Efficient Queuing Architecture for a Buffered Crossbar Switch

MICHAEL S. BERGER
Department of Communications, Optics and Materials
Technical University of Denmark
Oersteds Plads 1, 2800 Kgs. Lyngby
DENMARK
http://www.com.dtu.dk

Abstract: - This paper presents a modified architecture for a buffered crossbar switch that overcomes the memory bottleneck with only a minor impact on performance. The proposed architecture uses two levels of backpressure with different constraints on round trip time. Buffered crossbars are considered an alternative to bufferless crossbars mainly because the latter requires a complex scheduling algorithm that matches input with output. Buffered crossbars require only a simple scheduler that operates independently for each output queue column. The memory amount required for a buffered crossbar is proportional to the square of the number of ports and the round trip time. The proposed architecture reduces the amount of memory in the buffered crossbar without increasing the scheduling complexity.

Key-Words: - Buffered Crossbar, Packet switching, SpeedUp

1 Introduction

Crossbar switch fabrics have been studied extensively in the literature. In combination with Virtual Output Queuing (VOQ), the architecture provides a scalable solution with respect to memory access bandwidth. The crossbar can be either unbuffered or contain a small amount of buffering in each crosspoint. A bufferless crossbar requires a complex scheduling mechanism that matches input with output. The scheduling algorithm can either calculate a maximum match or a maximal match. A maximum match algorithm pairs the maximum number of input and output, whereas a maximal match has no cell in any input queue destined to an unmatched output. A number of maximum weight matching algorithms have been presented in [1]. Their main disadvantage is timing complexity, leading to an interest for maximal matching algorithms such as PIM and iSLIP [2]. SLIP matches input with output by having a round-robin scheduler for each input and output. The input schedulers independently select an output, and the output scheduler selects among contending inputs. The iterative SLIP, iSLIP, performs a number of iterations of SLIP. To compensate for the lower performance of a maximum matching algorithm, speedup can be introduced between the VOQs and the crossbar. A speedup of 2 is sufficient to obtain 100 % throughput [3].

Due to the complexity of scheduling algorithms for bufferless crossbars, buffered crossbars are considered as an alternative. By adding a small buffer capacity in each crosspoint, it is possible to perform the scheduling decision independently among the output columns. The crosspoint buffers generate backpressure signals towards the VOQs in the port card to avoid overflow. The minimum crosspoint buffer size to maintain full throughput is determined by the round trip delay for the backpressure mechanism. As an alternative to small crossbar buffers in combination with VOQ, one may consider pure crosspoint buffering, however, this requires large buffer capacity in each crosspoint to reduce cell loss.

It has recently been shown that a buffered crossbar switch with a speedup of 2 can emulate a pure output queued switch [4]. A similar result is available for bufferless crossbars: Emulation of an output buffered switch of size NxN is obtainable with a speedup of 2-1/N 6[5]. The emulation algorithm proposed in [5] is, however, much more complex than the one proposed in [4]. This result indicates that QoS is more easily supported in the buffered crossbar architecture.

The performance of buffered crossbars with VOQ has been studied in various papers. The architecture was originally proposed in [6] where a simple round-robin scheduling scheme was compared to a more advanced scheme taking into account buffer size and cell age. In [7], a stability analysis is performed for a CICQ (Combined Input and Crosspoint Queued) switch with one cell sized crosspoints. The switch uses Longest Queue First VOQ schedulers. Different combinations of scheduling algorithms are compared in [8]. Longest Queue First, Oldest Cell First and round-robin were

considered for VOQ scheduling in combination with Oldest Cell First and round-robin for the crossbar. The paper concludes that the performance is quite similar and recommends the round-robin approach due to its simplicity. The combined input one cell crosspoint buffered switch (CIXB) is compared to iSLIP and pure output queuing (OQ) in [9]. The delay performance of CIXB is better than iSLIP and close to that of an OQ switch. For unbalanced traffic, that is traffic with an uneven distribution of destinations, the CIXB will not support 100 % throughput even if the traffic is admissible. The throughput for unbalanced traffic is, however, better for CIXB compared to iSLIP. In [11] the study is extended to cover more than one buffer location in the crosspoints. Due to the round trip time for backpressure signals, a single buffer location in each crosspoint is not feasible. The high memory consumption is the main drawback of this architecture, and the results are mainly interesting from a theoretical point of view.

Another benefit of a buffered crossbar compared to a bufferless crossbar is the less stringent synchronisation requirement between the port cards and the switch cards [12]. Bufferless crossbars require that all port cards are synchronized to the same clock.

This paper presents a modification to the buffered crossbar architecture that overcomes the memory problem with only a small impact on performance. Each crosspoint buffer is reduced to a minimum size of one cell, and a small, shared VOQ memory is added in front of each row of crosspoints. This configuration requires two levels of backpressure; a fast mechanism between the small crosspoints and the on-chip crossbar VOQs, and a slower mechanism between the VOQs in the port card and VOQs in the crossbar. The switch architecture is described in more detail in section 2. The simulation study presented in section 3 compares the performance of this switch architecture to a standard buffered crossbar system. Moreover, the simulation study is used as a guideline for system dimensioning. Finally, concluding remarks are given in section 4.

2 Switch Model

A crossbar buffered switch system CIXB of size N *x* N consists of N Input/Output port cards and a switch card implementing the N^2 crosspoint buffers as shown in Figure 1. Each input port card contains VOQs with one buffer for each of the N outputs. The switch model uses round-robin scheduling between VOQs in the port cards and also between crosspoint buffers in an output column. The output port card contains a buffer to store cells in case of speedup. In order to avoid overflow, the crossbar buffers will generate a backpressure signal towards the corresponding VOQ buffer in the port card. The round trip time for backpressure RT_BP is defined as the number of timeslots it takes to stop the cell flow to a specific crosspoint buffer measured from the time when backpressure is asserted by that crosspoint. The round trip time is composed of a propagation delay for the backpressure signal, the time it takes before the port card scheduler is blocked, and the data path delay from the port card scheduler to the crosspoint. To achieve 100 % throughput, the minimum crosspoint buffer size is $2*RT\_BP$. Backpressure is then asserted if the buffer level is larger than or equal to RT_BP.

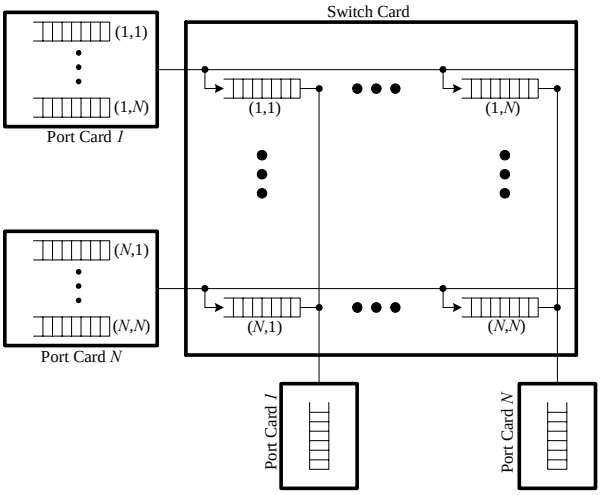

Figure 1: CIXB switch

The main advantage of the CIXB switch architecture shown in Figure 1 is the low scheduling complexity between crosspoint buffers in an output column. A simple round-robin scheduler can be implemented very efficiently [13]. However, the total amount of storage is $O(2 \cdot RT\_BP \cdot N^2)$. With a round trip time of four, a *32 x 32* switch will contain 4M memory bits for a packet size of 64B. Having e.g. eight traffic priorities results in 32 M bits memory, which is a very large amount, and this is not feasible on a single chip.

From a performance point of view, the CIXB switch behaves like an output buffered switch for very large crosspoint buffers, and 100% throughput is achieved for all admissible traffic patterns. This is, however, not the case for limited size crosspoint buffers. In [9], the reduction in switch throughput has been investigated for unbalanced traffic. To increase throughput, a speedup can be introduced between the port card and the switch card. In the following,

this is referred to as *External Speedup*. The egress port card must then contain buffering to adapt between the different rates, as shown in Figure 1.

Figure 2 presents a modified switch card architecture that uses a smaller amount of memory compared to CIXB. The objective of the proposed architecture is to reduce the crosspoint queues to a minimum possible size of one cell. This is achieved by adding an additional queue system in front of each crosspoint row. The new queue system has a dedicated VOQ for each crosspoint in that row. The VOQs are implemented in a shared memory following e.g. a linked list approach. In the following, the system is denoted CISXB.

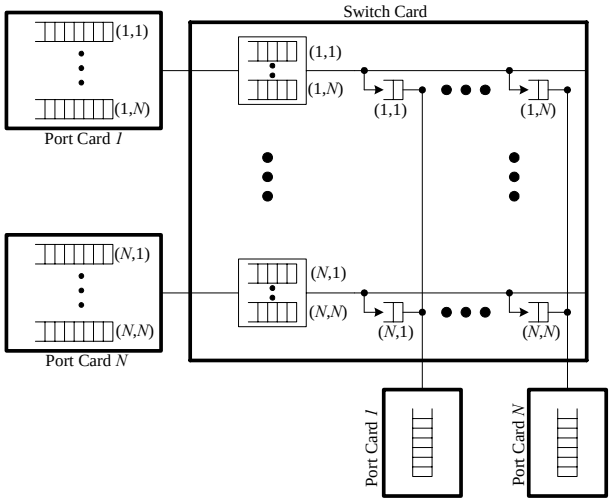

Figure 2: CISXB switch

The CISXB switch requires two levels of backpressure; the first level controls the filling of the crosspoint buffers, and the second level controls the filling of the shared memory. With a crosspoint buffer size of only one cell, the first level of backpressure must have very low latency. This can be achieved because the crosspoint buffers and the shared memory reside on the same switch chip. Each crosspoint in a row generates backpressure towards the corresponding VOQ in the shared buffer. The backpressure between the VOQs in the shared memory and the VOQs in the port cards has a higher round trip delay equal to *RT_BP* for the CIXB switch in Figure 1. The VOQs on the switch card share memory, which implies that the VOQs can accommodate buffer space for a limited number of destinations at the same time. If the shared memory occupancy exceeds a global threshold, then a global backpressure signal concerning all VOQs in the shared memory is generated.

The CISXB architecture requires an additional scheduler to select between the VOQs in the shared memory. In this work, a round-robin scheduling mechanism is considered. The switch chip has then two levels of round-robin arbitration; first among the VOQs in the shared memory, and then among the crosspoints in an output column. Other scheduling algorithms could be considered, but the main objective in this paper is to compare the CIXB and the CISXB architectures.

In the following, the number of cells in VOQ number i in the shared memory is denoted Q_i. The backpressure threshold for queue i is B_i, that is, a backpressure signal is generated if $Q_i \geq B_i$. Due to the round trip time for backpressure signals, the size of queue i can grow to $Q_{i,\max} = B_i + RT\_BP$. The total number of cells in the shared buffer is $Q = \sum Q_i$. The total capacity of the shared memory S is typically much smaller than $\sum Q_{i,\max}$, therefore a global backpressure threshold B is introduced to avoid queue overflow. The global backpressure signal is then asserted if $Q \geq B$. The global threshold must be selected such that $B + RT\_BP \leq S$ in order to avoid overflow in the shared buffer.

The total memory capacity of the CISXB switch chip in Figure 2 can be reduced compared to the CIXB in Figure 1 if the added size of the shared memory is smaller than the reduction in crosspoint memory size. In principle, the shared buffer could be as small as $2 \cdot RT\_BP$ to avoid queue overflow, however, the performance of the switch will suffer from frequent blocking due to the global backpressure signal. The size should be large enough to reduce the global blocking to a minimum; the CIXB switch can achieve 100 % throughput for uniform Bernoulli traffic [9[. The CISXB switch will not be able to support 100 % throughput if the global backpressure is invoked because the transmission from the port card is blocked completely. In general, the size of the shared buffer will depend on the traffic profile (e.g. uniform, bursty) and the load. In section 3, this subject is investigated further by a simulation study.

The performance of the CISXB switch can be increased by internal speedup between the shared buffers and the crosspoint buffers in the switch card. With an internal speedup of *IS*, the round-robin scheduler for the shared queue performs *IS* scheduling decisions for each decision of the crosspoint column scheduler. Internal speedup will not affect the bandwidth between the port cards and

the switch card, but the internal bandwidth between the shared memory and crosspoint memory must be *IS* times higher. Note that the system behaves like an output buffered switch if *IS*=*N* and the shared buffer is sufficiently large. An internal speedup of more than 2 is, however, seldom feasible.

3 Simulation and Results

A simulation study has been carried out in order to compare the CISXB switch with a well known buffered crossbar system CIXB. Each port card receives cells from a source. In each timeslot, the source generates a cell with probability equal to the load ρ. The switch size is 32x32. The destination is selected randomly according to a uniform distribution. Assigning the same destination to a number of consecutive cells generates bursty traffic. Figure 3 shows the average delay as a function of load for a burst length of 0, 10 and 20, respectively.

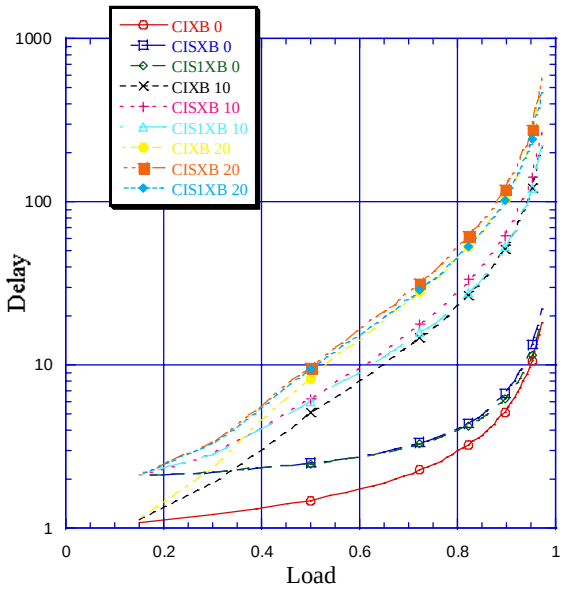

Figure 3: Delay vs. Load. The burst size is 0, 10 and 20, respectively

The round trip time for backpressure is set to four for both CIXB and CISXB. The backpressure threshold for each individual queue in the shared input buffer is set to the minimum value of four. The global threshold for the shared buffer is set to 92, which has shown to be sufficiently large to avoid global backpressure during all simulations. The total size of the shared buffer is then 92+4 = 96. The crosspoint size of the CIXB switch is at the minimum possible size of eight positions. The total number of cell positions in CIXB is then 8*32*32 = 8192. The number is 96*32 + 1*32*32 = 4096 for the CISXB switch, i.e. half the amount. For each burst length, the plot shows the delay for CIXB, CISXB and CISXB with an internal speedup of two, CIS1XB. The average delay of CISXB is a little larger than of CIXB. For small load values, the delay of CISXB is close to that of CIS1XB because the average number of cells in the shared crossbar buffer is very small, and consequently, there is no gain in having internal speedup. For large load values there is a reduction in delay from internal speedup. In this case, the delay of CIS1XB is close to that of CIXB. Since the average delay for an output buffered switch is very close to that of a CIXB, the gain in performance from further increasing the speedup is limited.

The larger delay of CISXB is mainly due to delay in the shared input buffer. The average size of the shared input buffer is depicted in Figure 4 for burst lengths of 0, 10 and 20, respectively. The buffer size depends strongly on the load, but not on the burst size. For load values close to 100 %, the required buffer size is quite large. The switch card load is reduced by use of external speedup between the port card and the switch card. Thereby, the shared buffer size is reduced as well. If the external speedup is e.g. 1.5, then the maximum load of the switch card becomes 66,7 %. From Figure 4, it is seen that the average buffer size is below 10 even for a burst size of 20. Note that the difference in buffer size between a burst length of 10 and 20 is rather small compared to the difference between 0 and 10. This indicates that the size of the buffer only slightly depends on the traffic burstiness. In the following, the dependence on bursty traffic is investigated in more detail.

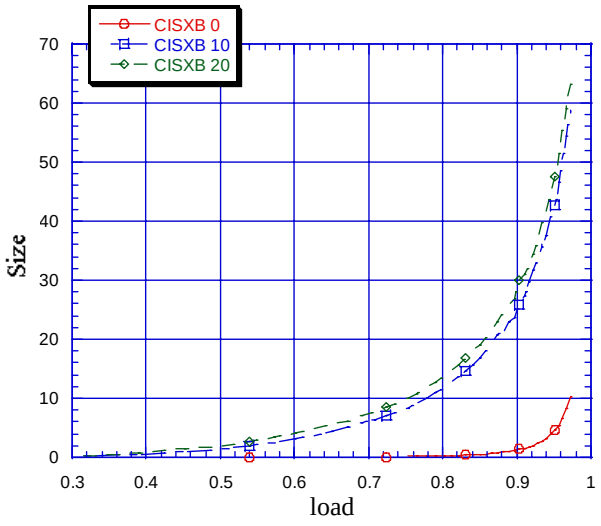

Figure 4: Size (occupancy) of shared input buffer vs. load. The burst size is 0, 10 and 20, respectively

As explained in section 2, the size of the shared crossbar buffer should be sufficiently large to ensure that the probability of global backpressure is minimized. In order to determine the size of the shared crossbar buffer, the average buffer size as a function of burst length has been determined. The results are shown in Figure 5. The figure shows both the average buffer size in the port card and the shared memory size in the switch card. The load values are 70 %, 80 % and 90 %. The size of the shared crossbar buffer increases rather slowly with the burst size. A detailed investigation of the plot shows that the size grows only logarithmically as a function of burst size. This result is used to dimension the buffer by taking only the system load into account. Assuming an external speedup of 1.5 (load = 66,7 %), the average shared buffer size will be below 10 according to Figure 5. By allocating 32 buffer locations, global backpressure becomes extremely rare even for very bursty traffic. The total number of memory locations for the CISXB architecture then becomes 32*32 + 1*32*32 = 2048, a reduction of 75 % compared to the CIXB architecture. In average, only 2 cells per crosspoint are needed. The performance in terms of cell delay is only slightly degraded. For unbalanced traffic, the shared buffer size requirement becomes even lower even if the overall throughput is reduced so the system behaves properly under both bursty and unbalanced traffic scenarios.

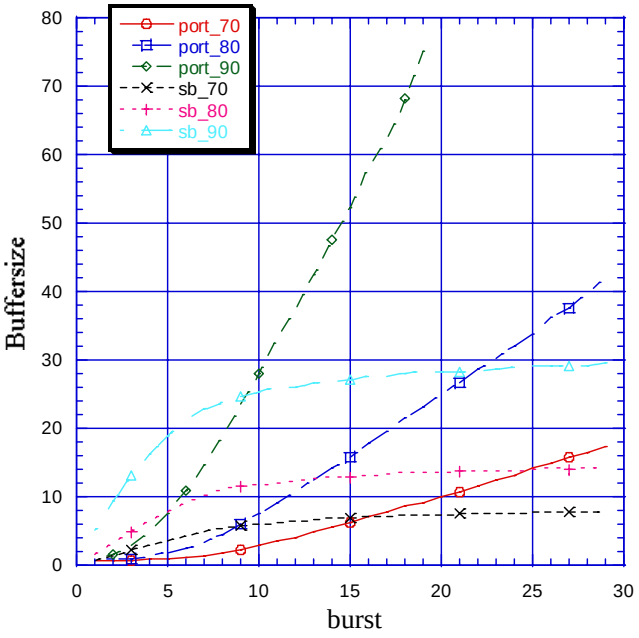

Figure 5: Size (occupancy) of shared input buffer and port card buffer vs. burst size. The load is 70 %, 80 % and 90 % respectively

In the previous discussion a speedup of 1.5 was taken as an example without further explanation. The switch behaviour under unbalanced traffic is now investigated in more detail. The model for unbalanced traffic from [9] is used below. The model is commonly used to describe unbalanced traffic [10].

The unbalanced weight ω defines the degree of unbalance. The load from input port s to output port d is denoted $\rho_{s,d}$:

$$\rho_{s,d} = \begin{cases} \rho(\omega + \frac{1-\omega}{N}) & \text{If } s=d, \\ \rho(\frac{1-\omega}{N}) & \text{Otherwise} \end{cases}$$

Note that

$$\sum_s \rho_{s,d} = \sum_d \rho_{s,d} = \rho$$

The traffic matrix is thus admissible, and all input and output have a load equal to ρ. If $\omega = 0$, there is no unbalance, and if $\omega = 1$, the traffic is completely unbalanced.

Figure 6 shows the performance degradation under unbalanced traffic. The switch parameters are identical to those used in Figure 3. The throughput penalty is highest for the CISXB switch. The throughput for a CIXB switch with 1-cell crosspoint buffers is shown in [9], and the result is quite close to that of CISXB shown in Figure 6. The CIXB with 8-cell crosspoint buffers has a smaller reduction in throughput according to Figure 6. It is concluded that the throughput reduction for unbalanced traffic mainly depends on the crosspoint buffer size. Internal speedup between the shared input buffer and the 1-cell crosspoints increases performance; the throughput for CIS1XB is slightly better than for CIXB. The CIS1XB switch, however, requires double internal speed and data path bandwidth so it is more feasible to use CISXB with a slightly higher external speedup.

The throughput for iSLIP with four iterations is shown in [9]. The minimum throughput is 0.8, which is lower than for CISXB with a minimum value around 0.85 according to Figure 6.

To compensate for the throughput reduction of unbalanced traffic for CISXB, an external speedup is required. To equalise the difference in throughput between CIXB and CISXB, the CISXB must have speedup that is approximately 10 % higher compared to CIXB. Also, with a 10 % increase in speed, the delay performance of CISXB will reach that of CIXB according to Figure 3.

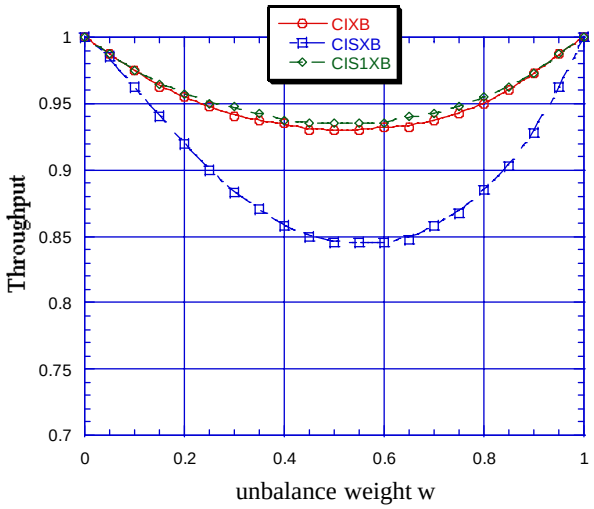

Figure 6: Throughput with unbalanced traffic

The results obtained above are valid for a single priority only. However, it is believed that the reduction in memory would be even higher if the switch supports more than one priority: For the pure buffered crossbar, the memory consumption will become *P times* higher with *P* priorities in the system. The modified architecture with 2 levels of backpressure will also require *P* times more memory for the one-cell crosspoint buffers, but the size of the shared input buffer is expected to be less than *P* times higher. The exact buffer requirements for the shared input buffer in case of priorities are for further study.

The proposed architecture solves another potential problem related to large switching systems covering several racks. This could lead to round trip times between port cards and switch card much higher than four, which was assumed in the simulations. This will lead to very high memory consumption for a buffered crossbar switch, whereas the modified architecture requires only additional memory in the shared input buffer to compensate for the increased delay. Also, for a bufferless crossbar with iSLIP scheduling, it has been shown that large round trip delays lead to a significant drop in throughput [12]. However, the reduction in throughput could be prevented by a modification to iSLIP so that only packet arrivals (and not the whole state of the VOQs) are send to the central scheduler. Of course this introduces the usual problem of reliability when only state changes are communicated.

4 Conclusion

Buffered crossbars have several advantages compared to non-buffered crossbars including simpler arbitration, synchronisation relaxation and better performance. The main drawback, however, is the total amount of crossbar memory, which is proportional to the square of the number of input/output ports and backpressure latency.

This paper introduced a new architecture for a buffered crossbar that uses two levels of backpressure to reduce the amount of memory used in the switch card. The proposed switch uses a small, shared VOQ memory in combination with a one-cell deep crosspoint buffer. Each shared queue system uses independent schedulers so the time complexity of the arbitration is identical to that of a pure crossbar buffered switch card.

The performance has been investigated by a simulation study. It was shown that the amount of memory is reduced significantly with only a small reduction in performance. The switch was insensitive to the burstiness of traffic, and the study shows that a reduction of 75 % in memory could be obtained for a 32x32 switch with a backpressure round trip time of 4 timeslots. The performance reduction can be compensated by an additional speedup of 10 %, or it can be compensated by an internal speedup of 2 between the shared input buffer and the crosspoint buffers. It is also expected that the memory savings will become even higher in a switch containing several priorities.

References:

[1] N. McKeown, A. Mekkittikul, V. Anantharam, J. Walrand, “Achieving 100% throughput in an input-queued switch”, IEEE Transactions on Communications, Vol.47 Issue.8 1999.

[2] N. McKeown, “The iSLIP scheduling algorithm for input-queued switches”, IEEE/ACM Transactions on Networking, p 188 -201, Vol.7 Issue.2, 1999.

[3] J. G. Dai, B. Prabhakar, “The throughput of data switches with and without speedup”, Proceedings IEEE INFOCOM 2000, p556-64 vol.2 2000.

[4] R. B. Magill, C. E. Rohrs, R. L. Stevenson, "Output-queued switch emulation by fabrics with limited memory", IEEE Journal on Selected Areas in Communications, Volume: 21, Issue: 4, May 2003.

[5] S. T. Chuang, A. Goel, N. McKeown, B. Prabhakar, "Matching Output Queuing with a Combined Input Output Queued Switch", IEEE Journal on Selected Areas in Communications, vol.17, n.6, Dec.1999, pp. 1030-1039.

[6] M. Nabeshima, “Performance evaluation of a combined input- and crosspoint-queued switch”,

IEICE Transactions on Communications, Vol.E83-B Issue.3, p737-41, 2000

[7] T. Javidi, R. Magill, T. Hrabik, "A high-throughput scheduling algorithm for a buffered crossbar switch fabric", Proceedings of IEEE ICC 2001, p 1586-1591, vol.5.

[8] I. Radusinovic, M. Pejanovic, Z. Petrovic, "Impact of scheduling algorithms on performances of buffered crossbar switch fabrics", Proceedings of IEEE ICC 2002, p 2416-2420, Vol.4

[9] R. Rojas-Cessa, E. Oki, Z. Jing, H. J. Chao, "CIXB-1: combined input-one-cell-crosspoint buffered switch", Proceedings of IEEE HPSR 2001, p 324-329.

[10] L. Mhamdi, M. Hamdi, "MCBF: A High-Performance Scheduling Algorithm for Buffered Crossbar Switches", IEEE Communications Letters, Vol. 7, No. 9, September 2003.

[11] R. Rojas-Cessa, E. Oki, H.J. Chao, "CIXOB-k: combined input-crosspoint-output buffered packet switch", Proceedings of IEEE GLOBECOM 2001, p 2654-2660, Vol.4

[12] F. Abel, C. Minkenberg, R. P. Luijten, M. Gusat, I. Iliadis, "A Four-Terabit Single-Stage Packet Switch with Large Round-Trip Time Support", 10th Symposium on High Performance Interconnects HOT Interconnects, p.5, 2002

[13] P. Gupta, N. McKeown, "Designing and implementing a fast crossbar scheduler", IEEE Micro, Vol.19 Issue.1, p 20-28, 1999.