Captured-Frame Selection Schemes for Scalable Input-Queued Packet Switches

Roberto Rojas-Cessa and Chuan-Bi Lin

***Abstract*— Input-queued (IQ) switches are attractive for the implementation of high-performance routers because they require no speedup in the used memory. It has been shown that IQ switches can provide 100% throughput under admissible traffic when using maximum weight matching schemes or speedup of two, and pre-computed switch configurations, for pre-known traffic patterns. These three different approaches require either high computation complexity or high memory costs. Therefore, there is a need of low-complexity and fast matching schemes that provide high throughput under several admissible traffic patterns, including those with nonuniform distributions, without recurring to speedup or multiple iterations. In this paper, we propose matching schemes that use the captured frame-size concept, and show that the application of this concept in matching schemes for IQ switches provides high throughput under a variety of admissible traffic patterns with a single iteration and no speedup. We present two weightless matching schemes, one based on round-robin selection, called uFORM, and the other based on random selection, called uFPIM. Furthermore, we study the application of the captured frame concept in matching schemes for multiple stage switches and show the achieved improvement on switching performance.**

***Index Terms*— captured frame, frame eligibility, service frame, nonuniform traffic, input queued**

I. Introduction

Input-queued (IQ) switches are attractive because their memories work without the speedup requirement of an output-queued (OQ) switch. As a result, IQ switch architectures have been adopted by several manufacturers of switches/routers. The introduction of virtual output queues (VOQs), where one queue per output port is placed in an input port of an IQ packet switch, is used to remove the head-of-line (HOL) blocking problem [1]. HOL blocking causes idle outputs to remain so, even in the existence of traffic for them at an idle input, thus impeding the delivery of high throughput.

This paper follows the common practices in packet switch design: 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, 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 IQ switch is the delivery of high throughput under different traffic conditions. In this paper, we consider admissible traffic [2] with Bernoulli and bursty arrivals that have destinations with uniform and nonuniform distributions.

The matching scheme used in 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] while using no speedup. However, MWM schemes have intrinsically high computation complexity that is translated into long resolution time and high hardware complexity. This makes these schemes prohibitively expensive for a practical implementation of high-speed switches with currently available technologies. An alternative is to use maximal-weight matching schemes. These schemes can provide higher throughput performance under uniform and nonuniform traffic patterns with, however, a large number of iterations. The hardware and time complexity of these schemes can be considered high for the ever increasing data rates because of the large number of iterations, and because of the number of parameters that need to be compared in the selection process. Furthermore, some weight-based schemes may starve queues with little traffic to provide more service to the congested ones, therefore, presenting unfairness [3].

Maximal-size matching can be used to resolve contention in IQ switches in a fast manner. An example of these size matching schemes is PIM [4], which is based on random selection. However, PIM cannot achieve 100% throughput under admissible uniform traffic because contentions cannot be avoided in this scheme. Schemes based on round-robin selection can provide higher throughput than PIM [5]. Some example of round-robin schemes are iSLIP [5], iDRRM [6], [7], and SRR [8] and these can deliver 100% throughput under uniform traffic with a single iteration. 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 this traffic pattern. However, schemes based on round-robin selections have not been shown to provide nearly 100% throughput under nonuniform traffic patterns without speedup, or pre-calculated configurations [9] that are tailored for traffic with pre-known distributions. The exhaustive dual round-robin matching (EDRRM) scheme [10] has shown a throughput higher than iSLIP under nonuniform traffic patterns at the cost of reduced performance under uniform traffic.

An alternative to is to perform matching for a batch of cells, instead of for a single cell. Matching in train-of-cells basis have been shown to improve throughput under optimal train sizes for each different traffic scenarios [11]. This approach seems to be beneficial for nonuniform distributions as outputs

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

Correspondence author. Email: rrojas@njit.edu

receiving large amount of traffic may utilize efficiently an achieved match. However, it is difficult to define one train size for all traffic distributions. Some train sizes may be optimal for some distributions and at the same time non-optimal for others.

This paper introduces the captured-frame concept and shows its application on maximal-size matching schemes for IQ switches. The resulting schemes are maximal size based and can provide service to VOQs with a rate proportional to their input loads in similar way weight-based schemes do. This is achieved by using frame service, where the frame length depends on the accumulation of cells in a given period of (service) time, and therefore, the frame length is adjusted dynamically. We called the resulting schemes as the unlimited frame-size occupancy-based round-robin matching (uFORM), and the unlimited frame-size occupancy-based PIM (uFPIM). This paper demonstrates that the captured-frame concept, used for cell matching eligibility, improves the performance of the arbitration schemes, which are run in a cell-basis. We analyze the achievable throughput of uFPIM in single-stage IQ switches under uniform traffic. We also show the switching performance of uFORM and uFPIM under uniform and nonuniform admissible traffic. We show that uFORM retains the high performance of round-robin schemes under uniform traffic and high performance under nonuniform traffic.

Looking towards switch scalability and knowing that single-stage switches are difficult to scale up (due to VLSI constraints, such as chip's pin count), we consider multiple-stage Clos-networks [12]. Clos-network switches have either three or five stages, where each stage in comprised by two or more small switches, called modules. Here, we focus on three-stage switches to use the minimum amount of hardware. Clos-network switches are more complex to configure than single-stage IQ switches and are more sensitive to the efficiency of matching algorithms than single-stage IQ switches. The configuration of three-stage Clos-network switches includes the matching of input and output ports, located in the first and second stage modules, and the selection of the interconnecting path in the second stage module.

The configuration complexity of these three-stage switches can be reduced by using queues in the first and third stage modules [13], in addition to the queues at the inputs. These switches are known as memory-space-memory (MSM) Clos-network switches [14], [15]. The configuring process (port matching plus central module routing) is then converted into a dispatching problem (matching between the queues and output links of the first-stage modules, plus another matching between input-module requests and output links of the central-stage modules), where only the first and second stage switches needs to be configured. Scheduling schemes that have been originally developed for single-stage IQ switches and have been adapted for MSM Clos-network switches have shown that the achieved throughput performance of MSM Clos-networks is equal to or lower than that of single-stage IQ switches [13], [16]. Furthermore, as the modules in these MSN Clos-network switches are located in different chips or boards, transmission delays between modules, may be long. Therefore, for schemes where their implementation uses exchange of information between different modules in more than one iteration (for scheduling), the resolution time becomes long enough to be considered impractical. One way to make the implementation of this switch practical is by designing scheduling schemes that can provide high throughput with a single iteration.

As a solution to this need, we extend the study of the captured frame concept to dispatching schemes for MSM Clos-network switches. We show that our developed dispatching schemes can achieve high throughput under uniform and several nonuniform traffic patterns.

This paper is organized as follows. Section II describes the single-stage switch and introduces preliminary definitions used in this paper. Section III introduces the uFPIM and uFORM scheduling schemes. Section IV analyzes the throughput of uFPIM in single-stage IQ switches. Section V presents a simulation study of the throughput and delay performance of uFORM and uFPIM under uniform and nonuniform traffic patterns. Section VI introduces the framed concurrent round-robin dispatching (FCRRD) scheme for a MSM Clos-network switch. Section VII shows the performance of FCRRD under uniform and nonuniform traffic models. Section VIII presents our conclusions.

II. Single-Stage Input-Queued Switch Model and Definitions

We consider a single-stage $N \times N$ switch, with VOQs in the inputs. A VOQ that stores cells from input i to output j is denoted as $VOQ(i, j)$. Unless otherwise stated, we consider that a VOQ can store a large number of cells. In this switch, each input can dispatch one cell each time slot and each output can receive up to one cell per time slot (i.e., no speedup is used). Figure 1 shows the switch model.

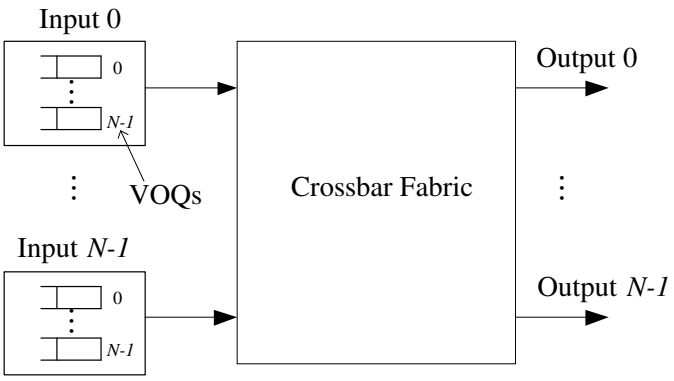

Fig. 1. Input-queued switch with VOQs.

We use the following definitions in the description of the proposed matching schemes.

Frame. A frame is related to a VOQ. A frame is the set of one or more cells in a VOQ that are eligible for dispatching. Only the HOL cell of the VOQ is eligible per time slot.

On-service status. A VOQ is said to be in 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. An input is said to be on-service status if the status of a VOQ becomes on.

Off-service status. A VOQ is said to be in off-service status if the last cell of the VOQ's frame has been matched or no cell of the frame has been matched. Note that for frame sizes of one cell, the associated VOQ is off-service during the matching of its one-cell frame.

Captured frame size. 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 at $VOQ(i,j)$. Cells arriving in $VOQ(i,j)$ 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. We call this captured frame as it is the equivalent of having a snapshot of the VOQ occupancy at time t_c, where the occupancy determines the frame size.

Figure 2 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 size of the frame is four cells and that the VOQ is first matched during time slot t, the VOQ becomes on service at time slot $t+1$. 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, as these cells are the only ones in the queue at this time. Then, the status of the VOQ changes to off service in the following time slot.

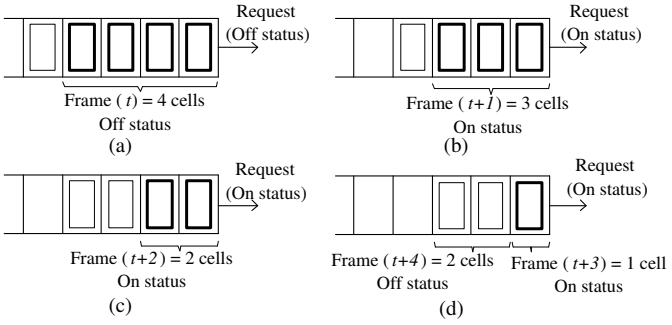

Fig. 2. Example of a frame and the service status of a VOQ.

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, $F_{i,j} = 0$.

III. Matching schemes with Captured Frame for Single-Stage IQ Switches

A. uFPIM Scheme for Single-Stage IQ Switches

The uFPIM scheme has CF counters and F flags. 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, $F_{i,j} = 1$.
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 a matching in 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 off-service requests to all outputs they have at least a cell 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 requests 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.

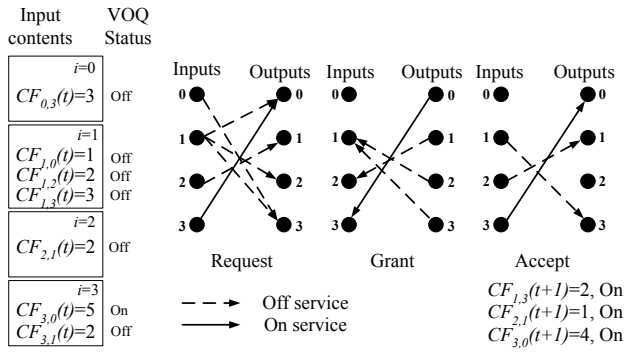

Fig. 3. Example of uFPIM in a 4×4 switch.

B. uFORM Scheduling Scheme for Single-stage IQ Switches

uFORM follows request-grant-accept steps as in uFPIM, and uses round-robin selection instead of random-based selection. 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 the input arbiter a_i receives any grants, it accepts an 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, $F_{i,j} = 0$.

We give the prefix unlimited to the names of these two matching schemes because the captured-frame size is not limited to a maximum value at the capture time.

Figure 4 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 cell 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.

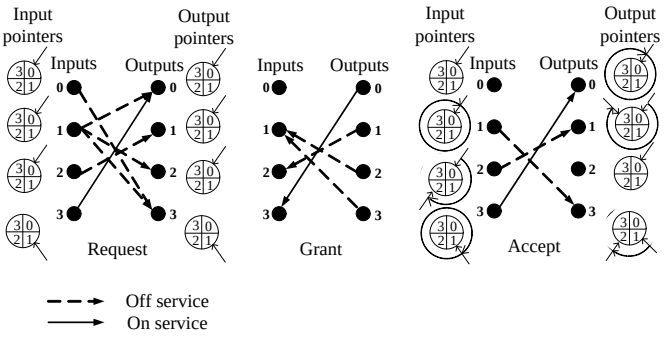

Fig. 4. Example of the uFORM scheme in a 4×4 switch.

IV. Throughput Analysis of Randomized Selection using the Captured-Frame Concept

In this section, we analyze the throughput of uFPIM and show the improvement of over the throughput of PIM. It has been shown that the throughput of an IQ switch using PIM under uniform traffic [17], [18] for a large N and a single iteration, where PIM's throughput (T_{PIM}) can be defined by:

$$T_{PIM} = 1 - (1 - \frac{\rho}{N})^N. \quad (1)$$

where, N is the number of input/output ports and ρ is the probability of a cell arrival in a time slot. As presented in [17], the probability of a request that is being granted by output j is ρ/N. The probability that output j does not receive a cell from any inputs is $(1-\rho/N)$. When N is large and $\rho = 1.0$, T_{PIM} is known to be 63.2% under uniform traffic with Bernoulli arrivals.

The uFPIM scheme uses the captured-frame concept. In this scheme, a frame is defined at the end of the VOQ service and those cells that arrived during the (frame) servicing time are considered part of the next frame. Therefore, cell arrivals (after a frame is defined) 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, thus, 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 last during the time that a frame is served. Therefore, the probability of a request of being granted by an output i is

$$\rho/(N - E(m))$$

and the throughput of uFPIM, T_{uFPIM}, is defined by

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

where $E(m)$ is the average number of on-service inputs.

Then $E(m)$ is defined by the number of cells in a frame. Because of the two states of an input (on-service or off-service), we consider the average of the duration of a frame, P_m, follows a binomial distribution:

$$P_m = (\begin{matrix} N \\ m \end{matrix})p^m(1-p)^{N-m}, m = 0, 1, 2, \ldots, N, \quad (3)$$

where p is the probability that an input becomes on service.

$$E(m) = \sum_{m=0}^{N} m \cdot P_m \quad (4)$$

then

$$\begin{aligned} E(m) &= 0 \cdot P_0 + 1 \cdot P_1 + 2 \cdot P_2 + \ldots + N \cdot P_N \\ &= 0 + 1 \cdot (\begin{matrix} N \\ 1 \end{matrix})p^1(1-p)^{N-1} + \ldots + N \cdot (\begin{matrix} N \\ N \end{matrix})p^N \\ &\geq N \cdot p \end{aligned} \quad (5)$$

When the switch size N is large,

$$E(m) = N \cdot p. \quad (6)$$

Recalling that a VOQ that has a frame size of two or more cells and after the first cell of the frame is matched, the status of the VOQ becomes on-service, that is, while a VOQ is to become on-service, the VOQ can not matched at the least two time slots (as the cells that will form the frame arrive in the

first two time slots, and the VOQ gets matched in the third time slot) the captured a new frame-size. Let's have p_{1st-um} and p_{2nd-um} denote the probability that there an output does not get matched in the first and second time slots, respectively, and p_{3rd-m} is the probability that the output gets matched in the third time time slot, p becomes:

$$\begin{aligned} p &= 1 - p_{1st-um} \cdot p_{2nd-um} \cdot p_{3rd-m} \quad (7) \\ &= 1 - (1 - \frac{1}{N - A_1})^{N-A_1} \cdot (1 - \frac{1}{N - A_2})^{N-A_2} \\ &\cdot \quad (1 - \frac{1}{N - A_3})^{N-A_3} \\ &\geq 1 - ((1 - \frac{1}{N})^N)^3, \quad (8) \end{aligned}$$

where A_1, A_2, and A_3 represent the number of on-service inputs at the first, second, and third time slots, respectively. Here, because $E(m)$ is difficult to calculate, we use the approximation

$$1 - ((1 - \frac{1}{N})^N)^3$$

to calculate p. From Eq. (8), we get p, and from Eq. (5), we get $E(m)$. Using $E(m)$ in Eq. (2) we estimate the maximum throughput of uFPIM when $\rho = 1.0$. For example, if $N = 32$, $p \geq 0.952$ and $E(m) \simeq N \cdot p \simeq 30$, and then $T_{uFPIM} = 0.986$.

When N is large,

$$p = 1 - ((1 - \frac{1}{N})^N)^3 \simeq 1 - (\frac{1}{e})^3 = 0.95 \quad (9)$$

then

$$E(m) = 0.95N.$$

Now, using this value in Eq.(2) and

$$\begin{aligned} T_{uFPIM} &= \frac{0.95N}{N} + \\ &\quad \frac{N - 0.95N}{N} \cdot \\ &\quad (1 - (1 - \frac{1}{N - 0.95N})^{N-0.95N}) \\ &= 0.95 + 0.05 \cdot (1 - \frac{1}{e}) \\ &= 0.982 \end{aligned} \quad (10)$$

In this way, for a large N, the actual T_{uFPIM} is higher than 0.982.

Figure 5 shows the throughput of uFPIM produced by analysis and simulation when considering Bernoulli uniform traffic. The analysis result is close to the simulation result under different switch sizes. The actual throughput is expected higher than the one obtained through analysis because we use Eqs. (6) and (8).

V. Performance Evaluation of uFPIM and uFORM

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, where results are obtained with a 95% confidence interval, not greater than 5% for the average cell delay. The traffic models considered have destination with uniform and nonuniform distributions. The simulation does not consider the segmentation and reassembly delays for variable size packets.

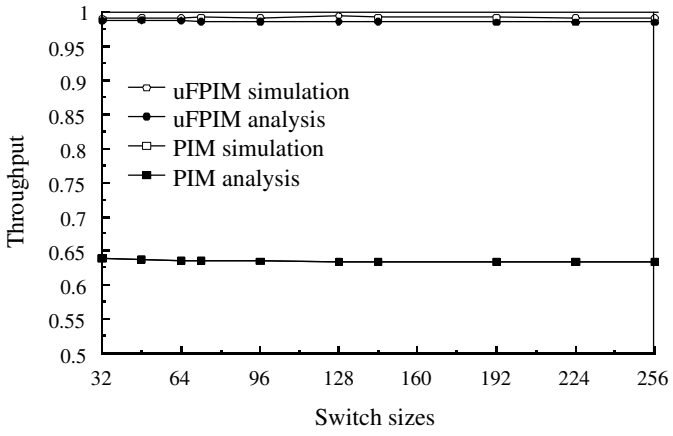

Fig. 5. Throughput comparison of analysis and simulation of uFPIM with different switch sizes.

A. Uniform Traffic

Figure 6 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 iSLIP, delivers 100% throughput under uniform traffic. Under this traffic, PIM delivers about 63% throughput, however, when using the captured frame-size concept in uFPIM, the throughput improves to nearly 100%. The reason for the improvement by uFPIM is 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.

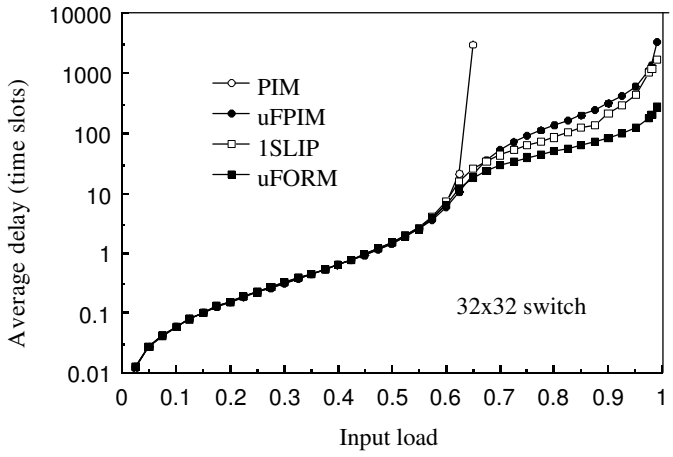

Fig. 6. Average delay of uFORM and uFPIM schemes under Bernoulli uniform traffic.

Figures 7 and 8 show the average latency produced by uFORM and uPIM schemes as a function of the offered load for switches with 8, 16, 32, 64, 128, and 256 ports. It can be seen that as the switch size increases, the average cell delay

increases. However, in a load close to 1.0, small switches, $N = \{8\}$ of uFORM and $N = \{8, 16\}$ of uFPIM, produce a long average delay.

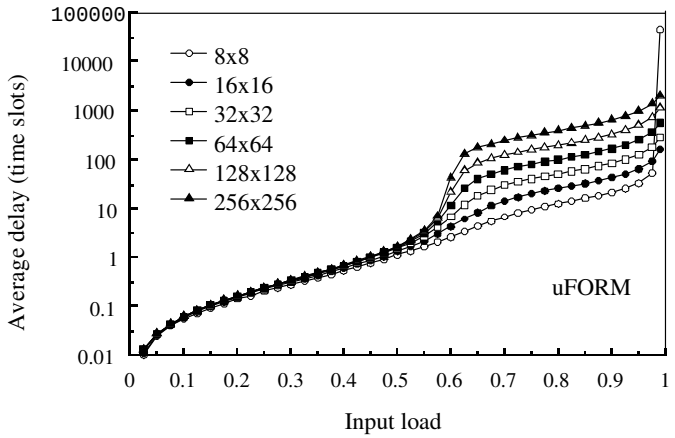

Fig. 7. Average delay of uFORM in function of switch size, under Bernoulli uniform traffic.

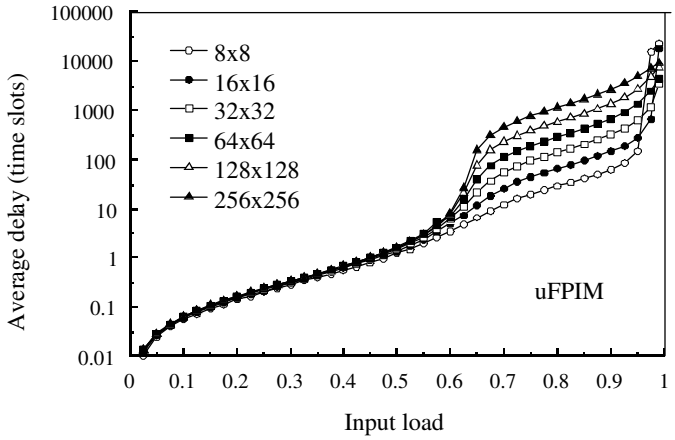

Fig. 8. Average delay of uFPIM in function of switch size, under Bernoulli uniform traffic.

B. Nonuniform Traffic

We simulated these four schemes under several nonuniform traffic models: unbalanced [19], Chang's [9], asymmetric [20] 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,

$$\rho_{s,d} = \begin{cases} \rho\left(w + \frac{1-w}{N}\right) & \text{if } s = d \\ \rho\frac{1-w}{N} & \text{otherwise.} \end{cases} \quad (11)$$

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 9 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 up to 99% throughput, while both PIM and 1SLIP reach 64% throughput. The high throughput of uFORM and uFPIM under this traffic model is the product of considering the 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.

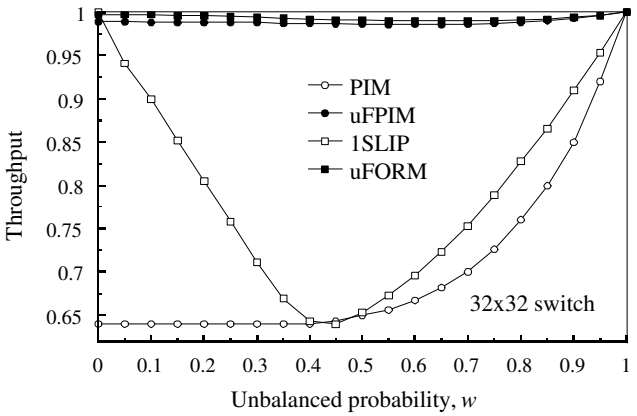

Fig. 9. Throughput performance of uFORM and uFPIM under unbalanced traffic.

Figure 10 shows the throughput performance of uFPIM and uFORM with different switch sizes under Bernoulli uniform traffic with 1.0 input load. This figure shows that uFORM provides over 99% throughput and that uFPIM reaches just 99% throughput for large switches. However, small switches, $N = \{8, 16\}$, have the lower throughput because they are more sensitive to the value of the captured frame sizes, and the frame size depends on the actual arrivals.

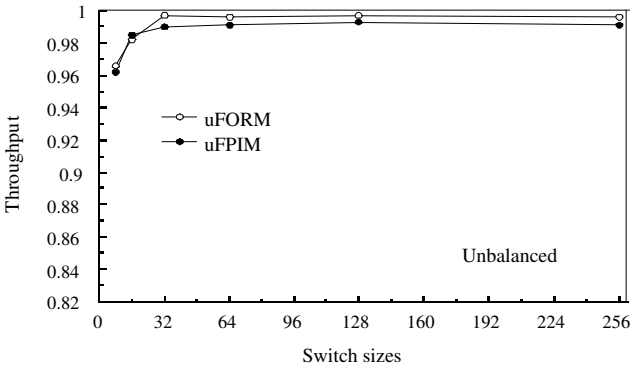

Fig. 10. Throughput performance of uFORM and uFPIM in function of switch size, under unbalanced traffic.

Chang's traffic model can be defined as $\rho = 0$ for $i = j$, and $\rho_{i,j} = \frac{1}{N-1}$ otherwise. Figure 11 shows the average cell delay achieved by the four matching schemes under this traffic model. The results show that the obtained throughput is 64% by PIM, 97% by 1SLIP, and 99% by both uFORM and uFPIM.

Figure 12 shows the average cell delay of the matching schemes under the asymmetric traffic model. The results show

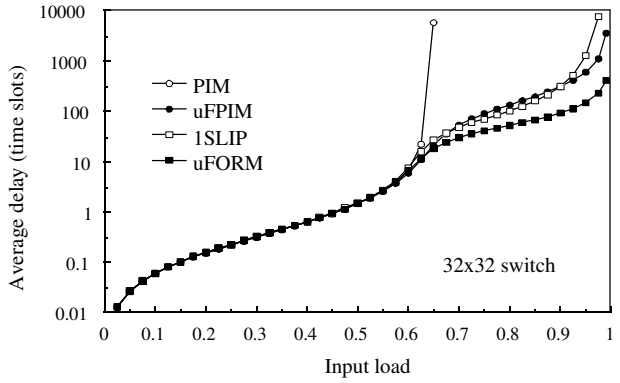

Fig. 11. Throughput performance of uFORM and uFPIM under Chang's traffic.

that the obtained throughput is 70% by PIM, 72% by 1SLIP, and above 99% by uFORM and uFPIM.

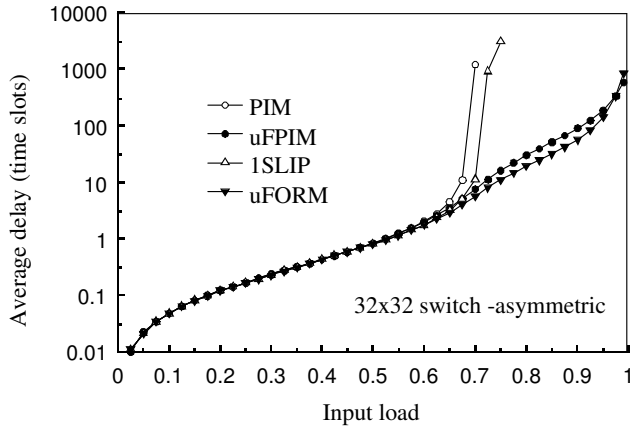

Fig. 12. Throughput performance of uFORM and uFPIM under asymmetric traffic.

The diagonal traffic model distributes all the load of an input between two different outputs, making the distribution heavily distributed among a small number of output. This traffic model is defined as $\rho_{i,j} = 0.5$ for $j = i$ and $j = (i+1) \bmod N$, and $\rho_{i,j} = 0$ otherwise. Figure 13 shows the average cell delay of our matching schemes under this traffic model. These results show that

Figure 14 shows the performance of the four matching schemes under the PO2 traffic model. Because of the complexity of describing the PO2 traffic model in our simulation program, we consider 30×30 switches for simulation. 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, nevertheless, performance improvement.

In general, Figures 6-14 show that the throughput is improved by using the captured-frame concept to define the set of eligible cells for the matching process.

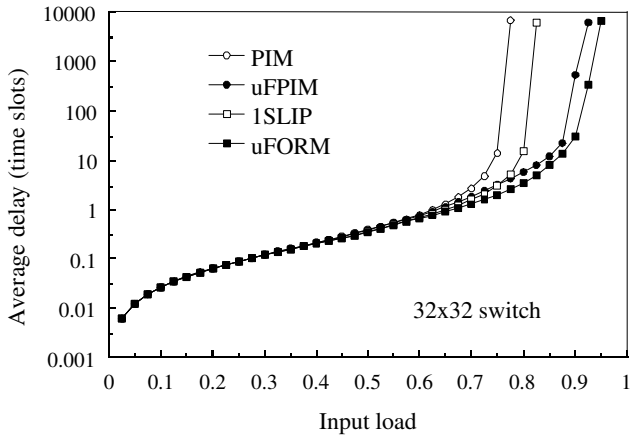

Fig. 13. Throughput performance of uFORM and uFPIM under diagonal traffic.

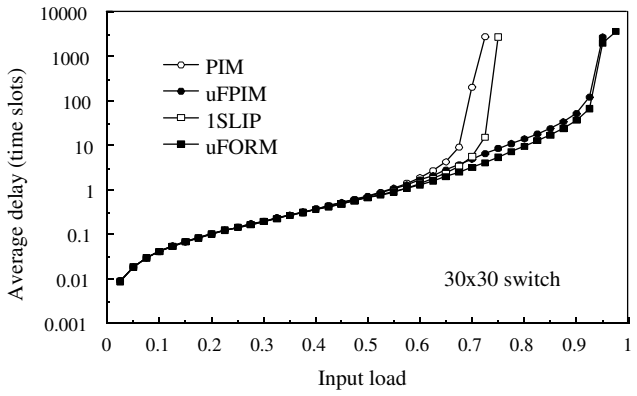

Fig. 14. Throughput performance of uFORM and uFPIM under PO2 traffic.

C. Discussion of Performance Results

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

When a VOQ changes its status to on-service, that 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 includes 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.

VI. Captured Frame in Dispatching Schemes for Clos-network Switches

In an MSN (3-stage) Clos-network switch, matching needs to be performed between VOQs in the first-stage modules and the links going from first-stage modules to second-stage modules, and a second match in second-stage modules between its input requests and output links. Furthermore, the first match uses the results of the second match process, therefore, if several iterations are to be performed, these two matches are performed a number of times as the number of iterations. As each iteration consumes the resolution time of each match, a single-iteration matching scheme that can achieve high (throughput) performance is desirable.

A. Clos-Network Switch Model and Preliminary Definitions

The Clos-network switch is a three-stage switch architecture [12], as Figure 15 shows. The first stage is also called as input stage, the second stage is called as central stage, and the third stage is called output stage. The modules in the input stage are called input modules (IMs), the modules in the middle stage are called central modules (CMs), and the modules in the output stage are called output modules (OMs). Here, the IMs and OMs have queues to store cells. We use the same terminology in [13], as follows:

$IM(i)$: $(i+1)$th input module, where $0 \leq i \leq k-1$.
$CM(r)$: $(r+1)$th central module, where $0 \leq r \leq m-1$.
$OM(j)$: $(j+1)$th output module, where $0 \leq j \leq k-1$.
n: number of input/output ports in each IM/OM, respectively.
k: number of IMs/OMs.
m: number of CMs.
$IP(i,h)$: $(h+1)$th input port (IP) at $IM(i)$, where $0 \leq h \leq n-1$.
$OP(j,l)$: $(l+1)$th output port (OP) at $OM(j)$, where $0 \leq l \leq n-1$.
$VOQ(i,j,l)$: Virtual output queue at $IM(i)$ that stores cells destined for $OP(j,l)$.
$L_I(i,r)$: output link of $IM(i)$ that is connected to $CM(r)$.
$L_C(r,j)$: output link at $CM(r)$ that is connected to $OM(j)$.

The switch has k IMs, m CMs, and k OMs. An $IM(i)$ has n input ports, each of which is denoted as $IP(i,h)$. Each $IM(i)$ has $n \times k$ VOQs. A $VOQ(i,j,l)$ stores cells going from $IM(i)$ to $OP(l)$ at $OM(j)$. In an IM, there are m output links. An output-link $L_I(i,r)$ is connected from $IM(i)$ to $CM(r)$. A $CM(r)$ has k output links, each of which is $L_C(r,j)$, which are connected to k OMs. An $OM(j)$ has n output ports, each of which is denoted as $OP(j,l)$, and has an output buffer.

In the following section, we described a dispatching scheme based on round-robin selection, FCRRD. However, this description can be adopted for a scheme based on random selection by changing the selection criteria.

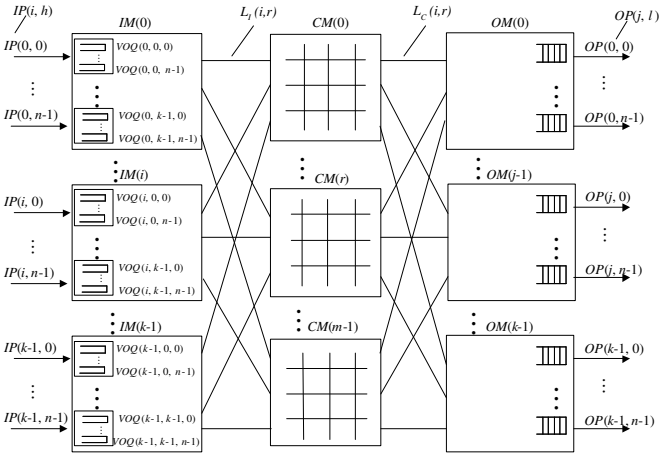

Fig. 15. Clos-network switch with VOQs in the IMs.

B. Frame Occupancy-Based Concurrent Round-Robin Dispatching Scheme

In $IM(i)$, there are m output-link round-robin arbiters and nk VOQ round-robin arbiters. An output-link arbiter, which is associated with $L_I(i,r)$, has its own pointer $P_L(i,r)$. A VOQ has an arbiter associated with it. For the sake of simplicity, VOQs in IMs are re-denoted as $VOQ(i,v)$, where $v = hk+j$ and $0 \leq v \leq nk-1$) and each VOQ has a pointer $P_V(i,v)$. In $CM(r)$, there are k round-robin arbiters, which have their own pointer $P_C(r,j)$

For each VOQ there is a captured frame-size counter, $CF_{i,j,l}(t)$. The value of $CF_{i,j,l}(t)$, indicates the frame size at time slot t; that is, the maximum number of cells that a $VOQ(i,j,l)$ can have as matching candidates in the current and future time slots. $CF_{i,j,l}(t)$ takes a new value when the last cell of the current frame of $VOQ(i,j,l)$ is matched. $CF_{i,j,l}(t)$ decreases its count each time a cell is matched, other than the last.

The arbitration process includes two phases. This scheme follows request-grant-accept approach, as in the CRRD algorithm [13]:

Phase 1: Matching within IM

- First iteration

- Step 1: Non-empty VOQs send a request to the output-link arbiter L_I, where each request indicates the on-service or off-service status of the VOQ.
- Step 2: If an output-link arbiter receives any request, it chooses an on-service request in a round-robin fashion starting from the position of $P_L(i,r)$. If none on-service request exists, the L_I chooses an off-service request in a round-robin fashion starting from the position of $P_L(i,r)$. L_I then sends a grant to the selected VOQ.
- Step 3: If the VOQ arbiter receives any grant, it accepts an on-service grant in a round-robin fashion, starting from the position of $P_V(i,v)$. If none on-service grant exists, the VOQ arbiter accepts an off-service grant that appears next in round-robin schedule, starting from the position of $P_V(i,v)$.

- ith iteration

- Step 1: Each unmatched VOQ sends another request to all unmatched output-link arbiters.
- Step 2 and 3: The same procedure is performed as in the first iteration for matching between unmatched nonempty VOQs and unmatched output links.

Phase 2: Matching between IM and CM

- Step 1: After phase 1 is complete, $L_I(i,r)$ sends the request to $CM(r)$. Each round-robin arbiter associated with $OM(j)$ then chooses a request from the on-service $L_I(i,r)$ that appears next in a round-robin schedule, starting from the position $P_C(r,j)$ and sends the grant to $L_I(i,r)$ of IM. $P_C(r,j)$ is updated to one position beyond the granted one. If none on-service request exists, the $OM(j)$ chooses an off-service request that appears next in a round-robin schedule, starting from its position $P_C(r,j)$ and sends the grant to $L_I(i,r)$.
- Step 2: If the IM receives the grant from the CM, P_V and P_L are updated to one position beyond the granted link and VOQ, respectively. IM sends the corresponding cell from that VOQ at the next time slot. The request from the CM that is not granted will be attempted again at the next time slot because the pointers that are related to the ungranted requests are not moved. In addition to the pointer update, the $CF_{i,j,h}$ counter updates the value according to the following:
  If an IM received a grant from a CM, the counters are updated as follows:
  i) If $CF_{i,j,l}(t) > 1$: $CF_{i,j,h}(t+1) = CF_{i,j,l}(t) - 1$ and this $VOQ(i,j,l)$ is set as on-service.
  ii) else ($CF_{i,j,l}(t) = 1$): $CF_{i,j,l}(t+1)$ is assigned the occupancy of $VOQ(i,j,l)$, and $VOQ(i,j,l)$ is set as off-service.

Note that the matching within IM can have several iterations as the arbiters can be placed in the IM modules. The matching between IM and CM is considered with one iteration only as, depending on the implementation, the IM and CM modules may be located far from each other.

VII. Simulation Evaluation of FCRRD

As in the case for single-stage IQ switches, we study the performance of the proposed scheme using computer simulation. We compare the performance of the RD scheme [21], and our framed version of it, FRD (the description of this scheme is similar to that of FCRRD with the difference that selections are random based instead). Here, we consider CRRD and FCRRD with multiple iterations in IM, which are denoted as I_{IM}, and only a single iteration between IMs and CMs. FRD, as RD, assumes that up to r non-empty VOQs are matched, disregarding of the number of iterations, and therefore, we don't indicate the number of iterations in the results. The traffic models considered have destinations with uniform and nonuniform distributions and Bernoulli and bursty arrivals. The bursty traffic follows an on-off Markov modulated process and has an average burst length, l, of 10 cells. 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 16 shows the simulation results of RD, FRD, CRRD and FCRRD, all under uniform traffic with Bernoulli arrivals in a $n = m = k = 8$ switch (i.e., 64×64 switch without internal expansion). This figure shows that FRD delivers higher performance than RD, and therefore, showing the improving effect of occupancy-based framing. This figure also shows that FCRRD, as CRRD, delivers 100% throughput with any number of iterations under this traffic type. The average delay of FCRRD with two iterations is lower than that of CRRD with four iterations. Therefore, FCRRD converges when the number of iterations in the IM reaches two. The reason for this improvement is that once a match is achieved, the match is kept during the frame duration. Therefore, contention in IM is reduced as the number of participant VOQs is reduced with each match. Figure 17 shows that FCRRD provides 100% throughput under Bernoulli uniform traffic while using different switch sizes $n = m = k = \{2, 4, 8, 16\}$. The average delay increases when the switch size increases.

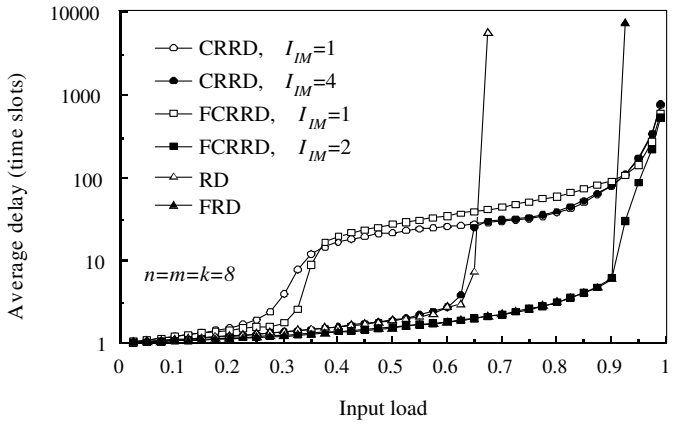

Fig. 16. Average delay of FCRRD and CRRD schemes (*n=m=k=8*) under Bernoulli uniform traffic with multiple iterations.

Figure 18 shows that FCRRD provides 100% throughput even when the input traffic is bursty, with $l = 10$. As the figure shows, the average delay of FCRRD with one and two iterations is smaller than that of CRRD with any number of iterations.

B. Nonuniform Traffic

As in the case for single-stage switches, we simulate RD, FRD, CRRD and FCRRD with multiple iterations under four nonuniform traffic patterns: unbalanced, Chang's, asymmetric, and diagonal [16].

Figure 19 shows the throughput performance of RD, FRD, CRRD, and FCRRD under unbalanced traffic. The throughput of RD when $w = 0$ is about 65% and increases as w increases. However, FRD is above 90% when $w = 0$ and increases to 100% as w increases, because of the effect of the occupancy-based framing. This figure also shows that the

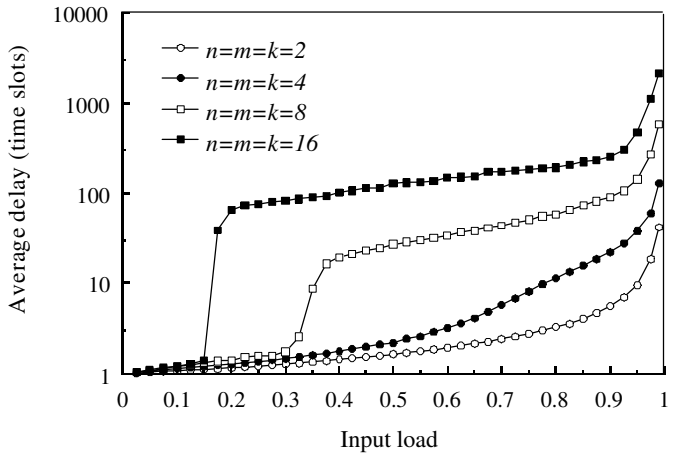

Fig. 17. Average delay of FCRRD scheme with different switch sizes under Bernoulli uniform traffic.

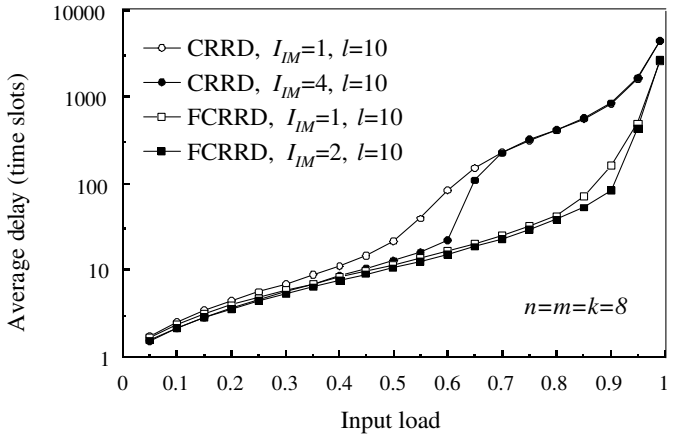

Fig. 18. Average delay of FCRRD and CRRD schemes (n=m=k=8) under bursty traffic with multiple iterations.

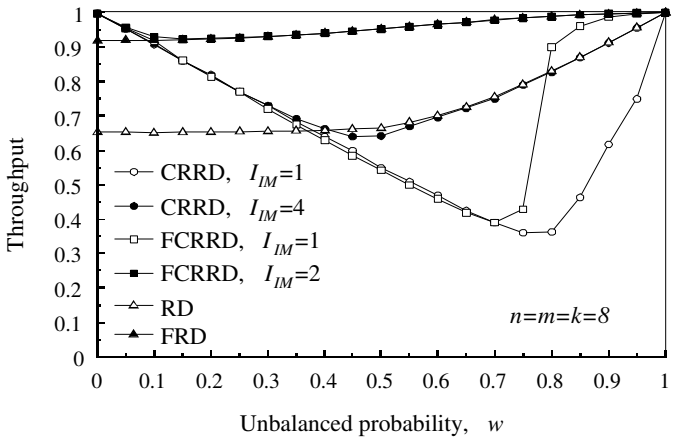

Fig. 19. Throughput performance of FCRRD and CRRD (n=m=k=8) with multiple iterations under unbalanced traffic.

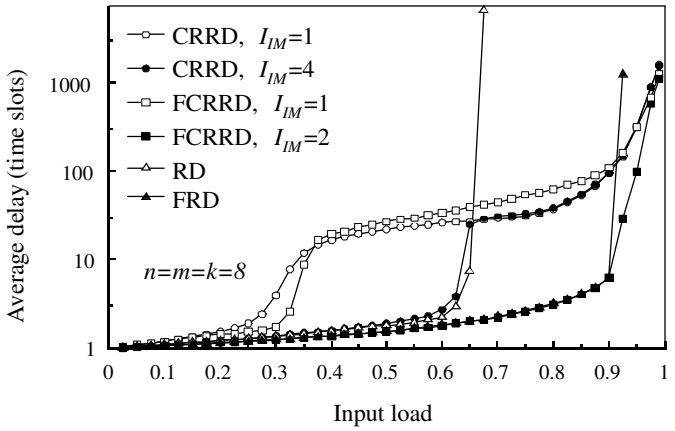

Fig. 20. The performance of FCRRD and CRRD (n=m=k=8) under Chang's traffic.

throughput of FCRRD with two iterations is higher than that of CRRD with four iterations under the complete range of w. The high throughput of FCRRD under this traffic model is the product of considering the VOQ occupancy and traffic isolation. FCRRD, as 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. The captured-frame size allows the scheduler to isolate the stored cells from incoming cells that could make the queuing delay longer, as observed in single-stage switches.

Figure 20 shows the performance of our dispatching schemes under Chang's traffic model. This figure shows that RD has about 63% throughput while FRD can achieve about 91% throughput under this traffic pattern. This figure also shows that the average cell delay of FCRRD with two iterations is higher than that of CRRD with 4 iterations under Chang's traffic, as seen in the previous traffic models.

Figure 21 shows the simulation results under asymmetric traffic. These results show that RD delivers about 75% throughput as CRRD with four iterations, while CRRD and FCRRD, both with one iteration, deliver below 45% throughput. FRD and FCRRD with two iterations provide close to 100% throughput. This figure shows that random selection is as effective as round-robin selection.

Figure 22 shows the simulation results under diagonal traffic. These results show that FCRRD, with two iterations, delivers higher throughput than FCRRD with one iteration and than CRRD with any number of iterations. Also, RD and FRD show higher throughput than round-robin based schemes. The performance of FRD is comparable to that of the FCRRD with two iterations, of about 95% throughput. One of the reasons for this improvement is that the pointer update in round-robin schemes might not provide effective desynchronization of pointers as traffic is directed to only two different outputs per input.

The use of the captured frame and the service concepts make FRD and FCRRD deliver high switching performance under uniform and nonuniform traffic patterns. This is because when a VOQ is on-service status, service remains for the next time slots until the current frame is depleted.

VIII. Conclusions

In this paper, we introduced the captured frame size concept to determine cell eligibility in the matching process for input queued packet switches. Also, we proposed two matching

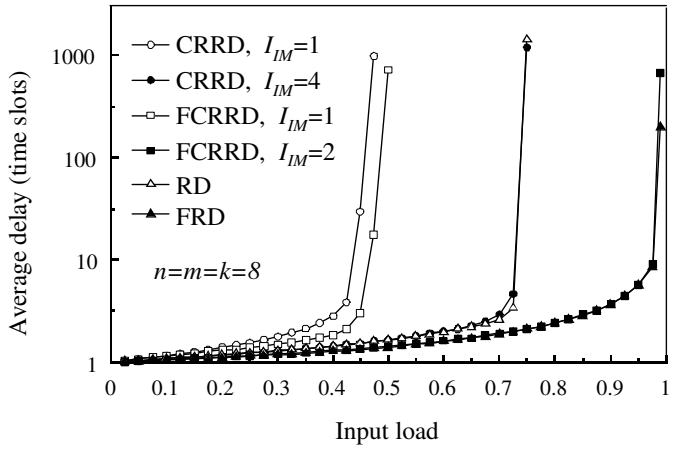

Fig. 21. The performance of FCRRD and CRRD (n=m=k=8) under asymmetric traffic.

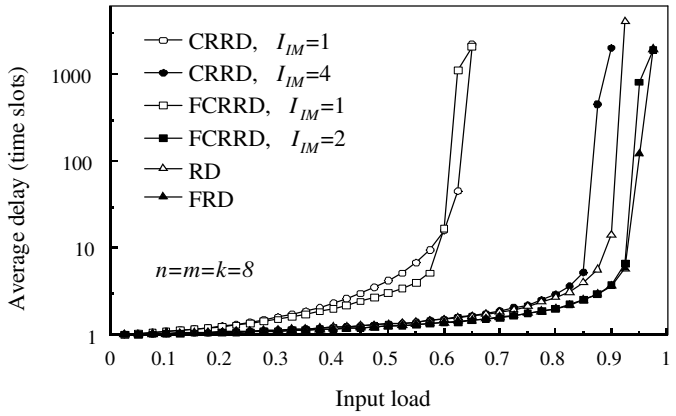

Fig. 22. The performance of FCRRD and CRRD (n=m=k=8) under diagonal traffic.

schemes, uFORM and uFPIM, that use the captured frame concept, a single iteration, and no speedup, both for single-stage IQ switches. We analyzed the throughput performance of uFPIM under uniform traffic with independent and identical distributions, and showed that uFPIM can achieve higher throughput than PIM. Furthermore, we tested the proposed schemes under several nonuniform traffic patterns. The presented schemes show above 99% throughput under the unbalanced traffic model, using a single iteration and no speedup. uFORM and uFPIM were also studied under Chang's, asymmetric, and PO2 traffic models and these schemes showed higher switching performance than those schemes without the captured-frame concept. The new scheme give similar performance to that of weight-based matching schemes under nonuniform traffic patterns without recurring to queue comparisons and keep the high throughput of weightless schemes under uniform traffic. Furthermore, the proposed concept is scalable as the throughput performance increases as the switch size increases.

In addition, we adopted the captured frame concept in multiple-stage Clos-network switches as the resolution time in large scale multiple-stage switches may be affected more than in single-stage switches by time-consuming schemes. Schemes based on the proposed concept decrease the needed number of iterations performed in Clos-network switches to achieve high throughput. For this, we looked into the adaptation of this concept into RD and CRRD dispatching schemes, and therefore giving place to the creation of two dispatching schemes, FRD and FCRRD. These schemes use random and round-robin selection, respectively. As compared to RD and CRRD, FRD and FCRRD show higher performance under several nonuniform traffic patterns. Furthermore, FCRRD keeps 100% throughput under uniform traffic as CRRD does. FCRRD, with two iterations in IMs, is sufficient to achieve a high switching performance under nonuniform traffic. This reduction of number of iterations is important in Clos-network switches as the input modules are located in different physical locations from the central modules. As uFPIM and uFORM for single-stage switches, FRD and FCRRD do not need to compare the status of different VOQs as they are based in random and round-robin selection, respectively.

The complexity of adding the captured frame concept to matching schemes is low as the schemes do not need to compare the status of the VOQs. The addition of the hardware and timing complexity of schemes are negligible as only the update of the CF counters and F flags is added to the implementation, and the time to update CF and F can be performed in parallel with the pointer update in FCRRD.

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] N. McKeown, "Scheduling algorithms for input-queued cell switches," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. California at Berkeley, Berkeley, CA, 1995.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] C-S. Chang, D-S. Lee, and Y-S. Jou, "Load Balanced Birkhoff-von Newman 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] C. Clos,"A study of nonblocking switching networks," *Bell Syst. Tech. J.*, pp. 406-424, March 1953.

[13] E. Oki, Z. Jing, R. Rojas-Cessa, and H. J. Chao, "Concurrent roun-robin dispatching scheme for Clos-network switches," *IEEE/ACM Trans. Networking*, vol. 10, no. 6, pp. 830-844, May 2001.

[14] T. T. Lee, and S-Y Liew, "Parallel Routing Algorithm in Benes-Clos Networks," *IEEE INFOCOM' 96*, pp. 279-286, 1996.

[15] H.J. Chao, S.Y. Liew, and Z. Jing, "A dual-level Matching algorithm for 3-stage Clos-network Packet Switches," 11th Symposium on High Performance Interconnects, pp. 38-44, August 2003.

[16] K. Pun and M. Hamdi, ”Static Round-Robin Dispatching Schemes for Clos-Network Switches, in Proc. IEEE HPSR 2002, pp. 239-243, May 2002.
[17] 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 .
[18] 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.
[19] 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.
[20] 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
[21] F. M. Chiussi, J. G. Kneuer, and V. P. Kumar “Low-cost scalable switching solutions for broadband networking: the ATLANTA architecture and chipset,” *IEEE Communications Mag.*, pp. 44-53, Dec. 1997.