MCBF: A High-Performance Scheduling Algorithm for Buffered Crossbar Switches

Lotfi Mhamdi, *Student Member, IEEE*, and Mounir Hamdi, *Member, IEEE*

***Abstract*—The buffered crossbar architecture is becoming very attractive for the design of high performance routers due the unique features it offers. Recently, many distributed scheduling algorithms have been proposed for this architecture. Despite their distributed nature, the existing schemes require quite a bit of hardware and timing complexity. In this letter, we propose a novel scheduling scheme named the *most critical buffer first* (MCBF). This scheme is based only on the internal buffer information and requires much less hardware than the existing schemes. Yet, it exhibits good performance and outperforms all its competitors. More interestingly, MCBF shows optimal stability performance while being almost a stateless algorithm.**

***Index Terms*—Buffered crossbar fabric, scheduling algorithms.**

I. Introduction

Buffered crossbar switch (BCS) architecture, that employs input virtual output queues (VOQ) in the ingress ports, is gaining increasing interest and is being considered as a robust solution in facing the challenging design of today's routers. In fact, The VOQ/BCS, which has been first introduced in [4] as shown in Fig. 1, has key advantages that can serve to ensure that the scheduling algorithm can be simple and efficient at the same time. The presence of internal buffers improves drastically the overall performance of the switch due to the advantages it offers. First, the adoption of internal buffers makes the scheduling totally distributed, hence reducing the arbitration complexity and makes it linear. Second, and most importantly, these internal buffers reduce (or avoid) the output contention. Meaning, they allow the inputs to send cells to an output irrespective of simultaneous cell transfer to the same output.

Recently, many scheduling schemes for the VOQ/BCS architecture were proposed. The simplest scheme is based on round-robin (RR-RR) arbitration in both the input and the output side [6]. A scheme, based on the oldest cell first (OCF) in the input as well as in the internal buffers, was proposed in [4]. An algorithm, based on the longest queue first (LQF) at the input side followed by round-robin arbitration at the output side, was introduced in [7]. All these algorithms were just a simple mapping of earlier algorithms proposed for buffer-less crossbar switch into the new VOQ/BCS architecture.

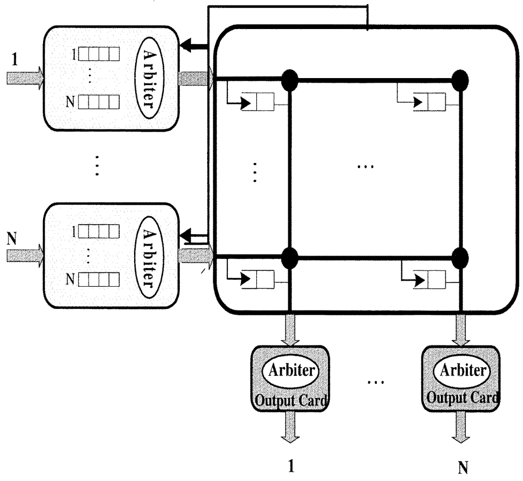

Fig. 1. The VOQ/BCS architecture.

The good performance that LQF-RR and OCF-OCF exhibited was at the expense of being quite complex in hardware implementation. For both input scheduling, LQF and OCF, the arbiters decisions are very time consuming [1] due to the large number of input values (i.e., queue length or cell age). These arbiters take almost 75% of the whole arbitration time [1]. Recall that a packet, while being inside the switch, needs additional processing such as overhead and quality-of-service (QoS) information. Given the very short time constraints to switch packets from the line cards to their outgoing ports, the arbitration time can become soon a bottleneck.

In this letter, we propose a scheduling scheme based on the *Shortest internal Buffer First* (SBF) at the input side with a scheme based on the *Longest internal Buffer First* (LBF) at the output side. We show that information based only on the internal buffers is sufficient for the schedulers to make effective decisions while being simple to implement in hardware. Our scheme does not use any input state information, such as VOQs occupancies or VOQs head-of-line (HoL) cells waiting time. Yet, SBF-LBF yields very high throughput and outperforms all the previously proposed algorithms under many traffic patterns. A stability study was carried out on the behavior of the input VOQs to investigate the performance of our scheme, as a stateless scheme. The outcomes are surprisingly good, especially when compared to [7] which was proven to be stable under any admissible input traffic that obeys the strong law of large numbers.

The rest of the paper is organized as follows. In Section II, we present our proposed scheme along with its properties. Section III contains a simulation study. Finally, Section IV concludes the letter.

Manuscript received February 6, 2003. The associated editor coordinating the review of this letter and approving it for publication was Prof. J. Evans. This work was supported in part by the Hong Kong Research Grant Council under Grant HKUST6181-01E.

The authors are with the Department of Computer Science, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: lotfi@cs.ust.hk; hamdi@cs.ust.hk).

Digital Object Identifier 10.1109/LCOMM.2003.815665

II. Most Critical Buffer First Algorithm (MCBF)

A. Motivation

To keep pace with the Internet's exponential growth, building routers with large number of ports and high line rates is becoming a must. For example, switches of size 256 to 1024, and running at OC768 (40 Gb/s) or even higher speed are becoming a necessity for most IP core networks. Generally, the interconnect runs faster than the line speed to amortize the time spent on some additional requirements such as QoS related processing and imperfect output contention resolution. If we consider transferring packets (or ATM cells), of size 53 B each, through a 40-Gb/s switch port with a speed up of 2, the scheduler has approximately 5.3 ns to decide which packet to forward. This short time constraint requires the scheduler to make its arbitration as fast as it possibly can.

The schemes proposed so far for the VOQ/BCS architecture are mainly based on sorting, such as LQF-RR and OCF-OCF. If we consider the hardware complexity of LQF scheduling for example, we can see that it takes relatively long time to make its arbitration. This is mainly due to the large number of input values (i.e., number of packets in a line card) and the basic building blocks of the arbiters, which are mainly two-integer comparators and two-integer MUXes [8]. In a similar implementation, it was shown that the arbitration time is more than 7 ns for a 32 × 32 switch with 10 b representing the input weight [1]. Even with the fastest implementation, the two-input integer comparator still takes $O(\log B)$ time units to complete the comparison [2], where B is the number of bits equaling $\log L_{\max}$ (the maximum number of packets a line card can hold). The 10 b representing the weight above corresponds to a maximum of 53 KB as the buffer space at the line card. However, it is usually required that the buffer size at each line card should hold up to 100 ms worth of packets [3]. Meaning that, at 40 Gb/s, the buffer size can be as large as 500 MB. Thus, it is clear that employing LQF (or OCF) arbitration will result in much longer arbitration time and therefore will most likely be the bottleneck of the whole switch.

In an attempt to reduce the arbitration complexity while keeping good performance, our new scheme MCBF is proposed. It is based only on the internal buffers information. It favors the least occupied internal buffer at the input side. While the output gives priority to the most occupied internal buffer. This means that, the scheduler keeps the information about the internal buffers only, instead of the input queues length in the case of LQF. Doing so, instead of $\log L_{\max}$, B will equal to $\log(P.S.)$, where P is the number of switch ports and S equals to the internal buffer size in number of packets.

B. Notation

We consider the switch model defined in Fig. 1. There are N input cards; each one maintains N logically separated VOQs. When a packet (cell), destined to output j, $1 \leq j \leq N$, arrives to the input card i, $1 \leq i \leq N$, it is held in $VOQ_{i,j}$.

- Eligible *VOQ* (*EVOQ*): A $VOQ_{i,j}$, is said to be eligible (denoted $\text{EVOQ}_{i,j}$) for being scheduled in the input scheduling process if it is not empty and the internal buffer $XP_{i,j}$ is empty (or not full).
- The internal fabric consists of N^2 buffered crosspoints (XP). A crosspoint $XP_{i,j}$, holds cells coming from input i and going to output j.
- The line of crosspoint buffers $LXPB_i$ is the set of all the internal buffers ($XP_{i,j}$) that correspond to the same input, i, and holding cells for all outputs. NLB_i is the number of cells held in $LXPB_i$.
- The column of the crosspoint buffers $CXPB_j$ is the set of the internal buffers $(XP_{i,j})$ that correspond to the same output, j, and receiving cells from all inputs. NCB_j is the number of cells held in $CXPB_j$.

C. MCBF Specification

The MCBF scheme is based on the *Shortest internal Buffer First* (SBF) as its input scheduling. Its output arbitration is based on the *Longest internal Buffer First* (LBF). The specification of each 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 $(XP_{i,j})$. Move the highest priority pointer to the location $j + 1 (\bmod N)$.
- *Output Scheduling* (**LBF**)

  For each output j: Starting from the highest priority pointer's location, select the first $XP_{i,j}$ corresponding to: $\max_i \{NLB_i\}$ and send its HoL cell to the output. Move the highest priority pointer to the location $i + 1 (\bmod N)$.

D. MCBF Properties

The MCBF scheme has three major properties when compared to other schemes. First, MCBF is simpler in hardware complexity when compared to LQF-RR or OCF-OCF for example. Recall that MCBF's scheduling decision is based on the number of cells in the internal buffers (NLB_i, NCB_j). That is, for $N \times N$ one-cell internally buffered crossbar switch, an arbiter's encoder consists only of log N bits ($P = N$ and $S = 1$). This is much faster than comparing $\log B$, where B is equal to $\log L_{\max}$ in the case of comparing the queues' occupancies [1]. More interestingly, the product $P.S$ remains small irrespective of the internal buffer size. It grows linearly. Second, MCBF is a scheme which is almost stateless. It makes its arbitration without any type of state information about the input VOQs. The only feedback information that MCBF needs to have during its arbitration process is whether an input VOQ is empty or not. Finally, MCBF is designed to be a matched pair of input and output scheduling. The internal buffer element is of key importance in finding matched scheduling because of its shared nature. No output is idle so long as $NCB_j \geq 1, \forall 1 \leq i, j \leq N - 1$. To keep the outputs as busy as possible, MCBF maintains a load balancing among the internal buffers.

III. Performance Study

We simulated MCBF and compared it to LQF-RR and OCF-OCF using a 32 × 32 VOQ/BCS switch. The performance evaluation is done through two traffic models: Bursty uniform and Bernoulli nonuniform.

A stability performance study was carried out along with the delay study. Similar to [5], the input queues occupancies can serve to prove the stability of the scheduling algorithm. That is, if under a service policy X, we can show that $E(\|L(n)\|) < \infty$, then we can conclude that X is stable. $\|L(n)\|$ is the l-two norm

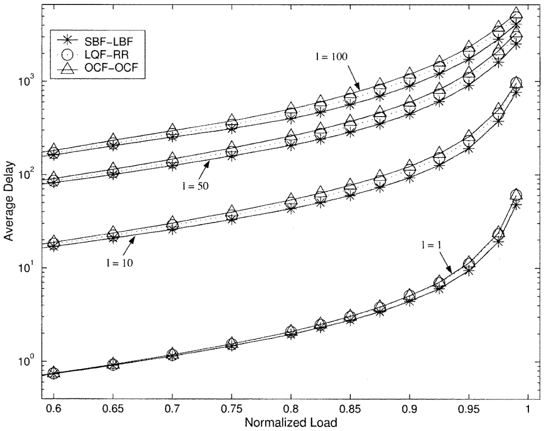

Fig. 2. Performance under bursty uniform traffic.

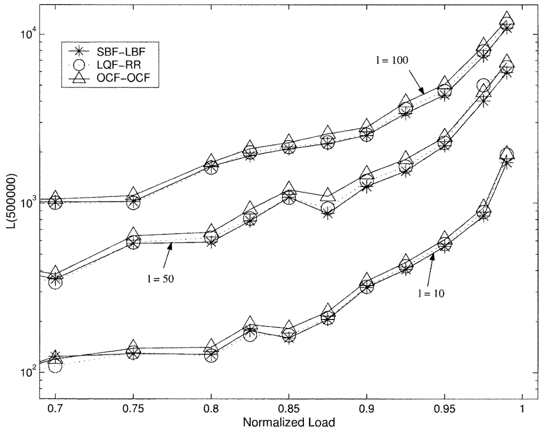

Fig. 3. The l-two norm vector under Bursty uniform traffic.

vector representing the occupancy of the VOQs a time n and is defined as follows:

$$||L(n)|| = \sqrt{\begin{array}{r} VOQ_{1,1}(n)^2 + \cdots + VOQ_{1,N}(n)^2 + \cdots \\ +VOQ_{N,1}(n)^2 + \cdots + VOQ_{N,N}(n)^2. \end{array}}$$

Fig. 2 shows the average delay performance under bursty uniform traffic with burst lengths equal 1, 10, 50, and 100, respectively. Under heavy load, SFB-LBF exhibits the shortest delay amongst all the schemes. Note that when the burst length equals 1, the traffic is Bernoulli uniform. At 99% load and burst length of 10, SBF-LBF has an average queuing delay less than 80% that of LQF-RR. With a burst length of 50 and at 99% load SBF-LBF has an average delay of 2523, then LQF-RR with an average delay of 3014 and finally OCF-OCF with an average delay of 3311.

As for the l-two norm vector's stability, shown in Fig. 3, SBF-LBF has the best performance amongst all despite the fact that it maintains no information at all about the input VOQs.

As for the nonuniform traffic, we used the same unbalanced traffic as in [6]. As shown in Fig. 4, we can see that SBF-LBF can achieve high throughput irrespective of the unbalanced coefficient, W. It is expected that the performance of SBF-LBF increases as the internal buffer size increases. Fig. 5 depicts the performance of each algorithm with an internal buffer size of eight cells. As expected, SBF-LBF exhibits the best performance, because as the internal buffer size increases SBF-LBF emulates more the Longest Port First (LPF) algorithm [1].

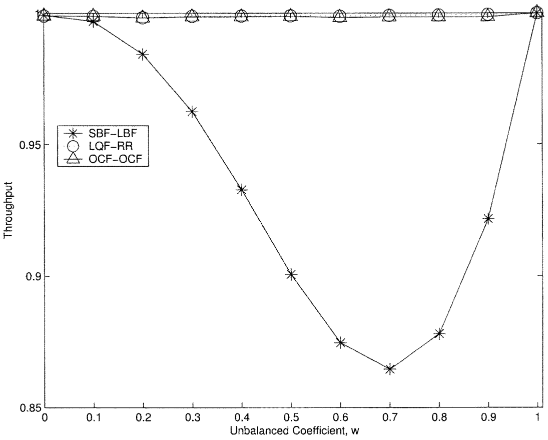

Fig. 4. Stability under nonuniform traffic, internal buffer $= 1$ cell.

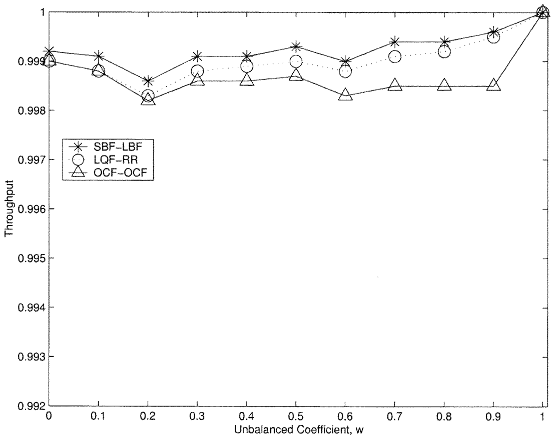

Fig. 5. Stability under unbalanced traffic and internal buffer $= 8$ cells.

IV. Conclusion

This paper presents a new scheduling algorithm for the buffered crossbar switch. This scheme takes full advantage of the VOQ/BCS architecture, and in particular, the interaction between the internal buffers and the VOQs. Unlike the previously proposed algorithms, basing their arbitration on the input VOQ's occupancies or waiting time, the *MCBF* algorithm is almost stateless and keeps no state information about the input VOQs. Yet, it shows surprisingly good performance especially with respect to the stability of the input VOQs.

References

[1] A. Mekkittikul, "Scheduling nonuniform traffic in high speed packet switches and routers," Ph.D. dissertation, Stanford Univ., Stanford, CA, Nov. 1998.

[2] T. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA: MIT Press, Mar. 1990.

[3] H. J. Chao, "Next Generation Routers," *Proc. IEEE*, vol. 90, no. 9, Sept. 2002.

[4] M. Nabeshima, "Performance evaluation of combined input-and cross-point-queued switch," *IEICE Trans. Commun.*, vol. E83-B, no. 3, Mar. 2000.

[5] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in input-queued switch," *IEEE Trans. Commun.*, vol. 47, Aug. 1999.

[6] R. Rojas-Cessa, E. Oki, Z. Jing, and H. J. Chao, "CIXB-1: Combined input one-cell-crosspoint buffered switch," in *Proc. 2001 IEEE WHPSR*, 2001, pp. 324–329.

[7] T. Javadi, R. Magill, and T. Hrabik, "A high-throughput algorithm for buffered crossbar switch fabric," in *Proc. IEEE ICC*, June 2001, pp. 1581–1591.

[8] H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks," *Proc. IEEE*, vol. 83, pp. 1374–1376, Oct. 1995.