitation on the number of outstanding memory requests a processor can have before being blocked to wait for the completion of a request. Let m denote the maximum al-

lowable number of outstanding requests. A processor keeps on generating requests at the rate of λ packets/cycle until it has m outstanding requests. It then gets blocked until the completion of a request.

To model multiple outstanding memory requests, the same set of equations (12-14) can be used to compute the delay D for a given λ. Let $z = (\frac{1}{\lambda})$ be the *thinking time* of a processor. The maximum effective packet generation rate could be m requests per $(z + D)$ cycles. This rate is also bounded by λ. The modified value of λ, denoted as λ', is obtained from

$$\lambda' = min\left\{\lambda,\ \frac{m\lambda}{1 + \lambda D}\right\}. \tag{15}$$

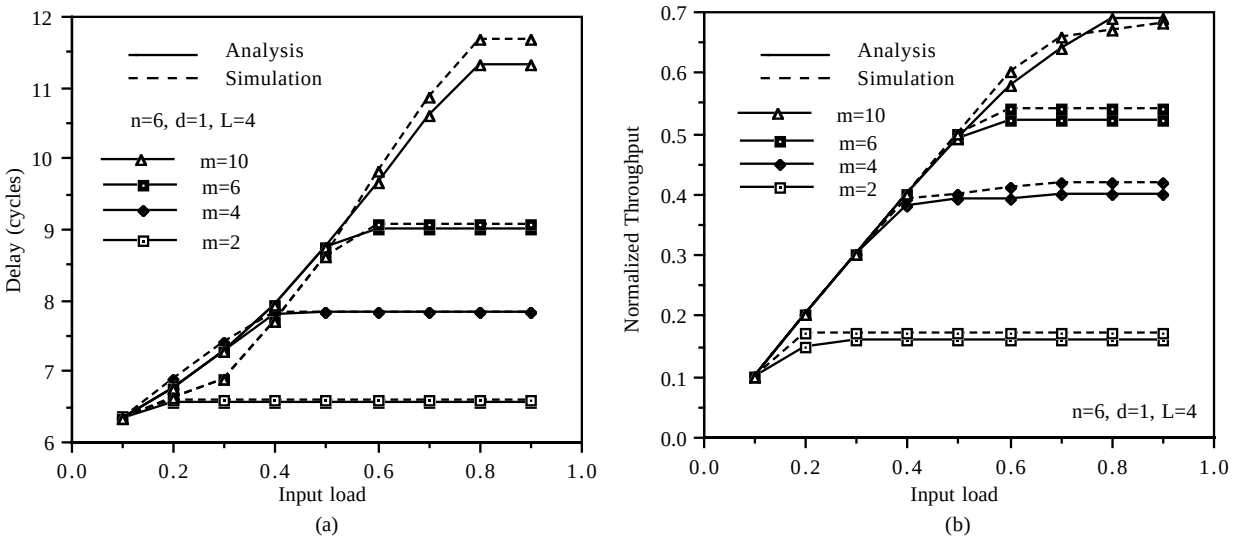

Fig. 11. Performance Variation for Multiple Outstanding Requests.

This modified value of λ' is used in the set of equations (12-15) to compute delay and throughput. Figure 11 depicts the comparison of analytical and simulation results for different values of m for a (64x64) system. As expected, better throughput is observed by allowing multiple outstanding requests. It is observed that the system saturates early for low values of m. This saturation is attributed to the fact that the effective input load to the system is limited by m. The throughput improves for higher values of m, but the packets incur more delay. For a given buffer size, we can determine the maximum input load at which the network saturates for different values of m. On the other hand, for

a given value of m and λ, the minimum buffer length can be computed for any desired performance level (delay and throughput).

Throughput and delay are not necessarily sufficient measures of system performance. It is observed that higher throughputs result in longer delay. This can also be inferred from Figures (6-11). A combined metric called *system power* is sometimes more meaningful than D or X alone. System power is defined as the ratio of throughput to delay [19]. A higher power means either a higher throughput or lower delay. In either case, a higher power is better than a lower power. Denoting P as the system power, we get $P = \frac{NX}{D}$.

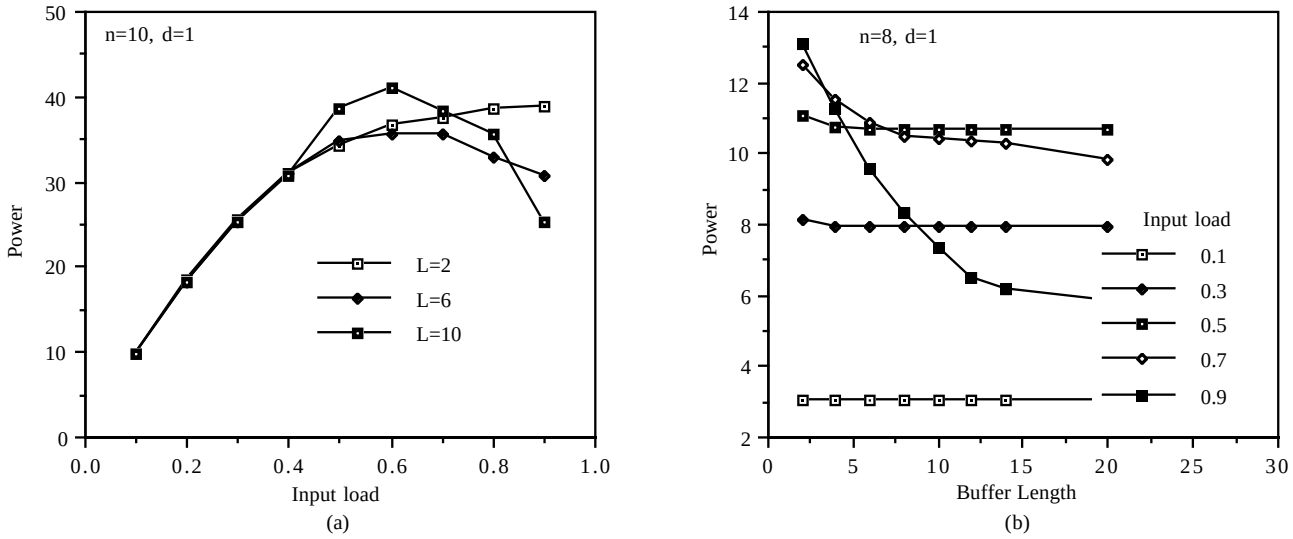

Fig. 12. Variation of System Power.

The variation of system power with respect to the input load is shown in Figure 12(a) for various buffer lengths. It is observed that the system power increases with the input load for small buffers. For large buffers, the system power increases with the input load and attains a peak value after which it decreases with the increase in load. This can be explained as follows. For a large buffer size, the throughput first increases with the input load until it saturates. On the other hand, delay increases monotonically. This can be also observed from the graphs of Figures 6 and 7. Thus, after a certain input load, the power reduces. The model can be used for predicting the optimum load to maximize the power of a MIN. The variation of power with respect to buffer length is plotted in Figure

12(b). System power is insensitive to buffer length for light load. However, system power decreases under heavy loads with the increase in buffer length.

6. Concluding Remarks

A queueing model for evaluating performance of finite-buffered, asynchronous MINs is presented in this paper. The uniqueness of this model compared to previous finite-buffered analyses is that it captures asynchronous operations, deterministic service time of switches, message blocking, and behavior in a multiprocessor environment. Both uniform as well as non-uniform traffic patterns are considered while analyzing MIN performance. Comparison with simulation results shows that the analytical model is highly accurate. The MIN is then included in a multiprocessor environment to study its effect on the overall system performance. It is observed that there is a considerable gain in throughput by allowing multiple outstanding requests in a multiprocessor. Various design alternatives based on performance requirements are discussed. It is difficult to come up with an optimal set of design parameters to satisfy all performance measures. The model can be used to compute suitable values of MIN parameters based on the priorities of performance metrics. Current investigation is focussed on the extension of the model to analyze multi-path MINs and other routing protocols such as virtual cut-through and wormhole routing.

References

[1] J. Konicek, T. Tilton, et al, "The Organization of the Cedar System," Int. Conf. on Parallel Processing, pp. 49-56, Aug. 1991.

[2] A. Gottlieb, R. Grishman, *et al*, "The NYU Ultracomputer - Designing a MIMD Shared Memory Parallel Computer," IEEE Trans. on Computers, pp. 175-189, Feb. 1983.

[3] G. F. Pfister, W. C. Brantly, *et al*, "The IBM Research Parallel Processor Prototype (RP3) : Introduction and Architecture," Int. Conf. on Parallel Processing, pp. 764-771, Aug. 1985.

[4] H. J. Siegel, L. J. Siegel, et al, "PASM: A Partitionable SIMD/MIMD System for Image Processing and Pattern Recognition," IEEE Trans. on Computers, vol. C-30, pp. 934-947, Dec. 1981.

[5] C. -l. Wu and M. Lee, "Performance Analysis of Multistage Interconnection Network Configurations and Operations," IEEE Trans. on Computers, pp. 18-27, Jan. 1992.

[6] H. Jiang, L. N. Bhuyan, and J. K. Muppala, "MVAMIN: Mean Value Analysis Algorithms for Multistage Interconnection Networks," Journal of Parallel and Distributed Computing, pp. 189-201, July 1991.

[7] D. M. Dias and J. R. Jump, "Analysis and Simulation of Buffered Delta Network," IEEE Trans. on Computers, pp. 273-282, Aug. 1981.

[8] D. L. Willick and D. L. Eager, "An Analytical Model of Multistage Interconnection Networks," ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems, pp. 192-202, 1990.

[9] A. Merchant, "A Markov Chain Approximation for the Analysis of Banyan Networks," ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems, pp. 60-67, May, 1991.

[10] T. Lin and L. Kleinrock, "Performance Analysis of Finite-Buffered Multistage Interconnection Networks with a General Traffic Pattern," ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems, pp. 68-78, May, 1991.

[11] Y. -C. Jenq, "Performance Analysis of a Packet Switch Based on a Single-Buffered Banyan Network," IEEE Jou. Selected Areas Commun., vol. SAC-1, pp. 1014-1021, Dec. 1983.

[12] H. Yoon, K. Y. Lee, and M. T. Liu, "Performance Analysis of Multibuffered Packet-switching networks in Multiprocessor Systems," IEEE Trans. on Comput., vol. C-39, no.3, pp. 319-327, Mar. 1990.

[13] J. Ding and L. N. Bhuyan, "Performance Evaluation of Multistage Interconnection Networks with Finite Buffers," Int. Conf. on Parallel Processing, pp. I-592-I-599, Aug. 1991.

[14] T. N. Mudge and B. A. Makrucki, "An Approximate Queueing Model for Packet Switched Multistage Interconnection Networks," Int. Conf. on Distributed Computing Systems, pp. 556-562, Oct. 1982.

[15] G. F. Pfister and V. A. Norton, "Hot Spot Contention and Combining in Multistage Interconnection Networks," Int. Conf. on Parallel Processing, pp. 790-797, 1985.

[16] P. -C. Yew, N. -F. Tzeng, and D. H. Lawrie, "Distributing Hot-Spot Addressing in Large-Scale Multiprocessors," Int. Conf. on Parallel Processing, pp. 51-58, Aug. 1986.

[17] G. Lee, "A Performance Bound of Multistage Combining Networks," IEEE Trans. on Computers, pp. 1387-1395, Oct. 1989.

[18] S. R. Dickey and O. E. Percus, "Performance Differences among Combining Switch Architectures," Int. Conf. on Parallel Processing, vol. I, pp. 110-117, Aug. 1992.

[19] R. Jain, *The Art of Computer Systems Performance Analysis,* John Wiley and Sons Inc., New York, 1991.

[20] B. V. Gnedenko and I. N. Kovalenko, *Introduction to Queueing Theory,* Second Edition, Birkhauser, Boston, 1989.

[21] Y. Mun and H. Y. Youn, "Performance Analysis of Finite Buffered Multistage Interconnection Networks," IEEE Trans. on Computers, pp. 153-162, Feb. 1994.