Effect of nonuniform traffic on the performance of multistage interconnection networks

M. Atiquzzaman
M.S. Akhtar

Indexing terms: Multistage interconnection networks, Multiprocessor systems, Nonuniform memory reference

Abstract: Multistage interconnection networks are used to connect processors to memories in shared memory multiprocessor systems. The performance evaluation of such networks is usually based on the assumption of a uniform memory reference pattern. Hot spots in such networks give rise to a nonuniform memory reference pattern and result in a degradation in performance. A comparison of the performance of unbuffered and buffered networks under a nonuniform traffic pattern is given. Analytical models have been developed for the evaluation of performance. An analytical model for unbuffered networks is developed in this paper, while the model for buffered networks is presented elsewhere. Results from the models are used to find the impact of different degrees of hot spot traffic and network size on the performance of the network. It is shown that an unbuffered network may perform better than a buffered network under a nonuniform traffic pattern. Finally, a hybrid mode network is suggested for optimum performance under different traffic conditions.

1 Introduction

In recent years, there has been a significant increase in interest in the use of multiprocessor systems. Tightly-coupled multiprocessor systems with a variety of interconnection networks have been proposed, analysed, and built in the last two decades [1]. Because of the modularity, simplicity and fault-tolerant capabilities of crossbar and multiple-bus systems, such systems have been widely investigated [2]. The disadvantage of the crossbar is the large number of switches required, $O(N^2)$ switches are required to connect N processors to N memories.

Multistage interconnection networks reduce the number of switches to $O(N \log N)$. A Delta network has been proposed by Patel [3]. It is an $a^n \times b^n$ switching network of n stages constructed using $a \times b$ crossbar switches. An Omega network has been proposed by Lawrie [4] which uses $\log_2 N$ stages of switches, each stage consisting of $N/2$ crossbar switches of size 2×2. A perfect shuffle permutation [5] is used to connect the switches in adjacent stages. Delta, Omega, indirect binary n-cube [6] etc. are subsets of the Banyan network [7].

The omega network is a blocking type of network where contention arising at a switch results in performance degradation. Performance evaluation of Omega and Delta networks have been reported [3, 8–12] using both analytical modelling and simulation techniques. Most of the work reported assumes a uniform memory reference model (URM), i.e. a request issued by a processor has an independent and equal probability, equal to $1/N$, of being directed to any of the N memory modules. Patel [3] has analysed the unbuffered Delta network using a recursive technique. Analytical and simulation results on the performance of buffered Delta networks are presented in Reference 8. Modelling of the buffered network is usually carried out using Markov chains. Banyan networks have been analysed [13]. Thanawastien and Nelson [9] have analysed a buffered network under synchronous and asynchronous operations.

URM is rather restrictive and does not hold in many real-world applications. For example, each processor may have a different local/private memory which it accesses more frequently than others. This type of configuration, called the favourite memory, has been studied by Bhuyan [10], Du [11] and Chen and Sheu [12].

Nonuniform memory references arise in multiprocessor systems due to shared variables used for locking, global and barrier synchronisation, pointers to shared queues etc. These are indivisible primitives and must be stored in a single shared memory. The primitives are accessed by all processors, giving rise to an increased request rate for the memory module which contain them. These memory modules are called hot memories, and the phenomenon which is known as hot spot contention was first reported by Pfister and Norton [14]. In the case of a buffered network, a phenomenon called tree saturation severely degrades the performance of the network. Combining and feedback schemes have been suggested as solutions to the problem [15–17]. In the case of an unbuffered network, the performance degradation is due to a high rate of contention at the switches carrying the hot memory traffic.

The first objective of this paper is to evaluate the performance of an unbuffered Omega network under hot spot traffic. This will give an insight into the degree of degradation in the performance of the network due to hot spot traffic as compared to a network operating under a uniform traffic pattern. An analytical model has been developed to measure the performance of such a network. A recursive technique is used to develop the model. Although the analysis is carried out for the Omega network, the approach can easily be modified to analyse

Paper 9999E (C2), first received 5th January and in revised form 11th October 1993

M. Atiquzzaman is with the Department of Computer Science, La Trobe University, Melbourne, Australia 3083

M.S. Akhtar is with the Department of Electrical Engineering, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia

other multistage interconnection networks like the indirect binary n-cube, Banyan network etc.

It has previously been reported that buffered multistage networks have higher bandwidths than unbuffered networks. This conclusion was based on the assumption of uniform traffic. A comparison between the buffered and unbuffered networks under nonuniform traffic is not available in the literature. The second objective of this paper is to compare the performance of an unbuffered network with that of a buffered network when both networks are operating under a nonuniform traffic pattern. Hot spot traffic will be used as an example of nonuniform traffic for this study. It will be shown that an unbuffered network may perform better than a buffered network under a nonuniform traffic pattern. Consequently, a hybrid mode multistage network is proposed to optimise the performance of the network for both uniform and nonuniform traffic patterns.

The performance figures for the unbuffered network are obtained from the analytical model to be described herein. The figures for the buffered network are obtained from a different analytical model described in Reference 18–20.

2 Model

An Omega network [4] will be used as an example of a multistage interconnection network (MSIN). This section describes the Omega network, followed by the assumptions under which the analytical model is developed.

2.1 The Omega network

The Omega network is a subset of the Delta network originally proposed by Patel [3]. It is used to connect N processors to N memories using $n = \log_2 N$ stages of $N/2$ switches per stage, each switch having two input and two output lines. The kth stage of switches will be denoted by S_k, $0 \leqslant k \leqslant n-1$. A perfect shuffle permutation is used to connect adjacent stages as shown in Fig. 1 for $N = 8$.

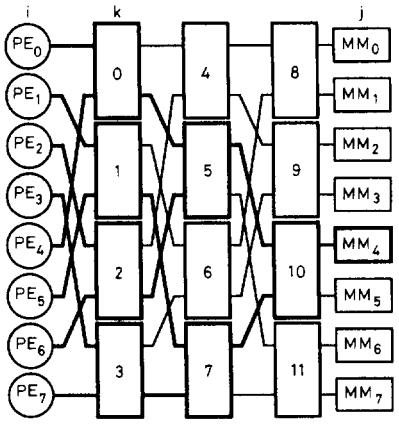

Fig. 1 *Omega network for $N = 8$*

Each switch is a 2 × 2 crossbar switch allowing any input link to be connected to any one of the output links. A conflict arises if both the inputs need to be connected to the same output simultaneously. Arbitration schemes are used to resolve such conflicts.

Requests generated from processors are bundled into packets consisting of the data and the destination address. The destination address is an n-bit number represented by $D = (d_0 d_1 d_2, \ldots, d_{n-2} d_{n-1})_2$. Destination tag routing is used to route the packets through the network. A switch at S_k inspects bit d_k and, in the case of no conflict, routes the message to the upper or lower output of the switch depending on whether d_k is 0 or 1, respectively. A unique path of constant length exists between a processor and a memory.

2.2 Assumptions

For modelling purposes, the following assumptions are made regarding the network and its operation.

(i) There are $N = 2^n$ processors and N memory modules in the system, where n is an integer. The ith processor and the jth memory will be denoted by PE_i and MM_j, respectively, where $0 \leqslant i, j \leqslant N-1$.

(ii) Synchronous operation is assumed. All processors request memories at the beginning of a memory cycle. The service times of all memory modules are also equal.

(iii) Packet switching is assumed for the routing of messages. A packet contains both the data and the destination address.

(iv) Fair routing arbitration logic at the switches is assumed. In case of conflict at any switch, the switch randomly selects one input, and the rejected (or blocked) one is ignored. This also implies unbuffered switches.

(v) Temporal independence of requests is assumed, i.e., the request generated by a processor in a cycle is independent of whether requests at previous cycles were accepted or rejected (blocked).

(vi) Memory requests are assumed to be spatially independent, i.e., requests generated by a processor are independent of requests generated by other processors.

(vii) Processors generate random request at the beginning of a memory cycle. The probability that a processor generates a memory request at the beginning of a cycle is p_0. Thus p_0 is the average number of requests generated per cycle by each processor.

(viii) Memory reference patterns are uniform except for the hot module. MM_h is a hot memory module for all processors. If PE_i generates a request, the probability that it will request MM_h is q, whereas, the probability of requesting any module MM_j, $j \neq h$, is $q' = (1-q)/(M-1)$. This implies that the nonhot memory modules are equally likely to be referenced. Also $q > q'$. Note that $q = q'$ is the well known uniform memory reference model assumed by most authors.

(ix) Because of nonuniform memory reference pattern, packets arriving at the inputs to a switch are not necessarily uniformly distributed over the outputs of the switches.

In practice, rejected requests are resubmitted in the next cycle. The assumption of temporal independence makes the analysis simple without introducing too many errors as is evident from the studies reported in References 3, 21 and 10. Analyses of even more complex systems for similar problems [22–26] have shown that the assumption of temporal independence introduces negligible errors. Thus, models based on the assumption of temporal independence provide a good measure for comparing different networks.

3 Properties

In the case of URM, the data rates at all the links between any two stages are the same [3]. Due to the nonuniform memory reference pattern, data rates at the different links between any two stages are not the same.

The rates at the different links of the network will be evaluated in Section 4. In this section, some definitions and important properties of the network under hot-spot conditions are presented. Fig. 1 shows an Omega network for $N = 8$ with MM_4 being the hot memory module.

Definition 1: Requests directed for hot memory module are called hot requests. In Fig. 1, requests for MM_4 are called hot requests.

Definition 2: The links which carry hot requests are called hot links. The hot links in Fig. 1. are shown as thick lines.

Definition 3: All switches connected to hot links are called hot switches. The hot switches in Fig. 1 are shown as thick boxes.

Theorem 1: All the hot links at the output of a stage are either the lower outputs or the upper outputs of that stage.

Proof: If the address of the hot memory is $d_0 d_1, \ldots, d_k, \ldots, d_{n-1}$, then all the hot switches at S_k will route the hot request to either the upper or lower output link depending on whether d_k is 0 or 1. Therefore, the hot link outputs of a stage are either the lower or upper outputs.

Theorem 2: Both the inputs to a switch at S_k, $1 \leqslant k \leqslant n-1$ are either lower outputs or upper outputs of switches at S_{k-1}.

Proof: It follows directly from the construction of the Omega network where perfect shuffle interconnection between stages is used.

Theorem 3: The hot switches and hot links form a bipartite 'hot tree' rooted at the hot memory. All the processors form the leaves, and the hot switches and hot links form the vertices and edges, respectively, of the tree. There are 2^n paths of this tree, the paths correspond to the paths traversed by hot requests from the 2^n processors.

Proof: In an Omega network, there is a unique path from any processor to any memory, and hence there are unique paths from the processors to a particular memory. This gives rise to a tree, the edges of which are hot links forming the paths from the different processors.

Theorem 4: The data rates at the outputs of a hot switch are different.

Proof: The two output links of a hot switch carry requests for two disjoint sets of memories. The output link carrying requests for the set containing the hot memory will have higher data rate than the other output link carrying requests for the memory set containing non-hot memories. As an example, the upper and lower output links of switch 0 in Fig. 1 carry traffic for the memory sets $\{MM_0, MM_1, MM_2, MM_3\}$ and $\{MM_4, MM_5, MM_6, MM_7\}$, respectively. The data rate at the lower link will be higher than the upper one because of hot requests being routed to the lower link.

Theorem 5: The data rates at the outputs of a nonhot switch are the same.

Proof: The two output links carry traffic for two disjoint sets of memories which have the same probabilities of being requested because all the memories in the two sets are equally likely to be referenced by the processors (see assumption (viii) in Section 2.2).

Theorem 6: Both output links of a switch are never hot—either both of them are nonhot, or one hot and one nonhot.

Proof: Nonhot switches do not carry hot traffic, and hence both outputs are nonhot links. Hot requests arriving at a hot switch will be routed to the hot output link which may also contain nonhot requests, whereas, only nonhot requests will be routed to the other output. Therefore, one output of a hot switch is a hot link while the other one is a nonhot link.

Theorem 7: Both inputs to a switch will have the same data rate.

Proof: Since both the inputs to any switch carry traffic originating from an equal number of identical processors and directed towards the same set of memory modules, the data rate will be equal.

Lemma 1: The number of hot switches reduce by half at succeeding stages, and the number of hot links also reduce by half at the outputs of succeeding stages. The number of hot links at the output of stage S_k is $2^n/2^{k+1}$, and the number of hot switches at stage S_k is 2^{n-1-k}.

Proof: Since the hot switches and the hot links form a bipartite hot tree (see theorem 3), the number of hot switches and hot links reduce by half towards the root of the tree.

Lemma 2: Both inputs to a switch are either hot links or nonhot links.

Proof: All hot switches are nodes of the hot tree and will have both inputs as hot links. The nonhot switches are not part of the hot tree and hence will have nonhot input links.

Lemma 3: If there is a packet at any input x_u, $0 \leqslant u \leqslant 1$, to a switch, the probability that it will be routed to output y_v, $0 \leqslant v \leqslant 1$, is $r_v / \sum_{w=0}^{1} r_w$, where r_w is the sum of the probabilities with which a processor requests the set of memories reachable from the output y_w.

Proof: This follows directly from the probability scaling-up theorem of elementary probability theory.

4 Performance analysis

The performance of the network will be measured by the average memory bandwidth (AMBW) of the network. The AMBW is defined as the expected number of memory modules active during any memory cycle.

$$AMBW = \sum_{k=1}^{N} \beta q(\beta) \tag{1}$$

where, $q(\beta)$ is the probability of β memory modules being active. It can also be expressed as

$$AMBW = \sum_{j=0}^{N-1} p(j) \tag{2}$$

where, $p(j)$ is the data rate at the input link of the jth memory module. Using the properties of the network described in Section 3, we will calculate the $p(j)$s for the different memory modules, and then calculate the AMBW using eqn. 2. Starting with the rate at which the PEs request memory (also called the request rate), the data rates at the different stages of the network will be calculated recursively. Since each switch is a 2 × 2 crossbar, the data rate at each output of a switch can be determined if the input data rates to the switch and the probability of routing to the outputs are known. The rest of this section presents a recursive method of calculating the data rates at the different stages.

Assume that each processor has a request rate of $p_{-1,0}$. The requests are fed to the switches at S_0 (see Fig. 1). A request for MM_j, arriving at the input of a switch at S_0 will be routed to the upper or lower output link depending on whether $j \leqslant N/2 - 1$ or $j \geqslant N/2$, respectively. Either the upper or the lower output link of S_0 will be hot links (see theorem 6) depending on whether $h \leqslant N/2 - 1$ or $h \geqslant N/2$, respectively. Note that, the input links to all the switches at S_0 are hot. According to lemma 1, $N/2$ of the output links of S_0 will be hot resulting in $N/4$ of the switches at S_1 being hot.

According to theorem 4, the data rates at the two output links of a switch at S_0 will be different. Let these data rates be $p_{0,0}$ and $p_{0,1}$ for the hot and the nonhot output links, respectively.

By lemma 2, both inputs to a switch at any stage are either hot or nonhot links. Therefore, in S_1, the $N/4$ nonhot switches will have nonhot input links with data rates $p_{0,1}$, and the $N/4$ hot switches will have hot input links with data rates $p_{0,0}$. Let the data rates at the hot and nonhot output links of a hot switch at S_1 be denoted by $p_{1,0}$ and $p_{1,1}$, respectively. According to theorem 6, at S_1 there will be $N/4$ hot output links having data rates of $p_{1,0}$ and $N/4$ nonhot output links with data rates $p_{1,1}$. Each of the $N/4$ nonhot switches in S_1 will have input data rates of $p_{0,1}$. According to theorem 5, both the outputs of any of these switches will have the same data rate, say $p_{1,2}$. Therefore, there will be $N/2$ nonhot output links with data rates $p_{1,2}$. Note that $p_{1,2}$ depends only on $p_{0,1}$, whereas $p_{1,0}$ and $p_{1,1}$ depend on $p_{0,0}$ and the distribution of requests to the two outputs of a hot switch.

We observe that the output links of S_0 and S_1 have two and three different data rates. Let P_k denote the set of data rates at the output links of stage k. For example, $P_1 = \{p_{1,0}, p_{1,1}, p_{1,2}\}$. At the output of S_1 there are:

(i) $N/2$ nonhot links with data rates $p_{1,2}$. These are produced by nonhot switches having nonhot input links with data rates $p_{0,1}$.

(ii) $N/4$ nonhot links with data rates $p_{1,1}$. These are produced by hot switches having hot input links with data rates $p_{0,0}$.

(iii) $N/4$ hot links with data rates $p_{1,0}$. These are produced by hot switches having hot input links with data rates $p_{0,0}$.

Following the above procedure, we can recursively calculate the data rates at the output links of all the stages. We note that $|P_{k+1}| = |P_k| + 1$, where, $|P_k|$ denotes the number of elements of P_k. This is due to the generation of two different data rates at the outputs of a hot switch.

Let us denote the data rates at the hot and nonhot output links of a hot switch at S_k by $p_{k,0}$ and $p_{k,1}$, respectively. These are produced by hot switches having hot input links with data rates $p_{k-1,0}$. Let the other nonhot data rates at the outputs of S_k be $p_{k,2}, p_{k,3}, \ldots, p_{k,k+1}$ which are dependent on the data rates $p_{k-1,1}, p_{k-1,2}, \ldots, p_{k-1,k}$, respectively at the nonhot input links of the nonhot switches. This input-output data rate dependency for the different stages can be expressed by a data rate dependency tree as shown in Fig. 2. The tree shows

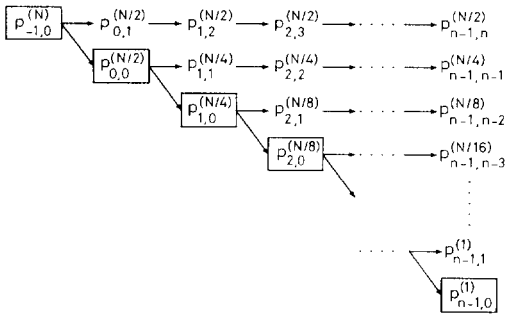

Fig. 2 *Data rate dependency tree for N × N omega network*

which output data rates of a stage depend on which input data rates of that stage. The data rates in boxes are those at the hot links. The data rates in bold typeface are those which depend not only on the data rate indicated in the tree, but also on the ratio of q to q' as will be explained below. The superscript associated with a data rate shows the number of links at that stage having that data rate.

Theorem 8: $|P_k| = k + 2$, i.e., at the output of S_k, $0 \leqslant k \leqslant n - 1$, there will be a total of $k + 2$ different data rates from $p_{k,0}$ to $p_{k,k+1}$. The number of output links having nonhot data rate $p_{k,s}$, $1 \leqslant s \leqslant k + 1$, is $N/2^{k-s+2}$. The number of links having the hot data rate $p_{k,0}$, $0 \leqslant k \leqslant n - 1$ is $N/2^{k+1}$.

The data rates at the output links of S_0 and S_1 which are determined first will be used to generalise the data rates at the output link of any stage. Let us consider a switch at S_0 as shown in Fig. 3 with the input and output

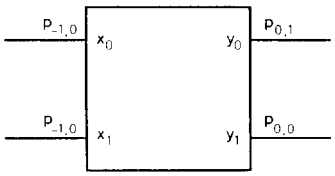

Fig. 3 *Typical 2 × 2 crossbar switch used in omega networks*

data rates as marked on the links. For the purpose of illustration, assume that $h \geqslant N/2$. Then y_1 will be the hot output link and will be labelled with $p_{0,0}$. (Note that if $h < N/2$, y_0 will be labelled with $p_{0,0}$). The data rate at y_1 is equal to the probability that there will be a request routed at the y_1 output, and is given by

$$\begin{aligned} p_{0,0} = {} & \Pr[x_0 \rightarrow y_1] \Pr[x_1 \rightarrow y_1] \\ & + \Pr[x_0 \rightarrow y_1] \Pr[x \not\rightarrow y_1] \\ & + \Pr[x_1 \rightarrow y_1] \Pr[x_0 \not\rightarrow y_1] \end{aligned} \tag{3}$$

where, $\Pr[x_u \rightarrow y_v]$ denotes the probability that a memory request is routed to y_v from x_u, and $\Pr[x_u \not\rightarrow y_v]$ denotes the probability that no request is routed to y_v from x_u. $\Pr[x_0 \rightarrow y_1] = \Pr$ [a message is present at x_0] $\times \Pr$ [the message at x_0 is routed to y_1]. By lemma 3, provided that a message has arrived at x_0, the pro-

bability that the message is routed to y_0 is $((N/2)q')/((N-1)q'+q)$ while that of being routed to y_1 is $((N/2-1)q'+q)/((N-1)q'+q)$. This is because requests for $N/2$ nonhot memories are routed through y_0, and requests for $(N/2-1)$ nonhot and one hot memory are routed through y_1. Therefore

$$\Pr[x_0 \rightarrow y_0] = (p_{-1,0})\left(\frac{(N/2)q'}{(N-1)q'+q}\right) \quad (4)$$

$$\Pr[x_0 \rightarrow y_1] = (p_{-1,0})\left(\frac{(N/2-1)q'+q}{(N-1)q'+q}\right) \quad (5)$$

The other probability terms in eqn. 3 can be derived similarly. Substituting them in eqn. 3 gives

$$\begin{aligned} p_{0,0} &= (p_{-1,0})\left(\frac{(N/2-1)q'+q}{(N-1)q'+q}\right) \\ &\quad \times (p_{-1,0})\left(\frac{(N/2-1)q'+q}{(N-1)q'+q}\right) \\ &\quad + (p_{-1,0})\left(\frac{(N/2-1)q'+q}{(N-1)q'+q}\right) \\ &\quad \times \left(1-(p_{-1,0})\left(\frac{(N/2-1)q'+q}{(N-1)q'+q}\right)\right) \\ &\quad + (p_{-1,0})\left(\frac{(N/2-1)q'+q}{(N-1)q'+q}\right) \\ &\quad \times \left(1-(p_{-1,0})\left(\frac{(N/2-1)q'+q}{(N-1)q'+q}\right)\right) \\ &= 2(p_{-1,0})\left(\frac{(N/2)q'+q-q'}{(N-1)q'+q}\right) \\ &\quad - (p_{-1,0})^2\left(\frac{(N/2)q'+q-q'}{(N-1)q'+q}\right)^2 \end{aligned} \quad (6)$$

By a similar reasoning the expression for $p_{0,1}$ can be derived as

$$\begin{aligned} p_{0,1} &= 2(p_{-1,0})\left[\frac{(N/2)q'}{(N-1)q'+q}\right] \\ &\quad - (p_{-1,0})^2\left[\frac{(N/2)q'}{(N-1)q'+q}\right]^2 \end{aligned} \quad (7)$$

The data rates at the outputs of S_1 will be calculated next. An output link of S_1 can access any of a set of $N/4$ memory modules. The set of $N/4$ MMs accessed by a hot output link includes the hot memory and other $N/4-1$ equiprobable memories, while $N/4$ equiprobable memory modules are accessed by the nonhot output link. In the case of a nonhot switch, each of the two sets of MMs accessed by the output links contain $N/4$ equiprobable memory modules. Following the same approach and reasoning as used for S_0, the three different data rates at the outputs of S_1 can be shown to be as follows:

$$\begin{aligned} p_{1,0} &= 2(p_{0,0})\left(\frac{(N/4)q'+q-q'}{(N/2-1)q'+q}\right) \\ &\quad - (p_{0,0})^2\left(\frac{(N/4)q'+q-q'}{(N/2-1)q'+q}\right)^2 \end{aligned} \quad (8)$$

$$\begin{aligned} p_{1,1} &= 2(p_{0,0})\left(\frac{(N/4)q'}{(N/2-1)q'+q}\right) \\ &\quad - (p_{0,0})^2\left(\frac{(N/4)q'}{(N/2-1)q'+q}\right)^2 \end{aligned} \quad (9)$$

$$\begin{aligned} p_{1,2} &= 2(p_{0,1})\left(\frac{(N/4)q'}{(N/2)q'}\right) - (p_{0,1})^2\left(\frac{(N/4)q'}{(N/2)q'}\right)^2 \\ &= p_{0,1} - (p_{0,1})^2\left(\frac{1}{2}\right)^2 \end{aligned} \quad (10)$$

Using eqns. 6–10, the general expressions for $p_{k,s}$ can be expressed as follows:

$$p_{k,s} = p_{k-1,s-1} - (p_{k-1,s-1})^2\left(\frac{1}{2}\right)^2 \quad \text{for } 1 \leq k \leq n-1,\ 2 \leq s \leq k+1 \quad (11)$$

$$\begin{aligned} p_{k,s} &= 2(p_{k-1,0})\left(\frac{(N/2^{k+1})q'+(q-q')(1-s)}{(N/2^k-1)q'+q}\right) \\ &\quad - (p_{k-1,0})^2\left(\frac{(N/2^{k+1})q'+(q-q')(1-j)}{(N/2^k-1)q'+q}\right)^2 \end{aligned} \quad \text{for } 0 \leq k \leq n-1,\ 0 \leq s \leq 1 \quad (12)$$

Eqns. 11 and 12 recursively define the data rates at all the links in the network. Note that eqns. 11 and 12 give the data rates at the output links of nonhot and hot switches, respectively.

To calculate the AMBW, the set of data rates $P_{n-1} = \{p_{n-1,0}, p_{n-1,1}, \cdots, p_{n-1,n}\}$ are calculated recursively using eqns. 11 and 12. The bandwidth is then given by eqn. 2 as

$$AMBW = p_{n-1,0} + \sum_{s=1}^{n} 2^{s-1} p_{n-1,s} \quad (13)$$

where, 2^{s-1} is the number of output links at S_{n-1} having data rate $p_{n-1,s}$.

The above model will be used to calculate the performance of an unbuffered network under hot spot traffic pattern in the next section.

5 Results

In this section, the performance figures of an unbuffered network operating under hot spot traffic pattern for different degrees of network traffic and hot spot probabilities are presented. This is followed by a comparison of performance between unbuffered and buffered networks.

5.1 Unbuffered network

Fig. 4 shows the AMBW plotted against the hot spot probability q for various values of processor request rates $(p_{-1,0})$ and a network size of 256. Because of increased contention at the hot switches with increasing q, the bandwidth decreases. As expected, the bandwidth increases with increased processor request rates. The decrease in bandwidth with increasing q is almost linear except for high request rates and hot spot probabilities when the decrease is almost exponential.

The bandwidth as a function of the hot spot probability for different network sizes and for a processor request rate of 0.8 is shown in Fig. 5. The degradation in performance with increasing hot spot probability is much more pronounced for a large network size.

Fig. 6 shows the bandwidth as a function of the network size for different q. For each curve, a constant ratio of q/q' has been maintained. If the ratio q/q' can be

maintained constant when the network size is increased, a linear increase in bandwidth can be expected. $q = q'$ corresponds to the URM case. It is observed that even

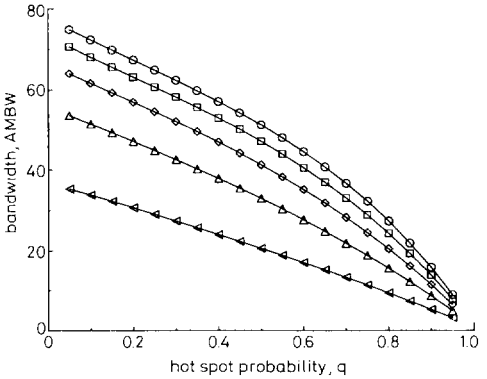

Fig. 4 *Bandwidth plotted against probability of hot spot for different request rates*

$N = 256$
○—○ $p_0 = 1$
□—□ $p_0 = 0.8$
◇—◇ $p_0 = 0.6$
△—△ $p_0 = 0.4$
◁—◁ $p_0 = 0.2$

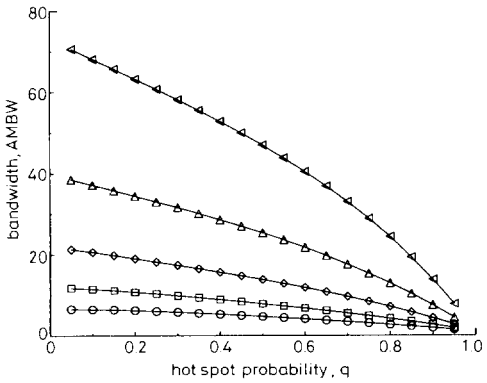

Fig. 5 *Bandwidth plotted against probability for hot spot for different network sizes*

$p = 0.8$
○—○ $N = 16$
□—□ $N = 32$
◇—◇ $N = 64$
△—△ $N = 128$
◁—◁ $N = 256$

with high hot spot probability ($q = 40q'$), the degradation as compared to the URM is not significant.

5.2 Comparison between unbuffered and buffered networks

In Fig. 7 we compare unbuffered and single-buffered Omega networks for uniform traffic and for different degrees of hot spot traffic. The buffers are assumed to be at the inputs of the switches, and a fair conflict resolution strategy at the switches has been used. Results for the buffered network have been obtained from an analytical model developed by the authors and described in Reference 19.

It is well established that, for uniform traffic the bandwidth of a buffered network is higher than that of an unbuffered network. This is also confirmed by the two uniform traffic curves in Fig. 7 where the buffered network has a higher bandwidth than the unbuffered one.

The above fact is not always true in the case of non-uniform traffic. As an example, for $q = 0.1$, the unbuffered

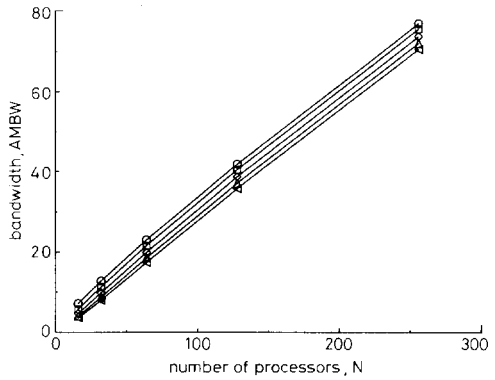

Fig. 6 *Bandwidth plotted against network size for different values of q*

○—○ $q = q'$
□—□ $q = 10q'$
◇—◇ $q = 20q'$
△—△ $q = 30q'$
◁—◁ $q = 40q'$

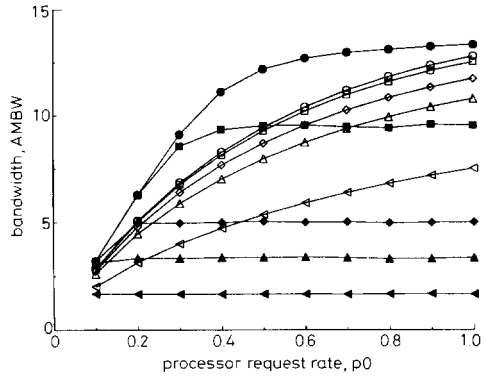

Fig. 7 *Bandwidth comparison between unbuffered and buffered networks*

○—○ $q = 0.03125$ (unbuffered, URM)
□—□ $q = 0.1$ (unbuffered)
◇—◇ $q = 0.2$ (unbuffered)
△—△ $q = 0.3$ (unbuffered)
◁—◁ $q = 0.6$ (unbuffered)
●—● $q = 0.03125$ (buffered, URM)
■—■ $q = 0.1$ (buffered)
◆—◆ $q = 0.2$ (buffered)
▲—▲ $q = 0.3$ (buffered)
◀—◀ $q = 0.6$ (buffered)

network performs better than the buffered network for processor request rates greater than 0.5. If the hot spot probability is increased to 0.2, the performance of the unbuffered network outperforms that of a buffered network for processor request rates greater than 0.2. For higher hot spot probabilities, the unbuffered network performs better than the buffered network for even lower processor request rates. This is due to tree saturation in buffered networks operating under hot spot traffic. As the hot spot probability increases, tree saturation occurs at lower processor request rates. If multiple buffers are used at the switches, it simply takes more time for the onset of tree saturation, but the performance is the same once tree

saturation occurs. We therefore conclude that an unbuffered network performs better than a buffered network whenever tree saturation occurs under a hot spot traffic pattern. This leads us to suggest a hybrid network which can switch between buffered and unbuffered modes. Under normal traffic conditions, the network will operate in a buffered mode, but will switch itself to the unbuffered mode whenever tree saturation is detected. Such a hybrid network will provide optimum performance under both uniform and nonuniform traffic patterns.

A simulator for an unbuffered network has been built to validate the results from the analytical model presented herein. The simulator is driven by a nonuniform random number generator. At the beginning of each cycle, a set of memory requests are generated from a random number generator. The distribution of the requests over the outputs follow the hot spot traffic pattern described in assumption (viii). The requests are forwarded through the stages and the number of requests reaching the output of the network at that cycle is determined. The above procedure is repeated over a large number of cycles. The average bandwidth is determined by averaging the number of requests coming out of the network. In case of a routing conflict between two requests at a switching element, a random number is used to resolve the conflict and route one of the requests.

Table 1 shows the percentage errors between analytical and simulation results for different network size and hot spot probabilities with a request rate of unity. The errors are small, thereby justifying the validity of the analytical model. The accuracy of the simulation depends on the number of memory cycles used. For our study, 12000 memory cycles have been used for $N = 2$ to 64. Due to the excessive simulation time required for larger networks, 7500 memory cycles have been used for $N = 128$ to 256.

Table 1: Percentage errors between simulation and analytical results

N	$q = 0.2$	$q = 0.4$	$q = 0.6$	$q = 0.8$
2	0.08205	−0.29845	0.10696	0.15780
4	0.13272	−0.49581	0.08955	0.01039
8	0.00049	0.03530	0.21179	−0.21902
16	0.28410	0.07466	0.33078	0.07624
32	−0.01225	−0.05007	−0.18703	−0.32316
64	−0.23501	0.04071	−0.28599	−0.13787
128	−0.20695	0.02320	0.23894	0.16657
256	−0.16347	0.01467	−0.17856	−0.12376

6 Conclusions

Analytical models offer a significantly faster method of performance evaluation than simulation methods. In this paper, an analytical model to study the bandwidth of unbuffered multistage interconnection networks under hot spot traffic has been described.

Bandwidths have been presented as functions of different network parameters, like the size of the network, the nonuniformity of requests, and traffic load on the network. It has been found that the degradation in the bandwidth with an increase in the hot spot traffic is more pronounced for larger networks than for smaller ones. Other measures of performance, for example, processor blocking probability can easily be determined from the bandwidth.

To validate the results from the analytical model simulation has been carried out and the results have been found to be in close agreement. The percentage errors between the analytical and simulation results for a wide range of network sizes and hot spot probabilities have been found to be less than 0.25% in most cases.

It has been verified that buffered networks perform better than unbuffered networks for the uniform traffic pattern. But, unbuffered networks perform better than buffered networks in the case of hot spot traffic. A hybrid network has been proposed to optimise the bandwidth of a multistage network under both uniform and nonuniform traffic conditions. The network will normally work in the buffered mode but will switch to an unbuffered mode whenever congestion in the network is detected.

The analysis in this paper has been carried out for the Delta network. The method used applies equally well for other Banyan-type networks, i.e. networks in which there is a unique path between the source and the destination.

7 References

1 FENG, T.Y.: 'A survey of interconnection networks', *Comput.*, 1981, **14,** pp. 12–27
2 MUDGE, T.N., HAYES, J.P., and WINSOR, D.C.: 'Multiple-bus architectures', *Comput.*, 1987, **20,** pp. 42–48
3 PATEL, J.H.: 'Performance of processor-memory interconnections for multiprocessors', *IEEE Trans.*, 1981, **C-30,** (10), pp. 771–780
4 LAWRIE, D.H.: 'Access and alignment of data in a array processor', *IEEE Trans.*, 1975, **C-24,** (12), pp. 1145–1155
5 STONE, H.S.: 'Parallel processing with the perfect shuffle', *IEEE Trans.*, 1971, **C-20,** (2), pp. 153–161
6 PEASE, M.C.: 'The indirect binary n-cube microprocessor array', *IEEE Trans.*, 1977, **C-26,** (5), pp. 458–473
7 GOKE, L.R.: 'Banyan networks for partitioning multiprocessor systems'. First annual symp. on computer architecture, 1973, pp. 21–28.
8 DIAS, D.M., and JUMP, J.R.: 'Analysis and simulation of buffered Delta networks', *IEEE Trans.*, 1981, **C-30,** (4), pp. 271–282.
9 THANAWASTIEN, S., and NELSON, V.P.: 'Interference analysis of shuffle/exchange networks', *IEEE Trans.*, 1981, **C-30,** (8), pp. 545–556
10 BHUYAN, L.N.: 'An analysis of processor-memory interconnection networks', *IEEE Trans.*, 1985, **C-34,** (3), pp. 279–283
11 DU, H.C.: 'On the performance of synchronous multiprocessor systems', *IEEE Trans.* 1985, **C-34,** (5), pp. 462–466
12 CHEN, W.T., and SHEU, J.P.: 'Performance analysis of multistage interconnection networks with hierarchical requesting model', *IEEE Trans.*, 1988, **C-37,** (11), pp. 1438–1442
13 KRUSKAL, C.P., and SNIR, M.: 'The performance of multistage interconnection networks for multiprocessors', *IEEE Trans.*, 1983, **C-32,** (12), pp. 1091–1098
14 PFISTER, G.F., and NORTON, V.A.: 'Hot spot contention and combining in multistage interconnection networks', *IEEE Trans.*, 1985, **C-34,** (10), pp. 943–948
15 LEE, G., KRUSKAL, C.P., and KUCK, D.J.: 'The effectiveness of combining in shared-memory parallel computers in the presence of hot spots'. 1986 Int. Conf. on Parallel Processing, 1986, pp. 35–41
16 SCOTT, S.L. and SOHI, G.S.: 'The use of feedback in multiprocessors and its applications to tree saturation control', *IEEE Trans.* 1990, **PDS-1,** (4), pp. 385–398
17 YEW, P.C., TZENG, N.F., and LAWRIE, D.H.: 'Distributing hot-spot addressing in large-scale multiprocessors', *IEEE Trans.*, 1987, **C-36,** (4), pp. 388–395
18 ATIQUZZAMAN, M., and AKHTAR, M.S.: 'Effect of hot spots on the performance of multistage interconnection networks'. FRONTIERS 92: Fourth Symp. Frontiers of Massively Parallel Computation, Virginia, October 1992, pp. 504–505
19 ATIQUZZAMAN, M., and AKHTAR, M.S.: 'Performance of buffered multistage interconnection networks in non uniform traffic environment'. 7th Int. Parallel Processing Symposium, California, April 1993, pp. 762–767
20 ATIQUZZAMAN, M., and AKHTAR, M.S.: 'Performance modeling of switching fabric for ATM switching node'. Australian Broadband Switching and Services Symp., Melbourne, 1992, pp. 581–588
21 BHUYAN, L.N., and AGRAWAL, D.P.: 'Design and performance of generalized interconnection networks', *IEEE Trans.*, 1983, **C-32,** (12), pp. 1081–1090.

22 CHANG, D.Y., KUCK, D.J., and LAWRIE, D.H.: 'On the effective bandwidth of parallel memories', *IEEE Trans.* 1977, **C-26,** pp. 480–489
23 BASKETT, F., and SMITH, A.J.: 'Interference in multiprocessor computer systems with interleaved memory', *Comm. ACM*, 1976, **19**, (6), pp. 327–334
24 BHANDARKAR, D.P.: 'Analysis of memory interference in multiprocessors', *IEEE Trans.* 1975, **C-24,** (9), pp. 897–908
25 VALERO, M., LLABERIA, J.M., LABARTA, J., SANVICENTE, E., and LANG, T.: 'A performance evaluation of the multiple bus network for multiprocessors'. Sigmetrics Conf. on Measurement and Modelling of Computer Systems, August 1983, pp. 200–206
26 RAU, B.R.: 'Interleaved memory bandwidth in a model of a multiprocessor computer system'. *IEEE Trans.* 1979, **C-28,** (9), pp. 678–681