CR Switch: A Load-Balanced Switch with Contention and Reservation

Chao-Lin Yu, *Student Member, IEEE,* Cheng-Shang Chang, *Fellow, IEEE,* and Duan-Shin Lee, *Senior Member, IEEE,*
Institute of Communications Engineering
National Tsing Hua University
Hsinchu 300, Taiwan, R.O.C.
Email:clyu@gibbs.ee.nthu.edu.tw
cschang@ee.nthu.edu.tw
lds@cs.nthu.edu.tw

***Abstract*—Load-balanced switches have received a great deal of attention recently as they are much more scalable than other existing switch architectures in the literature. However, as there exist multiple paths for flows of packets to traverse through load-balanced switches, packets in such switches may be delivered out of order. In this paper, we propose a new switch architecture, called the CR switch, that not only delivers packets in order but also guarantees 100% throughput. The key idea, as in a multiple access channel, is to operate the CR switch in two modes: (i) the contention mode in light traffic and (ii) the reservation mode in heavy traffic. To do this, we invent a new buffer management scheme, called I-VOQ (virtual output queue with insertion). With the I-VOQ scheme, we give rigorous mathematical proofs for 100% throughput and in order packet delivery of the CR switch. By computer simulations, we also demonstrate that the average packet delay of the CR switch is considerably lower than other schemes in the literature, including the uniform frame spreading scheme [10], the padded frame scheme [8] and the mailbox switch [5].**

***Index Terms*—load-balanced switches, contention, reservation, I-VOQ, delay performance.**

I. Introduction

Load-balanced switches (see e.g., [3], [5], [6], [8], [10], [11]) have received a great deal of attention recently as they are much more scalable than other existing switch architectures in the literature. A typical load-balanced switch (see Figure 1) consists of two stages: the first stage is for load-balancing that converts incoming traffic into the uniform traffic, and the second stage is for switching of the uniform traffic. The connection patterns in the switches of both stages are *deterministic* and *periodic*. As such, there is no need to find matchings as required in most input-buffered switches.

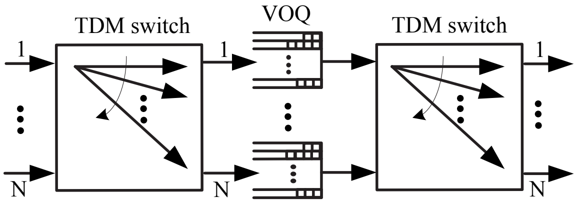


Fig. 1. The generic load-balanced switch architecture

The problem of load-balanced switches is that there are multiple paths between each input/output pair. As such, packets of the same flow may be delivered out of sequence. To cope with this problem, there are several tentative solutions proposed in the literature. Among them, the Uniform Frame Spreading (UFS) scheme [10] is the most simple one. The idea of the UFS scheme is to add virtual output queues (VOQ) at the inputs of the whole switch and operate the system in frames. Packets destined for the same output are stored in the same VOQ. Once a VOQ has more packets than the number of input/output ports, that VOQ is called a full-framed VOQ. At the beginning of a frame, a full-framed VOQ is selected and transmitted to the second stage. If there is no full-framed VOQ, then nothing is transmitted. By so doing, a full-framed VOQ "reserves" a frame (of time slots) and transmits its packets *consecutively* in that frame. Though the UFS scheme is shown to achieve 100% throughput [10], the packet delay is large (even in light traffic). This is known as the starvation problem as it takes time to accumulate packets for a full-framed VOQ.

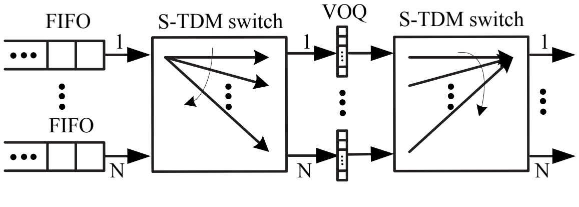


Fig. 2. The architecture of the mailbox switch (with $\delta = 0$ in [5])

On the other hand, the mailbox switch (with $\delta = 0$ in [5]) has only one buffer (for storing a packet) between two stages (see Figure 2). Packets have to contend for that buffer and packets might be rejected in the central buffer due to contention. To obtain the information about whether a transmission is successful or not, the mailbox switch utilizes the symmetric TDM (S-TDM) switch to provide a feedback

This research was supported in part by the National Science Council, Taiwan, R.O.C., under Contract NSC-93-2213-E-007-040, Contract NSC-93-2213-E-007-095, Contract NSC-94-2213-E-007-046, and the Program for Promoting Academic Excellence of Universities NSC 94-2752-E-007-002-PAE. Part of this paper was presented in the IEEE International Conference on Computer Communications (INFOCOM'07), Anchorage, AK, USA, May 6–12, 2007.

path. As there is only one buffer, packets from the same flow are delivered in order. However, as packets have to contend for that buffer, 100% throughput cannot be achieved. In fact, it was shown in [5], the throughput for such a switch is only 58%. The advantage of the mailbox switch is its low packet delay in light traffic. In light traffic, collisions seldom occur and packets can be transmitted immediately after their arrivals.

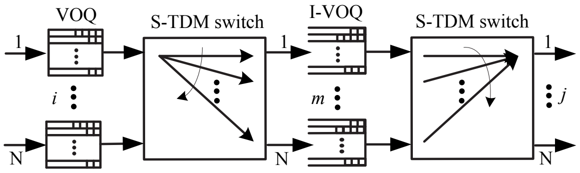


Fig. 3. The architecture of the CR switch

The main contribution of our work is to propose a switch architecture, called the CR switch (see Figure 3), that can have the advantages of both the UFS scheme in [10] and the mailbox switch in [5]. We show that the CR switch achieves 100% throughput and delivers packets in order (as in the UFS scheme), while maintaining low packet delay in light traffic (as in the mailbox switch). The main idea, as pointed out in the pioneer work by Tobagi and Kleinrock [16] for a multiple access channel, is to have the CR switch operating in two modes: the contention mode (in light traffic) and the reservation mode (in heavy traffic). As in the UFS scheme, when there is a full-framed VOQ, the CR switch operates in the reservation mode and transmits a full frame of packets. However, when there is no full-framed VOQ, it is operated in the contention mode like the mailbox switch. The difference between our scheme and [16] is that our system has multiple parallel channels while there is only one in [16]. The challenge in multiple CR (Contention and Reservation) channels is to maintain packets in sequence.

The key innovation that enables us to do this is a new buffer management scheme, called I-VOQ (virtual output queue with insertion). There are three types of packets in an I-VOQ: fake packets, contention packets and reservation packets. A fake packet is generated by the I-VOQ itself every time an I-VOQ becomes empty. A reservation packet (a packet transmitted in the reservation mode) is always stored at the end of an I-VOQ. A contention packet (a packet transmitted in the contention mode) can only be stored at the head-of-line position of an I-VOQ if the head-of-line packet is a fake packet. Otherwise, a contention packet is blocked and has to be retransmitted later.

With the I-VOQ scheme, we give rigorous mathematical proofs for 100% throughput and in order packet delivery of the CR switch. By computer simulations, we also demonstrate that the average packet delay of the CR switch in light traffic is almost the same as that in the mailbox switch and it is considerably smaller than that in the UFS scheme. Moreover, when compared with the Padded Frame scheme [8], an improved scheme for the starvation problem in the UFS scheme, our delay performance is also much better in light traffic and comparable in heavy traffic.

In summary, the CR switch has the following advantages:

1) The CR switch achieves 100% throughput.
2) The CR switch maintains packets in order.
3) The communication overhead of the CR switch is $O(1)$.
4) The online computation overhead of the CR switch can be in the order of $\log N$.
5) In light traffic, the average delay of the CR switch is about $N/2$ as in the mailbox switch.
6) In heavy traffic, the average delay of the CR switch is still finite as in the UFS scheme.
7) The CR switch transits between the contention mode and the reservation mode based on local queue lengths at each input. Hence, the control of the CR switch is *distributed*.
8) The size of each input buffer is bounded by N^2.

From simulation, we will show that the CR switch performs much better in average delay than the Padded Frame (PF) scheme [8], the UFS scheme and the mailbox switch under all traffic loadings with uniform and nonuniform destination distributions. Compared with an input-buffered switch executing the iSLIP matching algorithm, the CR switch performs distinctly better under heavy traffic condition. When the traffic has nonuniform destination distributions, the iSLIP algorithm cannot achieve 100% throughput, while the CR switch can. However, the iSLIP switch has a better delay performance under light to medium traffic conditions. By simulation, we study two fairness problems of the CR switch. The first fairness problem arises because of the deterministic and periodic TDM connection pattern that the CR switch uses. This connection pattern produces a fixed priority order among inputs for any given output port. We propose a port re-mapping method to solve this fairness problem. In the second fairness problem, we observe that packets transmitted in the contention mode are likely to have longer delays than packets transmitted in the reservation mode. We note that a similar fairness problem exists in input-buffered switches with maximum weighted matching with longest queue first algorithm [14].

This paper is organized as follows: in Section II we propose the CR switch architecture and its operation. We then show that the CR switch delivers packets in order in Section III and achieves 100% throughput in Section IV. In Section V, by computer simulation, we study the delay of the CR switch, and compare it with the padded frame scheme. The paper is concluded in Section VI, where we address further research problems of the CR switch.

II. The Switch Architecture

In Figure 3, we show the switch architecture for an $N \times N$ CR switch. In the $N \times N$ CR switch, there are N input ports (*resp.* output ports), indexed by $i = 1, 2, \ldots, N$ (*resp.* $j = 1, 2, \ldots, N$). As in the generic load-balanced switches [3], [4], the CR switch also consists of two crossbar switches. The buffers between the two crossbar switches are called *central buffers*, indexed by $m = 1, 2, \ldots, N$ and the buffers in front of the first crossbar switch are called *input buffers*, indexed by $i = 1, 2, \ldots, N$. In the CR switch, we assume that packets are of the same size. Also, time is slotted and synchronized so that a packet can be transmitted within a time slot. We index time slots by $t = 1, 2, \ldots, \infty$. Unless otherwise specified, by input/output ports we mean those of the whole CR switch instead of a single crossbar switch.

In each input buffer, there are N Virtual Output Queues (VOQs). Each VOQ stores packets of the same output destination. We index the VOQ in input buffer i with output destination j by VOQ (i, j). Packets arriving at an input port are stored in one of the N VOQs according to their output destinations. Then packets in the N input buffers are sent to the N central buffers by the first $N \times N$ symmetric TDM (S-TDM) switch. There are two modes to send packets from the input buffers to the central buffers. One is the contention mode; the other is the reservation mode. A packet transmitted under the contention (*resp.* reservation) mode is called a contention (*resp.* reservation) packet. In the central buffers, there are N I-VOQs (VOQ with Insertion). Similar to a VOQ, each I-VOQ stores packets of the same output destination. We index the I-VOQ in cental buffer m with destination j by I-VOQ (m, j). Finally, packets stored in the N central buffers are transmitted to the N output ports through the second $N \times N$ symmetric TDM switch.

In the following subsections, we will illustrate the function of the S-TDM switch, the I-VOQ, and the contention and reservation modes. Finally, we present an example to visualize the operation of the whole CR switch.

A. Symmetric TDM switches

As shown in Figure 3, there are two $N \times N$ symmetric TDM switches in the CR switch. The connection patterns of these two switch fabrics are identical at the same time slot. Each symmetric TDM switch consists of N input ports (*resp.* output ports) generically indexed by $i_s = 1, 2, \ldots, N$ (*resp.* $j_s = 1, 2, \ldots, N$). As in the mailbox switch [5], an $N \times N$ symmetric TDM switch is merely an $N \times N$ crossbar switch that implements the following periodic connection patterns: input i_s is connected to output j_s at time t if and only if

$$(i_s + j_s) \text{ mod } N = (t + 1) \text{ mod } N. \tag{1}$$

In other words, for any positive integer g, input i_s is connected to output 1 at time $i_s + (g - 1)N$, output 2 at time $i_s + 1 + (g - 1)N, \ldots,$ and output N at time $i_s - 1 + gN$. Also, it is clear from (1) that every connection pattern in a symmetric TDM switch is *symmetric* (as input i_s is connected to output j_s if and only if output i_s is connected to input j_s). As such, output j_s is connected to input 1 at time $j_s + (g - 1)N$, input 2 at time $j_s + 1 + (g - 1)N$,..., and input N at time $j_s - 1 + gN$. If each input/ouput pair of the whole CR switch is built in the same line card, the symmetric connection patterns provide each central buffer a feedback path to its connected input buffer through its connected output port.

B. I-VOQs

To maintain packets (both contention packets and reservation packets) in order, we invent a new buffer management scheme, called Virtual Output Queue with Insertion (I-VOQ), for the *central* buffers. Similar to a standard VOQ, an I-VOQ stores packets of the same destination. The difference is that in an I-VOQ an arriving packet is allowed to replace its head-of-line (HOL) packet. There are three kinds of packets in an I-VOQ: fake packets, contention packets, and reservation packets. A fake packet is generated by the I-VOQ itself every time an I-VOQ becomes empty. By so doing, a fake packet is always stored as a HOL packet and this guarantees that there exists at least one packet in an I-VOQ. When a contention packet arrives and the HOL packet of an I-VOQ is a fake packet, then the fake packet is replaced by the contention packet and the contention packet becomes the HOL packet. Otherwise, the arriving contention packet is *rejected*. On the other hand, when a reservation packet arrives, it is attached to the tail of an I-VOQ (we assume that the size of every I-VOQ is infinite so that no reservation packet is lost due to buffer overflow). As there is at least one packet in an I-VOQ, we note that a reservation packet cannot be stored as a HOL packet upon its arrival at an I-VOQ. When an I-VOQ is connected to its destination output, its HOL packet (fake or not) is transmitted to the output and removed from the I-VOQ. Packets behind the HOL packet are then moved up one position, i.e., the p^{th} packet becomes the $(p - 1)^{th}$ packet.

We note that the CR switch needs one bit of feedback information from the central buffer to the connected input buffer to indicate whether the transmission of a contention packet is successful. (In practice, one also needs this for a reservation packet as it might also be rejected due to buffer overflow.) As in the mailbox switch [5], this one bit information can be sent via the feedback path provided by the two symmetric TDM switches.

C. Contention mode and reservation mode

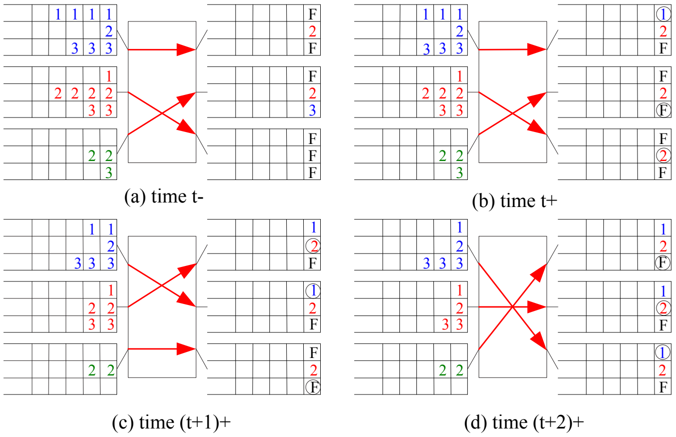
As pointed out in the pioneer work by Tobagi and Kleinrock [16] for a multiple access channel, one should have the CR switch operating in the contention mode under light traffic to have low delay, and in the reservation mode under heavy traffic to maintain system stability. The question is then how the CR switch knows whether the traffic is light or heavy without measuring it.

To answer this question, we operate the CR switch in a frame-based manner as in the UFS scheme [10]. Every frame consists of N consecutive time slots. However, the beginning time slots of frames are different for different inputs/outputs. Specifically, frame f of input i (*resp.* output j) begins at the f^{th} time when input i (*resp.* output j) is connected to the *first* central buffer. As such, we have from (1) that frame f of input i (*resp.* output j) consists of time slots $i + (f - 1)N, \ldots, i - 1 + fN$ (*resp.* output $j + (f - 1)N, \ldots, j - 1 + fN$). If the number of packets in a VOQ at an input port is not less than N, that VOQ is called a full-framed VOQ. At the beginning of a frame, if an input has a full-framed VOQ, then it is considered in heavy traffic and is operated in the reservation mode. That frame is then called a reservation frame. Otherwise, it is considered in light traffic and is operated in the contention mode. Accordingly, that frame is called a contention frame.

Now we describe the detailed operations for these two modes.

The reservation mode: each input i keeps a reservation pointer for selecting a full-framed VOQ as in iSLIP [13]. At the beginning of a reservation frame, the full-framed VOQ that is *clockwise* the closest to the pointer is selected. The pointer is then incremented clockwise to one location beyond the selected VOQ. Suppose that VOQ (i, q) is selected. In each time slot of that frame, the HOL packet from VOQ (i, q) is sent to the connected central buffer m. One bit of information is also transmitted to indicate that this packet is a reservation packet. The packet is then stored at the tail of I-VOQ (m, q).
The contention mode: each input i keeps a contention pointer for selecting a nonempty VOQ as in iSLIP [13]. In each time slot of a contention frame, the nonempty VOQ that is *clockwise* the closest to the pointer is selected. The pointer is then incremented clockwise to one location beyond the selected VOQ. Suppose that VOQ (i, q) is selected in a time slot of that frame. The HOL packet of VOQ (i, q) is copied and sent to the connected central buffer m in that time slot. One bit of information is also transmitted to indicate that this packet is a contention packet. If the HOL packet of I-VOQ (m, q) is a fake packet, we replace the HOL packet of I-VOQ (m, q) by this contention packet and feed back one bit of information to indicate a successful transmission. Otherwise, we reject the contention packet and feed back one bit of information to indicate a failed transmission. If the transmission is successful, the HOL packet of VOQ (i, q) is removed and packets behind it are moved up one position. Otherwise, the HOL packet remains the HOL packet of VOQ (i, q).

Note that there are various ways to select VOQs in the contention mode. This could result in different delay performance. We will discuss this issue in Section V-B.


(a) time t-

(b) time t+

(c) time (t+1)+

(d) time (t+2)+

Fig. 4. An example to illustrate the operation of the CR switch

Before we leave this section, we present an example to illustrate the operation of the CR switch. In this example, we consider a 3 × 3 switch and demonstrate the operation of the switch for 3 time slots. Assume that time $t = 3n+1$ for some integer n. The connection pattern and the buffer contents right before the packets are moved from input buffers to central buffers and from central buffers to outputs are shown in Figure 4 (a). The buffer contents after the packets are moved are shown in Figure 4 (b). Note that from the second paragraph of Section II, packets in the input buffers are moved to the central buffers first and then packets in the HOL of I-VOQs are then moved to the connected outputs. The numbers in the buffers are the destination ports of the packets. Encircled numbers in the central buffers correspond to the packets that are moved in the shown time slot. In this example, we focus on the operation of the CR switch and ignore the new arriving packets for simplicity. Note that for input 1, frames begin at time slots t, $t+3$, $t+6$, etc. For input 2 (resp. input 3), frames begin at $t-2, t+1, t+4$, (resp. $t-1, t+2, t+5$) etc. Since input 1 has full-framed VOQs (VOQ $(1, 1)$ and VOQ $(1, 3)$), input 1 chooses to operate in the reservation mode and transmit packets from VOQ $(1, 1)$. In this example, we assume that at time $t-2$ input 2 chooses to operate in the reservation mode. Thus, at time t input 2 sends a packet from VOQ $(2, 2)$ to I-VOQ $(3, 2)$. Assume that at time $t-1$ input 3 chooses the contention mode. Thus, at time t, input 3 selects VOQ $(3, 2)$ and transmits its HOL packet to central buffer 2. Since the HOL of I-VOQ $(2, 2)$ is occupied, this transmission fails and the transmitted packet remains in VOQ $(3, 2)$ for retransmission in the future. Then, the HOL packets in the I-VOQs are transmitted to their connected outputs. Specifically, since the connection patterns are symmetrical, central buffer 1 transmits a fake packet to output 1 and moves the newly arrived packet to the HOL position of I-VOQ $(1, 1)$. I-VOQ $(2, 3)$ transmits a packet to output 3 and inserts a fake packet to its HOL position. Similarly, central buffer 3 is connected to output 2. Thus, I-VOQ $(3, 2)$ transmits the fake HOL packet to output 2 and moves the newly arrived packet to its HOL position. The resulting buffer contents are shown in Figure 4 (b).

At time $t+1$, input 2 is connected to central buffer 1. Since input 2 has a full-framed VOQ (VOQ $(2, 2)$), it chooses to operate in the reservation mode (see Figure 4 (c)). At time $t+2$, input 3 is connected to central buffer 1. Input 3 does not have a full-framed VOQ and it can only operate in the contention mode in this frame (Figure 4 (d)).

III. In Order Delivery

In this section, we show how the CR switch delivers packets in order. Let flow (i, j) be the sequence of packets from input i to output j. Let packet k be the k^{th} packet of flow (i, j). The CR switch delivers packets of the same flow in order if packet k departs the switch earlier than packet $k+1$.

A. General Properties of I-VOQs

Now we show some general properties of I-VOQs that are needed for proving in order delivery. Unless otherwise specified, we consider flow (i, j) and central buffer m. For clarity, indices i, j, and m are sometimes omitted.

Now suppose that input i is connected to central buffer m at time t. Let w_ℓ be the offset from time t that central buffer m is connected to output j for the ℓ^{th} time. Then central buffer m is connected to output j at time $t + w_\ell$. Clearly, we have $w_1 = 0$ if $j = i$ as the connection is symmetric (and the central buffers receive packets first and send packets later). As in (1), the connection is sequential and periodic. Thus, we have that $w_1 = j - i$ if $j > i$ and $w_1 = N + j - i$ if $j < i$. For all these three cases, we have

$$w_1 = (j - i) \bmod N. \tag{2}$$

As the connection patterns in symmetric TDM switches are periodic with period N, it then follows that

$$w_\ell = (j - i) \bmod N + (\ell - 1)N. \tag{3}$$

Note that the waiting time w_ℓ only depends on i and j and it does not depend on m.

As every time central buffer m is connected to output j, the HOL packet (fake or not) of I-VOQ (m, j) is sent to output j and every packet behind the HOL packet is moved up one position. As such, w_ℓ can be viewed as the (virtual) waiting time for the ℓ^{th} packet in I-VOQ (m, j) at time t. This leads to the following properties:

Proposition 1 *Suppose that input i is connected to central buffer m at time t. If packet k is the p^{th} packet of I-VOQ (m, j) at time t, then*

(i) *packet k becomes the $(p-\ell)^{th}$ packet at time $t+w_\ell$, and*

(ii) *packet k departs I-VOQ (m, j) at time $t + w_p$.*

In the operation of the CR switch, contention packets can only be stored as HOL packets of I-VOQs. On the other hand, reservation packets, transmitted in a frame of N consecutive time slots, can only be stored at the tails of I-VOQs. As in the UFS scheme [10], one might expect that any N reservation packets transmitted in the same frame from an input are also stored in the same position of N I-VOQs. Moreover, as w_ℓ in (3) does not depend on m, it follows from Proposition 1(ii) that any N reservation packets transmitted in a frame of an input are also sent to their output consecutively in a frame of their output. This is stated in the following property. Its formal proof is given in Appendix A.

Proposition 2 *Suppose that frame f of input i is a reservation frame that contains packets for output j.*

(i) *For $m = 1, 2, \ldots, N$, the packet transmitted in the m^{th} slot of frame f is stored as the p^{th} packet in I-VOQ (m, j) for some fixed $p \geq 2$.*

(ii) *For $m = 1, 2, \ldots, N$, the packet transmitted in the m^{th} slot of frame f is sent to its output in the m^{th} time slot of frame $\tilde{f}$ of output j for some $\tilde{f}$.*

In view of Proposition 2(ii), a frame of an output can also be classified as a reservation frame if it contains all reservation packets, and as a contention frame otherwise.

B. The proof for in order delivery

Now we show that packets of the same flow are always delivered in order. Recall in the beginning of Section III that packet k represents the k^{th} packet of flow (i, j). To prove in order delivery, we will prove that packet k departs earlier than packet $k+1$ for any integer k. There are three cases that need to be considered: (i) packet k is a contention packet, (ii) both packet k and packet $k+1$ are reservation packets, and (iii) packet k is a reservation packet and packet $k+1$ is a contention packet.

Firstly, if packet k is a contention packet, then packet k departs earlier than packet $k+1$, no matter whether packet $k+1$ is a contention packet or a reservation packet. This is because packet k is a contention packet and it is stored as the HOL packet of an I-VOQ. From Proposition 1(ii), we know that if packet k is transmitted to an I-VOQ at time t_1, it will depart the switch at time $t_1 + w_1$. Also, if packet $k+1$ is transmitted at time t_2, it will depart the switch at time t_2+w_p, for some $p \geq 1$. Since $t_1 < t_2$ and $w_1 \leq w_p$, packet k departs earlier than packet $k+1$.

Then, if both packet k and packet $k+1$ are reservation packets and they are in the same reservation frame, this is the case addressed in Proposition 2(ii). On the other hand, if packet $k+1$ belongs to a later frame, it is clear that packet $k+1$ departs in a later frame.

In the third case, packet k must be the last packet in a reservation frame and packet $k+1$ belongs to a later contention frame. Suppose that packet k is transmitted to I-VOQ (N, j) as the p^{th} packet at time t for some $p \geq 2$. Then it follows from Proposition 2(i) that packets $k-N+m$, $m = 1, 2, \ldots, N-1$, are also transmitted to I-VOQ (m, j) as the p^{th} packet at time $t - N + m$. As reservation packets are attached to the tails of I-VOQs (and only reservation packets can be stored behind the HOL packet), we know that the ℓ^{th} packet, $\ell = 2, \ldots, p$, of I-VOQ (m, j) are all reservation packets at time $t-N+m$ for $m = 1, 2, \ldots, N$. From Proposition 1(ii), the ℓ^{th} packet of I-VOQ (m, j) departs the switch at time $t - N + m + w_\ell$. As $m = 1, 2, \ldots, N$ and $\ell = 2, 3, \ldots, p$, time slots of output j from $t - N + 1 + w_2 = t + 1 + w_1$ to $t + w_p$ are reserved before packet $k+1$ is transmitted to central buffers. Therefore packet $k+1$ can not depart the switch between $t + 1 + w_1$ and $t + w_p$. As packet $k+1$ is transmitted after time t, from Proposition 1 (ii), packet $k+1$ departs the switch on or after $t+1+w_1$. As time slots from $t+1+w_1$ to $t+w_p$ are reserved by reservation packets, we conclude that packet $k+1$ departs the switch after $t + w_p$ which is, from Proposition 1(ii), the departure time of packet k. Thus, packet $k+1$ must depart later than packet k.

From these three cases, we have the following theorem.

Theorem 3 ***(in order delivery)*** *The CR switch delivers packets of the same flow in order.*

IV. 100% Throughput

In this section, we show that the CR switch indeed achieves 100% throughput. This is done by showing two stronger results: (i) the total number of packets in every input buffer is bounded above by N^2 in Corollary 8, and (ii) the total number of packets in the central buffers (I-VOQs) destined for a particular output is bounded above by the sum of the total number of packets in the corresponding output buffer of the output-buffered switch and $N^3 + 2N$ in Corollary 12.

To study the number of packets in the input buffers and the I-VOQs, we need to introduce the concepts of work conserving modes for queues that have *at most one packet departure* in a time slot.

Definition 4 ***(WC mode)*** *A queue is in the work conserving (WC) mode if there is one departure in each time slot whenever the queue is nonempty.*

Clearly, each output buffer of an output-buffered switch is in the work conserving mode for every time slot. However, both the input buffers and the I-VOQs of the CR switch are not in the work conserving mode for every time slot. They fall in a weaker concept of work conserving mode defined below.

Definition 5 ***(WC(K, D) queue)*** *A queue is work conserving with response workload K and response delay D (denoted by $WC(K, D)$) if it satisfies the following: when the queue length is smaller than K at time $t-1$ and becomes longer than or equal to K at time t, this queue begins to be in the WC mode not later than time $t + D$. Moreover, this mode must continue until the queue length becomes smaller than K again.*

In the following lemma, we derive a bound between the queue length of a WC queue and that of a $WC(K, D)$ queue.

Lemma 6 *Let $Q_{WC}(t)$ (resp. $Q_{WC(K,D)}(t)$) be the number of packets in a WC (resp. $WC(K, D)$) queue at time t. Suppose that both queues are subject to the same arrival process and they both are empty at time 0. Then*

$$Q_{WC(K,D)}(t) \leq Q_{WC}(t) + K + D - 1. \tag{4}$$

Proof. Let a busy period of a $WC(K, D)$ queue be the period of time in which there are more than or equal to K packets in the $WC(K, D)$ queue. All we need to proof is that (4) holds for every time slot in a busy period of the $WC(K, D)$ queue.

Let the busy period of the $WC(K, D)$ queue start from time a. Also, let $A(\tau)$ be the cumulative number of packets arriving at the $WC(K, D)$ queue by time τ. We first show that if $a \leq t \leq a + D - 1$, then (4) holds. By definition, we have

$$Q_{WC(K,D)}(a - 1) \leq K - 1. \tag{5}$$

As there is at most one departure in each time slot, we have

$$Q_{WC}(t) \geq Q_{WC}(a-1)+A(t)-A(a-1)-(t-a+1). \tag{6}$$

As a $WC(K, D)$ queue might have no departure, we have

$$Q_{WC(K,D)}(t) \leq Q_{WC(K,D)}(a - 1) + A(t) - A(a - 1). \tag{7}$$

From (5), (6) and (7), we have

$$Q_{WC(K,D)}(t) \leq Q_{WC}(t) + (t - a + 1) + K - 1. \tag{8}$$

Thus, (4) holds at t where $a \leq t \leq a + D - 1$.

On the other hand, if $t \geq a+D$, then the $WC(K, D)$ queue is in the work conserving mode between $a + D$ and t. Then we have

$$\begin{aligned} Q_{WC(K,D)}(t) = Q_{WC(K,D)}(a + D - 1) + \\ A(t) - A(a + D - 1) - (t - a - D + 1). \end{aligned} \tag{9}$$

From (8) with t substituted by $a+D-1$, (6) with a substituted by $a + D$ and (9), we have $Q_{WC(K,D)}(t) \leq Q_{WC}(t) + K + D - 1$. In this case, (4) holds, too. This completes the proof. ■

We have the following work conserving property for input buffers.

Proposition 7 *Each input buffer is work conserving with response workload $N(N - 1) + 1$ and response delay $N - 1$.*

Proof. Note that if there are more than $N(N - 1)$ packets in an input buffer, then there is a full-framed VOQ in that input buffer. As such, the input buffer will be in the reservation mode at the beginning slot of the next frame and it will continue to be in the reservation mode until there is no full-framed VOQ. Note that there is exactly one packet sent out from that input buffer in every time slot when the input buffer is in the reservation mode. Thus, the response workload is $N(N-1)+1$. As the time it takes to the beginning time slot of the next frame is bounded above by $N-1$, the response delay is $N-1$. This completes the proof. ■

Note that there is at most one packet arrival at an input buffer in a time slot. If we put the same arrival process to a work conserving queue, the number of packets in that work conserving queue is at most 1. Thus, along with Lemma 6 and Proposition 7, we have the following corollary.

Corollary 8 ***(packets in input buffers)*** *The number of packets in an input buffer is bounded above by N^2.*

From Corollary 8, large memory space is only needed in the central buffers. To show the work conserving property for the I-VOQs, we need to introduce the following definition.

Definition 9 *We define Q_j as a conceptual queue which contains the union of non-HOL packets in I-VOQ (m, j) for $m = 1, 2, \ldots, N$.*

As a fake packet or a contention packet can be stored only as a HOL packet in an I-VOQ, a non-HOL packet must be a reservation packet. Thus, Q_j contains all non-HOL reservation packets with destination j stored in the N I-VOQs.

Proposition 10 *For each $j = 1, 2, \ldots, N$, Q_j is work conserving with response workload 1 and response delay $N - 1$.*

Proof. Suppose Q_j is empty at time $t - 1$ and becomes nonempty at time t. Since the first packet of a reservation frame of any input is always transmitted to the first central buffer, there is exactly one packet, called packet k, transmitted at time t to I-VOQ $(1, j)$ and stored as the second packet of I-VOQ $(1, j)$. Without loss of generality, assume that packet k is transmitted from input i. From Proposition 1(i), we know that at time $t+w_1$ packet k becomes the HOL packet of I-VOQ $(1, j)$ and thus leaves Q_j. Since $w_1 \leq N-1$, the response time of Q_j is at most $N-1$ time slots and the response workload is 1.

It remains to show that there is exactly one departure in each time slot from Q_j after $t+w_1$ until Q_j becomes empty again. From Proposition 2(i) and Proposition 1(i), there are packets departing Q_j from time $t+w_1$ to time $t+w_1+N-1$. Also, we know that $t+w_1$ is the beginning time slot of a frame of output j. At the beginning time slot of the next frame of output j, i.e., $t+w_1+N$, if Q_j is empty, then we complete our argument. On other hand, if Q_j is still nonempty at $t+w_1+N$, then there is a reservation packet stored as the second packet of I-VOQ $(1,j)$ (since the first packet of a reservation frame of any input is always transmitted to I-VOQ $(1,j)$). Using Proposition 2(i) and Proposition 1(i) again, there are packets departing Q_j from time $t+w_1+N$ to time $t+w_1+2N-1$. Repeating the same argument, we conclude that there is a departure from Q_j until Q_j is empty. ■

Using Proposition 10 and Lemma 6, we derive in the following lemma a bound for the difference between the queue length of Q_j and that of the corresponding output-buffered switch. The proof is given in Appendix B.

Lemma 11 *Suppose that the CR switch and the output-buffered switch are subject to the same arrival process. Let $Q^R(t)$ be the number of packets in Q_j at time t and $Q^O(t)$ be the number of packets in the j^{th} output buffer of the corresponding output-buffered switch at time t. Then,*

$$Q^R(t) \leq Q^O(t) + N^3 + N. \tag{10}$$

Observe that there are at most N HOL packets destined for output j in the central buffers at any time t. This leads to the following corollary.

Corollary 12 ***(packets in central buffers)*** *Suppose that the CR switch and the output-buffered switch are subject to the same arrival process. Let $Q^C(t)$ be the number of packets destined for output j in the central buffers at time t and $Q^O(t)$ be the number of packets in the j^{th} output buffer of the corresponding output-buffered switch at time t. Then,*

$$Q^C(t) \leq Q^O(t) + N^3 + 2N. \tag{11}$$

V. Simulations

In this section, we study the delay of the CR switch. In the experiments, we set the switch size N to be 32. The number of time slots for each experiment is 10^6. Let ρ be the average arrival rate to an output of the switch. We assume that arrival processes to the N input ports are independent, and consider the following four traffic models:

(i) uniform i.i.d. traffic,
(ii) uniform Pareto traffic,
(iii) hotspot i.i.d. traffic, and
(iv) hotspot Pareto traffic.

For the i.i.d traffic models in (i) and (iii), a packet is generated *independently* in a time slot in an input with probability ρ. On the other hand, for the Pareto traffic models (see [3]) in (ii) and (iv), packets are generated in bursts. With probability ρ, there are packets in a burst (and with probability $1-\rho$ there are no packets in a burst). Packets in the same burst are sent to the same destination. The length of each burst is generated *independently* according to the following (truncated) Pareto distribution:

$$P(\text{burst length} = s) = \frac{C}{s^{2.5}}, \tag{12}$$

where $s = 1, 2, \ldots, 1000$, and $C = (\sum_{s=1}^{1000} \frac{1}{s^{2.5}})^{-1}$ is the normalization constant.

For the uniform traffic models in (i) and (ii), the destination of a packet (or packets in the same burst) is selected according to the uniform distribution in $[1, N]$, i.e., each output port is selected as the destination of a packet (or packets in the same burst) with the same probability $1/N$. On the other hand, for the hotspot traffic models (see [6]) in (iii) and (iv), packets from input i are destined to output i with probability 0.5 and to each of the other outputs with probability $0.5/(N-1)$.

A. Average delay

In the first experiment, we study the average delay of the CR switch under the uniform i.i.d. traffic. In Figure 5, we plot the average delay of three two-stage switches: the CR switch, the contention scheme and the UFS scheme in [10]. Among them, the *contention scheme* is the CR switch without the reservation mode. On the other hand, the UFS scheme is the CR switch without the contention mode and with I-VOQs replaced by VOQs. In Figure 5, we observe that the advantage of the contention scheme yields very low delay under light traffic while the advantage of the UFS scheme is maintaining system stability under heavy traffic. The CR switch, however, has both advantages.

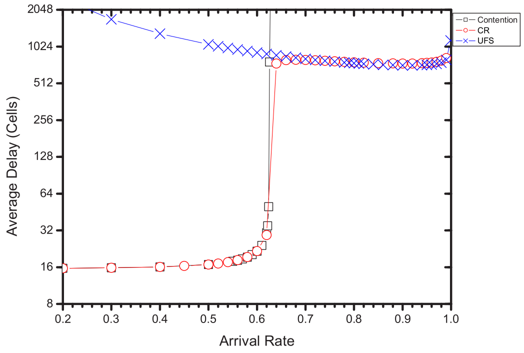


Fig. 5. The average delay of the contention scheme, the UFS scheme and the CR switch.

For the contention scheme, the maximum throughput seems to be around $1-e^{-1} \approx 0.63$ and the average delay seems to be around $N/2$ before reaching the maximum throughput. The intuition behind this is that there are few collisions under light traffic. A packet, upon its arrival, is transmitted immediately to the central buffer as a HOL packet. Thus, the delay of a packet is almost the same as the time that a HOL packet needs to wait for the connection to its destined output. Therefore, the average delay is around $N/2 = 16$ under light traffic. The quantity, $1-e^{-1}$, is known as the maximum throughput of an input-buffered switch with collision dropping [9]. As argued in [5] for the mailbox switch with $\delta = 0$, one can argue that the contention scheme has the same maximum throughput as that of an input-buffered switch with collision dropping.

For the CR switch, the average delay is low under light traffic as in the contention scheme. Then it transits to the UFS scheme under medium traffic. As in the UFS scheme, the CR switch still has finite average delay under heavy traffic. In Figure 5, we observe that there are three regions in the delay curve of the CR switch. In the first region, $0 \leq \rho \leq 0.63$, the delay curve coincides with that of the contention scheme. This is because there is almost no full-framed VOQ when the load is under 0.63. In the transition region, $0.63 \leq \rho \leq 0.7$, the delay curve is below those of the other two schemes, since in the CR switch packets can still be transmitted to I-VOQs before some full-framed VOQs are formed. In the heavy load region, $0.7 \leq \rho < 1$, the delay curve is close to that of the UFS scheme. This is because it is very likely to have some full-framed VOQs in input buffers under very heavy traffic.

For the UFS scheme, even though the average delay is finite under heavy traffic, the average delay is large under light traffic. Moreover, as shown in Figure 5, there are two regions in the delay curve for the UFS scheme. In the first region, $0 \leq \rho \leq 0.92$, the delay curve is monotonically decreasing, while in the second region, $0.92 \leq \rho < 1$, the delay curve becomes monotonically increasing. This is because the delay of a packet consists of two parts: (i) the delays incurred in an input buffer, and (ii) the delay incurred in a central buffer. In light traffic, the major portion of the delay of a packet is from the delay in an input buffer as it needs to wait until a full-framed VOQ in an input buffer is formed. Clearly, the lighter the traffic is, the longer it takes to heap up a full-framed VOQ. As such, the delay curve is decreasing in the first region. On the other hand, in heavy traffic, the delay of a packet is dominated by the queueing delay in a central buffer. As the queueing delay is increasing in the average arrival rate, the delay curve is increasing in the second region.

B. Advancing the contention pointers

For the CR switch, the delay in the transition region can be affected by how the VOQs are selected when their inputs are in the contention mode. We use a pointer called contention pointer to designate the selected VOQ from which a packet will be transmitted in contention mode. In the transition region, the arrival rate exceeds the maximum throughput of the contention scheme, and some full-framed VOQs start to form. As described in Proposition 2(ii), a full-framed VOQ, when selected, reserves a frame of N consecutive output time slots and hence N consecutive HOL packets of the N I-VOQs during that frame of output time slots. As such, when a HOL packet transmitted in the contention mode to an I-VOQ is rejected, it is very likely that it will be rejected again if it is transmitted immediately in the next time slot. Thus, when the previous transmission is failed, it might be better to select another input VOQ by advancing the contention pointer. On the other hand, if the previous transmission is successful, it might be better to select the same input VOQ until it is empty. Before we present our study on the mechanisms to update the contention pointer, we present the following acronyms for easy referencing.

- **SAFA**: if **S**uccess, **A**dvance the pointer to the next nonempty VOQ; if **F**ailed, **A**dvance the pointer to the next nonempty VOQ.
- **SAFP**: if **S**uccess, **A**dvance the pointer to the next nonempty VOQ; if **F**ailed, **P**ersist.
- **SPFA**: if **S**uccess, **P**ersist; if **F**ailed, **A**dvance the pointer to the next nonempty VOQ.
- **SPFP**: if **S**uccess, **P**ersist; if **F**ailed, **P**ersist.
- **SPFA-Longest**: if **S**uccess, **P**ersist; if **F**ailed, **A**dvance the pointer to the **Longest** VOQ.
- **SPFA-LMQ**: if **S**uccess, **P**ersist; if **F**ailed, **A**dvance the pointer to the VOQ whose queue length is **L**onger than or equal to the **M**edium **Q**ueue length of the nonempty VOQs in the input.

Since a contention can succeed or fail and the contention pointer can be advanced or persisted, one has four possible schemes, *i.e.* SPFA (Success: Persist/Failure: Advance), SAFA (Success: Advance/Failure: Advance), SAFP (Success: Advance/Failure: Persist), and SPFP (Success: Persist/Failure: Persist) schemes. In the generic algorithm presented in Section II-C, the contention pointer is advanced using the SAFA scheme as the contention pointer is always advanced. To verify the intuition described in the last paragraph, we simulate these four methods for both the uniform i.i.d. traffic and the hotspot Pareto traffic in Figure 6 and Figure 7. As shown in these figures, the SPFA scheme has the least average delay for the entire region of the arrival rates. As such, we suggest the SPFA scheme be used in the CR switch for advancing the contention pointers.

In the SPFA scheme described in the last paragraph, we simply advance the pointer to the next non-empty VOQ when the previous transmission is failed. The question is whether there is a better choice. Intuitively, the longer the VOQ is, the more consecutive packets can be transmitted successfully to reduce the average delay. In this experiment, three methods of selecting VOQs are investigated: (i) the next nonempty VOQ, (ii) the next VOQ whose queue length is Longer than or equal to the Median Queue length (LMQ) of the nonempty VOQs in the input, and (iii) the longest VOQ among the VOQs in the input. The first method is simply the SPFA scheme. We denote the second and the third methods by SPFA-LMQ and SPFA-Longest. In Figure 6 and Figure 7, we plot the average delay for these three methods of selecting VOQs under the uniform i.i.d. traffic and the hotspot Pareto traffic respectively. As expected (from the intuition of selecting a longer queue to reduce the average delay), the curve of the nonempty queue is higher than that of the curve of the LMQ. However, to our surprise, the curve of the longest queue is higher than that of the curve of the LMQ in most traffic conditions. This might be explained as follows: if the longest queue is selected and it results in a failed transmission, then with high probability the longest queue will be selected again. As such, it behaves like the SPFP scheme that yields large delay. As such, the right intuition is to select a VOQ long enough to have consecutive successful transmissions, but not too long to keep the freedom of advancing to other VOQs when there is a failed transmission. It seems that the LMQ method fits the intuition very well as there are often several VOQs with queue length longer than the median queue length. As such, there is no problem to advance the contention pointer to other VOQs in the LMQ method. To summarize, we suggest the contention pointer be advanced using the SPFA-LMQ scheme.

Before we close this subsection, we discuss the computation complexity of the LMQ method. The online computation overhead of the LMQ method involves searching for the median among the queue lengths of nonempty VOQs in an input buffer. We note that this can be done in the order of $\log N$ time complexity by maintaining heap structures. To do this, we maintain two heaps, H_S and H_L. Let N^* be the number of nonempty VOQs in the input buffer. Heap H_S keeps the queue length information of the lower $\lfloor N^*/2 \rfloor$ VOQs while heap H_L keeps the queue length information of the remaining $N^* - \lfloor N^*/2 \rfloor$ VOQs. Heap H_S (*resp.* H_L) is maintained as a max-heap (*resp.* min-heap) in which each father is not smaller (*resp.* not larger) than all his children. Then the root of H_S can be considered as the median. The change of the value in one node requires $O(\log N)$ steps to percolate or sift [1]. As there is at most one arrival and one departure in each time slot, the complexity of such an approach is then $O(\log N)$.

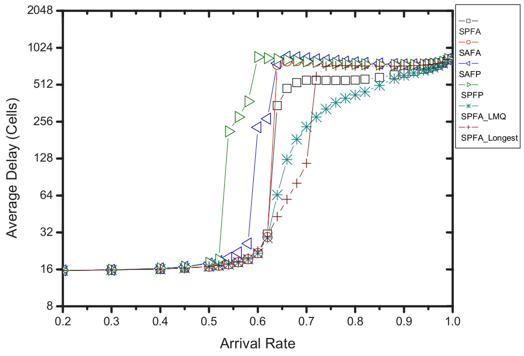


Fig. 6. The average delay of selecting a VOQ under the uniform i.i.d. traffic

C. *Comparison with the padded frame scheme*

In this section, we compare the average delay between the Padded Frame (PF) scheme in [8] and the CR switch with SPFA-LMQ. The PF scheme is an improved version of the UFS scheme. As the UFS scheme, it also operates in frames. If there is a full-framed VOQ, the longest VOQ is selected and N packets from that VOQ are sent to the N central buffers. Otherwise, the longest VOQ is selected and the partial frame of that VOQ is padded with fictitious packets to form a padded frame with N packets. The padded frame is sent only if the total number of padded frames in the central buffers does not exceed a threshold TH. By so doing, the average packet delay can be reduced in light traffic. Clearly, when TH is 0, it reduces to the UFS scheme. The reason that we choose the PF scheme for comparison is that both the PF scheme and the CR switch are based on the load-balanced architecture. They both achieve 100% throughput without speedup and deliver packets in sequence without re-sequencing buffers.

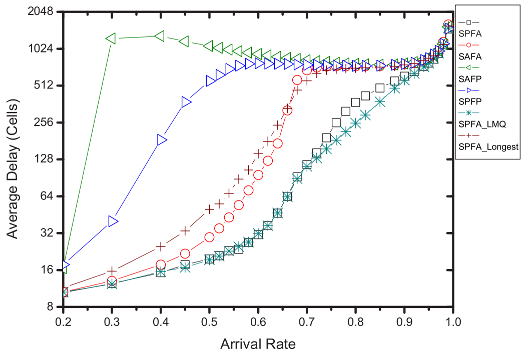


Fig. 7. The average delay of selecting a VOQ under the hotspot Pareto traffic

In our experiments, we choose $TH = 4$ as it is the suggested threshold in [8]. As shown in Figure 8, Figure 9, Figure 10 and Figure 11, the average delay of the CR switch (CR_avg) is much lower than that of the PF scheme (PF_avg) under light traffic. Moreover, these two curves are very close to each other under heavy traffic.

D. *Comparison with the iSLIP algorithm and the ideal output-buffered switch*

In this section, we first compare the delay performance of the CR switch with a famous practical input-buffered switch: the iSLIP [13] in Figure 8, Figure 9, Figure 10 and Figure 11. From those figures, we observe the followings:

1) Under the uniform traffic in Figure 8 and Figure 9, the delay of both the CR switch and the iSLIP are finite.
2) Under the hotspot traffic in Figure 10 and Figure 11, the iSLIP algorithm cannot achieve 100% throughput when the arrival rate is greater than 0.8. Nevertheless, the delay of the CR switch remains finite.
3) Under the uniform i.i.d. traffic in Figure 8, the delay of the iSLIP algorithm is much lower than that of the CR switch.
4) Under the Pareto traffic, the delay difference between the iSLIP and the CR switch is much smaller than under the i.i.d. traffic.

The last observation is due to the burst reduction property (as previously reported in [3]) that the CR switch inherits from a generic load-balanced switch. As pointed out in [12], the Internet traffic could be very bursty. Thus, we expect that the average delay of the CR switch might be comparable to that of the iSLIP algorithm when the Internet traffic is lightly loaded. However, the delay performance of the CR switch is much better when the Internet traffic is heavily loaded.

From Figure 8, Figure 9, Figure 10 and Figure 11 we see that the average delay of the CR switch converges to that of an ideal output-buffered switch under heavy traffic condition. This observation is consistent with the theoretical result in [3] that the average delay of a generic load balanced switch converges to that of the ideal output-buffered switch for a certain uniform bursty traffic model in heavy load. As discussed in [3], the first stage of a load balanced switch effectively reduces burst lengths and thus can approach the performance of an ideal output-buffered switch under heavy traffic. From Figure 9 and Figure 11 we see that the average delay of the CR switch is very close to that of the ideal output-buffered switch under the heavily loaded Pareto traffic. As the average queue length can be derived from the average delay by using Little's formula, we also expect that the average memory requirement for the CR switch should be comparable to that for the ideal output-buffered switch when the traffic is heavy and bursty (even though the worst case memory bound in Corollary 12 is $O(N^3)$). Finally, we note that there exist switches in the literature that guarantee $O(1)$ delay bounds (see e.g., [7] [15]). However, these delay bounds are at the cost of speedup of 2.

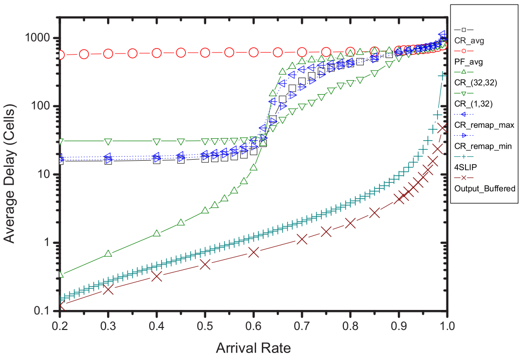


Fig. 8. The average delay of the PF scheme, the CR switch, the iSLIP algorithm and the ideal output-buffered switch under the uniform i.i.d. traffic

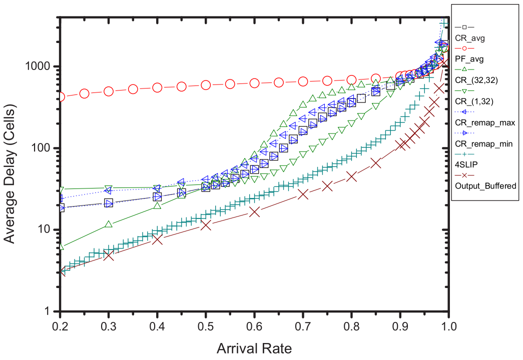


Fig. 9. The average delay of the PF scheme, the CR switch, the iSLIP algorithm and the ideal output-buffered switch under the uniform Pareto traffic

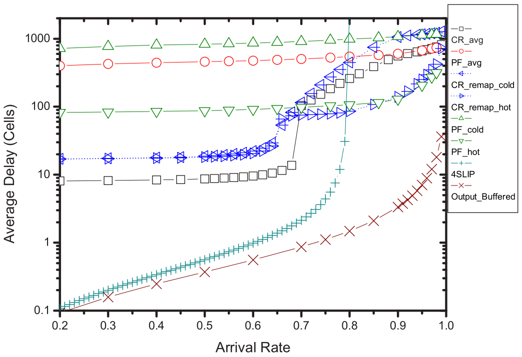


Fig. 10. The average delay of the PF scheme, the CR switch, the iSLIP algorithm and the ideal output-buffered switch under the hotspot i.i.d. traffic

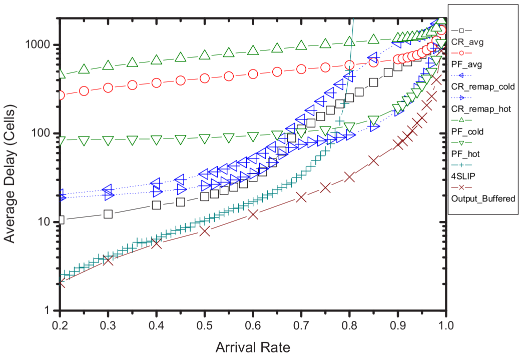


Fig. 11. The average delay of the PF scheme, the CR switch, the iSLIP algorithm and the ideal output-buffered switch under the hotspot Pareto traffic

E. Fairness Issues

In this section we discuss some fairness problems associated with the CR switches. It is well known that switches using deterministic and periodic TDM connection patterns can have fixed priority order among inputs for the same output port. One example is the mailbox switch [5]. The CR switch inherits a fairness problem from its predecessor, the mailbox switch. We now briefly describe this problem. Consider packets destined to output 32. Suppose input 1 is connected to central buffer m at time t. Then, output $N = 32$ was connected to central buffer m at time $t-1$ and retrieved the HOL packet of I-VOQ $(m, 32)$. So, the HOL packet in VOQ 32 of input 1 can contend as the HOL packet of I-VOQ $(m, 32)$ successfully at time t if there was no reservation packets in I-VOQ $(m, 32)$ at time $t-1$. In general, suppose output j retrieves the HOL packet of I-VOQ (m, j) at time t. Then, input i can contend as the HOL packet of I-VOQ (m, j) at time $t + (i - j - 1)$mod $N + 1$. Therefore, the contention priority among input VOQs of destination j should decreases with the input indices in the right modulated fashion after j, i.e. input $(j + k)$mod N has higher priority than input $(j+k+1)$mod N for $k = 1 \ldots N-1$. To demonstrate the priority, we show the average delay of packets destined to 32 from input 32 and that from input 1

(CR_$(32,32)$ and CR_$(1,32)$) by Figure 8 and Figure 9. In these figures, we observe that the contention priority appears in the region between $0.63 \leq \rho \leq 0.95$ because the curve of CR_$(32,32)$ is higher than that of CR_$(1,32)$ in this region. As such, the fixed contention priority among flows might be a concern.

We can solve this fairness problem by re-mapping port indices. This technique was proposed to solve the fairness problem for the mailbox switch [5]. There are $N!$ one-to-one and onto mappings from the set $\{1, 2, \ldots, N\}$ to itself. We uniformly select a mapping from those $N!$ mappings. Then, we use that mapping for $CT = 200$ time frames in our simulation experiments. To uniformly select a mapping, we toss $N-1$ fair dice with values $1, 2, \ldots, N-k+1$ for the k^{th} dice. Then we select one value from the remaining unused $N-k+1$ values as the k^{th} value of the permutation mapping. If we utilize enough mappings, by the law of large number, we can eventually equalize the priority orders of all output ports. In order to keep packets in sequence, we need to pause two frames during the transition from one mapping to another. During the pause, the output ports clear the possible HOL contention packets in the central buffers. This pause of sending and receiving packets would result in approximately $2/CT$ throughput loss. In Figure 8 and Figure 9, we simulate for N^2 mappings at each data point. In these figures, the maximum average delay among all flows in the CR switch (CR_remap_max) and the minimum average delay among all flows in the CR switch (CR_remap_min) are very close to each other. Therefore, this fairness problem due to contention priorities can be successfully solved by re-mapping port indices. One can also observe that CR_remap_min is even higher than CR_avg in some data points. This is because of the throughput reduction due to the pause. This throughput reduction, however, can be made as small as possible by setting CT large enough.

There is another fairness problem associated with the CR switch. In a CR switch, a flow of packets delivered mostly in the reservation mode may experience less expected delay than a flow of packets delivered mostly in the contention mode, even if the flow delivered in the contention mode has less arrival intensity. This phenomenon is more likely to happen if the arrival traffic is extremely unbalanced. Suppose that an input port sends most of its traffic to a particular output port. We call such a pair of input and output ports a hotspot flow. Packets generated by a hotspot flow with medium to heavy traffic loads are most likely delivered in the reservation mode. As a result, contention packets from other inputs to the same destination are more likely blocked because reservation packets from the hotspot flow are likely to occupy the HOL positions. The contention packets from other input ports can only use the remaining bandwidth left by reservation packets from the hotspot flow. Therefore, a fairness issue can arise under extremely unbalanced traffic.

To study the fairness issue for extremely unbalanced traffic, we simulate the CR switch equipped with port re-mapping and loaded with the hotspot traffic. We simulate for the total average delay of hotspot flows (CR_remap_hot) ($i = j$) and the total average delay of all other flows (CR_remap_cold) ($i \neq j$) in Figure 10 and Figure 11. As shown in those figures, there is a fairly wide gap between the curve CR_remap_cold and the curve CR_remap_hot. In comparison we also simulate the PF scheme loaded with the hotspot traffic. The results are shown as curves PF_hot and PF_cold in Figure 10 and Figure 11. We can observe that the gap between CR_remap_cold and CR_remap_hot is much smaller than the gap between PF_cold and PF_hot. Thus, the CR switch has a less serious fairness problem than the PF scheme under such traffic. However, the port re-mapping method cannot effectively equalize the average delay of packets delivered in the reservation mode and that of the packets delivered in the contention mode due to *blocking of service* in the reservation mode. Similar fairness problems among flows could also exist in the input-buffered switch under the Maximum Weighted Matching with Longest Queue First (MWM-LQF) algorithm [14]. This is because an input VOQ is more likely to build up when its arrival traffic is bursty and heavy. As the MWM-LQF algorithm assigns the weight of an input VOQ proportional to its queue length, the input VOQ with bursty and heavy traffic will be matched most of the time and result in *blocking of service* for other input VOQs.

VI. Conclusions

In this paper, we proposed a new switch architecture, called the CR switch, that solved the reordering problem in load-balanced switches. This is done without using any resequencing buffer. Also, we showed that the number of packets in the CR switch is bounded by the sum of the number of packets in the corresponding output-buffered switch and a constant that only depends on the size of the switch. As such, the CR switch still achieves 100% throughput.

The key invention of the CR switch is the I-VOQ buffer management scheme that allows the CR switch to be operated in two modes: (i) the contention mode in light traffic and (ii) the reservation mode in heavy traffic. By so doing, the CR switch has low average delay in light traffic and still maintains system stability in heavy traffic. By computer simulations, we also demonstrated that the average packet delay of the CR switch is considerably lower than other schemes in the literature, including the Uniform Frame Spreading scheme [10], the Padded Frame scheme [8] and the mailbox switch [5].Compared with an input-buffered switch executing the iSLIP matching algorithm, the CR switch performs distinctly better under heavy traffic condition. When the traffic has nonuniform destination distributions, the iSLIP algorithm cannot achieve 100% throughput, while the CR switch can. However, the iSLIP switch has a better delay performance under light to medium traffic conditions. By simulation, we have studied two fairness problems of the CR switch. The first fairness problem arises because of the deterministic and periodic TDM connection pattern that the CR switch uses. This connection pattern produces a fixed priority order among inputs for any given output port. We propose a port re-mapping method to solve this fairness problem. In the second fairness problem, we observe that packets transmitted in the contention mode are likely to have longer delays than packets transmitted in the reservation mode.

Finally, we note that there are still some problems and issues that require further study for the CR switch as listed below:

1) Large propagation delay:
   In the CR switch, we need one bit of feedback information from the connected central buffer to indicate whether a transmission is successful or not. There might be a problem if the propagation delay from the connected central buffer to an input is large.
2) Heterogeneous line speeds:
   We assume that the input line speeds are identical. This is a very common assumption in the literature for input-buffered switches and load-balanced switches that require synchronous transmissions. To deal with the case with heterogeneous input line speeds, one common practice is to implement line-grouping, i.e., multiplexing the low speed lines into high speed lines before they go into the CR switch and then de-multiplexing the traffic after leaving the CR switch. The drawback of doing line-grouping is that some bandwidth could be wasted as there could be residual bandwidth left unpacked.
3) Priority services:
   In order to provide quality of service in the CR switch, one might need to consider the problem of providing priority services in the CR switch. A simple and straightforward method is to provide priority services directly in the input VOQs. However, we might not be able to to retain the 100% throughput property by doing that. The problem arises when there does not exist a full-framed VOQ of high priority packets while there are still full-framed VOQs of low priority packets. If we choose to serve the high priority packets in the contention mode, then we will waste some bandwidth and cannot maintain 100% throughput for low priority packets. One tentative solution for this is to set a threshold like the PF scheme in [8], and serve the high priority packets in the contention mode only when the total queue length of full-framed VOQs is below the threshold. However, how to set the threshold to achieve the right tradeoff between high priority packets and low priority packets requires further study.

References

[1] G. Brassard and P. Bratley, "Fundamentals of Algorithmics" New Jersey: Prentice Hall, 1996.

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

[3] 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.

[4] 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.

[5] C.-S. Chang, D.-S. Lee, and Y.-J. Shih, and Chao-Lin Yu, "Mailbox switch: a scalable two-stage switch architecture for conflict resolution of ordered packets," to appear in *IEEE Trans. on Communications*.

[6] H. J. Chao, J. Song, N. S. Artan, G. Hu, and S. Jiang, "Byte-focal: a practical load balanced switch," *IEEE HPSR*, 2005.

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

[8] J.-J. Jaramillo, F. Milan, and R. Srikant, "Padded frames: a novel algorithm for stable scheduling in load balanced switches," *Proceedings of CISS*, Princeton, NJ, March 2006.

[9] M. J. Karol, M. G. Hluchyj, and S. P. Morgan, "Input versus output queueing on a space-division packet switch," *IEEE Transactions on Communications*, Vol. COM 35, NO. 12, Dec. 1987.

[10] I. Keslassy, S.-T. Chuang, K. Yu, D. Miller, M. Horowitz, O. Solgaard, and N. McKeown, "Scaling internet routers using optics," *Proceedings of ACM SIGCOMM*, Karlsruhe, Germany, August 2003.

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

[12] W.E. Leland, M.S. Taqqu, W. Willinger and D.V. Wilson, "On the self-similar Nature of Ethernet Traffic," *IEEE/ACM Trans. on Networking*, Vol. 2, pp. 1-15, 1994.

[13] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches " *IEEE/ACM Transactions on Networking*, Vol. 7, pp. 188-201, 1999.

[14] N. McKeown, A. Mekkittikul, V. Anantharam, J. Walrand, "Achieving 100% Throughput in an Input-Queued Switch" *IEEE TRANSACTIONS ON COMMUNICATIONS*, VOL. 47, NO. 8, pp. 1260-1267, AUGUST 1999

[15] J. Turner, "Strong Performance Guarantees for Asynchronous Crossbar Schedulers," *Proceedings of IEEE INFOCOM*, Barcelona, April 2006.

[16] F. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part III –Polling and (Dynamic) Split-Channel Reservation Multiple Access", *IEEE Transactions on Communications*, Vol. 24, No. 8, pp. 832-844, August 1976.

[17] C.-Y. Tu, C.-S. Chang, D.-S. Lee, and C.-T. Chiu, "Design a simple and high performance switch using a two stage switch architecture," *Proceedings of IEEE Globecom*, 2005.

Appendix

A. Proof of Proposition 2

We prove (i) and (ii) simultaneously by induction on time. Suppose that Proposition 2(i) and (ii) are true up to time t as the induction hypothesis. Without loss of generality, assume that t is the m^{th} slot of frame f of input i. Moreover, frame f is a reservation frame that contains packets for output j. As such, a packet, say packet h, is transmitted to I-VOQ (m, j) at time t and stored as the p^{th} packet for some $p \geq 2$. As a reservation packet is always attached to the tail of an I-VOQ, to prove Proposition 2(i), it suffices to argue that before transmitting another packet from input i at $t+1$, the queue length of I-VOQ $(m+1, j)$ is exactly $p-1$.

We first show that the queue length is at least $p-1$. If $p = 2$, nothing needs to be proved as an I-VOQ contains at least one packet. For $p \geq 3$, it suffices to show that the $(p-1)^{th}$ packet of I-VOQ $(m+1, j)$ exists at $t+1$. Since packet h is stored as the p^{th} packet in I-VOQ (m, j) at time t, the $(p-1)^{th}$ packet of I-VOQ (m, j), say packet h_1, exists at time t. From Proposition 1(ii), packet h_1 will depart the switch at time $t + w_{p-1}$. As $p \geq 3$, packet h_1 must be a reservation packet and it is transmitted to I-VOQ (m, j) at some time $t_1 < t$ from some input i_1. As reservation packets are transmitted consecutively in each reservation frame, there is another reservation packet, say packet h_2, transmitted from input i_1 to I-VOQ $(m+1, j)$ at time $t_1+1 \leq t$. Thus from (ii) in the induction hypothesis, packet h_2 will depart the switch at time $t + w_{p-1} + 1$. From Proposition 1(ii), the $(p-1)^{th}$ packet in I-VOQ $(m+1, j)$ at time $t+1$ will depart the switch at time $t+1+w_{p-1}$ which is the same as packet h_2. Thus, packet h_2 exists as the $(p-1)^{th}$ packet of I-VOQ $(m+1, j)$ at time $t+1$. Thus there are at least $p-1$ packets.

Now we show that the queue length is at most $p-1$. Suppose that the p^{th} packet, say packet h_3, exists in I-VOQ $(m+1, j)$ at $t+1$. Then packet h_3 is a reservation packet transmitted to I-VOQ $(m+1, j)$ before $t+1$. Also, from

Proposition 1(i), packet h_3 will depart the switch at time $t{+}1{+}w_p$. As reservation packets are transmitted consecutively in a reservation frame, there is another packet, say packet h_4, transmitted from the input of packet h_3 to I-VOQ (m, j) before t. From (ii) in the induction hypothesis, packet h_4 will depart the switch at time $t{+}w_p$. As argued in the previous paragraph, packet h_4 is the p^{th} packet in I-VOQ (m, j) at time t. This contradicts to the assumption that packet h is stored as the p^{th} packet of I-VOQ (m, j) at time t.

The induction of Proposition 2(ii) at time $t + 1$ follows directly the inducted result of Proposition 2(i) at time $t + 1$ and Proposition 1(ii).

B. Proof of Lemma 11

Let $A_1(t)$ (*resp.* $A_2(t)$, $A_3(t)$) be the cumulative number of packets arriving at the CR switch (*resp.* the central buffers, Q_j) for output j by time t. Let $Q^R(t)$ be the number of packets in Q_j at time t. Also, let $Q^O_g(t)$ be the number of packets in output buffer j of the output-buffered switch at time t when the arrival process is $A_g(t)$, for $g = 1, 2, 3$.

Note that an output-buffered switch is work conserving. Thus, we have the following Lindley's equation:

$$Q^O_g(t{+}1) = \max\{Q^O_g(t) + [A_g(t{+}1) - A_g(t)] - 1, 0\}, \quad (13)$$

for $g = 1, 2, 3$. From Section 1.3 in [2], these Lindley's equations can be expanded recursively to the following forms.

$$Q^O_g(t) = \max_{0 \le s \le t}\{[A_g(t) - A_g(s)] - (t - s)\}, \quad (14)$$

for $g = 1, 2, 3$.

As we have shown in Proposition 10 that Q_j is work conserving with response workload 1 and response delay $N - 1$, it follows from Lemma 6 that

$$Q^R(t) \le Q^O_3(t) + N. \quad (15)$$

Since the packets arriving at Q_j are all reservation packets, they are only a subset of the packets arriving at the central buffers. This implies that $A_3(t) - A_3(s) \le A_2(t) - A_2(s)$, for all $s \le t$. Along with (14), we have

$$Q^O_3(t) \le Q^O_2(t). \quad (16)$$

As the packets arriving at the central buffers are the packets departing from the input buffers, we have

$$A_2(t) \le A_1(t). \quad (17)$$

Also, Let $Q^I(t)$ be the number of packets destined to output j stored in the input buffers of the CR switch at time t. Then, we have

$$Q^I(t) = A_1(t) - A_2(t). \quad (18)$$

Since the number of packets in an input buffer cannot exceed N^2 (see Corollary 8), we have $Q^I(t) \le N \cdot N^2$. This leads to

$$A_1(t) \le A_2(t) + N^3. \quad (19)$$

Using (19) and (17) in (14), we have

$$Q^O_2(t) \le Q^O_1(t) + N^3. \quad (20)$$

Finally, (15) together with (16) and (20) implies

$$Q^R(t) \le Q^O_1(t) + N^3 + N. \quad (21)$$

This completes the proof.

Chao-Lin Yu Chao-Lin Yu (S'04) received the B.S. degree in Electrical Engineering from National Tsing Hua University (NTHU), Hsinchu, Taiwan, in 2001. He is pursuing his PhD degree in NTHU, focusing on computer communications. He visited IBM Thomas J. Watson research center, Hawthorne, N. Y. during the summer in 2001. He visited IBM T. J. Watson research center again from Nov. 2006 to Nov. 2007.

Cheng-Shang Chang Cheng-Shang Chang (S'85-M'86-M'89-SM'93-F'04) received the B.S. degree from the National Taiwan University, Taipei, Taiwan, in 1983, and the M.S. and Ph.D. degrees from Columbia University, New York, NY, in 1986 and 1989, respectively, all in Electrical Engineering. From 1989 to 1993, he was employed as a Research Staff Member at the IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y. Since 1993, he has been with the Department of Electrical Engineering at National Tsing Hua University, Taiwan, R.O.C., where he is a Professor. His current research interests are concerned with high speed switching, communication network theory, and mathematical modeling of the Internet. Dr. Chang received an IBM Outstanding Innovation Award in 1992, an IBM Faculty Partnership Award in 2001, and Outstanding Research Awards from the National Science Council, Taiwan, in 1998, 2000 and 2002, respectively. He also received Outstanding Teaching Awards from both the college of EECS and the university itself in 2003. He was appointed as the first Y. Z. Hsu Scientific Chair Professor in 2002. He is the author of the book "Performance Guarantees in Communication Networks," and he served as an editor for Operations Research from 1992 to 1999. Dr. Chang is a member of IFIP Working Group 7.3.

Duan-Shin Lee Duan-Shin Lee (S'89-M'90-SM'98) received the B.S. degree from National Tsing Hua University, Taiwan, in 1983, and the MS and Ph.D. degrees from Columbia University, New York, in 1987 and 1990, all in electrical engineering. He worked as a research staff member at the C&C Research Laboratory of NEC USA, Inc. in Princeton, New Jersey from 1990 to 1998. He joined the Department of Computer Science of National Tsing Hua University in Hsinchu, Taiwan, in 1998. Since August 2003, he has been a professor. He received a best paper award from the Y.Z. Hsu Foundation in 2006. His research interests are switch and router design, wireless networks, performance analysis of communication networks and queueing theory. He is a senior IEEE member.