Shared-Memory Combined Input-Crosspoint Buffered Packet Switch for Differentiated Services

Ziqian Dong and Roberto Rojas-Cessa
Department of Electrical and Computer Engineering
New Jersey Institute of Technology
University Heights, Newark NJ 07102

***Abstract*— Combined input-crosspoint buffered (CICB) packet switches with dedicated crosspoint buffers require a minimum amount of memory in the buffered crossbar of $N^2 \times k \times L$, where N is the number of ports and k is the crosspoint buffer size, which is defined by the distance between the line cards and the buffered crossbar, to achieve 100% throughput under high-speed data flows. A long distance between these two components can make a buffered crossbar costly to implement. In this paper, we study a shared-memory crosspoint buffered packet switch that uses small crosspoint buffers and no speedup to support differentiated services and long distances between the line cards and the buffered crossbar in practical implementations. The proposed switch requires a buffer memory of $\frac{1}{m}$ of that in a CICB switch with dedicated crosspoint buffers to achieve similar throughput performance to that of a CICB switch.**

***Index Terms*— Buffered crossbar, round-trip time, no speedup, shared memory, differentiated services.**

I. Introduction

As more voice and video traffic emerge along with data traffic in the internet nowadays, the demand for switches providing differentiated services has increased. Prioritized scheduling has been studied in input-queued switches [1]. Combined input-crosspoint buffered (CICB) switches are used to relax arbitration timing in an input-queued switch while providing high-performance switching and high-speed ports [3]. These packet switches use time efficiently as input and output port arbitrations are performed separately [2]-[13]. Incoming variable-size packets are segmented into fixed-length packets, called cells, at the ingress side of a switch and re-assembled at the egress side, before the packets depart from the switch. This paper considers the use of cells.

For a CICB switch, the required amount of memory in a buffered crossbar to avoid buffer underflow for high-rate flows is

$$N^2 \times k \times L, \tag{1}$$

where N is the number of input/output ports, k is the crosspoint buffer size in number of cells, and L is the cell size in bytes. The value of k is defined by the length of the round-trip time (RTT), where

$$RTT = IA + d1 + OA + d2. \tag{2}$$

Roberto Rojas-Cessa is the corresponding author. Email: rrojas@njit.edu

RTT is defined as the sum of the delays of 1) the input arbitration IA, 2) the transmission of a cell from an input to the crossbar $d1$, 3) the output arbitration OA, and 4) the transmission of the flow-control information back from the crossbar to the input, $d2$ [7]. Cell and bit alignments are included in the transmission time. For example, the switch proposed in [7] requires the size of k be equal to or larger than the round-trip time to avoid throughput degradation or crosspoint-buffer underflow for flows (here defined as the data arriving at input i and destined to output j, where $0 \leq i, j \leq N-1$) with high data rates.

In a CICB switch, the crosspoint-buffer size, k, must be at least RTT cells long to avoid underflow [7]. Furthermore, as the buffered crossbar can be physically located far from the input ports, actual RTTs can be long. To support long RTTs in a buffered-crossbar switch, the crosspoint-buffer size needs to be increased, such that at least RTT cells can be buffered. However, the memory amount that can be allocated in a chip is limited, and therefore, it can make the implementation costly or infeasible when the distance between line cards and the buffered crossbar is long, or else to provide a size of k cells where $k < RTT$ without supporting high data rates. The effect of long RTT is studied in detail in [16].

In this paper, we focus on providing differentiated services, where traffic is assigned different switching priorities. High priority cells are from delay sensitive traffic such as voice and video. They need to be served prior to other traffic classes to minimize queueing delay within the switch. An interesting scheme using limited memory is presented in [10] for a switch with P traffic classes, where the crosspoint buffer size is larger than RTT for a single class, and smaller than $P \times RTT$.

A solution to keep the crosspoint buffer small while supporting differentiated services, long RTTs, and high data rates is needed.

To reduce the memory amount while supporting longer RTT values and multiple priority traffic, we use a CICB switch that shares the crosspoint buffers, called the shared-memory crosspoint buffered (SMCB) switch, among two inputs [16]. This switch uses shared memories as crosspoint buffers to reduce the total amount of memory in the crossbar such that the available area in the buffered crossbar chip is efficiently used, specially for flows with high data rates and long distances between the line cards and the buffered

crossbar. We show that the SMCB switch supports a given RTT with half or less memory than that of a buffered crossbar with dedicated crosspoint buffers, and that the switch delivers equivalent or better switching performance for different classes of traffic than a CICB switch. We also show that no speedup is needed to achieve high throughput when using the shared-memory approach and the equivalent amount of memory in the crossbar.

This paper is organized as follows. Section II presents a brief description of the pre-existing CICB switch with round-robin input and output arbitrations, or the CIXB switch with prioritized service. Section III introduces the proposed SMCB switch. Section IV presents the throughput performance of the SMCB switch. Section V presents our conclusions.

II. CICB Switch with Dedicated Crosspoint Buffers for Differentiated Services

Here, we describe a CICB switch with dedicated crosspoint buffers. The CICB switch architecture follows the one presented in [7]. We consider that traffic is classified in P different classes in this paper. A buffered crossbar has N inputs and outputs. A crosspoint (CP) element in the buffered crossbar that connects input port i to output port j is denoted as $CP(i,j)$. There are NP VOQs at each input. A VOQ at input i that stores cells for output j of priority p, where $0 \leq p \leq P-1$, is denoted as $VOQ(i,j,p)$. Here, we consider that $P = 3$ for sake of clarity. The CP buffer of $CP(i,j)$ is denoted as $CPB(i,j)$, and this is not prioritized. The size of $CPB(i,j)$ is k cells, where $k \geq 1$.

A credit-based flow control mechanism is used to indicate at input i whether $CPB(i,j)$ has room available for a cell or not. Each VOQ has a credit counter, where the maximum count is the number of cells that $CPB(i,j)$ can hold. When the number of cells sent by $VOQ(i,j,p)$ reaches the maximum count, the VOQ is considered not eligible for input arbitration and overflow on $CPB(i,j)$ is avoided. The count is increased by one each time a cell is sent to $CPB(i,j)$ and decreased by one each time that $CPB(i,j)$ forwards a cell to output j. If $CPB(i,j)$ can receive at least one cell, then $VOQ(i,j,p)$ is considered eligible by the input arbiter.

An input arbiter at input i selects $VOQ(i,j,p)$ among the eligible VOQs to send a cell to CPB for output j at buffered crossbar. An output arbiter at output port j in the buffered crossbar selects a $CPB(i,j)$, among occupied CPBs from input i, to send a cell to output j. The eligibility of VOQs is determined by the flow control mechanism.

Different from the switch in [7], this CICB switch has round-robin with strict priority for input arbitration. In this input arbitration scheme, round-robin selection is performed among all queues of the highest priority. If there are no cells of the highest priority, round-robin is performed among the second highest priority, and so on. Output arbitration considers only round-robin with no priorities to avoid to starving cells with low priority.

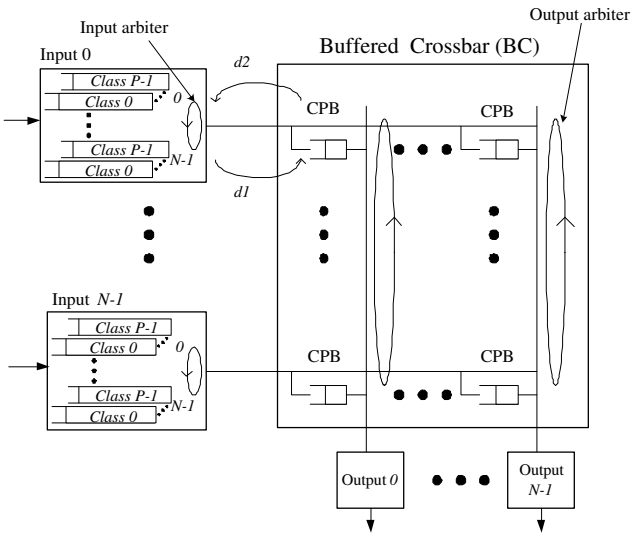

Fig. 1. CICB switch with round-robin input and output arbitrations.

This switch has dedicated crosspoint buffers, this is, $VOQ(i,j,p)$ can access only the CPB at input row i that is connected to output j in the buffered crossbar, and each CPB can be accessed by P VOQs with the same i,j pair. Therefore, if the VOQs of the i,j pair have no cells for $CPB(i,j)$, the crosspoint buffer remains idle.

III. Shared-Memory Crosspoint Buffered Switch (SMCB)

As discussed in [16], the largest throughput degradation of a CICB switch with dedicated crosspoint buffers occurs when the rate of flow $f(i,j)$, denoted as $r_{f(i,j)}$, is equal to port capacity, R_c. Under these conditions, all traffic at input i goes to the crosspoint that connects to output j and the other crosspoints receive no traffic. This motivates the sharing of the crosspoint memory by two or more inputs. In a SMCB switch, the crosspoint buffer is shared by m inputs, where $2 \leq m \leq N$ [16].

We adapted the SMCB switch to support different classes of traffic. The SMCB switch uses input access schedulers to arbitrate the access from the inputs to the crosspoint buffers. The input access scheduler matches the sharing inputs and the shared crosspoints to eliminates the speedup of the shared memory. In paper, we consider P classes of traffic with class 0 being the highest priority and class $P-1$ the lowest priority. Here we define high priority traffic as delay sensitive traffic that needs to be served prior to the lower priority traffic.

The proposed switch has NP VOQs at each input. A crosspoint in the shared-memory buffered crossbar that connects input port i to output j is also denoted as $CP(i,j)$ as in the CICB switch. To simplify our explanation and because the switch delivers the highest throughput [16], we consider that a crosspoint buffer is shared by two inputs (i.e., $m = 2$). However, the number of inputs sharing the buffer can be from 2 to N. The buffer for $CP(i,j)$ and $CP(i',j)$, where $0 \leq i,i' \leq N-1$ and $i \neq i'$, that stores cells for output

port j and is shared by these two crosspoints (or inputs i and i') is denoted as $SMB(q, j)$, where $0 \leq q \leq \frac{N}{2} - 1$. We assume an even N for the sake of clarity. However an odd N can also be considered. Therefore, in a switch where each crosspoint buffer is shared by two inputs, there are $\frac{N^2}{2}$ SMBs in the buffered crossbar.

Figure 2 shows the architecture of the SMCB switch with two inputs sharing the buffered crosspoints. To eliminate the speedup at SMBs, only one input is allowed to access an SMB at a time. To schedule the SMB access between two inputs, an input access scheduler, $S(q)$, is used among the two inputs that share the same SMB. The size of an SMB, in number of cells that can be stored, is k_s. There are $\frac{N}{2}$ input access schedulers in the buffered crossbar.

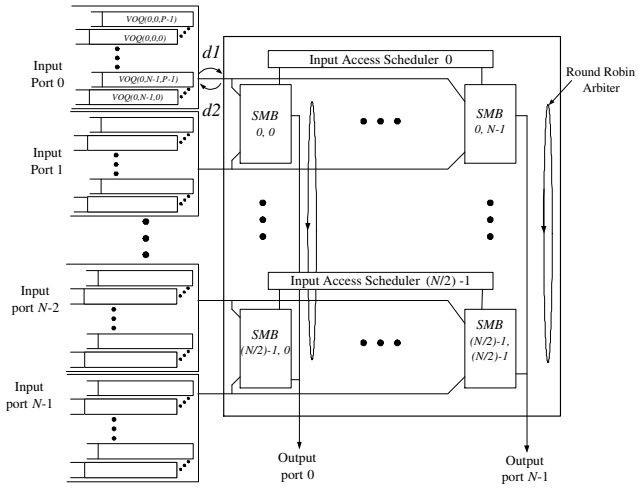

Fig. 2. $N \times N$ buffered crossbar with shared crosspoints.

$S(q)$ performs a matching process among the N SMBs and the $m = 2$ non-empty inputs that share them. Figure 3 shows the inputs and the shared crosspoint buffers that take place in the matching. In this paper, the matching follows a three-phase process, as performed for input-queued (IQ) switches. The matching scheme used in this switch is round-robin based [14] with strict priority.

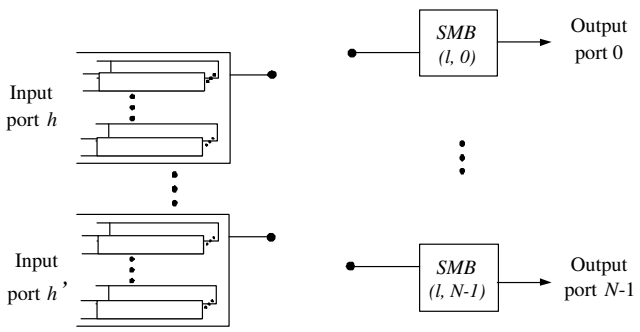

Fig. 3. Bipartite matching in an input access scheduler.

At each output j in the buffered crossbar, there is an output arbiter to select the outgoing cell from non-empty SMBs. The arbitration scheme for outputs is round-robin without considering priorities to avoid SMB blocking. This blocking, produced when strict priority selection is used as output arbitration, occurs when a low priority cell in an SMB blocks the forwarding of a higher priority cell in the VOQ, because the low priority cell is neglected service by the output arbiter as other SMBs may have high priority cells. Since an SMB holds cells from two different inputs, then an output arbiter considers up to two cells from an SMB, where each cell belongs to one of the inputs. Figure 2 shows the output arbiters, where the transmission delays between ports and the crosspoint are denoted by $d1$ and $d2$.

The way the SMCB switch works is as follows. Cells with priority p, where $0 \leq p \leq P - 1$, destined to output j arrive at $VOQ(i, j, p)$ and wait for dispatching. Input i notifies $S(q)$ about the new cell arrival. $S(q)$ selects the next cells to be forwarded to the crossbar by performing a parallel match between the inputs and the SMBs using prioritized selection. A cell going from input i to output j enters the buffered crossbar and is stored in $SMB(q, j)$. Cells leave output j after being selected by the output arbiter. To reduce the complexity at the buffered crossbar, the output arbiter uses round-robin selection without considering the cell priority.

$S(q)$ considers eligible VOQs to those non empty and for which the SMBs that are not full. The input access scheduler information is sent from the SMB to the corresponding VOQ in the next time slot. Therefore, the flow control mechanism between the inputs and the buffered crossbar is stop-and-go and no arbiter is needed at the line cards. Cells and flow-control data experience transmission delay between input ports and the buffered crossbar.

IV. Performance Evaluation of the SMCB Switch under Differentiated Traffic

In this section, we study the switching performance of the SMCB switch and the CICB switch (labeled as CICB in the remainder of this paper). The throughput of a SMBC switch with different number of inputs and with a single-priority traffic under long RTTs flows with high data rates is studied in [16]. Here, we focus on handling multi-priority traffic. For this, we first compare the switching performance of 32×32 SMCB and CICB switches under traffic with Bernoulli and bursty arrivals with uniform distribution and P classes, where $P = 3$, with class 0 being the highest and class 2 being the lowest. We study the average cell delay of each priority under uniform traffic and bursty traffic with different burst lengths. Here, a burst is modeled as a two-state Markov modulated process, with an average length l for the active state. Then, we show the throughput performance under nouniform prioritized traffic, using the unbalanced and diagonal traffic models.

A. Uniform traffic

Figures 4 and 5 show the average cell delay of the SMCB switch under uniform traffic with average burst lengths of 1 and 100 cells respectively. The RTT under uniform traffic

is 1. The minimum average delay of one time slot of the SMCB switch from loads of 0.1 to 0.8 is because the VOQs notifies the input access scheduler when a new cell arrives, and this takes one time slot, and before forwarding the cell to the buffered crossbar. Note that the magnitude of one time slot is small to be considered significant. For loads over 0.8, the average cell delay of the highest priority traffic, class 0, is significantly lower than that of lower priorities, classes 1 and 2. The average cell delay of SMCB switch under bursty traffic with an average burst length $l = 100$ shows similar magnitude to that of the CICB switch when the load is less than 0.85. When the input load is heavier, the service for the lowest priority class degrades because higher priority classes use most of the service. In this case, Class 2 traffic can no longer achieve 100% throughput in both the SMCB and CICB switches. However, the amount of memory used in the SMCB switch is $\frac{N^2}{2}$, which is $\frac{1}{2}$ of that in the CICB switch, and no speedup is needed in the shared memory.

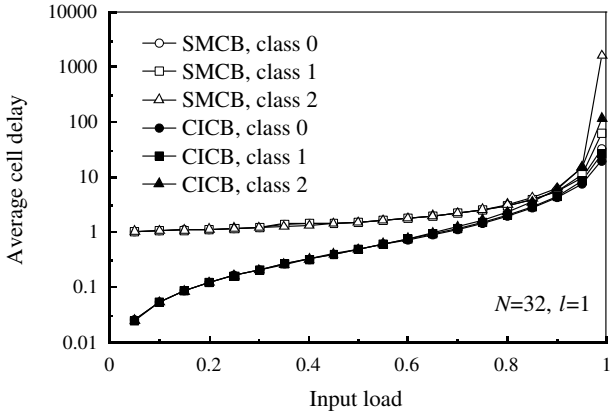

Fig. 4. Average queuing delay of 32×32 SMCB and CICB switch when $l = 1$, $k_s = k = 1$, and $RTT = 1$.

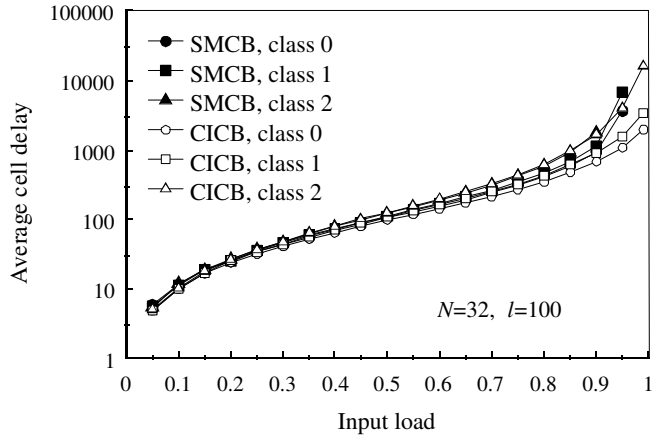

Fig. 5. Average queuing delay of a 32×32 SMCB and CICB switch when $l = 100$, $k_s = k = 1$, and $RTT = 1$.

B. Unbalanced traffic

We consider the unbalanced traffic model as a nonuniform traffic. This model also can be used to describe flows with different data rates [16]. This traffic model uses the probability w as the fraction of input load directed to a single predetermined output, while the rest of the input load is directed to all outputs with uniform distribution. Let us consider input port i, output port j, and the offered input load for each input port ρ. The traffic load from input port i to output port j, $\rho(i, j)$ is given by,

$$\rho(i,j) = \begin{cases} \rho\left(w + \frac{1-w}{N}\right) & \text{if } i = j \\ \rho\frac{1-w}{N} & \text{otherwise.} \end{cases}$$

When $w = 0$, the offered traffic is uniform. On the other hand, when $w = 1$, the traffic is completely directional, from input i to output j, where $i = j$. This means that all traffic of input port i is destined for only output port j. This is the maximum port capacity.

In this section, we compare the throughput performance between SMCB and CICB switch with different crosspoint buffer sizes and RTTs under unbalanced traffic.

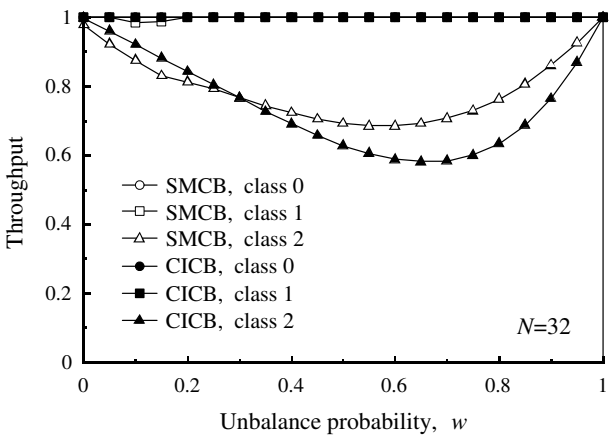

Fig. 6. Throughput of 32×32 SMCB and CICB switch when k_s=k=RTT=1.

Figure 6 shows that for the CICB switch, when $k = RTT = 1$, class 0 and class 1 traffic achieve 100% throughput under unbalanced traffic. However; the throughput for class 2 traffic is low. When $k_s = 1$, the total amount of memory in a SMCB switch is $\frac{N^2}{2}$ cells. When $k = 1$, the total amount of memory in the CICB switch is N^2 cells. The SMCB switch achieves similar or better throughput performance that that of the CICB switch with only half the amount of memory in the CICB switch under unbalanced traffic when RTT is not an issue.

When RTT increases from 1 to 3 time slots (or cells), a crosspoint buffer size of 1 is not enough to ensure 100% throughput for classes 1 and 2 in the CICB switch. Figure 7 shows that the throughput for traffic classes 1 and 2 degrades drastically as the unbalance probability increases, which indicates an increase in the flow rate. For the SMCB switch, the traffic for classes 0 and 1 achieve 100% throughput.

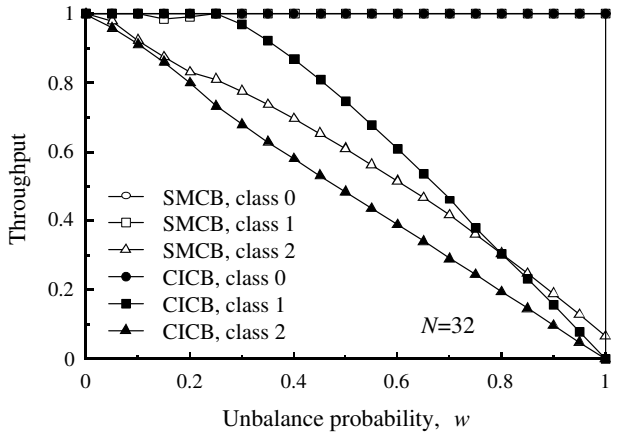

Fig. 7. Throughput of 32×32 SMCB and CICB switch when $k_s = k = 1$, and RTT=3.

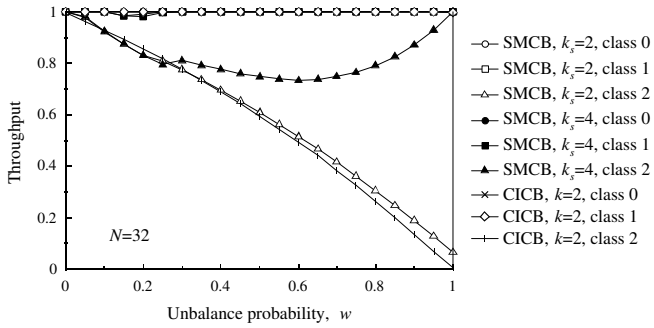

Fig. 8. Throughput of 32×32 SMCB and CICB switch when SMCB switch has half ($k_s = k = 2$), the same amount of memory as CICB switch ($k_s = \frac{1}{2}k = 4$), and $RTT = 3$.

However, the throughput of class 2 traffic degrades as the unbalance probability increases. These results further show the importance of addressing the effect of long RTTs.

Figure 8 shows the throughput of the SMCB and CICB switches when the SMCB switch has half and the same amount of memory as that of the CICB switch in the crossbars, and $RTT = 3$. By increasing the crosspoint buffer size by one cell ($k_s = k = 2$), the CICB and SMCB switch achieve higher throughput performance for class 1 traffic than when $k_s = k = 1$. In this figure, when the SMCB switch has the same amount of memory ($k_s = 4$ and $k = 2$) as that of the CICB switch, all traffic classes have a guaranteed throughput of 100% for $w = 1.0$. Class 2 traffic shows similar throughput performance when $0 < w < 1.0$ as in Figure 6.

C. Diagonal traffic

The Diagonal traffic model can be represented as $d\rho(i,j) = d\rho_i$ for $i = j$, $(1-d)\rho_i$ for $j = (i+1) \bmod N$, where ρ_i is the load at input i. or by the matrix $\bar{\rho}$ as:

$$\bar{\rho} = \rho \begin{pmatrix} d & (1-d) & 0 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ (1-d) & 0 & \dots & 0 & d \end{pmatrix}$$

This traffic model distributes the load of an input among two outputs only. The distribution is given by the diagonal degree probability, d.

In this section, we compare the throughput performance between SMCB and CICB switch with different crosspoint buffer sizes and RTTs under diagonal traffic with maximum allowable input load, $\rho_i = 1$. Since the results under diagonal traffic are symmetric at $d = 0.5$, the following graphs show throughput results at $0 \le d \le 0.5$.

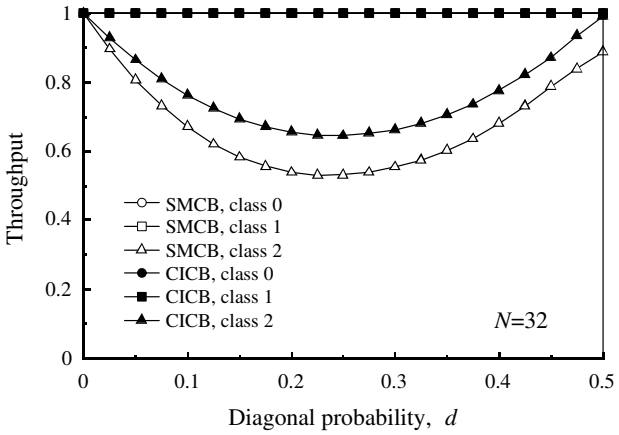

Fig. 9. Throughput of 32×32 SMCB and CICB switch when SMCB switch has half the amount of memory as CICB switch ($k_s = k = 1$) and $RTT = 1$.

Figure 9 shows the throughput of the SMCB and CICB switch when SMCB switch has half the amount of memory as that of the CICB switch. Class 0 and class 1 traffic achieve 100% throughput. The throughput of class 2 traffic for both switches degrade when $d \neq 0, 1$. The throughput of class 2 traffic of CICB switch is a slightly higher than that of the SMCB switch. Considering the fact that the SMCB switch has half the amount of memory as that of the CICB switch, the throughput advantage is not significant.

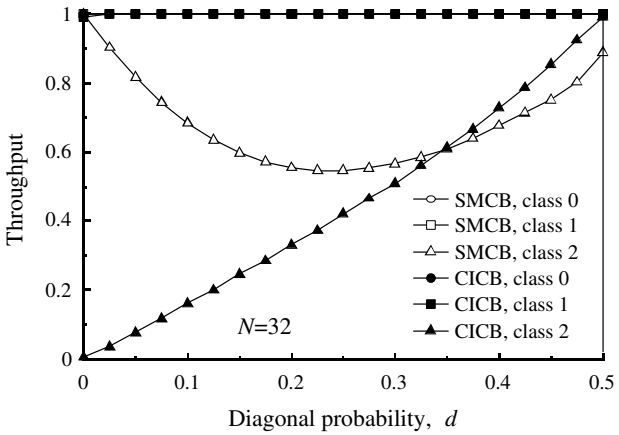

Fig. 10. Throughput of 32×32 SMCB and CICB switch when they have the same amount of memory ($k_s = 4$ and $k = 2$) and $RTT = 3$.

Figure 10 shows the throughput of the SMCB and CICB

switch when both switches have the same amount of memory, $k_s = 4, k = 2$, and $RTT = 3$. Both switches achieve 100% throughput for class 0 and class 1 traffic. The crosspoint buffer size here is set to one cell smaller than RTT to show the throughput difference of class 2 traffic of the two switches. With the same amount of memory, SMCB switch shows higher throughput performance than the CICB switch for class 2 traffic when $0 \leq d \leq 0.25$.

V. Conclusions

We presented the support for differentiated services by a buffered crossbar switch with dedicated buffers, or CICB switch, and for different RTTs. In a CICB switch, it is required that the crosspoint buffer size k be at least RTT cells long to achieve high throughput for all priority traffic classes, and specially for flows with high data rates. We observed that CICB switches, with an architecture as in [7], have their maximum throughput as the ratio of $\frac{k}{RTT}$, when input ports handle a single flow with a data rate equal to the port capacity. To minimize the crosspoint-buffer size, we use a shared-memory crosspoint buffer switch, SMCB, where the crosspoint buffers are shared by 2 inputs, such that RTT can be twice as long as that supported by a CICB switch with dedicated buffers, without decreasing switching performance. In this way, the SMCB switch provides 100% throughput for high data-rate flows under long round-trip times and when traffic has uniform distributions. This switch relaxes the amount of memory to $\frac{1}{2}$ of the amount required by a CICB switch with dedicated buffers to support P classes of traffic and flows with high data rates. This memory requirement relaxation can be further extended to $\frac{1}{m}$ when data flows are expected with high data rates (e.g., rates close to the port capacity), with however; a decrease of switching performance for traffic with nonuniform distributions.

In addition, we showed that the shared memory used in the crosspoint buffers needs no speedup to achieve high throughput. The timing relaxation that a CICB switch has for cell selection is partially lost in the SMCB switch as the input access schedulers perform parallel matching. However, the arbiter in the input access scheduler are placed in the same chip, the buffered crossbar, such that the matching time is rather small. The advantage of the SMCB switch is that memory is more efficiently handled than in a CICB switch such that higher switching performance is achieved without recurring to speedup and large amounts of memory. This trade-off well serves the cases when the distance between the line cards and the buffered crossbar is long.

References

[1] Minagar, A.R.; Safavi, S.M., "The optimized prioritized iSLIP scheduling algorithm for input-queued switches with ability to support multiple priority levels,"*Telecommunications, 2003. ICT 2003. 10th International Conference on*, Vol. 2, No. 23, pp. 1680-1685, March 2003.

[2] S. Nojima, E. Tsutsui, H. Fukuda, and M. Hashimmoto, "Integrated Packet Network Using Bus Matrix," *IEEE J. Select. Areas Commun.*, Vol. 5, No. 8, pp. 1284-1291, October 1987.

[3] Y. Doi and N. Yamanaka, "A High-Speed ATM Switch with Input and Cross-Point Buffers," *IEICE Trans. Commun.*, Vol. E76, No.3, pp. 310-314, March 1993.

[4] E. Oki, N. Yamanaka, Y. Ohtomo, K. Okazaki, and R. Kawano, "A 10-Gb/s (1.25 Gb/s x8) 4 x 0.25-μm CMOS/SIMOX ATM Switch Based on Scalable Distributed Arbitration," *IEEE J. Solid-State Circuits*, Vol. 34, No. 12, pp. 1921-1934, December 1999.

[5] M. Nabeshima, "Performance Evaluation of a Combined Input- and Crosspoint-Queued Switch," *IEICE Trans. Commun.*, Vol. E83-B, No. 3, pp. 737-741, March 2000.

[6] K. Yoshigoe, K.J. Christensen, "A parallel-polled Virtual Output Queue with a Buffered Crossbar," *Proceedings of IEEE HPSR 2001*, pp. 271-275, May 2001.

[7] R. Rojas-Cessa, E. Oki, Z. Jing, and H. J. Chao, "CIXB-1: Combined Input-One-Cell-Crosspoint Buffered Switch," *Proceedings of IEEE HPSR 2001*, pp. 324-329, May 2001.

[8] T. Javadi, R. Magill, and T. Hrabik, "A High-Throughput Algorithm for Buffered Crossbar Switch Fabric," *Proceedings of IEEE ICC 2001*, pp.1581-1591, June 2001.

[9] F. Abel, C. Minkenberg, R.P. Luijten, M. Gusat, and I. Iliadis, "A Four-Terabit Single-Stage Packet Switch with Large Round-Trip Time Support," *Proceedings of IEEE Hot Interconnects, 2002. 10th Symposium on*, pp. 5-14, August 2002.

[10] R. Luijten, C. Minkenberg, and M. Gusat, "Reducing Memory Size in Buffered Crossbars with Large Internal Flow Control Latency," *Proceedings of IEEE Globecom 2003*, Vol. 7, pp. 3683-3687, December 2003

[11] R. Rojas-Cessa, E. Oki, and H. J. Chao, "CIXOB-1: Combined Input-crosspoint-output Buffered Packet Switch," *Proceedings of IEEE GLOBECOM 2001*, Vol. 4, pp. 2654-2660, November 2001.

[12] L. Mhamdi and M. Hamdi, "MCBF: a high-performance scheduling algorithm for buffered crossbar switches," *IEEE Commun. Letters*, Vol. 7, Issue 9, pp. 451-453, September 2003.

[13] L. Mhamdi, M. Hamdi, "Practical Scheduling Algorithms for High-Performance Packet Switches," *Proceedings of IEEE ICC 2003*, Vol. 3, pp. 1659-1663, May 2003.

[14] N. McKeown, "The *i*SLIP Scheduling Algorithm for Input-Queue Switches", *IEEE/ACM Trans. Networking*, Vol. 7, No. 2, pp. 188-200, April 1999.

[15] R. Rojas-Cessa and E. Oki, "Round-Robin Selection with Adaptable-Size Frame in a Combined Input-Crosspoint Buffered Switch," *IEEE Commun. Letters*, Vol. 7, Issue 11, pp. 555-557, November 2003.

[16] Z. Dong and R. Rojas-Cessa, "Long Round-Trip Time Support with Shared-Memory Crosspoint Buffered Packet Switch," *Proceedings of IEEE Hot Interconnects, 2005. 13th Symposium on*, pp. 138-143, August 2005.