

## PERFORMANCE IMPROVEMENT OF MULTISTAGE INTERCONNECTION NETWORKS

SUBAŞI, Abdülhamit  
Sakarya University

Electrical and Electronic Engineering Department  
E-mail: asubasi@usa.net

Etem Köklükaya  
Sakarya University

Electrical and Electronic Engineering Department  
E-mail: Ekaya@esentepe.sau.edu.tr

**Abstract**-Multistage Interconnection networks (MIN's) have a number of applications in the areas of computer and communication. The presented work is a study of performance improvement of MINs. MINs are used for interconnecting processors to memory modules in multiprocessor systems. Buffered MINs have buffers to store the incoming packets. In this work, a new model for the multibuffered networks, which is called as dynamic buffered networks, are proposed and examined; the buffered networks were proposed before is called as static buffered networks. The performance of static and dynamic buffered networks are also compared by considering throughputs and delays. It is demonstrated that dynamic buffered networks improves the network performance by a considerable amount.

**Index Terms:** Dynamic buffered networks, static buffered networks, multistage interconnection networks, performance analysis.

## 1. INTRODUCTION

(MINs) have been widely used as efficient interconnection structures for parallel computer systems and the switching nodes for high-speed communication networks. MINs are also useful for broadband integrated services digital network (B-ISDN) and transport systems based on the asynchronous transfer mode (ATM). In the ATM, informations are encapsulated in fixed length packets. To accommodate services requiring high bandwidth such as video communication or high speed data communication, these packets must be switched at high speed. Banyan network has been proposed as a candidate for this high speed switching system [9].

It has been recognized that buffers in MINs can increase the MIN's performance significantly especially when the traffic is nonuniform. They also prevent a packet from being lost when a path conflict occurs. If buffers are put in each switch node and the back pressure mechanism is employed, then a packet can leave its buffer only when the destination buffer at the succeeding stage is able to accept it. For various implementations of MINs, buffered or unbuffered, analytical performance evaluation is crucial for justifying the merit of the design in different operational conditions. It is also important that the performance model needs to be simple and easily generalized [9],[10],[11].

## 2. PACKET-SWITCHING MINs

A packet switching MIN is consisted of simply, several  $a$  input- $b$  output ( $a \times b$ ) switches. A packet at any one of the  $a$  input terminals can be passed by a switch to any one of the  $b$  output terminals. Therefore, when a packet has arrived at an input terminal, by using information in the packet, the switch selects one of its output terminals. Then it transfers the packet to that output terminal [1].

A MIN is formed by arranging switches into stages with data paths, which are called links. These links connect an output terminal of a switch in one stage to an input terminal of the switch in the next stage. Input terminals of the first stage switches are called input ports, and output terminals of the last stage switches are called output ports.

Packet-Switching MINs can be categorized as unbuffered MINs, and buffered MINs. Although buffered MINs have buffers at each switch input, unbuffered MINs have no such buffers. In unbuffered MINs, when more than one packet in the same switch tend to go through the same output terminal of the switch, a conflict occurs there. Hence, usually a randomly chosen packet is forwarded, the remaining packets are rejected. But, in buffered MINs, when a conflict occurs, randomly chosen one is forwarded, remaining packets are stored in the buffers.

Delta networks were introduced by Patel [1]. A delta network is an  $a' \times b'$  switching network with  $n$  stages consisting of  $a \times b$  crossbar modules. The interconnection or link patterns between stages are such that there exists a unique path of constant length from any input to any output. Moreover, the path is digit controlled such that a switch connects an input to one of its  $b$  outputs depending on a single base- $b$  digit taken from the output address [1], [5]-[8].

We can determine the number of switch in each stage from the above definition. Specifically, the conditions of constant length paths and no unconnected terminals require that all the output terminals of one stage be connected to all the input terminals of the next stage in one-to-one fashion. The inputs of the first stage are connected to the source and outputs of the final stage are connected to the destination. Therefore,  $a' \times b'$  delta network has  $a'$  sources and  $b'$  destinations. Numbering the stages of the network as 1,2,... starting at the source side of the network requires that there be crossbar modules in the first stage. Then, the first stage has output terminals. This implies that the second stage must have input terminals, which requires crossbar

modules at this stage. In general,  $i$ th stage has crossbar modules of size  $a \times b$  [1].

We can specify the size of switch as replacing  $b$  output with  $a$ , such that we will have an  $a \times a$  switch. Hence, such a delta network with  $a \times a$  crossbar modules, has output as many as input, i.e.  $N$  input  $N$  output. An  $(N \times N)$  delta network consists of  $n$  stages of  $N/a$  ( $a \times a$ ) crossbar modules. Therefore, delta networks are a subclass of banyan networks, and superclass of the cube, indirect binary  $n$ -cube, omega, and baseline networks [1], [5]. An example of an  $(8 \times 8)$  delta network with  $(2 \times 2)$  switches is given in figure 1.

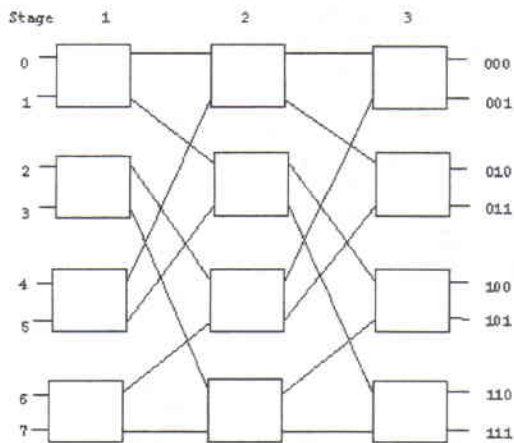


Figure 1. An  $(8 \times 8)$  delta network.

### 3. ANALYSIS OF BUFFERED NETWORKS

Two important performance measures to evaluate and compare the performance of buffered networks are *throughput* and *delay*. The *throughput* of a network is defined as the average number of packets passed by the network per stage cycle. The *delay* is the average number of stage cycle required for a packet to pass through the network [5].

The values of these measures depend on several network parameters as well as the external environment of the network. Therefore, it is necessary to place some restrictions on the network and its environment and to make some simplifying assumptions about the environment that generates and accepts packets. Hence, to evaluate these network parameters independent of their external environment, some assumptions are made as in [1]-[11], which is the uniform traffic model. The uniform traffic means that:

- 1) Packets are generated by  $N$  independent random processes for each of the  $N$  network input ports.
- 2) The destination addresses of packets are uniformly distributed over all the network output

port addresses. i.e. each switch buffer has the same amount of packets.

3) Packets are removed from the network as soon as they arrive at an output port.

4) Conflicts in each switch are resolved randomly.

The assumption that packets are generated by independent, uniform, random processes at the network input ports is made to facilitate the analysis and estimate the upper limits of the network performance. If the modules generating packets depend on the generation of other packets at other modules, then the actual rate of packet transfer through the network will be less than the observed transfer rate when the modules are independent. Additionally, if the distribution of destination addresses is not uniform, some parts of the network will be more congested than others, resulting a decrease in throughput. Similarly, the assumption that packets are removed from the network as soon as they arrive at an output port means that the environment must be able to accept packets at a faster rate than they can be passed by the system. Thus, these assumptions imply that, for each switching stage of the network, the packet distribution is identical and statistically independent for all switching elements (SEs).

We can classify delta networks as buffered and unbuffered delta networks. Unbuffered networks have been analyzed by Patel [1]. Buffered networks have been analyzed by Jenq [3] and Dias and Jump [4], and Yoon et al. [5].

In unbuffered networks, output data are delivered in discrete time intervals. An attempt is made to pass all input packets along the unique path from the network input link to the destination output link. When more than one packet request to pass through the same link, a *conflict* is said to occur. In this case, one of the conflicting packets is randomly selected and passed, others are rejected. In order to simplify the analysis, the rejected packets are assumed to be lost. In reality, the rejected packets would be resubmitted in the next time slot.

In buffered networks, each SE has predefined amount of buffer. If a packet conflict occurs, one of the conflicting packet is randomly selected and passed, other packets are stored in the buffers. This structure can be called *fixed* or *static* buffered networks.

In the static buffered networks each input of a SE has a predefined amount of buffer as shown in figure 2.a. On the other hand, if a network has a predefined amount of buffer pool for each SE as shown in figure 2.b, the system can be called *dynamic* buffered network. Static buffered networks are completely studied before [1], [2] and [5]. Performance analysis of multibuffered delta networks are studied in [5].

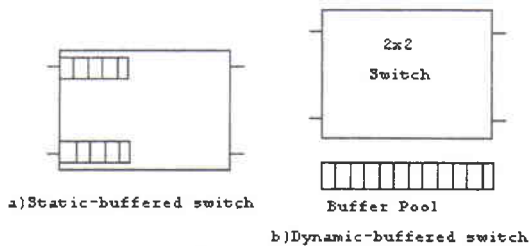


Figure 2. Demonstration of Static and Dynamic Buffers.

For the operation of static buffered networks, following assumptions are considered as in [2], [3], [4] and [5].

1) Each input port of an SE has a single buffer to accommodate a single packet.

2) A buffered network operates synchronously at a rate of  $t$ , called stage cycle, which consists of two phases.

- In the first phase, the availability of the buffer space at the subsequent stage along the destined path of a packet in the current buffer is determined; the packet is informed whether it may go to the next stage or should stay in the current buffer.

- In the second phase, packets may move forward one stage.

3) A packet is able to move forward only if it is selected among competing packets by the routing logic of the current SE, and either the buffer of the SE it would go to at the next stage is empty, or the packet in that buffer is able to move forward.

Related to static buffered networks, some variables are defined and a set of state equations are derived by Yoon et al. [5].

Analysis of multibuffered delta network is given in [5]. We will only give the results of [5], without considering the detailed formulation of the equations given in [5]. In a multibuffered network, each input port of every SE has a finite number of buffers so that multiple packets can be accommodated. For some discussion in this section, we need to distinguish between the buffers for an input port of an SE as a whole and an individual buffer slot. We shall refer to the former as a "buffer module," and the latter as a "buffer" in this section. We will also denote a buffer module of size  $m$  as an  $m$ -buffer [5].

In a multibuffered network, a packet is able to move forward one stage if either there is at least one empty buffer at the next stage, or a packet in the full buffer module at the next stage is able to move forward. We have already introduced a new variable  $m$  to denote the size of a buffer module.

The major difference between a single-buffer and an  $m$ -buffer is in the different possible number of buffer states. While a single-buffer can have only two possible states, full or empty, an  $m$ -buffer can be

in the one of  $m+1$  possible states, containing  $j$  packets with the probability  $P_j(k,t)$  for  $0 < j < m$ . Since only a single packet may be transmitted between stages per stage cycle, an  $m$ -buffer can change its state only among adjacent neighbor states ( $-1$  or  $+1$ ) plus the old state itself. The transition probabilities of moving from one state to another can be obtained by considering the ways in which one could move between two states and the associated probabilities for movements. For example, a buffer module remains in the old state  $j$ , if there is one departure and one arrival, or no departure and no arrival

#### 4. PERFORMANCE EVALUATION OF BUFFERED DELTA NETWORKS

Equations derived by Yoon et al. [5] for static  $m$ -buffered networks are very powerful, in that they can be used to determine the normalized throughput and the normalized delay of static buffered delta networks with the following four parameters:

- 1)  $a$  = SE size,
- 2)  $n$  = number of stages =  $\log_a N$  ( $N$  = network size),
- 3)  $m$  = static buffer size, i.e., the size of the buffer at each input port of an SE,
- 4)  $g(1)$  = input load applied to the network.

Among the many possible variations of the parameters, the most interesting cases are computed and plotted according to the equations given in [5].

Figures 3 and 4 are the analytic results which show normalized throughput versus network size, and normalized delay versus network size respectively for various static buffer sizes. It is seen that the normalized throughput decreases as the network size increases, and also as the buffer size decreases. The normalized delay decreases as the network size increases, and as the buffer size decreases.

The normalized throughput reaches the saturation point very quickly after the six buffer sizes. The increase in normalized throughput is very significant up to the buffer size of 3-4. The normalized delay increases almost linearly with the buffer size, especially for not very large networks. So, adding buffers to a packet switching network can increase throughput. But, for most applications, the number of buffers should be limited to three or four [5].

Figure 5 shows the normalized throughput versus static buffer size for which the network size is six ( $n=6$ ). Figure 6 shows the normalized delay versus static buffer size for which the network size is six ( $n=6$ ). These figures also compare the simulation results with analytical results. As seen from these figures, the normalized throughput taken from simulation results are less than those taken from analytical results; the normalized delay taken from simulation results are more than those taken from analytical results.

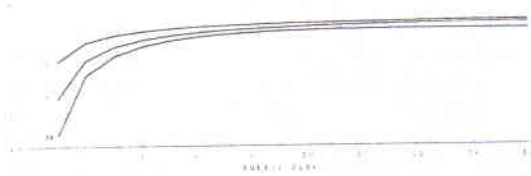


Figure 3. Normalised throughput versus static buffer size for static buffered delta networks ( $\alpha=2, q(1)=1$ ).

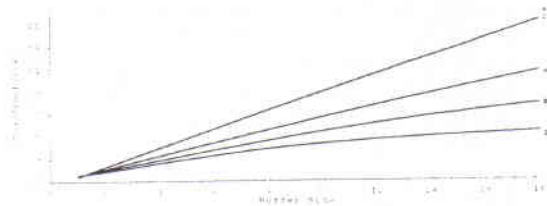


Figure 4. Normalised throughput versus static buffer size for static buffered delta networks ( $\alpha=2, q(1)=1$ ).

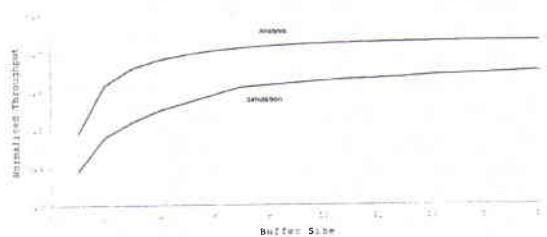


Figure 5. Normalised delay versus static buffer size for static buffered delta networks ( $\alpha=2, q(1)=1$ ).

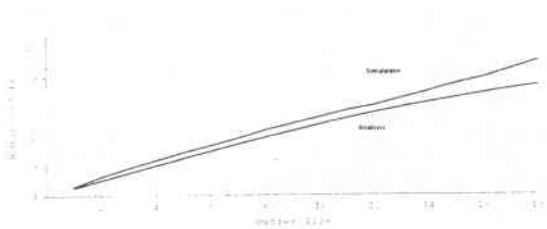


Figure 6. Normalised delay versus static buffer size for static buffered delta networks ( $\alpha=2, q(1)=1$ ).

Simulation studies on multibuffered networks have shown that some of the buffers are not full because of the random nature of the packets' destination addresses. Therefore, one of the buffer module of a SE may have more packets than another buffer module of the same SE; or especially when the input load is equal to one ( $q(1)=1$ ), the buffer modules of the SE of the first stage will have drastically more packets than the buffers of other SE of the other stages. We can conclude from here that, empty parts of a buffer module can be used by others which are filled by the packets. Hence, if we make buffers so that they can be used by more than one switch input, we may improve the performance of the network. So we can collect the buffer modules so that we will have a buffer pool as seen in figure 2. By doing so we can improve the network performance by either decreasing normalized delay or increasing normalized throughput or both. Computer simulation studies of dynamic buffers are made and performance of these networks are studied.

When the simulator is run, the packets are generated randomly from the  $N$  network input ports. Then by using the destination addresses, these packets pass through the IN. At each stage cycle, total packet time, which is used to calculate delay, is increased for each packet. At the output port, packets are counted for the calculation of throughput and delay. In order to find the steady state values of throughput and delay, the simulator is run for several times by increasing time ( $t$ , which is composed of discrete stage cycles). After some  $t$ ,  $S$  and  $d$  reaches a steady state value. This  $t$  is small for small network and buffer size (5000 is acceptable); but for large network and buffer size, it is big (10000 is acceptable). We have made the same assumptions during the implementation of simulator worked in [5]. These are as follows:

- 1) The  $N$  processors generate packets at each stage cycle with probability  $q(1)$  (input load).
- 2) The destination of each packet at each stage cycle by processors is set randomly by a random number generator to simulate uniform traffic.
- 3) If there is a conflict among packets within an SE, one of the conflicting packets is selected randomly.
- 4) The throughput and the delay are measured at each output port of the network, and averaged over the network size and simulation time span to get the normalized throughput and the normalized delay of the network.

5) Packet service at each buffer module is done by using first-in-first-out (FIFO) policy. Normalized delay ( $d$ ) and normalized throughput ( $S$ ) are calculated as:  
 $d = \text{total packet time} / \text{total packet} / n$  and  
 $S = \text{total packet} / t / N$ .

Simulation results for dynamic multibuffered delta networks have also been obtained by using the simulation program. In the plots for dynamic multibuffered delta networks, although size

of buffer pool are changed, the buffer size per switch input are not changed. In order to compare the static and dynamic buffered networks, figures are plotted such that the buffer size per switch input are the same. These are shown in figures 7 and 8.

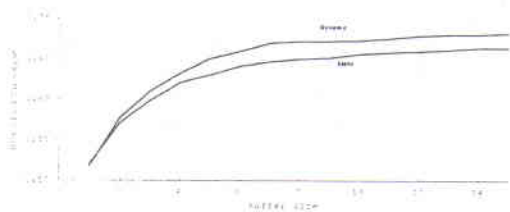


Figure 7. Comparison of static and dynamic buffered delta networks ( $n=3$ ,  $a=2$ ,  $q(1)=1$ ).

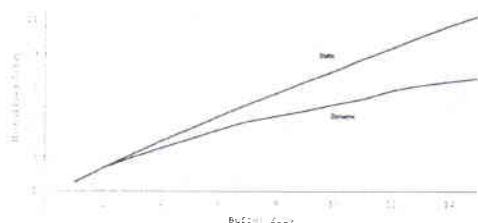


Figure 8. Comparison of static and dynamic buffered delta networks ( $n=3$ ,  $a=2$ ,  $q(1)=1$ ).

## 5. RESULTS AND CONCLUSIONS

The principal goal of this work is to find a multiprocessor interconnection network which improves the network performance. A new model for multibuffered delta networks was introduced in a multiprocessor environment. In this work, this new model is called as dynamic buffered delta networks; other buffered delta networks are called as static buffered networks. Analysis and simulation of the static buffered delta networks had been done by Jenq [3] and Yoon et. al. [5]. In this work, simulation of the dynamic buffered delta networks have been done. Also performance of the static and the dynamic buffered delta networks have been compared to each other.

It is well known that a network performance is better for uniform traffic than for nonuniform traffic. Because, with uniform traffic, network load is well distributed resulting in fewer path conflicts. Therefore analytic results are more optimistic than simulation results as seen in the figures plotted before.

Since the dynamic buffered networks are more closer to the uniform traffic model than the static buffered networks, the network performance can be improved or some buffers can be saved by making them dynamic.

As a result, in the dynamic buffers per switch case, when the same amount of buffer as in the static case is used, normalized throughput increases and normalized delay decreases. Hence, the network performance can be increased, or some amount of buffer can be saved, if buffers are made dynamic.

## REFERENCES

- [1] J. H. Patel, "Performance of Processor-Memory Interconnections for Multiprocessors," *IEEE Trans. Comput.*, C-30 (10), 771 (Oct. 1981).
- [2] D. M. Dias and J. R. Jump, "Analysis and Simulation of Buffered Delta Networks," *IEEE Trans. Comput.* C-32 (4), 273 (April 1981).
- [3] Y. C. Jenq, "Performance Analysis of a Packet Switch Based on Single Buffered Banyan Network," *IEEE J. Select. Areas Commun.*, SAC-3 (6), 1014 (Dec. 1983).
- [4] D. M. Dias and J. R. Jump, "Packet Switching Interconnection Networks for Modular Systems," *IEEE Computer*, 43 (Dec. 1981).
- [5] H. Yoon, K. Y. Lee, and M. T. Liu, "Performance Analysis of Multibuffered Packet-Switching Networks in Multiprocessor Systems," *IEEE Trans. Comput.*, 39 (3), 319 (March 1990).
- [6] T. H. Theimer, E. P. Rathgeb, and M. N. Huber, "Performance Analysis of Buffered Banyan Networks," *IEEE Trans. On Comm.*, 269, Feb. 1991.
- [7] S. Hsiao and C. Y. R. Chen, "Performance Analysis of Single-Buffered Multistage Interconnection Networks," *Proceedings of the Third IEEE Symp. On Parallel and Distributed Processing*, 1991.
- [8] T. Stouraitis, "Performance Evaluation of BIN/ABIN Networks in Buffered/Unbuffered Packet-Switched Environments," *IEE Proc. Computer and Digital Tech.*, Vol. 141, No.1, 29, Jan. 1994.
- [9] Y. Mun and H. Y. Youn, "Performance Analysis of Finite Buffered Multistage Interconnection Networks," *IEEE Trans. On Comput.* Vol. 43, No. 2, 153, Feb. 1994.
- [10] J. Ding and L. N. Bhuyan, "Finite Buffer Analysis of Multistage Interconnection Networks," *IEEE Trans. On Comput.* Vol. 43, No. 2, 243, Feb. 1994.
- [11] S. K. Bhogavilli and H. Abu-Amara, "Design and Analysis of High Performance Multistage Interconnection Networks," *IEEE Trans. On Comput.* Vol. 46, No. 1, 110, Jan. 1997.