An Analytical Model of Multistage Interconnection Networks

Darryl L. Willick
Derek L. Eager

Department of Computational Science
University of Saskatchewan

ABSTRACT

Multiprocessors require an interconnection network to connect processors with memory modules. The performance of the interconnection network can have a large effect upon overall system performance, and, therefore, methods are needed to model and compare alternative network architectures.

This paper is concerned with evaluating the performance of multistage interconnection networks consisting of $k \times s$ switching elements. Examples of such networks include omega, binary n-cube and baseline networks. We consider clocked, packet switched networks with buffers at switch output ports. An analytical model based on approximate Mean Value Analysis is developed, then validated through simulations.

1. Introduction

As the need for computational power has grown, multiprocessor systems have become a promising means of providing high performance at reasonable cost. A common architecture for these systems consists of a number N of processors and memory modules connected via some form of interconnection network. There is a very wide range of interconnection networks that have been proposed. On the one extreme is the crossbar interconnection, where N connections can be made simultaneously but where the number of switches and therefore cost rises as the square of N. At the other extreme is the single global bus which is very inexpensive but does not allow scaling of systems to large sizes. Between these two extremes there are a number of possibilities including meshes, hypercubes and multistage networks [4].

This paper concerns multistage interconnection networks. These networks have received considerable attention and several classes of multistage networks have been proposed and investigated in the literature [1, 6, 11]. Several machines have been designed using this type of interconnection network including the NYU Ultracomputer [7], the IBM RP3 [15], the BBN Butterfly [17], and the Illinois Cedar system [5].

* This material is based upon work supported by the Natural Sciences and Engineering Research Council of Canada. Authors' current address: Department of Computational Science, University of Saskatchewan, Saskatoon, Saskatchewan, Canada, S7N OWO.

Multistage interconnection networks connect processors (PEs) to memory modules through stages of switches. The switches are k-input, s-output ($k \times s$) devices which can route data arriving on any input port to any output port. Fig. 1 shows an example multistage interconnection network, termed an omega network [11], constructed from 2×2 switches. An omega network consisting of $k \times k$ switches connects N processors to N memory modules through $\log_k N$ stages with N/k switches in each stage. The routing of memory requests and replies through multistage interconnection networks is a distributed process. Each network switch can route an incoming request/reply to the appropriate output port by examining a single digit of the destination address, as specified in base k.

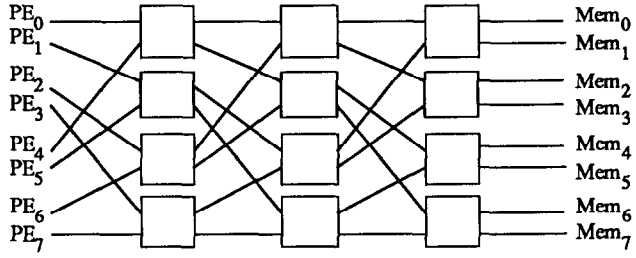


Fig. 1. 3 stage, 2×2 omega network.

Multistage networks may employ either circuit-switching, packet-switching or some hybrid strategy (such as virtual cut-through). With circuit switching, a circuit must be established between a processor and memory before data can be transferred. With packet-switching, packets of fixed size are transferred in a store-and-forward fashion through the stages of the network. Packet-switching networks may be either synchronous, in which case switches transfer packets only at times defined by discrete clock cycles, or asynchronous, in which case packet transfers may occur at arbitrary points in time. When two or more packets on different switch inputs concurrently require the same output port, a "conflict" is said to occur. There are two main approaches to handling such conflicts. The first is to simply discard all but one of the conflicting packets. The second approach is to supply buffers at switches so that packets will only be delayed rather than lost. In this paper, we are only interested in those multistage interconnection networks that are synchronous (clocked) packet-switching networks, and that utilize buffers to resolve conflicts.

There have been a considerable number of studies investigating the performance of multistage interconnection networks (see [2, 3, 8, 9, 10, 13, 14], among others). These studies have employed both analytical and simulation models.

Of major importance in the analytical modeling of clocked, buffered, packet-switched networks has been the work of Kruskal and Snir [9], whose analytical model of interconnection network performance has seen use in design studies of the IBM RP3 [15] and the NYU Ultracomputer [7].

In this paper, we develop an analytical model for clocked, buffered, packet-switched multistage interconnection networks that is based on queueing network models (QNM) and approximate Mean Value Analysis [12]. Our goal is to increase the range of applicability of analytical models in systems design. Previous models, although very useful, typically rely on a number of simplifying assumptions, including those of constant memory request generation rates at the network input ports (i.e., wholely nonblocking processors), and uniform memory reference patterns. (If non-uniform patterns are permitted, typically a special purpose analysis is required for each pattern of interest.) In addition, the interconnection network is typically modeled in isolation (although not in [8]) and, thus, the effects that other system resources (e.g., memory) will have upon performance are neglected. (Note that the service characteristics of these other resources may alter the pattern as well as the achieved rate of network traffic.)

The analytical model proposed in this paper is intended to have quite general applicability. Arbitrary network topologies and memory reference patterns are permitted, processors may have limits on the number of memory requests that may be outstanding, and, rather than modeling a multistage interconnection network in isolation, the entire multiprocessor (processors, memory and interconnection network) is modeled.

The remainder of this paper is organized as follows. Section 2 describes the overall modeling approach used. This includes a description of the particular machine architectures treated in Section 2.1, our conceptual system model in Section 2.2, and the model inputs and outputs in Sections 2.3 and 2.4, respectively. Section 3 develops our analytical model, and evaluates its accuracy, in the case where each memory request and reply can be contained in a single network packet. Section 4 considers the case where multiple packets are required for requests and replies. Finally, Section 5 concludes the paper.

2. Modeling Approach

2.1 Architectures Modeled

In this paper, as in [9, 10], we consider clocked (i.e., synchronous), packet-switched multistage interconnection networks in which each switch has buffers located at each of its output ports. A network switch is able to receive a packet from an input port and route it to the appropriate output port buffer queue in one clock cycle. We will assume here distinct "forward" (for requests sent from processors to memory modules) and "return" (for replies sent from memory modules to processors) networks, although a single network fulfilling both functions can also be modeled. Essentially arbitrary topologies for the networks can be treated, although in the presentation that follows it will be assumed that each processor and memory module pair are connected by a unique path in both the forward and return networks (i.e., banyan networks [6] are assumed), and that the forward and return networks have identical topologies. With respect to processor behavior, it is assumed that there is some fixed limit on the number of memory requests from a processor that may be outstanding before the processor must block to wait for the return of a reply. Whenever a processor has fewer than the maximum allowable number of requests outstanding it will generate a memory request during a clock cycle with some fixed probability.

The memory modules as well as the processors and interconnection network are modeled. Each memory module is assumed to have a single input port for receiving requests from the forward interconnection network, and a single output port through which replies are placed on the return network. The input port is assumed to be buffered. The memory service time (the number of clock cycles it takes to process a memory request once it reaches the head of the memory queue) is assumed to be deterministic.

In order to simplify the analysis, it is assumed that all buffers (at switch output ports and memory input ports) have unbounded length. Although this is clearly unrealistic, it is well known that, at least under uniform traffic, quite moderately sized buffers provide approximately the same performance as unbounded buffers. Current work involves extending our analysis to allow finite buffers.

2.2 Conceptual System Model

The proposed modeling approach employs a closed, multiclass queueing network model (QNM) [12] to represent the multiprocessor system. The customers in the QNM represent network requests and replies. The number of customer classes is equal to the number of processors; each customer class corresponds to the requests generated by one particular processor (i.e., processor i generates requests and receives replies (customers) of class i). The number of customers in each class is equal to the number of requests a processor may have outstanding before it must block and wait for some reply to return from memory. The routing patterns (visit ratios) of the QNM represent the topology of the interconnection network and the memory referencing patterns of the processors. The service centers of the QNM represent the processors, the memory modules, and the switch output ports (together with their associated output links) of the interconnection network.

The representation of the processors by (queueing) service centers requires further comment. The service time at such a center corresponds to the mean time between the generation of memory requests by the modeled processor (whenever it is not blocked waiting for a reply), so that as long as there are customers at the center, departures (modeling memory requests) will occur at a fixed rate. However, whenever the queue is emptied there cannot be any departures from the center, corresponding to the case where the processor has the maximum number of allowable requests outstanding and cannot generate a new one until a reply returns.

Performance measures for the multiprocessor are obtained from the representative QNM via approximate Mean Value Analysis (MVA) [12], with the appropriate reflections in the MVA mean residence time equations of service center peculiarities such as deterministic service times and synchronous arrivals.

2.3 Model Inputs

The analytical model requires inputs which can be classified into two basic categories; those that describe the particular hardware configuration to be studied and those that describe the expected workload.

The hardware configuration can be described by:

N — the number of processors.

M — the number of memory modules.

$k \times s$ — the size of the switches making up the forward and return interconnection networks (k-input, s-output).

network topology — a description of the topology of the interconnection networks and the connections to the processors and memory modules.

The workload can be described through the following parameters, the first three of which are assumed identical for all like resources for clarity of exposition:

NC — the maximum number of requests a processor can have outstanding before it must block to await a reply.

S_{pe} — the average processor interrequest time when not blocked (in clock cycles); interrequest times are assumed to be geometrically distributed.

S_{mm} — the memory service time (in clock cycles); service times are assumed to be deterministic.

P_{ij} — the memory referencing pattern to be studied. P_{ij} gives the probability that a request generated by processor i will be destined for memory module j. Obviously $\sum_j P_{ij} = 1$. Any arbitrary memory referencing pattern can be studied simply by modifying these inputs.

m — the request/reply size (number of packets). A constant size is assumed here, although our analysis can be easily extended to the case of different request and reply sizes.

The QNM used to represent the multiprocessor requires visit ratios (V_{ij}'s), where Vij is the probability of a class i customer visiting the jth queueing center. These visit ratios are not included in the list of inputs given above since they must be derived from both the network topology and memory referencing pattern, P_{ij}, to be studied. The visit ratio for a class i customer at processor center i is equal to one and at any other processor center j is zero since processor i only generates requests of class i. For a memory module center j, V_{ij} is just equal to the probability of processor i sending a request to memory module j (i.e., P_{ij}). The visit ratios for switch output port centers are easily calculated, once the path (forward and return) between each processor and memory module pair is extracted from the network topology description, through the following simple algorithm:

Initialize $V_{ij} = 0$ for all classes i and switch output port centers j

For each processor and memory module pair (i,m) **do**
  For each switch output port j on the path between i and m **do**
    $V_{ij} = V_{ij} + P_{im}$

2.4 Model Outputs

From the model inputs, we can derive approximate expressions for the following performance measures:

R_{ij} — the average "residence time" (queueing plus service) of a class i customer at service center j.

R_i — the average "response" time of a class i customer (time from the departure of a class i customer from the center representing processor i until its return time).

R — the average response time over all classes.

X_{ij} — the throughput of class i customers at center j.

X_i — the system throughput of class i customers.

X — the total system throughput (over all classes).

Q_{ij} — the average number of class i customers at center j (queued and in service).

U_{ij} — the average utilization of center j by class i customers.

In the following sections, we show that R_{ij} can be approximately expressed in terms of the other quantities (model inputs and outputs). Then, since the other outputs can be easily expressed in terms of R_{ij} and the model inputs, a set of non-linear equations can be developed which may be solved iteratively. The outputs may then be used to estimate the performance of the multiprocessor being studied.

3. Modeling Single Packet Requests/Replies

This section describes the proposed analytical model for the case where all requests and replies are composed of a single packet. Section 3.1 presents further assumed architectural details. Section 3.2 describes the details of the analytical model. Finally, Section 3.3 describes how the model was validated through the use of simulations and through consideration of special cases.

3.1 Architectural Details

The hardware architecture which we model is as described in Section 2.1. However, several details of how the processors, switches and memory modules are assumed to operate have been left until now since they can be more clearly described separately for the single packet requests/replies context and the multiple packet context. Below are the details of operation assuming single packet requests/replies.

A processor generates memory requests with an average interrequest delay of S_{pe} (geometrically distributed) clock cycles, provided that there are fewer than NC requests from that processor currently outstanding. If NC requests are outstanding, the processor is unable to send another until the next clock cycle after a reply returns. Each memory request requires S_{mm} clock cycles of service (a constant). Following these S_{mm} clock cycles, the next request in the memory module queue will begin service, and, in parallel, the reply for the request just served is sent to the first switch in the return network, where it will arrive at the end of the S_{mm} + 1st clock cycle.

Each switch input port can accept one packet from the connected output port of an upstream switch per clock cycle, and route it to the appropriate output port which has a FIFO buffer queue. Thus, the "service time" for a request/reply at a switch output port and its associated output link is one clock cycle. Conflicts occur when two or more input ports simultaneously try to route their incoming requests/replies to the same output port. These conflicts are resolved by queueing. Conflicting requests/replies are assumed to be placed in the FIFO buffer queue attached to the output port in a random order.

3.2 Analytical Model

This section describes, in detail, our proposed analytical model for the case of single packet requests and replies. We begin by developing equations approximating the average residence time that a customer will expect to incur at each QNM service center using approximate Mean Value Analysis[12]. There are three distinct residence time equations necessary; one for the centers representing the output ports of the switches (and their associated links), one for the centers representing the memory modules, and one for the centers representing the processors.

The average residence time (queueing plus service) for a class i customer at a switch output port center j can be approximated as follows:

$$R_{ij} = V_{ij}\Bigg[1 + \overbrace{\sum_{s\neq i}(Q_{sj} - U_{sj})}^{\text{term (a)}} + \overbrace{(Q_{ij} - U_{ij})\left[\frac{NC-1}{NC}\right]}^{\text{term (b)}} + \underbrace{\frac{1}{2}\sum_{k\in IN_j} p_{ikj}\left[\sum_{s\neq i}(1-p_{skj})X_{sj} + (1-p_{ikj})X_{ij}\left[\frac{NC-1}{NC}\right]\right]}_{\text{term (c)}}\Bigg] \quad (1)$$

where

p_{ikj} is the probability that a class i customer that passes through a switch output port center j arrives at that switch on input port k rather than some other input port (these probabilities can be calculated without too much difficulty from the visit ratios and network topology),

and IN_j is the set of switch input ports which are on the same network switch as output port j.

The initial 1 in the equation represents the service time at a switch output port center. Term (a) represents the average time spent queueing for customers of other classes that a class i customer finds in the queue (but not in service) during the clock cycle in which it arrives. Term (b) represents the average time spent queueing for other class i customers that a class i customer finds in the queue (but not in service) when it arrives. The factor $\frac{NC-1}{NC}$ is present here, as in other commonly used approximate MVA algorithms, since a new class i customer can never see itself in the queue [12]. Note that since the network switches are synchronous, and the service time is a single clock cycle, any customer in service during the arrival of a new customer will have departed by the beginning of the next clock cycle, and thus does not itself delay the new customer. Term (c) represents the average time spent queueing for other customers that arrive at the switch during the same clock cycle as an arriving class i customer, contend for the same output port center j, and are placed in the queue ahead of the class i customer. Since a switch is assumed to resolve conflicts by queueing in a random order, and since the service time is one, this quantity is equal to one half the expected number of other customers that arrive during the same clock cycle. This latter quantity is derived by noting that all such customers must arrive on other input ports, and that X_{sj} is the equilibrium probability of an arrival of a class s customer at switch output port center j.

The average residence time (queueing plus service) for a class i customer at the center representing memory module j can be approximated as follows:

$$R_{ij} = V_{ij}\Bigg[S_{mm} + \overbrace{\sum_{s\neq i}(Q_{sj} - U_{sj})}^{\text{term (a)}} + \overbrace{(Q_{ij} - U_{ij})\left[\frac{NC-1}{NC}\right]}^{\text{term (b)}}\Bigg]S_{mm} + \underbrace{\left[\sum_{s\neq i}U_{sj} + U_{ij}\left[\frac{NC-1}{NC}\right]\right]\left[\frac{S_{mm}-1}{2}\right]}_{\text{term (c)}} \quad (2)$$

The first S_{mm} represents the service time at a memory module center. Terms (a) and (b) serve the same purpose as in the switch residence time equation, but are multiplied by S_{mm} to reflect the multiple clock cycles that must be spent waiting for each queued customer to be served. Term (c) is the probability of encountering a customer in service, multiplied by the average remaining service time of such a customer. Note that since the network is synchronous the remaining service time is at most $S_{mm} - 1$, and that the total memory service time is deterministic.

Centers representing processors will be distinguished by subscripts of the form PE_i, as the residence time at these centers must be considered separately in what follows. A class i customer never visits any processor center except processor center i, so all R_{iPE_j} for $j \neq i$ are equal to zero. The average residence time (queueing plus service) for a class i customer at processor center i can be approximated as follows:

$$R_{iPE_i} = S_{pe} + \overbrace{(Q_{iPE_i} - U_{iPE_i})\left[\frac{NC-1}{NC}\right]S_{pe}}^{\text{term (a)}} + \underbrace{U_{iPE_i}\left[\frac{NC-1}{NC}\right](S_{pe}-1)}_{\text{term (b)}} \quad (3)$$

Similarly as before, the initial S_{pe} represents the center service time, while term (a) represents the average queueing delay due to other customers found in the queue but not in service. Term (b) gives the probability of encountering a customer in service, multiplied by the average remaining service time of such a customer. Note that since the network is synchronous the remaining service time is at most $S_{pe} - 1$, and that the total processor center service time is geometrically distributed.

The other equations that are needed for an approximate MVA analysis include the following equations for class throughputs and response times (derived from Little's Law [12]):

$$X_i = \frac{NC}{\sum_j R_{ij} + 1 + R_{iPE_i}}$$

$$R_i = \sum_j R_{ij} + 1$$

In both these equations, the sum is over only switch output port and memory module centers. The 1 in each equation represents the transmission time on the link from a memory module center to the network, which, unlike the case with the processors, is not included in the center service time. Other measures are obtained as follows (where center j may be a switch output port center, a memory module center or a processor center):

$$X_{ij} = V_{ij} \cdot X_i$$

$$Q_{ij} = X_i \cdot R_{ij}$$

$$U_{ij} = \begin{cases} X_{ij} & \text{for a center } j \text{ representing a switch output port} \\ X_{ij} \cdot S_{mm} & \text{for a center } j \text{ representing a memory module} \\ X_{ij} \cdot S_{pe} & \text{for a center } j \text{ representing a processor} \end{cases}$$

$$X = \sum_i X_i$$

$$R = \sum_i \frac{R_i \cdot X_i}{X}$$

This concludes the description of the equations that comprise the analytical model for the case where requests and replies are a single packet in length. These equations may be solved iteratively, as in other approximate MVA methods [12].

3.3 Validation

This section addresses the validity of the analytical model. First, some special cases are considered. Then, results are presented from several experiments in which simulation results are compared to the predictions of the analytical model.

Kruskal and Snir [9] obtain a residence time equation for a $k \times k$ switch in a banyan network assuming uniform traffic patterns and constant request generation rates (i.e., wholely nonblocking processors). It is interesting to note that under the assumptions made by Kruskal and Snir (uniform traffic, unbounded NC) our switch residence time equation reduces to one identical to theirs. That is, for a switch output port j, we get:

$$R_{ij} = V_{ij}\left[1 + \left[\frac{(1 - 1/k)\,X_j}{2(1 - X_j)}\right]\right]$$

where X_j is the throughput of switch output port center j.

This result helps to increase our confidence in approximate MVA (as contrasted to queueing theoretic) approaches for modeling interconnection networks. It should also be noted that for a 1×1 synchronous switch our equation reduces to $R_{ij} = V_{ij}$, which is exact since no queueing should occur in this case. Similarly, no queueing at processors is predicted for $S_{pe} = 1$, nor at memory modules for $S_{mm} = 1$, as desired.

In order to further validate the model, results were compared to those obtained from simulation. All experiments reported here used an omega interconnection network [11] for the forward network, and an identical network for the return network. In addition, the memory service times were consistently chosen to be quite low (1, 2 or 4 clock cycles). Although these low memory service times are quite unrealistic (memory is usually considerably slower than a network switch), they are necessary to allow consideration of cases where significant queueing occurs within the network. Such cases are important since, in practice, "system balance" will likely be achieved through the multiplexing of multiple memory modules on a single switch output link, which will result in similar effects (in terms of network loads that may be achieved) as one high speed memory module. The latter configuration is more of a stress test on model accuracy, however, so although both can be easily modeled, we have considered only the latter here.

Fig. 2(a) (average response time) and Fig. 2(b) (throughput) show analytical and simulation results for a 64 processor/memory module system using an omega network constructed from 2×2 switches, under a uniform memory reference pattern. Here, S_{pe} is chosen as 1, which is a stress test for our analytical model. Three values of S_{mm} were chosen, and NC was varied from 1 to 32. Note that the analytical results shown are very accurate, with less than 5 percent deviation from simulation results in all cases.

To illustrate the accuracy of the model with the same hardware configuration but under non-uniform memory reference patterns, Fig. 3 shows results for a "hot spot" [16] reference pattern in which 32 of the processors (0 to 31) are involved in "hot spot" contention for memory module 0. Each references the "hot" module with 20 percent higher probability than any other module, so that their probability of referencing module 0 is $(0.8/64 + 0.2)$. The other memory modules are referenced uniformly so that the probability of referencing a particular one is equal to 0.8/64. The 32 processors not involved in "hot spot" traffic reference each memory module except the "hot" module 0 (which they never reference) with probability 1/63. Again, the analytical model is quite accurate in its predictions.

Other experiments were performed with uniform memory reference patterns but different hardware configurations. Fig. 4 gives the results obtained for a 64 processor/memory module system using an omega network composed of 4×4 switches. Fig. 5 gives the results for a 128 processor/memory module system using an omega network of 2×2 switches. In all cases, the analytical results are seen to be very accurate.

The accuracy of the analytical results for the individual center average residence times is indicated in Tables 1 and 2 for the $S_{mm} = 1$ and $S_{mm} = 2$ cases of Fig. 2. Here, network stage $F1$ is the stage closest to the processors in the forward network, while stage $F6$ is the stage closest to the memory modules. Similarly, stage $R6$ is the stage closest to the memory modules in the return network, while stage $R1$ is the stage closest to the processors. Note that the model tends to underestimate residence times in the forward network (particularly in those stages closest to the memory modules) and overestimate residence times in the return network (again, particularly in those stages closest to the memory modules). This behavior may be explained by the fact that requests in the simulated forward network tend to bunch together, inflating residence times there. In the return network, on the other hand (particularly in those stages close to the memory modules), replies that potentially might collide often tend to be spaced apart by collisions of the corresponding requests in the (identical topology) forward network.

4. Modeling Multiple Packet Requests/Replies

In this section we develop our analytical model for the case where all requests and replies are m packets in length ($m \geq 1$). Section 4.1 presents further assumed architectural details. Section 4.2 describes the analytical model in detail. Finally, Section 4.3 presents validation results.

4.1 Architectural Details

The basic machine architecture is again as described in Section 2.1; however, several details of operation must be clarified here for the context of multiple packet requests/replies. A processor may generate a new request whenever it is not blocked (has fewer than NC outstanding requests) and is not already in the process of transmitting the packets of a request to

the first stage of the network. The parameter S_{pe}, in this context, becomes the mean delay from the time when the last (mth) packet of a request is sent until the first packet of the next request is sent (as before, this delay is geometrically distributed). If a processor is blocked due to having NC requests outstanding, it remains blocked until the last packet of a reply arrives - not the first packet.

A request can not begin service at a memory module until the last packet arrives. Once it does arrive, and there are no preceding requests in the queue, it will be served for S_{mm} (assumed to be greater than or equal to m) clock cycles. After these S_{mm} cycles, the memory module may begin servicing the next request (provided that all m packets of it have arrived) immediately on the next clock cycle. Meanwhile, in parallel, the first packet of the reply for the request just completed is sent to the first switch of the return network and the tail packets are sent on successive cycles.

Network switches operate in a similar fashion as in the single packet request/reply case. A packet can be accepted by an input port from the connected output port of an upstream switch and routed to the appropriate output port in one clock cycle. It is assumed that the packets constituting a request or reply are never split up by the network, implying that once the lead packet is served, the switch will give priority to all further packets of that request or reply arriving from the same input port. Any new requests/replies that arrive while the tail of a request/reply is arriving will be queued. When the lead packets of multiple requests/replies arrive simultaneously on different input ports of a switch, and are destined for the same output port, they are queued in a random order as in the single packet case. However, since each request is m packets long, each request other than that placed first will require m buffer locations where its tail packets will be placed as they arrive.

4.2 Analytical Model

As before, we begin our development with average residence time equations. For a switch output port center j, it is convenient to define R_{ij} as the average residence time of only the lead packet of a class i customer. The residence time of the entire customer is obtained from this simply by adding $(m-1)$ cycles to the per-visit delay, since all further packets follow immediately after the lead packet. We obtain:

$$R_{ij} = V_{ij}\Bigg[1 + \overbrace{\left[\sum_{s\neq i}(Q_{sj} - U_{sj}) + (Q_{ij} - U_{ij})\left[\frac{NC-1}{NC}\right]\right]m}^{\text{term (a)}} + \overbrace{\frac{1}{2}\sum_{k\in IN_j} p_{ikj}\left[\sum_{s\neq i}(1-p_{skj})X_{sj} + (1-p_{ikj})X_{ij}\left[\frac{NC-1}{NC}\right]\right]m + \sum_{k\in IN_j} p_{ikj}\left[\sum_{s\neq i}(1-p_{skj})U_{sj} + (1-p_{ikj})U_{ij}\left[\frac{NC-1}{NC}\right]\right]\left[\frac{m-1}{2}\right]}^{\text{term (b)}}\Bigg] \quad (4)$$

where

p_{ikj} and the set IN_j are defined as in the single packet case,

$$Q_{ij} = X_i V_{ij}\left[\frac{R_{ij}}{V_{ij}} + m - 1\right],$$

and $U_{ij} = X_{ij} \cdot m$.

The initial 1 represents the service time of the lead packet. Term (a) corresponds to terms (a) and (b) in the switch residence time equation for the single packet case (equation (1)), and represents the average time a class i customer spends queueing for other customers found in the queue but not in service. The factor m is required since each customer is m packets long. Term (b) corresponds to term (c) in equation (1), and represents the average time spent queueing for other customers which arrive during the same clock cycle, contend for the same output port, and are placed in the queue ahead of an arriving class i customer. Term (c) represents the probability of encountering a customer in service, multiplied by the average remaining service time of such a customer. (Note that term (c) does not reflect the small, but usually non-zero, probability of encountering a customer in service which had arrived on the same input port as the arriving class i customer. This probability is hard to estimate reliably.)

The average residence time equation for a class i customer at the center representing memory module j is as follows:

$$R_{ij} = V_{ij}\Bigg[(m-1+S_{mm}) + \overbrace{\left[\sum_{s\neq i}(\hat{Q}_{sj} - U_{sj}) + (\hat{Q}_{ij} - U_{ij})\left[\frac{NC-1}{NC}\right]\right]S_{mm}}^{\text{term (a)}} + \overbrace{\left[\sum_{s\neq i}U_{sj} + U_{ij}\left[\frac{NC-1}{NC}\right]\right]\left[\frac{(S_{mm}-m+1)(S_{mm}-m)}{2S_{mm}}\right]}^{\text{term (b)}}\Bigg] \quad (5)$$

term (c)

where

$\hat{Q}_{ij} = X_i V_{ij}\left[\frac{R_{ij}}{V_{ij}} - (m-1)\right]$, and represents the average number of complete customers in the queue or in service (excluding any partially arrived customers),

and $U_{ij} = X_{ij} \cdot S_{mm}$.

The $(m-1+S_{mm})$ term represents the memory service time of a customer plus (since a customer can not begin service until its last packet arrives) the time from when the lead packet of the customer arrives until its last packet arrives. Terms (a) and (b) multiplied by S_{mm}, as in equation (2), represent the average time spent queueing for other customers found in the queue but not in service. (Note that "partially arrived" customers cannot be found by an arriving customer since each memory module has only a single input port.) Term (c), as in the single packet case, is the probability of encountering a customer in service, multiplied by the average remaining service time of such a customer. This latter quantity is the average portion of the remaining service time at the time of the arrival of the lead packet that is still left once the tail packet arrives $m-1$ cycles later.

The average residence time for a class i customer at processor center i is as follows:

$$R_{iPE_i} = (m - 1 + S_{pe}) + \overbrace{(Q_{iPE_i} - U_{iPE_i})\left\lceil \frac{NC-1}{NC} \right\rceil (m - 1 + S_{pe})}^{\text{term (a)}} + \underbrace{U_{iPE_i}\left\lceil \frac{NC-1}{NC} \right\rceil \left\lceil \frac{(S_{pe}-1)S_{pe}}{S_{pe}+m-1} \right\rceil}_{\text{term (b)}} \quad (6)$$

where

$Q_{iPE_i} = X_{iPE_i} R_{iPE_i}$,

and $U_{iPE_i} = X_{iPE_i}(S_{pe} + m - 1)$.

Only the time until the first packet of a class i customer departs from processor center i is included in R_{iPE_i}, although it is important to note that the processor center is busy for an additional $m - 1$ cycles during which the remaining packets are transmitted. The initial $(m - 1 + S_{pe})$ represents the processor service time of a customer plus the time from when the lead packet of a customer arrives until its last packet arrives. Term (a) represents the average queueing delay due to customers found in the queue but not in service. (Note that Q_{iPE_i} excludes any partially arrived customers, but correspondingly includes any partially departed customers.) Term (b) gives the probability of encountering a customer in service multiplied by the average remaining service time of such a customer, which is derived in a corresponding fashion as for a memory center.

The other required equations include the following equations for X_i and R_i, derived using Little's Law:

$$X_i = \frac{NC}{\sum_j R_{ij} + 1 + R_{iPE_i}}$$

$$R_i = \sum_j R_{ij} + (m - 1) + 1$$

In both these equations, the sum is over only switch output port centers and memory module centers. The 1 in each equation represents the transmission time of the lead packet on the link from a memory module to the return network. The additional $(m - 1)$ in the R_i equation is the time for the last packet of a customer to catch up to the lead packet after the lead packet arrives back at the processor. (This is not included in the equation for X_i since it is incorporated into R_{iPE_i}.)
Other measures are obtained as follows:

$$X_{ij} = V_{ij} \cdot X_i$$

$$X = \sum_i X_i$$

$$R = \sum_i \frac{R_i \cdot X_i}{X}$$

This concludes our description of the required equations for the case of m packet requests/replies. As with the single packet case, these equations may be solved iteratively.

4.3 Validation

Kruskal and Snir derive a switch residence time equation for multiple packet requests in [9]. It is interesting to note that under the assumptions made by Kruskal and Snir (uniform traffic, unbounded NC) our switch residence time equation again reduces to one identical to theirs:

$$R_{ij} = V_{ij}\left[1 + \left\lceil \frac{(1 - 1/k)m^2 X_j}{2(1 - mX_j)} \right\rceil\right]$$

where X_j is the throughput of switch output port center j.

Also note that our switch residence time equation is correct for a 1×1 synchronous switch, our memory equation is correct for $S_{mm} = m$, and our processor equation is correct for $S_{pe} = m$ since in all of these cases no queueing is predicted.

In order to further validate the model, results were compared to those obtained from simulation for several different request/reply sizes ($m =$ 2, 4, 8). For each size, experiments were done with $S_{mm} = m$ (in which case no memory queueing occurs) and $S_{mm} = 2m$. In all cases, a 64 processor/memory module multiprocessor using an omega network with 2×2 switches was modeled, uniform memory reference patterns were assumed, and S_{pe} was set equal to 1 to test higher load situations. Figs. 6, 7, and 8 show response time and throughput results as NC is varied for the cases $m = 2$, $m = 4$, and $m = 8$, respectively. As in the single packet case, the analytical model is quite accurate in its performance predictions.

5. Conclusions

We have developed an analytical queueing network model, using approximate Mean Value Analysis, for estimating the performance of clocked, buffered, packet-switched multistage interconnection networks. Rather than modeling the network in isolation, the other system resources (memory modules and processors) are also explicitly modeled in order to capture the effects their service characteristics have upon the overall system performance. The analytical model permits very general interconnection network topologies (including arbitrary switch sizes), arbitrary memory reference patterns, and arbitrary request/reply sizes. In addition, various degrees of processor blocking may be modeled (i.e., a limit may be imposed on the number of outstanding requests a processor may have before it must block and wait for a reply to return from memory). The model was validated by comparisons to simulations and was shown to be quite accurate in its system performance predictions.

Current research efforts concern, in part, possible algebraic exploitation of regularities in memory referencing patterns in an attempt to reduce the number of system components which must be modeled (in our current implementation all switch output ports, memory modules and processors are explicitly modeled). Also of interest is the extension of our model to allow multiple distinct sizes for requests and replies. Of particular interest is the case where read requests, read replies, write requests, and write replies each have a potentially distinct size. Finally, work in progress concerns relaxing the assumption of unbounded buffer queues, and treating the bounded queue case.

Acknowledgements

This work was motivated by discussions with Mary Vernon, Ed Lazowska, and John Zahorjan concerning the performance analysis of shared memory multiprocessors. We thank these individuals, and the anonymous referees, for their helpful suggestions. In addition, we thank David Cargill who assisted in validating the multiple packet memory residence time equation.

References

[1] V. E. Benes, "Mathematical Theory of Connecting Networks and Telephone Traffic," Academic Press, New York, 1965.

[2] D. M. Dias and J. R. Jump, "Analysis and Simulation of Buffered Delta Networks," IEEE Trans. Comput., Vol. C-30, pp. 273-282, Apr. 1981.

[3] D. M. Dias and J. R. Jump, "Packet Switching Interconnection Networks for Modular Systems," Computer, Vol. 14, pp. 43-53, Dec. 1981.

[4] T. Y. Feng, "A Survey of Interconnection Networks," Computer, Vol. 14, pp. 12-27, Dec. 1981.

[5] D. Gajski, D. Kuck, D. Lawrie and A. Sameh, "Cedar - A Large Scale Multiprocessor," Proc. 1983 Int'l Conf. Parallel Processing, pp. 524-529, 1983.

[6] L. R. Goke, and G. L. Lipovski, "Banyan Networks for partitioning multiprocessor systems," Proc. 1st Annual Symp. on Computer Architecture, pp. 21-28, Dec. 1973.

[7] A. Gottlieb, R. Grishman, C. P. Kruskal, K. P. McAuliffe, L. Rudolph and M. Snir, "The NYU Ultracomputer - Designing an MIMD Shared Memory Parallel Computer," IEEE Trans. Comput., Vol. C-32, pp. 175-189, Feb. 1983.

[8] D. T. Harper and J. R. Jump, "Performance Evaluation of Reduced Bandwidth Multistage Interconnection Networks," Proc. 14th Annual Symp. on Computer Architecture, pp. 171-175, 1987.

[9] C. P. Kruskal and M. Snir, "The Performance of Multistage Interconnection Networks for Multiprocessors," IEEE Trans. Comput., Vol. C-32, pp. 1091-1098, Dec. 1983.

[10] C. P. Kruskal, M. Snir and A. Weiss, "The Distribution of Waiting Times in Clocked Multistage Interconnection Networks," IEEE Trans. Comput., Vol. C-37, pp. 1337-1352, Nov. 1988.

[11] D. H. Lawrie, "Access and Alignment of Data in an Array Processor," IEEE Trans. Comput., Vol. C-24, pp. 1145-1155, Dec. 1975.

[12] E. D. Lazowska, J. Zahorjan, G. S. Graham, K. C. Sevcik, "Quantitative System Performance," Prentice Hall, New Jersey, 1984.

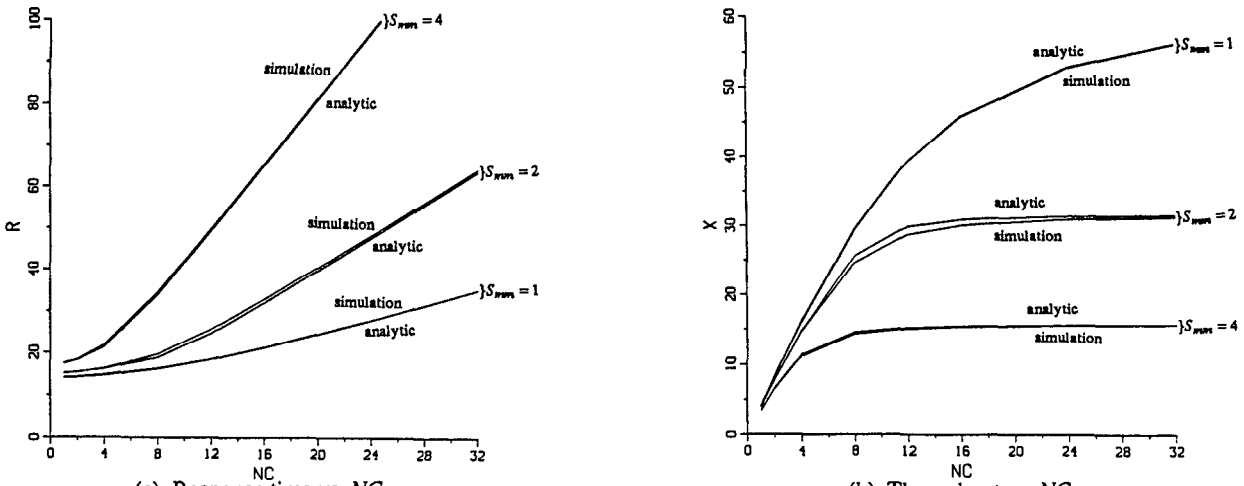
[13] Y. Liu and S. Dickey, "Simulation and Analysis of Different Switch Architectures for Interconnection Networks in MIMD Shared Memory Machines," Ultracomputer Note #141, June 1988.

[14] J. A. Patel, "Performance of Processor-Memory Interconnections for Multiprocessors," IEEE Trans. Comput., Vol. C-30, pp. 771-780, Dec. 1981.

[15] G. F. Pfister, W. C. Brantley, D. A. George, S. L. Harvey, W. J. Kleinfelder, K. P. McAuliffe, E. A. Melton, V. A. Norton and J. Weiss, "The IBM Research Parallel Processor Prototype (RP3): Introduction and Architecture," Proc. 1985 Int'l Conf. Parallel Processing, pp. 764-771, 1985.

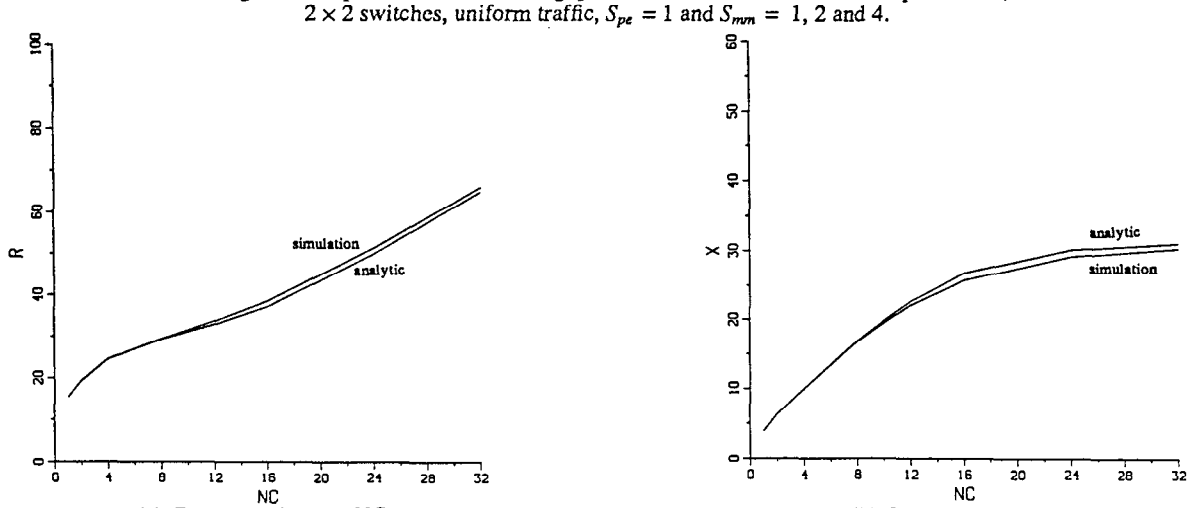
[16] G. F. Pfister and V. A. Norton, "'Hot Spot' Contention and Combining in Multistage Interconnection Networks," Proc. 1985 Int'l Conf. Parallel Processing, pp. 790-797, 1985.

[17] R. Rettberg and R. Thomas, "Contention is No Obstacle to Shared-Memory Multiprocessing," Communications of the ACM, Vol. 29, No. 12, pp. 1202-1212, Dec. 1986.



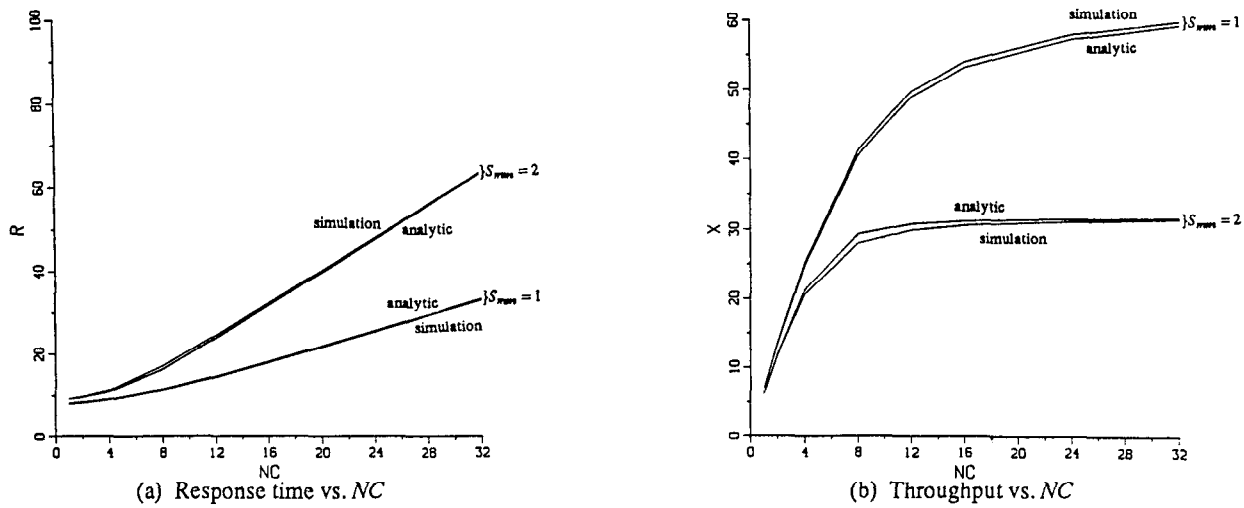
(a) Response time vs. *NC* (b) Throughput vs. *NC*

Fig. 2. Response time and throughput as functions of *NC* with 64 processors, 2×2 switches, uniform traffic, $S_{pe} = 1$ and $S_{mm} = 1$, 2 and 4.



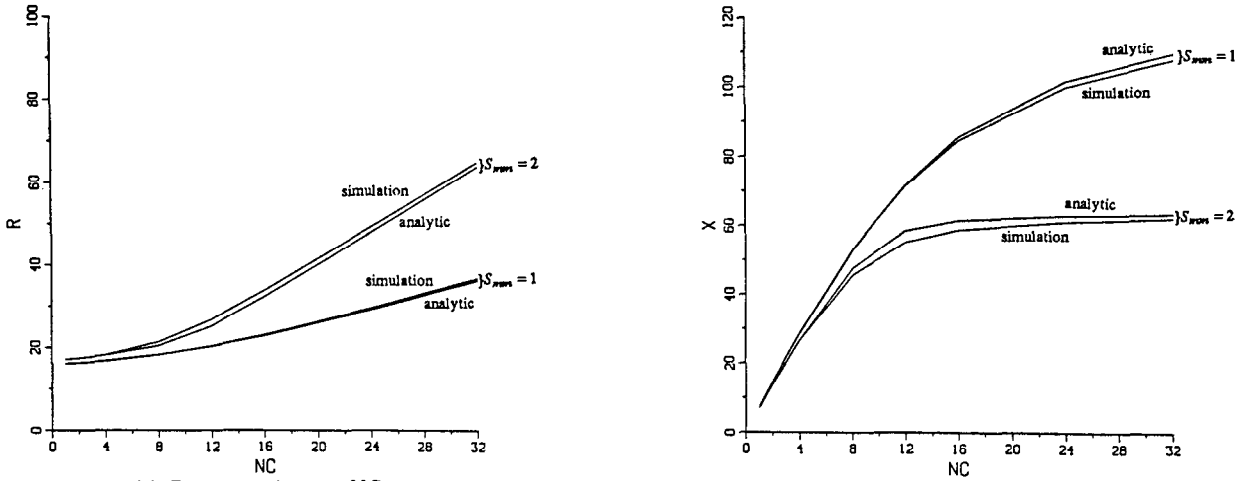
(a) Response time vs. *NC* (b) Throughput vs. *NC*

Fig. 3. Response time and throughput as functions of *NC* with 64 processors, 2×2 switches, "hot spot" traffic, $S_{pe} = 1$ and $S_{mm} = 2$.



(a) Response time vs. *NC* (b) Throughput vs. *NC*

Fig. 4. Response time and throughput as functions of *NC* with 64 processors, 4×4 switches, uniform traffic, $S_{pe} = 1$ and $S_{mm} = 1$ and 2.



(a) Response time vs. NC (b) Throughput vs. NC

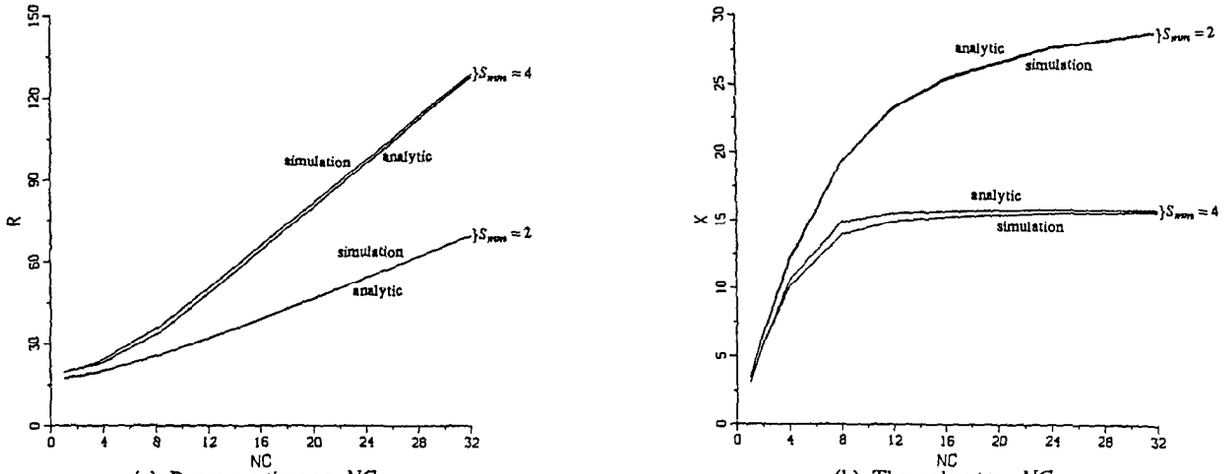
Fig. 5. Response time and throughput as functions of NC with 128 processors, 2×2 switches, uniform traffic, $S_{pe} = 1$ and $S_{mm} = 1$ and 2.

Network Stage	Average Residence Time									
	$NC = 2$		$NC = 4$		$NC = 8$		$NC = 16$		$NC = 32$	
	analytic	simulation	analytic	simulation	analytic	simulation	analytic	simulation	analytic	simulation
F1	1.035	1.009	1.080	1.040	1.201	1.150	1.588	1.563	2.659	2.772
F2	1.036	1.021	1.082	1.071	1.206	1.221	1.612	1.714	2.771	3.189
F3	1.036	1.032	1.082	1.087	1.209	1.241	1.624	1.760	2.830	3.335
F4	1.037	1.036	1.083	1.091	1.210	1.248	1.630	1.783	2.861	3.380
F5	1.037	1.038	1.083	1.093	1.211	1.253	1.633	1.792	2.877	3.405
F6	1.037	1.039	1.083	1.095	1.211	1.254	1.635	1.797	2.885	3.391
R6	1.036	1.004	1.082	1.021	1.210	1.105	1.632	1.401	2.875	2.099
R5	1.035	1.007	1.081	1.036	1.209	1.153	1.628	1.496	2.858	2.317
R4	1.034	1.009	1.080	1.046	1.206	1.172	1.619	1.537	2.823	2.443
R3	1.032	1.009	1.076	1.049	1.200	1.178	1.602	1.548	2.757	2.475
R2	1.026	1.008	1.070	1.046	1.188	1.166	1.569	1.518	2.634	2.388
R1	1.017	1.005	1.057	1.038	1.167	1.145	1.508	1.443	2.414	2.127
Memory	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Average Response Time R	14.409	14.223	14.950	14.718	16.437	16.291	21.293	21.358	35.270	35.327

Table 1. Average residence time at each network stage and at the memory modules. 64 processors/memory modules, 2×2 switches, uniform traffic, $S_{pe} = 1$, $S_{mm} = 1$.

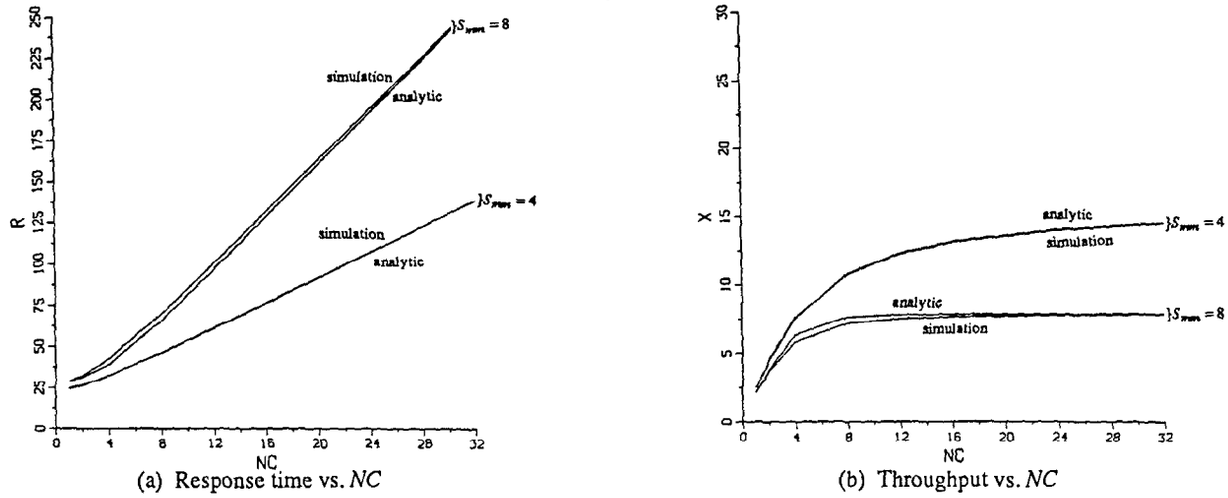
Network Stage	Average Residence Time									
	$NC = 2$		$NC = 4$		$NC = 8$		$NC = 16$		$NC = 32$	
	analytic	simulation	analytic	simulation	analytic	simulation	analytic	simulation	analytic	simulation
F1	1.033	1.012	1.072	1.041	1.160	1.115	1.227	1.191	1.241	1.222
F2	1.033	1.023	1.073	1.065	1.164	1.160	1.230	1.242	1.243	1.270
F3	1.034	1.030	1.074	1.075	1.165	1.174	1.232	1.255	1.244	1.282
F4	1.034	1.034	1.074	1.079	1.166	1.178	1.233	1.260	1.244	1.285
F5	1.034	1.035	1.075	1.080	1.167	1.179	1.233	1.262	1.244	1.286
F6	1.034	1.035	1.075	1.082	1.167	1.179	1.234	1.262	1.244	1.287
R6	1.033	1.013	1.074	1.040	1.166	1.089	1.233	1.115	1.244	1.120
R5	1.033	1.014	1.073	1.051	1.165	1.140	1.232	1.210	1.243	1.228
R4	1.031	1.015	1.072	1.056	1.163	1.155	1.230	1.236	1.243	1.259
R3	1.029	1.015	1.069	1.056	1.158	1.153	1.227	1.240	1.241	1.266
R2	1.024	1.013	1.063	1.052	1.150	1.144	1.220	1.231	1.237	1.263
R1	1.016	1.008	1.052	1.041	1.135	1.123	1.207	1.207	1.230	1.249
Memory	2.157	2.240	2.427	2.674	4.008	4.949	16.339	17.266	47.764	48.199
Average Response Time R	15.536	15.493	16.284	16.399	18.946	19.747	32.092	32.994	63.687	64.289

Table 2. Average residence time at each network stage and at the memory modules. 64 processors/memory modules, 2×2 switches, uniform traffic, $S_{pe} = 1$, $S_{mm} = 2$.



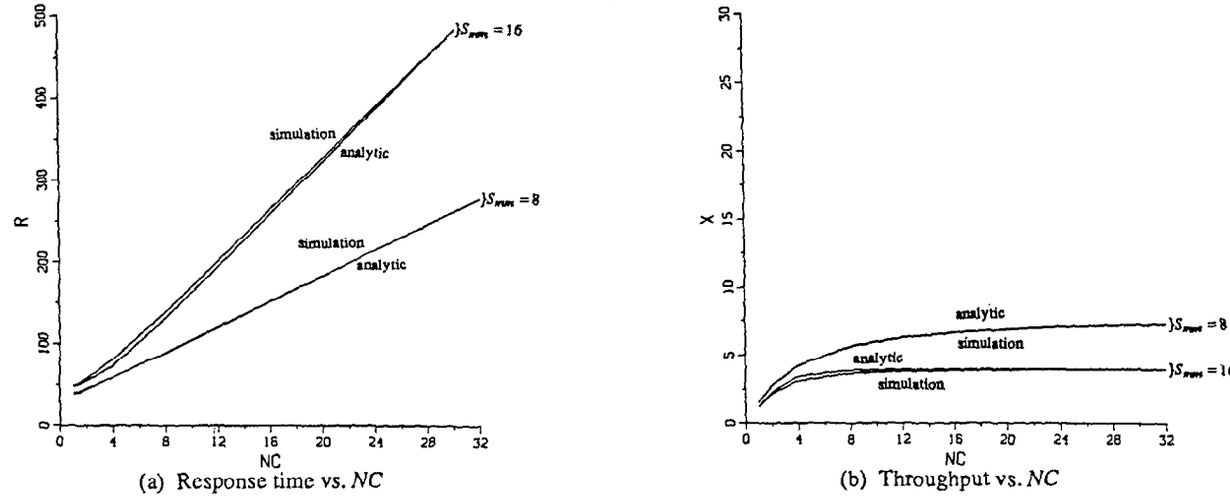
(a) Response time vs. *NC* (b) Throughput vs. *NC*

Fig. 6. Response time and throughput as functions of *NC* with 64 processors, 2×2 switches, uniform traffic, $S_{pe} = 1$, $m = 2$ and $S_{mm} = 2$ and 4.



(a) Response time vs. *NC* (b) Throughput vs. *NC*

Fig. 7. Response time and throughput as functions of *NC* with 64 processors, 2×2 switches, uniform traffic, $S_{pe} = 1$, $m = 4$ and $S_{mm} = 4$ and 8.



(a) Response time vs. *NC* (b) Throughput vs. *NC*

Fig. 8. Response time and throughput as functions of *NC* with 64 processors, 2×2 switches, uniform traffic, $S_{pe} = 1$, $m = 8$ and $S_{mm} = 8$ and 16.