CHAPTER 11

CROSSPOINT BUFFERED SWITCHES

The performance of packet switches are constrained by memory speed and the efficiency of the arbitration scheme used to select the cells that will traverse the switch at a given time. Output buffered (OB) switches use no arbitration at the inputs as the output buffers run at N times faster than the speed of external links. Therefore, for a limited memory speed, the size of such switches is restricted to a small number of ports as link rates increase. On the other hand, the memory in input buffered (IB) switches runs at link-rate speeds, making these switches attractive. However, IB switches need to perform matching between input and outputs to resolve input and output contentions. Fast matching schemes for IB switches can be modeled as parallel matchings, where N input and output arbiters perform the parallel selection and communication among them to decide the matching results. This communication adds overhead time to the matching process. The parallel matching process can be characterized by three phases: request, grant, and accept. Therefore, the resolution time would be the time spent in each of the selection phases plus the transmission delays for the exchange of request, grant, and accept information. As an example, a matching scheme must perform input or output arbitration within 6.4 ns in an IB switch with 40 Gbps (OC-768) ports and 64-byte cells, as input and output arbitrations may use up to half of a time slot, assuming that the transmission delays are decreased to a negligible value (e.g., the arbiters are implemented in a single chip, in a centralized way).

A solution to minimize the time overhead is to use buffers in the crosspoints of a crossbar fabric, or buffered crossbar. In a buffered crossbar switch, an input can avoid waiting for a matching to be completed before a cell is forwarded to the crossbar. However, the number of buffers grows in the same order as the number of crosspoints, $O(N^2)$, making implementation costly or infeasible for a large buffer size or a large number of ports. As VLSI technology has matured, buffered crossbars are considered feasible for a moderate size switch.

High Performance Switches and Routers, by H. Jonathan Chao and Bin Liu

The rest of this chapter is organized as follows. Section 11.1 introduces the basic combined input and crosspoint buffered (CIXB) switch structure. CIXB switches with virtual output queues (VOQs) at the input ports are discussed in Section 11.2. Sections 11.2.1 to 11.5 describe some variants of CIXB, namely CIXB-*k* and CIXOB-*k*, and several scheduling schemes employed in these switches. More work related to crosspoint buffered switches can be found in References [1, 2].

11.1 COMBINED INPUT AND CROSSPOINT BUFFERED SWITCHES

A conventional crosspoint buffered switch is shown in Figure 11.1. It consists of a crossbar $N \times N$ switch fabric and buffers that are placed at each crosspoint. This switch and the memory operate at the line speed. When a cell arrives from an input port, it will enter the designated crosspoint buffer and wait to be scheduled to the output link. Although it has high throughput performance equivalent to that of output buffered switches, it is not feasible in implementation. The required number of buffers grows quadratically with N. Since crosspoint buffers are not shared among them, the total required buffer size is large, making it difficult to include the buffers on a chip.

To reduce the memory in the crosspoint buffer, CIXB switches were proposed [3], where large input buffers and small crosspoint buffers were adopted. The CIXB switch operates at the line rate, and its architecture is shown in Figure 11.2. When a cell arrives, it is first buffered at the input buffer, and transferred to the crosspoint buffer when it has a space. The number of crosspoint buffers and input buffers are N^2 and N, respectively. Different from the conventional crosspoint buffered switch, CIXB switches only need a small buffer at crosspoint. Thus, the total buffer size of the CIXB switch is much smaller than that of the conventional crosspoint buffered switch for the same switching performance.

Simulations show that the CIXB switch has superior traffic characteristics to the conventional crosspoint buffered switches. In the simulation, the switch size is assumed to be 32 and the input offered load is set to 0.8. Input buffer size is set to 8, 16, and 24

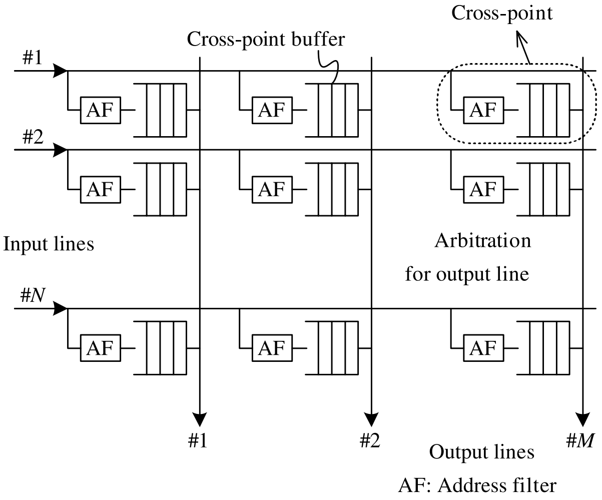

Figure 11.1 Conventional crosspoint buffered switch.

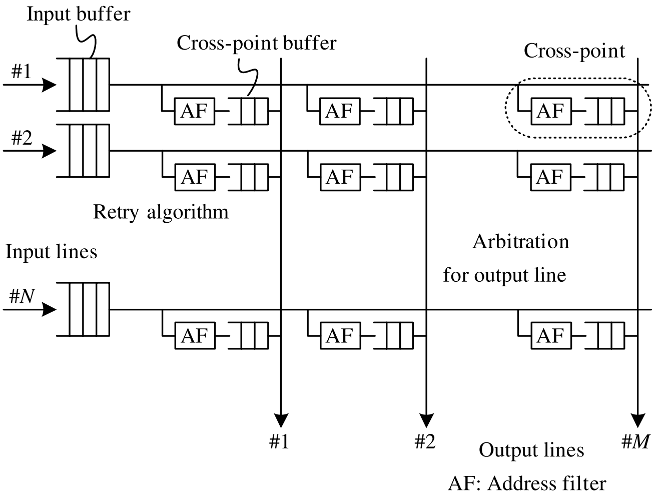

Figure 11.2 Combined input and crosspoint buffered (CIXB) switch with FIFO queuing at inputs.

to 32 cells, respectively. A simple ring arbitration mechanism (i.e., round robin) is used to select cells from the crosspoint buffers on the same column. The simulation results in Figure 11.3 show the cell loss rates as a function of crosspoint buffer size. To achieve a cell loss rate of less than 10^{-8}, the conventional crosspoint buffered switch needs 8-cell buffers at each crosspoint, and the CIXB switch needs only 4-cell buffers if 32-cell input buffers are used.

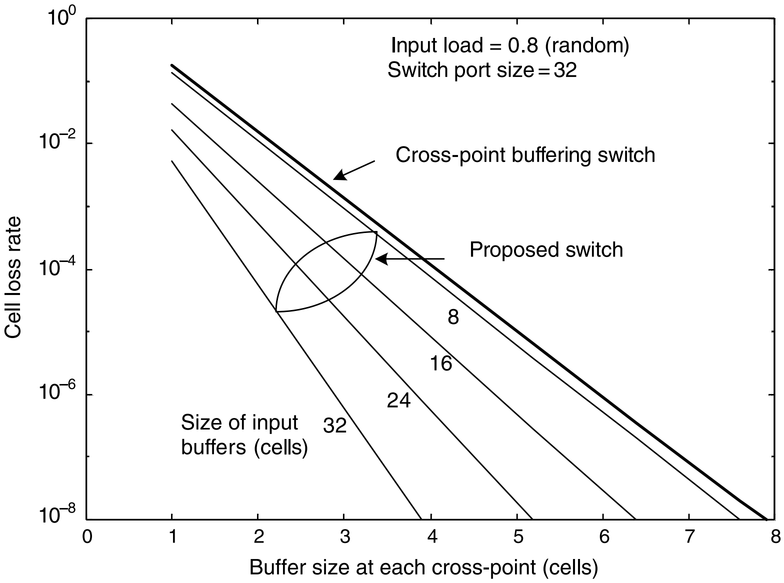

Figure 11.3 Cell loss rates as a function of crosspoint buffer size.

11.2 COMBINED INPUT AND CROSSPOINT BUFFERED SWITCHES WITH VOQ

Although the CIXB switch can have a very high throughput performance, it suffers from head-of-line (HOL) blocking, and thus does not achieve 100 percent throughput. To overcome the throughput degradation due to the HOL blocking, a CIXB switch with virtual output queuing (VOQ) at the input ports was proposed in [4]. From now on, we will refer to a CIXB switch in which VOQ is used at each input port simply as a CIXB/VOQ or just CIXB switch.

Figure 11.4 shows the architecture of a CIXB switch. A crosspoint (XP) element in the crossbar (BX) that connects input port i $(0 \leq i \leq N-1)$ to output port j $(0 \leq j \leq N-1)$ is denoted as $XP_{i,j}$. The switch has the following major components:

Input Buffers. There are N VOQs at each input. A VOQ at input i that stores cells for output j is denoted as $VOQ_{i,j}$.

Crosspoint Buffer (*XPB*). The XPB of $XP_{i,j}$ is denoted as $XPB_{i,j}$. The size of $XPB_{i,j}$ is k cells. We call a CIXB switch with an XPB size k, CIXB-k.

Flow Control. A credit-based flow-control mechanism indicates input i whether $XPB_{i,j}$ has room available for a cell or not. Each VOQ has a credit counter, where the maximum count is the number of cells that $XPB_{i,j}$ can hold. When the number of cells sent by $VOQ_{i,j}$ reaches the maximum count, $VOQ_{i,j}$ is considered not eligible for input arbitration and overflow on $XPB_{i,j}$ is avoided. The count is increased by one every time a cell is sent to $XPB_{i,j}$, and decreased by one every time that $XPB_{i,j}$ forwards a cell to output j. If $XPB_{i,j}$ has room for, at least, one cell, then $VOQ_{i,j}$ is considered eligible by the input arbiter.

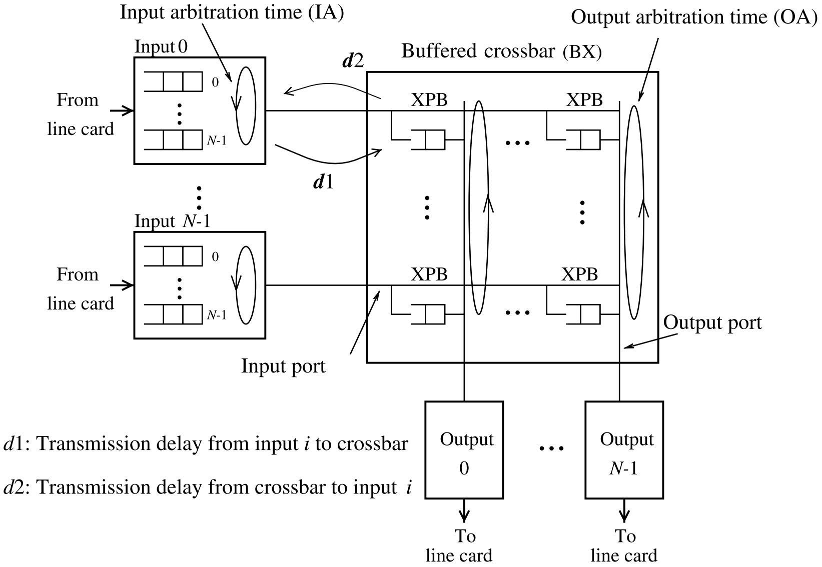

Figure 11.4 CIXB switch model with VOQ at inputs.

Input and Output Arbitration. Round-robin arbitration is used at the input and output ports. An input arbiter selects an eligible $VOQ_{i,j}$ for sending a cell to $XPB(i,j)$. VOQ elegibility is determined by the flow-control mechanism. An output arbiter selects a non-empty $XPB_{i,j}$ for forwarding. Input and output arbitrations are performed separately, therefore, reducing cell selection complexity.

11.2.1 CIXB with One-Cell Crosspoint Buffers (CIXB-1)

Let us assume CIXB-1 has the following traffic model: All inputs have uniform traffic where each $VOQ_{i,j}$ receives cells at a rate $\rho_{i,j}$ such that $\sum_{i=0}^{N-1} \rho_{i,j} = 1.0$, $\sum_{j=0}^{N-1} \rho_{i,j} = 1.0$, and each $VOQ_{i,j}$ has a capacity $\sigma_{i,j}(t)$ such that $0 < \sigma_{i,j}(t) < \infty$ and $\sigma_{i,j}(t)$ is large. CIXB-1 reaches a state such that the number of cells entering the system equals the number of cells leaving it, defining 100 percent throughput. CIXB-1 has the following property:

Property 1. CIXB-1 achieves 100 percent throughput under uniform traffic [5].

For this property to be effective, we need to consider the round-trip time. Round-trip time (RT) is defined as the sum of the delays of the input arbitration (IA), the transmission of a cell from an input port to the crossbar ($d1$), the output arbitration (OA), and the transmission of the flow-control information back from the crossbar to the input port ($d2$). Figure 11.4 shows an example of RT for input 0 by showing the transmission delays for $d1$ and $d2$, and arbitration times, IA and OA. Cell and data alignments are included in the transmission times. The condition for CIXB-1 to provide 100 percent throughput is such that, in general:

$$RT = d1 + OA + d2 + IA \leq k. \tag{11.1}$$

In other words, RT time slots must be absorbed by the number of available cells in the XPB. The arbitration times IA and OA are constrained to $IA \leq 1$ and $OA \leq 1$.

As the cost of implementing memory in a chip is still considerable, although feasible with currently available VLSI technologies, it is important to minimize the XPB size within the implementation limits. A back-pressure flow control is expensive to consider with CIXB as the XPB size needs to be at least twice RT ($k \geq 2RT$) to avoid underflow and, therefore, performance degradation. To evaluate this, consider the worst-case scenario: there is only a single flow in BX from input i to output j. In this case, $XPB_{i,j}$ must have room available for all cells that can be transmitted to their outputs while the notification of back-pressure release travels back to input i and a cell travels from input i to $XPB_{i,j}$ (RT time), so that output port j can continue forwarding cells from input port i. This is why the use of a credit-based flow control is more cost-effective in CIXB.

11.2.2 Throughput and Delay Performance

The performance of an IB switch using *i*SLIP scheduling, two combined input-crosspoint buffered switches, and an OB switch are studied by computer simulation. One of the CIXB switches is the CIXB-1 and the other uses a pre-determined permutation for input and output selections, where the permutation changes cyclically and in a fixed sequence, as time division multiplexing (TDM) works. TDM also uses one-cell XPB. *i*SLIP is used as the scheduling scheme in the IB switch, as *i*SLIP is based on round-robin selection. A comparison on the performance between IB and CIXB switches, where round-robin selection is used, is given. The comparison with the OB switch is used for the best possible performance. The traffic patterns studied in this section are uniform and non-uniform traffic with Bernoulli arrivals. Traffic with uniform distribution with bursty arrivals is also considered. Here, segmentation and re-assembly delays are not considered. Our simulation results are obtained with a 95 percent confidence interval.

Uniform Traffic. The performance of various switches is compared here, including IB switches with 1SLIP and 4SLIP, CIXB-1, TDM, and OB switches under uniform traffic with Bernoulli and bursty arrivals (on–off modulated Markov process) for a switch size of 32 × 32. Note that 4 is almost the optimum number of iterations for *i*SLIP, as more iterations offer no significant measurable improvements [6]. Figure 11.5 shows that, under this traffic model, all switches provide 100 percent throughput. CIXB-1 provides an average delay close to OB. 1SLIP and TDM have similar average delay for high loads, and CIXB-1 delivers shorter average cell delay than 4SLIP and TDM. The reason why CIXB delivers this high performance is because at a given time slot, there could be more than one cell from input *i* stored in crosspoints for different outputs that depart at the same time. Therefore, several cells from input *i* could leave the switch in one time slot. In an IB switch, the number of cells that can traverse to an output port is one.

This small delay of CIXB-1 remains independent of the switch size according to results in the work of Rojas-Cessa et al. [7]. A low average cell delay is expected as it is similar to that of an OB switch.

Figure 11.6 shows the average delay performance of TDM, CIXB-1, and OB under on–off arrival uniform traffic. The traffic in this figure has an average burst length (*l*) of 1, 10, and 100 cells, where the burst length is exponentially distributed. The figure shows that CIXB-1 follows the throughput and average cell delay of an OB switch. TDM has a significantly larger delay than CIXB-1. Some differences can be noted between CIXB-1 and OB when the input load is close to 1.0.

Nonuniform Traffic. *i*SLIP, TDM, and CIXB switches were evaluated under three traffic models with Bernoulli arrivals and nonuniform distributions: unbalanced [7], asymmetric [8], and Chang's models [9].

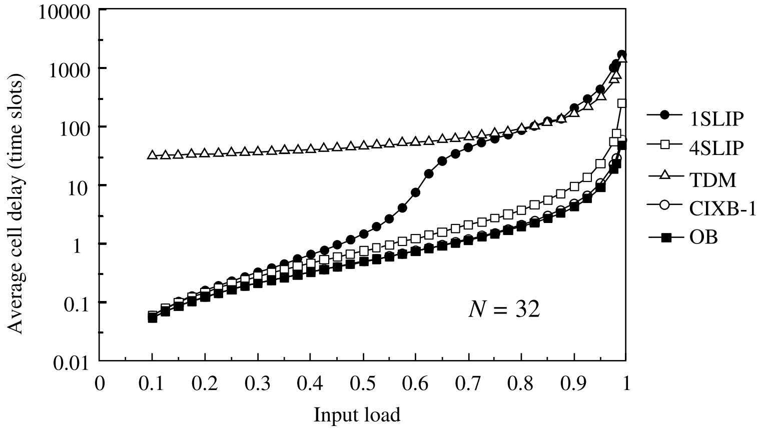

Figure 11.5 Performance of CIXB, TDM, and OB under uniform traffic.

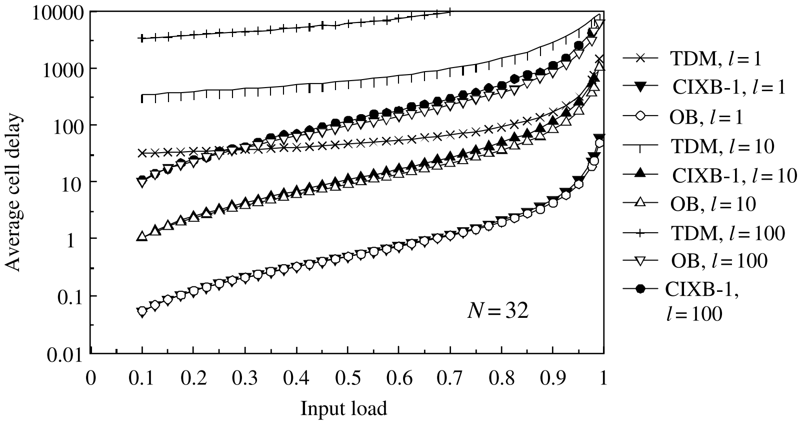

Figure 11.6 Performance of CIXB, TDM, and OB under bursty uniform traffic.

Unbalanced Traffic. The unbalanced traffic model uses a probability w as the fraction of input load directed to a single predetermined output, while the remaining load is directed to all outputs with a uniform distribution. Let us consider input port s, output port d, and the offered input load ρ for each input port. The traffic load from input port s to output port d, $\rho_{s,d}$ is given by $\rho_{s,d} = \rho\{w + [(1 - w)/N]\}$ if $s = d$, and $\rho_{s,d} = \rho[(1 - w)/N]$ otherwise, where N is the switch size. Here, the aggregate offered load ρ_d that goes to output d from all input ports is given by $\rho_d = \sum_s \rho_{s,d} = \rho\{w + N \times [(1 - w)/N]\} = \rho$. When $w = 0$, the offered traffic is uniform. On the other hand, when $w = 1$, the traffic is completely directional. This means that all the traffic of input port s is destined for output port d only, where $s = d$. Figure 11.7 shows *i*SLIP, CIXB-1, and TDM under unbalanced traffic for $0 \leq w \leq 1$, and observed the minimum throughput in terms of w achieved by each switch. The throughput of *i*SLIP with 1 and 4 iterations are 64 percent and 80 percent, respectively. The throughput of CIXB-1 is 84 percent, higher than *i*SLIP. The throughput for TDM drops to almost 0 when $w = 1.0$, because of the predetermined connectivity that TDM provides, independently of the existing traffic. CIXB provides a higher throughput

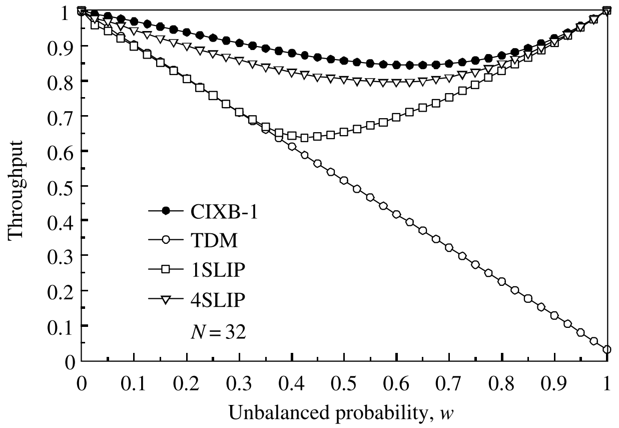

Figure 11.7 Througput of *i*SLIP, TDM, and CIXB-1 under unbalanced traffic.

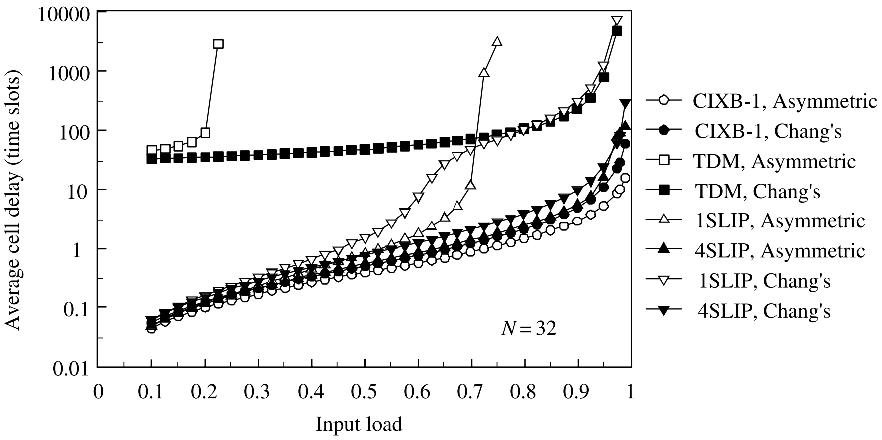

Figure 11.8 Delay performance of TDM and CIXB-1 under asymmetric and Chang's traffic models.

than IB switches with round-robin schemes [7] at the expense of having buffers in the crosspoints. These results clearly show the advantage of using a round-robin scheme and crosspoint buffers under unbalanced traffic.

Chang's and Asymmetric Traffic. Chang's traffic model can be defined as $\rho = 0$ for $i = j$ and $\rho = 1/(N-1)$, otherwise. The asymmetric traffic model can be defined as having different load for each input–output pair, such as $\rho_{i,(i+j) \bmod N} = a_j\rho$, where $a_0 = 0$, $a_1 = (r-1)/(r^N - 1)$, $a_j = a_1 r^{j-1}$ $\forall j \neq 0$, and $\rho_{i,j}/\rho_{(i+1) \bmod N}$, $j = r$, $\forall i \neq 0$, $(i+1)$ mod $N \neq 0$, and $r = (100:1)^{-1/(N-2)}$. Figure 11.8 shows the average cell delay of *i*SLIP, TDM, and CIXB-1 under asymmetric and Chang's traffic models. *i*SLIP delivers close to 71 percent throughput under asymmetric traffic and close to 95 percent throughput under Chang's traffic. 4SLIP delivers close to 100 percent throughput under these two traffic patterns. TDM provides 20 percent throughput under asymmetric traffic, and close to 95 percent throughput under Chang's traffic. CIXB delivers 100 percent throughput and lower average cell delay than the other switches under both traffic models. These results show that having buffers in the crosspoints and round-robin arbitration improves the switching performance.

Buffer and Switch Size Effect on CIXB-k. A 32 × 32 CIXB-*k* for $k = \{1, 2, 8\}$ under uniform traffic was evaluated for the average delay performance and tail delay distribution. Because the similarity of the average cell delay in CIXB-1 and in OB switches, the value of *k* in CIXB-*k* produces non-measurable differences on the average delay performance under Bernoulli and bursty ($l = 10$) uniform traffic, as shown previously [7].

A more stringent test is performed by measuring the tail delay distribution of CIXB-*k*. Figure 11.9 shows that the tail delay distribution under Bernoulli uniform traffic, for different input loads, presents non-measurable differences for any *k* value. This is because as long as the *k* value complies with (11.1), the crosspoint buffers avoid underflow states. By providing a larger *k*, the number of cells leaving BX will not be larger as the optimum performance of the switch has already been reached. Therefore, the crosspoint buffer size can be kept as small as possible to minimize the amount of in-chip memory without having performance degradation.

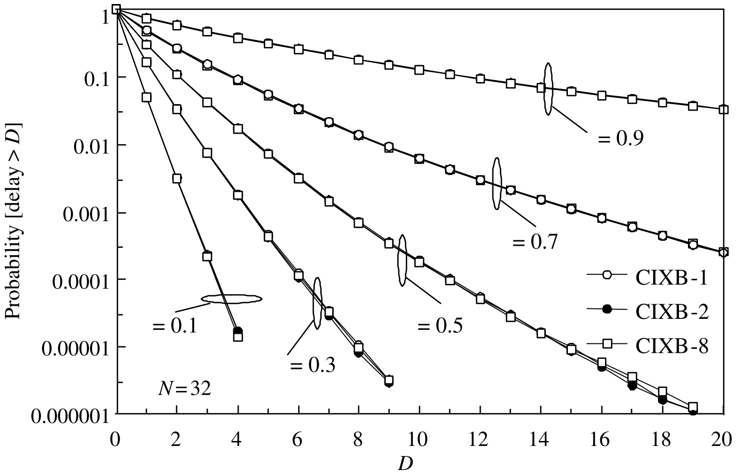

Figure 11.9 Tail delay distribution for different input loads and crosspoint buffer size k under uniform traffic.

The effect of k under non-uniform traffic is studied in a 32 × 32 CIXB-k, with k values between 1 and 32. Figure 11.10 shows the effect of k in CIXB under unbalanced traffic. This figure shows that the throughput of CIXB under non-uniform traffic is slightly improved when a larger k is provided. When $k = N = 32$, the throughput barely reaches 99 percent. These results indicate that to achieve 100 percent throughput under this traffic pattern, a very large k is needed. To provide a higher throughput under unbalanced traffic, a weighted round-robin scheme [6, 10] can be used. The weight of an input or output can be assigned by considering the queue occupancy or cell ages.

According to previous results, the high performance of CIXB is also independent of the switch size [7]. This result differs from the one that has been observed in IB switches.

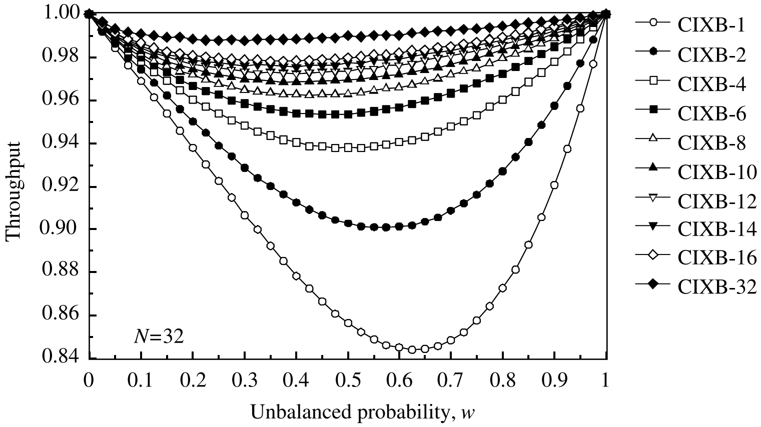

Figure 11.10 Effect of crosspoint buffer size under unbalanced traffic for CIXB-k.

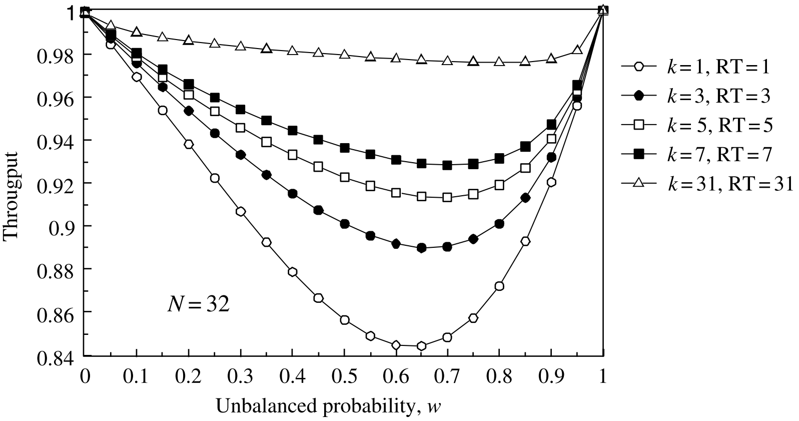

Figure 11.11 Effect of buffer size and RT under unbalanced traffic in CIXB-($k - RT$).

11.2.3 Non-Negligible Round-Trip Times in CIXB-*k*

In the cases above, we assumed small round-trip times, such that (11.1) is satisfied with $k = 1$. Although the consideration of $k = 1$ is practical, the distance between the buffered crossbar and the port cards may be longer than one time slot. Therefore, k needs to be increased to comply with (11.1).

In general, CIXB can be denoted as CIXB-($k - RT$), where $k \geq RT$, for non-negligible RT values (i.e., $RT \geq 1$). We call this the general CIXB model. To observe the performance of a CIXB-($k - RT$) switch, we simulated a switch with different k and RT values for $k - RT = 0$ under unbalanced traffic. Figure 11.11 shows the throughput performance of CIXB-($k - RT$). The throughput performance is improved by larger k values and slightly decreased by larger RT.

11.3 OCF_OCF: OLDEST CELL FIRST SCHEDULING

The scheduling algorithms for CIXB/VOQ switches used in [4] are OCF_OCF (oldest cell first), which selects a cell based on delay time. The algorithms that select a cell based on delay time have better cell delay time performance than the algorithms based on the simple ring arbitration [11].

Output Scheduling. The scheduler at output port j selects the HOL cell with largest delay time. If more than one buffer is selected, then the scheduler at output port j selects, from among these buffers, the buffer that queues a cell from the input port whose identifier is the smallest. Then the HOL cell at the selected buffer is transmitted to output port j.

Input Scheduling. The scheduler at input port i selects the logical queue that has the cells and the smallest queue length of the crosspoint buffer. If more than one logical queue is selected, then the logical queue in which the HOL cell has the longest delay time will be selected. The HOL cell of the selected queue is transmitted to its crosspoint buffer.

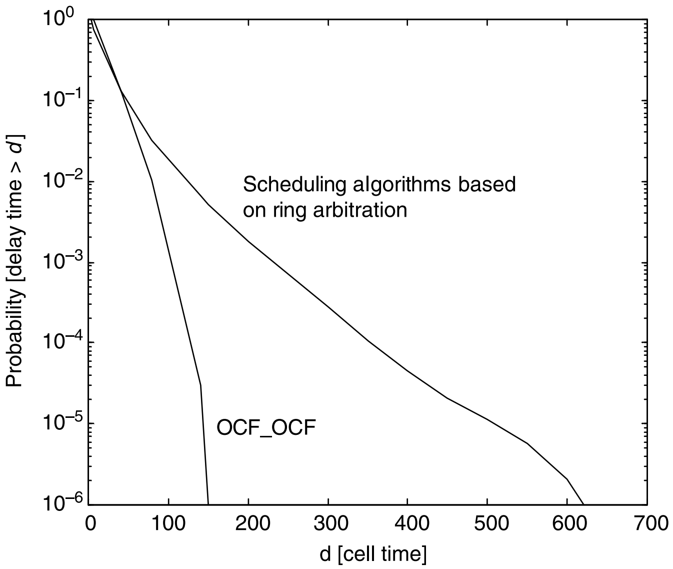

Figure 11.12 Probability of delay time in different scheduling algorithms.

To evaluate the performance of OCF_OCF, several simulations are performed. The traffic model used in simulations is assumed to be uniform, independent, identically distributed (i.i.d.).

The OCF_OCF is compared with a scheduling algorithm based on the ring arbitration. Figure 11.12 shows the probability of the delay time exceeding a certain time d. We assume that the offered load $\rho = 0.97$ and the buffer size at each crosspoint is one. It can be seen that the CIXB switch using the OCF_OCF has better cell delay time performance.

A CIXB switch using the OCF_OCF is compared with an input queued switch using the i-OCF algorithm [12] in terms of mean cell delay time in another simulation. With $i = 32$, it means the scheduling algorithm tries up to 32 times to find a conflict-free match between input and output ports. The buffer size at each input port in the CIXB switch and the input queued switch is infinite. The simulation results are shown in Figure 11.13, where CIXB_M1 and CIXB_M2 mean that the buffer size at each crosspoint in the CIXB switch is either one or two. A cell arriving at the CIXB switch must go through a buffer at an input port and a buffer at a crosspoint. The minimum cell delay time for the CIXB switch is two and that for the input-queued switch is one. From the figure, we can see that the mean cell delay time for the CIXB switch is larger than that for the input-queued switch at $\rho \leq 0.7$, and smaller when $\rho \geq 0.7$. This is because throughput of the CIXB switch is larger than that of the input-queued switch. It can also been seen that CIXB_M2 performs better than CIXB_M1.

A CIXB switch using OCF_OCF is also compared with a conventional crosspoint switch in terms of required buffer size. Here, it is assumed that the OCF algorithm is used as the scheduling algorithm for the crosspoint queued switch. Figure 11.14 shows the required buffer size per port for the CIXB switch and the conventional crosspoint queued switch with a cell loss ratio of under 10^{-9}. It can been seen that the required buffer size for the CIXB switch is much less than that for the crosspoint queued switch, because the CIXB switch allows the input buffer to be shared by multiple output ports.

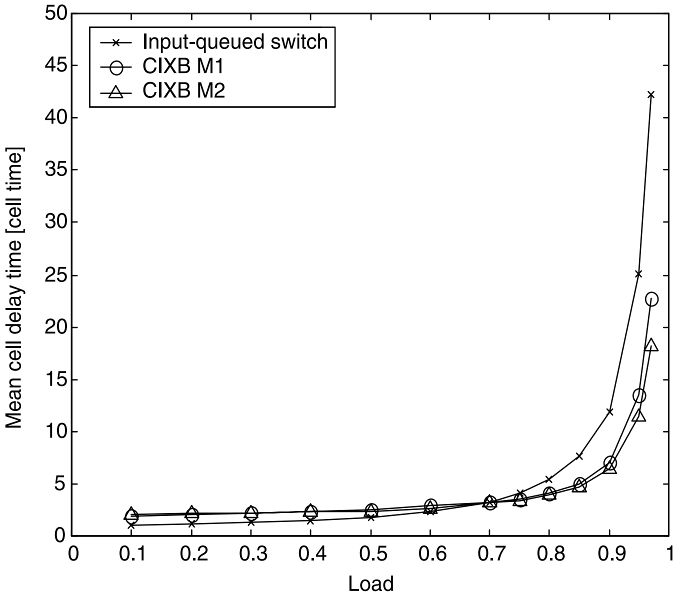

Figure 11.13 Mean delay time: CIXB versus input-queued switch.

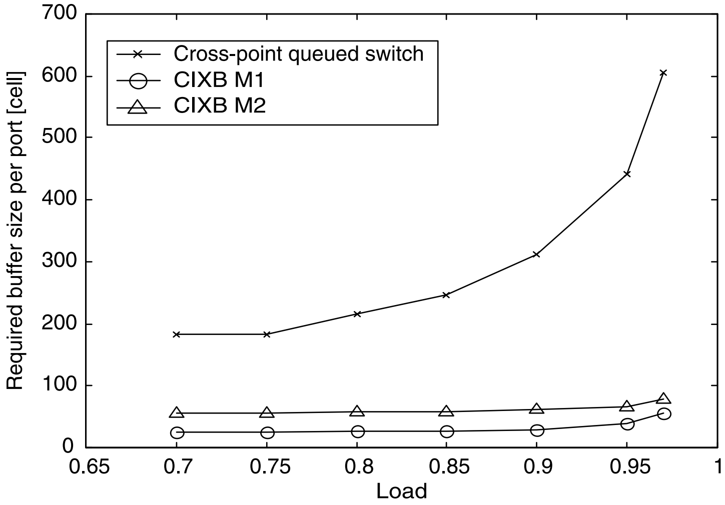

Figure 11.14 Buffer size requirements.

11.4 LQF_RR: LONGEST QUEUE FIRST AND ROUND-ROBIN SCHEDULING IN CIXB-1

An $N \times N$ CIXB switch fabric with one buffer per crosspoint is cost-effective because the required memory speed is only twice the line rate for both the inputs and for the N^2 crosspoints. A scheduling algorithm in such a CIXB-1 switch, which uses longest queue first (LQF) scheduling for VOQs at the inputs and round-robin (RR) scheduling for the crosspoints, was presented by Javidi et al. [13]. Through fluid model techniques, it was shown that LQF_RR scheduling achieves 100 percent throughput for input traffic that satisfies the strong law of large numbers and has a load $\leq 1/N$ for any input/output pair.

Some definitions and notations are given below:

- $Z_{i,j}(n)$: The number of packets in $VOQ_{i,j}$ at the beginning of time slot n.
- $A_{i,j}(n)$: The cumulative number of packets that have arrived at $VOQ_{i,j}$ by time n.
- $D_{i,j}(n)$: The cumulative number of packets that have departed from $VOQ_{i,j}$ by time n.

Assume that the arrivals follow the strong law of large number; that is, with probability one,

$$\lim_{n\to\infty} \frac{A_{i,j}(n)}{n} = \lambda_{i,j} \qquad i, j = 1, \ldots, N, \tag{11.2}$$

where $\lambda_{i,j}$ is the arrival rate at $VOQ_{i,j}$.

By definition, a pair of input and output service algorithms is called "rate stable" if with probability one,

$$\lim_{n\to\infty} \frac{D_{i,j}(n)}{n} = \lambda_{i,j} \qquad i, j = 1, \ldots, N. \tag{11.3}$$

In [14], it has been proved that a switch operating under rate-stable service achieves 100 percent throughput.

Theorem 1. If $\lambda_{i,j} < 1/N$, a CIXB switch using the LQF_RR algorithm is rate stable.

Theorem 1 is proved using a fluid model for the switch. Its proof can be found by referring to the work of Javidi et al. [13].

11.5 MCBF: MOST CRITICAL BUFFER FIRST SCHEDULING

Up until to now, we have introduced several scheduling schemes for the CIXB switch, such as round-robin (RR_RR), OCF_OCF, and LQF_RR. For the input scheduling schemes such as LQF and OCF, the arbiter decisions are very time consuming due to the need for comparing a large number of values (i.e., queue length or cell age). In an attempt to reduce the arbitration complexity while keeping good performance, a scheduling scheme based on the shortest internal buffer first (SBF) at the input side and based on the longest internal buffer first (LBF) at the output side was proposed [15]. Most critical buffer first (MCBF) does not use any input state information, such as VOQ occupancies or VOQ head-of-line cells waiting time. The scheduling is based only on the internal buffers information.

Some notations are defined here to explain the MCBF scheduling.

- *Eligible VOQ* (*EVOQ*): A $VOQ_{i,j}$ is said to be eligible (denoted $EVOQ_{i,j}$) for being scheduled in the input scheduling process if it is not empty and the internal buffer $XPB_{i,j}$ is not full.

- $LXPB_i$ (the line of crosspoint buffers): The set of all the internal buffers $XPB_{i,j}$ that correspond to the same input i.
- NLB_i: The number of cells held in $LXPB_i$.
- $CXPB_j$ (the column of the crosspoint buffers): The set of internal buffers $XPB_{i,j}$ that correspond to the same output j.
- NCB_j: The number of cells held in $CXPB_j$.

The operation of the MCBF is as follows:

Input-Scheduling (*SBF*). For each input i: starting from the highest priority pointer's location, select the first EVOQ corresponding to $\min_j\{NCB_j\}$ and send its HOL cell to the internal buffer $XPB_{i,j}$. Move the highest priority pointer to the location $j+1 \pmod N$.

Output-Scheduling (*LBF*). For each output j: starting from the highest priority pointer's location, select the first $XPB_{i,j}$ corresponding to $\max_i\{NLB_i\}$ and send its HOL cell to the output. Move the highest priority pointer to the location $i+1 \pmod N$.

The MCBF scheme has two properties different from LQF and OCF schemes:

1. MCBF is almost stateless. It does its arbitration without any state information of the input VOQs.
2. MCBF maintains a load balancing among the internal buffers, that is, it keeps the outputs as busy as possible (work conserving).

The performance evaluation of MCBF, LQF_RR, and OCF_OCF were compared through two traffic models, bursty uniform and Bernoulli non-uniform, in a 32 × 32 CIXB switch.

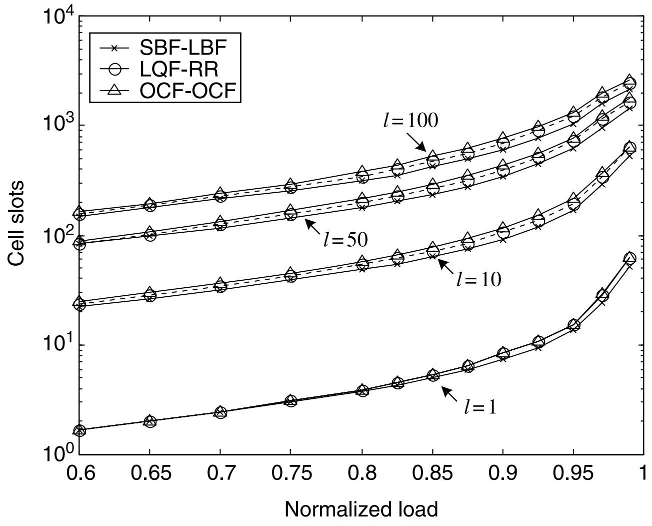

Figure 11.15 Average delay under bursty uniform traffic.

Figure 11.15 shows the average delay performance under bursty uniform traffic with average burst length (l) of 1, 10, 50, and 100 cells, respectively. Under heavy load, SBF_LBF has the smallest delay amongst all the schemes. At 99 percent load and burst length of 10, SBF_LBF has an average delay less than 80 percent that of LQF_RR. With an average burst length of 50 and at 99 percent load SBF_LBF has an average delay of 2523, while LQF_RR has 3014 and OCF_OCF has 3311.

REFERENCES

[1] S.-T. Chuang, S. Iyer, and N. Mckeown, "Practical algorithm for performance guarantees in buffered crossbars," in *Proc. IEEE INFOCOM'05*, Miami, Florida, vol. 2, pp. 981–991 (Mar. 2005).

[2] G. Passas and M. Katevenis, "Packet mode scheduling in buffered crossbar (CICQ) switches," in *Proc. High Performance Switching and Routing (HPSR) 2006*, Poznan, Poland (June 2006).

[3] Y. Doi and N. Yamanaka, "A high-speed ATM switch with input and cross-point buffers," *IEICE Transactions on Communications*, vol. E76-B, no. 3, pp. 310–314 (Mar. 1993).

[4] M. Nabeshima, "Performance evaluation of a combined input- and crosspoint-queued switch," *IEICE Transactions on Communications*, vol. E83-B, no. 3, pp. 737–741 (Mar. 2000).

[5] Rojas-Cessa, E. Oki, and H. J. Chao, "CIXOB-*k*: Combined input-crosspoint-output buffered packet switch," in *Proc. IEEE GLOBECOM'01*, San Antonio, Texas, pp. 2654–2660 (Nov. 2001).

[6] N. McKeown, "*i*SLIP: a scheduling algorithm for input-queued switches," *IEEE/ACM Transactions on Networking*, vol. 7, no. 2, pp. 188–201 (Apr. 1999).

[7] R. Rojas-Cessa, E. Oki, Z. Jing, and H. J. Chao, "CIXB-1: combined input-one-cell-crosspoint buffered switch," in *Proc. IEEE Workshop on High Performance Switching and Routing*, Dallas, Texas, pp. 324–329 (May 2001).

[8] R. Schoenen, G. Post, and G. Sander, "Weighted arbitration algorithms with priorities for input-queued switches with 100% throughput," in *Proc. Broadband Switching Symposium'99*, Kingston, Canada, 1999. [Online]. Available at: http://www.schoenen-service.de/assets/papers/Schoenen99bssw.pdf

[9] C. S. Chang, D. S. Lee, and Y. S. Jou, "Load balanced Birkhoff–von Neumann switches," in *Proc. IEEE HPSR 2001*, Dallas, Texas, pp. 276–280 (May 2001).

[10] B. Li, M. Hamdi, and X.-R. Cao, "An efficient scheduling algorithm for input-queuing ATM switches," in *Proc. IEEE Broadband Switching Systems'97*, Taiwan, pp. 148–154 (Dec. 1997).

[11] E. Oki and N. Yamanaka, "A high-speed ATM switch based on scalable distributed arbitration," *IEICE Transactions Communications*, vol. E80-B, no. 9, pp. 1372–1376 (Sept. 1997).

[12] N. McKeown, "Scheduling algorithms for input-queued cell switches," Ph.D. Thesis, UC Berkeley, May 1995.

[13] T. Javidi, R. Magill, and T. Hrabik, "A high throughput scheduling algorithm for a buffered crossbar switch fabric," in *Proc. IEEE International Conference on Communications*, Helsinki, Finland, vol. 5, pp. 1586–1591 (June 2001).

[14] J. G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," in *Proc. IEEE INFOCOM'00*, Tel Aviv, Israel, pp. 556–564 (Mar. 2000).

[15] L. Mhamdi and M. Hamdi, "MCBF: a high-performance scheduling algorithm for buffered crossbar switches," *IEEE Communications Letters*, vol. 7, no. 9, pp. 451–553 (Sept. 2003).