PERFORMANCE ANALYSIS OF CLUSTER-BASED MULTIPROCESSORS*

Prasant Mohapatra, Chita R. Das, and Tse-yun Feng

Department of Electrical & Computer Engineering
The Pennsylvania State University
University Park, PA 16802

Abstract

A queueing model for performance evaluation of cluster-based multiprocessors is proposed in this paper. Most system components are modelled as $M/D/1/L$ queues to capture deterministic service time and finite buffer behavior. Various subsystems are analyzed independently and then integrated for the system level analysis. Average delay, throughput, and processor utilization are the performance parameters studied in this analysis. The analytical results are first validated via simulation. Next, several design alternatives are discussed using the model. These include the effect of buffer length and identification of bottleneck centers for various design configurations.

* This research was supported in part by the National Science Foundation under grant MIP-9104485.

I. INTRODUCTION

Cluster-based multiprocessors, also known as hierarchical systems, are designed to reduce the network complexity by incorporating hierarchies of interconnection networks [1-3]. These systems take advantage of the locality of reference exhibited by many programs. Hierarchical multiprocessors offer several advantages over the single-level designs and provide various design alternatives [4]. The design of such a system is quite complex and needs careful study of many interacting performance parameters. A performance model is a useful tool for exploring the design space and examining various parameters. This paper reports a simple, yet powerful performance evaluation model for shared-memory cluster-based multiprocessors.

Performance evaluation of cluster-based multiprocessors has been studied by a few researchers [4, 5]. These models assume infinite buffer capacity and/or exponential service time distribution. Modelling of the interconnection network (IN) is an essential part of multiprocessor analysis. Normally, three types of INs have been used for the shared-memory design. These are bus, crossbar, and multistage interconnection network (MIN). Different techniques for the performance evaluation of these INs are summarized in [6]. The MIN analysis, being more complex than bus or crossbar, has drawn more attention. Performance of MINs with infinite buffers is analyzed in [7, 8]. Probabilistic analysis of finite-buffered networks have been studied in [9, 10] assuming synchronous IN. A queueing model for finite-buffered MINs is developed in [11] assuming exponential service time and non-blocking capability.

The clusters in hierarchical systems operate independently and the inputs to the network or the memory are asynchronous in nature. This paper thus attempts to model the asynchronous behavior through queueing analysis. A detailed queueing network model of the complete system consisting of all the processors, memories, and network components could be prohibitive even for a small multiprocessor. A decomposition approach seems a natural choice to keep the analysis tractable. We use a hierarchical decomposition technique to model a two-level cluster-based system. The system is modelled as a queueing network where each subsystem is represented as a service/delay center. The subsystems are analyzed independently and then integrated to form the system-level model. The salient features of the proposed model are highlighted as follows.

- The interconnection networks are modeled as finite-buffered service centers to reflect the practical system behavior.

- Deterministic service time is considered for elements of the IN and the memory.
- A packet is blocked at a center due to the unavailability of buffer space at the destination center.
- The interdeparture rate of one center affects the arrival rate of the next center in a finite-buffered queueing network.

The performance parameters discussed here are average delay, throughput and utilization. Due to the approximations included in the analysis, a simulation study is conducted to validate the model. System behavior with respect to different workloads and design constraints is examined. Contrary to previous results [8], it is shown that the length of the buffer could have a significant effect on the system performance. Large buffers are shown to be suitable for throughput oriented systems, whereas smaller buffers should be used where response time is of high priority. The model can be used for identifying the bottleneck center by analyzing the utilization of various components.

The rest of the paper is organized as follows. The system architecture is described in Section II. In Section III, the model assumptions and system decomposition are presented. Queueing analyses for various subsystems are presented in Section IV. In Section V, integration of the decomposed subsystems is described. Numerical results from the analysis and simulation are given in Section VI, followed by the conclusions in Section VII.

II. SYSTEM DESCRIPTION

A generic organization of a two level cluster-based shared memory system is shown in Figure 1. The clusters of processing elements (PEs) form the first level and the connection of the clusters through a global network (GN) constitutes the second level. The depicted structure is an $(N \times N)$ multiprocessor designed using K clusters. Each cluster has n PEs. A PE consists of a processor, a memory module (MM) and a processor node controller (PNC). A memory module is called the *private memory* of the processor present in the same node. For a particular processor, the MMs of other PEs of the same cluster are called *local memories* (LMs), and the MMs in other clusters are called *global memories* (GMs). The PNC is responsible for handling the requests from the local and global memories. Each cluster is connected to the GN by a bus which is shared by all the PEs of the cluster. This bus, termed as *cluster-to-global bus* (CGB), can be accessed directly by the PEs of a cluster without going through the local network (LN).

Fig. 1. A generic two-level cluster-based multiprocessor.

A request through the GN goes to any MM via a bus which is referred as *global-to-cluster bus* (GCB). Each cluster is thus associated with two busses as shown in Figure 1 - *cluster-to-global bus* and *global-to-cluster bus*. This type of an architecture is also studied in [4].

A processor can access its private memory, local memory (MMs of its cluster) or global memory (MMs of other clusters). An access to the private memory does not go through any network. A local memory reference goes through the LN. A global memory request is first transmitted to the *cluster-to-global* bus which connects the cluster to the GN. Then, it passes through the GN, *global-to-cluster* bus, and finally reaches the destination MM. After memory service, an acknowledgement is sent to the requesting PE through a return path, which includes the CGB, GN, and GCB.

III. SYSTEM LEVEL MODEL

The key concept of modeling the system is hierarchical decomposition - the process of splitting the system model into smaller submodels, each of which is analyzed in isolation. The solution of the original model is formed by combining the submodels taking into account

the dependency of various parameters. We consider packet switching communication where all packets are of fixed length. The terms request and packet are used interchangeably throughout this paper. The model is based on the following assumptions.

(i) Each processor generates packets independently at a rate λ and the intermessage times are exponentially distributed.

(ii) A request could be directed to one of the n MMs of its own cluster (local request) or to a global MM. The local memory request is uniformly distributed among the local MMs and similarly, a global memory request is uniformly distributed among the global MMs.

(iii) A conflict occurs when two or more packets are routed to the same port. Conflicts are randomly resolved by allowing only one packet to move to the destined port if there is buffer space.

(iv) If a request generated by a processor finds the buffer at the first service center full, then the packet is rejected. A packet is never lost in the network.

A. Overall System Model

A request generated by a PE traverses its own path while interfering with the traffic due to the requests generated by the other PEs. However, under the uniform traffic load, all the paths are statistically indistinguishable. The system behavior can therefore be obtained by modelling any one path.

The system is represented as a network of queues as shown in Figure 2, where the path of a packet through various queueing centers is illustrated. A request from a processor accesses the global memory with a probability g, and with a probability $(1-g)$ the request is directed to the local memory. A global memory request or acknowledgement first accesses the CGB as shown in the figure. If the buffer at the CGB is full, all the arriving requests are rejected. In Figure 2, β_0 represents additional requests to the CGB from other PEs of the same cluster. β_1 reflects the traffic into the GN from other clusters. Effect of the GN is captured by representing it as a delay center. β_2 denotes the additional packets in the path which are not directed to the particular global MM with respect to the request under consideration.

We have not shown the model for the *global-to-cluster* bus in Figure 2. The input to a *global-to-cluster* bus will be from the output of a queueing center of the GN. There can be only one packet coming into this bus in a cycle. A packet from the *global-to-cluster* bus is transmitted to an MM. Thus, the service time of the GCB can be merged with that of the GN subsystem.

Fig. 2. Queueing model of the cluster-based system.

The traffic at an MM comes from three sources: private processor (processor of the same node), local PEs, and PEs of other clusters (global requests). β_3 represents the additional packets from the private processor and the local PEs. β_4 represents the acknowledgements to the private processor and the local PEs ($\beta_3 = \beta_4$). A request to the global MM could be a *read* or a *write*. The memory sends the requested packet for a *read* request or an acknowledgement for a *write*. For the return path, β_0, β_1, and β_2 are the same as described earlier.

A request to one of the local MMs goes through the LN as shown in Figure 2. In the return path, it again accesses the LN to get back to the node which originated the request. The additional inputs at the LN, β_5, are due to the requests or acknowledgements generated by other PEs of the same cluster. β_6 represents the requests or acknowledgements directed to the other MMs or PEs (except the one under consideration). At the local MM, there could be requests from the private processor and from other clusters which are denoted as β_7. The acknowledgements for the private and global requests are identified as β_8 and is quantitatively equal to β_7.

Let D_l and D_g represent the average delay due to local memory access and global memory access, respectively. Then the average delay D for a request completion is given by

$$D = gD_g + (1-g)D_l. \tag{1}$$

B. System Decomposition

The system queueing model depicted in Figure 2 consists of four major subsystems. These are the LN, CGB, GN and memory. The LN is usually a crossbar and so there is no contention in the network. The delay is only due to the time taken for message transfer. Although we have neglected this fixed delay, it can be included in the LN subsystem. Local requests thus incurs delay at the memory subsystem only. Bus or MIN-based LNs can be modelled as described later for GN subsystem. For the GN, we need to model a bus or a MIN subsystem. Message transfer time in crossbar GN can be included as a fixed delay in the GN subsystem. In summary, we need to model the following three subsystems - MIN, bus, and the memory.

IV. QUEUEING MODELS FOR THE SUBSYSTEMS

Buses, and switching elements (SEs) of a MIN have finite-length buffers. Service times of these components are assumed deterministic (fixed). Hence, they are modeled as $M/D/1/L$ queueing centers (exponential arrival time, deterministic service time, single server and finite length buffer). A survey of queueing networks with blocking can be found in [12], and the detailed analysis of an $M/D/1/L$ queue is reported in [13]. Here, we directly state the results required for this study.

Let λ be the arrival rate at a queueing center. The traffic intensity, ρ, is equal to $\lambda \cdot d$, where d is the service time. Let L be the buffer length. The probability that there are k customers at the center, denoted as $p_k^{(L)}$, is given as

$$p_k^{(L)} = \frac{(1-x)p_k}{\sum_{i=0}^{L} p_i}, \qquad 0 \le k \le L, \tag{2}$$

where, x denotes the probability that the buffer is full (blocking probability), and is given as

$$x = p_{L+1}^{(L)} = \frac{p_0 - (1-\rho)\sum_{k=0}^{L} p_k}{p_0 + \rho \sum_{k=0}^{L} p_k}. \tag{3}$$

The values of p_k's can be obtained by analyzing the steady state probabilities of an $M/D/1$ queueing center [13]. Using Little's law, the average time, $E[T]$, spent at the center is

$$E[T] = \frac{\sum_{k=1}^{L+1} k p_k^{(L)}}{\lambda(1-x)}. \tag{4}$$

A. MIN Analysis

A complete MIN queueing model is described in [13]. Here, we summarize the results. The SEs are modelled as $M/D/1/L$ queueing centers. A MIN with s stages is represented as a series of s $M/D/1/L$ queues. The uniform memory reference assumption makes all the queues of a particular stage statistically indistinguishable. The interdeparture time distribution of an $M/D/1/L$ queue is not the same as that of the arrival time distribution. In order to keep the model tractable, we can approximate the interdeparture time distribution from stage i to the next stage as exponential with an average value of $\lambda_{i+1} = 1/E[\tau_i]$ requests/cycle, where $E[\tau_i]$ is the expected value of the interdeparture time of stage i [14]. The departure rate from stage i, λ_i, becomes [13]

$$\lambda_i = \begin{cases} \dfrac{\lambda_{i-1}(1-x_{i-1})}{p_0^{(L_s)}(i) + \lambda_{i-1}(1-x_{i-1})d_s}, & \text{for } 1 \le i \le s; \\ \lambda, & \text{for } i = 0 \text{ (input to the MIN)}, \end{cases} \tag{5}$$

where L_s is the buffer length of the SEs, and d_s is the switch service time. The blocking probabilities, x_i's, are obtained using equation (3).

The above expression is used to compute λ_i from $i = 1$ to s. The blocking at a stage affects the arrival rate at its preceeding stage. Hence, equation (5) is solved iteratively until the network reaches a steady state ($\lambda_1 = \lambda_2 = \ldots \lambda_s$). After each iteration, we start with a new value of λ_1 equal to λ_s obtained in the previous iteration. The average time spent at the ith stage can be obtained from equation (4), and the average delay for a packet is obtained by summing up the delays of all the stages.

B. Bus Analysis

The bus is modeled as an $M/D/1/L$ queue. The steady state probabilities, $p_i^{(L_b)}$'s, can be obtained from equation (2), for $0 \le i \le L_b$, where L_b is the buffer length of the bus. Using Little's law, we get the average time spent at the center. Let λ_o denote the output rate of a queueing center representing the bus. λ_o can be computed as

$$\lambda_o = \frac{\lambda_b}{p_0^{(L_b)} + \lambda_b d_b} \tag{6}$$

where, λ_b is the traffic input rate to the bus, and d_b is the bus service time.

C. Memory Analysis

Inputs to a memory is from the n PEs of the cluster and the global requests from the non-local PEs via the GN. Under the finite buffer assumption, when the memory buffer gets full, the incoming packets to the memory are rejected if they were generated by the processors of the same cluster and the global requests are blocked in the GN. A blocked packet in the GN affects the packets in the CGB which in turn affects the request generation rate from the PNC. This chained reaction is difficult to capture in a model without sacrificing simplicity. All the reported models therefore assume infinite buffer capacity at the PNC or memory queue. In order to keep the model tractable, we assume that the memory buffer is large enough to store any number of requests.

The requests to the memory from the GN do not have an exponential interarrival time distribution due to the $M/D/1/L$ queues as discussed earlier. It is extremely difficult to characterize the input process to the memory modules. We have assumed the request interarrival time at an MM as exponentially distributed. The validity of this assumption lies in the fact that the number of requests to an MM from the local processors is higher compared to the number of global requests, and thus dominates the arrival pattern.

A memory module can now be modelled as an $M/D/1$ service center. Let λ_m be the total arrival rate to a memory module. Each processor of a cluster generates local requests at a rate $(1-g)\lambda$. There are n PEs in a clusters and the total generation rate is $n(1-g)\lambda$. The request rate to any one of the n MMs from local processors is $n(1-g)\lambda(\frac{1}{n}) = (1-g)\lambda$. Processors from other clusters generate global requests at a rate $(K-1)ng\lambda$. Each of these requests can be directed to any of the global memories. There are $(K-1)n$ global memories for any processor. This gives a rate at an MM due to global request as $(K-1)ng\lambda\frac{1}{(K-1)n} = g\lambda$. The total request rate, λ_m, to an MM is thus the sum of local and global request rates and is equal to λ.

Let d_m be the service time of an MM. The traffic intensity, ρ_m, is $\lambda_m d_m$. The average delay at the memory (D_m) can be written from the $M/D/1$ queue results as [13]

$$D_m = d_m + \frac{\rho_m d_m}{2(1-\rho_m)}. \tag{7}$$

V. SYSTEM COMPOSITION

The individually modelled subsystems need to be combined to reconstruct the system level model. Interdependence of traffic flow must be taken into account while combining the subsystems. In essence, one needs to determine the request arrival rate at each of the subsystems.

The request generation rate of a processor is λ requests/cycle. A local memory access has no contention in the crossbar LN and is not blocked because of the infinite buffer assumption in the MMs. A global request is rejected if the CGB buffer is full. The arrival rate at a CGB is affected by the newly generated global requests as well as the acknowledgements from the local MMs. A CGB serves n PEs (of a single cluster), each of which generates global requests at a rate $g\lambda$ requests/cycle. The total global request rate from a single cluster is equal to $ng\lambda$. Global requests to a particular cluster can be from the remaining $(K-1)$ clusters which generate global requests at a rate $(K-1)ng\lambda$. With a probability $(\frac{1}{K-1})$, these requests are destined to a particular cluster. The acknowledgement traffic is equal to $(K-1)ng\lambda \cdot \frac{1}{(K-1)} = ng\lambda$. The arrival rate at a CGB, λ_b, is the sum of the global request rate and the acknowledgement rate, and is equal to $2ng\lambda$.

Traffic arrival rate at the GN depends upon the departure rate from the cluster buses. Equation (6) is used to compute λ_o, the output rate from the CGB. The arrival rate at an input port of a GN, denoted as λ_{gn}, is equal to $K\lambda_o$ for a bus-based GN, and λ_o for a crossbar or MIN-based GN.

A packet is blocked at the CGB if the buffer at the entry to the GN is full. The probability that the buffer in the GN is full is $p_{L_b+1}^{(L_b)}$ for a bus-based GN. For a MIN-based GN, the probability that a buffer is full in the first stage is $p_{L_s+1}^{(L_s)}(1)$. The input rate to the GN is thus modified to

$$\lambda'_{gn} = \begin{cases} \lambda_{gn}(1 - p_{L_b+1}^{(L_b)}), & \text{for bus-based GN;} \\ \lambda_{gn}[1 - p_{L_s+1}^{(L_s)}(1)], & \text{for MIN-based GN;} \\ \lambda_{gn}, & \text{for crossbar-based GN.} \end{cases} \tag{8}$$

This new value λ'_{gn} is the actual input rate to the GN. To have an effective input rate of λ'_{gn} to the GN, the output rate of CGB, λ_o, must also be adjusted to a new value λ'_o. λ'_o is equal to λ'_{gn} for MIN-based, and λ'_{gn}/K for bus-based GN. In order to have an interdeparture rate of λ'_o, the input rate to the CGB needs to be modified. We get the adjusted input rate λ'_b to the CGB from equation (6) as

$$\lambda'_b = \frac{p_0^{(L_b)}\lambda'_o}{1 - d_b\lambda'_o}. \tag{9}$$

The blocking initiates a chain reaction which eventually affects the input rate which is modified to a new value $\lambda' = \lambda'_b/2ng$. The system model is solved using the new values, λ', λ'_b, λ'_{gn}, and is iterated until the steady state is achieved. The CGB delay, denoted as D_c, is computed as described in section IV.B using the steady state value of λ_b. The global network delay, denoted as D_{gn}, is computed similarly for a global bus or a MIN by using the steady state value of λ_{gn}. The fixed delay in message transfer in crossbar interconnection is considered as D_{gn} for crossbar-based GN. Finally we get, $D_l = D_m$, and $D_g = 2D_c + 2D_{gn} + D_m$.

The system delay D is obtained from equation (1). System throughput is equal to the average number of request completions per cycle. The number of jobs served at an MM in a cycle (λ_m) indicates the throughput per PE or the *normalized throughput*. Let X and SX denote the normalized throughput and the system throughput, respectively. These are expressed as, $X = \lambda_m = \lambda'$, and $SX = N \cdot X = N \cdot \lambda'$.

VI. RESULTS AND DISCUSSION

A. Model Validation

The modelling technique described in the previous sections has inducted a few approximations at certain stages of analysis to preserve simplicity. In order to validate the technique and justify the approximations, the system was simulated. Requests are generated randomly by each processor with an exponential distribution of interarrival time with a mean of $1/\lambda$. The destination MM is determined by using a uniform random number generator. Each packet is time-stamped after its generation. The request completion time is checked to compute the delay. Throughput is obtained by counting the request completions per cycle. The average number of request completions per cycle per processor gives the normalized throughput. The simulations were run until the 95% confidence interval was within 3% of the mean.

Figures 3(a) and 3(b) show the comparison of analytical and simulation results of a (256x256) multiprocessor with a MIN-based GN. There are 32 clusters each having 8 PEs. The GN is a (32x32) MIN that uses (2x2) SEs. The closeness of the results indicates that the analytical model is fairly accurate. The model has been validated for other configurations and over a wide range of input parameters.

Fig. 3. Performance comparison of a (256x256) system (MIN-based GN).
K=32, n=8, s=5, d_m=8,d_b=2, g=0.3, L_b=8, L_s=2

B. Design Trade-Offs

A performance model is useful not only to quantify a set of parameters for a given configuration, but also to investigate the effect of different parameters on system performance. The following discussions illustrate these concepts.

The effect of buffer length on delay and throughput of a (256x256) system is depicted in Figure 4. It is mentioned in [8] that a small buffer length shows performance equivalent to an infinite buffer. It can be inferred from Figures 4(a) and 4(b) that this is true only when the input load is less. Under light traffic, i.e. for $\lambda \leq 0.05$, a finite-buffered MIN with $L \geq 4$ mimics the performance of infinite-buffered MINs. 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 size of buffer to achieve performance equivalent to the infinite buffer case. For example, the minimum buffer length required to mimic the performance of infinite buffer is approximately 20 at $\lambda = 0.1$.

Designers have the choice to increase or decrease the buffer length of various subsystems. This can be decided by comparing the relative utilization of various subsystems. Relative utilization can be obtained with respect to the utilization of the processor (considered as 1.0). If the traffic rate at a subsystem is θ and the service time is d, the utilization U is given as, $U = \theta \cdot d$. Using this formula, relative utilizations of various subsystems for two multiprocessor configurations are plotted in Figures 5(a) and 5(b), respectively.

Fig. 4. Effect of buffer length on the performance of a (256x256) system.
K=32, n=8, s=5, d_m=8, d_b=2, d_s=2, g=0.3, L_b=8

Fig. 5. Relative utilization of various subsystems.
(a) K=4, n=8, d_m=12, g=0.1, L_b=4
(b) K=32, n=8, s=5, d_m=8,d_b=2, g=0.3, L_b=8, L_s=2

It is inferred from Figure 5(a) that the global bus is relatively over-utilized as expected. The system throughput can thus be improved by increasing the buffer size of the global bus. Futhermore, as the global bus could be the bottleneck at high input load, a multiple-bus implementation may be required to improve the system performance. Increasing the buffer size of the CGB will have little effect on the performance as it is relatively under-utilized. Similarly, from Figure 5(b), one can deduce that the buffer length of the CGB and/or the global switch can be increased to improve the throughput. Here, the CGB and the MIN are

the potential bottleneck centers. Multiple path implementation from the LN to the GN can be used to alleviate this problem as has been done in the Cedar design.

Performance of a cluster-based multiprocessor is dependent on several interrelated parameters and various constraints. It is extremely difficult to come up with an optimal design satisfying all requirements. The proposed model can be used to decide the trade-offs depending upon the priorities of various performance parameters. For throughput oriented systems, large sized buffers should be used and smaller buffers should be used where response time is of high priority (Figure 4). Increasing the buffer length at a bottleneck center improves the performance of the system. The system bottleneck can be pinned down by analyzing the relative utilizations (Figure 5).

VII. CONCLUSIONS

A performance model is an essential tool for predicting the behavior of a system. It can also be used to analyze intricate details and various design optimization issues. One such model is presented in this paper for predicting the performance of two-level cluster-based multiprocessors. System throughput, average delay, and processor utilization are computed using this model. The analysis captures the effect of finite buffers and deterministic service time on system performance. The novelty of the approach lies in the aggregation of subsystems by considering the interdependence of parameters. The effect of buffer length on the system performance have been analyzed. Various design alternatives are suggested based on performance requirements.

The proposed model could be extended for multi-level hierarchical designs and other types of hierarchical architectures. A complete system model of this nature provides better insight to design a well-balanced system as compared to an analysis of the network only.

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] E. F. Gehringer, A. K. Jones, and Z. Z. Segall, "The Cm* Testbed," IEEE Computer, pp.40-53, Oct. 1982.

[3] D. Lenoski, J. Laudon, et al, "The Stanford DASH Multiprocessor," IEEE Computer, pp. 63-79, March, 1992.

[4] W. T. Hsu and P. -C. Yew, "The Performance of Hierarchical Systems with Wiring Constraints," Int. Conf. on Parallel Processing, pp. I-9 - I-16, Aug. 1991.

[5] S. P. Dandamudi and D. L. Eager, "Hierarchical Interconnection Networks for Multicomputer Systems," IEEE Trans. on Computers, pp. 786-797, June 1990.

[6] L. N. Bhuyan, Q. Yang, and D. P. Agrawal, "Performance of Multiprocessor Interconnection Networks," IEEE Computer, pp. 25-37, Feb. 1989.

[7] C. P. Kruskal and M. Snir, "The Performance of Multistage Interconnection Networks for Multiprocessors," IEEE Trans. on Computers, pp. 1091-1098, Dec. 1983.

[8] 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.

[9] 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.

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

[11] 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.

[12] R. O. Onvural, "Survey of Closed Queueing Networks with Blocking," ACM Computing Surveys, pp. 83-121, June 1990.

[13] P. Mohapatra and C. R. Das, "Performance Analysis of Finite-Buffered Multistage Interconnection Networks," Computer Engineering Technical Report, TR-92-99, The Pennsylvania State University, 1992.

[14] T. N. Mudge and B. A. Makrucki, "A Queueing Model for Delta Networks," SEL Report 159, Dept. of Elect. & Comp. Engr., Univ. of Michigan, Jan. 1982.