Long Round-Trip Time Support with Shared-Memory Crosspoint Buffered Packet Switch

Ziqian Dong and Roberto Rojas-Cessa
Department of Electrical and Computer Engineering
New Jersey Institute of Technology
University Heights, Newark NJ 07102
Email: {zd2, rrojas}@njit.edu

***Abstract*— The amount of memory in buffered crossbars in combined input-crosspoint buffered switches is proportional to the number of crosspoints, or $O(N^2)$, where N is the number of ports, and to the crosspoint buffer size, which is defined by the distance between the line cards and the buffered crossbar, to achieve 100% throughput under port-rate data flows. A long distance between these two components can make a buffered crossbar costly to implement. In this paper, we propose and examine two shared-memory crosspoint buffered packet switches that use small crosspoint buffers to support a long round-trip time, which is mainly affected by the transmission delay caused by the distance between line cards and the buffered crossbar. The proposed switch reduces the required buffer memory of the buffered crossbar by 50% or more. We show that a shared-memory crosspoint buffer switch can provide high this improvement without speedup.**

I. Introduction

Combined input-crosspoint buffered (CICB) switches are becoming attractive as an alternative to input-buffered switches to relax arbitration timing and to provide high-performance switching for packet switches with high-speed ports [3]. These packet switches use time efficiently as input and output arbitrations are performed separately [2]-[14] and memory speed needs to be no faster than that for input-buffered switches. In this paper, we consider that 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.

The amount of memory in a buffered crossbar is

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

where N is the number of input and 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), which is defined in [7] 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, $d2$. Cell and bit alignments are included in the transmission times. For example, the switch proposed in [7] requires the size of k be equal to or larger than the RTT to avoid crosspoint-buffer underflow or throughput degradation 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 required crosspoint-buffer size to avoid underflow by flows of data rate R_c b/s, where R_c is the port speed, is

$$RTT = d1 + OA + d2 + IA \leq k, \tag{2}$$

such that cells are transmitted continuously every time slot [7].

Furthermore, as the buffered crossbar can be physically located far from the input ports, actual RTTs can be long. To support long RTTs by a buffered-crossbar switch, the crosspoint-buffer size needs to be increased [9], such that up to RTT cells can be buffered. However, as the on-chip interconnection technology requires large real-state, the memory amount that can be allocated in a chip may be 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, the switch may have $k < RTT$, without supporting high data rates. This problem has been addressed by [10], which proposes a switch that supports p traffic classes with the crosspoint buffer size larger than RTT for a single class, and smaller than $p \times RTT$.

A solution to keep the crosspoint buffer small while supporting long RTTs and high data rates is needed. In this paper, we study a CICB switch that uses round-robin arbitration and credit-based flow control, named CIXB switch, under long round-trip times and high data-rate flows. We show the throughput degradation as a function of the round-trip time and the crosspoint buffer size.

This work is supported in part by National Science Foundation under Grants 0435250 and 0423305, and by NJIT under Grant 421070.

To reduce the memory amount or support longer RTT values, we propose a CICB switch that shares the crosspoint buffers, called the shared-memory crosspoint buffered (SMCB) switch, among m inputs. This switch uses shared memory in the crosspoint buffers to reduce the total crosspoint buffer size such that flows with high data rates can be handled with smaller amount of memory than a switch with dedicated buffers. We show that a SMCB switch supports a given round-trip time with half or less memory than a buffered crossbar with dedicated crosspoint buffers and deliver equivalent switching performance. Furthermore, we show that no speedup is needed when using the shared-memory approach. The matching scheme used in the proposed switches is round-robin [15] to have a fair comparison with the CIXB switch.

This paper is organized as follows. Section II describes the CIXB switch, which uses dedicated crosspoint buffers. Section III discusses the effect of long round-trip times in the CIXB switch. Section IV introduces two proposed SMCB switches. Section V presents the throughput performance of the SMCB switches under different situations of memory amount, traffic distributions, and data rates. Section VI presents the conclusions.

II. Combined Input-Crosspoint Buffered (CIXB) Switch

A buffered crossbar has N inputs and N outputs. A crosspoint (XP) element in the buffered crossbar that connects input port i to output port j is denoted as $XP(i,j)$.

There are N VOQs at each input. A VOQ at input i that stores cells for output j is denoted as $VOQ(i,j)$. The XP Buffer of $XP(i,j)$ is denoted as $XPB(i,j)$. The size of $XPB(i,j)$ is k cells, where $k \geq 1$.

A credit-based flow control mechanism indicates input i whether $XPB(i,j)$ is 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 each time a cell is sent to $XPB(i,j)$ and decreased by one each time that $XPB(i,j)$ forwards a cell to output j. If $XPB(i,j)$ can receive at least one cell, then $VOQ(i,j)$ is considered eligible by the input arbiter.

Round-robin arbitration is used at the inputs and output ports. An input arbiter at input i selects a $VOQ(i,j)$, among the eligible VOQs, to send a cell to XPB for output j at buffered crossbar. An output arbiter at output port j in the buffered crossbar selects a $XPB(i,j)$, among occupied XPBs from input i, to send a cell to output j.

III. Effects of Long Round-Trip Time and Limited k

To keep up with high data rates, switch ports must be able to handle flows of up to R_c b/s, where R_c is the data-rate capacity of a port (i.e., port-speed rate) of a switch or router In contrast, switches unable to support such flows can only handle aggregated data rates of R_c b/s, where each flow might have a data rate r_{single}, such that $r_{single} < R_c$. In a CICB switch (e.g., the CIXB switch presented in [7]), the maximum flow rate that the switch can handle is $R_c \frac{k}{RTT}$. Note that when $r_{f(i,j)} = R_c$, where $r_{f(i,j)}$ is the rate of $f(i,j)$, the maximum flow rate that the CIXB switch can transfer from inputs to outputs is equivalent to its achievable throughput.

We simulated the CIXB switch to observe the throughput obtained under different k and RTT values in a 32×32 switch, and to validate the traffic model to test the proposed architecture. Different from [7], we consider $RTT > 0$ in this paper. Here, we assume that the distances between input ports and the buffered crossbar are identical (the results in this paper also apply for non-identical distances). To model flows with different rates, we use the unbalanced traffic model [7].

The unbalanced traffic model uses the probability w, as the fraction of input load directed to a single pre-determined output, while the rest of the input load is directed to all outputs with uniform distribution. Let us consider input port s, output port d, and the offered input load for each input port ρ. The traffic load from input port s to output port d, $\rho(s,d)$ is given by,

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

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 s is destined for only output port d, where $s = d$.

Therefore, the fraction of R_c that $f(i,j)$ uses is $r_{f(i,j)} = w + \frac{1-w}{N}$. The maximum data rate of $f(i,j)$ is represented by setting $w = 1$ or $r^{max}_{f(i,j)} = R_c$, and the minimum data rate is represented when $w = 0$ or $r^{min}_{f(i,j)} = \frac{1}{N}$. We emphasize our observations in these two w values of the unbalanced traffic model.

Figure 1 shows that when flows have a rate $r_{f(i,j)} = r^{min}_{f(i,j)}$ (i.e., w=0) for different k values, such that $RTT - k < N$, the throughput is 100%, as shown by curves 1) and 5), where $RTT - k = 0$, and by curves 4) and 6), where $RTT - k = 31$. The uniform distribution of traffic relaxes the demand for buffer space, resulting in high throughput. The figure also shows that when $RTT - k \geq N$, the throughput is less than 100%, as shown by curve 2), where $RTT - k = 32$.

As the data rate of the flow increases (i.e., w), throughput degradation occurs. The worst-case scenario is observed when $r_{f(i,j)} = R_c$ b/s (i.e., w=1) where the achieved throughput

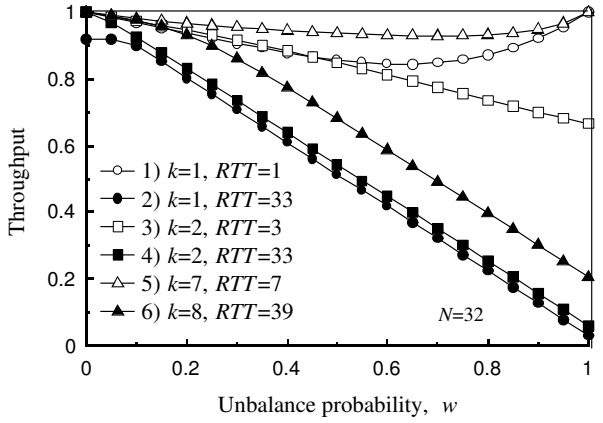

Fig. 1. Throughput performance of the CIXB switch [7] with $RTT > 0$.

is $\frac{k}{RTT}$ for $RTT - k > 0$, as curves 2)-6) show. Note that this case occurs when an input/output pair is used by a single flow for a period of time (or burst) as the simplest switching scenario, which however, may compromise the performance of a switch.

IV. Shared-Memory Crosspoint Buffered Switch (SMCB)

As discussed in Section III, the largest throughput degradation occurs when the $r_{f(i,j)} = R_c$ b/s, or $w = 1$ in the unbalanced traffic model. 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 $1 \leq m \leq N$. Here, we propose two SMCB switch architectures, a SMCB switch that uses dynamic memory allocation among the sharing inputs and speedup of 2, or SMCBx2, and a SMCB switch that uses no speedup and arbitrates the access of m inputs to the shared memory, or mSMCB.

A. Shared-Memory Crosspoint Buffered Switch with Dynamic Memory Allocation and Speedup=2 (SMCB×2)

This switch has N VOQs at each input, N^2 crosspoints and $\frac{N^2}{m}$ crosspoint buffers in the buffered crossbar. Each crosspoint buffer is shared by m inputs. Here, we denote each shared crosspoint buffer as SMB to differentiate from the notation of the CIXB switch. Each VOQ has a service counter, which counts the number of outstanding cells, and a counter limit $C_{i,j}^{max}$, which indicates the maximum number of cells that can be sent to the SMB. These counters are used by a credit-based flow control mechanism. A sharing control unit (SCU) at each SMB sets up the amount of memory (or threshold) of the shared memory for each input based on the VOQ occupancy. Since m inputs may need to access the shared memory at the same time, this architecture requires the shared memory to have a speedup of m. To minimize the speedup of the shared memory, we consider two inputs sharing a crosspoint buffer. A crosspoint in the buffered crossbar that connects input port i to output j is also denoted as $XP(i,j)$ as in the CIXB switch. The buffer for $XP(i,j)$ and $XP(i',j)$, 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 be used. Note that for switches with odd number of ports, one port is left with dedicated buffers of size 0.5 to 1.0 the capacity of a SMB.

Figure 2 shows the architecture of the switch with two inputs sharing the buffered crosspoint. The arbitration scheme for inputs and outputs is round-robin. The switch works as follows. The occupancy $Z_{i,j}$ of $VOQ(i,j)$ is sent to the corresponding SCUs. Based on the occupancy of the competing VOQs, the SCU at every SMB sets up the amount of the shared memory available for each input. The allocated amount of memory for an input is given as the maximum number of cells that a VOQ can send to the SMB. Table I shows the values of the allocated memory based on the input occupancy of the two VOQs sharing a SMB. These values set $C_{i,j}^{max}$ dynamically. The service counter is reduced by one when a cell at $SMB(q,j)$ of $VOQ(i,j)$ is served to the output. When the service counter reaches $C_{i,j}^{max}$, $VOQ(i,j)$ is inhibited of sending more cells to the SMB. In this architecture, the size of each SMB in number of cells is k_s, where $k_s \geq RTT$.

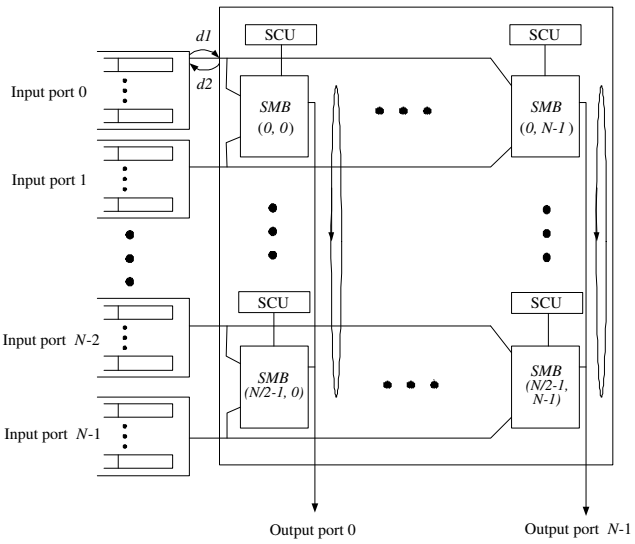

Fig. 2. $N \times N$ SMCB×2 switch.

There are four possible occupancy states for $VOQ(i,j)$: $Z_{i,j} = 0$, $Z_{i,j} \leq \frac{RTT}{2}$, $\frac{RTT}{2} < Z_{i,j} \leq RTT$, and $Z_{i,j} >$

TABLE I

MEMORY ALLOCATION OF A SMB IN THE SMCB×2 SWITCH.

$Z_{i,j}$	$Z_{i',j}$	$C^{max}_{i,j}$	$C^{max}_{i',j}$
0	0	0	0
$[0, RTT)$	0	$Z_{i,j}$	0
$[RTT, \infty)$	0	RTT	0
$[0, RTT/2]$	$[0, RTT/2]$	$RTT/2$	$RTT/2$
$(RTT/2, \infty)$	$[0, RTT/2]$	$RTT - Z_{i',j}$	$Z_{i',j}$
$(RTT/2, \infty)$	$(RTT/2, \infty)$	$RTT/2$	$RTT/2$

RTT.

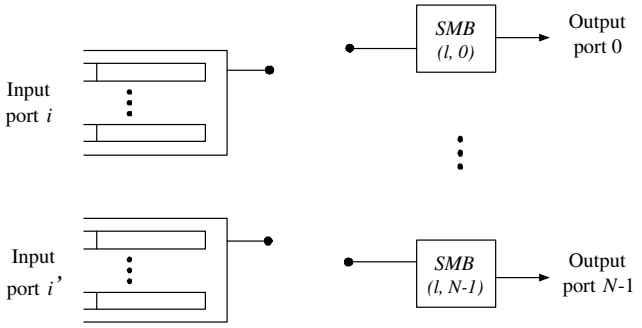

Fig. 4. Bipartite matching in an input-access scheduler.

B. Shared-Memory Crosspoint Buffered Switch with Input-Crosspoint Matching (mSMCB)

To eliminate the speedup at SMBs, only one input is allowed to access a SMB at a time. To schedule the SMB access between two inputs that are physically separated, an input-access scheduler is used among the m inputs that share N SMBs. Figure 3 shows the architecture of the mSMCB switch when $m = 2$. The size of a SMB, in number of cells that can be stored, is k_s. There are $\frac{N}{m}$ input-access schedulers in the buffered crossbar, each denoted as $S(q)$. An input-access scheduler matches non-empty inputs to the SMBs that have room for storing a new cell.

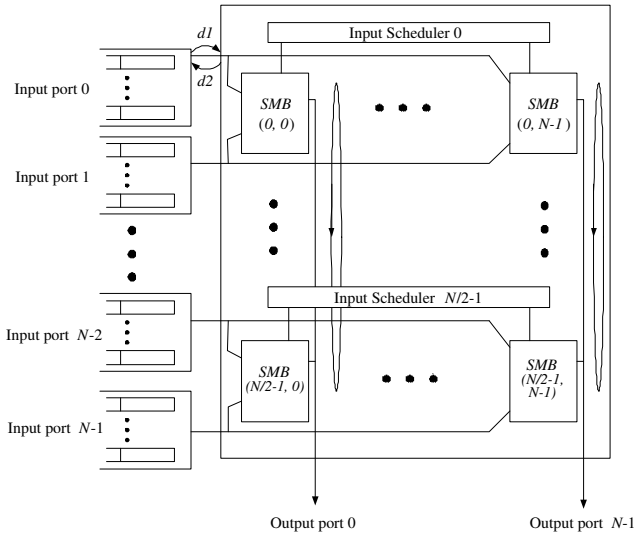

Fig. 3. $N \times N$ mSMCB switch, $m = 2$.

The input-access scheduler performs a matching process among the shared-crosspoint buffers and the inputs that share them. Figure 4 shows the inputs and the shared crosspoint buffers that participate in the matching process. In this paper, the matching follows a three-phase process, as performed for input-buffered switches. The matching scheme used in this switch is round-robin based [15] to have a fair comparison with the CIXB switch. However, other matching schemes can be used.

At each output j in the buffered crossbar, there is an output arbiter to select the outgoing cell from non-empty XPs. An output arbiter considers up to two cells from each SMB, where each cell belongs to one input. The output arbiter is represented as a loop in Figure 3.

The way the mSMCB switch works is as follows. Cells destined to output j arrive at $VOQ(i, j)$ 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 matching. 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. The output arbiter uses round-robin selection.

The matching performed by $S(q)$ among the shared-crosspoint buffers and the inputs works as flow control as it controls the forwarding of cells according to the buffer availability, as $S(q)$ considers eligible those VOQs whose corresponding SMBs have available room.

V. THROUGHPUT OF THE SMCB SWITCH

We compare the switching performance of two 32×32 switches, SMCB×2 and mSMCB. We show the throughput performance and average cell delay of a CIXB switch and the SMCB switches. The traffic considered has Bernoulli and bursty arrivals with uniform distribution, and Bernoulli arrivals with unbalanced distribution.

A. Uniform Traffic

Figure 5 shows average cell delay of the CIXB, SMCBx2 and 2SMCB switch under uniform traffic. We use $k = 2$ for the CIXB switch, and $k_s = 2$ for SMCBx2 and 2SMCB switches. By using these crosspoint buffer sizes, the amount of memory used in SMCB switches is half the amount of memory in the CIXB switch for a given cell size. The average delay of SMCB switch only considers the queuing delay. Figure 5 shows that the average cell delay of SMCBx2 and 2SMCB switches is similar. Furthermore, the average cell delay of the SMCB switches shows similar magnitude to that of the

CIXB switch without the effects of RTT (i.e, $RTT = 0$). The larger average delay shown by SMCB switches at loads from 0.1 to 0.8 are a constant time slot because the VOQs notify the input-access scheduler when a new cell arrives, and a the VOQ is matched before forwarding a cell. This process always takes at least one time slot. For loads over 0.9, the SMCB switches have the average delay similar to that of the CIXB switch. The average delay of all switches under bursty traffic with an average burst length l has similar magnitude. The small advantage of the CIXB switch is caused by the larger memory amount provisioned in the buffered crossbar. Therefore, the SMCB switches, with $m = 2$, have equivalent performance under uniform traffic while using half memory amount of the CIXB switch.

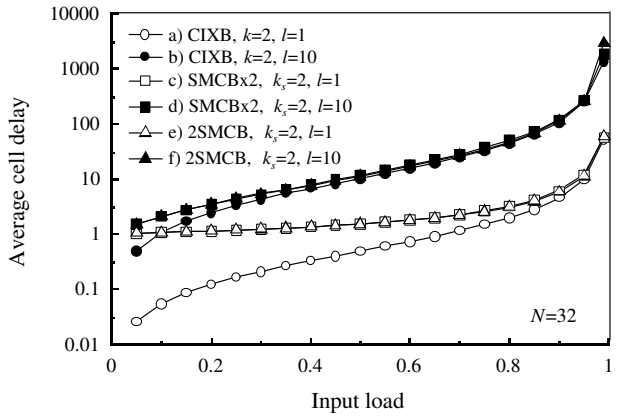

Fig. 5. Average queuing delay of a 32×32 SMCBx2 and 2SMCB switches.

B. Unbalanced Traffic

Figure 6 compares the throughput of the SMCBx2 and 2SMCB switches. The two switches have the same throughput performance. The following simulations only consider the 2SMCB switch.

We observe the effect of long RTTs in the proposed switches by measuring the switch throughput under the unbalanced traffic model, as in Section III. Figure 7 shows the throughput performance of the 2SMCB switch, with $k_s \geq 1$. This switch has a symmetric throughput when $w = 0$ and $w = 1$ or $r_{f(i,j)} = r^{max}_{f(i,j)} = r^{min}_{f(i,j)}$, and achieves 100% throughput for $k_s - RTT \geq 0$, as the figure shows in all curves, except for b) and d), which have $k_s - RTT < 0$. For these values of w, the throughput can be 100% using half of the total amount of memory used by the CIXB switch.

For the other values of w, we see that for $k_s = k$ the throughput is similar. This is because the buffered crossbar switches seems to have small sensitivity to the crosspoint buffer size. The decreased throughput around $w = 0.6$ in

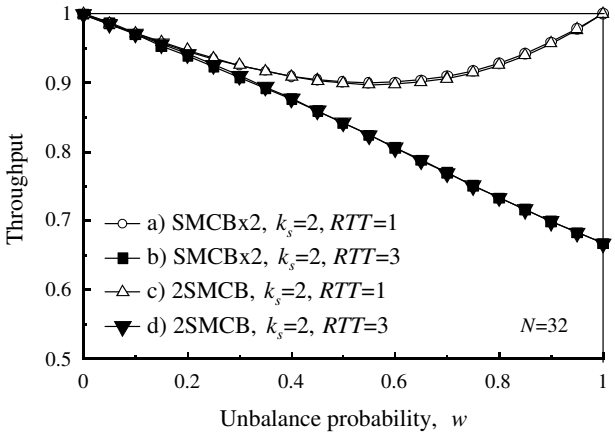

Fig. 6. Throughput of SMCBx2 and 2SMCB switches with same amount of memory.

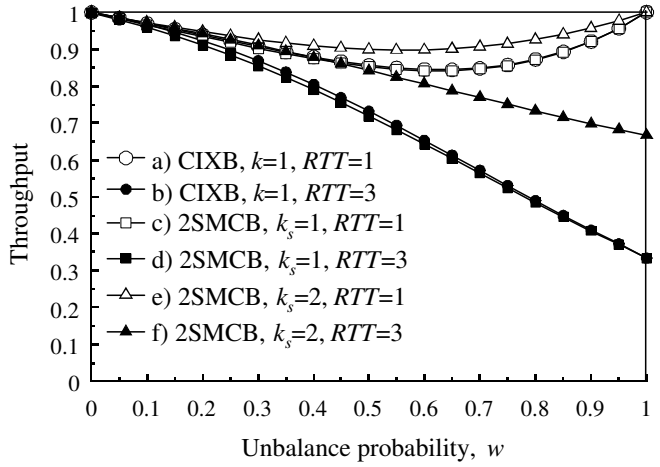

Fig. 7. Throughput of a 2SMCB switch with half ($k_s = 1$) and equal ($k_s = 2$) amount of memory of the CIXB switch.

curves a) and c), where k_s, $k \geq RTT$, is the result of having a limited and small buffer size, mixed traffic (the high data-rate flow is mixed with a large number of low data-rate flows) as described in Section III, and round-robin arbitration. In these cases, a more elaborate arbitration scheme [16] can be used to improve the throughput for small $k_s - RTT$ values.

As seen in curves e) and f) in Figure 7, when the 2SMCB switch have the same amount of memory of the CIXB switch (i.e., $k_s = 2k$) in the buffered crossbar, the throughput of the 2SMCB switch is higher than that of the CIXB switch under the same RTT values.

Figure 8 shows the throughput of the 32×32 mSMCB switch with m inputs sharing the buffered crosspoint, where $1 < m \leq N$. With the same k_s value, the total amount of memory is $\frac{1}{m}$ of a CIXB switch. The throughput of the mSMCB switch still achieves 100% when $w = 0$ and $w = 1$.

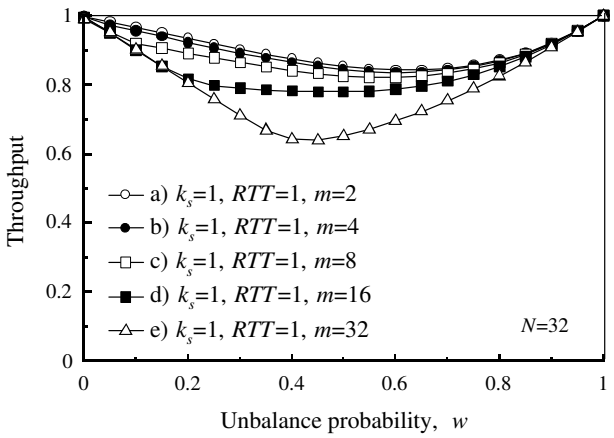

Fig. 8. Throughput of the 32×32 mSMCB with m inputs sharing the buffered crosspoint.

VI. Conclusions

We presented the effect of long round trip times RTTs, where the crosspoint buffer size k is such that $k < RTT$, in a combined input-crosspoint buffered switch. We observed that switches based on buffered crossbars, with the 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 proposed two switches that shared the crosspoint buffers among m inputs, such that RTT can be m times as long as that supported by a CICB switch with dedicated buffers without decreasing switching performance, and providing 100% throughput for port-rate flows and under uniform traffic. Therefore, these switches relax the amount of memory to $\frac{1}{m}$ of the amount required by a CICB switch with dedicated buffers. We showed that the higher performance is achieved when $m = 2$. In addition, the performance study shows that the shared memory used in the crosspoint buffers needs no speedup to provide high throughput.

References

[1] M. Karol, M. Hluchyj, "Queuing in High-performance Packet-switching," *IEEE J. Select. Areas Commun.*, vol. 6, pp. 1587-1597, December 1988.

[2] S. Nojima, E. Tsutsui, H. Fukuda, and M. Hashimmoto, "Integrated Packet Network Using Bus Matrix," *IEEE J. Select. Areas Commun.*, vol. SAC-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, 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 Hot Interconnects, 2002. 10th Symposium on*, pp. 5-14, Aug. 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, Dec. 2003

[11] M. Katevenis, G. Passas, D. Simos, I. Papaefstathiou, N. Chrysos, "Variable Packet Size Buffered Crossbar (CICQ) Switches," *Proceedings of IEEE ICC 2004*, vol. 2 , pp. 1090-1096, June 2004.

[12] 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.

[13] 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.

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

[15] N. McKeown, "The iSLIP Scheduling Algorithm for Input-Queue Switches", *IEEE/ACM Trans. Networking*, vol. 7, no. 2, pp. 188-200, April 1999.

[16] 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.