Matching Schemes with Captured-Frame Eligibility for Input-Queued Packet Switches

Roberto Rojas-Cessa and Chuan-bi Lin

Abstract— **Virtual output queues (VOQs) are widely used by input-queued (IQ) switches to eliminate the head-of-line (HOL) blocking phenomena, which limits switching performance. An effective matching scheme must provide high throughput under several admissible traffic patterns and keep implementation complexity low. A variety of matching schemes for IQ switches that deliver high throughput under uniform traffic have been proposed. However, there is a need of matching schemes that provide high throughput under several admissible traffic patterns, including those with nonuniform distributions. In this paper, we introduce the captured frame-size concept for matching schemes in IQ switches. We use the captured-frame eligibility concept in a round-robin based scheme, uFORM, and in a random-base scheme, uFPIM, to improve switching performance under nonuniform traffic patterns. The uFPIM scheme is based in the parallel iterative matching (PIM) scheme and shows the throughput improvement achieved with the captured frame concept. The uFORM scheme provides high performance under nonuniform traffic while keeping the high performance that round-robin schemes are known to have under uniform traffic.**

Index Terms— **input-queued switch, round-robin matching, virtual output queue, captured frame, frame eligibility, service frame, unbalanced traffic**

I. Introduction

Virtual output queues (VOQs), where one queue per output port is allocated at an input port of an input queued (IQ) packet switch, is known to remove the head-of-line (HOL) blocking problem. HOL blocking causes idle outputs to remain so, even in the existence of traffic for them at an idle input, thus impeding high throughput delivery [1].

This paper follows the common practices in packet switch design: 1) segmentation of incoming variable-size packets at the ingress side of a switch to perform internal switching with fixed-size packets, or cells, and re-assembling the packets at the egress side before they depart from the switch; 2) use of VOQs, to avoid head-of-line (HOL) blocking [1]; and 3) use of crossbar fabrics for implementation of packet switches because of their non-blocking capability, simplicity, and market availability.

One major requirement for an input-queued switch is the delivery of high throughput under different traffic conditions. We consider admissible traffic [2], with Bernoulli arrivals and uniform and nonuniform distributions. The matching scheme used in crossbar-based IQ switches determines in large measure the achievable throughput.

Maximum weight matching (MWM) schemes have been used to show that IQ switches with VOQs can provide 100% throughput under admissible traffic [2], but MWM schemes have intrinsically high computation complexity that is translated into long resolution time and high hardware complexity. An alternative is to use maximal-weight matching schemes. However, the hardware and time complexity of these schemes can be considered high for the ever increasing data rates, and a large number of iterations may be needed to achieve satisfactory matching results. Maximal-size matching are another way to resolve contention in IQ switches [3]. Schemes based in round-robin matching, such as iSLIP [4], DRRM [5], [6], and SRR [7] have been proposed to deliver 100% throughput under uniform traffic. iSLIP showed that the desynchronization effect, where arbiters reach the point where each of them prefers to match with different input/outputs, is beneficial for switching under uniform traffic. Other schemes have employed further the advantage of this effect [8].

However, schemes based on round-robin selection have not been shown to provide nearly 100% throughput under nonuniform traffic patterns without speedup or load-balancing stages [9]. The proposed exhaustive dual round-robin matching (EDRRM) scheme [10] has shown a throughput higher than iSLIP and DRRM under nonuniform traffic patterns at the cost of reduced performance under uniform traffic. Furthermore, scheduling on a set-of-cell basis have been shown to have a positive effect for switching under different traffic scenarios [11], [12].

This paper introduces the captured-frame concept and shows its application in maximal-size matching schemes. The resulting schemes are the unlimited frame-size occupancy-based round-robin matching (uFORM), and the unlimited frame-size occupancy-based PIM (uFPIM). This paper shows that the captured-frame concept, used for cell matching eligibility, improves the performance of the arbitration schemes run in a cell-basis. These arbitration schemes can achieve high throughput under several nonuniform traffic patterns. This paper also shows that uFORM retains the high performance of round-robin schemes under uniform traffic.

This paper is organized as follows. Section II presents the switch model and preliminary definitions. Section III introduces the uFPIM scheme and discusses the expected performance. Section IV introduces the uFORM scheme. Section V presents a simulation study of the throughput and delay performance of uFORM and uFPIM under uniform and nonuniform traffic patterns. Section VI discusses the properties of the presented matching schemes. Section VII presents the conclusions.

II. Switch Model and Preliminary Definitions

In this paper, we consider a single-stage IQ switch with N input and output ports. There are N VOQs at each input port.

The authors are with the Department of Electrical Engineering, New Jersey Institute of Technology, Newark, NJ 07102.

This work was supported in part by National Science Foundation under grant contracts 0435250 and 0423305, and by NJIT under grant 421070.

Correspondence author. Email: rrojas@njit.edu.

0-7803-8939-5/05/$20.00 (C) 2005 IEEE

A VOQ at input port i, where $0 \leq i \leq N-1$, that stores cells for output port j, where $0 \leq j \leq N-1$, is denoted as $VOQ_{i,j}$. The following definitions are used in the description of the proposed matching scheme.

A frame is related to a VOQ. A frame is the set of one or more cells in a VOQ that are eligible for matching, where only the HOL cell of the frame is considered per time slot. A VOQ is said to be on-service status if the VOQ has a frame size of two or more cells and the first cell of the frame has been matched (i.e., started service). An input is said to be on-service status if there is at least one on-service VOQ.

A VOQ is said to be off service if the last cell of the frame has been matched (i.e., ended service) or no cell of the frame has been matched (i.e., awaiting service). Note that for frame sizes of one cell, the associated VOQ is off-service during the matching of the single-cell frame. An input is said to be off-service if all VOQs are in off-service status.

At the time t_c of matching the last cell of the frame associated to $VOQ(i,j)$, the next frame is assigned a size equal to the cell occupancy of $VOQ(i,j)$ at this time. We define this as captured frame size.[1] Cells arriving to $VOQ(i,j)$ at time t_d, where $t_d > t_c$, are not considered for matching until the current frame is totally served and a new frame is captured. Figure 1 shows an example of the frame capture and the service status of a VOQ. At time slot t, the frame is off service, and the request for a match of the HoL cell is off service as well. Assuming that the VOQ is matched during time slot t, at time slot $t+1$, the size of the frame is three cells, and the VOQ, as well as the request, becomes on service. The status of the VOQ remains on service for the rest of the frame duration, or until time slot $t+3$. After the last cell of the frame is matched, a new frame is captured, with a size of two cells, which are the only cell in the queue. Therefore, the status of the VOQ changes to off service.

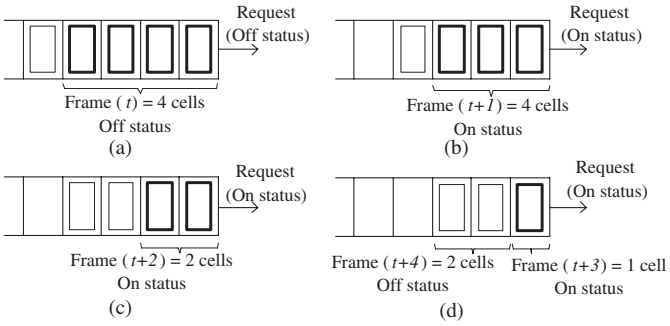

Fig. 1. Example of a frame and the service status of a VOQ.

For each VOQ there is a captured frame-size counter, $CF_{i,j}(t)$. The value of $CF_{i,j}(t)$ indicates the frame size; that is, the maximum number of cells that a $VOQ(i,j)$ can have as candidates in the current and future time slots. $CF_{i,j}(t)$ takes a new value when the last cell of the current frame of $VOQ(i,j)$ is matched. $CF_{i,j}(t)$ decreases its count each time a cell is matched, other than the last. Each VOQ has a status flag $F_{i,j}$ to indicate the on/off service status. If VOQ is in on-service status, $F_{i,j} = 1$. Otherwise, $F_{i,j} = 0$.

[1]We call this captured frame as it is the equivalent of having a snapshot of the occupancy of a VOQ at time t_c, where the occupancy determines the frame size.

III. Unlimited Frame Occupancy-based PIM Scheme

In this scheme, we use random selection as in the parallel iterative matching (PIM) scheme [3] to describe the matching process using the captured-frame size and to show the effect of this concept on the throughput performance.

uFPIM follows three steps as in the PIM scheme:

Step 1: Request. Non-empty on-service VOQs send a request to their destined outputs. Non-empty off-service VOQs send a request to their destined outputs if input i is off-service.

Step 2: Grant. If an output arbiter a_j receives any requests, it chooses a request from the on-service VOQ (also called an on-service request) in a random fashion. If none on-service request exists, the output arbiter chooses an off-service request in a random fashion.

Step 3: Accept. If the input arbiter a_i receives any grants, it accepts one on-service grant in a random fashion. If none on-service grant exists, the arbiter chooses an off-service grant in a random fashion. The CF counter updates the value according to the following: If the input arbiter a_i accepts a grant from a_j, and if:

i) $CF_{i,j}(t) > 1$: $CF_{i,j}(t+1) = CF_{i,j}(t) - 1$ and this VOQ is set as on-service.
ii) If $CF_{i,j}(t) = 1$: $CF_{i,j}(t+1)$ is assigned the occupancy of $VOQ(i,j)$, and $VOQ(i,j)$ is set as off-service.

Figure 2 shows an example of the uFPIM scheme. The CF values are shown as input contents. In this example, we only show the captured-frame sizes and the service status at each VOQ. In the request phase, inputs 0, 1, and 2 send request to all outputs they have a frame (or cells) for. Input 3 sends a single on-service request to output 0, as the off-service VOQ is inhibited as described in the scheme. The output and input arbiters select a request by service status and in a random fashion among all request of the same service status, as shown by the grant and accept phases. Output 0 selects the on-service request from input 3 over the off-service request from input 1. After the match is completed, the CF values are updated as shown in the figure. Note that at time slot $t+1$, three VOQs become on service.

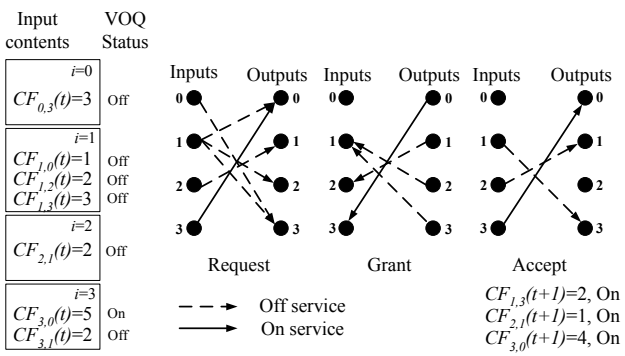

Fig. 2. Example of uFPIM in a 4×4 switch.

We call unlimited to this scheme because the captured-frame size is not limited to a maximum value at the capture time.

It has been shown that PIM delivers a limited throughput under uniform traffic [14], [13] for a large N and a single iteration. The throughput of PIM, (T_{PIM}), can be described as in [14]:

$$T_{PIM} = 1 - (1 - \frac{p}{N})^N, \quad (1)$$

where p is the probability of a cell arrival in a time slot. As presented in [14], the probability of a request being granted by an output j is p/N. The probability that the output j does not receive a cell from all inputs is $(1 - p/N)$. When N is large and $p = 1.0$, T for PIM is known to be 63.2% under uniform traffic.

The uFPIM scheme uses the captured-frame concept, where cells become eligible if they arrive before service for the current frame ends. Therefore, cell arrivals do not affect arbitrarily the matching process. Furthermore, once a match is achieved, the match is kept during the frame duration and the input is on-service, therefore, reducing the number of contending ports that participate in random selection. In subsequent time slots, the number of matches is increased because the use of frames makes a match remains for the time that a frame is served. Therefore, the probability of a request of being granted by an output j is $p/(N - E(m))$ and the throughput of uFPIM, T_{uFPIM}, is:

$$T_{uFPIM} = \frac{E(m)}{N} + \frac{N - E(m)}{N}[1-(1-\frac{p}{N - E(m)})^{N-E(m)}], \quad (2)$$

where $E(m)$ is the average number of on-service inputs. From Eq. 2, T increases beyond 63.2% under uniform traffic when $E(m) \geq 1$. Furthermore, as the input load increases, $E(m)$ increases. The effect that $E(m)$ has over the matching performance is the similar to PIM with several iterations because the number of matches keeps adding up in subsequent time slots and during the frame size of those VOQs that are in on-service status.

IV. Unlimited Frame Occupancy-Based with Round-Robin Matching

To study the effect of the captured frame size on round-robin based matching schemes, we introduce the unlimited frame occupancy-based with round-robin matching (uFORM). This scheme follows three steps: request, grant, and accept. The inputs have an input arbiter a_i and the output have an output arbiter a_j. The matching process is as follows:

Step 1: Request. Non-empty on-service VOQs send a request to their destined outputs. Non-empty off-service VOQs send a request to their destined outputs if input i is off-service.

Step 2: Grant. If an output arbiter a_j receives any requests, it chooses a request from the on-service VOQ (also called an on-service request) that appears next in a round-robin schedule, starting from the pointer position. If none on-service request exists, the output arbiter chooses an off-service request that appears next in a round-robin schedule, starting from its pointer position.

Step 3: Accept. If an input arbiter a_i receives any grants, it accepts one on-service grant in a round-robin schedule, starting from the pointer position. If none on-service grant exists, the arbiter chooses an off-service grant that appears next in round-robin schedule starting from its pointer position. The input and output pointers are updated to one position beyond the matched one. In addition to the pointer update, the CF counter updates the value according to the following: If the input arbiter a_i accepts a grant from a_j, and if:

i) $CF_{i,j}(t) > 1$: $CF_{i,j}(t+1) = CF_{i,j}(t) - 1$ and this VOQ is set as on-service, $F_{i,j} = 1$.

ii) If $CF_{i,j}(t) = 1$: $CF_{i,j}(t+1)$ is assigned the occupancy of $VOQ(i,j)$, and $VOQ(i,j)$ is set as off-service, $F_{i,j} = 0$.

Figure 3 shows an example of uFORM in a 4×4 switch. In this example, the contents of the VOQs are the same as that of the uFPIM example. The pointers of the input and output arbiters are positioned as shown in the request phase. The off inputs send request to all outputs they have a frame for. In the grant phase, the output arbiters select the request according to the request status and the pointer position. Output 0 selects the on-service request over the off-service request. Output 3 receives two off-service request, and selects input 1 because that input has higher priority, according to the pointer position. Outputs 1 and 2 receive a single off-service request, therefore, the requests are granted. In the accept phase, input 1 selects output 3 by using the pointer position. Input 2 accepts the single grant issued by output 1. Input 3 accepts the single grant, issued by output 0. Since the results are the same as in the uFPIM example, the CF values and service status are updated as in that example. Note that the input and output arbiters for the on-service ports (input 3 and output 0) are updated, but since the service status takes higher precedence, the pointer position in this case becomes secondary in the selection process.

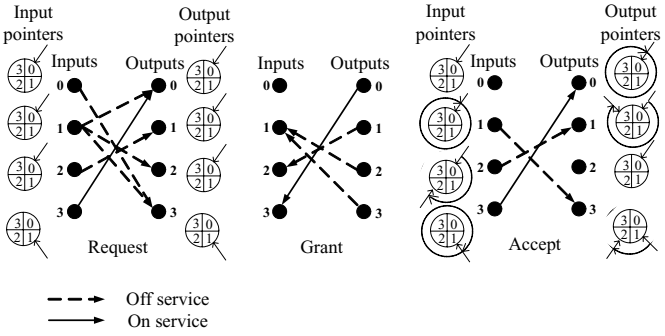

Fig. 3. Example of the uFORM scheme in a 4×4 switch.

V. Performance Evaluation

We consider $i$slip with one iteration (or 1SLIP) and PIM on this study for comparison purposes. The performance evaluations are produced by computer simulation. The traffic models considered have destination with uniform and nonuniform distributions. The simulation does not consider the segmentation and re-assembly delays for variable size packets. Simulation results are obtained with a 95% confidence interval, not greater than 5% for the average cell delay.

A. Uniform Traffic

Figure 4 shows the simulation results of four 32×32 IQ switches, each one with a different matching scheme:

1SLIP, PIM, uFORM, and uFPIM, all under uniform traffic with Bernoulli arrivals. This figure shows that uFORM, as 1SLIP, delivers 100% throughput under uniform traffic. PIM is known to offer about 63.2% throughput [13]. However, while using the captured frame-size concept in uFPIM, the performances improves to nearly 100% throughput. The reason for the improvement shown by uFPIM are that once a match is achieved, the match is kept during the frame duration. Therefore, contention among the others ports is reduced with each time slot.

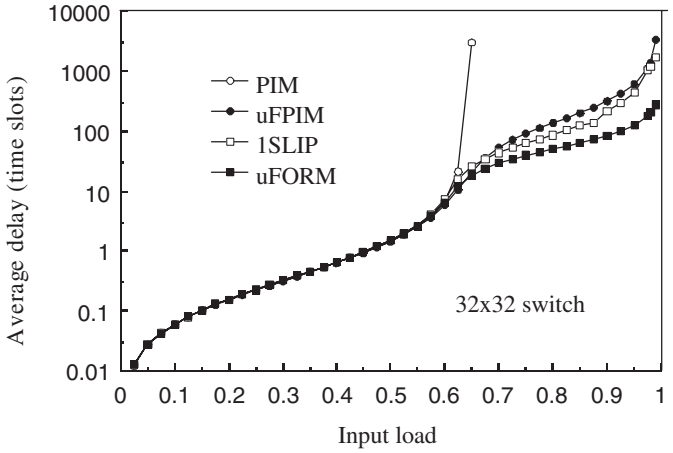

Fig. 4. Average delay of uFORM and uFPIM schemes under Bernoulli uniform traffic.

B. Nonuniform Traffic

We simulated these four schemes under several nonuniform traffic models: unbalanced [15], Chang's [9], asymmetric [16] and power-of-two (PO2) [11]. The unbalanced traffic model uses a 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 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,

$$\rho_{s,d} = \begin{cases} \rho\left(w + \frac{1-w}{N}\right) & \text{if } s = d \\ \rho\frac{1-w}{N} & \text{otherwise.} \end{cases} \quad (3)$$

When $w = 0$, the offered traffic is uniform. On the other hand, when $w = 1$, it 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 $s = d$. Figure 5 shows the throughput performance of 1SLIP, PIM, uFPIM, and uFORM under unbalanced traffic. This figure shows that uFORM provides over 99% throughput under the complete range of w and that uFPIM reaches just 99% throughput. The high throughput of uFORM under this traffic model is the product of considering VOQ occupancy. uFORM ensures service to queues with high load by capturing a large frame size for each, and to the queues with low load by using round-robin selection.

Chang's traffic model can be defined as $\rho = 0$ for $i = j$ and $\rho_{i,j} = \frac{1}{N-1}$. Figure 6 shows the average cell delay achieved

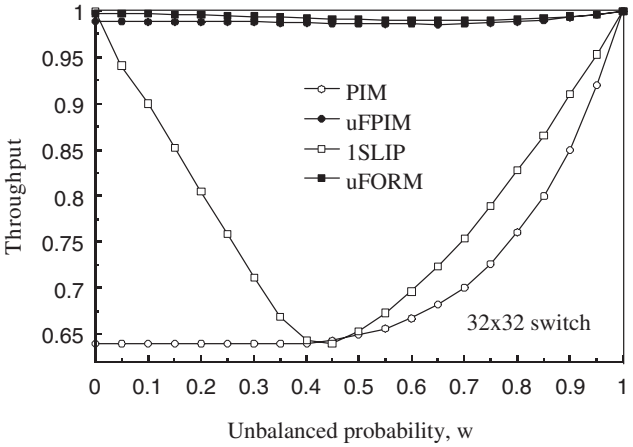

Fig. 5. Throughput performance of uFORM and uFPIM under unbalanced traffic.

by the four matching schemes under Chang's traffic model. Under Chang's traffic, the throughput is 64% by PIM, 97% by 1SLIP, and 99% by uFORM and uFPIM.

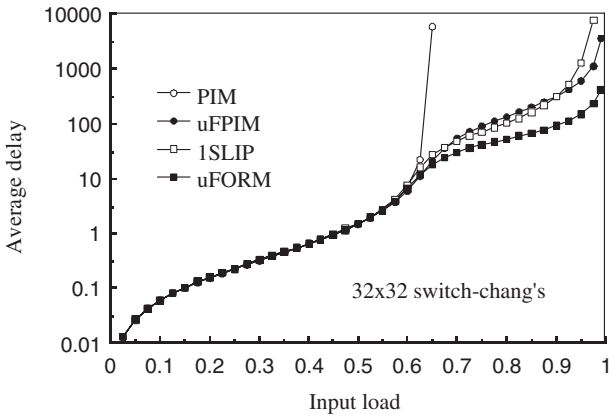

Fig. 6. Throughput performance of uFORM and uFPIM under Chang's traffic.

Figure 7 shows the four matching schemes under the asymmetric traffic model. Under asymmetric traffic, the throughput obtained is 70% by PIM, 72% by 1SLIP, and above 99% by uFORM and uFPIM. Figure 8 shows the performance of the four matching schemes under the PO2 traffic model. We consider 30×30 switches for simulation under PO2 traffic. Under this traffic model, the obtained throughput is 72% by PIM, 75% by 1SLIP, and 95% under uFPIM and uFORM. Although uFPIM and uFORM have under 99% throughput under PO2 than in the other traffic models, these schemes show performance improvement. In general, Figures 4-8 show that the throughput is improved by using the captured-frame concept to define the set of eligible cells for the matching process.

VI. Properties

The use of a captured frame size and the service concepts used here make uFORM and uFPIM deliver high performance under uniform and unbalanced traffic patterns. Note that in the

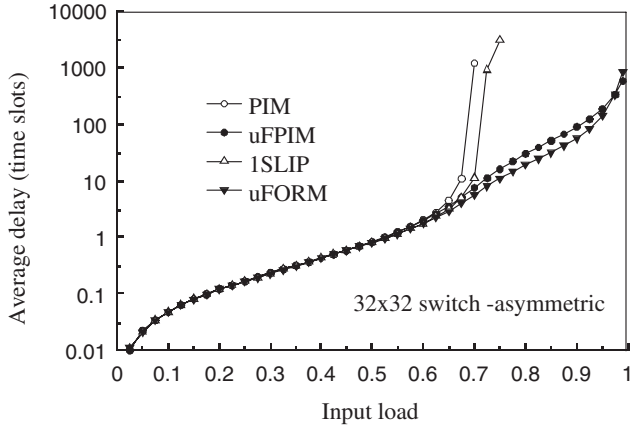

Fig. 7. Throughput performance of uFORM and uFPIM under Asymmetric traffic.

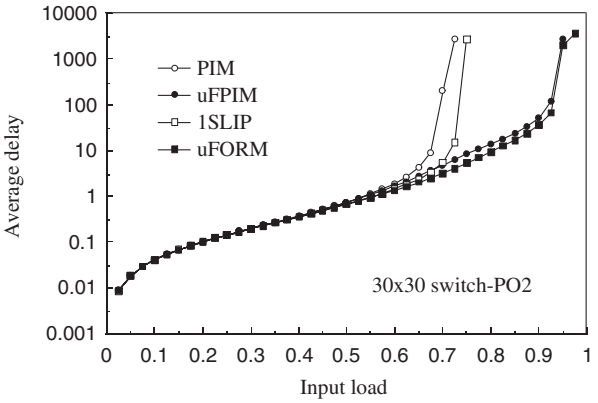

Fig. 8. Throughput performance of uFORM and uFPIM under PO2 traffic.

case where a VOQ has no cells at the capturing time, the VOQ can still participate in a matching when after a cell arrives, as long as the input is off-service.

When a VOQ changes its status to on-service, the VOQ has higher priority than the others to continue sending its request in subsequent time slots. When an input is off-service, all nonempty VOQs (independently of their CF value) send a request to their respective outputs.

Under uniform traffic, the captured frame sizes are not expected to reach large values because of the cell distribution among all queues. Therefore, most queues may remain in off-service status while completing service for one-cell frames. The performance is then determined by the selection policy. Furthermore, as the captured frame may include old cells, the delay may be smaller than pure round-robin or random based matching. Under unbalanced traffic, some queues are expected to have heavier loads than others. The queues with large occupancies have a higher service than the queues with lower occupancy. The difference on frame sizes results in more service for queues with a larger number of arrivals than those for queues with a small number of arrivals. Moreover, the selection policy ensures that all queues receive service.

The implementation complexity of uFORM and uFPIM are low because of the following reasons: the arbitration schemes are round-robin and random based, they perform no comparisons among different queues, and the hardware additions are the CF counters to each VOQ and a service flag.

VII. Conclusions

In this paper, we introduced the captured frame size concept to determines eligibility of cell in the matching process for input queued packet switches. We also proposed the matching schemes, uFORM and uFPIM, that use the presented concept, a single iteration, and no speedup. The presented schemes shows above 99% throughput under the unbalanced traffic model. uFORM and uFPIM were studied under several traffic models and they showed higher switching performance than those schemes without the captured-frame concept. The complexity of these matching schemes is low as they do not need to compare the status of the VOQs; the hardware and timing complexity of uFORM are low because only the update of the CF counters and F flags to the implementation, and the time to update CF and F can be performed in parallel with the pointer update in uFORM.

References

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

[2] N. McKeown, A. Mekkittikul, V. Anantharam, J. Walrand, "Achieving 100% Throughput in an Input-queued Switch," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1260-1267, August 1999.

[3] T.E. Anderson, S.S. Owicki, J.B. Saxe, and C.P. Tacker, "High-speed Switch Scheduling for Local Area Networks," ACM Trans. on Computer Systems, vol. 11, no. 4, pp. 319-352, November 1993.

[4] N. McKeown, "The iSLIP scheduling algorithm for Input-queued Switches," IEEE/ACM Trans. Networking, vol. 7, no. 4, pp. 188-201, April 1999.

[5] H.J. Chao, J-S. Park, "Centralized Contention Resolution Schemes for a large-capacity Optical ATM Switch," IEEE ATM Workshop 1998, pp. 11-16, May 1998.

[6] E. Oki, R. Rojas-Cessa, and H. J. Chao, "PMM: A Pipelined Maximal-Sized Matching Scheduling Approach for Input-Buffered Switches," IEEE Globecom 2001, pp. 35-39, Nov. 2001.

[7] Y. Jiang and M. Hamdi, "A fully Desynchronized Round-robin Matching Scheduler for a VOQ Packet Switch Architecture," IEEE HPSR 2001, pp. 407-411, May 2001.

[8] Y. Jiang and M. Hamdi, "A 2-stage Matching Scheduler for a VOQ Packet Switch Architecture," IEEE ICC 2002, vol. 4, pp. 2105-2110, May 2002.

[9] C-S. Chang, D-S. Lee, and Y-S. Jou, "Load Balanced Birkhoff-von Neumann Switches," IEEE HPSR 2001, pp.276-280, April 2001.

[10] Y. Li, S. Panwar, H.J. Chao, "The Dual Round-robin Matching Switch with Exhaustive Service," IEEE HPSR 2002, pp. 58-63, 2002.

[11] A. Bianco, M. Franceschinis, S. Ghisolfi, A.M. Hill, E. Leonardi, F. Neri, R. Webb, "Frame-based Matching Algorithms for Input-queued Switches," IEEE HPSR 2002, pp. 69-76, 2002.

[12] S. Li and N. Ansari, "Input-queuing Switching with QOS Guarantees," IEEE INFOCOM 1999, vol.3, pp. 1152-1159, March 1999.

[13] G. Nong, J. K. Muppala, and M. Hamdi, "Analysis of Nonblocking ATM Switches with Multiple Input Queues," Networking, IEEE/ACM Transactions on, vol. 7 , issue: 1 , pp.60 - 74, February 1999.

[14] S. Motoyama, D. W. Petr, and V. S. Frost, "Input-queued switch based on a scheduling algorithm," Electronics Letters, vol:31, Issue:14, Pages:1127 - 1128, July 1995 .

[15] R. Rojas-Cessa, E. Oki, Z. Jing, and H. J. Chao, "CIXB-1: Combined Input-One-cell-crosspoint Buffered Switch," IEEE HPSR 2001, pp. 324-329, May 2001.

[16] R. Schoene, G. Post, and G. Sander, "Weighted Arbitration Algorithms with Priorities for Input-Queued Switches with 100% Throughput," Broadband Switches Symposium'99,1999. http://www.schoenen-service.de/assets/papers/Schoenen99bssw.pdf