Chapter Four

Buffer Performance Evaluation with Synchronous Networks

This chapter presents evaluations of the performance of communication networks composed of switches implemented with the DAMQ buffer. The goal is to determine the performance characteristics of the DAMQ buffer and evaluate various design parameters associated with this buffer. In order to make this evaluation, the DAMQ buffer's performance is compared to the performance of several other buffer types: FIFO buffers, two statically allocated non-FIFO buffer architectures (SAMQ and SAFC buffers), and a central buffer pool whose space is dynamically allocated to incoming packets.

Chapter 3 argued that the use of FIFO buffers reduces network performance due to the susceptibility of FIFO buffers to output port contention and that non-FIFO buffer architectures which utilize static buffer allocation inefficiently utilize their available buffer space. Further, a strong case as made for the difficulty of implementing statically allocated non-FIFO buffers (the need for pre-routing packets, etc.) and of implementing central buffer pools. Nevertheless, there are a number of reasons for comparing the performance of these particular buffers to the performance of DAMQ buffers. First, they facilitate the evaluation of specific architectural features of the DAMQ buffer. For instance, by including the SAMQ buffer in the comparisons, the impact of dynamic buffer allocation on the performance of the DAMQ buffer can be isolated. Second, some of thse buffer organizations (specifically, SAFC) may be worthy of consideration in their own

right. Finally, while the CBDA buffer presents impossible implementation requirements (as it is simulated in this chapter — see Sec. 4.1), it serves as a ‘‘pseudo-optimal’’ buffer architecture under uniform traffic conditions, providing an estimate of the available communication network performance which the other buffer architectures fail to tap.

The next section discusses the difference between *synchronous* and *asynchronous* communication networks and discusses why we present results from evaluations of synchronous networks in this chapter. Sec. 4.2 describes the Markov model used to evaluate single 2×2 switches, presents the results obtained from the model and examines these results. The following section describes the event-driven simulator used to measure the performance of synchronous multistage interconnection networks. This section describes the switch behavioral model used in the simulator, including the algorithms for discarding and blocking flow control and for arbitrating the crossbar of each switch. Secs. 4.4 and 4.5 present and discuss the results obtained by simulating networks of discarding and blocking switches, respectively, which are transmitting uniformly distributed traffic. Sec. 4.6 examines the performance of networks of blocking switches when communication traffic is not uniform. The chapter concludes with a summary of the results and analyses presented in this chapter.

4.1. Synchronous and Asynchronous Networks

This chapter and the following use analysis and simulations of a *synchronous* communication network in order to compare the performance of different buffer architectures. However, Chs. 6 and 8 perform their evaluations in the context of an *asynchronous* network. In the context of a multicomputer communication network, the terms *synchronous* and *asynchronous* do not refer to the presence or absence of a clock, but rather to the synchrony (or lack thereof) with which packets are transmitted.

In a synchronous network, time is globally divided into *stage cycles* [Yoon90]. A stage cycle is the time required for a packet to travel one hop in the network. Each switch of a synchronous network may transmit packets at each stage cycle boundary, with at most one packet traversing a link each stage cycle. In an asynchronous network, on the other hand, each switch operates independently of its neighbors and is capable of transmitting or receiving a packet during any clock cycle that an output or input port is available. Packets require multiple clock cycles to be transmitted received (as opposed to a single stage cycle), and the number of cycles is dependent upon the size of the packet.

The use of a synchronous network has several implications. First, virtual cut-through [Kerm79] is not supported. As was discussed in previous chapters, virtual cut-through can dramatically reduce communication latency in lightly-utilized communication networks. However, in a synchronous network, packets can only travel one hop in a stage cycle. Using a synchronous network also eliminates the possibility of transmitting variable length packets; since transmission service time is constant (one link per stage cycle per packet), it is not meaningful to consider

variable-length packets.

Asynchronous networks are used by most of the academic and commercial multicomputers built in recent years: the Intel Delta and Paragon [Lill91], MIT's J-Machine [Dall87b], *T [Nikh92], Alewife [Agar90], the Cray T3D [Oed], and others. Exceptions include the BBN Butterfly [Crow85] and Monarch [Rett90] machines, which use a synchronous networks. Further, the DAMQ buffer architecture was intended for use in an asynchronous network, with features specifically designed to support virtual cut-through and variable-length packets. However, simulations of synchronous communication networks have been widely used in research on packet switching networks [Dias81, Kuri89, Lang88, Dias89, Yoon90, Kuma84, Pfis85b, Lee86, Yew86, Kuma86, Knig89, Scot90, Tami88b, Tami92a, Tami92b]. For this reason and because a synchronous switch is simpler to model, this chapter evaluates the DAMQ buffer in the context of a synchronous network (as does Ch. 5, which examines the ability of various buffer architectures including the DAMQ buffer to support real-time communication).

4.2. Evaluation of 2×2 Discarding Switches Using Markov Models

We have evaluated the performance of individual 2×2 discarding switches using Markov models. An entire network was not modeled due to the intractable number of states which would result. Several simplifying assumptions were made: (1) fixed length packets, (2) uniform distribution of packet destinations, (3) when there is contention for an output port, the input port that is allowed to transmit is chosen randomly, and (4) synchronous store-and-forward operation of the switch so that during each *stage cycle* [Yoon90] packets either completely arrive or

completely depart. Since we assume a uniform packet size, the amount of buffer memory occupied by each packet is a constant. Therefore, the *packet slot* (the amount of memory occupied by a packet) is used as the unit of buffer storage in this analysis.

The model used to generate the Markov state transition graph for each switch allows the switch to simultaneously receive and transmit messages during each stage cycle. A packet is discarded if and only if it arrives at a full buffer which is not currently transmitting a packet. This corresponds to the architecture described in Ch. 3, where the buffer supports simultaneous read and write operations. The operation of the switch is equivalent to a switch that alternately sends and receives packets, where the state of the switch is defined by the packets in its buffers *after* packet transmission but *before* packet reception. With this model, a packet is discarded when it arrives at a full buffer. Thus, when a packet arrives at the switch, the probability that it will be discarded is:

$$P(discard) = \sum_{i \in I} P(i) + 0.5 \sum_{j \in J} P(j)$$

where I is the set of states in which both input ports are full, J is the set of states in which one of the input ports is full, and $P(k)$ is the probability of being in state k after any given stage cycle. Packets arrive at each input port of the switch with a probability equal to the applied traffic rate and with equal probability of being destined for either output port. The transition probabilities between states is calculated in two stages: packet arrival puts the switch into an intermediate state, and then arbitration and transmission moves the switch to the next state. The product of these probabilities is the probability of transferring from one state to the

next via a particular intermediate state. The sum of all such transfers is the total probability of that state transition.

All four practical switches as well as the CBDA switch were evaluated at varying applied traffic rates and different buffer sizes. The applied traffic rate corresponds directly to the probability of a packet arriving at an input port, i.e., for a switch operating with 70% applied input traffic rate each input port has a probability of 0.70 of having a packet arrive at each long clock cycle. From our model we could determine the probability that a given packet arriving at a switch will be discarded for a given level of traffic. The results are presented in Tab. 4.1. Since the SAMQ and SAFC switches statically allocate buffer space to each of the output ports, they can only have an even number of packet slots in each buffer.

As shown in Tab. 4.1, the switch with DAMQ buffers performs better (lower probability of discarding) than any of the other practical switches with the same amount of storage at any rate of traffic. The single exception to this is the SAFC switch with six packet slots per buffer, operating with a 0.99 applied input traffic rate. This result demonstrates that, as the size of the buffer grows, the ability to dynamically allocate buffer space becomes less important. When large buffers make dynamic buffer allocation less relevant, the additional connectivity of the SAFC switch gives it the performance edge over the DAMQ buffer.

It should be noted that a DAMQ switch with space for three packets per input buffer discards as few or fewer packets than the FIFO switch with space for up to six packets for all traffic rates. Further, the DAMQ switch performs *significantly better* than the FIFO switch for high traffic rates. The savings in chip area due to this dramatic decrease in storage requirements is several times greater than the area

Buffer Type	Pkts per Port	Percentage of discarded packets vs. applied input traffic rate							
		0.25	0.50	0.75	0.80	0.85	0.90	0.95	0.99
FIFO	1	1.7	7.1	15.5	17.4	19.3	21.2	23.1	24.6
	2	0^+	1.2	8.7	11.4	14.5	17.8	21.3	24.2
	3	0^+	0.2	6.1	9.2	13.0	17.0	21.0	24.2
	4	0^+	0^+	4.7	8.1	12.3	16.7	21.0	24.2
	5	0^+	0^+	3.8	7.5	12.0	16.7	21.0	24.2
	6	0^+	0^+	3.2	7.1	11.9	16.6	21.0	24.2
SAMQ	2	0.9	4.7	11.3	12.9	14.5	16.1	17.8	19.1
	4	0^+	0.3	3.0	4.2	5.5	7.1	8.9	10.5
	6	0^+	0^+	0.9	1.5	2.4	3.7	5.4	7.1
SAFC	2	0.8	3.8	9.1	10.5	11.9	13.4	15.0	16.3
	4	0^+	0.2	2.0	2.8	3.8	5.1	6.6	8.1
	6	0^+	0^+	0.5	0.9	1.5	2.4	3.8	5.2
DAMQ	2	0^+	0.6	4.8	6.4	8.3	10.5	12.9	15.0
	3	0^+	0^+	1.4	2.4	3.9	5.8	8.3	10.6
	4	0^+	0^+	0.4	0.9	1.8	3.3	5.6	8.1
	5	0^+	0^+	0.1	0.4	0.9	2.0	3.9	6.5
	6	0^+	0^+	0^+	0.1	0.4	1.2	2.8	5.4
CBDA	2	0^+	0^+	1.8	3.0	4.6	6.7	9.3	11.8
	3	0^+	0^+	0.2	0.5	1.2	2.6	4.9	7.5
	4	0^+	0^+	0^+	0.1	0.3	1.1	2.9	5.4
	5	0^+	0^+	0^+	0^+	0.1	0.4	1.8	4.1
	6	0^+	0^+	0^+	0^+	0^+	0.2	1.1	3.3

Table 4.1: *Results of Markov Analysis.* A single 2×2 discarding switch. The percentage of discarded packets for a variety of buffer sizes (given as packets per input port) for various applied input traffic rates (the fraction of the raw input port capacity).

for the extra control circuitry needed for the DAMQ buffer [Fraz89] (Ch. 7).

For light traffic and only two slots per buffer, the FIFO switch performs better

than the SAMQ and SAFC switches. Under these conditions the probability of discarding is determined by available storage and the FIFO buffer, having a single pool of slots instead of statically partitioned storage, delivers better performance. This effect is overshadowed by the unnecessary blocking of packets due to FIFO handling when the traffic rate is high or when there are more than four packet slots per buffer. In general, the SAMQ buffer performs almost as well as the SAFC buffer, indicating that the additional throughput provided by fully connecting the inputs with the outputs does not provide a significant boost in performance for 2×2 switches.

The benefits of non-FIFO access to the buffers are demonstrated by the fact that the FIFO buffers perform significantly worse than the three other buffer types for traffic rates above 80% and a wide range of buffer sizes. Furthermore, the performance of the FIFO buffers cannot be improved by simply increasing the buffer size — for example, with a traffic rate of 90%, increasing the FIFO buffer size from three to six slots does not significantly reduce the discarding rate. This is due to there being a theoretical limit to the throughput obtainable from FIFO buffers [Irla78]. Under uniform traffic conditions, a 2×2 switch implemented with FIFO buffers cannot support throughputs over 0.75 (consider that there are only two packets available for transmission on any given stage cycle, and each packet is equally likely to be destined for either output port). Thus, it is not always possible to trade off implementation and control complexity for increased storage. FIFO switches perform poorly at high traffic rates *regardless of their buffer size*.

We have previously described the benefits of the DAMQ buffer over the SAMQ and SAFC buffers in the areas of implementation, flow control, and

routing. The performance advantage of the DAMQ compared with the SAMQ and SAFC buffers is based on more efficient use of storage. As shown in Tab. 4.1, the DAMQ switch performs significantly better (lower discarding rate) than a SAMQ or SAFC switch with small buffers.

This relatively simple analysis of a single discarding switch has exposed two buffering fundamentals. The first is the impact of the performance restriction imposed by FIFO buffering. Even with a small number of packet slots, switches implemented with multi-queue buffers can exceed the theoretical maximum throughput of switches implemented with FIFO buffers — and the maximum throughput FIFO switches under uniformly distributed traffic decreases as the number of ports increases (0.65542 for a 4×4 FIFO). The second is the importance of dynamic buffer allocation when there are a small number of packet slots per input port.

4.3. The Simulator

We used a general purpose simulation tool written in Modula2 called *SIMON* [Swop86] to construct the simulator. *SIMON* provides facilities for the creation of *objects* (co-routines) and an event queue which allows the objects to pass messages to each other. The simulator uses three types of objects to create a communication network: *senders, switches* and *receivers*. We simulate a clocked system; each object in the system is synchronous, and the clock is global. As explained in Sec. 4.1, the network being synchronous is independent of the fact that the individual objects with the network are synchronous — the asynchronous network simulator described in Sec. 6.1 also uses a global clock.

Network traffic is stochastically generated. Each sender injects packets into the network using geometrically distributed inter-message delays. The destination address of each message is chosen randomly — in Secs. 4.4 and 4.5, a uniform distribution of the possible destination addresses is used. For Sec. 4.6, which explores network performance in the presence of a hot spot, each packet generated has probability h of being addressed to the hot spot, probability $1-h$ of having its address chosen from the uniform distribution. The simulator follows a set structure in performing the simulations. First, the simulations are run until x packets (in total) are received (reach their destinations). This number must be set high enough that, by the end of the x packets, the system is stabilized. At this point, the statistics-gathering variables are reset and the simulations continue to execute until n more packets are received. Finally, the simulation ends and the statistics are printed to a file. This chapter presents the results of simulating 64×64 omega networks [Lawr75] constructed from three stages of 4×4 switches. The network connects sixty-four processors (message generators) to sixty-four memory modules (message receivers).

The next section discusses results obtained using *discarding flow control*, while the two sections after it present the results of simulations of networks which perform *blocking flow control*. Under discarding flow control, decisions as to whether or not to transmit a packet are made without consideration of the state of the destination buffer. If there is not buffer space available to receive an incoming packet, then the packet is discarded. In this simulator, a switch discards a packet by transmitting it back to its original sender. Each sender object has a queue for packets which have been discarded — while a sender can generate a new packet

while this queue is not empty, it cannot transmit a new packet while there are packets in the discarded queue (i.e. senders give priority to packets which have been discarded).

Under blocking flow control, when a buffer becomes filled with packets, it signals this state to the neighboring switch(es) which transmit to that buffer. Senders / switches do not transmit packets to a switch which has asserted the blocking signal. Only after the signal is dropped can transmissions recommence. In the network simulated for this dissertation, if a full buffer transmits a packet on stage cycle i (and thus becomes unfull), the neighboring switch will be able to transmit to the once-full buffer on stage cycle $i+1$.

An important issue in the design of communication switches is the scheme used to arbitrate between multiple packets which require conflicting resources in order to be forwarded. The switches have to arbitrate between multiple buffers with packets destined for the same output port. The SAMQ and DAMQ switches also need to arbitrate between packets from the same buffer which were routed to different output ports (they are multi-queue buffers which can only read queue at a time). The arbitration scheme used in the evaluation involved examining the buffers of a switch one at a time, transmitting packets from the longest unblocked queue in the buffer. To support fairness, the order in which buffers are examined is not fixed — a modified round-robin priority scheme is used. In successive arbitration ‘‘rounds’’ each buffer in turn is the first buffer to be examined. If a non-empty buffer has the top priority but is unable to transmit (for example, due to a full buffer at the destination node), the buffer retains its top priority for the next round. To maintain fairness within the buffers, a stale count is used on the

queues [Stev86] to determine which queues within a buffer have held packets for a long period of time and should therefore get top priority. This is not an arbitration scheme which one could efficiently implement in a communication switch. Tamir and Chi present a detailed discussion of arbitration schemes for small $n \times n$ switches [Tami93].

Network performance is measured by the maximum throughput achieved and by the average packet latency at a given throughput. In a synchronous network, throughput is measured as the fraction of the network bandwidth utilized, which is equal to the average number of packets reaching the destination nodes per stage cycle divided by the number of destination nodes (i.e. $0 \leq thpt \leq 1$). Latency is calculated in *stage cycles* and is measured from the time of a packet's creation until it reaches its destination.

4.4. Evaluation of Discarding Switches via Event-Driven Simulation

Section 4.2 discussed results obtained from an analytical model of a single discarding 2×2 switch. This section extends the evaluation by simulating a network of discarding switches.

The results of simulating a network of discarding switches under uniform traffic are presented in Table 4.2. The table shows the percentage of packets which are discarded for various rates of introducing packets to the network. The applied input traffic rates are expressed as the fraction of the network's raw input port capacity. For each applied input traffic rate R, with a resulting discarding rate D, the throughput of the network is $R \cdot (1-D)$.

The results in Tab. 4.2 show that, for a given buffer size and wide range of

Buffer Type	Pkts per Port	Percentage of Packets Discarded vs. Applied Input Traffic Rate								Max. Thpt.
		0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	
FIFO	1	1.5	5.8	12.1	19.6	27.0	33.9	40.3	45.8	0.45
	2	0^+	0.2	1.5	4.9	11.2	19.6	28.0	35.7	0.52
	3	0	0^+	0.2	1.3	5.2	13.4	22.3	31.1	0.55
	4	0	0^+	0^+	0.4	2.5	10.3	18.6	27.2	0.57
	8	0	0	0	0^+	0.2	5.3	13.6	24.0	0.61
SAMQ	4	0.4	1.9	4.6	8.4	13.2	18.6	23.9	29.1	0.61
	8	0^+	0^+	0.1	0.4	1.2	3.1	6.2	10.5	0.78
SAFC	4	0.4	1.5	3.6	6.4	9.9	14.2	18.6	23.2	0.67
	8	0	0^+	0.1	0.3	0.8	2.0	3.9	6.9	0.84
DAMQ	2	0^+	0.1	0.4	1.8	5.0	10.7	17.3	24.5	0.63
	3	0	0^+	0^+	0.1	0.7	3.0	7.2	13.3	0.72
	4	0	0	0^+	0^+	0.1	0.7	3.9	9.6	0.78
	8	0	0	0	0	0	0^+	0^+	0.7	0.88
CBDA	1	0^+	0.2	1.1	4.4	10.5	18.7	26.8	34.5	0.53
	2	0	0	0	0^+	0.1	1.3	4.7	10.9	0.73
	3	0	0	0	0	0^+	0.1	0.8	3.5	0.82
	4	0	0	0	0	0	0^+	0.1	1.1	0.86
	8	0	0	0	0	0	0	0	0^+	0.93

Table 4.2: *Discarding Percentage vs. Throughput.* The performance of a 64×64 omega network composed of 4×4 discarding switches. Results obtained from simulation. Uniform traffic. Shown is the percentage of packets discarded for various applied input traffic rates (the fraction of the network's raw input port capacity).

applied input traffic rates, the DAMQ switch network has significantly lower discarding rates than networks with FIFO, SAMQ, and SAFC switches. Furthermore, the maximum achievable throughput is significantly higher with the DAMQ switch, for the buffer sizes examined. At low applied input traffic rates, the FIFO switch performed *better* than the SAMQ and SAFC switches, due to the

four-way static partitioning of the buffer space in the SAMQ and SAFC buffers. However, non-FIFO handling of packets allows the switches implemented with SAMQ and SAFC buffers to operate at throughputs above the FIFO switch's theoretical maximum. In all cases, the hypothetical CBDA switch performs better than the other switches, thus demonstrating the value of complete sharing of storage and arbitrary random access to any packet buffered in the switch when dealing with uniformly-distributed traffic.

4.5. Evaluation of Blocking Switches via Event-Driven Simulation

The previous section described the results of simulating networks of discarding switches. This section explores the performance of networks using blocking flow control. The performance of a multistage interconnection network composed of blocking switches can be characterized by the relationship between packet latency and throughput. In general, as throughput increases, so does the latency. For low throughputs (i.e. before the network approaches saturation), latency grows very slowly with increasing throughput. As the throughput approaches saturation, the latency increases rapidly. Near saturation, small increases in throughput are accompanied by large changes in latency. This relationship between latency and throughput is shown in Fig. 4.1 and elsewhere [Dias81, Pfis85a].

We have simulated 64×64 omega networks of blocking switches using all five switch types and several buffer sizes under uniform traffic with a wide range of traffic loads. The results of our simulations are shown in Tab. 4.3. The throughput is reported as the fraction of the aggregate raw link bandwidth. This is the fraction

Buffer Type	Pkts per Port	Average latency vs. network throughput						Sat. Thpt
		0.10	0.20	0.30	0.40	0.50	sat.	
FIFO	1	3.67	5.51	Sat.	Sat.	Sat.	8.89	0.24
	2	3.14	3.39	3.88	5.41	Sat.	7.95	0.44
	4	3.14	3.38	3.79	4.65	9.34	13.14	0.51
	6	3.15	3.34	3.79	4.63	7.78	17.87	0.55
	8	3.14	3.38	3.79	4.60	6.90	23.03	0.57
	12	3.15	3.38	3.79	4.61	6.78	33.00	0.59
SAMQ	4	3.24	3.58	4.09	4.90	6.57	6.68	0.50
	8	3.14	3.36	3.68	4.07	4.95	9.39	0.71
	12	3.15	3.36	3.68	4.16	4.91	13.00	0.78
SAFC	4	3.22	3.50	3.88	4.42	5.28	5.88	0.54
	8	3.13	3.29	3.51	3.80	4.21	7.53	0.75
	12	3.13	3.29	3.50	3.79	4.20	9.80	0.82
DAMQ	2	3.14	3.36	3.74	4.48	Sat.	7.19	0.50
	4	3.14	3.36	3.68	4.16	4.91	10.66	0.71
	6	3.14	3.36	3.68	4.16	4.90	14.85	0.80
	8	3.14	3.36	3.68	4.17	4.89	19.10	0.84
	12	3.14	3.36	3.68	4.16	4.92	29.15	0.90
CBDA	1	3.24	3.53	4.64	Sat.	Sat.	6.63	0.33
	2	3.13	3.30	3.50	3.81	4.35	6.31	0.59
	4	3.13	3.29	3.50	3.80	4.19	9.71	0.80
	6	3.13	3.29	3.51	3.79	4.20	13.84	0.86
	8	3.13	3.29	3.51	3.79	4.20	18.07	0.90
	12	3.13	3.29	3.51	3.79	4.21	26.07	0.94

Table 4.3: *Latency vs. Throughput, Blocking Switches.* The performance of a 64×64 omega network composed of 4×4 blocking switches. Results obtained from simulation. Uniform traffic. Shown are the latencies of packets through the network for various network throughputs.

of the maximum network throughput that would be achieved if all the network links were transmitting at all times (this would be possible if there were no conflicts in any of the switches). The latency is the number of stage cycles from the moment the packet is created to its delivery at the destination. The minimum latency through the omega network is three stage cycles. At each sender, the interval to the next packet creation is calculated from the time the current packet enters the network.

Table 4.3 shows that for low throughput rates all the switches achieve nearly the same latencies. Furthermore, for particular buffer type, as long as the throughput is well below saturation, the buffer size does not have a significant impact on the latency. For the buffer sizes shown in the table, the DAMQ switch network can achieve significantly higher maximum throughput than networks with FIFO, SAMQ, or SAFC switches. For example, with a buffer size of four packet slots per input, the maximum throughput of a network composed of DAMQ switches is at least 30% higher than a network composed of any of the other three practical switches. If one were to increase the buffer sizes, as one approaches infinity, the performance of switches implemented with SAMQ and DAMQ buffers would be equal, and SAFC buffers better; static versus dynamic buffer allocation is irrelevant in extremely large buffers.

As shown in Tab. 4.3 and in Fig. 4.1, for these same switches, at a throughput of 0.50, the DAMQ network results in lower latency than with the other three practical networks. This difference in latency is due to the fact that the other switches are at or near their saturation throughputs. At lower throughputs (below 0.40) the average latencies with all the networks are very close to each other.

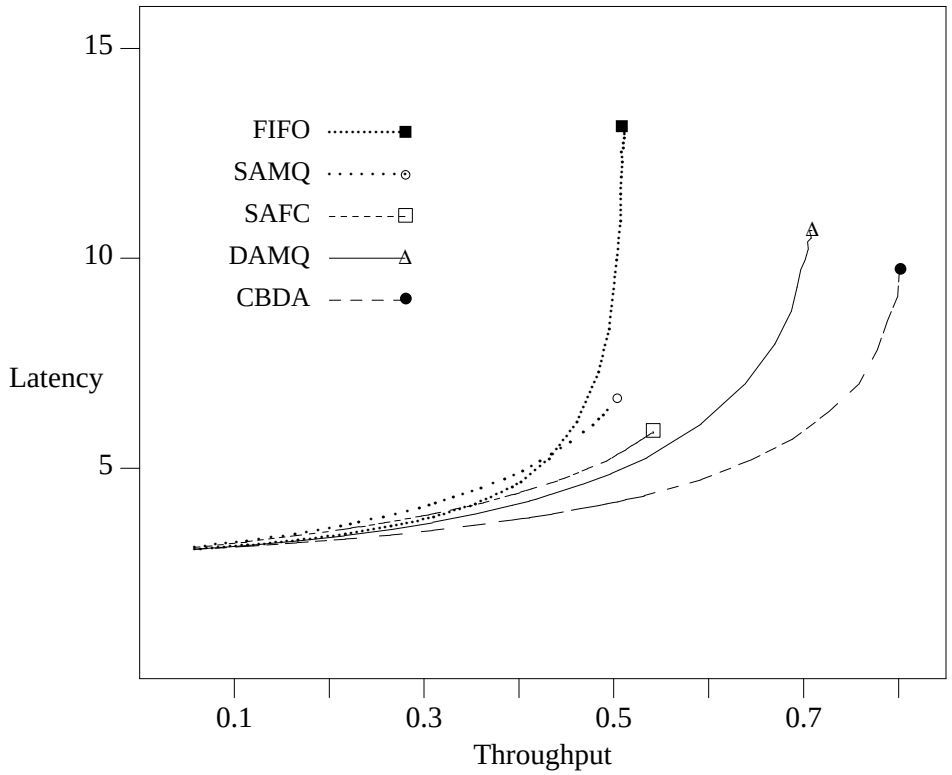

Figure 4.1: *Latency vs. Throughput.* The performance of a 64×64 omega network composed of 4×4 blocking switches with four packet slots per input port buffer. Results obtained from simulation. Uniform traffic.

Hence, the major advantages of the DAMQ buffer are (a) its ability to support high throughput communication and (b) its ability to provide low-latency communication at throughput rates which are approaching saturation.

As shown in Fig. 4.1, the latency of the network of SAMQ and SAFC switches with four packet slots per input buffer does not increase significantly near saturation as it does with the other switches. The reason for this is that with four packet slots per buffer allocated to four queues per buffer, each queue has only one buffer slot. Hence, once a packet enters a switch, it is forwarded the next time its

queue gets priority from the arbiter (if not sooner). With the other buffer types, and with larger SAMQ and SAFC buffers, the packet may be queued behind several other packets waiting for the same output port.

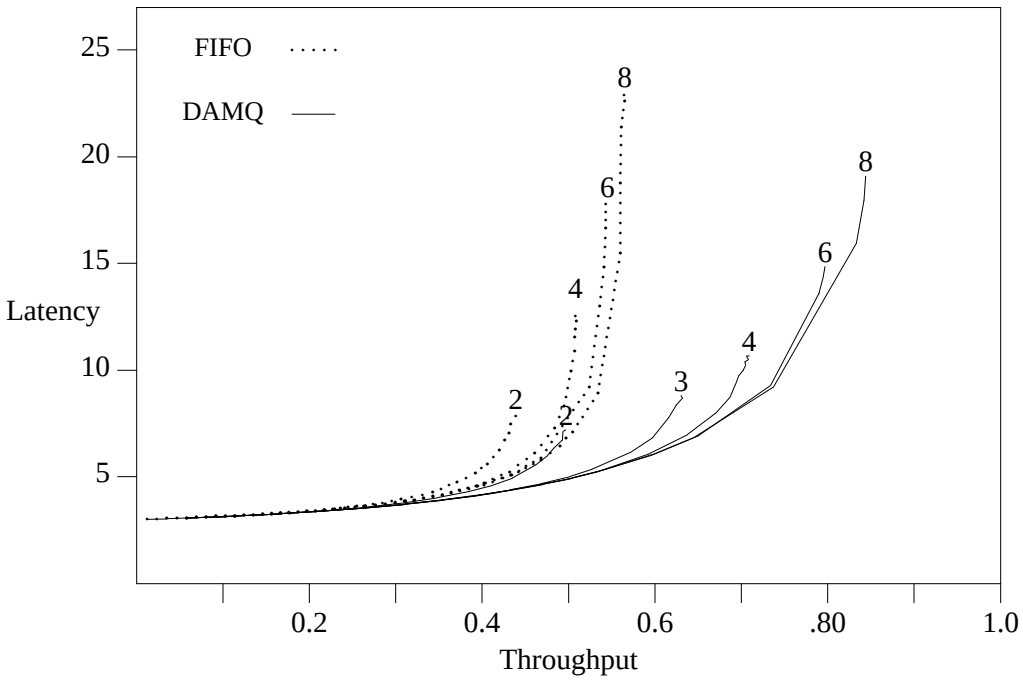

Figure 4.2: *Blocking Switches with Varying Buffer Sizes.* The performance of a 64×64 omega network composed of 4×4 FIFO or DAMQ blocking switches with several sizes of input buffers. Results obtained from simulation. Uniform traffic.

The results in Tab. 4.3 are similar to the results of the Markov model and the simulations of discarding switches — in all cases the benefit derived from non-FIFO handling of packets is the improved network performance under high network loads. At low throughputs, the FIFO switches perform as well as the SAMQ, SAFC, and DAMQ switches. However, multi-queue buffers can provide higher maximum throughput as well as lower latency at ‘‘moderate’’ throughputs.

This problem cannot be resolved by increasing the size of the FIFO buffers — increases beyond three or four slots results in very small increases in the maximum network throughput (as the work of ourselves and others [Dias81] has shown).

Multi-queue buffers, on the other hand, increase the maximum throughput far beyond what is achievable with FIFO buffers. This result is seen in Fig. 4.2, which shows the diminishing returns obtained from increasing the size of FIFO buffers. This figure also shows that implementing DAMQ buffers with only three packets slots results in significantly better performance than a FIFO buffer with eight slots. Hence, beyond minimal buffer space (two or three packet slots), it is more beneficial to allocate hardware resources to the more complex control of the DAMQ buffer than to additional buffer memory.

4.6. Non-Uniform (Hot Spot) Traffic

For some applications, the handling of non-uniform traffic by the network may be a critical factor in determining system performance [Pfis85a]. Of particular interest is ‘‘hot-spot’’ traffic, where a particular (‘‘hot’’) destination receives a higher percentage of the packets than any other destination. Tab. 4.4 presents the results of simulating networks composed of the various switches with hot-spot traffic [Pfis85a]. Five percent of the packets from all senders are sent to the ‘‘hot’’ destination, while the destinations of the rest of the packets are uniformly distributed among all the receivers. The results demonstrate that with hot-spot traffic, the buffer type does not matter. Below saturation, the switches display almost equal latencies, just as with uniform traffic. However, unlike the situation with uniform traffic, the switches all reach saturation at the same throughput

(≈ 0.24).

Buffer Type	Average latency vs. network throughput					Saturation Throughput
	0.05	0.10	0.15	0.20	saturated	
FIFO	3.07	3.17	3.32	3.81	23.58	0.24
SAMQ	3.12	3.27	3.48	3.88	10.92	0.24
SAFC	3.11	3.25	3.43	3.78	10.53	0.24
DAMQ	3.07	3.16	3.30	3.67	25.20	0.24
CBDA	3.10	3.15	3.25	3.55	16.96	0.24

Table 4.4: *Blocking Switches with Hot Spot Traffic.* A 64×64 omega network composed of 4×4 blocking switches. Four packet slots per input port. Five percent hot-spot traffic. Shown are the latencies of packets through the network for various network throughputs. Results obtained from simulation.

With hot-spot traffic (Table 4.4), the saturation throughput for all switches is significantly lower than with uniform traffic (Table 4.3). This is caused by the increased probability of contention within each switch for the output port on the path to the hot-spot. With FIFO buffers, when there is contention for an output port only one packet from one of the contending input ports is forwarded. All the other contending input ports are idle. Thus, within a short period of time, all of the switches which are on the path to the hot-spot have a high probability of having packets destined for the hot-spot at the head of their buffers and of having their buffers completely full. Pfister and Norton [Pfis85a] call this effect ‘‘tree saturation.’’ Since there is a path from every sender (processor) to each receiver (memory bank), when a hot-spot tree saturates, the traffic backs up to block every single sender. For all buffer types the network saturates at a throughput t that satisfies the equation $t(1-h) + thp = 1$, where h is the fraction of packets destined for the hot-spot and p is the number of inputs/outputs of the network [Pfis85a].

Tree saturation occurs with DAMQ switches just as with FIFO switches. The DAMQ switches forward all of the non hot-spot traffic, but cannot forward hot-spot traffic since the bottleneck is bandwidth of the single link connecting the network to the hot destination. This causes the buffers to fill up with hot-spot traffic. Once that happens, each DAMQ buffer is dominated by the queue to the output port on the path to the hot-spot and the network becomes saturated just like a network with FIFO switches. With the SAMQ and SAFC switches buffer space cannot become completely occupied by hot-spot traffic since it is statically partitioned. However, the blocked hot-spot traffic at the inputs to the network quickly block all non hot-spot traffic attempting to enter the network, thus leading to saturation at the same levels as the other switches.

Since the buffer organization does not help in reducing the impact of tree saturation, other solutions must be found. One possibility is to follow the design of the IBM RP3 multiprocessor [Pfis85b] and use two separate networks: one for general traffic and the second, a combining network [Gott83], for hot-spot traffic caused by synchronization traffic, such as accesses to semaphores. In a system such as this, the hot-spot traffic would not interfere with the uniform memory accesses, so significant performance gains would be made by using the DAMQ buffer instead of the FIFO buffer in the general traffic network. If the hot-spot traffic originates from only a subset of the nodes, an alternative scheme for limiting tree saturation is the use of feedback flow control to ‘‘throttle’’ the hot-spot traffic and the source [Scot90, Dias89]. In Ch. 8, we discuss the possibility of a flow control scheme to minimize the impact on the rest of the communication traffic in the network caused by hot spots and other forms of congestion.

4.7. Summary and Conclusions

We have evaluated the DAMQ buffer by comparing its performance with that of three alternative practical buffers in the context of a synchronous store-and-forward multistage interconnection network. Both discarding and blocking switches were considered. The DAMQ buffer provides two key features: non-FIFO handling of packets and dynamic partitioning of buffer storage. We have shown that both of these features are critical for supporting high throughput, low latency communication.

Also, we examined buffer performance in communication networks experiencing hot spot contention. We have shown that with hot-spot traffic that involves all the senders in the network, multi-queue buffering and dynamic buffer allocation do not improve performance, since they do not prevent tree saturation. For uniform traffic, our modeling and simulations show that these features result in large performance improvements over conventional FIFO buffers and/or static storage partitioning. The DAMQ buffer provides significantly lower latencies and higher maximum throughput than other practical buffer organizations with the same total buffer storage capacity.