International Journal of Parallel and Distributed Systems and Networks, Vol. 5, No. 1, 2002

A PERFORMANCE COMPARISON OF FOUR BUFFERING SCHEMES FOR MULTISTAGE INTERCONNECTION NETWORKS

B. Zhou* and M. Atiquzzaman**

Abstract

Multistage interconnection networks (MINs) are used to connect processors and memories in large-scale multiprocessor systems. MINs have also been proposed as switching fabrics in ATM networks. A MIN consists of several stages of small crossbar switching elements (SEs). A number of buffering schemes are used in the SEs to increase the throughput of MINs and prevent internal loss of packets. the objective of this article is to compare the performance of MINs using different buffering schemes in the presence of uniform and nonuniform traffic patterns. The results obtained from the study will help computer architects and network designers in choosing appropriate buffering strategies for fabric design and configuration of MINs. The normalized throughput, packet loss, and packet mean delay have been used as the performance measures for comparing the different buffering strategies. Results show that the performance of split-shared and output-buffered MINs is considerably better than that of input-buffered MINs when the hot request rate is low. However, the performance is identical for all the buffering schemes when the hot request rate is medium or high.

Key Words

Multistage interconnection networks, ATM switches, buffering schemes, performance analysis, hot spot traffic, throughput, packet loss, packet delay

1. Introduction

Multistage interconnection networks (MIN) have been found to be highly suitable for interconnecting a large number of processors and memories in large-scale multiprocessor systems. A MIN consists of a number of small crossbar switching elements interconnected by a permutation function. MINs can be broadly classified into two main categories: internally *blocking* and internally *nonblocking*. In an internally nonblocking MIN, two or more packets at different input ports can be simultaneously forwarded to two different output ports. A MIN is called internally blocking if two or more packets with distinct output port destinations cannot always be transferred to the output ports due to routing conflict within the MIN. For instance, resource contentions occur in MINs when more than one packet accesses the same internal link. Buffers are used in the SEs to store the packets that lose the routing conflicts in an internally blocking MIN. The packets are queued in the buffers for transmission during subsequent cycles.

The proper placement and arrangement of buffers in the SEs has a dramatic impact on the performance of the MIN. The implementation of *input buffered* SEs, operating in the first-in first-out (FIFO) fashion, is very simple in the sense that the internal links of the MIN have to operate at the same speed as the external input/output lines of the MIN. Therefore, internal speedup of the MIN is not required, and the hardware complexity can be lower than other buffering schemes to be discussed below. However, when a packet at the head of a queue in an SE waits for transmission to its destined output link, successive packets (which may be destined to different output links) in the queue must also wait. This phenomenon, called head-of-line (HOL) blocking, in input-buffered SEs reduces the throughput of the MIN.

In an *output-buffered* SE with separate buffers for each output link [1], a buffer must be able to receive up to d packets at a time, where d is the size (number of inputs) of the SE (Fig. 1). Output-buffered SEs do not suffer from the above-mentioned head-of-line blocking effect, and hence have higher throughput than input buffered SEs. The main drawback of output-buffered SEs is that they need to operate d times faster than the input (or output) lines of the MIN. This higher speed increases the implementation complexity and cost of the MIN. There are also SEs that combine input-buffering and output-buffering techniques, and in this case, the operating speed of the SEs can be lower than in the case of output buffered SEs [2].

The buffers in an SE can also be located at the *cross-point* inside the SE [3]. This buffering scheme removes the blocking of packets by a packet destined to a different output of the SE. All packets arriving at the inputs of a SE can, in principle, be transferred to their target buffers within one clock cycle.

Finally, another possibility to obtain high performance is through the use of a shared buffer [4]. In shared-buffer SEs of size $d \times d$, all input and output links of an SE have access to a shared buffer module that is able to write up

* Motorola Network Solutions Sector, Data Solutions Group, IL75 2C8, 1475 W. Shure Drive, Arlington Heights, IL 60004, USA

** School of Computer Science, University of Oklahoma, Norman, OK 73019-6151, USA; e-mail: atiq@ieee.org

(paper no. 204-0124)

to d incoming and read up to d outgoing packets in a clock cycle. There is no HOL blocking in shared-buffer SEs, and optimal throughput/delay performance is achieved. Furthermore, buffer utilization is better than input-, output-, or crosspoint-buffered SEs, thereby requiring a smaller number of buffers for the same performance. A shared-buffer MIN also has some additional features; for example, its basic architecture can be easily modified to handle several service classes through priority control functions to meet different service requirements. Multicasting and broadcasting can also be easily implemented in contrast to other types of architectures. The limitations of shared-buffer MINs arise from technological limitations. A buffer in an SE needs to queue d incoming and dequeue d outgoing packets per clock cycle. Therefore, the bandwidth of a buffer must be at least the sum of the bandwidths of the incoming and the outgoing lines.

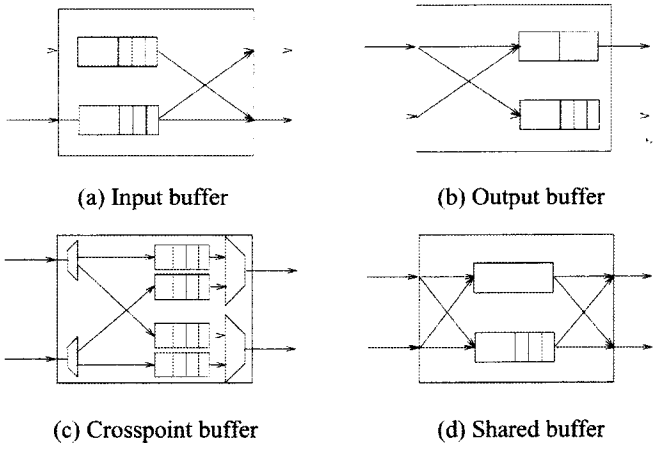

Figure 1. Four buffering schemes in switching elements.

In addition to be being used in multiprocessor systems, MINs have also been proposed as the switching fabrics in the future Broadband Integrated Services Digital (B-ISDN) networks [5]. The CCITT has standardized the asynchronous transfer mode (ATM) as the multiplexing and switching principle for the B-ISDN network [6, 7]. ATM is a packet-based and connection-oriented transfer mode, based on statistical time division multiplexing techniques. The information flow is organized in fixed-size packets called cells, consisting of a user information field (48 octets) and a header (5 octets). A very low cell loss probability must be guaranteed ($< 10^{-12}$), and intensive error and flow control protocols are provided at the edges of the switch. The line speeds are specified with nominal rates of 155.52 Mb/s and 622.08 Mb/s [5]. ATM will provide flexibility in bandwidth allocation and will allow a switch to carry heterogeneous services ranging from narrow-band to wide-band services requiring real time. However, the challenge is to build fast high-performance switches that are able to match the high speeds of the input links.

Early work in the performance analysis of unbuffered MINs was done by Patel [8]. Performance of unbuffered MINs in the presence of nonuniform traffic was reported in [9. 10]. The performance of input-buffered Banyan switches has been discussed in various publications. Dias [11] and Jenq [12] have analyzed MINs consisting of 2×2 SEs with single-packet input buffers and operating in the presence of uniform traffic in the MIN. Kruskal and Snir [13] have discussed buffered Banyans with output-buffered SEs for the case where the buffer capacity is infinite. Kim [14] reported a queuing analysis and simulation study of output-buffered Banyans with an arbitrary (finite) buffer size. All of the above performance analysis are based on the assumption that the MIN operates in the presence of a uniform traffic pattern.

A first approach to the analysis of single-buffered MINs in a nonuniform traffic environment is described in [15]. It is shown that certain nonuniform traffic patterns can have a crucial influence on the performance of the MIN. Kim [16] presented an analytical model to evaluate the performance of a single input-buffered Banyan switch under nonuniform traffic patterns.

Pfister and Norton [17] reported on a quantitative investigation of the performance impact of memory contention in highly parallel shared-memory multiprocessors. They first investigated the effect of a nonuniform traffic pattern consisting of a single hot spot of higher access rate superimposed on a background of uniform traffic. They found the potential degradation due to even moderate hot spot traffic to be very significant—severely degrading all memory access, not just access to shared lock locations—due to an effect they call tree saturation. They also found the technique of message combining to be an effective means of eliminating this problem if it arises due to lock or synchronization contention. Combining and feedback schemes have been suggested as partial solutions to the problem [18–20]. Atiquzzaman [10] proposed an efficient Markov chain model for the performance evaluation of a single-buffered Omega switch in the presence of a hot spot.

Three different buffer types for 2×2 SEs are analyzed in [21] for the unbounded queue size and queue size equal to one. The aim of this work is to study the performance of MINs with four different buffering schemes in the presence of uniform and hot spot traffic patterns. The results of this research work will enable the network designer to consider the buffering options for hardware implementation of buffered SEs in a MIN, to characterize the performance of low-cost hardware implementations, to obtain insight into the throughput limitations for different SE architectures, and to quantify the performance differences between the different types of SEs. Designers may use the results to weigh a higher cost implementation with higher-performance SE against a lower cost implementation with lower performance SE. In this study, *normalized throughput*, *packet loss*, and *mean delay* have been used as the performance measures.

The article is organized as follows. We describe the operating assumptions of internally buffered blocking MINs in Section 2. The simulation methodology is presented in Section 3. In Section 4 we present the performance results of the MINs using different buffering schemes in the SEs, in the presence of both uniform and hot spot traffic patterns; this is followed by concluding remarks in Section 5.

2. Switch Operating Assumptions

A MIN connects N inputs to N outputs using $n = \log_2 N$ stages of $N/2$ SEs per stage. We use a perfect shuffle permutation to connect adjacent stages as shown in Fig. 2 for $N = 8$. Each SE is a 2×2 crossbar allowing any input link to be connected to any output link. An SE has a finite number of buffers.

A packet arriving at an input port of the MIN consists of data and a destination address. The destination address is an n-bit number represented by $D = (d_1 d_2 \ldots d_{n-1} d_n)_2$. Destination tag routing is used to route a packet through the MIN. A SE at stage k inspects bit d_k, and in the case of no-conflict routes the packet to the upper or lower output of the SE depending on d_k being 0 or 1, respectively. A unique path of constant length exists between any input-output pair of the MIN, thereby rendering the MIN a blocking type of switch.

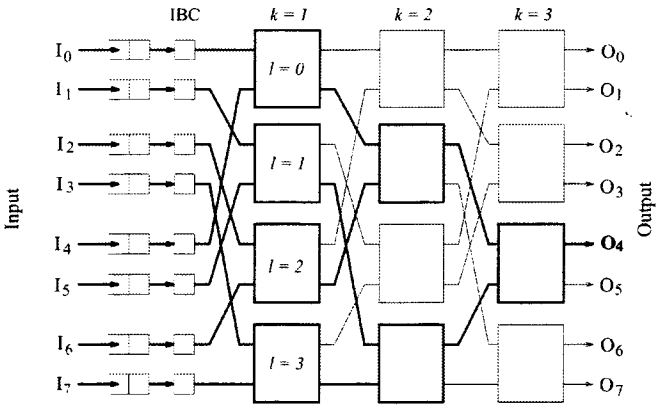

Figure 2. A three-stage MIN.

In addition to the buffers in the SEs, the MIN has input buffer controllers (IBCs) at every input of the MIN. To prevent packet loss in a MIN having finite-sized buffers at the SEs, IBCs with large buffer space are required in a MIN employing backpressure as the flow control mechanism.

We make the following assumptions regarding the operation and the environment of the MINs [10, 22].

1. The MIN operates *synchronously*, implying that packets move from one stage to the next only at the beginning of a time slot given by the stage clock, and thus the time axis is considered to be discrete. This reflects the situation in an ATM environment where all packets have a fixed length and fit exactly into one stage clock.
2. A *backpressure* mechanism [12] ensures that no packets are lost within the MIN. Thus, a packet can only leave its buffer if the corresponding destination buffer at the next stage is able to accept it.
3. As in [12], the arrival process at each input of the MIN is a simple *Bernoulli* process, that is, the probability that a packet arrives within a time slot is constant and the arrivals are independent of each other. This implies that the interarrival time between two packets is geometrically distributed with a minimum distance of one time slot.
4. Each input link of the MIN is offered the same *traffic* load. The probability that an input link generates a request at the beginning of a cycle is r.
5. There is *no blocking* at the output links of the MIN. This means that the output links have at least the same speed as the internal links.
6. The *conflict resolution* logic at each SE is fair for input-, output-, and split shared-buffer schemes, that is, routing conflicts among packets at the inputs of a SE are randomly resolved. We consider the following three selection policies for the cross-point buffering scheme:
   - *Random selection* (RS): the multiplexer randomly selects a packet from the buffer of contending packets for the given output.
   - *New packet selection* (NS): the multiplexer selects a packet from that buffer which has a new packet at the head of the queue. If there is no such packet, it selects a packet on RS basis.
   - *Blocked packet selection* (BS): the multiplexer selects a packet from that buffer which has a blocked packet at the head of the queue. If there is no such packet, it selects a packet on RS basis.
7. There are $N = 2^n$ inputs and N outputs in the MIN, where n is an integer.
8. The minimum possible *delay* of a packet is equal to $n + 1$, where n is the number of stages. It includes the delay at the IBC buffer, as at least one time unit is spent in each buffer even when there is no waiting.
9. The total amount of buffer per SE is $2m$. Therefore, the size of buffer per input or output port for input-, output-, or split shared-buffered SE is m, and $m/2$ for crosspoint buffered SE.
10. The uniform traffic pattern is defined to be the traffic pattern in which every source port has the same rate of incoming packets and they are destined to every destination port with the same probability $1/N$, where N is the number of output port.
11. For a hot-spot traffic pattern [17], a large proportion of the traffic is directed to a particular output called the *hot output*. The packets destined to the hot output are called *hot packets*. In Fig. 2, O_4 is the hot output and requests for O_4 are called hot requests. The internal links of the switch that carry hot packets are called *hot links*. Buffers that queue hot packets are called *hot buffers*. All SEs connected to hot links are called *hot SEs*. In Fig. 2, the SEs and links that carry hot traffic are shown in boldface.
12. For a hot spot traffic pattern, the probability that a generated request will be directed to a non-hot or a hot output port are $(1 - h)/N$ and $h + (1 - h)/N$ respectively, where h is defined to be the hot spot probability.

3. Simulation Method

The assumptions mentioned in Section 2 are implemented in the simulator as follows.

1. At each cycle, a packet generator generates a packet (processor requests memories in computer systems)

with probability r (traffic load) at an input of the MIN. The packet generation is independent of packets generated at previous cycles and those at the other input ports.

2. The destination of a generated packet is taken from a uniform random number generator in the case of uniform traffic, and in the case of hot spot traffic from a nonuniform random number generator that generates requests according to the hot spot probability (h) distribution mentioned in Section 2.
3. If there is a routing conflict among packets within a SE, a packet is selected randomly by another random number generator for input-, output-, and split-shared buffered SEs. In the case of crosspoint-buffered SEs, either the randomly selection (RS) or the blocked packet selection (BS) is used (see Section 2).
4. FIFO queuing policy is used at the buffers in the SEs of the input-, output-, and crosspoint-buffered SEs. Window selection policy is employed in the shared-buffered SEs.
5. The throughput and delay are measured at each output of the MIN, and averaged over the MIN size and simulation time span (typically 50,000 cycles) to get the normalized throughput and the mean delay of the MIN. The outputs for the first 500 cycles are discarded to allow the MIN to reach a steady state.

The simulator was written in C language. The following input data values to the simulator were varied each time to have a comprehensive picture of the switch behavior:

1. Number of simulation cycles (t_2) performed is large, typically 50,000, and initial simulation cycles t_1.
2. Seed for the random number generator: The simulator required two independent streams of numbers, one for the generation of the request and another other for resolution of the conflicts.
3. System size (N): Different MIN sizes are simulated.
4. Probability of a packet arrival (r).
5. Probability (h) of a packet being destined for the hot output.
6. Internal buffer size (m) and IBC buffer size (f).

3.1 Request Generation

The built-in random number generator in the C language library is used to obtain random requests at the beginning of each cycle. The random number generator is appropriately divided to generate requests according to the input parameters (i.e., rate of request generation and the probability of accessing the hot output). The actual demarcation process is portrayed in Fig. 3. We define the following:

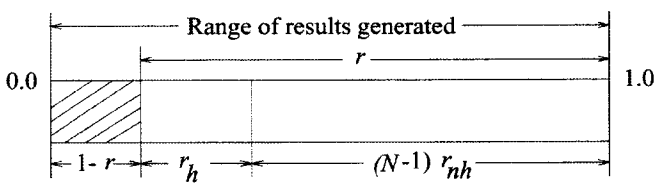

Figure 3. Demarcation of requests generated by a pseudo-random generator.

r = portion that is valid request
$1 - r$ = portion that is invalid request
h = probability of hot spot
$(N-1)r_{nh}$ = portion that is uniformly distributed among $N-1$ nonhot outputs

The effective non-hot and hot spot probabilities are given by $r_{nh} = \frac{(1-h)r}{N}$ and $r_h = rh + \frac{(1-h)r}{N}$ respectively.

3.2 Parameters Evaluated

Normalized throughput, packet loss, and mean delay of packets are used as the criteria for comparing the performance of the different buffering schemes. When the MIN reaches a steady state after t_1 clock cycles, the number of valid packets at the outputs of the MIN is counted at the end of each cycle. These are averaged over a large number of cycles to give the normalized throughput (μ) as follows:

$$\mu = \frac{1}{N(t_2 - t_1)} \sum_{l=0}^{N-1} \sum_{t=t_1}^{t_2} \mu(l,t) \tag{1}$$

where $\mu(l,t)$ is the throughput at the lth output of the MIN during cycle t.

For uniform traffic, $\mu(l,t) = \mu(t)$ for all l and t. The packet loss probability (η) for uniform traffic is therefore given by:

$$\eta = \frac{r - \mu}{r} \tag{2}$$

The mean packet delay in the MIN is obtained by averaging the delay experienced by the packets over a large number of cycles. It is given by:

$$\tau = \frac{1}{N(t_2 - t_1)} \sum_{l=0}^{N-1} \sum_{t=t_1}^{t_2} \tau(l,t) \tag{3}$$

where $\tau(l,t)$ is the delay experienced by a packet (if there is one) at the lth output of the MIN during cycle t, where t_1 is the number of initial simulation cycles allowed for the MIN to stabilize.

4. Results and Discussion

Four simulators have been developed for the simulation of MINs using input-, output-, crosspoint-, and split shared-buffered SEs. In this study, we have considered two types of traffic pattern: uniform and hot spot traffic. In the hot spot traffic pattern, there is an output port that is accessed more often than other output ports. For example, many telephone callers may contend in calling a popular location; many nodes may synchronously report some information to one node (say, the switch control center) for administrative purposes. Such traffic can be characterized by a single hot spot of a higher access rate, superimposed on a background of uniform traffic [17]. Hot spot traffic causes tree saturation in an MIN. Tree saturation degrades the performance of the entire switch

system, including those not participating in the hot spot activity.

We have simulated various MIN sizes under uniform and hot spot traffic patterns. Due to space limitations, we show only the results for switches of size 64×64. The total amount of buffer space is assumed to be the same for each buffering scheme. For instance, a total buffer space of 12 implies that each input buffer is of size 6 and each crosspoint buffer is of size 3.

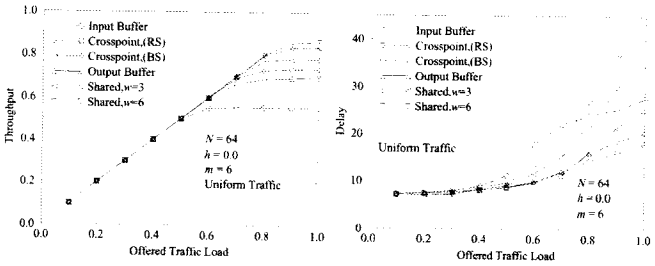

Figure 4. Normalized throughput and mean delay versus offered traffic load under uniform traffic with $m = 6$.

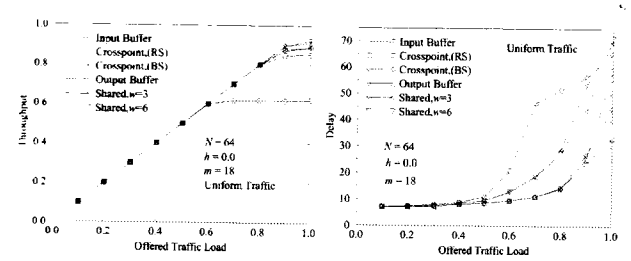

Figure 5. Normalized throughput and mean delay versus offered traffic load under uniform traffic with $m = 18$.

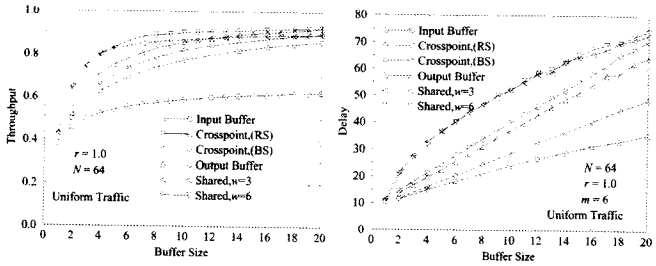

Figure 6. Normalized throughput and mean delay versus buffer size under uniform traffic.

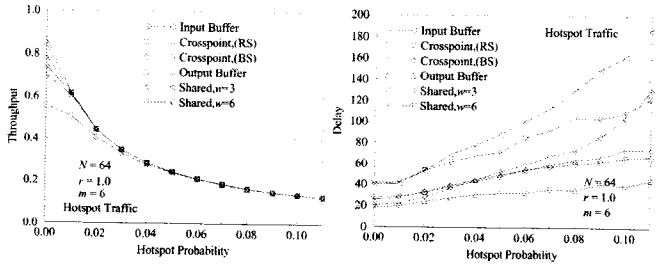

Figure 7. Normalized throughput and mean delay versus low hot spot probabilities.

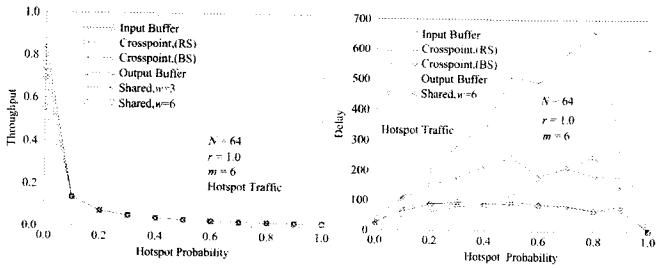

Figure 8. Normalized throughput and mean delay versus high hot spot probabilities.

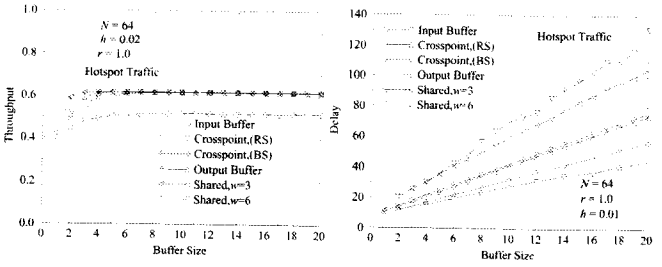

Figure 9. Normalized throughput and mean delay versus buffer size under hot spot traffic.

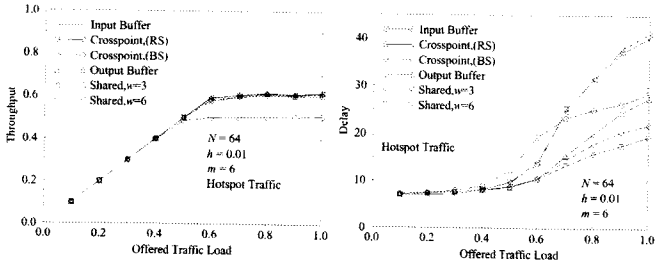

Figure 10. Normalized throughput and mean delay versus offered traffic load under hot spot traffic, $h = 0.01$.

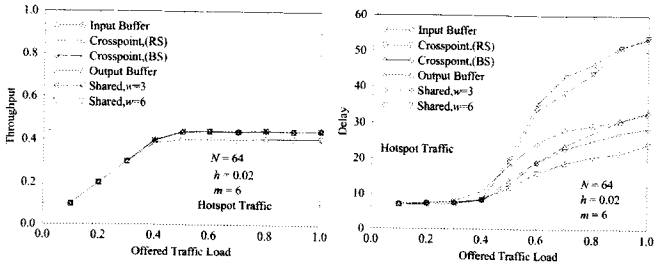

Figure 11. Normalized throughput and mean delay versus offered traffic load under hotspot traffic, $h = 0.02$.

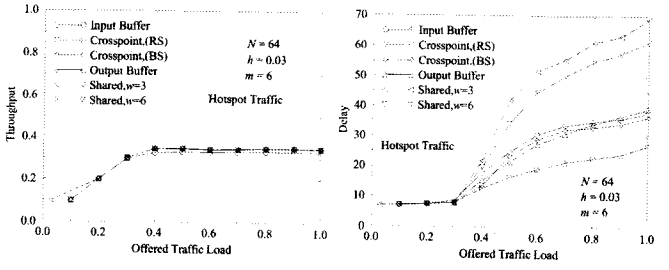

Figure 12. Normalized throughput and mean delay versus offered traffic load under hot spot traffic, $h = 0.03$.

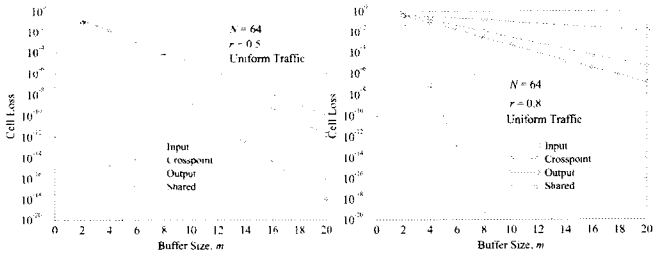

Figure 13. Packet loss versus buffer size, $N = 64$.

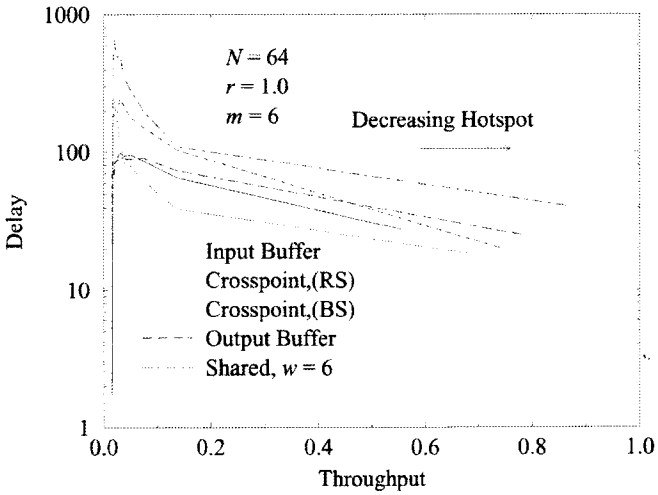

Figure 14. Mean delay versus normalized throughput under hot spot traffic.

Fig. 4 shows the normalized throughput and mean delay versus input offered traffic load (r) for MINs using four types of SEs under uniform traffic pattern ($h = 0.0$). For input-buffered SEs, the maximum normalized throughput of a MIN with buffer size six is limited to 0.56 under uniform input traffic pattern. This bottleneck, due to the head-of-the-line contention at each SE, is intrinsic to input queuing. When a packet at the head of a queue loses a contention, it prevents the rest of the packets in the same buffer from progressing forward, if packets are served on a FIFO basis. Another bottleneck arises when two or more packets contend for the same buffer in a SE. As only one packet can be admitted to the buffer in one clock cycle, one of them is blocked and will have to retry in the next clock cycle. When buffers are placed at the output port of each SE or are shared, a very high throughput can be achieved. From Fig. 4, we see that the maximum normalized throughput of 0.78 is achieved for output buffer, and 0.83 and 0.86 for split shared buffer with $w = 3$ and 6, respectively, where w is the window size.

In Fig. 5, the normalized throughput of various buffering schemes is shown as a function of the arrival rate for $m = 18$. The maximum normalized throughput of the input buffered MIN built with 2×2 SEs is limited to about 62% even with a very large buffer size. However, with crosspoint-, output-, or shared-buffering strategies, a normalized throughput of almost 90% is possible with moderately large buffer sizes. Split shared buffering performs the best under uniform traffic. Performance of the window selection policy can be drastically improved even with a small window size (w). The performance of crosspoint buffering approaches the performance of output and split shared buffering when the buffer size is increased to 18. Crosspoint buffering provides performance comparable to output and split shared buffering under operating loads below 80%. Figs. 4 and 5 also show the packet mean delay as a function of the arrival rate for various buffering schemes. When $m = 6$, reasonable delays can be achieved for both crosspoint buffering and output buffering up to a load of 0.6. However, when the buffer size is increased to 18, reasonable delays can be achieved for loads up to 0.8. The mean packet delay of the four buffering schemes is also compared. The offered load is varied from 0.1 to 1.0. It is shown the mean packet delay for MINs using crosspoint-buffered SEs is smaller than that of input-, output-, or split shared-buffered SEs.

Fig. 6 shows the normalized throughput and mean delay versus buffer size for MINs employing different buffering schemes under a uniform traffic pattern. Split shared buffering has the highest normalized throughput, followed, in decreasing order, by output buffering, crosspoint buffering, and input buffering. For mean delay, crosspoint buffering is the lowest, followed by output buffering, input buffering, and split shared buffering.

Fig. 7 plots the normalized throughput and mean delay for low hot spot probabilities (h) for four buffering schemes under full load. h is varied from 0 to 0.11. Fig. 8 shows the normalized throughput and mean delay for high hot spot probabilities for various buffering schemes. h is varied from 0 to 1. As the hot spot probability increases, the normalized throughput decreases due to tree saturation. The mean delay increases when h is low and decreases when h is high ($h \geq 0.8$). This can be explained as follows: There are two factors influencing the switch mean delay: the tree saturation due to the hot traffic, which increases the packet's delay time; and the fact that as h increases, the switch's normalized throughput decreases (i.e., the number of packets that are successfully switched to the output port is decreases). The net result of the above two factors determines the mean delay of the switch. When h is low, as h increases, hot traffic saturates the hot buffers, and the packets delay increases. When h is high ($h \geq 0.8$), as the normalized throughput is very small, the mean delay is decreases. For $h = 1$, the larger the switch size, the smaller the switch's mean delay is (see Table 1).

Fig. 9 shows the normalized throughput and mean delay versus buffer size for MINs with different buffering schemes under a hot spot traffic pattern. Figs. 10–12 show the normalized throughput and mean delay versus offered load for MINs under hot spot probabilities of 1%, 2% and 3%. When h equals 0.01, the tree saturation occurs approximately at 0.6 for output-, crosspoint-, and split shared-buffered SEs and at 0.5 for input buffering (see Fig. 10). When $h = 0.02$ (see Fig. 11), tree saturation occurs at $r = 0.5$ for output- crosspoint-, and split-shared buffered SEs and at $r = 0.4$ for input-buffered SEs. At a low hot spot probability ($h = 0.01$), the output-, crosspoint-, and split shared-buffer SEs have higher normalized throughput

Table 1
Mean Delay and Normalized Throughput versus Switch Size

Input Buffering, $h = 1.0$, $m = 6$, $\lambda = 1.0$								
Var.	$N = 2$	$N = 8$	$N = 32$	$N = 64$	$N = 128$	$N = 256$	$N = 512$	$N = 1024$
Delay	6.998667	11.495917	12.320958	4.689625	1.368812	1.187500	1.105469	0.007812
Throughput	0.500000	0.125000	0.031250	0.015625	0.007812	0.003906	0.001957	0.000977

than the input-buffered SEs, as shown in Fig. 10. This is due to the HOL blocking in the case of input-buffered SEs. And in high hot spot probability (say, h larger than 0.02), the normalized throughput is almost the same for all the buffering schemes, because of the tree saturation mentioned in Section 1. Comparison of results shows that the performance of the split shared-buffer MIN is the best, and that of an output-buffered and cross-point MIN is much better than an input-buffered MIN when the hot request rate (h) is low. But the performance is almost the same for all the types of MINs when the level of hot requests is medium or high. This is again due to tree saturation. Fig. 13 shows the packet loss probability with respect to the number of buffers in uniform traffic situations. The traffic load, r, is chosen for 0.5 and 0.8. The packet loss probability linearly decreases with increasing buffer size m. It decreases slowly at a heavy offer load of networks but sharply at a small load of networks.

In split shared-buffering switch, good packet loss characteristics can be achieved because of the probability of the blocked packet being very small. To achieve packet loss probability of 10^{-8} with offered load $r = 0.5$, only a packet buffer 3 is required for split shared buffering, 9 for output buffering, and 14 for crosspoint buffering. With offered load $r = 0.8$, it is possible with a split shared buffer size of 3 to have a packet loss probability of 10^{-4}. For the same loss probability, 12 and 16 buffers are needed for output buffering and crosspoint buffering, respectively. And input buffering never reaches this value.

In Fig. 14, the mean packet delay in MINs using input-, output-, crosspoint-, and split shared-buffered SEs as a function of normalized throughput is shown for h decreasing from 1 to 0. We note an interesting phenomenon: the mean packet delay first increases to a maximum and then decreases as the normalized throughput increases due to a decrease in the hot spot probability.

Certain general conclusions can be drawn:

- Crosspoint, output-, and split shared-buffered SEs with a large buffer size have similar normalized throughput. The normalized throughput of these three types of SEs is much better than that of input-buffered SEs. For a small buffer size, the split shared buffer has the best performance, followed by output buffering, crosspoint buffering, and input buffering.
- For offered traffic loads under 60%, the normalized throughput of the input-buffered MIN is reasonably close to that of output buffering, crosspoint buffering and split shared buffering. Because of its lower cost, input buffering may be the choice for implementation. However, in addition to the normalized throughput limitation, input-buffered SEs show a significant increase in mean packet delay even at loads as low as 60% (see Fig. 5).
- Adding large buffers to a input-buffered SE will not bring about a substantial performance improvement, as the normalized throughput is limited to 62% due to head-of-the-line contention. Output-, crosspoint-, and split shared-buffered SEs with small buffer sizes have significantly better performance than input-buffered SE with a large buffer size. Thus, if additional hardware resources are available to improve switch performance, they are better spent on implementing output-, crosspoint-, or split shared-buffering designs than on making larger buffers for a input-buffering design.
- The optimal buffering strategy depends on the demands of types of traffic.

5. Conclusion

A number of simulators have been developed to evaluate the performance of MINs with different internal buffering schemes under uniform and hot spot traffic environments. Results confirm the intuition that, under uniform traffic, split shared-buffer SEs have better performance than SEs using input buffering, output buffering, or crosspoint buffering. In addition to performance, there are other issues, such as implementation, that must be considered in designing a MIN. We also compared the performance of SEs with buffers at the inputs, outputs, crosspoints, and shared between the inputs and outputs, under the hot spot traffic pattern. The results show that the performance of split shared and output-buffered MINs is considerably better than that of input-buffered MINs when the hot request rate is low. But the performance is almost the same for all the buffering schemes when the hot request rate is medium or high. This is due to the onset of tree saturation at medium traffic loads. It is difficult to say that one buffering scheme is better than another in all aspects. A buffering scheme appropriate for one type of application may not be appropriate for another type of application. For example, a shared-buffered switch is suitable for a small-scale network but has limitations in a growing system. A comparison of the four approaches to providing queuing for SEs in MINs is given below.

Input Buffer

- The buffering structure is simple.
- The internal link speed of the MIN is equal to the speed of the inputs or outputs of the MIN.
- Throughput is limited to 75% of the offered load due to head-of-the-line blocking when 2×2 SEs are employed.

Table 2
Recommendation for Choosing Appropriate Buffering Strategies

Buffering Scheme	Packet Loss	Delay	Suitable Service	Packet Loss	Delay
Input	Very high	High	Voice	10^{-3}	500ms
Crosspoint	High	Very low	Data	10^{-6}	50ms
Output	Low	Low	Hifi sound	10^{-7}	1000ms
Split shared	Very low	Very high	Video	10^{-8}	1000ms

- Suitable for voice service.

Output Buffer

- There is a separate buffer for each output.
- Achieves optimal throughput/delay performance.
- The buffers should operate at a speed that is equal to the sum of the speeds of the input links of a SE.
- Suitable for hifi sound service.

Crosspoint Buffer

- The buffering structure is simple.
- The internal speed can be equal to the speed of the input/output links of an SE.
- Reduces the effect of head-of-the-line blocking.
- The total buffer required is much greater.
- Suitable for data transmission service.

Split Shared Buffer

- The buffering structure is complex.
- Achieves high utilization of the buffers.
- The total amount of buffer memory required is small.
- Suitable for broadcast video service.

References

[1] B. Zhou & M Atiquzzaman, Performance of output-multibuffered multistage interconnection networks under general traffic patterns, *IEEE INFOCOM'94: Conf. Computer Communications*, Toronto, June 1994, 1448–1455.

[2] Y. Oie, M. Murata, K. Kuboto, & H. Miyahara, Effect of speedup in nonblocking packet switch, *ICC'89 Conf.*, Boston, MA, June 1989, 410–414.

[3] P. Goli & V. Kumar, Performance of a crosspoint buffered ATM switch fabric, *IEEE INFOCOM'92*, 1992, 426–435.

[4] J.S. Turner, Queuing analysis of buffered switching networks, *IEEE Trans. on Communications, 41*(2), 1993, 412–420.

[5] CCITT Recommendation I.121, *Broadband Aspects of ISDN*, 1989.

[6] B. Zhou, K.E. Forward, & G.J. Armitage, Simulation study of the interaction between a multi-media terminal and the ATM network, *J. Electrical and Electronics Engineering, Australia, 13*(1), 1993, 41–52.

[7] M. Atiquzzaman & C.K. Chen, Realistic modeling of blocked packets for accurate performance evaluation of multistage ATM switches, *IEEE Proc.—Communications, 146*(4), 1999, 213–221.

[8] J.H. Patel, Performance of processor-memory interconnection for multiprocessors, *IEEE Trans. Comput., C-30*(10), 1981, 771–780.

[9] M. Atiquzzaman & M.S. Akhtar, Effect of hot spots on the performance of multistage interconnection networks, *FRONTIERS 92: The 4th Symp. on the Frontiers of Massively Parallel Computation*, Virginia, October 1992, 504–505.

[10] M. Atiquzzaman & M.S. Akhtar, Effect of non-uniform traffic on the performance of multistage interconnection networks, *9th Int. Conf. on Systems Engineering*, Las Vegas, NV, July 1993, 31–35.

[11] D.M. Dias & R. Jump, Analysis and simulation of buffered delta network, *IEEE Trans. Comput. C-30*, 1981, 273–282.

[12] Y.C. Jenq, Performance analysis of a packet switch based on single-buffered banyan network, *IEEE J. Select. Areas Commun. SAC-1*, 1983, 1014–1021.

[13] C. Kruskal & M. Snir, The performance of multistage interconnection networks for multiprocessors, *IEEE Trans. Comput. C-32*, 1983, 1091–1098.

[14] H.S. Kim, I. Widjaja, & A. Leon-Garcia, Performance of output-buffered banyan networks with arbitrary buffer sizes. *IEEE INFOCOM'91: Conf. on Computer Communications*. Bal Harbour, FL, April 1991, 701–710.

[15] D.S. Meliksetian & C.Y.R. Chen, A Markov-modulated Bernoulli process approximation for the analysis of banyan networks, *ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems*, May 1993, 183–194.

[16] H.S. Kim & A.L. Garcia, Performance of buffered banyan networks under nonuniform traffic patterns, *IEEE Trans. Commun., 38*(5), 1990, 648–658.

[17] G.F. Pfister & V.A. Norton, Hot spot contention and combining in multistage interconnection networks, *IEEE Trans. Comput., C-34*(10), 1985, 943–948.

[18] S.L. Scott & G.S. Sohi, The use of feedback in multiprocessors and its applications to tree saturation control, *IEEE Trans. on Parallel and Distributed Systems*, 1990, 943–948.

[19] G. Lee, C.P. Kruskal, & D.J. Kuck, The effectiveness of combining in shared memory parallel computers in the presence of hot-spots, *Int. Conf. on Parallel Processing*, Pennsylvania, August 1986, 35–41.

[20] P.C. Yew, N.F. Tzeng, & D.H. Lawrie, Distributing hot-spot addressing in large-scale multiprocessors, *IEEE Trans. Comput., C-36*(4), 1987, 269–277.

[21] O.E. Percus & S.R. Dickey, Performance analysis of clock-regulated queues with output multiplexing in three different 2×2 crossbar switch architectures, *J. Parallel and Distributed Computing, 16*(1), 1992, 27–40.

[22] B. Zhou & M. Atiquzzaman, Performance of output-multibuffered multistage interconnection networks under nonuniform traffic patterns, *Int. Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'94)*, North Carolina, Jan. 31–Feb. 2, 1994, 405–406.

Biographies

Bin Zhou received the B.E. in 1982 and the M.Sc. in 1993 from University of Melbourne, and his Ph.D. in 1995 from Monash University of Australia, all in electrical and computer engineering. He has been an academic staff member at the Institute for Telecommunications Research, University of South Australia. As a senior software researcher and development engineer, he has worked with

the Motorola Australia Software Center. Currently he is a lead systems engineer in the Data Solutions Group of the Network Solutions Sector in Motorola. His research interests include asynchronous transfer mode, third-generation cellular phones network, mobile IP, general packet radio services, and code division multiple access.

Mohammed Atiquzzaman received the M.Sc. and Ph.D. degrees in electrical engineering and electronics from the University of Manchester Institute of Science and Technology, U.K., in 1984 and 1987, respectively. He has been an academic staff member at Monash University (Melbourne, Australia) and the University of Dayton (Ohio, USA). Currently he is a faculty member in the School of Computer Science at University of Oklahoma, USA. He is a senior editor of the *IEEE Communications Magazine* and serves on the editorial boards of *Computer Communications*, *Telecommunication Systems*, and the *Journal of Real-Time Imaging*. He has been a guest editor of special issues on ATM switching and ATM networks for the *International Journal of Computer Systems Science & Engineering*, a special issue on enterprise Networking in *Computer Communications*, a special issue on next generation Internet in *European Transactions on Telecommunications*, and feature topics of *IEEE Communications* on traffic management and switching for multimedia and on optical networks, communication systems and devices. He is the conference co-chair of the SPIE Conference on Quality of Service over Next Generation Data Networks, Colorado, 2001. He has also served on the technical program committee of many national and international conferences, including IEEE INFOCOM, IEEE Globecom, and IEEE Annual Conference on Local Computer Networks. His current research interests are in quality-of-service over next-generation Internet, including broadband ISDN and ATM networks, multiprocessor systems, interconnection networks, and parallel processing. He has over 100 refereed publications in the above areas, most of which can be accessed at www.cs.ou.edu/$\sim$ atiq.