Providing Guaranteed Rate Services in the Load Balanced Birkhoff-von Neumann Switches

Cheng-Shang Chang, Duan-Shin Lee, and Chi-Yao Yue
Institute of Communications Engineering
National Tsing Hua University
Hsinchu 300, Taiwan, R.O.C.
Email: cschang@ee.nthu.edu.tw
lds@cs.nthu.edu.tw
cyyue@gibbs.ee.nthu.edu.tw

***Abstract*— In this paper, we propose two schemes for the load balanced Birkhoff-von Neumann switches to provide guaranteed rate services. As in [7], the first scheme is based on an Earliest Deadline First (EDF) scheduling policy. In such a scheme, we assign every packet of a guaranteed rate flow a *targeted departure time* that is the departure time from the corresponding work conserving link with capacity equal to the guaranteed rate. By adding a jitter control mechanism in front of the buffer at the second stage and running the EDF policy at the output buffer, we show that the end-to-end delay for every packet of a guaranteed rate flow is bounded by the sum of its targeted departure time and a constant that only depends on the number of flows and the size of the switch.**

Our second scheme is a frame based scheme as in Keslassy and McKeown [18]. There, time slots are grouped into fix size frames. Packets are placed in appropriate bins (buffers) according to their *arrival times* and their *flows*. We show that if the incoming traffic satisfies certain assumptions, then the end-to-end delay for every packet and the size of the central buffers are both bounded by constants that only depend on the size of the switches and the frame size. The second scheme is much simpler than the first one in many aspects: (i) the on-line complexity is $O(1)$ as there is no need for EDF, (ii) central buffers are finite and thus can be built into a single chip, (iii) connection patterns of the two switch fabrics are changed less frequently, (iv) there is no need for resequencing-and-output buffer after the second stage, and (v) variable length packets may be handled without segmentation and reassembly.

***Index Terms*— guaranteed rate services, Birkhoff-von Neumann switches, multicasting flows, variable length packets, multi-stage switches**

I. Introduction

In order to provide the needed speedup to match the speed of fiber optics, parallel buffered switches, capable of performing parallel read/write, have received a lot of attention recently (see e.g., [15], [16] and references therein). Traditionally, the study of parallel buffered switches is limited to the (single-stage) input-buffered crossbar switch (see e.g., [17], [22], [27], [23], [19], [9], [24], [28], [14], [20], [1], [21]), where each input has a segregated buffer. In such a switch, time is slotted and synchronized so that packets in different input buffers can be read out simultaneously within a time slot. There are two well-known problems in an input-buffered switch: low throughput due to head-of-line (HOL) blocking and the difficulty in controlling packet delay. The HOL problem can be solved by using the virtual output queueing (VOQ) technique. Instead of having a single First Come First Serve (FCFS) queue at each input port, the VOQ technique maintains a separate (logical) queue for each output port at each input port.

This research is supported in part by the National Science Council, Taiwan, R.O.C., under Contract NSC-91-2219-E007-003, and the program for promoting academic excellence of universities 89-E-FA04-1-4.

To control packet delay, one easy solution is to provide bandwidth guarantees in an input-buffered switch. In the paper [14], Hung, Kesidis and McKeown used an idling weighted round robin (WRR) algorithm in [2] to achieve rate guarantee for each input-output pair without internal speedup. Similar approaches are also addressed in [19], [20]. As the usual WRR algorithm, all these are frame based schemes and might have the granularity problem for bandwidth guarantees.

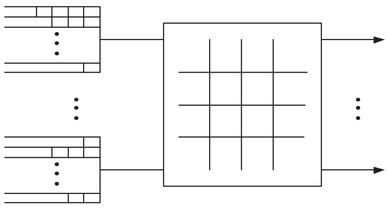

Fig. 1. The architecture of the Birkhoff-von Neumann switch

To cope with the granularity problem due to framing, the Birkhoff-von Neumann input-buffered switch is proposed in [5] and [6] for guaranteed rate service between each input-output pair (see Figure 1). As in most input-buffered switch, the Birkhoff-von Neumann switch uses the VOQ technique to solve the HOL blocking problem. The main idea of scheduling the connection patterns in the Birkhoff-von Neumann switch is to use the capacity decomposition approach by Birkhoff [3] and von Neumann [30] (for the details of the decomposition algorithm, we refer to [5] and [6]). The computational complexity of the decomposition is $O(N^{4.5})$ for an $N \times N$ switch. The on-line scheduling algorithm used there is a simplified version of the Packetized Generalized Processor (PGPS) algorithm in Parekh and Gallager [26] (or the Weighted Fair Queueing (WFQ) in Demers, Keshav, and Shenkar [12]). The complexity of the on-line scheduling algorithm is $O(\log N)$.

There are several drawbacks of the Birkhoff-von Neumann switches:

(i) Computational complexity: the Birkhoff-von Neumann decomposition itself is non-trivial (with the order of complexity $O(N^{4.5})$), even though such a decomposition only needs to be computed when the rates change.

(ii) Memory complexity: the number of permutation matrices generated from the Birkhoff-von Neumann decomposition is $O(N^2)$. These matrices have to be stored in the switch.

(iii) Multicast: the Birkhoff-von Neumann switch does not support multicast. Multicasting flows can only be supported through point-to-point flows.

(iv) Variable length packets: in the Birkhoff-von Neumann switch, time is slotted and packets are assumed to fit in a time slot. Variable length packets have to be segmented at the inputs and then re-assembled at the outputs.

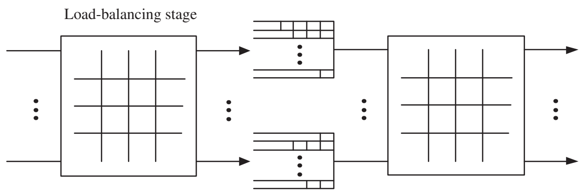

Fig. 2. The load balanced Birkhoff-von Neumann switch with one-stage buffering

To cope with the first three drawbacks in the Birkhoff-von Neumann switch, the load balanced Birkhoff-von Neumann switch with one-stage buffering is proposed in [7]. The main idea is to add a load balancing stage in front of the Birkhoff-von Neumann input-buffered switch (see Figure 2). In a time slot, the crossbar switch at the first stage sets up connection patterns corresponding to permutation matrices that are periodically generated from a one-cycle permutation matrix. By so ding, the first stage performs load balancing for the incoming traffic. As the traffic coming into the second stage is load balanced, it suffices to use the same simple periodic connection patterns as in the first stage to perform switching at the second stage. Thus, there is no need to carry out the Birkhoff-von Neumann decomposition. To support multicast, fan-out splitting is done at the central buffer (the buffer between two crossbars). It is shown in [7] that the load balanced Birkhoff-von Neumann switch indeed achieves 100% throughput (under a mild technical condition) for both point-to-point and multicasting flows. However, the main drawback of the load balanced Birkhoff-von Neumann switch with one-stage buffering in [7] is that packets might be out of sequence.

In [8], the load balanced Birkhoff-von Neumann switch with multi-stage buffering is proposed to solve the out-of-sequence problem. There, load-balancing buffers are added in front of the first switch and resequencing-and-output buffers are added after the second switch. As in [16], packets are distributed in the round-robin fashion according to their flows in the load balanced Birkhoff-von Neumann switch with multi-stage buffering. By so doing, it is shown in [8] that the delay through the first stage can be bounded by a constant that only depends on the size of the switch and the number of flows supported by the switch.

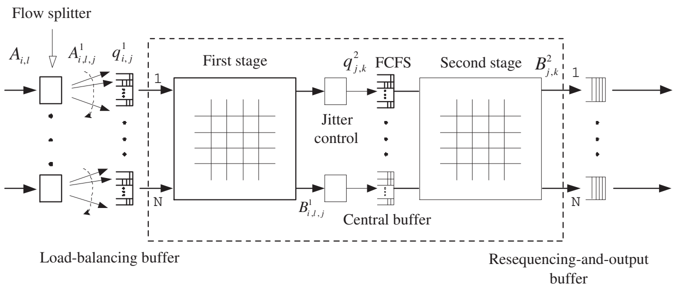

Fig. 3. The load balanced switch with multi-stage buffering under FCFS

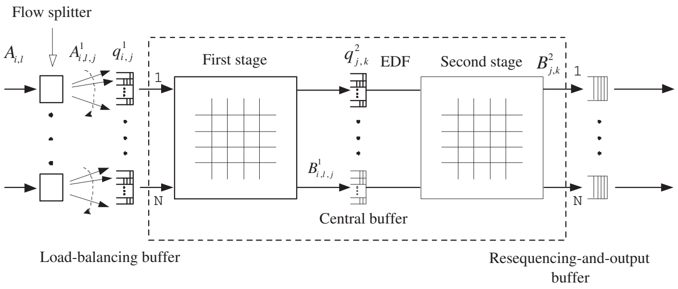

Fig. 4. The load balanced switch with multi-stage buffering under EDF

Two scheduling policies in the central buffers are presented in [8]: the First Come First Serve (FCFS) policy (see Figure 3) and the Earliest Deadline First (EDF) policy (see Figure 4). For the FCFS policy, a jitter control mechanism is added in the VOQ in front of the second stage. It delays every packet to its maximum delay at the first stage so that the flows entering the second stage are simply time-shifted flows of the original ones. For the EDF policy, every packet is assigned a deadline that is the departure time from the corresponding output-buffered switch. The central buffers then schedule packets according to their deadlines.

After the second stage, packets are stored in the resequencing-and-output buffer. The resequencing-and-output buffer conceptually consists of two virtual buffers: (i) the resequencing buffer and (ii) the output buffer. The objective of the resequencing buffer is to reorder the packets so that packets of the same flow depart in the same order as they arrive. After resequencing, packets are stored in the output buffer waiting for transmission from the output link. It is shown in [8] that for both the FCFS and EDF schemes the end-to-end delay is bounded above by the sum of the delay through the corresponding FCFS output-buffered switch and a constant that depends on the size of the switch and the maximum number of flows supported by the switch. Moreover, the size of the resequencing-and-output buffer for the FCFS (resp. EDF) policy is also bounded above by a constant that depends on the size of the switch and the maximum number of flows supported by the switch. In short, *the load balanced*

Birkhoff-von Neumann switch with multi-stage buffering is able to emulate the ideal FCFS output-buffered switch up to a constant delay, and this is done without speedup and conflict resolution. We also note the idea of using load balancing was previously explored in the literature via *randomization* (see e.g., [29], [25]). However, load balancing via randomization does not yield deterministic bounds.

The drawback of the load balanced Birkhoff-von Neumann switch with multi-stage buffering is its hardware implementation complexity for the resequencing-and-output buffer and the jitter control mechanism. In [18], Keslassy and McKeown developed a clever scheme that uses the Full Frame First (FFF) scheduling policy in the central buffers. In such a scheme, packets of the same flow at the central buffers are grouped into frames with frame size equal to the number of inputs. By so doing, packet of the same flow depart in the FCFS order. As such, there is no need for the resequencing-and-output buffer.

The load balanced Birkhoff-von Neumann switches in [7], [8], [18] only provide the best effort service. The main objective of this paper is to investigate schemes for providing guaranteed rate services in the load balanced Birkhoff-von Neumann switches. We develop two schemes for doing this. As in [8], the first scheme is based on an Earliest Deadline First (EDF) scheduling policy. Instead of using the departure time from the corresponding output-buffered switch, in the first scheme we assign every packet of a guaranteed rate flow a *targeted departure time* that is the departure time from the corresponding work conserving link with capacity equal to the guaranteed rate. The jitter control mechanism in front of the central buffer then uses the targeted departure time to regulate the traffic. By running the EDF policy with the targeted departure times as deadlines at the output buffer, we show that the end-to-end delay for every packet of a guaranteed rate flow is bounded by the sum of its targeted departure time and a constant that only depends on the number of flows and the size of the switch. The detailed architecture and its analysis for this scheme will be presented in Section II.

The second scheme is a much simpler one and has a framed structure as in Keslassy and McKeown [18]. There, time slots are grouped into fix size frames. Packets are placed in appropriate bins (buffers) according to their *arrival times* and their *flows*. We show that if the incoming traffic satisfies certain assumptions, then the end-to-end delay for every packet and the size of the central buffers are both bounded by constants that only depend on the size of the switches and the frame size. The second scheme is much simpler than the first one in many aspects: (i) the on-line complexity is $O(1)$ as there is no need for EDF, (ii) central buffers are finite and thus can be built into a single chip, (iii) connection patterns of the two switch fabrics are changed less frequently, (iv) there is no need for resequencing-and-output buffer after the second stage, and (v) variable length packets may be handled without segmentation and reassembly. The detailed architecture and its analysis will be shown in Section III.

II. AN EDF BASED SCHEME FOR GUARANTEED RATE SERVICES

In this section, we modify the scheme in the load balanced Birkhoff-von Neumann switch with multi-stage buffering in [8] so that guaranteed rate services can be provided for multicasting flows. The scheme has almost the same architecture as the FCFS scheme in Figure 3 (we will use Figure 3 for the analysis in this section). As in [8], we assume that packets are of the same size. Moreover, time is slotted and synchronized so that a packet can be transmitted within a time slot. We consider an $N \times N$ switch with multicasting flows. Packets from the same flow are distributed in the round-robin fashion to the second stage as described in [8]. As such, the delay through the first stage is bounded above by a constant in [8].

To provide guaranteed rate services, every packet of a (guaranteed rate) flow is assigned a *targeted departure time* that is the departure time from the corresponding FCFS work conserving link with capacity equal to the guaranteed rate of the flow. After leaving the first stage, a packet enters the jitter control stage in front of the central buffer. The time for a packet to leave the jitter control stage, called the *eligible time* of that packet, is set to be the sum of the targeted departure time and the maximum delay of the first stage. In the central buffer, packets are scheduled under the FCFS policy. We note that in implementation one may combine both the jitter control mechanism and the central buffer by using a single memory block. By time stamping every packet with its eligible time, the scheduling policy there is to schedule the *first eligible packet*. Another point is that best effort service can be provided as background traffic. Flows from best effort service can be assigned to a low priority queue and they are only served when there are no packets from guaranteed rate services in the central buffer.

After the second stage, packets are stored in the resequencing-and-output buffer as in the FCFS architecture. The key difference between this scheme and the FCFS scheme in Figure 3 is the scheduling policy at the output buffer. The scheduling policy in this guaranteed rate scheme is the EDF policy with packet deadlines being their targeted departure times. For this scheme, we will show that every packet departs from the switch not later than the sum of its targeted departure time and a constant that only depends on the size of the switch and the number of flows provided by the switch. Moreover, the size of the resequencing-and-output buffer can also be bounded by a constant that also depends on the size of the switch and the number of flows provided by the switch.

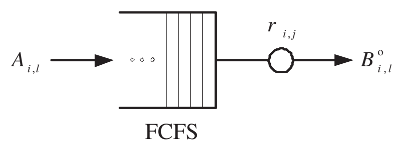

Fig. 5. The work conserving link corresponding to the $A_{i,\ell}$-flow

To be precise, let L_i be the number of multicasting flows through the i^{th} input port, $i = 1, 2, \ldots, N$. Denote by $A_{i,\ell}(t)$ the cumulative number of packet arrivals by time t from the

ℓ^{th} multicasting flow at the i^{th} input port, $i = 1, \dots, N$, $\ell = 1, \dots, L_i$. Let $r_{i,\ell}$ be the guaranteed rate of the $A_{i,\ell}$-flow. Now consider feeding the $A_{i,\ell}$-flow to a work conserving link with capacity $r_{i,\ell}$ (see Figure 5). Assume that the buffer in the work conserving link is infinite and empty at time 0. Let $B^o_{i,\ell}(t)$ be the cumulative number of departures at the output by time t. From [4], Lemma 1.3.1, one has the following well-known representation

$$B^o_{i,\ell}(t) = \min_{0 \le s \le t}[A_{i,\ell}(s) + r_{i,\ell}(t-s)]. \quad (1)$$

Let $d_{i,\ell}(k)$ be the targeted departure time of the k^{th} packet of the $A_{i,\ell}$-flow. Then it can be found by the following inverse mapping (see e.g., [4], Lemma 2.3.20)

$$\begin{aligned} d_{i,\ell}(k) &= \inf \Big[\tau : \tau \ge t \\ &\quad \text{and} \min_{0 \le u \le t-1}[A_{i,\ell}(u) + r_{i,\ell}(\tau - u)] \ge k\Big]. \end{aligned} \quad (2)$$

One key observation of the targeted departure times is that they are the outputs from a rate-controlled *traffic regulator*. Specifically, one can see from (1) that for all $s \le t$,

$$B^o_{i,\ell}(t) - B^o_{i,\ell}(s) \le r_{i,\ell}(t-s) \quad (3)$$

Such a property plays an important role in bounding delay in our scheme. We note that the technique of using targeted departure times to achieve rate guarantees has been studied extensively in the output-buffered switches (see e.g., [31], [13], [10], [4]).

Also, let $S^*(k)$ be the set of flows through the k^{th} output, and $M_k = |S^*(k)|$ be the number of multicasting flows through the k^{th} output port. Define $L_{\max} = \max_{1 \le i \le N} L_i$ as the maximum number of multicasting flow through an input port and $M_{\max} = \max_{1 \le k \le N} M_k$ as the maximum number of multicasting flow through an output port.

We present the main results of this scheme in the following theorem.

Theorem 1 *Suppose that all the buffers are empty at time 0. If*

$$\sum_{(i,\ell) \in S^*(k)} r_{i,\ell} \le 1, \quad (4)$$

for $k = 1, \cdots, N$, then

(i) *every packet of a guaranteed rate flow departs from the switch not later than the sum of its targeted departure time and $(N-1)L_{\max} + NM_{\max}$, and*

(ii) *the resequencing-and-output buffer at an output port of the second stage is bounded by $NM_{\max}$.*

The proof of Theorem 1 is based on a sequence of lemmas described in the following subsections.

A. Analysis for the central buffer

Now let $A^1_{i,\ell,j}(t)$ be the cumulative number of the $A_{i,\ell}$-flow packets that are split into the j^{th} VOQ at the i^{th} input port of the first stage by time t, and $D_{i,\ell,j}(t)$ be the number of the $A^1_{i,\ell,j}$-flow packets that have targeted departure times not greater than t. Without loss of generality, we assume that the first packet of a flow is always assigned to the first VOQ at the first stage. Since the targeted departure times are simply the departure times from the FCFS work conserving link with capacity $r_{i,\ell}$, we have

$$D_{i,\ell,j}(t) = \lceil \frac{B^o_{i,\ell}(t) - j + 1}{N} \rceil. \quad (5)$$

Moreover,

$$\sum_{j=1}^{N} D_{i,\ell,j}(t) = B^o_{i,\ell}(t). \quad (6)$$

Let $A^2_{i,\ell,j}(t)$ be the cumulative number of the $A_{i,\ell}$-flow packets at the j^{th} input port of the second stage by time t. Since the first stage is the same as that in [8], we know that the maximum delay at the first stage is bounded by

$$d_1 = (N-1)L_{\max}. \quad (7)$$

As discussed before, a jitter control stage is added in front of the VOQs in the second stage (see Figure 6) and the eligible time of a packet is set to be the sum of its targeted departure time and the maximum delay d_1. Thus, we have from (5) that

$$\begin{aligned} A^2_{i,\ell,j}(t) &= D_{i,\ell,j}(t - d_1) \\ &= \lceil \frac{B^o_{i,\ell}(t-d_1) - j + 1}{N} \rceil. \end{aligned} \quad (8)$$

Now consider the k^{th} VOQ at the j^{th} input port of the second stage (see Figure 6). Denote by $A^2_{j,k}(t)$ (resp. $B^2_{j,k}(t)$) the cumulative number of arrivals (resp. departures) at the k^{th} VOQ of the second stage by time t. Then

$$\begin{aligned} A^2_{j,k}(t) &= \sum_{(i,\ell) \in S^*(k)} A^2_{i,\ell,j}(t) \\ &= \sum_{(i,\ell) \in S^*(k)} \lceil \frac{B^o_{i,\ell}(t-d_1) - j + 1}{N} \rceil. \end{aligned} \quad (9)$$

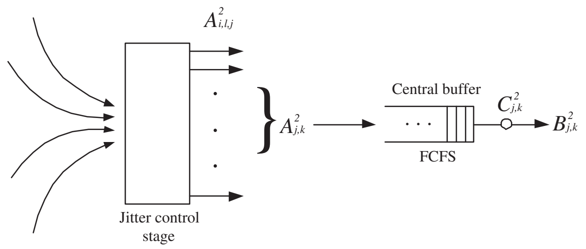

Fig. 6. The k^{th} VOQ at the j^{th} input port of the second stage

Let $q_{j,k}(t)$ be the number of packets queued at this queue at time t and $C^2_{j,k}(t)$ be the cumulative number of time slots assigned to this queue by time t. As shown in (17) of [8], we have

$$\begin{aligned} &q^2_{j,k}(t) \\ &= \max_{0 \le s \le t}[A^2_{j,k}(t) - A^2_{j,k}(s) - (C^2_{j,k}(t) - C^2_{j,k}(s))], \end{aligned} \quad (10)$$

and

$$B^2_{j,k}(t) = \min_{0\le s\le t}[A^2_{j,k}(s) + C^2_{j,k}(t) - C^2_{j,k}(s)]. \qquad (11)$$

Lemma 2 *Suppose the rate assumption in (4) holds.*

(i) *The maximum number of packets at the k^{th} VOQ of the j^{th} input of the second stage is bounded by M_k, i.e.,*

$$q^2_{j,k}(t) \le M_k. \qquad (12)$$

(ii) *Let*

$$d_2 = NM_k. \qquad (13)$$

The maximum delay of a packet at the k^{th} VOQ of the j^{th} input of the second stage is bounded by d_2, i.e.,

$$B^2_{j,k}(t+d_2) \ge A^2_{j,k}(t). \qquad (14)$$

A direct consequence of Lemma 2(ii) is that every packet leaves the k^{th} VOQ of the second stage not later than the sum of its targeted departure time and d_1+d_2. Therefore, we have from (9) and (14) that

$$B^2_{j,k}(t) \ge \sum_{(i,\ell)\in S^*(k)} D_{i,\ell,j}(t-d_1-d_2). \qquad (15)$$

For the proof of Lemma 2, we need to use the following well-known properties for the ceiling and floor functions.

Proposition 3 (i) $\lceil a+b\rceil \le \lceil a\rceil + \lceil b\rceil \le \lceil a+b\rceil + 1$.
(ii) $\lfloor a+b\rfloor \ge \lfloor a\rfloor + \lfloor b\rfloor$.
(iii) $\lceil a\rceil \le \lfloor a\rfloor + 1$.

Proof. (Lemma 2)

(i) Note from (9), Proposition 3(i), and (3) that

$$\begin{aligned} &A^2_{j,k}(t) - A^2_{j,k}(s) \\ &= \sum_{(i,\ell)\in S^*(k)} \lceil \frac{B^o_{i,\ell}(t-d_1)-j+1}{N}\rceil \\ &\qquad -\lceil \frac{B^o_{i,\ell}(s-d_1)-j+1}{N}\rceil \\ &\le \sum_{(i,\ell)\in S^*(k)} \lceil \frac{B^o_{i,\ell}(t-d_1)-B^o_{i,\ell}(s-d_1)}{N}\rceil \\ &\le \sum_{(i,\ell)\in S^*(k)} \lceil \frac{r_{i,\ell}(t-s)}{N}\rceil. \end{aligned} \qquad (16)$$

Since the connection patterns at the second stage are periodic with period N for some one-cycle permutation matrix,

$$C^2_{j,k}(t) - C^2_{j,k}(s) \ge \lfloor \frac{t-s}{N}\rfloor. \qquad (17)$$

From the assumption in (4) and Proposition 3(ii), it follows that

$$\begin{aligned} &C^2_{j,k}(t) - C^2_{j,k}(s) \\ &\ge \lfloor \frac{\sum_{(i,\ell)\in S^*(k)} r_{i,\ell}(t-s)}{N}\rfloor \\ &\ge \sum_{(i,\ell)\in S^*(k)} \lfloor \frac{r_{i,\ell}(t-s)}{N}\rfloor. \end{aligned} \qquad (18)$$

Using (16) and (18) in (10) yields

$$\begin{aligned} q^2_{j,k}(t) \quad &\le \quad \max_{0\le s\le t}\Big[\sum_{(i,\ell)\in S^*(k)} \lceil \frac{r_{i,\ell}(t-s)}{N}\rceil \\ &- \quad \lfloor \frac{r_{i,\ell}(t-s)}{N}\rfloor\Big]. \end{aligned} \qquad (19)$$

That (12) holds then follows from (19) and Proposition 3(iii).

(ii) It suffices to show that

$$B^2_{j,k}(t+d_2) - A^2_{j,k}(t) \ge 0.$$

Note from (11) that

$$\begin{aligned} &B^2_{j,k}(t+d_2) - A^2_{j,k}(t) \\ &= \min_{0\le s\le t+d_2}[A^2_{j,k}(s) - A^2_{j,k}(t) \\ &\qquad +C^2_{j,k}(t+d_2) - C^2_{j,k}(s)] \\ &= \min\Big[\min_{0\le s\le t}[A^2_{j,k}(s) - A^2_{j,k}(t) \\ &\qquad +C^2_{j,k}(t+d_2) - C^2_{j,k}(s)], \\ &\qquad \min_{t+1\le s\le t+d_2}[A^2_{j,k}(s) - A^2_{j,k}(t) \\ &\qquad +C^2_{j,k}(t+d_2) - C^2_{j,k}(s)]\Big]. \end{aligned} \qquad (20)$$

All the terms in the second minimum are clearly nonnegative as both $A^2_{j,k}(t)$ and $C^2_{j,k}(t)$ are non-decreasing in t. On the other hand, for $0\le s\le t$, we have from (17) and (4) that

$$\begin{aligned} C^2_{j,k}(t+d_2) - C^2_{j,k}(s) &\ge \lfloor \frac{t-s+d_2}{N}\rfloor \\ &\ge \lfloor \frac{(\sum_{(i,\ell)\in S^*(k)} r_{i,\ell}(t-s)) + NM_k}{N}\rfloor \\ &= \lfloor \frac{\sum_{(i,\ell)\in S^*(k)}(r_{i,\ell}(t-s)+N)}{N}\rfloor. \end{aligned}$$

Using (16) and Proposition 3 (ii),(iii) yields

$$\begin{aligned} &C^2_{j,k}(t+d_2) - C^2_{j,k}(s) \\ &\quad -(A^2_{j,k}(t) - A^2_{j,k}(s)) \\ &\ge \sum_{(i,\ell)\in S^*(k)} \lfloor \frac{r_{i,\ell}(t-s)+N}{N}\rfloor \\ &\quad - \sum_{(i,\ell)\in S^*(k)} \lceil \frac{r_{i,\ell}(t-s)}{N}\rceil \\ &= \sum_{(i,\ell)\in S^*(k)} (\lfloor \frac{r_{i,\ell}(t-s)}{N}\rfloor + 1) \\ &\quad - \sum_{(i,\ell)\in S^*(k)} \lceil \frac{r_{i,\ell}(t-s)}{N}\rceil \\ &\ge 0. \end{aligned}$$

■

B. Analysis for the resequencing-and-output buffer

In this section, we analyze the resequencing-and-output buffer. The resequencing-and-output buffer conceptually consists of two virtual buffers (see Figure 7): (i) the resequencing buffer and (ii) the output buffer. The objective of the resequencing buffer is to reorder the packets so that packets of the same flow depart in the same order as they arrive. After resequencing, packets are stored in the output buffer waiting for transmission from the output link. The scheduling policy at the output buffer is the EDF policy with the deadline of a packet being its targeted departure time.

Let $A^3_k(t)$ be the cumulative arrivals by time t to the k^{th} resequencing buffer, and $B^3_k(t)$ be its cumulative departures. Note that $B^3_k(t)$ is also the cumulative arrivals by time t to the k^{th} output buffer. Denote by $B^3_{i,\ell}(t)$ the cumulative arrivals of the $A_{i,\ell}$-flow by the time t to the output buffer. Thus, we have

$$A^3_k(t) = \sum_{j=1}^{N} B^2_{j,k}(t), \tag{21}$$

and

$$B^3_k(t) = \sum_{(i,\ell)\in S^*(k)} B^3_{i,\ell}(t). \tag{22}$$

Let $B^4_k(t)$ be the cumulative departures by time t from the k^{th} output buffer.

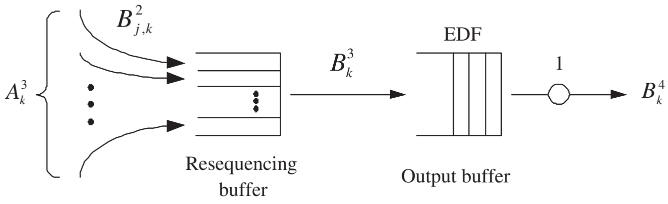

Fig. 7. The resequencing-and-output buffer

Lemma 4 *Suppose the rate assumption in (4) holds.*

(i) $A^3_k(t) \le \sum_{(i,\ell)\in S^*(k)} B^o_{i,\ell}(t-d_1)$.

(ii) *Every packet of the $A_{i,\ell}$-flow leaves the k^{th} resequencing buffer not later than the sum of its targeted departure time and d_1+d_2, i.e.,*

$$B^3_{i,\ell}(t) \ge B^o_{i,\ell}(t-d_1-d_2).$$

(iii) *Under the EDF scheduling policy, every packet leaves the output buffer not later than the sum of its targeted departure time and d_1+d_2.*

(iv) *The number of packets queued at the k^{th} resequencing-and-output buffer is bounded by d_2, i.e.,*

$$A^3_k(t) - B^4_k(t) \le d_2.$$

Proof. (i) Recall that $A^2_{j,k}$ is the cumulative number of departures from the jitter control stage to the k^{th} VOQ of the j^{th} input of the second stage by time t. From (21),(9) and (6), it follows that

$$\begin{aligned} A^3_k(t) &= \sum_{j=1}^{N} B^2_{j,k}(t) \le \sum_{j=1}^{N} A^2_{j,k}(t) \\ &= \sum_{j=1}^{N} \sum_{(i,\ell)\in S^*(k)} D_{i,\ell,j}(t-d_1) \\ &= \sum_{(i,\ell)\in S^*(k)} \sum_{j=1}^{N} D_{i,\ell,j}(t-d_1) \\ &= \sum_{(i,\ell)\in S^*(k)} B^o_{i,\ell}(t-d_1). \end{aligned}$$

(ii) From Lemma 2(ii), it follows that the departure time for a packet of the $A_{i,\ell}$-flow to leave the resequencing buffer is not later than the sum of its targeted departure time and d_1+d_2.

(iii) According to Theorem 5.6.1 in [4], it suffices to show that

$$\begin{aligned} &\sum_{(i,\ell)\in S(k)} B^o_{i,\ell}(t-d_1-d_2) \\ &\le \min_{0\le s\le t}\Big[\sum_{(i,\ell)\in S(k)} B^3_{i,\ell}(s) + (t-s)\Big], \end{aligned} \tag{23}$$

for all $S(k) \subseteq S^*(k)$.

From (ii) of this lemma, (3) and (4), we have for all $S(k) \subseteq S^*(k)$ and $0 \le s \le t$,

$$\begin{aligned} &\sum_{(i,\ell)\in S(k)} B^o_{i,\ell}(t-d_1-d_2) - B^3_{i,\ell}(s) \\ &\le \sum_{(i,\ell)\in S(k)} B^o_{i,\ell}(t-d_1-d_2) \\ &\qquad -B^o_{i,\ell}(s-d_1-d_2) \\ &\le \sum_{(i,\ell)\in S(k)} r_{i,\ell}(t-s) \\ &\le \sum_{(i,\ell)\in S^*(k)} r_{i,\ell}(t-s) \le (t-s). \end{aligned} \tag{24}$$

Thus, the inequalities in (23) hold.

(iv) Since every packet leaves the system not later than the sum of its targeted departure time and d_1+d_2, we then have

$$B^4_k(t) \ge \sum_{(i,\ell)\in S^*(k)} B^o_{i,\ell}(t-d_1-d_2).$$

From (i), (iii) of this Lemma, (3) and (4), it follows that

$$\begin{aligned} &A^3_k(t) - B^4_k(t) \\ &\le \sum_{(i,\ell)\in S^*(k)} B^o_{i,\ell}(t-d_1) - B^o_{i,\ell}(t-d_1-d_2) \\ &\le \sum_{(i,\ell)\in S^*(k)} r_{i,\ell} d_2 \le d_2. \end{aligned}$$

■

Proof. (Proof of Theorem 1) (i) It is shown in Lemma 4 (iii). (ii) It is shown in Lemma 4 (iv). ■

III. A FRAME BASED SCHEME FOR GUARANTEED RATE SERVICES

The drawback of the previous scheme is its hardware implementation complexity for the resequencing-and-output buffer and the jitter control mechanism. Moreover, only fixed size packets are considered. In order to provide guaranteed rate services for variable length packets, variable length packets have to be segmented into fixed size packets, transmitted through the switch and, re-assembled at the output. The objective of this section is to propose a simple scheme that does not require resequencing, EDF scheduling and jitter control. Furthermore, variable length packets may not need to be segmented.

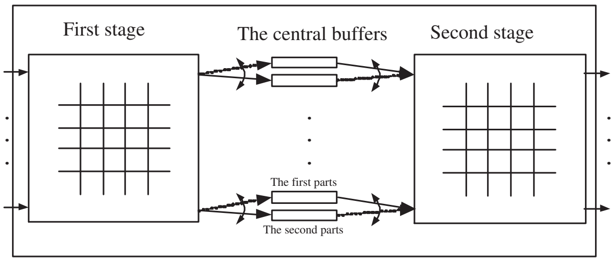

Fig. 8. The architecture for the frame based scheme

The idea of the second scheme, as in Keslassy and McKeown [18], is to use a framed structure so that resequencing is not needed. The architecture of the scheme is shown in Figure 8. To ease our presentation, we shall describe the scheme for fixed size packets and point-to-point flows. Extensions to variable length packets and multicasting flows will be addressed at the end of this section. As in the load balanced Birkhoff-von Neumann switches, there are two $N \times N$ crossbar switch fabrics and buffers between these two crossbar switch fabrics. In this scheme, time slots are grouped into fixed size frames. Each frame has F time slots. Thus, the m^{th} time frame is from time slot $(m-1)F+1$ to time slot mF (see Figure 9).

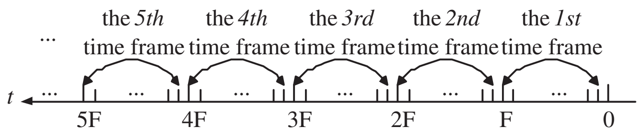

Fig. 9. The time frame structure

Let $A_{i,k}(t)$ be the cumulative number of (fixed size) packet arrivals by time t at the i^{th} input port to the k^{th} output port, $i = 1, \ldots, N$, $k = 1, \ldots, N$. Let $r_{i,k}$ be the guaranteed rate of the $A_{i,k}$-flow. Assume that F is chosen so that $M_{i,k} = r_{i,k}F$ is an integer for $i = 1, \ldots, N$, $k = 1, \ldots, N$. We will show that the switch architecture in Figure 8 provides guaranteed rate services under the following assumptions.

(A1) In a time slot, no more than one (fixed size) packet arrives at an input port of the switch fabric.

(A2) No more than $M_{i,k}$ (fixed size) packets of the $A_{i,k}$-flow arrive at the i^{th} input port in a time frame.

(A3) $\sum_{i=1}^{N} r_{i,k} \le 1$, for $k = 1, 2, \cdots, N$.

(A4) All the buffers are empty at the beginning

Note that (A2) implies that $\sum_{k=1}^{N} r_{i,k} \le 1$, for $i = 1, 2, \cdots, N$. These inequalities and those in (A3) are known as the "no overbooking" conditions in [14], as they simply state that neither the total rate coming out from an input port nor the total rate to an output port can be larger than 1.

First, we describe how the connection patterns of the two crossbar switch fabrics are set up. Unlike the last section, both switches now change their connection patterns according to time frames. In a time frame, both crossbar switches in Figure 8 set up connection patterns corresponding to a circular-shift matrix (note that a circular-shift matrix is also a one-cycle permutation matrix). Specifically, if the j^{th} output port is connected to the i^{th} input port during the m^{th} time frame, then the j^{th} output port will be connected to the $(i+1)^{th}$ input port during the $(m+1)^{th}$ time frame, for $i = 1, 2, \cdots, N-1$. If the j^{th} output port is connected to the N^{th} input port during the m^{th} time frame, then the j^{th} output port will be connected to the 1^{st} input port during the $(m+1)^{th}$ time frame. Initially, we set the connection patterns so that the j^{th} output port is connected to the 1^{st} input port during the j^{th} time frame. To be precise, we define the function $h(i, m) = (m - i + 1) \bmod N$ if $(m - i + 1) \bmod N \neq 0$ and $h(i, m) = N$ otherwise. During the m^{th} time frame, the i^{th} input port is connected to the $h(i, m)^{th}$ output port of these two crossbar switch fabrics. As such, all the packets that arrive at the i^{th} input port during the m^{th} frame are all routed to the $h(i, m)^{th}$ output port.

There are N central buffers between these two switch fabrics, indexed from 1 to N. Each central buffer consists of two alternating memory blocks. The buffer size of each memory block is NF, which is divided into N bins, each with buffer size of F. To ease the presentation for the operation of these central buffers, we introduce the concept of superframes. The p^{th} superframe of the i^{th} input port of the both stages is defined to be the set of time slots in the N time frames, starting from the $((p-1)N+i)^{th}$ frame to the $(pN+i-1)^{th}$ frame. Note that the p^{th} superframe of the i_1^{th} input and the p^{th} superframe of the i_2^{th} input are different if $i_1 \neq i_2$. Moreover,the j^{th} time frame in the p^{th} superframe of the i^{th} input port (of both stages) is the $((p-1)N + i + j - 1)^{th}$ frame. Since

$$h(i, (p-1)N + i + j - 1) = j,$$

it follows that *during the j^{th} time frame in the p^{th} superframe of the i^{th} input port, the i^{th} input port is always connected to the j^{th} output port.*

Consider a particular packet that arrives at the i^{th} input port of the first stage during the j^{th} time frame in the p^{th} superframe of the i^{th} input port. As just described, the i^{th} input is connected to the j^{th} output during that frame and the packet is thus sent to the j^{th} central buffer without delay. As there are two alternating memory blocks in the j^{th} central buffer, the packet is sent to the second (resp. first) memory block if p is odd (resp. even). If, furthermore, the packet is destined for the k^{th} output port, it will be placed in the k^{th} bin of that memory block. As each bin only has the buffer size F, one might wonder whether there is enough buffer space for such an assignment. We will show in Theorem 6 that under the assumptions in (A1-4) there are no packet overflows for such an assignment.

Without loss of generality, let us assume that p is *odd* and the packet is placed in the k^{th} bin of the *second* memory block of the j^{th} central buffer. During the k^{th} time frame in the $(p+1)^{th}$ superframe of the j^{th} input port of the second stage, the j^{th} input port of the second stage is connected to the k^{th} output of the second stage. As each frame has F times slots and each bin can hold at most F packets, during that frame all the packets in the k^{th} bin of the second memory block of the j^{th} central buffer are transmitted to the k^{th} output of the second stage.

Example 5 We illustrate this scheme by a 4×4 switch fabric. In Figure 10, we show the operation for the first stage. We denote by $I(i,m)$ the set of packets that arrive at the i^{th} input port of the first stage during the m^{th} time frame, and $I_s(i,p)$ the set of packets that arrive the i^{th} input port of the first stage during the p^{th} superframe of the i^{th} input port. Note that $I(1,1)$, $I(2,2)$, $I(3,3)$ and $I(4,4)$ are all routed to the second memory block of the *first* central buffer. Each of the four frames is the *first* frame in the superframe of its input. Upon the arrival of each packet in these four frames, it is placed immediately in the bin that corresponds to its destined output. At the end of the first superframe of the first input (i.e., the end of the 4^{th} frame), all the packets in the bins of the second memory block of the first central buffer are well packed and ready to be transmitted to the second stage. Similarly, $I(1,2)$, $I(2,3)$, $I(3,4)$ and $I(4,5)$ are all routed to the second memory block of the *second* central buffer, $I(1,3)$, $I(2,4)$, $I(3,5)$ and $I(4,6)$ are all routed to the second memory block of the *third* central buffer, and $I(1,4)$, $I(2,5)$, $I(3,6)$ and $I(4,7)$ are all routed to the second memory block of the *fourth* central buffer.

In Figure 11, we illustrate the operation for the second stage. We denote by $O(j,m)$ the set of packets that depart from the j^{th} *input* port of the second stage during the m^{th} time frame, and $O_s(j,p)$ the set of packets that depart from the j^{th} *input* port of the second stage during the p^{th} superframe of the j^{th} input port. Now consider the four bins at the second memory block of the first central buffer. Since they are ready at the end of the first superframe of the first input, packets in the *first* bin are routed to the *first* output during the *first* frame of the second superframe of the first input, i.e., the 5^{th} frame. Similarly, packets in the *second* bin are routed to the *second* output during the *second* frame of the second superframe of the first input, i.e., the 6^{th} frame, packets in the *third* bin are routed to the *third* output during the *third* frame of the second superframe of the first input, i.e., the 7^{th} frame, and packets in the *fourth* bin are routed to the *fourth* output during the *fourth* frame of the second superframe of the first input, i.e., the 8^{th} frame. In other words, $O(1,5)$ contains the packets in the first bin, $O(1,6)$ contains the packets in the second bin, $O(1,7)$ contains the packets in the third bin, and $O(1,8)$ contains the packets in the fourth bin.

The four bins in the second memory block of the *second* central buffer are ready at the end of the 5^{th} frame. These four bins are routed to the first output during the 6^{th} frame, the second output during the 7^{th} frame, the third output during the 8^{th} frame and the fourth output during the 9^{th} frame. The operation for the other two central buffers are done in a similar manner as shown in Figure 11.

Theorem 6 *Assume that (A1-4) hold. A packet that arrives at the i^{th} input and destined to the k^{th} output during the j^{th} time frame in the p^{th} superframe of the i^{th} input of the first stage (i.e., the $((p-1)N+i+j-1)^{th}$ time frame) will depart during the k^{th} time frame in the $(p+1)^{th}$ superframe of the j^{th} input of the second stage (i.e., the $(pN+j+k-1)^{th}$ time frame), for $i = 1, 2, \cdots, N$ and $k = 1, 2, \cdots, N$.*

There are several consequences of Theorem 6.

(i) Even though the central buffer is finite, no packets are lost inside the switch.

(ii) Packets of the same flow (the same i and k) depart in the FCFS order. This is trivial for packets of the same flow that arrive within the same frame. For packets of the same flow that arrive in different frames, one can see from Theorem 6 that the departure time of a packet is increasing in both j and p.

(iii) From Theorem 6, the maximum delay for all arrivals from the i^{th} input port to the k^{th} output port through the switch fabric is bounded by

$$\begin{aligned} &(pN+j+k-1)F \\ &\quad -((p-1)N+i+j-1)F+F \\ &= (N+k-i+1)F. \end{aligned} \tag{25}$$

Thus, the maximum delay for all arrivals from the i^{th} input port through the switch fabric is bounded by $(2N-i+1)F$, which in turn is bounded above by $2NF$.

Proof. (Theorem 6)

From (A2), the number of packets of the $A_{i,k}$-flow that arrive during the j^{th} time frame in the p^{th} superframe of the i^{th} input port of the first stage(i.e., the $((p-1)N+i+j-1)^{th}$ time frame) is bounded by $M_{i,k}$. Without loss of generality, assume that p is odd. The total number of packets that are placed in the k^{th} bin of the second memory block of the j^{th} central buffer during the p^{th} superframe of the j^{th} input port of the second stage is not greater than

$$\sum_{i=1}^{N} M_{i,k}.$$

From (A3), it follows that

$$\sum_{i=1}^{N} M_{i,k} = \sum_{i=1}^{N} r_{i,k} F \leq F.$$

Thus, if the k^{th} bin of the second memory block of the j^{th} buffer is empty at the beginning of the p^{th} superframe of the j^{th} input port of the second stage, then all of the packets that arrive during this superframe can be placed in that bin without causing buffer overflow. During the k^{th} time frame in the $(p+1)^{th}$ superframe of the j^{th} input port of the second stage (i.e., the $(pN+j+k-1)^{th}$ time frame), all of packets in that bin are routed to the k^{th} output port of the second stage. As a result, the k^{th} bin of the second memory block of the j^{th} buffer is *empty* again at the beginning of the $(p+2)^{th}$ superframe of the j^{th} input port of the second stage! By induction, all packets of the $A_{i,k}$-flow in the j^{th} time frame of the p^{th} superframe of the i^{th} input port of the first stage (i.e., the $((p-1)N+i+j-1)^{th}$ time frame) will depart during the k^{th} time frame in the $(p+1)^{th}$ superframe of the j^{th} input of the second stage (i.e., the $(pN+j+k-1)^{th}$ time frame), for $k = 1, 2, \cdots, N$ and $i = 1, 2, \cdots, N$.

The argument for the case that p is even is similar. ■

Now we describe how we extend the scheme for variable length packets. As there is a limit on the number of packets that can be transmitted within a time frame for a flow, buffers have to be provided at the input ports. Thus, one can use the VOQ technique for input buffers as shown in Figure 1. Specifically, packets from the $A_{i,k}$-flow are queued at the k^{th} VOQ of the i^{th} input. In every time frame, one can now assign *consecutive* $M_{i,k}$ time slots for the $A_{i,k}$-flow at the i^{th} input. As such, variable length packets (with packet length smaller than the quota $M_{i,k}$) can be transmitted without segmentation and reassembly.

It is also possible to support the multicasting flows considered in Section II. Now the no overbooking conditions are

$$\sum_{\ell=1}^{L_i} r_{i,\ell} \leq 1, \quad i = 1, 2, \ldots, N, \quad \text{and}$$
$$\sum_{(i,\ell) \in S*(k)} r_{i,\ell} \leq 1, \quad k = 1, 2, \ldots, N.$$

Moreover, fan-out splitting needs to be carried out at the central buffers. This implies that a packet needs to be placed in multiple bins at the same time. As such, the implementation that use pointers to the memory addresses of packets might be better than duplicating multiple packets directly.

IV. Conclusion

In this paper, we proposed two schemes for the load balanced Birkhoff-von Neumann switches to provide guaranteed rate services. The first scheme is an EDF based scheme. We assign every packet a targeted departure time that is the departure time from the corresponding work conserving link with capacity equal to the guaranteed rate. By adding a jitter control mechanism in front of the buffer at the second stage and running the EDF at the output buffer, we showed that the end-to-end delay for every packet of a flow is bounded by the sum of its targeted departure time and a constant that only depends on the number of flows and the size of the switch. In comparison with the scheme for guaranteed rate services in [5] and [6], this new scheme has the following advantages:

(i) There is no need to perform the Birkhoff-von Neumann decomposition in [5] and [6].
(ii) One only needs to implement N connection patterns for each crossbar switch and these connection patterns are independent of the incoming traffic.
(iii) This scheme can support multicasting flows.

The main drawback of this scheme is the hardware complexity of implementing the jitter control mechanism and the EDF scheduling policy at the output buffer.

Our second scheme is much simpler than the first one. There, time slots group into fix size frames. We showed that if the incoming traffic satisfied assumptions in $(A1)-(A4)$, then the end-to-end delay for every packet and the size of central buffers are both bounded by constants that only depend on the size of the switch and the frame size. The second scheme has the following advantages:

(i) The on-line complexity is $O(1)$.
(ii) We still only need N connection patterns for each crossbar switch.
(iii) Central buffers are finite and thus can be built into a single chip.
(iv) Since each crossbar switch changes its connection pattern according to time frames, the frequency of changing connection patterns for each switch in the second scheme is much slower than the frequency in the first scheme. This is a good aspect for an optical switch, since the frequency of changing connection patterns in an optical switch is constrained by its slow mechanical characteristic.
(v) Since all the packets from the same flow leave the switch fabric in the FCFS order, there is no need for the resequencing-and-output buffer after the second stage.
(vi) This scheme may be able to handle variable length packets without segmentation and reassembly.

To summarize, in Table I we compare various switch architectures, including the ideal output-buffered switch (OQ), the input-buffered switch with maximal matching (IQ(MM)) [11], the input-buffered switch with maximum weighted matching (IQ(MWM)) [23], the combined input-output queueing switch (CIOQ) [9], [28], the Birkhoff-von Neumann switch (BvN) [5], [6], the load balanced Birkhoff-von Neumann switch with one-stage buffering (LBvN(I)) [7], the load balanced Birkhoff-von Neumann switch with multi-stage buffering (LBvN(II)) [8], the EDF based scheme in this paper, and the frame based scheme in this paper.

Architecture	OQ	IQ(MM)	IQ(MWM)	CIOQ	BvN	LBvN(I)	LBvN(II)	EDF	Frame
Speedup	N	1	1	2	1	1	1	1	1
Throughput	100%	$\geq$50%	100%	100%	100%	100%	100%	100%	100%
On-line complexity for crossbar connections	N.A.	$O(N)$	$O(N^3 \log N)$	$O(N^2)$	$O(\log N)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$
Rate information needed	No	No	No	No	Yes	No	No	Yes	Yes
Rate Guarantee	Yes	No	No	Yes	Yes	No	No	Yes	Yes
Packet order preserved	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
Mulitcast	100%	No	No	No	No	100%	100%	100%	100%
Variable length packet	Yes	No	No	No	No	No	No	No	Yes
Delay with respect to OQ	N.A.	No	No	Exact	No	No	Bound	Bound	Bound

TABLE I

COMPARION OF VARIOUS SWITCH ARCHITECTURES.

REFERENCES

[1] M. Ajmone Marsan, A. Bianco, P. Giaccone, E. Leonardi and F. Neri, "On the throughput of input-queued cell-based switches with multicast traffic," *Proc. IEEE INFOCOM'01*, pp. 1664-1672, 2001.

[2] T. Anderson, S. Owicki, J. Saxes and C. Thacker, "High speed switch scheduling for local area networks," *ACM Trans. on Computer Systems*, Vol. 11, pp. 319-352, 1993.

[3] G. Birkhoff, "Tres observaciones sobre el algebra lineal," *Univ. Nac. Tucumán Rev. Ser. A*, Vol. 5, pp. 147-151, 1946.

[4] C.S. Chang. *Performance Guarantees in Communication Networks*. London: Springer-Verlag, 2000.

[5] C.S. Chang, W.J. Chen and H.Y. Huang, "On service guarantees for input buffered crossbar switches: a capacity decomposition approach by Birkhoff and von Neumann," *IEEE IWQoS'99*, pp. 79-86, London, U.K., 1999.

[6] C.S. Chang, W.J. Chen and H.Y. Huang, "Birkhoff-von Neumann input buffered crossbar switches," *IEEE INFOCOM2000*, pp. 1614-1623, Tel Aviv, Israel, 2000.

[7] C.S Chang, D.S. Lee and Y.S. Jou, "Load balanced Birkhoff-von Neumann switches, part I: one-stage buffering," *Computer Communications*, Vol. 25, pp. 611-622, 2002.

[8] C.S. Chang, D.S. Lee and C.M. Lien, "Load balanced Birkhoff-von Neumann switch, part II: Multi-stage buffering," *Computer Communications*, Vol. 25, pp. 623-634, 2002.

[9] S.-T. Chuang, A. Goel, N. McKeown and B. Prabhkar, "Matching output queueing with a combined input output queued switch," *IEEE INFOCOM'99*, pp. 1169-1178, New York, 1999.

[10] R.L. Cruz, " SCED+: efficient management of quality of service guarantees," *Proc. of IEEE INFOCOM'98*.

[11] J. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," *Proceedings of IEEE INFOCOM'00*, pp. 556-564, Tel Aviv, Isreal, March, 2000.

[12] A. Demers, S. Keshav, and S. Shenkar, "Analysis and simulation of a fair queueing algorithm," in *Proc. SIGCOMM'89*, pp. 1-12, Austin, TX, Sept. 1989.

[13] P. Goyal and H.M. Vin, "Generalized guaranteed rate scheduling algortihms: a framework," *IEEE/ACM Transactions on Networking*, Vol. 5, pp. 561-571, 1997.

[14] A. Hung, G. Kesidis and N. McKeown, "ATM input-buffered switches with guaranteed-rate property," *Proc. IEEE ISCC'98*, Athens, pp. 331-335, 1998.

[15] S. Iyer, A. Awadallah and N. McKeown, "Analysis of a packet switch with memories running at slower than line speed", *Proceedings of IEEE INFOCOM 2000*.

[16] S. Iyer and N. McKeown, "Making parallel packet switch practical," *Proc. IEEE INFOCOM 2001*, Anchorage, Alaska, U.S.A.

[17] M. J. Karol, M. G. Hluchyj, and S. P. Morgan, "Input Versus Output Queueing on a Space-Division Packet Switch," *IEEE Trans. Commun.*, Vol. COM35, NO.12, Dec. 1987.

[18] I. Keslassy and N. McKeown, "Maintaining packet order in two-stage switches," *Proc. of IEEE INFOCOM*, New York, 2002.

[19] T.T. Lee and C.H. Lam, "Path switching-a quasi-static routing scheme for large scale ATM packet switches," *IEEE Journal on Selected Areas of Communications*, Vol. 15, pp. 914-924, 1997.

[20] S. Li and N. Ansari, "Input-queued switching with QoS guarantees," *IEEE INFOCOM'99*, pp. 1152-1159, New York, 1999.

[21] Y. Li, S. Panwar and H.J. Chao, "On the performance of a dual round-robin switch," *Proc. IEEE INFOCOM'01*, pp. 1688-1697, 2001.

[22] N. McKeown, "Scheduling algorithms for input-queued cell switches," *PhD Thesis. University of California at Berkeley*, 1995.

[23] N. McKeown, V. Anantharam and J. Walrand, "Achieving 100% throughput in an input-queued switch," *Proc. IEEE INFOCOM'96*, pp. 296-302, 1996.

[24] A. Mekkittikul and N. McKeown, "A practical scheduling algorithm to achieve 100% throughput in input-queued switches," *Proc. IEEE INFOCOM'98*.

[25] D. Mitra and R.A. Cieslak, "Randomized parallel communications on an extension of the omega network," *Journal of the Association for Computing Machinery*, Vol. 34, No. 4, pp. 802-824, 1987.

[26] A.K. Parekh and R.G. Gallager, "A generalized processor sharing approach to flow control in integrated service networks: the single-node case," *IEEE/ACM Transactions on Networking*, Vol. 1, pp. 344-357, 1993.

[27] D. Stiliadis and A. Varma, "Providing bandwidth guarantees in an input-buffered crossbar switch," *Proc. IEEE INFOCOM'95*, pp. 960-968, 1995.

[28] I. Stoica and H. Zhang, "Exact emulation of an output queueing switch by a combined input output queueing switch," *IEEE IWQoS'98*, pp. 218-224, Napa, California, 1998.

[29] L. G. Valiant, "A scheme for fast parallel communication," *SIAM J. Comput.*, Vol. 11, No. 2, pp. 350-361, 1982.

[30] J. von Neumann, "A certain zero-sum two-person game equivalent to the optimal assignment problem, " *Contributions to the Theory of Games,* Vol. 2, pp. 5-12, Princeton University Press, Princeton, New Jersey, 1953.

[31] L. Zhang, "Virtualclock: a new traffic control algorithm for packet switching networks," *ACM Transactions on Computer Systems*, Vol. 9, pp. 101-124, 1991.

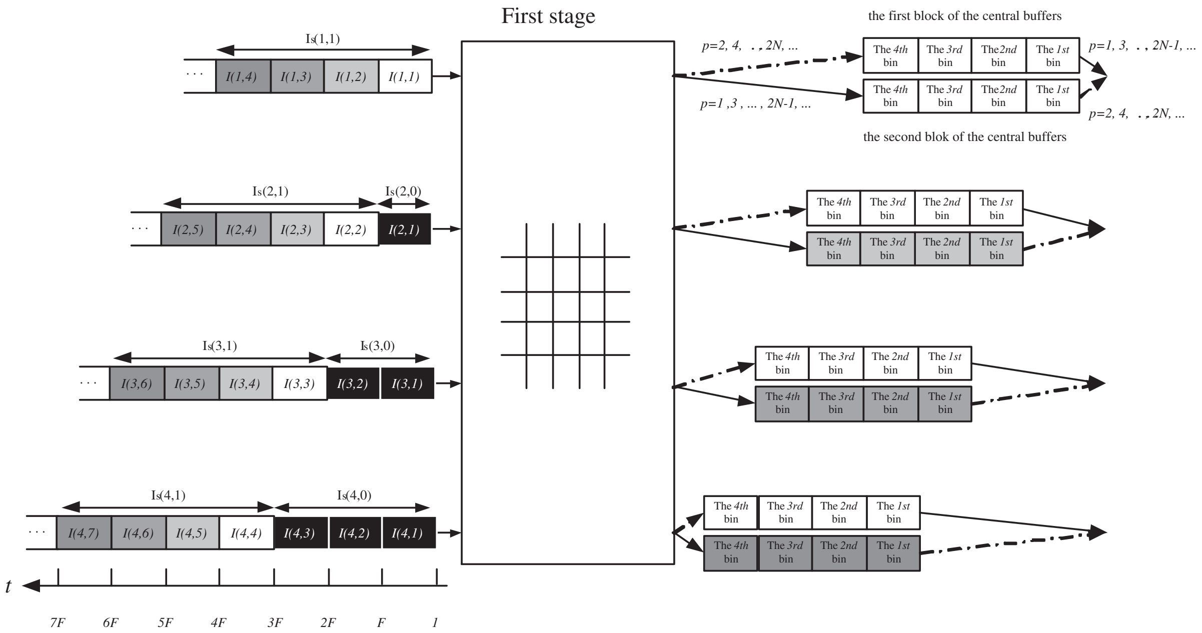

Fig. 10. The first stage of a 4×4 switch fabric

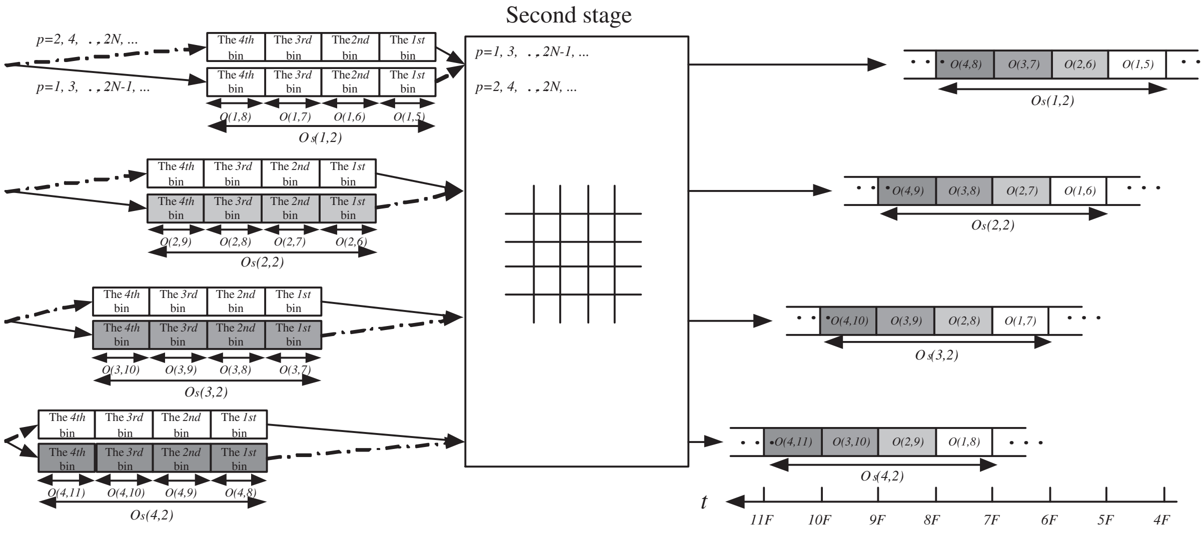

Fig. 11. The second stage of a 4×4 switch fabric