A New model for Evaluating Performance of Processor Memory Interconnections

Subramanyam.A[1] Prasad. E.V[2] Srinivasulu.K[3] Harikrishna.T[4]

ABSTRACT

Interconnection networks play a principal role in today's parallel computers. In the existing model, performance evaluation of multistage interconnection networks is based on a uniform reference model and the assumption of independent requests. Existing model and analytical results have been widely adopted by numerous researchers as a basis to investigate various aspects of MIN's. In this paper we study in detail the effects of the independence assumption on the accuracy of system performance and point out the factor which cause inaccuracy. A new queuing model is then proposed and is shown to be very accurate. Since only six states are needed, independent of the size of MIN's, this new model is very efficient computationally.

Keywords: interconnection networks, multistage interconnection networks, performance evaluation, queuing theory, simulation.

1. INTRODUCTION

Interconnection Networks provide communication between different processing elements and system's memory modules. Multistage Interconnection Networks (MIN's) have been used in multiprocessor systems for supporting processor-memory and interprocessor communications as well as in communication systems as basic switching devices. The advantage of using MIN's includes the low hardware complexity compared to that of crossbars, efficient distributed control schemes, partitionability, availability of multiple simultaneous paths, and ability to employ a variety of implementation techniques [19]. Depending on the clocking structure, switching mode and control strategy, MIN's can have different operational characteristics giving rise to different system behaviors [11].

In this paper single-path MIN's are only considered. In these MIN's internal blocking at any switching element (SE) may occur, i.e., multiple requests from the inputs may need to utilize the same output ports of some SE's or communication links and only one of them can occupy an output port or a link at a time such that the rest of the requests will be blocked and need to be reissued in the next cycle. In the existing model([1],[2],[4],[5]) a probabilistic approach is proposed to analyze the performance of MIN's based on a uniform reference model and the assumption of independent requests, i.e., a blocked request is discarded and a new request, which is independent of the blocked one, will be generated in the next cycle to replace the previous blocked yet unserved request. Thus this independent request model can be called as Drop Model (DM) as the request of previous cycle is not considered for the same port in the current

---

[1&4]CSIT dept. Annamacharya Institute of Technology & Sciences,Rajampet, Kadapa. Email : smarige@rediffmail.com, thk.itd@gmail.com

[2]Principal, J.N.T.U.College of Engineering, Kakinada, E-mail : drevprasad@yahoo.co.in

[3]CSE dept.,Madina Engineering College, Kadapa. E-mail : sreenu.kutala@gmail.com

cycle [6]. In a MIN, all requests from input ports must be issued at the beginning of a network cycle where a network cycle is defined as the time for a request to traverse the network and the time for data transmission. It is clear that this independent request assumption is not realistic because, in a real system, a blocked request will go to the same desired memory module in the next cycle. However, in the existing model ([1]-[4],[7],and [12]) compared to the results of simulation, the results are only slightly different if this independent request assumption is omitted. Thus, it is generally believed that the drop model (DM) not only simplifies the analysis but also accurately predicts the performance of a synchronous MIN.

Through extensive simulations it is found that the DM is indeed not accurate[6] enough in many cases, but it has been widely adopted by many people without further justifications ([9],[10],[14]-[19],[21]) which motivates the research work of this paper. It has also shown by Lee [6] that there is an as much as 30% difference between results using the existing model and those using dependent request models which are referred in this paper as Hold Model (HM) as the request of the previous cycle is also considered for current cycle to the same port. Of course, the HM is a much more realistic model. Yen, Patel, and Davidson [21] investigated, on crossbars, the accuracy of the DM (referred to as Model 1 in their paper) as well as two other proposed models (namely, Models 2-3) by comparing them with the simulation results of the HM. The other two models are briefly summarized as follows: Model 2 is a modified version of Model 1 by assuming that all the blocked requests will always be resubmitted as new independent requests, which is referred in this paper as modified drop model (MDM). By performing comparisons it is derived Model 3 is the most accurate model for the analysis of crossbars. However, since Model 3 is non intuitive ([20],[21]) it is very difficult to extend this model to analyze cascaded crossbars such as MIN's. MDM is slightly better than the DM in terms of maximum difference. Also, extending MDM to analyze a MIN will make the analysis more complicated. Thus, the DM is still widely used for the analysis of MIN's, when crossbars are cascaded into a MIN.

In this paper we first investigate the existing model (DM), through extensive simulations, point out that the existing model (DM) is not accurate enough. After that, a new queuing model is proposed, which takes into account that a blocked request will be reissued and will still go to a previously determined destination, and shows a very high degree of accuracy. Moreover, since only six states are needed, independent of size of a MIN, this new model is very efficient computationally. Since the accuracy of MDM in [22] has only been analyzed on crossbars, in this paper we also developed a model by adopting the concept of MDM to the analysis of MIN's and investigate its accuracy.

The remainder of the paper is organized as follows. In Section II, a brief review and investigation of the existing model is presented. In Section III we present a new model for analysis. The discussion of the performance of the system is presented in Section IV. In section V evaluation of the proposed model is given. Section VI gives the conclusions of this paper.

2. A Review Of Drop Model

In order to understand the accuracy of the existing model (DM), a brief review of the existing model (DM) and the analytical results ([2],[4],[7]and 19) are presented here. The accuracy of its results is investigated in detail in the next section.

The drop model (DM) is based on the following assumptions 1)A (P *x* M) MIN consists of n stages of switching elements (SE's) where P is the total number of inputs, M is the total number of outputs, and each SE is an (a *x* b) crossbar. For example, a (2*x*2) crossbar is shown in 1.

2) During each cycle a request will be generated at each input port with probability r_0, i.e. the request rate.

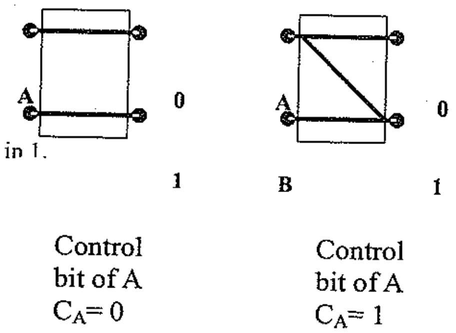

Figure1(A) : 2x2 Crossbar Switch

(a) Which is useful for construction of a (2^3 *x* 2^3) MIN as shown in Fig. 1(b).

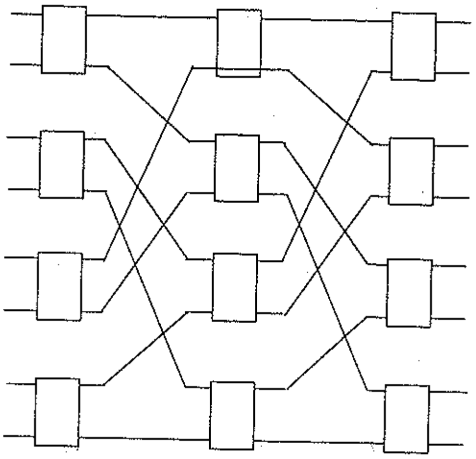

Figure1(B) : 3-Stage MIN With 2x2 Crossbar Switch

3) All the requests are uniformly distributed over all output ports.

4) The requests which are blocked (not accepted) are ignored and the requests issued in the next cycle are independent of the requests previously blocked.

The fourth assumption is used to simplify the analysis and of course, is not realistic. In a real system, a rejected request must be reissued in the next cycle and its destination must be the same as the blocked one.

Based on the assumptions stated above, in the existing model an analysis on crossbars of size a x b is performed. Here a is input size, b is the output size. For an arbitrary output, the probability that there is no request to the output is $(1-r_0/b)^a$. Thus, the probability that there is a request on an arbitrary output of the crossbar is

$$r_{out}=1-(1-r_0/b)^a \qquad (1)$$

The analytical result of a crossbar is then used as the basis for analyzing a MIN([13],[16]). Since the output rate of a stage is the input rate of the next stage, in the existing model an analysis on the synchronous MIN it can be shown that the probability that there is a request on any particular output at the (i+1)th stage of the network satisfies the recurrence relation

$$r_{i+1}=1-(1-r_i/b)^a (i=0,1.........n-1) \qquad (2)$$

Where r_0 is the request rate, and M x r_n is the throughput, which is defined to be the average number of unit messages delivered by the MIN in a unit time.

2.1. Investigations of Drop Model

Now we investigate the accuracy of the DM and point out the factors that cause the inaccuracy. The new models that can avoid those problems will be presented in the next section.

To verify that DM is not accurate enough in many cases and the independence assumption is indeed the major reason for the inaccuracy, we implement two versions of simulation for DM and HM. When a request is blocked, it will stop traveling through the network. For the DM the source node with a blocked request will generate a new request with probability r_0 and will generate another destination. However, for the HM when a request is blocked, the source node will reissue the same request in the next cycle to the same destination. The results of the DM are always too conservative for a small value of r_0 and too optimistic for a large value of r_0. This observation can be understood through the following interpretations. In the first case when the request rate r_0 is small, and once a request is blocked, it should be reissued in the next cycle. Thus, the probability of issuing a request in the next cycle should be 1. However, the probability of issuing a request under the existing model (i.e., the DM) will still be r_0. Thus, the existing model will have much lower throughput than the realistic model. The inaccuracy will depend on the difference between the value of r_0 and the probability for a request to be blocked. A decrease of b/a will increase the difference between the DM and the HM when r_0 is small. An increase of n will increase the probability for a request to be blocked through the network.

In the second case when the request rate r_0 is high, there will be a considerable probability that many requests need the same path and only one of them will be chosen and all others will be blocked. In a real system once those requests are blocked, they are likely to use the same path in the following cycle. However, the DM assumes that all requests are uniformly distributed over all paths in each cycle; those requests that needed the same path and were blocked in the previous cycle will again be assumed to be uniformly distributed over all paths. When request rate r_0 is high, there will be a large number of requests that need the same path in a cycle. The difference between the DM and the realistic HM is not negligible. A realization of this optimistic condition is shown in Fig.2.Box which is shown in fig.2(b) of existing model in the current cycle is just one of possible outcomes from box which is shown fig2.(c) is of proposed hold model(HM).

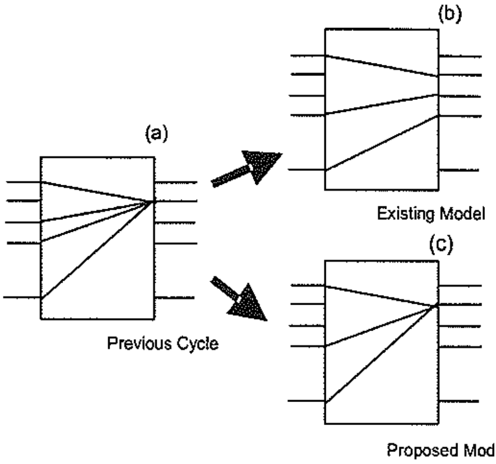

Figure 2 : Realization Of The Optimistic Condition

3. NEW QUEUING MODEL

We propose a new queuing model for analyzing crossbars[8] which utilizes the knowledge of the simulation results to adjust the dominant factors that cause the inaccuracy. From the discussion in the previous section it is clear that those factors which cause the inaccuracy in both the conservative and optimistic conditions of the DM need to be adjusted. Since the requests from all the input ports to all the output ports are assumed to be stochastically identical, we can then perform the analysis on the behavior of an arbitrary path. First, we perform analysis on a single crossbar. After that, we generalize it to the analysis of a MIN. The new queuing model for analyzing crossbars based on the understanding of the previous observation is then presented in Fig. 3. In the figure,

state $\mathbf{I}_0$ represents that, at the beginning of a network cycle, the source node is not issuing a request; state $\mathbf{R}_0$ represents that the source node is issuing a request which can be a new request or the same request that was

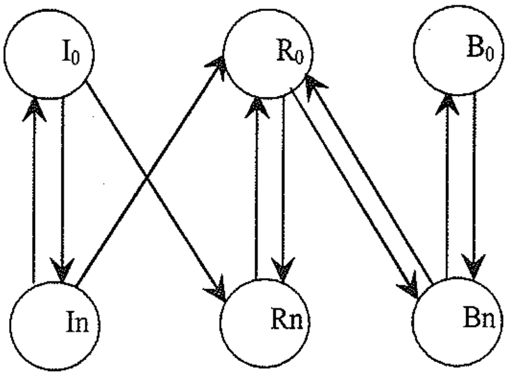

Figure3 : State Transition Diagram Of The Proposed Model

blocked in the previous cycle; and state **B**$_o$ represents that the source node is not joining the competition. Also, state **I**$_n$ represents that, at the end of a network cycle, the source node not issuing a request is staying idle; state **R**$_n$ represents that the requested path has been successfully established; and state **B**$_n$ represents that the request is blocked. Note that state B$_o$ is used to reduce the inaccuracy introduced in the optimistic condition. For example, in Fig. 2, all three requests will go to the same destination in the current cycle and two of them will be blocked again. Thus, when they are blocked, only one of the three requests will go to state R$_o$. and the rest of them will go to state B$_o$.

Utilizing the new queuing model, the problems which occur in the conservative condition of the DM can be improved by introducing a transition from state B$_n$ to state R$_o$ which ensures that the blocked request will be reissued in the next cycle. Also, the transition from state B$_n$ to state B$_o$ is used to adjust the overestimated input rate in the optimistic condition of the DM. The state transition probabilities can be derived through the following understanding:

1) Pr[R$_n$|Ro]: which is the probability that, at the end the probability that there will be of a network cycle, a request is in State R$_n$ giving that the request was in State R$_o$ at the beginning of that network cycle. Let m$_{out}$ be the probability that there is a request on any particular output of the crossbar. Since we assume that the requests from different source nodes are stochastically independent, m$_{out}$ can be obtained by utilizing the formula in [18] as follows:

$$m_{out} = 1 - \left(1 - \frac{P_r[R_0]}{b}\right)^a \qquad (3)$$

Since on the average there will be (a *x* Pr[R$_o$]) requests at the beginning of a cycle and only (b *x* m$_{out)}$ requests can reach the output of the crossbar, it is clear that

$$P_r[R_n \mid R_0] = \frac{m_{out} x b}{a x P_r[R_0]} \qquad (4)$$

2) Pr[Ro|B$_n$]: This transition is based on a concept, which ensures that the blocked request will be reissued in the next cycle. By setting Pr[R$_0$|B$_n$] = 1, though can effectively adjust the conservative condition, will result in an even more significant optimistic condition. We combine the effect of the conservative as well as the optimistic conditions by introducing the transition from state B$_n$ to states R$_0$ and B$_o$. The derivation of the transition probabilities will be shown later.

3) Pr[B$_0$|B$_n$]*:* This transition probability is used to reduce the inaccuracy introduced in the optimistic condition of the DM, i.e., some of the blocked requests that needed the same path in the previous cycle will need the same path again. Only one of those requests will successfully go through the network and the rest of them will be blocked again. Since the DM assumes that all requests are uniformly distributed over all paths in any cycle, those requests that needed the same path in the previous cycle will be assumed to be uniformly distributed over all paths. A straight forward approach for solving this problem is to build a queuing model where each state represents the number of requests that need the same path. However, this approach will require a large number of states and it

is very difficult to derive the state transition probabilities between states. Thus, we solve this problem by utilizing the knowledge that not all the blocked requests will go to the state of reissuing a request; instead, some of them will go to the state which represents that it will be blocked again. Consider an arbitrary blocked request, i requests from the other (a–1) source nodes which need the same path is

$$\binom{a-1}{i}\left(\frac{P_r[R_0]}{b}\right)^i\left(1-\frac{P_r[R_0]}{b}\right)^{a-1-i}$$

Since this request is blocked and there are i requests from other (a–1) source nodes in this cycle, one of the i requests will successfully establish the desired path and the other (i–1) requests will need the same path in the next cycle. In this new model, we allow only one of the i blocked requests to go to state R_o and the rest of them to state B_0 at the beginning of the next cycle. Since all the requests are randomly selected, the probability that this request will not be blocked by the other i-1 requests and can go to state R_0 is 1/i. Then the conditional probability is derived as follows:

$Pr[R_0|B_n]=$

$$\frac{\sum_{i=1}^{a-1}\left(\frac{a-1}{i}\right)\left(\frac{P_r[R_0]}{b}\right)^i\left(1-\frac{P_r[R_0]}{b}\right)^{a-1-i}\frac{1}{i}}{1-\left(1-\frac{P_r[R_0]}{b}\right)^{a-1}} \qquad (5)$$

and $Pr[R_0|B_n]=$

$$\frac{\sum_{i=1}^{a-1}\left(\frac{a-1}{i}\right)\left(\frac{P_r[R_0]}{b}\right)^i\left(1-\frac{P_r[R_0]}{b}\right)^{a-1-i}\frac{i-1}{i}}{1-\left(1-\frac{P_r[R_0]}{b}\right)^{a-1}} \qquad (6)$$

The rest of the state transition probabilities can be easily obtained and are given as follows

$$Pr[B_n|R_0]=1-\frac{m_{out}xb}{axPr[R_0]}$$

$$Pr[B_n|R_0]=Pr[I_n|I_o]=1$$

$$Pr[R_0|R_n]=Pr[R_0|I_n]=r_0$$

$$Pr[I_0|R_n]=Pr[I_0|I_n]=1-r_0 \qquad (7)$$

3.1. Proposed Model for the MIN's

The state transition diagram for a MIN can be obtained similarly by using six states at each stage and using the output rate of a stage as the input rate of the next stage. However, this approach will significantly increase the cost of analysis. Thus, for a MIN, we propose a model which uses the same number of states as that for a single crossbar independent of the system size. In the new model the meanings of all the states remain the same as they are in a crossbar but the state transition probabilities have to be modified. Let r_k be the probability that there is a request on any particular output at the kth stage. We can then have the following recurrence relation:

$$m[k]=1-\left(1-\frac{m[k-1]}{b}\right)^a \qquad (8)$$

where m[0] is the adjusted initial request rate, i.e., $m[0]=Pr[R_0]$, also

$$Pr[R_n|Ro]=\frac{m[n]xM}{NxP_r[R_0]} \qquad (9)$$

We also propose to keep the state transition probabilities from B_n to B_0 and from B_0 to R_0 to reduce the analysis cost. Those conditional probabilities, of course, are only an approximation of the model. A further partitioning of the states will produce a more accurate result. However, through extensive simulation verification, this model has been found to be accurate enough for the analysis of the system.

Deriving the state flow balance equations, we can then summarize the system of equations as follows:

$$Pr[Bo]=Pr[B_n]\,x\,Pr[Bo|B_n]$$

$$Pr[B_n]=Pr[Bo]+Pr[R_o]\,x\,Pr[B_n|R_o]$$

$$Pr[R_0]=Pr[B_n]\,x\,Pr[R_0|B_n]+(Pr[R_n]+Pr[I_n]\,x\,r_0$$

$$\Pr[R_n] = \Pr[Ro] + \Pr[R_n \mid R_o]$$
$$\Pr[I_o] = (\Pr[R_n] + \Pr[I_n])(1 - r_o)$$
$$\Pr[I_n] = \Pr[I_o] \tag{10}$$

where

$$\Pr[B_n \mid B_o] = \Pr[I_n \mid I_o] = 1$$
$$\Pr[R_0 \mid R_n] = \Pr[R_0 \mid R_n] = r_0$$
$$\Pr[I_0 \mid R_n] = \Pr[I_0 \mid I_n] = 1 - r_0$$
$$\Pr[B_n \mid R_o] = \frac{m[n]xM}{NxP_r[R_0]}$$
$$\Pr[B_n \mid R_o] = 1 - \frac{m[n]xM}{NxP_r[R_0]}$$
$$\Pr[B_0 \mid B_n = 1 - \frac{\sum_{i=1}^{N-1} \left(\frac{N-1}{i}\right)\left(\frac{P_r[R_0]}{M}\right)^i \left(1 - \frac{P_r[R_0]}{M}\right)^{N-1-i} \frac{1}{i}}{1 - \left(1 - \frac{P_r[R_0]}{M}\right)^{N-1}}$$
$$\Pr[R_0 \mid B_n] = \frac{\sum_{i=1}^{N-1} \left(\frac{N-1}{i}\right)\left(\frac{P_r[R_0]}{M}\right)^i \left(1 - \frac{P_r[R_0]}{M}\right)^{N-1-i} \frac{1}{i}}{1 - \left(1 - \frac{P_r[R_0]}{M}\right)^{N-1}} \tag{11}$$

with two boundary conditions

$$\Pr[I_i] + \Pr[B_i] + \Pr[R_i] = 1, \text{for } i = 0, n. \tag{12}$$

These equations are useful in solving the balancing equations for crossbars and MIN's numerically, so that we can derive performance measures from them.

4. Analysis Of Hold Model

The performance of MIN's depends greatly on many different system parameters like the number of stages in the MIN and the size of SE's, and workload parameters like request rate etc. The optimum design of a system with respect to various parameters is discussed in this section. There is a tradeoff between cost and performance of a system. Depending on applications, it is desirable to ensure that some measures will always meet certain performance criteria. Also, in [15] and [17], the authors pointed out that the bandwidth of a squared MIN is asymptotic independent of request rate. Thus, in this paper, all the performance measures will be taken into account for analysis. The designer, with all the information, can then design a system that best meets some desired performance criteria and is also cost-effective.

The performance measures used in the rest of this paper are described as follows:

1) Throughput (thr) is defined as the average number of unit messages delivered by the MIN in a unit time. It can be shown that throughput

$$\text{Throughput} = \Pr[R_n] x N. \tag{13}$$

Normalized throughput (nor_thr) is defined as throughput of each input/output port. Thus for a non-squared MIN, there are input normalized throughput $nor\_thr_{in} = \Pr[R_n]$ and output normalized throughput $nor\_thr_{out} = \Pr[R_n] x N/M$.

2) Transmission delay (T_d) is defined as the mean number of network cycles required to transfer a message. Since the probability that a request is accepted given that it is issued is $\Pr[R_n]/r_o$ clear that

$$\text{Transmission delay} T_d = r_o / \Pr[R_n]. \tag{14}$$

A system with larger number of stages may have a longer network cycle which is not necessarily proportional to number of stages of a MIN because a cycle contains not only the time to setup a path for a request but also the time for data transmission.

The probability that a request is accepted will be independent of the value of b/a of an SE. When r_o is small, a system with a large number of input ports and a smaller number of output ports will be more cost-effective, which will have similar nor_thr_{in} but much higher $nor\text{-}thr_{out}$ compared to a system with a large number of output ports. When r_0 is high, the performance of a MIN will become complicated. A system with a large number of output ports will have much higher nor_thr_{in} yet much lower nor_thr_{Out}. As r_o increases, T_d will increase significantly. It should be noted that when r_o becomes large, since throughput will become asymptotically independent of r_o, transmission delay will increase linearly with the rate of r_o.

5. EVALUTION USING PROPOSED MODEL

The performance measures can be evaluated using equations (15), (16) which were derived in the previous section. The following program is useful for calculating the transmission delay and throughput etc. performance measures.

Program for Calculation of Normalized throughput and transmission delay of Multistage Interconnection Networks.

```
#include<stdio.h>
#include<math.h>
void main()
{float td,r,tp,rin;
int b,a,n;
clrscr();
printf(“\nEnter memory,nthstage, nthstagerequest,
processor, reqrate values”);
scanf(“%d%d%f%d%f’,&b,&n,&rin,&a,&r);
rin=1-(pow((1-rin/b),a));
td=r/rin;
printf(“\nNormalized throughput is %f”, rin);
printf(“\nTransmission delay is %f” td);
getch();}
```

Table1 :Throughput of 2x2 MIN

p r	1.0	0.8	0.6	0.4	0.2
4	1.6410	1.4881	1.3484	1.2210	1.1051
8	1.9360	1.7191	1.5171	1.3299	1.1569
16	2.2230	1.9455	1.6836	1.4381	1.2098
32	2.5050	2.1684	1.8483	1.5455	1.2619
64	2.7824	2.3887	2.0121	1.6622	1.3139
128	3.0572	2.6070	2.1764	1.7589	1.3660
256	3.3295	2.8237	2.3347	1.8648	1.4179
512	3.5997	3.0388	2.4950	1.9705	1.4697
1024	3.8685	3.2530	2.6546	2.0759	1.5215
2048	4.1356	3.4660	2.8136	2.1810	1.5732
4096	4.4017	3.6783	2.9719	2.2858	1.6248

Table2 : Transmission delay of 2x2 MIN

r p	1.0	0.8	0.6	0.4	0.2
4	0.6094	0.5376	0.4450	0.3276	0.1810
8	0.5165	0.4654	0.3955	0.3008	0.1729
16	0.4498	0.4112	0.3563	0.2782	0.1653
32	0.3992	0.3689	0.3246	0.2588	0.1585
64	0.3594	0.3349	0.2982	0.2407	0.1522
128	0.3271	0.3069	0.2760	0.2274	0.1464
256	0.3004	0.2833	0.2570	0.2145	0.1411
512	0.2778	0.2632	0.2405	0.2030	0.1361
1024	0.2585	0.2459	0.2260	0.1927	0.1315
2048	0.2418	0.2308	0.2133	0.1834	0.1271
4096	0.2272	0.2175	0.2019	0.1750	0.1231

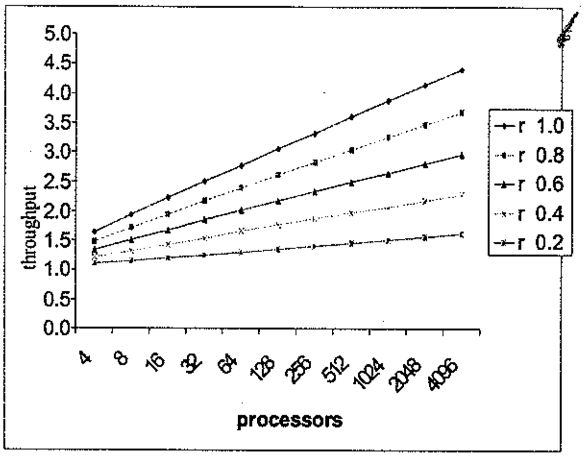

Figure 4 : Throughput Vs Processors

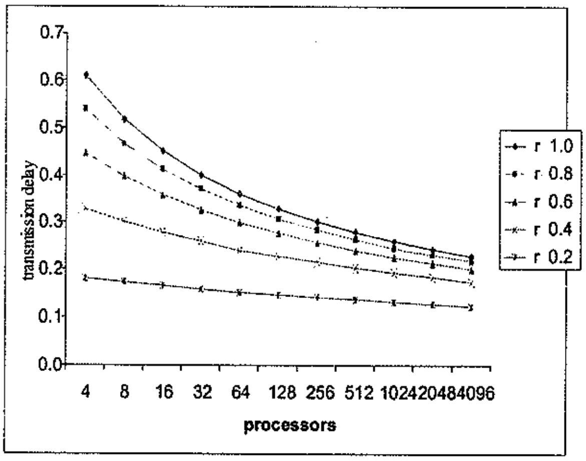

Figure 5 : Transmission Delay Vs Processors

The performance of MIN's with the various input and output ports are studied. The performance of MIN's in terms of nor_thr and T_d with respect to when $r_o = 1$ to varying upto 0.2, p indicates the number processors are presented in Table.1 and Table.2 . The graphs are plotted Fig.4 and Fig.5, since the value of N changes exponentially with respect to n, a logarithmic scale is more appropriate to represent the switch size. As mentioned earlier, the performance of a MIN will be independent of the sizes of SE's when r_o is small. The larger the value r_o the more significantly a MIN with larger SE's will improve the performance.

6. CONCLUSIONS

WE HAVE SHOWN THAT THE MODEl for the performance evaluation of synchronous multistage interconnection networks in existing model is not accurate enough in many cases, by carefully pointing out the causes for inaccuracy. A new queuing model is then proposed which has been shown to be very accurate in performance analysis and very efficient in computation. The performance of MIN's has been discussed in detail with respect to various system parameters, such as the number of stages in the MIN, the sizes of switching elements in the MIN and the request rate.

Further research can be done in extra stage MIN's considering the hold model using on queuing model and transitions for evaluating the performance measures.

REFERENCES

[1] A.Subramanyam, E.V.Prasad, C. Nadamuni Reddy, *" E-MIN: An efficient cost effective fault tolerant network for multiprocessor systems"*, Journal of Computer Science in March-April 2007, Vol.2, No. 5, PP. 378-388.

[2] A.Subramanyam, E.V.Prasad, C. Nadamuni Reddy, *"Performance evaluation of single stage and multistage interconnection networks connectivity in multiprocessor"* , International Journal ANIJMIT - 2008, Vol.4, PP. 7-21, 2007.

[3] A.Subrmanyam, E.V.Prasad, C. Nadamuni Reddy, *"Permutation capability and connectivity of enhanced multistage interconnection network (E-MIN)"*, IEEE proceedings of XIV international conference ADCOM-2006, PP. 8-12.

[4] A.Subramanyam,Dr.E.V.Prasad, Dr.C.Nadamuni Reddy, *"Performance of Processor Memory interconnection Networks proceedings of "NCMCM – 2003"*, Proceedings PP. 389 – 394.

[5] J. H. Patel, *"Performance of processor-memory interconnections for multiprocessors"*, IEEE Trans. Comp., Vol. C-30,PP. 771-780, Oct. 1981.

[6] M. Lee and C.L. Wu, *"Performance analysis of circuit switching baseline interconnection network"*, in Proceedings of Symposium Computer Architecture, PP. 82–90, 1984.

[7] F.Baskett and A.J.Smith, *"Interference in multiprocessor computer systems with interleaved memory"*, Communications ACM, Vol. 19, PP. 327-334, June 1976.

[8] C.Y.Roger, *"A New Model for the Performance Evaluation of Multistage Interconnection Networks"*, IEEE Transactions on Networking, Vol. 3 , No.6 PP. 708-716, December. 1995

[9] L. N. Bhuyan, *"An analysis of processor-memory interconnection networks"*, IEEE Transactions on Computers, Vol. C-34, No. 3, PP. 279–283, Mar. 1985.

[10] *"Design and performance of generalized interconnection networks"*, IEEE Transactions on Computers, Vol. C-32. PP. 1081-1090, Dec. 1983.

[11] L. N. Bhuyan, Q. Yang. and D. P. Agrawal, *"Performance of multiprocessor Interconnection networks"*, IEEE Computer Vol. 22, No. 2, PP. 25-37, Feb. 1989.

[12] D. Y. Chang, D. J. Kuck and D. H. Lawrie, *"On the effective bandwidth of parallel memories"*, IEEE Transactions on Computers, Vol. C-26, PP. 480-490, May 1977.

[13] S. Cheemalavagu and M. Malek, *"Analysis and simulation of banyan interconnection networks with 2 x 2, 4 x 4, and 8 x 8 switching elements"*, in Real time Systems Symp.,PP. 83–89, 1982.

[14] W.T. Chen and J.P. Sbeu, *"Performance analysis of multistage interconnection networks with hierarchical requesting model"*, IEEE Trans. Computer, Vol. C-37. No. 11, PP. 1438-1442. Nov. 1988.

[15] V. S. Cherkassky, *"Performance evaluation of non rectangular multistage interconnection networks"*, IEEE Trans. Computer, Vol. C-37, No. 10, PP. 1269-1272, Oct. 1988.

[16] D. M, Dias and J. R. Jump, *"Analysis and simulation of buffered delta networks"*, IEEE Transactions on Computer, Vol. C-30. No. 10, PP. 273–282, Oct. 1981.

[17] C. P. Kruskal and M. Snir, *"The performance of multistage interconnection networks for multiprocessors"*, IEEE Tram Comput., Vol. C-32, PP. 1091-1098. Dec. 1983.

[18] M. Kumar and J. R. Jump, *"Performance of unbuffered shuffle-exchange net works"*, IEEE Transactions on Computer, Vol. C-35, PP. 573–578, June 1986.

[19] T. N. Mudge and B. A. Makrucki, *"Probabilistic analysis of a crossbar switch"*, in Proceedings of Symposium on Computer Architecture, PP. 311-320, 1982.

[20] H. J. Siegel, W. G. Nation, C. P. Kruskal and L. M. Napolitano, Jr, *"Using the multistage cube network topology in parallel supercomputers"*, Proceedings IEEE, Vol. 77, No. 12, PP. 1932-1953, Dec. 1989.

[21] K. O. Siomalas and B. A. Bowen, *"Performance of crossbar multiprocessor systems"*, IEEE Transactions on Computer, Vol. C-32,PP. 689-695, July 1983.

[22] D. W. Yen, J. H. Patel and E. S. Davidson, *"Memory interference in synchronous multiprocessor systems"*, IEEE Transactions Computer, Vol. C-31, PP. 1116-1121, November 1982.

Author's Biography

Dr.A.Subramanyam received his Ph.D. degree in Computer Science and Engineering from JNTU College of Engineering, Anantapur. He has obtained his B.E.(ECE) from University of Madras and M.Tech.(CSE) from Visweswaraiah Technological University. He is having 17 years experience in teaching. He is currently working as Professor & HOD in the Department of Computer Science and Information Technology of Annamacharya Institute of Technology & Sciences, Rajampet, Kadapa dt. A.P. He has presented and published 7 technical papers in international and national conferences and 2 technical papers in international and national journals. He is guiding few Ph.D.s. His research areas of interest are parallel processing, network security and data warehousing.

Dr. E.V.Prasad, received his Ph.D. degree in Computer Science and Engineering from I.I.T., Roorke . He is having 29 years of experience in teaching. He joined in the JNTU university in the year 1979 and served in various positions. Now he is Professor in computer science and engineering department and also principal of JNTU college of engineering, Kakinada. He has published 41 technical papers in international and national journals and conferences. He has already guided few Ph.D.,'s and guiding several Ph.D.s. His research areas of interest including parallel processing, image processing ,computer networks, network security and data ware housing.

T.Harikrishna working as Assistant Professor in Computer science & Information Technology department in Annamacharya Institute of Technology & Sciences, Rajampet, Kadapa dt. A.P. He is having 5 years of experience in teaching. He completed his B.Tech. in Information Technology from JNTU. He is also having M Tech degree in Information Technology.

K.Sreenivasulu presently working as Associate professor & HOD CSE in Madina Engineering College Kadapa. He is currently pursuing Ph.D from JNTU He received B.E in Computer science from Bangalore University. He received his M.Tech in computer science from JNT University. He is having more than 11 years of experiance in teaching. He has guided several graduate & post graduate students in their Academic projects.