Frame Occupancy-Based Dispatching Schemes for Buffered Three-stage Clos-Network Switches

Chuan-Bi Lin and Roberto Rojas-Cessa

***Abstract*— The three-stage Clos-network switch architecture has attractive scalability features that makes it appealing as an alternative for scalable switches. However, scheduling packets in a Clos-network switch is complex. This complexity can be simplified by adding buffers to the first and third stages. By adding these buffers, the scheme used for dispatching packets from the first stage of the switch becomes important. Several dispatching schemes for Clos-network, without internal expansion, that deliver high throughput under uniform traffic model have been proposed. However, there is a need of dispatching schemes that provide high throughput under several admissible traffic patterns, including those with nonuniform distributions, with a small number of matching iterations and without internal expansion. In this paper, we propose two frame-occupancy based dispatching schemes to increase throughput performance in Clos-network switches without using internal expansion. We show that frame-occupancy based schemes deliver high throughput under uniform and nonuniform traffic patterns.**

***Index Terms*— Dispatching, frame, Clos-network, admissible traffic, round-robin.**

I. Introduction

There are two broad approaches to implement a high-performance switch: single and multiple stages. Single-stage switches are mainly based on crossbar switch fabrics. Several single-stage high-speed switch are described in [1], [2]. However, the single-stage approach makes it difficult to implement a large-scale switch, in terms of the number of ports, because a larger number of switch chips are needed to form a bi-dimensional array of chips.

A multiple-stage switch, such as a three-stage Clos-network switch [3], needs fewer switch chips for implementing a switch with large number of ports. This makes the Clos-network switch very attractive for scalable switches.[1] We consider two broad types of Clos-network switches: bufferless and buffered. A bufferless Clos-network switch has no memory in any of the three stages. Although the design of the switch modules is rather simple, this switch may require a complex matching process and a long resolution time. A variety of matching schemes for bufferless Clos-network switches have been proposed [4], [5].

Here, we consider that variable-length packets are segmented into fixed-size packets or cells at the input side of the switch, and re-assembled at the output side of the switch, before departure.

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

The authors are with Department of Computer and Electrical Engineering, New Jersey Institute of Technology Newark, NJ 07102 Email: {cl23, rrojas}@njit.edu

[1]Clos-network switches also use a smaller number of crosspoint elements.

Within the buffered Clos-network switches, we can categorize a switch into two types: one has the buffers in the second-stage modules and the other has no buffers in the second-stage modules. Implementing buffers in the second-stage modules helps to resolve contention among cells from different first-stage modules [6]. However, switches with buffers in the second-stage modules may suffer from serving packets in out-of-sequence order, which is undesirable as re-sorting packets might increase the switch complexity and cost. The other option is to use a switch with bufferless second-stage modules, where buffers are only placed in the first and third stages. This architecture, which avoids the out-of-sequence problem, is called a buffered Clos-network switch [7]. We use this definition in the remainder of this paper. By adding buffers to the first stage of the switch, a dispatching scheme needs to be used to avoid contention within the input module. This matching is implemented as a matching process.

There are several studies on matching schemes for dispatching packets from the first stage of buffered Clos-network switches. As in single-stage switches, a maximum-weight matching dispatching (MWMD) scheme has been used in Clos-network switches to provide high throughput under admissible traffic [8], but the MWMD scheme has high computation complexity that could slow down high-speed switches. An alternative is to use maximal-weight matching dispatching schemes. However, the hardware and time complexity of these schemes can be considered high for the ever increasing data rates. Schemes based in round-robin dispatching matching, which are maximal-size matching schemes, such as CRRD [9], have been proposed to deliver 100% throughput under uniform traffic and with a low implementation complexity. CRRD showed that the desynchronization effect, where arbiters reach the point where each of them prefers to match with different input/outputs, improves switching performance under uniform traffic. However, CRRD has a limited throughput under some nonuniform traffic patterns.

Frame-based scheduling with fixed-size frames has been shown to improve switching performance [10]. However, how to choose a suitable frame size is complex. In this paper, we apply framing, based on queue occupancy citerrc, to improve throughput under under nonuniform traffic patterns, without allocating any buffers in the second stage to avoid the out-of-sequence problem, and to offer a low implementation complexity. Here, we introduce the frame occupancy-based random dispatching (FRD), which is based on random dispatching [7], and the frame-occupancy concurrent round-robin dispatching (FCRRD) scheme, which is based on the CRRD scheme and on the captured-frame concept [11]. This paper shows that the captured-frame concept, used for matching

eligibility, improves the performance of dispatching schemes. We use the RD scheme and FRD for this purpose. In addition, our results show that FCRRD can achieve higher throughput than that of CRRD under nonuniform traffic patterns, while retaining the high performance under uniform traffic and the low implementation complexity of round-robin schemes.

This paper is organized as follows. Section II presents the Clos-network switch model and preliminary definitions, used along this paper. Section III proposes the captured frame eligibility and the frame occupancy-based round-robin dispatching scheme. Section IV presents the performance study of the proposed scheme under uniform and nonuniform traffic patterns. Section V presents the conclusions.

II. Clos-Network Switch Model and Preliminary Definition

The Clos-network switch is a three-stage switch architecture [3], as Figure 1 shows. We use the same terminology in [9], 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 input modules (IM), m central modules (CM), and k output modules (OM). An $IM(i)$ has n input

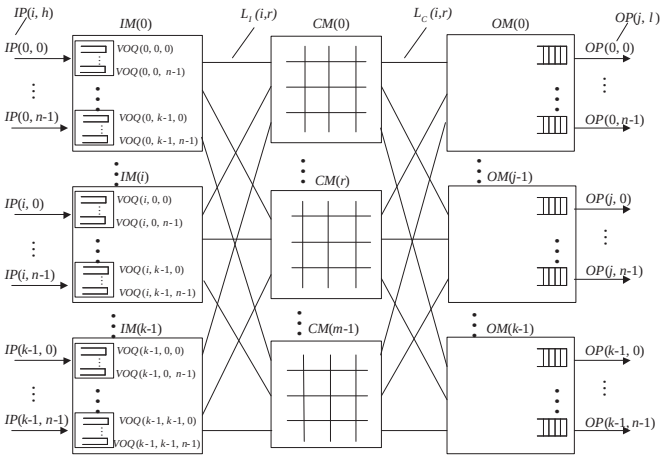

Fig. 1. Clos-network switch with VOQs in the IMs.

ports, each of which is denoted as $IP(i,h)$. Each $IM(i)$ has $n \times k$ VOQs to eliminate head-of-line (HOL) blocking. 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.

The following definitions, adapted from [11], are used in the description of the proposed dispatching scheme.

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.

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 $VOQ(i,j,l)$, the next frame is assigned a size equal to the cell occupancy at $VOQ(i,j,l)$. Cells arriving to $VOQ(i,j,l)$ 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.

III. 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 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 [9]:

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. Otherwise, the IM cannot send the cell at the next time slot. The request from the CM that is not granted will again be attempted 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. If:
  i) $CF_{i,j,l}(t) > 1$: $CF_{i,j,h}(t+1) = CF_{i,j,l}(t) - 1$ and this $VOQ(i,j,l)$ is set as on-service.
  ii) If $CF_{i,j,l}(t) = 1$: $CF_{i,j,l}(t+1)$ 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.

IV. Simulation evaluation

We studied the performance of the proposed scheme using computer simulation. We compare the performance of the RD scheme [7], and our framed version of it, FRD. 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 2 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 2 iterations is lower than that of CRRD with 4 iterations. Different from CRRD, 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 3 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.

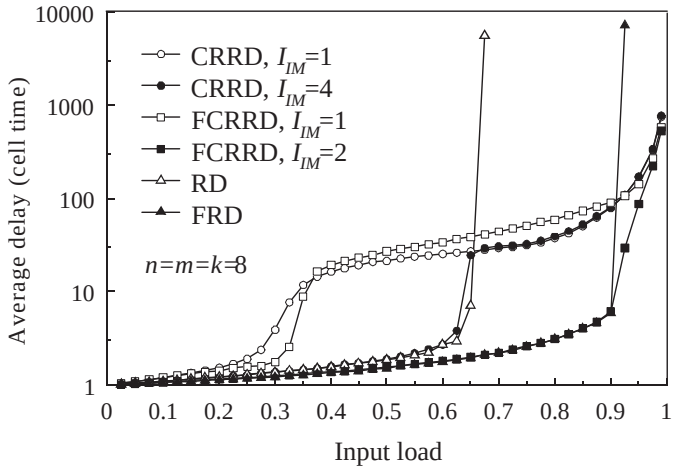

Fig. 2. Average delay of FCRRD and CRRD schemes (*n=m=k=8*) under Bernoulli uniform traffic with multiple iterations.

Figure 4 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 2 iterations is smaller than that of CRRD with any number of iterations.

B. Nonuniform Traffic

We simulated the RD, FRD, CRRD and FCRRD schemes with multiple iterations under four different nonuniform traffic patterns: unbalanced [12], Chang's [13], asymmetric [14] and bi-diagonal [15].

The unbalanced traffic model uses a probability, w, as the fraction of the input load directed to a single predetermined output, while the rest of the input load is directed to all outputs

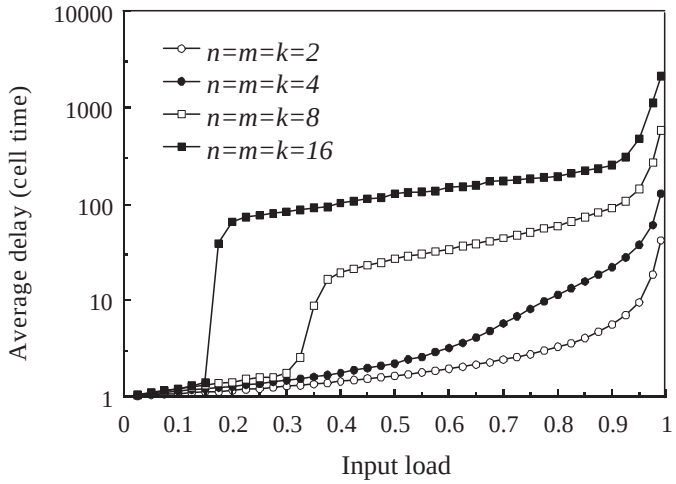

Fig. 3. Average delay of FCRRD scheme with different switch sizes under Bernoulli uniform traffic.

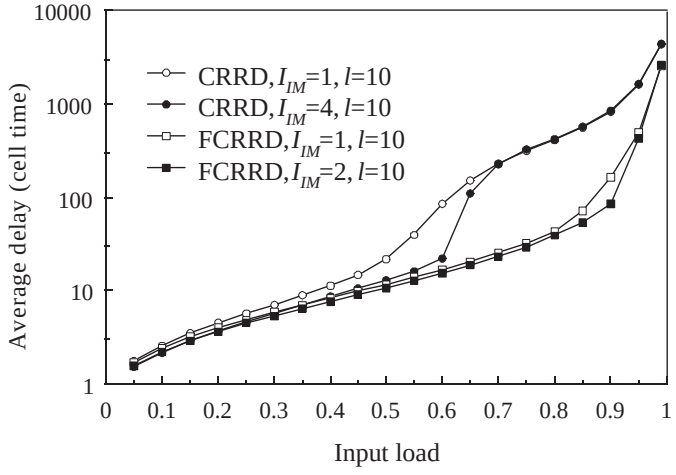

Fig. 4. Average delay of FCRRD and CRRD schemes (*n=m=k=8*) under bursty traffic with multiple iterations.

with uniform distribution. Let us consider input port s, output port d, and the offered input load for each input port is ρ. The traffic load from input port s to output port d, $\rho_{s,d}$ is given by,

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

Where N (i.e., nk) is the switch size. When $w = 0$, the offered traffic is uniform. On the other hand, when $w = 1$, the traffic is completely directional. This means that all traffic of input port s is destined for output port d, where $s = d$.

Figure 5 shows the throughput performance of RD, FRD, CRRD, and FCRRD under unbalanced traffic. The throughput of RD when $w = 0$ is about 65% and starts increasing 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 throughput of FCRRD with 2 iterations is higher than that of CRRD with 4 iterations under the complete range of w. The high throughput of FCRRD under this traffic model is

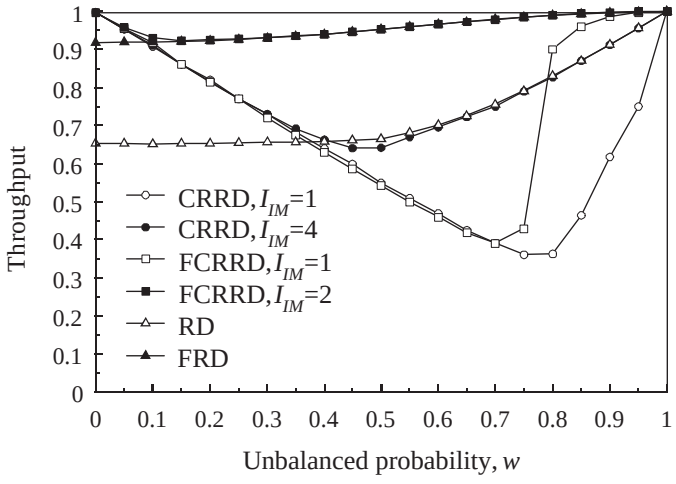

Fig. 5. Throughput performance of FCRRD and CRRD (*n=m=k=8*)with multiple iterations under unbalanced traffic.

the product of considering the VOQ occupancy and traffic isolation. FCRRD 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 isolate the buffered cells from incoming cells that could make the queuing delay longer.

Another non-uniform traffic pattern is Chang's traffic model, which is defined as $\rho_{i,j} = 0$ when $i = j$, and $\rho_{i,j} = 1/(N-1)$, otherwise, where $N = nk$ is switch the size and $\rho_{i,j}$ is the input load. Figure 6 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 2 iterations is higher than that of CRRD with 4 iterations under Chang's traffic, as seen in the previous traffic models.

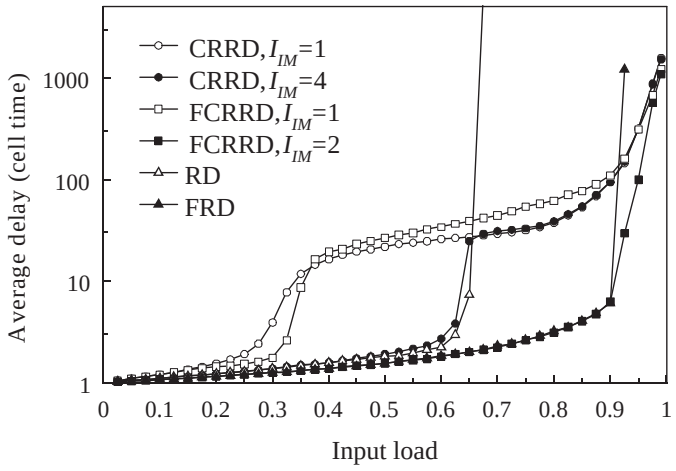

Fig. 6. The performance of FCRRD and CRRD (*n=m=k=8*) under Chang's traffic.

Figure 7 shows the simulation results under asymmetric traffic. This figure shows that RD delivers about 75% throughput as CRRD with 4 iterations, while CRRD and FCRRD, both with 1 iteration, deliver below 45% throughput. FRD and

FCRRD with 2 iterations provide close to 100% throughput. This figure shows that random selection is as effective as round-robin selection.

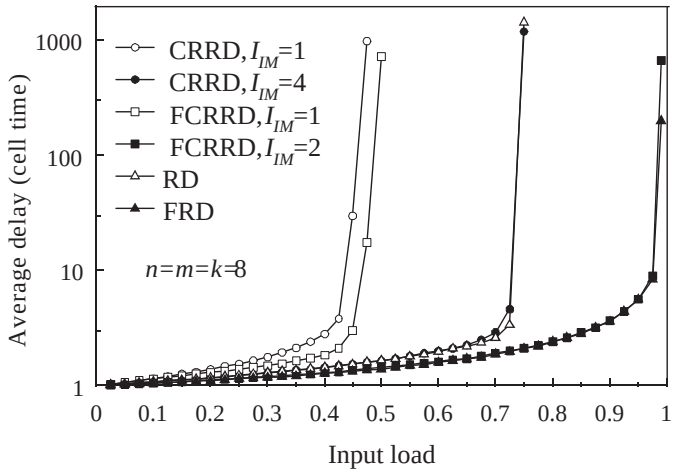

Fig. 7. The performance of FCRRD and CRRD (*n=m=k=8*) under asymmetric traffic.

The last non-uniform traffic pattern considered here is the bi-diagonal traffic model, which is defined as $\rho_{i,j} = 0.5$ for $j = i$ and $j = (i+1) \bmod N$, and $\rho_{i,j} = 0$ otherwise. Figure 8 shows that FCRRD, with 2 iterations, delivers higher throughput than FCRRD with 1 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 2 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.

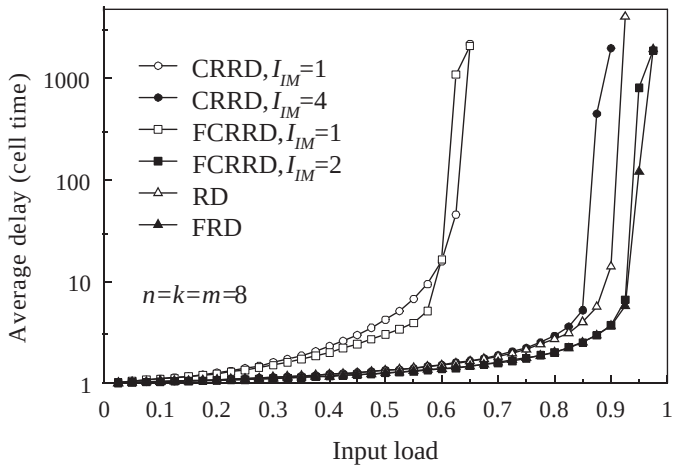

Fig. 8. The performance of FCRRD and CRRD (*n=m=k=8*) under bi-diagonal traffic.

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

V. Conclusions

In this paper, we proposed two dispatching schemes for Clos-network switches FRD and FCRRD. These schemes use random and round-robin selection, respectively, and the concept of unlimited captured frame-size, where the frame size depends on VOQ occupancy at completed-service time. As compared to RD and CRRD, FRD and FCRRD show higher performance under several nonuniform traffic patterns. Furthemore, FCRRD keeps 100% throughput under uniform traffic as CRRD does. We also showed that FCRRD, with 2 iterations, is sufficient to achieve a high switching performance. The reduction of the number of iterations is important in Clos-network switches as the input modules are located in different physical locations from the central modules. The FRD and FCRRD schemes do not need to compare the status of different VOQs as they are based in random and round-robin selection, respectively. The hardware and timing complexity of FCRRD is comparable to that of CRRD because only the frame counters and the on/off service flags are added.

Acknowledgment

The authors thank Eiji Oki for graciously sharing the CRRD program.

References

[1] N. McKeown, M. Izzaed, A. Mekkittikul, W. Ellerisick, and M. Horowitz, "Tiny-Tera: A packet switch core," *IEEE Micro.*, pp. 26-33, Jan.-Feb. 1997.

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

[3] C. Clos,"A study of nonblocking switching networks," *Bell Syst. Tech. J.*, pp. 406-424, March 1953.

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

[5] K. Pun and M. Hamdi, "Distro: A Distributed Static Round-robin Scheduling Algorithm for Bufferless Clos-network switches," *IEEE Globecom 2002*, vol. 3, pp. 2298-2302, 2002.

[6] J. Turner, and N. Yamanaka, "Architectural Choices in Large Scale ATM switches," *IEICE Trans. Commun.*, vol. E81-B, no. 2, pp. 120-137, Feb. 1998.

[7] 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.

[8] R. Rojas-Cessa, E. Oki, and H. J. Chao, "Maximum weight matching dispatching scheme in buffered Clos-network packet switches," *IEEE International Conf. on Commun.*, vol. 2, pp. 1075-1079, June. 2004.

[9] 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.

[10] 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.

[11] R. Rojas-Cessa, and C. Lin, "Captured-frame Eligibility and Round-Robin Matching for Input-Queued Packet Switches," *Communications Letters, IEEE*, Vol. 8, pp. 585-587, Sept. 2004.

[12] 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.

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

[14] 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

[15] K. Pun and M. Hamdi, "Static round-robin dispatching schemes for Clos-network switches," *IEEE HPSR*, pp. 239 - 243, May 2002.