Using a Single Switch with $O(M)$ Inputs/Outputs for the Construction of an Optical Priority Queue with $O(M^3)$ buffer

Hsien-Chen Chiu, Cheng-Shang Chang, Jay Cheng, and Duan-Shin Lee
Institute of Communications Engineering
National Tsing Hua University
Hsinchu 30013, Taiwan, R.O.C.
Email: hcchiu@gibbs.ee.nthu.edu.tw
cschang@ee.nthu.edu.tw
jcheng@ee.nthu.edu.tw
lds@cs.nthu.edu.tw

***Abstract*— In this paper, we consider the construction of an optical priority queue with a single $(M+1)\times(M+1)$ switch and M fiber delay lines. The M fiber delay lines are connected from M outputs of the switch back to M inputs of the switch, leaving one input (resp. output) of the switch for the input (resp. output) of the priority queue. It was known in [8][9] that with an appropriate choice of the lengths of the delay lines, such a construction can be used for exact emulation of an optical priority queue with $O(M^2)$ buffer size. In this paper, we show that the buffer size can be further extended to $O(M^3)$ using the same construction. The improvement relies on establishing a *partial ordering* for all the packets stored in the delay lines.**

I. Introduction

One of the main problems for optical packet switching is the lack of optical buffers. As optical packets, composed of a train of photons, cannot be easily stopped, stored, and forwarded, the only known way to construct optical buffers is to direct optical packets through a set of optical Switches and fiber Delay Lines (SDL) so that optical packets come out at the right place and at the right time. Recent advances in the SDL constructions have shown various interesting results, including first-in-first-out (FIFO) multiplexers in [1]–[5], FIFO queues in [6], linear compressors, non-overtaking delay lines, and flexible delay lines in [7], and priority queues in [8][9].

In this paper, we focus on the constructions of optical priority queues. In a priority queue, every packet is associated with a label, called priority. The packet with the highest priority is always the next one to depart. Both FIFO queues and LIFO queues are special cases of priority queues as one can simply use the arrival time of a packet as its priority. As such, the construction of an optical priority queue is much more difficult than that of an optical FIFO/LIFO queue.

The first construction of an optical priority queue was proposed by Sarwate and Anantharam [8]. In [8], they considered a feedback system as shown in Figure 1. In such a feedback system, there is an $(M+1)\times(M+1)$ crossbar switch and M fiber delay lines with delays d_i, $i=1,2,\ldots,M$. If $M=2k-1$ for some positive integer k, $d_i=i$ for $i=1,\ldots,k$, and $d_i=1$ for $i=k+1,\ldots,2k-1$, then it was shown in [8] that such a system can be used for exact emulation of a priority queue with buffer $\sum_{i=1}^{k} d_i$. The proof in [8] is quite elaborated. A simpler proof was provided in [9]. The key idea of the approach in [9] was to use a sorter to sort packets according to their priorities. By so doing, there is a total order for the packets at the outputs of the sorter, and it can then be used for delaying packets before their departures. It was further shown in [9] that if $M=2k-1$, then one can choose $d_i=i$ for $i=1,\ldots,k$ and $d_i=2k-i$ for $i=k+1,\ldots,2k-1$ for exact emulation of an optical queue with buffer $\sum_{i=1}^{M} d_i$.

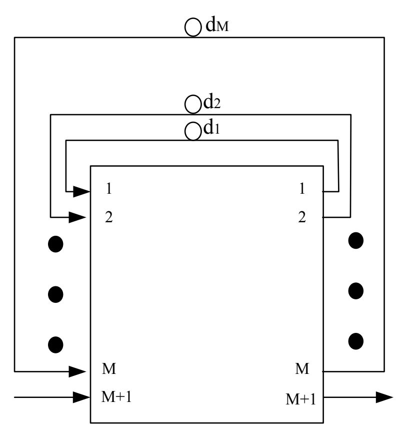

Fig. 1. A construction of a priority queue via a single switch and fiber delay lines.

Both the constructions in [8] and [9] showed that one can construct an optical priority queue with $O(M^2)$ buffer size using the feedback system in Figure 1. Our main contribution in this paper is to show that the buffer size can in fact be extended to $O(M^3)$ using the same feedback system. Specifically, we show that if for some $0\le m\le\lceil M/2\rceil$, we choose $d_i = d_{M+1-i} = i$ for all $i = 1, 2, \dots, m$, and

$$m \le d_i = d_{M+1-i} \le i + \sum_{j=2}^{m} \lceil ((i - M + 2m - 4j + 1)/2)^+ \rceil$$

for all $i = m+1, \dots, \lceil M/2 \rceil$, then the feedback system in Figure 1 can be operated as a priority queue with buffer $\sum_{i=1}^{M} d_i$. The improvement relies on establishing a *partial ordering* for all the packets stored in the buffer. The partial ordering is derived from the *total ordering* for the packets at the outputs of the sorter in [9] and the fact that the relative order for these packets (at the outputs of the sorter) is preserved as time evolves. By numerical computation, we find that the optimal choice of m to maximize the buffer size is approximately $0.433M$ for large M. For such a choice, the buffer size is roughly $0.000929M^3$ for large M. These are further verified by approximating sums by integrals.

The paper is organized as follows. In Section II, we introduce the definitions of priority queues and the constructions of priority queues in [9]. We then present our main results in Section III and the proofs in Section IV. The paper is concluded in Section V, where we summarize our results.

II. Review of Known Results

In this section, we give a brief review of known results.

A. Priority Queues and Complementary Priority Queues

We first introduce some basic assumptions and concepts for Switches and fiber Delay Lines (SDL).

(i) Packets are of the same size.

(ii) Time in all our optical links is slotted and synchronized so that a packet can be transmitted within a time slot.

(iii) A fiber delay line with delay d is an optical link that requires d time slots for a packet to traverse through.

(iv) An $M \times M$ crossbar switch is a network element with M input links and M output links that realizes all the $M!$ permutations between its inputs and outputs.

In the following, we introduce the definition of a (discrete-time) priority queue with buffer B in [9].

Definition 1 (Priority queues [9]) *A priority queue with buffer B is a network element that has one input link, one control input link, and two output links (see Figure 2). One output link is for departing packets and the other is for lost packets. When a packet arrives at the queue, it is associated with a label, called* priority. *We assume that there is a total order for the priorities of all the packets. As shown in Figure 2, let $c(t)$ be the state of the control input at time t. When $c(t) = 1$, we say the priority queue is enabled at time t. On the other hand, the priority queue is disabled at time t if $c(t) = 0$. Also, let $a(t)$ be the set of the packet arriving at time t (if any[1]), $d(t)$ be the set of the packet departing at time t (if any), $\ell(t)$ be the set of the lost packet at time t (if any), and $q(t)$ be the set of packets queued at the priority queue at time t (at the end of the t^{th} time slot). Then the priority queue with buffer B satisfies the following five properties:*

(P1) *Flow conservation: arriving packets from the input link are either stored in the buffer or transmitted through the two output links, i.e.,*

$$q(t) = (q(t-1) \cup a(t)) \backslash (d(t) \cup \ell(t)). \tag{1}$$

(P2) *Non-idling: if the control input is enabled, i.e., $c(t) = 1$, then there is always a departing packet if there are packets in the buffer or there is an arriving packet, i.e.,*

$$|d(t)| = \begin{cases} 1, & \text{if } c(t) = 1 \text{ and } |q(t-1) \cup a(t)| > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

(P3) *Maximum buffer usage: if the control input is not enabled, i.e., $c(t) = 0$, then there is a lost packet only when buffer is full and there is an arriving packet, i.e.,*

$$|\ell(t)| = \begin{cases} 1, & \text{if } c(t) = 0 \text{ and } |q(t-1) \cup a(t)| > B, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

(P4) *Priority departure: if there is a departing packet at time t, the departing packet is the one with the highest priority among all the packets in $q(t-1) \cup a(t)$.*

(P5) *Priority loss: if there is a lost packet at time t, the lost packet is the one with the lowest priority among all the packets in $q(t-1) \cup a(t)$.*

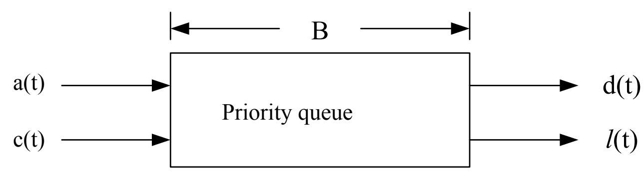

Fig. 2. A priority queue with buffer B.

The following concept of a complementary priority queue was introduced in [9] that reduces the five properties for a priority queue into two simple properties. As such, it is much easier to verify a construction of a complementary priority queue.

Definition 2 (Complementary priority queues [9]) *A complementary priority queue with buffer B is a network element that has one input link, one control input link, and one output link (see Figure 3). As in a priority queue, every packet is associated with a label, called* priority, *and there is a total order for the priorities. At time 0, there are B packets stored in the network element. Unlike a priority queue, there is always an arriving packet and a departing packet in every time slot. As shown in Figure 3, let $c(t)$ be the state of the control input,*

[1] This means that $a(t)$ is an empty set if there is no packet arriving at time t, and is a singleton otherwise.

$a(t)$ be the set of the packet arriving at time t, $b(t)$ be the set of the packet departing at time t, and $q^c(t)$ be the set of packets queued at the complementary priority queue at time t (at the end of the t^{th} time slot). Then the complementary priority queue with buffer B satisfies the following two properties:

(C1) *Flow conservation: arriving packets from the input link are either stored in the buffer or transmitted through the the output link, i.e.,*

$$q^c(t) = (q^c(t-1) \cup a(t)) \backslash b(t). \quad (4)$$

(C2) *Complementary priority departure: if $c(t) = 1$, then the departing packet is the one with the* highest *priority among all the packets in $q^c(t-1) \cup a(t)$. On the other hand, if $c(t) = 0$, then the departing packet is the one with the* lowest *priority among all the packets in $q^c(t-1) \cup a(t)$.*

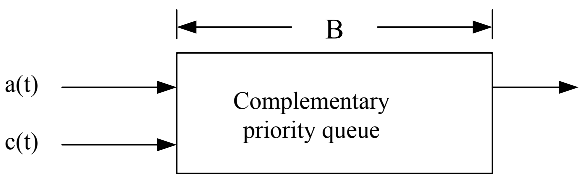

Fig. 3. A complementary priority queue.

It was shown in [9] that a priority queue with buffer B can be constructed by a concatenation of a complementary priority queue with buffer B and a 1×2 switch (see Figure 4). The key idea in [9] was to view empty time slots as *fictitious* packets that have priorities lower than those of real packets. Moreover, the priorities among the fictitious packets are decreasing in the order of their arrival times. As such, there is a total order among all the packets, including both the real packets and the fictitious packets. At time 0, there are B fictitious packets stored in the complementary priority queue. To emulate a priority queue, the input of the 1×2 switch in Figure 4 is connected to $d(t)$ (resp. $\ell(t)$) when $c(t) = 1$ (resp. $c(t) = 0$).

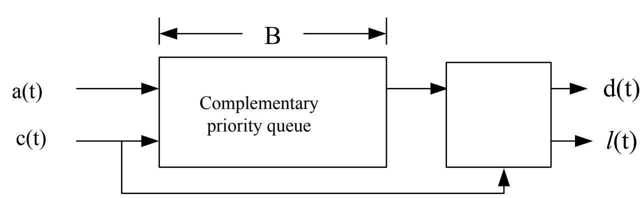

Fig. 4. A construction of a priority queue with buffer B via a concatenation of a complementary priority queue with buffer B and a 1×2 switch.

B. A Construction of a Complementary Priority Queue

Now we introduce the construction of a complementary priority queue with buffer $\sum_{i=1}^{M} d_i$ in [9]. This is also the construction that we will use in this paper. In Figure 5, there are two $(M+1) \times (M+1)$ crossbar switches: a sorter (on the left hand side) and a shifter (on the right hand side). The M outputs of the shifter, indexed from $i = 1, 2, \ldots, M$, are

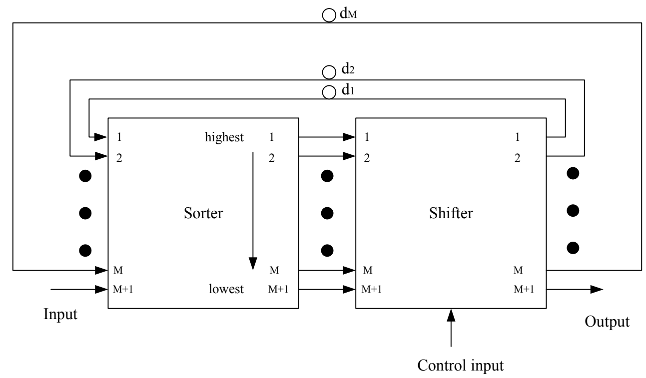

Fig. 5. A construction of a complementary priority queue with buffer $\sum_{i=1}^{M} d_i$.

connected back to the the corresponding M inputs of the sorter via M fiber delay lines with delays d_i, $i = 1, 2, \ldots, M$. For a fiber delay line with delay d, there are at most d packets stored in that delay line. As such, there are at most $\sum_{i=1}^{M} d_i$ packets stored in the M fiber delay lines. It was shown in [9] if one chooses

$$0 < d_i \leq \min[i, M+1-i] \quad (5)$$

for all $i = 1, 2, \ldots, M$, then the construction in Figure 5 can be operated as a complementary priority queue with buffer $\sum_{i=1}^{M} d_i$. For $M = 2k-1$, the maximum buffer size that can be achieved by (5) is to set $d_i = i$ for $i = 1, 2, \ldots, k$, and $d_i = 2k - i$ for $i = k+1, k+2, \ldots, 2k-1$. For this, one has buffer size $\sum_{i=1}^{2k-1} d_i = k^2$. And for $M = 2k$, the maximum buffer size that can be achieved by (5) is to set $d_i = i$ for $i = 1, 2, \ldots, k$, and $d_i = 2k + 1 - i$ for $i = k+1, k+2, \ldots, 2k$. For this, one has buffer size $\sum_{i=1}^{2k} d_i = k^2 + k$. As one can combine the two crossbar switches into a single one, the condition in (5) implies that one can construct a priority queue with $O(M^2)$ buffer size via a single $(M+1) \times (M+1)$ switch.

To emulate a complementary priority queue, the construction in Figure 5 is operated in a way so that the following assumption is satisfied all the time:

(A1) All the packets stored in all the fiber delay lines in Figure 5 cannot be either the packet with the highest priority or the packet with the lowest priority until they appear at the inputs of the sorter.

From (A1), the packets that appear at the inputs of the sorter contain both the packet with the highest priority and the packet with the lowest priority. The function of the sorter on the left hand side is to sort the packets at the $M+1$ inputs (in the order of their priorities) so that the packet with the highest priority appears at the first output and the packet with the lowest priority appears at the $(M+1)^{th}$ output.

The function of the shifter on the right hand side is then to direct the packet with the lowest (resp. highest) priority to the output when $c(t) = 0$ (resp. $c(t) = 1$), and keep the remaining M packets in decreasing order of their priorities. For this, its connection pattern is realized by the $(M+1) \times (M+1)$ identity matrix when $c(t)=0$. On the other hand, when $c(t)=1$, its connection pattern is realized by the $(M+1)\times(M+1)$ circular-shift matrix, i.e., the matrix $P=(P_{ij})$ with $P_{i,j}=1$ for $i=(j \bmod (M+1))+1$ and $P_{i,j}=0$ otherwise.

Consider a packet at the the i^{th} output of the shifter and it is about to enter the i^{th} delay line. Call this packet the tagged packet. A sufficient condition for (A1) to hold is as follows:

(A2) There are at least d_i-1 packets that have priorities *higher* than the priority of the tagged packet, and there are at least d_i-1 packets that have priorities *lower* than the priority of the tagged packet.

This is because there is exactly one departure in a time slot from a complementary priority queue. As such, the tagged packet cannot be either the packet with the highest priority or the packet with the lowest priority until it appears at the input of the sorter if the condition in (A2) holds.

To see that the condition in (A2) holds under (5), note that the priorities of the packets at the M outputs of the shifter, indexed from $1,2,\ldots,M$, are decreasing. As such, there are $i-1$ (resp. $M-i$) packets that have priority higher (resp. lower) than the priority of the tagged packet. Thus, there are at least $\min[i,M+1-i]-1$ packets that have priority higher (resp. lower) than the priority of the tagged packet.

III. Main Results

We have known that the construction in Figure 5 can be used as a priority queue with $O(M^2)$ buffer size. This is still far from the exponential upper bound $O(2^M)$ in [8]. The question is whether it is possible to further increase the buffer size under the same construction in Figure 5. The answer to the question is affirmative. In this section, we show that one can increase the buffer size from $O(M^2)$ to $O(M^3)$. The result is stated in the following theorem and its proof is deferred to the next section.

Theorem 3 *For some $0\le m\le\lceil M/2\rceil$, if we keep $d_i=d_{M+1-i}=i$ for all $i=1,2,\ldots,m$, and*

$$\begin{aligned} m \le d_i &= d_{M+1-i} \\ &\le i+\sum_{j=2}^{m}\lceil((i-M+2m-4j+1)/2)^+\rceil \end{aligned} \qquad (6)$$

for all $i=m+1,\ldots,\lceil M/2\rceil$, then the construction in Figure 5 is still a complementary priority queue with buffer $\sum_{i=1}^{M} d_i$.

To see that $\sum_{i=1}^{M} d_i$ in Theorem 3 is $O(M^3)$, we may simply choose $m=\lceil M/3\rceil$ and

$$d_i=d_{M+1-i}=i+\sum_{j=2}^{m}\lceil((i-M+2m-4j+1)/2)^+\rceil$$

for all $i=m+1,\ldots,\lceil M/2\rceil$. Then

$$\begin{aligned} &\sum_{i=1}^{M} d_i \\ \ge& \sum_{i=\lceil M/3\rceil+1}^{\lceil M/2\rceil} d_i \\ \ge& \sum_{i=\lceil M/3\rceil+1}^{\lceil M/2\rceil}\sum_{j=2}^{\lceil M/3\rceil}\lceil((i-M+2\lceil M/3\rceil-4j+1)/2)^+\rceil \\ \ge& \sum_{i=\lceil 3M/8\rceil+1}^{\lceil M/2\rceil}\sum_{j=2}^{\lfloor M/192\rfloor}\lceil((i-M+2\lceil M/3\rceil-4j+1)/2)^+\rceil \\ \ge& \sum_{i=\lceil 3M/8\rceil+1}^{\lceil M/2\rceil}\sum_{j=2}^{\lfloor M/192\rfloor}\lceil((\lceil 3M/8\rceil+1-M \\ & \qquad +2\lceil M/3\rceil-4\lfloor M/192\rfloor+1)/2)^+\rceil \\ \ge& \sum_{i=\lceil 3M/8\rceil+1}^{\lceil M/2\rceil}\sum_{j=2}^{\lceil M/192\rceil}\lceil M/96\rceil. \end{aligned}$$

This shows that $\sum_{i=1}^{M} d_i$ is at least $O(M^3)$ for such a choice.

In Table I, we numerically compute the buffer size in Theorem 3 by choosing the optimal m. In the second column, we list the buffer size obtained by (5). The increment of the buffer size using Theorem 3 is shown in the third column and the optimal choice of m is shown in the fourth column. It is interesting to see from this table that the optimal choice m is roughly $0.433M$ for large M. To see this, note that the increment of the buffer size using Theorem 3 is

$$2\sum_{i=m+1}^{M/2}\sum_{j=2}^{m}\lceil((i-M+2m-4j+1)/2)^+\rceil \qquad (7)$$

when M is even. For large M, if we replace m by αM, j/M by x, and i/M by y, the double sum in (7) can be approximated by the following double integral

$$\begin{aligned} &M^3\int_{\alpha}^{1/2}\int_{0}^{\alpha}(y-1+2\alpha-4x)^+dxdy \\ &=M^3\int_{\alpha}^{1/2}\int_{0}^{(y-1+2\alpha)/4}(y-1+2\alpha-4x)dxdy. \end{aligned} \qquad (8)$$

With $z=y-1+2\alpha$, the integral in (8) can be simplified as follows:

$$\frac{M^3}{8}\int_{3\alpha-1}^{2\alpha-1/2}z^2dz=\frac{M^3}{24}((2\alpha-1/2)^3-(3\alpha-1)^3). \qquad (9)$$

The optimal α that maximizes (9) for α in [0,1] is

$$\frac{14+\sqrt{6}}{38}\approx 0.433. \qquad (10)$$

Using the optimal α in (10) for (9), we show that the maximum increment of the buffer size using Theorem 3 is roughly $0.000929M^3$ for large M.

M	Buffer size by (5)	Increment by using Theorem 3	Optimal m
127	$4096 = (\frac{M+1}{2})^2$	$1563 \approx 0.000763M^3$	$56 \approx 0.441M$
255	$16384 = (\frac{M+1}{2})^2$	$13994 \approx 0.000844M^3$	$111 \approx 0.435M$
511	$65536 = (\frac{M+1}{2})^2$	$118236 \approx 0.000886M^3$	$222 \approx 0.434M$
1023	$262144 = (\frac{M+1}{2})^2$	$971436 \approx 0.000907M^3$	$443 \approx 0.433M$
2047	$1048576 = (\frac{M+1}{2})^2$	$7875590 \approx 0.000918M^3$	$887 \approx 0.433M$
4095	$4194304 = (\frac{M+1}{2})^2$	$63423364 \approx 0.000924M^3$	$1773 \approx 0.433M$
8191	$16777216 = (\frac{M+1}{2})^2$	$509065684 \approx 0.000926M^3$	$3546 \approx 0.433M$
16383	$67108864 = (\frac{M+1}{2})^2$	$4079247391 \approx 0.000928M^3$	$7092 \approx 0.433M$

TABLE I

INCREMENT OF BUFFER SIZE BY THE OPTIMAL CHOICE IN THEOREM 3.

IV. PROOF OF THEOREM 3

As discussed in Section II-B, it suffices to show that the condition in (A2) holds for all time.

Our proof for Theorem 3 relies on establishing a *partial ordering* among all the packets stored in the buffer. In Section II-B, there is a total order for the M packets at the outputs of the sorter, and this was used in [9] to derive (5). By using the partial ordering among all the packets stored in the buffer, we can obtain better bounds for the number of packets with priorities higher (or lower) than that of the packet entering the i^{th} delay line for $i = m+1, \ldots, \lceil M/2 \rceil$. To establish the partial ordering, we view a fiber delay line with delay d as a "sequential" buffer that consists of d cells with each cell capable of holding a packet. We index the cells from the *input* of a fiber delay line. Specifically, the $(i,j)^{th}$ cell, $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, d_i$, (see Figure 6 for an illustration) is the j^{th} cell (from the input) of the i^{th} fiber delay line in Figure 5. As a fiber delay line is a "sequential" buffer, a packet entering the i^{th} delay line at time t will be stored in the $(i,j)^{th}$ cell at time $t+j-1$.

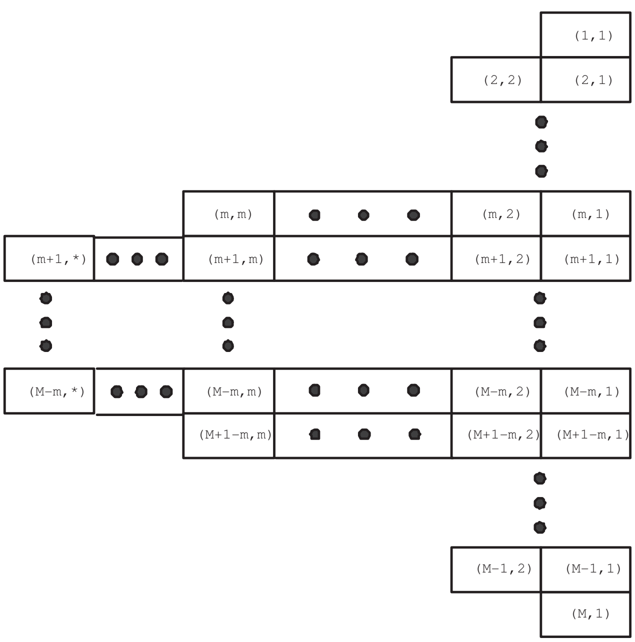

Fig. 6. The cells of the delay lines.

Now consider a packet enters the ${i_p}^{th}$ delay line at some time t_p. Call this packet the tagged packet. We would like to find a bound on the number of packets that have priorities higher than or equal to that of the tagged packet at any time t. For this, we let $p_{i,j}(t) = 1$ if the priority of the packet in the $(i,j)^{th}$ cell at time t is higher than or equal to the priority of the tagged packet and $p_{i,j}(t) = 0$ otherwise.

According to the operation rule of the complementary priority queue, we know that the priorities of the M packets at the inputs of the M fiber delay lines are sorted according to their priorities. As such, we have for all t that

$$p_{1,1}(t) \geq p_{2,1}(t) \geq \cdots \geq p_{M,1}(t). \tag{11}$$

Since the packet in the $(i,j)^{th}$ cell at time t is also in the $(i,j-1)^{th}$ cell at time $t-1$, we have $p_{i,j}(t) = p_{i,j-1}(t-1)$, and in general

$$p_{i,j}(t) = p_{i,j-1}(t-1) = \cdots = p_{i,1}(t-(j-1)). \tag{12}$$

As we assume that $d_j = j$ for all $j = 1, 2\ldots, m$, using (12) and (11), we have for all $j = 1, 2, \ldots, m$ that

$$p_{j,j}(t) \geq p_{j+1,j}(t) \geq \cdots \geq p_{M+1-j,j}(t). \tag{13}$$

The inequalities in (13) form the base of the partial ordering that we need in the proof of Theorem 3.

For $j = 1, 2, \ldots, m$, let

$$p_j(t) = \sum_{i=j}^{M+1-j} p_{i,j}(t) \tag{14}$$

be the total number of packets in the cells of the j^{th} column in Figure 6 that have priorities higher than or equal to that of the tagged packet at time t. Note that the definition for $p_j(t)$, $j = 1, 2, \ldots, m$, is *feasible* because we assume that $d_i = i$ for $i = 1, 2, \ldots, m$, and $d_i \geq m$ for $i = m+1, \ldots, M+1-(m+1)$. As $p_{i,j}(t)$ only has two values, i.e., 0 and 1, we have the following inequalities:

$$0 \leq p_j(t) \leq M - 2(j-1), \quad j = 1, 2, \ldots, m. \tag{15}$$

In the following lemma, we derive an upper bound on $p_{j-1}(t-1)$ in terms of $p_j(t)$.

Lemma 4 *For $j = 2, \ldots, m$, if $p_j(t) < M - 2(j-1)$, then*

$$p_{j-1}(t-1) \leq p_j(t) + 1. \tag{16}$$

Proof. Since $p_j(t) = \sum_{i=j}^{M+1-j} p_{i,j}(t)$ for $j = 1, 2, \ldots, m$, we have from (12) that

$$\begin{aligned} p_j(t) &= \sum_{i=j}^{M+1-j} p_{i,j}(t) = \sum_{i=j}^{M+1-j} p_{i,j-1}(t-1) \\ &= \sum_{i=j-1}^{M+1-(j-1)} p_{i,j-1}(t-1) - p_{j-1,j-1}(t-1) \\ &\quad - p_{M+1-(j-1),j-1}(t-1) \\ &= p_{j-1}(t-1) - p_{j-1,j-1}(t-1) \\ &\quad - p_{M+1-(j-1),j-1}(t-1). \end{aligned}$$

Thus,

$$p_{j-1}(t-1) \quad (17)$$
$$= p_j(t) + p_{j-1,j-1}(t-1) + p_{M+1-(j-1),j-1}(t-1).$$

Now we show (16). If $p_j(t) < M-2(j-1)$, then $p_{i,j}(t) = 0$ for some i. From (13), we have $p_{M+1-j,j}(t) = 0$. By (12), we also have $p_{M+1-j,j-1}(t-1) = 0$. Using (13) again yields $p_{M+1-(j-1),j-1}(t-1) = 0$. Thus, we have from (17) that for $p_j(t) < M - 2(j-1)$

$$p_{j-1}(t-1) = p_j(t) + p_{j-1,j-1}(t-1) \le p_j(t) + 1.$$

The proof is completed. ■

In the following lemma, we derive some key inequalities that will be used in the proof of Theorem 3.

Lemma 5 *Let* $x(t) = \sum_{i=1}^{m} p_{i,i}(t)$.
(i) $x(t) \le x(t-1) + 1$.
(ii) $x(t-1) - 1 \le p_1(t) \le 2x(t-1) + M + 1 - 2m$.
(iii) $x(t-1) \le p_j(t) + 2j - 1$ *for* $j = 2, 3, \ldots, m$.

Proof. (i) Using the fact that $p_{i,i}(t) = p_{i,i-1}(t-1)$ in (12) and the partial ordering $p_{i-1,i-1}(t-1) \ge p_{i,i-1}(t-1)$ in (13), we have that

$$\begin{aligned} x(t) &= \sum_{i=1}^{m} p_{i,i}(t) = p_{1,1}(t) + \sum_{i=2}^{m} p_{i,i}(t) \\ &= p_{1,1}(t) + \sum_{i=2}^{m} p_{i,i-1}(t-1) \\ &\le p_{1,1}(t) + \sum_{i=2}^{m} p_{i-1,i-1}(t-1) \\ &= p_{1,1}(t) + \sum_{i=1}^{m-1} p_{i,i}(t-1) \\ &\le p_{1,1}(t) + \sum_{i=1}^{m} p_{i,i}(t-1) \\ &\le 1 + x(t-1). \end{aligned}$$

(ii) Note that $p_1(t) = \sum_{i=1}^{M} p_{i,1}(t)$ is the total number of packets in the cells $(1,1)$, $(2,1), \ldots, (M,1)$ at time t that have priorities higher than or equal to that of the tagged packet. These M packets can only come from the arriving packet at time t and those packets stored at time $t-1$ in the cells (i,i) and $(M+1-i,i)$ for $i = 1, 2, \ldots, m$, and (i, d_i) for $i = m+1, \ldots, M+1-(m+1)$. Thus,

$$\begin{aligned} p_1(t) &\le \sum_{i=1}^{m} (p_{i,i}(t-1) + p_{M+1-i,i}(t-1)) \\ &\quad + \sum_{i=m+1}^{M+1-(m+1)} p_{i,d_i}(t-1) + 1 \\ &\le \sum_{i=1}^{m} (p_{i,i}(t-1) + p_{i,M+1-i}(t-1)) + M + 1 - 2m. \end{aligned} \quad (18)$$

Note from the partial ordering in (13) that

$$p_{i,M+1-i}(t-1) \le p_{i,i}(t-1)$$

for $i = 1, 2, \ldots, m$. Thus, we have from (18) that

$$\begin{aligned} p_1(t) &\le 2 \sum_{i=1}^{m} p_{i,i}(t-1) + M + 1 - 2m \\ &\le 2x(t-1) + M + 1 - 2m. \end{aligned}$$

This proves the upper bound.

To see the lower bound, note that there is at most one departure in every time slot. Thus,

$$\begin{aligned} p_1(t) &\ge \sum_{i=1}^{m} (p_{i,i}(t-1) + p_{M+1-i,i}(t-1)) \\ &\quad + \sum_{i=m+1}^{M+1-(m+1)} p_{i,d_i}(t-1) - 1 \\ &\ge \sum_{i=1}^{m} p_{i,i}(t-1) - 1 = x(t-1) - 1. \end{aligned}$$

(iii) Note that $x(t-1) = \sum_{i=1}^{m} p_{i,i}(t-1) \le m$. If

$$p_j(t) = M - 2(j-1),$$

then

$$p_j(t) + 2j - 1 = M + 1 \ge m \ge x(t-1).$$

Thus, it suffices to consider the case that

$$p_j(t) < M - 2(j-1).$$

In this case, we have from (16) that

$$\begin{aligned} p_{j-1}(t-1) &\le p_j(t) + 1 \\ &< M - 2(j-1) + 1 \\ &< M - 2(j-2). \end{aligned}$$

As such, we can apply (16) again to show that

$$\begin{aligned} p_{j-2}(t-2) &\le p_{j-1}(t-1) + 1 \\ &< M - 2(j-2) + 1 \\ &< M - 2(j-3). \end{aligned}$$

Repeating the same argument yields

$$p_{j-j'}(t-j') \le p_{j-j'+1}(t-j'+1) + 1$$

for all $j' = 1, \ldots, j-1$. Summing up for all $j' = 1, \ldots, j-1$, we derive

$$p_1(t-(j-1)) \le p_j(t) + j - 1. \quad (19)$$

From the lower bound in (ii) of this lemma and (19), it follows that

$$x(t-j) \le p_1(t-(j-1)) + 1 \le p_j(t) + j. \quad (20)$$

On the other hand, we have from (i) of this lemma that

$$x(t-1) \le x(t-2) + 1 \le \cdots \le x(t-j) + j - 1. \quad (21)$$

Using (21) in (20) yields

$$x(t-1) \le p_j(t) + 2j - 1. \tag{22}$$

The proof is completed. ■

Proof. (Proof of Theorem 3) It suffices to consider the tagged packet that enters the ${i_p}^{th}$ delay line at time t_p, where $m+1 \le i_p \le \lceil M/2 \rceil$. Note from the definition of $p_j(t)$ in (14) that the total number of packets with priorities higher than or equal to the priority of the tagged packet is at least $\sum_{j=1}^{m} p_j(t_p)$. From Lemma 5(iii), the upper bound in Lemma 5(ii), and the fact that $x(t_p-1)$ is an integer, it follows that

$$\begin{aligned} &\sum_{j=1}^{m} p_j(t_p) \\ &= p_1(t_p) + \sum_{j=2}^{m} p_j(t_p) \\ &\ge p_1(t_p) + \sum_{j=2}^{m} (x(t_p-1) - 2j + 1)^+ \\ &\ge p_1(t_p) + \sum_{j=2}^{m} \lceil ((p_1(t_p) - M + 2m - 4j + 1)/2)^+ \rceil . \end{aligned}$$

Since the tagged packet enters the ${i_p}^{th}$ delay line at time t_p, we have from (11) that $p_1(t_p) = i_p \ge \min[i_p, M+1-i_p]$. As such, the total number of packets with priorities higher than or equal to the priority of the tagged packet is not less than

$$\begin{aligned} &\min[i_p, M+1-i_p] \\ &+ \sum_{j=2}^{m} \lceil ((\min[i_p, M+1-i_p] - M + 2m - 4j + 1)/2)^+ \rceil . \end{aligned} \tag{23}$$

As there is a total order for the priorities of all the packets in a complementary priority queue, the only packet that has the same priority as the tagged packet is the tagged packet itself. Therefore, the total number of packets with priorities higher than the priority of the tagged packet is not less than

$$\begin{aligned} &\min[i_p, M+1-i_p] \\ &+ \sum_{j=2}^{m} \lceil ((\min[i_p, M+1-i_p] - M + 2m - 4j + 1)/2)^+ \rceil - 1. \end{aligned} \tag{24}$$

As the construction is symmetric, the same argument can also be used to show that the total number of packets with priorities *lower* than the priority of the tagged packet is not less than the quantity in (24). Since we assume that $d_i = d_{M+1-i} = i$ for all $i = 1, 2, \ldots, m$, and

$$d_i = d_{M+1-i} \le i + \sum_{j=2}^{m} \lceil ((i - M + 2m - 4j + 1)/2)^+ \rceil$$

for all $i = m+1, \ldots, \lceil M/2 \rceil$, the tagged packet cannot be either the packet with the *highest* priority or the packet with *lowest* priority until it reaches the ${i_p}^{th}$ input of the sorter. This shows that the construction in Figure 5 is indeed a complementary priority queue with buffer $\sum_{i=1}^{M} d_i$. ■

V. Conclusions

In this paper, we considered the construction of an optical priority queue with a single $(M+1) \times (M+1)$ switch and M delay lines. By establishing a *partial ordering* for all the packets stored in the delay lines, we showed that such a construction can be used for exact emulation of an optical priority queue with $O(M^3)$ buffer size. Even though we have increased the buffer size from $O(M^2)$ in [8][9] to $O(M^3)$, it is still much worse than the exponential upper bound $O(2^M)$ derived in [8]. As commented in [9], if a priority queue only has K priority classes of packets, then the exponential bound in [8] can be achieved by using K FIFO queues for these K classes of packets. This is possible because a FIFO queue with buffer B can be constructed with $O(\log B)$ 2×2 crossbar switches (see e.g., [6]). However, for a general priority queue like the one considered in this paper, it still requires further study to go beyond the $O(M^3)$ buffer size.

Acknowledgment

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.

References

[1] R. L. Cruz and J.-T. Tsai, "COD: alternative architectures for high speed packet switching," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 11–21, February 1996.

[2] D. K. Hunter, D. Cotter, R. B. Ahmad, D. Cornwell, T. H. Gilfedder, P. J. Legg, and I. Andonovic, "2×2 buffered switch fabrics for traffic routing, merging and shaping in photonic cell networks," *IEEE Journal of Lightwave Technology*, vol. 15, pp. 86–101, January 1997.

[3] C.-S. Chang, D.-S. Lee, and C.-K. Tu, "Recursive construction of FIFO optical multiplexers with switched delay lines," *IEEE Transactions on Information Theory*, vol. 50, pp. 3221–3233, December 2004.

[4] C.-S. Chang, D.-S. Lee, and C.-K. Tu, "Using switched delay lines for exact emulation of FIFO multiplexers with variable length bursts," *IEEE Journal on Selected Areas in Communications*, vol. 24, pp. 108–117, April 2006.

[5] C.-C. Chou, C.-S. Chang, D.-S. Lee and J. Cheng, "A necessary and sufficient condition for the construction of 2-to-1 optical FIFO multiplexers by a single crossbar switch and fiber delay lines," to appear in *IEEE Transactions on Information Theory*, vol. 52, October 2006.

[6] C.-S. Chang, Y.-T. Chen, and D.-S. Lee, "Constructions of optical FIFO queues," *IEEE Transactions on Information Theory*, vol. 52, pp. 2838–2843, June 2006.

[7] C.-S. Chang, Y.-T. Chen, J. Cheng, and D.-S. Lee, "Multistage constructions of linear compressors, non-overtaking delay lines, and flexible delay lines," submitted to *IEEE/ACM Transactions on Networking*. Conference version in *Proceedings IEEE 25th Annual Conference on Computer Communications (INFOCOM'06)*, Barcelona, Spain, April 23–29, 2006.

[8] A. D. Sarwate and V. Anantharam, "Exact emulation of a priority queue with a switch and delay lines," to appear in *Queueing Systems: Theory and Applications*, vol. 53, pp. 115–125, July 2006.

[9] H.-C. Chiu, C.-S. Chang, J. Cheng, and D.-S. Lee, "A simple proof for the constructions of optical priority queues," submitted to *Queueing Systems: Theory and Applications*, 2005.