Recursive Constructions of Parallel FIFO and LIFO Queues with Switched Delay Lines

Po-Kai Huang, Cheng-Shang Chang, *Fellow, IEEE,* Jay Cheng, *Member, IEEE,*
and Duan-Shin Lee, *Senior Member, IEEE*

***Abstract*— One of the most popular approaches for the constructions of optical buffers needed for optical packet switching is to use Switched Delay Lines (SDL). Recent advances in the literature have shown that there exist systematic SDL construction theories for various types of optical buffers, including First In First Out (FIFO) multiplexers, FIFO queues, priority queues, linear compressors, non-overtaking delay lines, and flexible delay lines. As parallel FIFO queues with a shared buffer are widely used in many switch architectures, e.g., input-buffered switches and load-balanced Birkhoff-von Neumann switches, in this paper we propose a new SDL construction for such queues. The key idea of our construction for parallel FIFO queues with a shared buffer is *two-level caching*, where we construct a dual-port random request queue in the upper level (as a high switching speed storage device) and a system of scaled parallel FIFO queues with a shared buffer in the lower level (as a low switching speed storage device). By determining appropriate dumping thresholds and retrieving thresholds, we prove that the two-level cache can be operated as a system of parallel FIFO queues with a shared buffer. Moreover, such a two-level construction can be recursively expanded to an n-level construction, where we show that the number of 2×2 switches needed to construct a system of N parallel FIFO queues with a shared buffer B is $O((N \log N) \log(B/(N \log N)))$ for $N >> 1$. For the case with $N = 1$, i.e., a single FIFO queue with buffer B, the number of 2×2 switches needed is $O(\log B)$. This is of the same order as that previously obtained by Chang *et al*. We also show that our two-level recursive construction can be extended to construct a system of N parallel Last In First Out (LIFO) queues with a shared buffer by using the same number of 2×2 switches, i.e., $O((N \log N) \log(B/(N \log N)))$ for $N >> 1$ and $O(\log B)$ for $N = 1$. Finally, we show that a great advantage of our construction is its fault tolerant capability. The reliability of our construction can be increased by simply adding extra optical memory cells (the basic elements in our construction) in each level so that our construction still works even when some of the optical memory cells do not function properly.**

***Index Terms*— Caches, FIFO queues, LIFO queues, optical buffers, switched delay lines.**

I. Introduction

One of the key problems of optical packet switching is the lack of optical buffers. Unlike electronic packets, optical packets cannot be easily stopped, stored, and forwarded. The only known way to “store” optical packets is to direct them via a set of optical switches through a set of fiber delay lines so that optical packets come out at the right place and at the right time. Such an approach, known as Switched Delay Line (SDL) construction, has received a lot of attention recently (see e.g., [1]–[15] and the references therein). Early SDL constructions for optical buffers, including the shared-memory switch in [1] and CORD (contention resolution by delay lines) in [2][3], focused more on the feasibility of such an approach. On the other hand, recent advances in SDL constructions have shown that there exist systematic methods for the constructions of various types of optical buffers, such as First In First Out (FIFO) multiplexers in [4]–[9], FIFO queues in [10], priority queues in [11][12], and linear compressors, non-overtaking delay lines, and flexible delay lines in [13].

In this paper, we focus on the constructions of optical parallel FIFO queues with a shared buffer as such queues are crucial in switch design. For instance, the virtual output queues in input-buffered switches (see e.g., [16][17]) and the central buffers in load-balanced Birkhoff-von Neumann switches (see e.g., [18][19]) can all be implemented by using parallel FIFO queues with a shared buffer. One of the main contributions of this paper is to provide a two-level recursive construction of parallel FIFO queues with a shared buffer. The key idea of our two-level construction is *caching* (see e.g., [20]–[22]). The upper level in our construction is a random request queue (see Definition 4 in Section II) that can be viewed as a high switching speed storage device, while the lower level in our construction is a system of scaled parallel FIFO queues with a shared buffer that can be viewed as a low switching speed storage device. By determining appropriate dumping thresholds and retrieving thresholds, we show that the two-level cache can be operated as a system of parallel FIFO queues with a shared buffer. Moreover, such a two-level construction can be recursively expanded to an n-level construction, where we show that the number of 2×2 switches needed to construct a system of N parallel FIFO queues with a shared buffer B is $O((N \log N) \log(B/(N \log N)))$ for $N >> 1$. For the case with $N = 1$, i.e., a single FIFO queue with buffer B, the construction complexity (in term of the number of 2×2 switches) is $O(\log B)$. This is of the same order as that in [10].

To our surprise, the two-level recursive construction can be extended to construct a system of N parallel LIFO queues with a shared buffer. The only modification of our architecture is to use a system of scaled parallel LIFO queues with a shared buffer in the lower level. Therefore, it can also be recursively

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.

The authors are with the Institute of Communications Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan, R.O.C. (e-mail: pkhuang@gibbs.ee.nthu.edu.tw; cschang@ee.nthu.edu.tw; lds@cs.nthu.edu.tw; jcheng@ee.nthu.edu.tw).

expanded to an n-level construction and the number of 2×2 switches needed for the system remains the same. For the case with $N = 1$, i.e., a single LIFO queue with buffer size B, the construction complexity is $O(\log B)$, which is better than $O(\sqrt{B})$ as obtained in [11][12] (we note that the designs in [11][12] are more general and work for priority queues).

We also note that one of the advantages of our construction is its fault tolerant capability. By adding extra optical memory cells (the basic elements in our construction) in each level, the reliability of our construction can be easily increased in the sense that our construction still works even after some of the optical memory cells are broken.

The paper is organized as follows: In Section II, we introduce basic construction elements, including optical memory cells, FIFO queues, and random request queues. In Section III, we propose our two-level recursive construction for N parallel FIFO queues with a shared buffer, the associated operation rules, and the main theorem. We show that the two-level recursive construction can be further expanded to an n-level construction that has a much lower construction complexity in terms of the number of 2×2 switches. The extension to parallel LIFO queues with a shared buffer is reported in Section IV. The paper is concluded in Section V.

In the following, we provide a list of notations used in the paper for easy reference.

$Q(t)$: the set of packets in N parallel FIFO (resp. LIFO) queues at the end of the t^{th} time slot

$Q_1(t)$: the set of packets in level 1 at the end of the t^{th} time slot

$Q_2(t)$: the set of packets in level 2 at the end of the t^{th} time slot

$Q_{1,i}(t)$: the set of packets in the i^{th} queue in level 1 at the end of the t^{th} time slot

$Q_{2,i}(t)$: the set of packets in the i^{th} queue in level 2 at the end of the t^{th} time slot

k: a scaling factor or a frame size

$F_i(t)$: the set of packets in the i^{th} *front* queue at the end of the t^{th} time slot (see Definition 5)

$T_i(t)$: the set of packets in the i^{th} *tail* queue at the end of the t^{th} time slot (see Definition 6)

R_T: Retrieving threshold $R_T = \left\lceil 1 + k \sum_{\ell=1}^{N} \frac{1}{\ell} \right\rceil$

D_T: Dumping threshold $D_T = R_T + k$

$R(t)$: the set of queues that have packets in level 2 at the end of the $(t-1)^{th}$ time slot

$I_i(p,t)$: the departure index of packet p in the i^{th} queue at the end of the t^{th} time slot

II. Basic Network Elements

A. Optical Memory Cells

In our previous papers [10][13], we used optical memory cells as basic network elements for the constructions of various types of optical queues. As in the constructions in [10][13], we assume that packets are of the same size. Moreover, time is slotted and synchronized so that a packet can be transmitted within a time slot. An optical memory cell (see Figure 1) is constructed by a 2×2 optical crossbar switch and a fiber delay line with one time slot (unit) of delay. As illustrated in [10][13], we can set the 2×2 crossbar switch to the "cross" state to write an arriving packet to the optical memory cell. By so doing, the arriving packet can be directed to the fiber delay line with one time slot of delay. Once the write operation is completed, we then set the crossbar switch to the "bar" state so that the packet directed into the fiber delay line keeps recirculating through the fiber delay line. To read out the information from the memory cell, we set the crossbar switch to the "cross" state so that the packet in the fiber delay line can be routed to the output link.

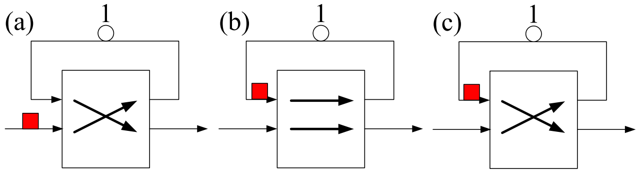

Fig. 1. An optical memory cell: (a) writing information (b) recirculating information (c) reading information

Network elements that are built by optical crossbar switches and fiber delay lines are called Switched Delay Line (SDL) elements in the literature (see e.g., [1]–[13]). Clearly, an optical memory cell that is constructed by a 2×2 switch and a fiber delay line with one unit of delay in Figure 1 is an SDL element. A scaled SDL element is said to be with scaling factor k if the delay in every delay line is k times of that in the original (unscaled) SDL element. For instance, if we scale the fiber length from 1 to 2 in Figure 1, then it is a scaled optical memory cell with scaling factor 2. As the length is now increased to 2, the scaled optical member cell with scaling factor 2 can be used for storing two packets. In general, each packet in a scaled SDL element with scaling factor k can be individually accessed as can be seen in our early papers (see e.g., [7][9][10][12][13]). However, in the proposed recursive constructions of parallel FIFO and LIFO queues in this paper we only need to access the packets *contiguously* as a block of k packets. In other words, in this paper we group every k packets into a block and view a scaled SDL element with scaling factor k as an unscaled SDL element for a block of k packets. For instance, if we group every two packets into a block, then a scaled optical memory cell with scaling factor 2 can be viewed as an unscaled optical memory cell for a block of two packets. This is the key observation that we will use in our construction of parallel FIFO and LIFO queues in this paper.

In the following, we extend the optical memory cell (with a single input and a single output) to a dual-port optical memory cell.

Definition 1 (Dual-port optical memory cells) *A dual-port optical memory cell in Figure 2 is an optical memory cell with one additional I/O port. It consists of a 3×3 switch and a fiber delay line with one unit of delay. The 3×3 switch has the following three connection states: accessing state by the first I/O port in Figure 2(a), recirculating state in Figure 2(b), and accessing state by the second I/O port in Figure 2(c).*

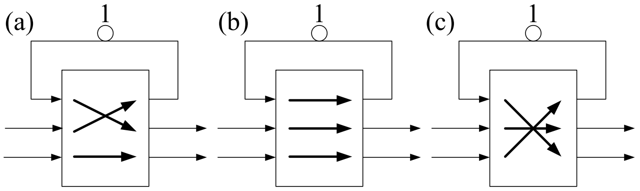

Fig. 2. The three connection states of a dual-port optical memory cell: (a) accessing state by the first I/O port (b) recirculating state (c) accessing state by the second I/O port

As an optical memory cell, a dual-port optical memory cell can be used for storing exactly one packet. Moreover, the stored packet can be accessed by either one of the two I/O ports. With the additional I/O port, we note that a packet arriving at one input of an I/O port may be first stored in a dual-port optical memory cell and then routed to the output of another I/O port in a different time slot. In Figure 3, we show a simple construction of a dual-port optical memory cell by adding a 2×2 switch in front of an optical memory cell and another 2×2 switch after the optical memory cell. It is easy to see that the recirculating state in Figure 2(b) can be realized by setting all the 2×2 switches in Figure 3 to the "bar" state. For the accessing states in Figure 2(a) and (c), the 2×2 switch in the middle of Figure 3 is set to the "cross" state. If it is accessed by the first I/O port, then the other two 2×2 switches are set to the "bar" state. On the other hand, the other two 2×2 switches are set to the "cross" state if it is accessed by the second I/O port. This shows that a dual-port optical memory cell can be constructed by three 2×2 switches. Clearly, the construction in Figure 3 can realize all of the six possible connection states for the 3×3 switch in a dual-port optical memory cell. However, we note that we only need the three connection states described in Definition 1 for the constructions of parallel FIFO and LIFO queues in this paper. The construction complexity (in terms of the number of 2×2 switches) is of the same order as that of a similar construction using all of the six possible connection states, but the control mechanism is much simpler than using all of the six possible connection states.

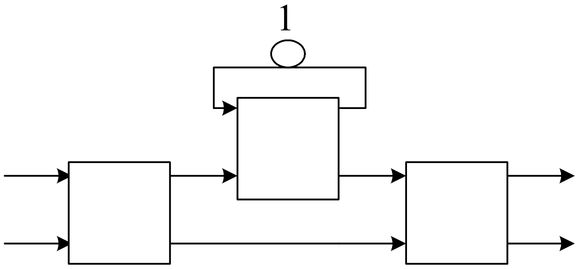

Fig. 3. A simple construction of a dual-port optical memory cell

B. Parallel FIFO Queues

In a FIFO queue, a packet joins the tail of the queue when it arrives. If the buffer of a FIFO queue is finite, then an arriving packet to a full queue is lost. When a packet departs from the head of a FIFO queue, every packet in the queue moves up one position. Specifically, a discrete-time FIFO queue is formalized in the following definition in [10].

Definition 2 (FIFO queues) *A single FIFO queue with buffer B is a network element that has one input link, one control input, and two output links. One output link is for departing packets and the other is for lost packets. Then the FIFO queue with buffer B satisfies the following four properties:*

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

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

(P3) *Maximum buffer usage: if the control input is not enabled, then an arriving packet is lost only when buffer is full.*

(P4) *FIFO: packets depart in the FIFO order.*

The definition of a *single* FIFO queue can be easily extended to *parallel* FIFO queues with a shared buffer as follows:

Definition 3 (Parallel FIFO queues with a shared buffer) *A system of N* parallel *FIFO queues with a* shared *buffer B is a network element that has one input link, N control inputs, and two output links (see Figure 4). As in Definition 2, one output link is for departing packets and the other is for lost packets. Also, each one of the N FIFO queues is associated with a control input under the constraint that at most one of the N control inputs is enabled at any time instant. Then the system of N parallel FIFO queues with a shared buffer B satisfies (P1), (P2), and (P4) in Definition 2 for each FIFO queue. However, as the buffer is shared by the N FIFO queues, the maximum buffer usage property needs to be modified as follows:*

(P3N) *Maximum buffer usage: if there is no departing packet at time t, then an arriving packet at time t is lost only when buffer is full.*

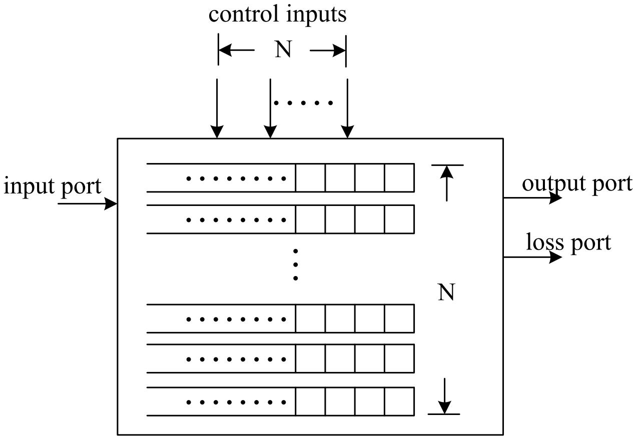

Fig. 4. The N parallel FIFO queues

Note that it is possible that one of the N queues is enabled at time t and there is still no departing packet at time t. This happens when the enabled queue is empty at time t.

The construction of a *single* FIFO queue with buffer B has been studied in [10]. It is shown in [10] that there is a three-stage recursive construction for a FIFO queue, and that a FIFO queue with buffer B can be constructed by using $O(\log B)$ 2×2 switches. However, using the construction of a single FIFO queue in [10] for the construction of a system of N parallel FIFO queues may not be efficient as each FIFO queue needs to be constructed with the same amount of buffer. In this paper, we will propose a new two-level recursive construction that allows the buffer to be shared among the N parallel FIFO queues.

C. Optical Random Request Queues (RRQs)

In this section, we introduce the notion of a Random Request Queue (RRQ). In an RRQ, the departing packet, instead of the first one in a FIFO queue, could be any packet in the queue (including the arriving one). As there is no particular order for departures, the construction complexity of an RRQ is expected to be much higher than that of a FIFO queue. In the following, we provide the formal definition for an RRQ.

Definition 4 (Random request queues) *As indicated in Definition 2 for a FIFO queue, an RRQ with buffer B is a network element that has one input link, one control input, and two output links. One output link is for departing packets and the other is for lost packets. Index the position in the buffer from $1, 2, \dots, B$. An arriving packet can be placed in any one of the B positions as long as it is not occupied (note that it is implicitly assumed that there exists an internal control for the placing of an arriving packet). For an RRQ, the flow conservation property in (P1) of Definition 2 and the maximum buffer usage property in (P3) of Definition 3 are still satisfied. The non-idling property in (P2) of Definition 2 is not needed. Moreover, (P4) needs to be modified as follows:*

(P4R) *Random request: the control input in an RRQ has the set of states $\{0, 1, 2, \dots, B + 1\}$. When the state of the control input is not zero, we say the control input is enabled. If the state of the control input is i for $i = 1, 2, \dots, B$, then the packet in the i^{th} position of the queue (if there is one) is sent to the output link. If the state of the control input is $B + 1$, then the arriving packet (if there is one) is sent to the output link.*

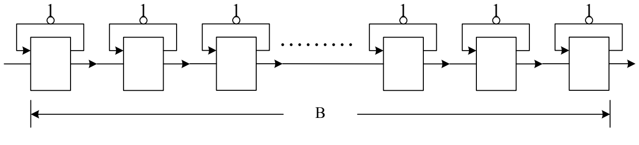

Fig. 5. A construction of an optical RRQ with buffer B

Now we show in Figure 5 a way to construct an optical RRQ with buffer B by a concatenation of B optical memory cells. As discussed in the previous section, an optical memory cell can be used for storing one arriving optical packet. To see the random request property in (P4R), we index the B buffer positions (optical memory cells) from left to right. Suppose that the i^{th} optical memory cell is empty, then an arriving packet can be written into the i^{th} optical memory cell by setting the 2×2 optical crossbar switch of the i^{th} optical memory cell to the "cross" state and the other 2×2 optical crossbar switches to the "bar" state. On the other hand, if the i^{th} optical memory cell is occupied and the state of the control input is i, where $i = 1, 2, \dots, B$, then the packet stored in the i^{th} optical memory cell can be routed to the output by setting the 2×2 optical crossbar switch of the i^{th} optical memory cell to the "cross" state and the other 2×2 optical crossbar switches to the "bar" state. If there is an arriving packet and the state of the control input is $B+1$, then the arriving packet can be sent to the output link immediately by setting all the 2×2 optical crossbar switches to the "bar" state. Note that it is possible for a packet stored at the i^{th} optical memory cell to depart from the RRQ while an arriving packet is routed to the i^{th} optical memory cell at the same time.

The problem of the construction in Figure 5 is the maximum buffer usage property. If all the B optical memory cells are occupied and there is no departing packet, then an arriving packet should be routed to the loss port. For this reason, one needs to add a 1×2 switch in front of the construction in Figure 5 for admission control. However, in the later development, we only operate all the RRQs in such a way that there is no buffer overflow. As such, the 1×2 switch needed for the construction of an RRQ is omitted for clarity.

Instead of using optical memory cells with a single I/O port, one can use dual-port optical memory cells in Figure 5. This results in a *dual-port RRQ* with buffer B in Figure 6. Note that we need two control inputs for a dual-port RRQ, one control input is for the random request from the first output link and the other is for the random request from the second output link. The dual-port RRQ in Figure 6 satisfies the flow conservation property and the random request property in Definition 4. However, as there are only three connection patterns in every dual-port optical memory cell, it is not possible for an arriving packet to be routed to the i^{th} dual-port optical memory cell from the input link of one I/O port while the packet stored in the i^{th} dual-port optical memory cell is departing from the output link of another I/O port at the same time. One consequence of such a restriction is that the maximum buffer usage property is not satisfied. This can be seen from the following worst-case scenario. Suppose that all of the B dual-port optical memory cells are occupied at time t. If at time $t+1$ there is a packet arriving at the input link of the second I/O port and the state of the first control input is i, then the packet stored in the i^{th} dual-port optical memory cell will be sent to the output link of the first I/O port, but the arriving packet at the input link of the second I/O port can not be placed in the i^{th} dual-port optical memory cell (which is empty now). As such, the arriving packet at the second input link has to be sent to the loss link and the maximum buffer usage property is not satisfied.

It is clear that if the maximum buffer usage property has to be satisfied, then the maximum buffer that can be achieved by the construction in Figure 6 is $B - 1$. Therefore, it would be technically correct to call the construction in Figure 6 a dual-port RRQ with buffer $B - 1$. However, in the recursive constructions of parallel FIFO and LIFO queues in this paper, we never require an arriving packet from the input link of one I/O port be routed to the output link of another I/O port of a dual-port optical memory cell at the same time, and hence the worst-case scenario mentioned above never occurs. In other words, there are always empty dual-port optical memory cells for arriving packets at the input links (see the proof of Theorem 8 in Appendix A for details). As such, in our proposed scheme the construction in Figure 6 achieves the maximum buffer B, and that is why we call the construction in Figure 6 a dual-port RRQ with buffer B in this paper. Finally, we would like to point out that the reason why the maximum buffer usage property is not satisfied is due to the fact that we only use three connection patterns in every dual-port optical memory cell in this paper. If we use all of the six possible connection states for the 3×3 switch in a dual-port optical memory cell, then the maximum buffer usage property is satisfied. However, the maximum buffer that can be achieved is still the same. This implies that a construction similar to that proposed in this paper and using all of the six possible connection states achieves the same order of buffer size, but undoubtedly increases the complexity of the control mechanism.

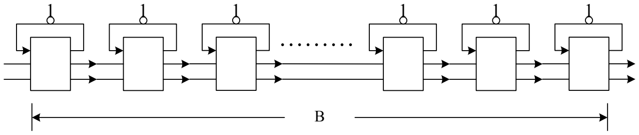

Fig. 6. A construction of a dual-port RRQ with buffer B via a concatenation of dual-port optical memory cells

III. Recursive Constructions of Parallel FIFO Queues with a Shared Buffer

A. A Two-level Construction of Parallel FIFO Queues with a Shared Buffer

It is obvious to see that an RRQ with buffer B can be operated as N parallel FIFO queues with a shared buffer B. However, the number of 2×2 switches needed for the construction of an RRQ with buffer B in Figure 5 is also B. As packets have to depart in the FIFO order, the construction complexity of N parallel FIFO queues with a shared buffer B (in terms of the number of 2×2 switches) should be much less than that of an RRQ with buffer B. To show this, in this section we provide a recursive construction of N parallel FIFO queues with a shared buffer $B_1 + kB_2$ in Figure 7. The construction in Figure 7 consists of two levels: a dual-port RRQ with buffer B_1 in level 1, and a scaled SDL network element that can be used as a system of N parallel FIFO queues with a shared buffer B_2 and scaling factor k in level 2. The 1×2 switch in front of the network element is for admission control. Its objective is to make sure that the total number of packets inside the network element does not exceed $B_1 + kB_2$. An arriving packet can only be admitted if the total number of packets inside the network element does not exceed $B_1 + kB_2$ after its admission. Otherwise, it is routed to the loss port.

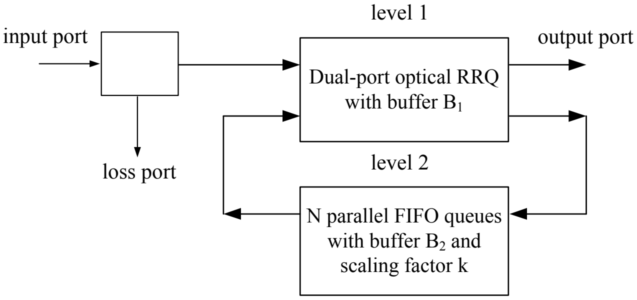
level 1
input port
output port
Dual-port optical RRQ with buffer B1
loss port
level 2
N parallel FIFO queues with buffer B2 and scaling factor k

Fig. 7. A recursive construction of N parallel FIFO queues with buffer $B_1 + kB_2$

The key idea behind the construction in Figure 7 is *caching*. Note that if we group every k time slots into a frame and operate the scaled SDL element in level 2 at the time scale of frames, then the scaled SDL element in level 2 can be used as a system of N parallel FIFO queues with a shared buffer B_2 by viewing k consecutive packets as a *block* of packets. As such, the scaled SDL element in level 2 can be viewed as a storage device with a much lower switching speed (k times slower) than that of the dual-port RRQ in level 1. As in most caching systems, the problems are about (i) when to dump packets from the high switching speed storage device in level 1 to the low switching speed storage device in level 2, and (ii) when to retrieve packets from the low switching speed storage device in level 2 to the high switching speed storage device in level 1.

Consequently, we let D_T be the *dumping threshold* and R_T be the *retrieving threshold*. These two thresholds will be used to determine when to dump packets and when to retrieve packets. To be precise, let $Q_{\ell,i}(t)$, $\ell = 1$ and 2, $i = 1, 2, \ldots, N$, be the set of packets in the i^{th} queue that are stored in level ℓ at the end of the t^{th} time slot. Then the set of packets in the i^{th} queue at the end of the t^{th} time slot is simply $Q_{1,i}(t) \cup Q_{2,i}(t)$. Furthermore, let $Q_1(t)$ (resp. $Q_2(t)$) be the set of packets in level 1 (resp. level 2) at the end of the t^{th} time slot. Clearly, for $\ell = 1$ and 2,

$$Q_\ell(t) = \bigcup_{i=1}^{N} Q_{\ell,i}(t). \tag{1}$$

Also, the set of packets in the N parallel FIFO queues at the end of the t^{th} time slot, denoted by $Q(t)$, is the union of the set of packets in each queue of each level, i.e.,

$$Q(t) = Q_1(t) \cup Q_2(t) = \bigcup_{\ell=1}^{2} \bigcup_{i=1}^{N} Q_{\ell,i}(t). \tag{2}$$

For all the packets in the i^{th} FIFO queue at time t, i.e., $Q_{1,i}(t) \cup Q_{2,i}(t)$, we can sort them according to their departure order. Specifically, we let $I_i(p,t)$ be the departure index of packet p in the i^{th} queue at time t, i.e., $I_i(p,t) = j$ if packet p is the j^{th} packet to depart in the i^{th} queue at the end of the t^{th} time slot.

In the following, we use the departure index to define the notions of front queues and tail queues that are needed for our operation.

Definition 5 (Front queues) *The i^{th} front queue at time t, denoted by $F_i(t)$, is a subset of the packets in the i^{th} queue in level 1 at time t, i.e., $F_i(t) \subseteq Q_{1,i}(t)$. A packet p is in $F_i(t)$ if*

(1) *there are packets in the i^{th} queue in level 2 and the departure index of packet p is smaller than that of any packet in the i^{th} queue in level 2, i.e., $|Q_{2,i}(t)| > 0$ and $I_i(p,t) < I_i(\tilde{p},t)$, $\forall\, \tilde{p} \in Q_{2,i}(t)$, or*

(2) *there are no packets in the i^{th} queue in level 2 and the departure index of packet p is not greater than the dumping threshold, i.e., $|Q_{2,i}(t)| = 0$ and $I_i(p,t) \leq D_T$.*

Definition 6 (Tail queues) *The i^{th} tail queue at time t, denoted by $T_i(t)$, is a subset of the packets in the i^{th} queue in level 1 at time t, i.e., $T_i(t) \subseteq Q_{1,i}(t)$. A packet p is in $T_i(t)$ if*

(1) *there are packets in the i^{th} queue in level 2 and the departure index of packet p is greater than that of any packet in the i^{th} queue in level 2, i.e., $|Q_{2,i}(t)| > 0$ and $I_i(p,t) > I_i(\tilde{p},t)$, $\forall\, \tilde{p} \in Q_{2,i}(t)$, or*

(2) *there are no packets in the i^{th} queue in level 2 and the departure index of packet p is greater than the dumping threshold, i.e., $|Q_{2,i}(t)| = 0$ and $I_i(p,t) > D_T$.*

We note from Definition 5 and Definition 6 that the departure index of a packet in the i^{th} front queue is always smaller than that of any packet in the i^{th} tail queue at any time, no matter the i^{th} queue in level 2 is empty or not. As such, the i^{th} front queue and the i^{th} tail queue are always disjoint at any time, i.e., $F_i(t) \cap T_i(t) = \phi$ for all t.

Now we describe the operations of our recursive construction in Figure 7. In our operations, every k time slots are grouped into a frame. The RRQ in level 1 is operated in every time slot, while the scaled N parallel FIFO queues in level 2 is operated in the time scale of frames.

(R0) Admission control: an arriving packet can be admitted to the network element in Figure 7 only if the total number of packets in the network element does not exceed $B_1 + kB_2$ after its admission. Otherwise, it is routed to the loss port by the 1 × 2 switch in front of the network element in Figure 7.

(R1) Write operation: suppose that there is an arriving packet to the i^{th} queue at time t. If the i^{th} queue is empty at time $t-1$ and the i^{th} queue is enabled at time t, then the arriving packet is routed to the output port immediately. Otherwise, the arriving packet is stored in the dual-port RRQ in level 1 (as long as the total number of packets in the construction does not exceed $B_1 + kB_2$ after its admission).

(R2) Read operation: suppose that the i^{th} queue is enabled at time t. If the i^{th} queue is empty at time $t-1$ and there is an arriving packet to the i^{th} queue at time t, then the arriving packet is routed to the output port immediately. If the i^{th} queue has packets in level 1 at time $t-1$, the packet that has the smallest departure index among all the packets of the i^{th} queue in the dual-port RRQ in level 1 is sent to the output port at time t. Otherwise, there is no departing packet at time t.

(R3) Retrieve operation (the shortest front queue below the retrieving threshold): suppose that t is the beginning time slot of the m^{th} frame, i.e., $t = k(m-1)+1$. Consider the set of queues $R(t)$ that have packets in level 2 at time $t-1$. Suppose that the i^{th} queue is the queue that has the smallest number of packets in its front queue at time $t-1$ among all the queues in $R(t)$. If the number of packets in the i^{th} front queue at time $t-1$ is less than or equal to the retrieving threshold R_T, i.e., $|F_i(t-1)| \leq R_T$, then the i^{th} FIFO queue in level 2 is enabled during the m^{th} frame. As such, there are k packets retrieved from the i^{th} FIFO queue in level 2 to the i^{th} front queue in $[t, t+k-1]$.

(R4) Dump operation (the longest tail queue with a full block of packets): suppose that t is the beginning time slot of the m^{th} frame, i.e., $t = k(m-1)+1$. Suppose that the i^{th} queue is the queue that has the largest number of packets in its tail queue at time $t-1$ among all the N queues. If there are at least k packets in the i^{th} tail queue at time $t-1$, i.e., $|T_i(t-1)| \geq k$, then the k packets with the *smallest* departure indices in the i^{th} tail queue are sent (starting from the packet with the *smallest* departure index among these k packets) to the i^{th} FIFO queue in level 2 (as a block of packets) provided that there is buffer space in level 2 (i.e., either the buffer of the N FIFO queues in level 2 is not full at time $t-1$ or there is a retrieve operation at time t).

We note that both the retrieve operation and the dump operation can only occur at the beginning time slot of a frame. Also, if the two-level recursive construction in Figure 7 is started from an empty system, then in our operations we always keep $Q_{1,i}(t) = F_i(t) \cup T_i(t)$. In other words, if the i^{th} queue in level 2 is not empty, then the departure index of a packet in the i^{th} queue in level 1 is either greater than that of any packet of the i^{th} queue in level 2 or smaller than that of any packet of the i^{th} queue in level 2. (As for the case that the i^{th} queue in level 2 is empty, it holds trivially that $Q_{1,i}(t) = F_i(t) \cup T_i(t)$ as in this case $F_i(t)$ contains all the packets in the i^{th} queue with departure indices less than or equal to the dumping threshold D_T, and $T_i(t)$ contains all the packets in the i^{th} queue with departure indices greater than D_T.) As this property will be very useful in the proof of our main theorem (Theorem 8 below) in this paper, we state this property formally in the following lemma.

Lemma 7 *Suppose that the two-level recursive construction in Figure 7 is started from an empty system. Then under (R0)–(R4), we have $Q_{1,i}(t) = F_i(t) \cup T_i(t)$ for all $t \geq 0$.*

Proof. We prove this lemma by induction on t. As the two-level recursive construction in Figure 7 is started from an empty system, $Q_{1,i}(t) = F_i(t) \cup T_i(t)$ holds trivially for $t = 0$.

Assume that it also holds for some $t-1 \geq 0$. We consider the following four cases.

Case 1: There is an arriving packet to the i^{th} queue at time t. According the write operation in (R1), the arriving packet is either routed to the output port immediately or stored in the dual-port RRQ in level 1 (as long as the total number of packets in the construction does not exceed $B_1 + kB_2$ after its admission). If the arriving packet is routed to the output port immediately or the i^{th} queue in level 2 is empty at time t, then there is nothing to prove. On the other hand, if the arriving packet is not routed to the output port immediately and the i^{th} queue in level 2 is not empty at time t, then it will be placed in the i^{th} tail queue as the arriving packet has the *largest* departure index among all the packets of the i^{th} queue. From the induction hypothesis, we easily conclude that $Q_{1,i}(t) = F_i(t) \cup T_i(t)$.

Case 2: The i^{th} queue is enabled at time t. According the read operation in (R2), the packet with the *smallest* departure index among all the packets, including the arriving packet (if there is one), of the i^{th} queue in the dual-port RRQ in level 1 is sent to the output port at time t. It follows trivially from the induction hypothesis that $Q_{1,i}(t) = F_i(t) \cup T_i(t)$.

Case 3: There is a retrieve operation performed on the i^{th} queue at time t. According the retrieve operation in (R3), the retrieved packet is the packet with the *smallest* departure index among all the packets in the i^{th} FIFO queue in level 2 as packets in a FIFO queue must depart in the FIFO order. If the i^{th} queue in level 2 is empty at time t, then there is nothing to prove. On the other hand, if the i^{th} queue in level 2 is not empty at time t, then the retrieved packet will be placed in the i^{th} front queue as it has a departure index smaller than all the packets of the i^{th} queue in level 2 at time t. As such, we have from the induction hypothesis that $Q_{1,i}(t) = F_i(t) \cup T_i(t)$.

Case 4: There is a dump operation performed on the i^{th} queue at time t. According the dump operation in (R4), the dumped packet is the packet with the *smallest* departure index among all the packets of the i^{th} tail queue. As in Case 2 above, it follows trivially from the induction hypothesis that $Q_{1,i}(t) = F_i(t) \cup T_i(t)$.

Finally, we note that although we discuss the above four cases separately, the arguments still hold if two or more of the above four cases occur at the same time. ■

Now we state the main theorem of our paper. The proof of Theorem 8 will be presented in Appendix A.

Theorem 8 *Suppose the two-level recursive construction in Figure 7 is started from an empty system. If we choose* $R_T = \left\lceil 1 + k \sum_{\ell=1}^{N} \frac{1}{\ell} \right\rceil$, $D_T = R_T + k$, *and* $B_1 \geq ND_T + N(k-1) + k + 1$, *then under (R0)–(R4) the construction in Figure 7 achieves the exact emulation of a system of N parallel FIFO queues with a shared buffer $B_1 + B_2k$.*

To see the intuition why we need to set $R_T = \left\lceil 1 + k \sum_{\ell=1}^{N} \frac{1}{\ell} \right\rceil$. We consider an ideal fluid model as in [21] and [22]. As can be seen in [21] and [22], the largest amount of fluid that can be drained from queue 1 in level 1 is achieved in the following scenario. Suppose that initially the number of packets in every front queue in level 1 is $R_T + \epsilon$ for some small $\epsilon > 0$. As such, no retrieve operation is performed. During the first frame, all the N front queues in level 1 are drained at the same rate. By the end of the first frame, the number of packets in the i^{th} front queue, $i = 1, 2, \ldots, N$, is roughly $R_T + \epsilon - k/N$. At the beginning of the second frame, a retrieve operation is performed on one of the N queues, say queue N. This takes another k time slots (a frame) and k packets of queue N are retrieved from level 2 to its front queue in level 1. During the second frame, the first $N-1$ front queues are drained at the same rate. By the end of the second frame, the number of packets in the i^{th} front queue, $i = 1, 2, \ldots, N-1$, is roughly $R_T + \epsilon - k/N - k/(N-1)$. At the beginning time slot of the third frame, a retrieve operation is performed on queue $N-1$. During the third frame, the first $N-2$ front queues are drained at the same rate. By the end of the third frame, the number of packets in the i^{th} front queue, $i = 1, 2, \ldots, N-2$, is roughly $R_T + \epsilon - k/N - k/(N-1) - k/(N-2)$. Repeating the same argument, one can argue that by the end of the N^{th} frame the number of packets in the first front queue is roughly $R_T + \epsilon - k \sum_{\ell=1}^{N} 1/\ell$. This has to be nonnegative so that the non-idling property can be satisfied.

We now discuss our choice of D_T, the threshold for dump operations. If we set $D_T \geq R_T + k$, then from the definitions of a front queue and a tail queue, all the k packets retrieved in one frame from a queue in level 2 will be stored in its front queue in level 1. Since a larger D_T would require a larger buffer size B_1 for the dual-port RRQ in level 1, we set $D_T = R_T + k$.

The reason why we need $B_1 \geq ND_T + N(k-1) + k + 1$ can be explained intuitively by the following scenario: suppose at the beginning time slot of a frame each of the N queues has D_T packets in its front queue and $k-1$ packets in its tail queue. As such, no dump operation is performed for that frame. During that frame, there are k arriving packets and they are stored in the dual-port RRQ in level 1. At the end of that frame, there are $N(D_T + k - 1) + k$ packets in the dual-port RRQ in level 1. As such, one of the tail queues must have at least k packets and a dump operation is performed at the beginning time slot of the next frame. Suppose that there is another arriving packet at the beginning time slot of the next frame. Even though there is a packet dumped from level 1 to level 2, this arriving packet has to be stored in a buffer space *different* from the one being freed by the dumped packet. This is because the dual-port RRQ in level 1, constructed by dual-port optical memory cells, only allows three connection patterns and the dumped packet and the arriving packet have to use different I/O ports. As such, we need $B_1 \geq ND_T + N(k-1) + k + 1$ at the beginning time slot of the next frame in this scenario.

Furthermore, if in the above scenario there is an arriving packet in each time slot, then the dump operations continue until the buffer in level 2 is full, from which time the arriving packets are stored in the buffer in level 1 until the entire queue is full. This shows that the maximum possible buffer $B_1 + B_2k$ could be achieved.

In short, our choices of B_1 and D_T ensure that empty memory cells are always available to store new arriving packets. As such, the flow conservation property and the maximum buffer usage property are satisfied. Also, our choice of R_T ensures that the non-idling property is satisfied.

B. Recursive Expansion to an n-level Construction of Parallel FIFO Queues with a Shared Buffer

One can recursively expand the two-level construction in Theorem 8 to an n-level construction in Figure 8. This n-level construction can be used for exact emulation of a system of N parallel FIFO queues with a shared buffer $B_1(k^{n-1}-1)/(k-1)+B_2k^{n-1}$. To see this, consider the case when $n=3$. Then we have from Theorem 8 that the dual-port RRQ with buffer B_1 and scaling factor k in level 2 and the system of N parallel FIFO queues with a shared buffer B_2 and scaling factor k^2 in level 3 can be used for exact emulation of a system of N parallel FIFO queues with a shared buffer B_1+kB_2 and scaling factor k. Using Theorem 8 again, one can show that the 3-level construction can be used for exact emulation of a system of N parallel FIFO queues with a shared buffer $B_1+k(B_1+kB_2)$.

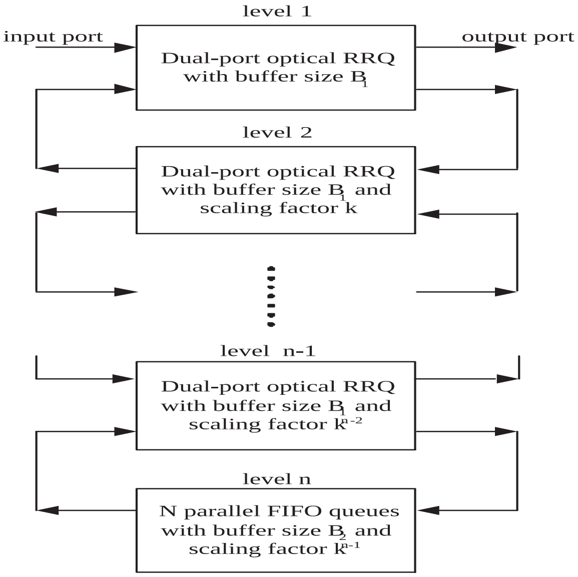

Fig. 8. An n-level construction of N parallel FIFO queues

Note that an RRQ with buffer B can be used for exact emulation of a system of N parallel FIFO queues with a shared buffer B and that the number of 2×2 switches needed for an RRQ with buffer B in Figure 5 is $O(B)$. If we choose $B_2=B_1$ in Figure 8, then the number of 2×2 switches needed for the n-level construction of N parallel FIFO queues with buffer $B_1(k^n-1)/(k-1)$ is $O(nB_1)$. This shows that one can construct N parallel FIFO queues with buffer B with $O(B_1\log_k(B/B_1))$ 2×2 switches. From Theorem 8, the minimum number that one can choose for B_1 is $ND_T+N(k-1)+k+1$. For $N=1$, one can simply choose $B_1=4k+1$ and construct a FIFO queue with $O(\log B)$ 2×2 switches. This is of the same order as that in [10]. In particular, if we choose $k=2$, then one only needs 9 dual-port optical memory cells in each level. In Figure 9, we show a 3-level construction of a FIFO queue with buffer 63.

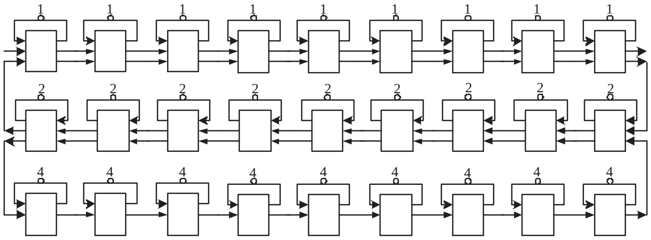

Fig. 9. A 3-level construction of a FIFO queue with buffer 63

For $N>>1$ and $k=2$, we know from the complexity of the harmonic function that $B_1=ND_T+N(k-1)+k+1$ is $O(N\log N)$. Thus, $O((N\log N)\log(B/(N\log N)))$ 2×2 switches can be used to construct N parallel FIFO queues with buffer B.

Note that in Theorem 8 the condition for the buffer in level 1 is $B_1\geq ND_T+N(k-1)+k+1$. This leads to a great advantage of our construction – the fault tolerant capability. Specifically, if each optical memory cell has a bypass circuit that sets up a direct connection between its input link and its output link once a fault within the optical memory cell is detected, then by setting $B_1=F+ND_T+N(k-1)+k+1$ our construction still works even after up to F optical memory cells are broken.

In [23], Bouillard and Chang provided a solution for the control of 2×2 switches in optical FIFO queues and non-overtaking delay lines, which were designed based on recursive constructions. As in this paper the parallel FIFO queues with a shared buffer are also recursively constructed, and from (R0)–(R4) we know that we only need to keep track of the front queues and tail queues of the RRQs, the control mechanism of the 2×2 switches in the proposed recursive construction of parallel FIFO queues with a shared buffer is expected to be simpler than that in [23].

Before we move on to the recursive constructions of parallel LIFO queues with a shared buffer in the next section, we briefly address a few practical implementation issues that are of concern to some researchers. With recent advances in optical technologies, the constructions of compact and tunable optical buffers have been made feasible by using the so-called "slow light" technique [24]–[28]. For instance, optical buffers can be implemented in the nano-scale in today's technology [24], and hence the construction of an optical buffer may not be as bulky as one might expect. Also it has been demonstrated that a 75-ps pulse can be delayed by up to 47 ps [28], and thus the synchronization issue that is usually of practical concern may not be a serious design obstacle.

Furthermore, current photonic technology allows to implement a 2×2 optical memory cell using photonic regeneration and reshaping (P2R) wavelength converters [29]–[31] and arrayed waveguide grating (AWG) [32][33]. According to [31], P2R wavelength converters have an excellent cascadability of up to fifteen cascaded stages using today's technology. With error correcting codes employed, it is expected that the number of cascaded stages can be much higher. As such, the power loss due to recirculations through the fiber delay lines may not be a critical design obstacle as one might expect. Of course, it will be unrealistic to allow a packet to recirculate through the fiber delay lines indefinitely, and there should be a limitation on the number of recirculations through the fiber delay lines. In an approximate implementation, one may simply drop the packets that have to be recirculated through the fiber delay lines for more than a certain number of times. Alternatively, one may take into consideration the limitation on the number of recirculations through the fiber delay lines during the design process. We have made some progresses on the constructions of optical 2-to-1 FIFO multiplexers with a limited number of recirculations, and the results will be reported later in a separate paper.

Finally, we mention that crosstalk interference is also a practical implementation issue of concern. We have made some progresses on the constructions of linear compressors with minimum crosstalk, and results along this line will be reported later in a separate paper.

IV. Recursive Constructions of Parallel LIFO Queues with a Shared Buffer

We have proposed a recursive construction in Figure 7 to construct parallel FIFO queues. One key condition that makes such a construction feasible is the constraint of FIFO order among the arriving packets. In this section, we will show that parallel LIFO queues can also be constructed using a similar architecture.

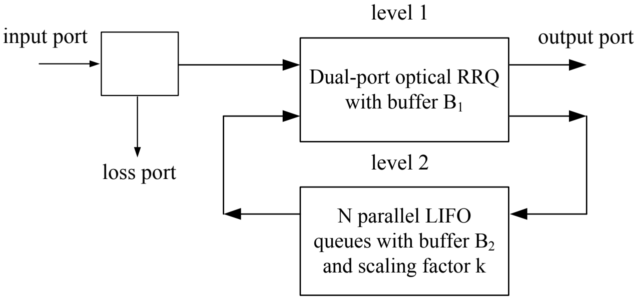
level 1
input port
output port
Dual-port optical RRQ with buffer B_1
loss port
level 2
N parallel LIFO queues with buffer B_2 and scaling factor k

Fig. 10. A recursive construction of N parallel LIFO queues with buffer $B_1 + kB_2$

The definition of N parallel LIFO queues is the same as that for N parallel FIFO queues except that packets depart in the Last In First Out (LIFO) order. In Figure 10, the construction consists of two levels: a dual-port RRQ with buffer B_1 in level 1, and a scaled SDL network element that can be used as a system of N parallel LIFO queues with a shared buffer B_2 and scaling factor k in level 2. The 1×2 switch in front of the network element is for admission control. Its objective is to make sure that the total number of packets inside the network element does not exceed $B_1 + kB_2$. An arriving packet can only be admitted if the total number of packets inside the network element does not exceed B_1+kB_2 after its admission. Otherwise, it is routed to the loss port.

We use the same notations as we did in the construction of parallel FIFO queues. Specifically, we let $Q_{\ell,i}(t)$, $\ell = 1$ and 2, $i = 1, 2, \cdots, N$, be the set of packets in the i^{th} queue that are stored in level ℓ at the end of the t^{th} time slot, and let $I_i(p, t)$ be the departure index of packet p at time t, i.e., $I_i(p, t) = j$ if packet p is the j^{th} packet to depart in the i^{th} queue at the end of the t^{th} time slot. Here, the departure index is labeled according to LIFO order. Moreover, as the operations for N parallel FIFO queues, the RRQ in level 1 is operated in every time slot, while the scaled N parallel LIFO queues in level 2 is operated in the time scale of frames. Note that the notations of front queues and tail queues are no longer needed, because under our operation rules the departure index of a packet stored in level 1 is always lower than that of any packet stored in level 2.

Now we present the operation rules of the recursive construction in Figure 10. The rule for admission control is the same as that in (R0). However, the write operation rule, the read operation rule, the retrieve operation rule, and the dump operation rule need to be modified as follows:

(LR1) Write operation: suppose that there is an arriving packet to the i^{th} queue at time t. If the i^{th} queue is enabled at time t, then the arriving packet is routed to the output port immediately. Otherwise, the arriving packet is stored in the RRQ in level 1 (as long as the total number of packets in the construction does not exceed $B_1 + kB_2$ after its admission).

(LR2) Read operation: suppose that the i^{th} queue is enabled at time t. If there is an arriving packet to the i^{th} queue at time t, then the arriving packet is routed to the output port immediately. If there is no arriving packet to the i^{th} queue at time t and the i^{th} queue has packets in level 1 at time $t-1$, the packet that has the smallest departure index among all the packets of the i^{th} queue in the RRQ in level 1 is sent to the output port at time t. Otherwise, there is no departing packet at time t.

(LR3) Retrieve operation (the shortest queue below the retrieving threshold): suppose that t is the beginning time slot of the m^{th} frame, i.e., $t = k(m - 1) + 1$. Consider the set of queues $R(t)$ that have packets in level 2 at time $t - 1$. Suppose that the i^{th} queue is the queue that has the smallest number of packets at time $t - 1$ among all the queues in $R(t)$. If the number of packets in the i^{th} queue at time $t - 1$ is less than or equal to the retrieving threshold R_T, i.e., $|Q_{1,i}(t-1)| \leq R_T$, then the i^{th} LIFO queue in level 2 is enabled during the m^{th} frame. As such, there are k packets retrieved from the i^{th} LIFO queue in level 2 to the i^{th} queue in level 1 in $[t, t + k - 1]$.

(LR4) Dump operation (the longest queue with a full block of packets): suppose that t is the beginning time slot of the m^{th} frame, i.e., $t = k(m-1)+1$. Suppose that the i^{th} queue is the queue that has the largest number of packets at time $t - 1$ among all the N queues. If there are at least $D_T + k$ packets in the i^{th} queue in level 1 at time $t - 1$, i.e., $|Q_{1,i}(t - 1)| \geq D_T + k$, then the k packets with the *largest* departure indices in the i^{th} queue in level 1 are sent (starting from the packet with the *smallest* departure index among these k packets) to the i^{th} LIFO queue in level 2 (as a block of packets) provided that there is buffer space in level 2 (i.e., either the buffer of the N LIFO queues in level 2 is not full at time $t-1$ or there is a retrieve operation at time t).

Now we state the main result for the construction of parallel LIFO queues in the following theorem. The proof of Theorem 9 is given in Appendix B.

Theorem 9 *Suppose the two-level recursive construction in Figure 10 is started from an empty system. If we choose* $R_T = \left\lceil 1 + k \sum_{\ell=1}^{N} \frac{1}{\ell} \right\rceil$, $D_T = R_T + k$, *and* $B_1 \geq ND_T + N(k-1) + k + 1$, *then under (R0) and (LR1)–(LR4) the construction in Figure 10 achieves the exact emulation of a system of* N *parallel LIFO queues with a shared buffer* $B_1 + B_2 k$.

The intuition for the choice of R_T and B_1 is the same as that in section Section III. Moreover, we can also expand the two-level construction in Theorem 9 to an n-level construction in Figure 11.

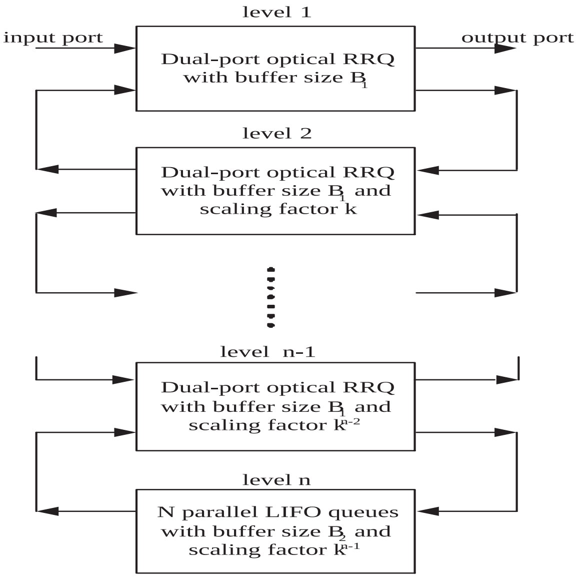
level 1
input port
output port
Dual-port optical RRQ with buffer size B1
level 2
Dual-port optical RRQ with buffer size B1 and scaling factor k
level n-1
Dual-port optical RRQ with buffer size B1 and scaling factor kn-2
level n
N parallel LIFO queues with buffer size B2 and scaling factor kn-1

Fig. 11. An n-level construction of N parallel LIFO queues

Since we use the same construction for parallel FIFO queues and parallel LIFO queues, the number of 2×2 switches needed for the two systems are the same. For a single LIFO queue with buffer size B (the case with $N=1$), the construction complexity is $O(\log B)$, which is better than $O(\sqrt{B})$ as given in [11][12] (we note that the constructions in [11][12] are more general and work for priority queues).

Moreover, as we do not need to distinguish between the front queue and the tail queue as in the construction for parallel FIFO queues, the control of the parallel LIFO queues is much easier than that for the parallel FIFO queues.

V. Conclusions

In this paper, we provide a new two-level recursive construction of a system of parallel FIFO (resp. LIFO) queues with a shared buffer. The key idea of our two-level construction is *caching*, where we have a dual-port RRQ in level 1 that acts as a high switching speed storage device and a system of scaled parallel FIFO (resp. LIFO) queues in level 2 that acts as a low switching speed storage device. By determining appropriate dumping thresholds and retrieving thresholds, we prove that the two-level cache can indeed be operated as a system of parallel FIFO (resp. LIFO) queues with a shared buffer.

We have shown that one of the advantages of our construction is its fault tolerant capability. By adding extra optical memory cells in each level, our construction still works even after some of the optical memory cells are broken. Furthermore, to construct a single LIFO queue with buffer size B, our construction only needs $O(\log B)$ 2×2 switches, which is sharper than $O(\sqrt{B})$ previously given in [11][12] (we note that the constructions in [11][12] are more general and work for priority queues).

There are some extensions that need to be further explored.

(i) N-port optical memory cells: for this paper, a dual-port optical RRQ in Figure 6 is constructed. Using the same architecture, an N-port optical RRQ can be constructed via a concatenation of N-port optical memory cells. It would be of interest to look for the efficient construction of an N-port optical memory cell.

(ii) N-to-1 multiplexer: the key condition to make our two-level recursive constrution feasible is the constraint of FIFO order or LIFO order among the arriving packets. Since an N-to-1 multiplexer has a similar constraint, is it possible to do the recursive construction of an N-to-1 multiplexer with an $(N+1)$-port optical RRQ in level 1 and a scaled N-to-1 multiplexer in level 2?

(iii) $N\times N$ output-buffered switch: based on the same reason of (ii), is it possible to do the recursive construction of an $N\times N$ output-buffered switch with a $2N$-port optical RRQ in level 1 and a scaled $N\times N$ output-buffered switch in level 2?

Appendix A
Proof of Theorem 8

In this appendix, we prove Theorem 8. The proof of Theorem 8 is based on the following three lemmas that bound the size of front queues and tail queues. In Lemma 10, we first show upper bounds for tail queues. We then show upper bounds for front queues in Lemma 11 and lower bounds for front queues in Lemma 12.

Lemma 10 *Suppose that* t *is the beginning time slot of a frame.*

(i) Suppose that no dump operation is performed at time t. *If either the buffer of the* N *parallel FIFO queues in level 2 is not full at time* $t-1$ *or there is a retrieve operation at time* t, *then* $\sum_{j=1}^{N} |T_j(t-1)| \leq N(k-1)$.

(ii) If the buffer of the N *parallel FIFO queues in level 2 is not full at time* $t-1$, *then* $\sum_{j=1}^{N} |T_j(t-1)| \leq N(k-1)+k$.

Proof. (i) Since we assume that either the buffer of the N parallel FIFO queues in level 2 is not full at time $t-1$ or there is a retrieve operation at time t, the only reason that no dump operation is performed at time t (see (R4)) is because the number of packets in every tail queue is less than k. The result then follows by summing all the packets in the N tail queues.

(ii) We prove this lemma by induction on time. Since the network element is started from an empty system, we have

$$\sum_{j=1}^{N}|T_j(0)| = 0 \leq N(k-1)+k.$$

Assume that the lemma holds at the beginning time slot of the $(m-1)^{th}$ frame as the induction hypothesis. We would like to show that this is also true at the beginning time slot of the m^{th} frame. Let t be the beginning time slot of the m^{th} frame, i.e., $t = k(m-1)+1$. There are two possible cases.

Case 1: The buffer of the N parallel FIFO queues in level 2 is full at time $t-k-1$. Since the buffer of the N parallel FIFO queues in level 2 is full at time $t-k-1$ and it is not full at time $t-1$, we know that there is no dump operation at time $t-k$ and there is a retrieve operation at time $t-k$. Thus, we have from (i) of this lemma that

$$\sum_{j=1}^{N}|T_j(t-k-1)| \leq N(k-1).$$

Since there are at most k arriving packets in a frame,

$$\begin{aligned}\sum_{j=1}^{N}|T_j(t-1)| &\leq \sum_{j=1}^{N}|T_j(t-k-1)|+k \\ &\leq N(k-1)+k.\end{aligned}$$

Case 2: The buffer of the N parallel FIFO queues in level 2 is not full at time $t-k-1$. If no dump operation is performed at time $t-k$, then the result follows from the same argument in Case 1.

Now suppose that there is a dump operation at time $t-k$. Then there are k packets that are sent from one of the tail queues in level 1 to the N parallel FIFO queues in level 2. Since there are at most k arriving packets in a frame, it then follows that

$$\begin{aligned}\sum_{j=1}^{N}|T_j(t-1)| &\leq \sum_{j=1}^{N}|T_j(t-k-1)|-k+k \\ &= \sum_{j=1}^{N}|T_j(t-k-1)|.\end{aligned}$$

The result then follows from the induction hypothesis. ■

Lemma 11 *(i) The number of packets in the i^{th} front queue is bounded above by the dumping threshold, i.e., $|F_i(t)| \leq D_T$ for all t.*

(ii) Suppose that t is the beginning time slot of a frame and a retrieve operation is performed at time t. Then

$$\sum_{i=1}^{N}|F_i(t-1)| \leq ND_T - k. \tag{3}$$

Proof. (i) We prove this by induction on t. As we assume the construction is started from an empty system. The inequality holds trivially for $t=0$. Suppose for some $t \geq 1$ that $|F_i(s)| \leq D_T$ for all $s \leq t-1$ as the induction hypothesis. Now we consider the following three cases:

Case 1: $|Q_{2,i}(t)| = 0$. In this case, the i^{th} queue does not have packets in level 2. The inequality holds trivially from the definition of a front queue in Definition 5.

Case 2: $|Q_{2,i}(t)| > 0$ and no retrieve operation is performed on the i^{th} queue in $[t-k+1,t]$. Note from Definition 6 that if there is an arriving packet to the i^{th} queue at time t, then this packet (if admitted) is added to the i^{th} tail queue in this case. As no retrieve operation is performed on the i^{th} queue in $[t-k+1,t]$, we know that at time t the number of packets in the i^{th} front queue cannot be increased. Thus, the inequality holds from the induction hypothesis.

Case 3: $|Q_{2,i}(t)| > 0$ and there is a retrieve operation on the i^{th} queue in $[t-k+1,t]$. Suppose that a retrieve operation is performed on the i^{th} queue at time τ in $[t-k+1,t]$. When this happens, we know from (R3) that the number of packets in the i^{th} front queue is less than or equal to R_T at time $\tau-1$. Since $D_T = R_T + k$ and there is at most one packet that can be retrieved to the i^{th} front queue in every time slot, it then follows that

$$\begin{aligned}|F_i(t)| &\leq |F_i(\tau-1)| + (t-\tau+1) \\ &\leq R_T + k = D_T.\end{aligned}$$

(ii) Without loss of generality, assume that a retrieve operation is performed at time t on the j^{th} queue. From (R3), we know that $|F_j(t-1)| \leq R_T = D_T - k$. As the rest of front queues are still bounded above by D_T, we then conclude that

$$\sum_{i=1}^{N}|F_i(t-1)| \leq ND_T - k,$$

and the proof is completed. ■

Lemma 12 *Suppose that t is the beginning time slot of a frame. Let $R(t)$ be the set of queues that have packets in level 2 at the end of the $(t-1)^{th}$ time slot. If U is a nonempty subset of $R(t)$, i.e., $U \subseteq R(t)$ and $|U| > 0$, then*

$$\sum_{j\in U}|F_j(t-1)| \geq |U|\left(1 + k\sum_{\ell=1}^{|U|-1}\frac{1}{\ell}\right), \tag{4}$$

with the convention that the sum on the RHS of (4) is 0 if the upper index is smaller than its lower index.

Proof. We prove this lemma by induction on time. Suppose the value of $|R(t)|$ changes from zero to one for the first time at time t_0 which is the beginning time slot of a frame. Therefore, a dump operation must have been performed for the first time at time $t_0 - k$. From (R4), the definition of a tail queue in Definition 6, and the definition of a front queue in Definition 5, we know that there is a queue, say the i^{th} queue, such that $|T_i(t_0-k-1)| \geq k$ and $|F_i(t_0-k-1)| = D_T$. As there are at most k packets that can depart during a frame,

$$\begin{aligned}\sum_{j\in R(t_0)} |F_j(t_0-1)| &= |F_i(t_0-1)| \\ &\geq |F_i(t_0-k-1)| - k \\ &= D_T - k = R_T \geq 1. \qquad (5)\end{aligned}$$

Since $|R(t_0)| = 1$, the only nonempty subset of $R(t_0)$ is itself. Thus, the lemma follows trivially from (5).

Assume that the inequality in (4) holds at some beginning time slot $t_1 \geq t_0$ of a frame as the induction hypothesis. Let U be a nonempty subset of $R(t_1+k)$. We need to consider the following two cases.

Case 1: $U \subseteq R(t_1)$. In this case, we know U is a nonempty subset of $R(t_1) \cap R(t_1+k)$. There are three subcases for this case.

Subcase (1a): A retrieve operation is performed at time t_1 on some queue in U. From the induction hypothesis, we have

$$\sum_{j\in U} |F_j(t_1-1)| \geq |U| \left(1 + k\sum_{\ell=1}^{|U|-1} \frac{1}{\ell}\right). \qquad (6)$$

Note that there are k packets retrieved from level 2 to some front queue in U during the frame and that there are at most k packets that can depart via the read operations during that frame. Thus,

$$\sum_{j\in U} |F_j(t_1+k-1)| \geq \sum_{j\in U} |F_j(t_1-1)| + k - k. \qquad (7)$$

From (6) and (7), it then follows that

$$\sum_{j\in U} |F_j(t_1+k-1)| \geq |U| \left(1 + k\sum_{\ell=1}^{|U|-1} \frac{1}{\ell}\right).$$

Subcase (1b): A retrieve operation is performed at time t_1 on some queue in $R(t_1)\backslash U$. For this subcase, we first show that

$$\sum_{j\in U} |F_j(t_1-1)| \geq |U| \left(1 + k\sum_{\ell=1}^{|U|-1} \frac{1}{\ell}\right) + k. \qquad (8)$$

Since there are at most k packets that can depart in a frame (k time slots), we then have from (8) that

$$\begin{aligned}\sum_{j\in U} |F_j(t_1+k-1)| &\geq \sum_{j\in U} |F_j(t_1-1)| - k \\ &\geq |U| \left(1 + k\sum_{\ell=1}^{|U|-1} \frac{1}{\ell}\right). \qquad (9)\end{aligned}$$

To show (8), suppose that a retrieve operation is performed at time t_1 on queue q in $R(t_1)\backslash U$. From (R3), we know that at time t_1-1 the number of packets in the q^{th} front queue is not greater than that of any other front queue on $R(t_1)$, i.e., $|F_q(t_1-1)| \leq |F_i(t_1-1)|$ for all $i \in R(t_1)$ and $i \neq q$. Thus, if $|F_q(t_1-1)| \geq 1 + k\sum_{\ell=1}^{|U|} \frac{1}{\ell}$, then

$$\begin{aligned}\sum_{j\in U} |F_j(t_1-1)| &\geq |U||F_q(t_1-1)| \\ &\geq |U| \left(1 + k\sum_{\ell=1}^{|U|} \frac{1}{\ell}\right) \\ &= |U| \left(1 + k\sum_{\ell=1}^{|U|-1} \frac{1}{\ell}\right) + k.\end{aligned}$$

On the other hand, if $|F_q(t_1-1)| < 1 + k\sum_{\ell=1}^{|U|} \frac{1}{\ell}$, we then have from the induction hypothesis that

$$\begin{aligned}&\sum_{j\in U} |F_j(t_1-1)| \\ &= \sum_{j\in U\cup\{q\}} |F_j(t_1-1)| - |F_q(t_1-1)| \\ &\geq (|U|+1) \left(1 + k\sum_{\ell=1}^{|U|} \frac{1}{\ell}\right) - |F_q(t_1-1)| \\ &= |U| \left(1 + k\sum_{\ell=1}^{|U|} \frac{1}{\ell}\right) + \left(1 + k\sum_{\ell=1}^{|U|} \frac{1}{\ell}\right) - |F_q(t_1-1)| \\ &> |U| \left(1 + k\sum_{\ell=1}^{|U|} \frac{1}{\ell}\right) = |U| \left(1 + k\sum_{\ell=1}^{|U|-1} \frac{1}{\ell}\right) + k.\end{aligned}$$

Subcase (1c): No retrieve operation is performed at time t_1. For this subcase, we also show that (8) holds. As there are at most k packets that can depart in a frame, we then derive the desired inequality in (9).

To show (8), we note from (R3) that at time t_1-1 the number of packets in every front queue in $R(t_1)$ is more than the retrieving threshold R_T because no retrieve operation is performed at time t_1. This implies that

$$\begin{aligned}\sum_{j\in U} |F_j(t_1-1)| &\geq |U|(R_T+1) \\ &\geq |U| \left(1 + k\sum_{\ell=1}^{N} \frac{1}{\ell}\right) + |U| \\ &\geq |U| \left(1 + k\sum_{\ell=1}^{|U|} \frac{1}{\ell}\right) \\ &= |U| \left(1 + k\sum_{\ell=1}^{|U|-1} \frac{1}{\ell}\right) + k.\end{aligned}$$

Case 2: $U \not\subseteq R(t_1)$. In this case, there is an element in U that is not in $R(t_1)$. Without loss of generality, assume that $q \in U$ and $q \notin R(t_1)$. Since $U \subseteq R(t_1+k)$, we know that $q \in R(t_1+k)$ and $q \notin R(t_1)$. Thus, a dump operation must have been performed on the q^{th} tail queue at time t_1. Moreover, by the definition of a tail queue in Definition 6 and the definition of a front queue in Definition 5, we have $|T_q(t_1-1)| \geq k$ and

$$|F_q(t_1-1)| = D_T = R_T + k. \qquad (10)$$

Let $\tilde{U} = U \backslash \{q\}$. If $\tilde{U}$ is an empty set, then $U = \{q\}$. As there are at most k packets that can depart during a frame, we have

$$\begin{aligned}\sum_{j \in U} |F_j(t_1 + k - 1)| &= |F_q(t_1 + k - 1)| \\ &\geq |F_q(t_1 - 1)| - k \\ &= R_T \geq 1,\end{aligned}$$

and the induction is completed. Therefore, in the following we assume that $\tilde{U}$ is nonempty.

Since a dump operation is already performed on the q^{th} tail queue at time t_1, no dump operation can be performed for any queue in $\tilde{U}$ at time t_1. As we assume that U is a nonempty subset of $R(t_1 + k)$, every queue in U has packets in level 2 at time $t_1 + k - 1$. Hence, we also know that every queue in $\tilde{U}$ has packets in level 2 at time $t_1 - 1$ because no dump operation is performed for any queue in $\tilde{U}$ at time t_1. Thus, we have $\tilde{U} \subseteq R(t_1) \cap R(t_1 + k)$ and all the property derived in Case 1 for U also hold for $\tilde{U}$ in this case.

We first note that in this case it suffices to show

$$\sum_{j \in U} |F_j(t_1 + k - 1)| \geq |\tilde{U}| \left(1 + k \sum_{\ell=1}^{|\tilde{U}|-1} \frac{1}{\ell}\right) + R_T + k. \quad (11)$$

This is because

$$\begin{aligned}|\tilde{U}| \left(1 + k \sum_{\ell=1}^{|\tilde{U}|-1} \frac{1}{\ell}\right) + R_T + k &= |\tilde{U}| \left(1 + k \sum_{\ell=1}^{|\tilde{U}|} \frac{1}{\ell}\right) + R_T \\ &\geq (|\tilde{U}| + 1) \left(1 + k \sum_{\ell=1}^{|\tilde{U}|} \frac{1}{\ell}\right) \\ &= |U| \left(1 + k \sum_{\ell=1}^{|U|-1} \frac{1}{\ell}\right).\end{aligned}$$

To show (11), we write

$$\sum_{j \in U} |F_j(t_1 - 1)| = \sum_{j \in \tilde{U}} |F_j(t_1 - 1)| + |F_q(t_1 - 1)|, \quad (12)$$

and consider the following three subcases as in Case 1.

Subcase (2a): A retrieve operation is performed at time t_1 on some queue in $\tilde{U}$. For this subcase, we have from the induction hypothesis that

$$\sum_{j \in \tilde{U}} |F_j(t_1 - 1)| \geq |\tilde{U}| \left(1 + k \sum_{\ell=1}^{|\tilde{U}|-1} \frac{1}{\ell}\right). \quad (13)$$

As there are k packets retrieved from level 2 to some front queue in $\tilde{U}$ during the frame and there are at most k packets that can depart during that frame, we have from (12), (13), and (10) that

$$\begin{aligned}\sum_{j \in U} |F_j(t_1 + k - 1)| &\geq \sum_{j \in U} |F_j(t_1 - 1)| + k - k \\ &\geq |\tilde{U}| \left(1 + k \sum_{\ell=1}^{|\tilde{U}|-1} \frac{1}{\ell}\right) + R_T + k.\end{aligned}$$

Subcase (2b): A retrieve operation is performed at time t_1 on some queue in $R(t_1) \backslash \tilde{U}$. We have from (8) in Case (1b) that

$$\sum_{j \in \tilde{U}} |F_j(t_1 - 1)| \geq |\tilde{U}| \left(1 + k \sum_{\ell=1}^{|\tilde{U}|-1} \frac{1}{\ell}\right) + k. \quad (14)$$

In conjunction with (12) and (10), it follows that

$$\sum_{j \in U} |F_j(t_1 - 1)| \geq |\tilde{U}| \left(1 + k \sum_{\ell=1}^{|\tilde{U}|-1} \frac{1}{\ell}\right) + R_T + 2k. \quad (15)$$

Since there are at most k packets that can depart in a frame, we have from (15) that

$$\begin{aligned}\sum_{j \in U} |F_j(t_1 + k - 1)| &\geq \sum_{j \in U} |F_j(t_1 - 1)| - k \\ &\geq |\tilde{U}| \left(1 + k \sum_{\ell=1}^{|\tilde{U}|-1} \frac{1}{\ell}\right) + R_T + k.\end{aligned}$$

Subcase (2c):No retrieve operation is performed at time t_1. As in Case (1c), we still have (14). The rest of the proof for (11) then follows from the same argument in Case (2b). ■

Proof. (Proof of Theorem 8) To prove that our construction indeed achieves the exact emulation of N parallel FIFO queues with a shared buffer $B_1 + kB_2$, we need to show the following four properties.

(P1) Flow conservation: The flow conservation property is satisfied trivially for the write operation, the read operation, and the dump operation because both the RRQ in level 1 and the system of N parallel FIFO queues in level 2 also satisfy the flow conservation property. The only problem is whether there is always a buffer space in level 1 for a packet retrieved from the N parallel FIFO queues in level 2. To show this, suppose that a retrieve operation is performed on the i^{th} queue at the beginning time slot t of the m^{th} frame, i.e., $t = k(m-1)+1$. Consider the following two cases:

Case 1: A dump operation is also performed at time t. As there is a packet dumped from level 1 to level 2 during each time slot of the m^{th} frame, there is always a buffer space for a packet retrieved from the i^{th} queue in level 2.

Case 2: No dump operation is performed at time t. Since a retrieve operation is performed on the i^{th} queue at time t, it follows from Lemma 11(ii) that

$$\sum_{j=1}^{N} |F_j(t - 1)| \leq ND_T - k.$$

As there is no dump operation at time t, we have from Lemma 10(i) that

$$\sum_{j=1}^{N} |T_j(t - 1)| \leq N(k - 1).$$

Thus,

$$\begin{aligned}\sum_{j=1}^{N} |Q_{1,j}(t - 1)| &= \sum_{j=1}^{N} |F_j(t - 1)| + \sum_{j=1}^{N} |T_j(t - 1)| \\ &\leq B_1 - 2k - 1,\end{aligned}$$

where we use the fact that $F_j(t-1)$ and $T_j(t-1)$ are disjoint, $Q_{1,j} = F_j(t-1) \cup T_j(t-1)$, and $B_1 \geq ND_T + N(k-1) + k + 1$. Since there are at most k arriving packets during the m^{th} frame and there are at least $2k+1$ unoccupied buffer spaces in level 1 at time $t-1$, we conclude that there is always one buffer space in level 1 for every packet retrieved from the i^{th} queue in level 2 during the m^{th} frame.

(P2) Non-idling: We prove this property by contradiction. Suppose the non-idling property is violated for the first time at time t for some queue i. Without loss of generality, assume that t is within the m^{th} frame, i.e., $k(m-1)+1 \leq t < km+1$ for some $m \in \mathbb{N}$. Let $t_0 = k(m-1)+1$ be the beginning time slot of the m^{th} frame. When this happens, we know that there are packets of queue i in level 2 at time $t-1$ and queue i in level 1 is empty at time $t-1$. This implies that $i \in R(t_0)$ and $|F_i(t-1)| = 0$. As there is at most one departure in a time slot and $t - t_0 < k$, we also know that

$$|F_i(t_0 - 1)| < k. \tag{16}$$

From Lemma 12, it follows that $|F_i(t_0-1)| \geq 1$ and thus $t \neq t_0$. As such, we have $t_0 < t < t_0 + k$.

Consider the following three cases.

Case 1: A retrieve operation is performed on queue i at time t_0. In this case, there is a packet retrieved from queue i in level 2 to its front queue from t_0 to t. As there is at most one packet departure in a time slot and $|F_i(t_0-1)| \geq 1$, it follows that $|F_i(t-1)| \geq 1$. This contradicts to $|F_i(t-1)| = 0$.

Case 2: A retrieve operation is performed on some queue $j \neq i$ at time t_0. According to (R3), we know that $j \in R(t_0)$ and $|F_j(t_0-1)| \leq |F_i(t_0-1)|$. Since $|F_i(t_0-1)| < k$ in (16), we have

$$|F_j(t_0-1)| + |F_i(t_0-1)| \leq 2|F_i(t_0-1)| < 2k.$$

On the other hand, we know from Lemma 12 that $|F_j(t_0 - 1)| + |F_i(t_0-1)| \geq 2(1+k)$. Thus, we reach a contradiction.

Case 3: No retrieve operation is performed at time t_0. From (R3), we know that at time $t_0 - 1$ the number of packets in every front queue in $R(t_0)$ is not less than or equal to R_T. Thus,

$$|F_i(t_0-1)| \geq R_T + 1 \geq k+1.$$

This contradicts to (16).

(P3) Maximum buffer usage: We prove this property by contradiction. Suppose the property of maximum buffer usage is violated for the first time at time t. Without loss of generality, assume that t is within the m^{th} frame, i.e., $k(m-1)+1 \leq t < km+1$ for some $m \in \mathbb{N}$. Let $t_0 = k(m-1)+1$ be the beginning time slot of the m^{th} frame. When this happens, the read operation is not performed at time t and no packet departs at time t. Moreover, we know that the buffer in the RRQ in level 1 is full at time $t-1$, i.e., $|Q_1(t-1)| = B_1$, and the buffer in the system of N parallel FIFO queues in level 2 is not full at time $t-1$, i.e., $|Q_2(t-1)| < B_2 k$. Consider the following two cases.

Case 1: The buffer in the system of N parallel FIFO queues in level 2 is full at time $t_0 - 1$. Since the buffer in the system of N parallel FIFO queues in level 2 is not full at time $t-1$, we know in this case that a retrieve operation is performed at time t_0 and no dump operation is performed at time t_0. We have from Lemma 10(i) that

$$\sum_{j=1}^{N} |T_j(t_0-1)| \leq N(k-1). \tag{17}$$

Also, since a retrieve operation is performed at time t_0, we have from Lemma 11(ii) that

$$\sum_{j=1}^{N} |F_j(t_0-1)| \leq ND_T - k. \tag{18}$$

From (17) and (18), it follows that

$$\begin{aligned} |Q_1(t_0-1)| &= \sum_{j=1}^{N} |F_j(t_0-1)| + \sum_{j=1}^{N} |T_j(t_0-1)| \\ &\leq ND_T - k + N(k-1) \\ &\leq B_1 - 2k - 1. \end{aligned}$$

Since the number of packets in the dual-port RRQ in level 1 can be increased by at most 2 packets in a time slot and $t - t_0 < k$,

$$\begin{aligned} |Q_1(t-1)| &\leq |Q_1(t_0-1)| + 2(t-t_0) \\ &\leq B_1 - 2k - 1 + 2(t-t_0) \\ &< B_1 - 1. \end{aligned}$$

We reach a contradiction to $|Q_1(t-1)| = B_1$.

Case 2: The buffer in the system of N parallel FIFO queues in level 2 is not full at time $t_0 - 1$. There are four subcases in this case.

Subcase (2a): A dump operation is performed and no retrieve operation is performed at t_0. By Lemma 10(ii),

$$\sum_{j=1}^{N} |T_j(t_0-1)| \leq N(k-1) + k.$$

On the other hand, we have from Lemma 11(i) that

$$\sum_{j=1}^{N} |F_j(t_0-1)| \leq ND_T.$$

As there are $t - t_0$ packets dumped from level 1 to level 2 and there are at most $t - t_0$ arrivals in $[t_0, t-1]$

$$\begin{aligned} |Q_1(t-1)| &\leq |Q_1(t_0-1)| - (t-t_0) + (t-t_0) \\ &= \sum_{j=1}^{N} |F_j(t_0-1)| + \sum_{j=1}^{N} |T_j(t_0-1)| \\ &\leq ND_T + N(k-1) + k \\ &< B_1 - 1. \end{aligned}$$

We reach a contradiction to $|Q_1(t-1)| = B_1$.

Subcase (2b): A dump operation is performed and a retrieve operation is performed at time t_0. By Lemma 10(ii),

$$\sum_{j=1}^{N} |T_j(t_0-1)| \leq N(k-1) + k.$$

As there is a retrieve operation at time t_0, we have from Lemma 11(ii) that

$$\sum_{j=1}^{N} |F_j(t_0 - 1)| \le ND_T - k.$$

Now there are $t - t_0$ packets dumped from level 1 to level 2, $t - t_0$ packets retrieved from level 2 to level 1, and at most $t - t_0$ packets arriving in $[t_0, t-1]$. Thus,

$$\begin{aligned}|Q_1(t-1)| &\le |Q_1(t_0-1)| - (t-t_0) + (t-t_0) + (t-t_0)\\ &= \sum_{j=1}^{N} |F_j(t_0-1)| + \sum_{j=1}^{N} |T_j(t_0-1)| + (t-t_0)\\ &\le ND_T - k + N(k-1) + k + (t-t_0)\\ &\le B_1 - 1 - k + (t-t_0)\\ &< B_1 - 1.\end{aligned}$$

We reach a contradiction to $|Q_1(t-1)| = B_1$.

Subcase (2c): No dump operation is performed and no retrieve operation is performed at time t_0. Since no dump operation is performed at time t_0, we have from Lemma 10(i) that

$$\sum_{j=1}^{N} |T_j(t_0 - 1)| \le N(k-1).$$

On the other hand, we have from Lemma 11(i) that

$$\sum_{j=1}^{N} |F_j(t_0 - 1)| \le ND_T.$$

Since there are at most $t - t_0$ arrivals in $[t_0, t-1]$

$$\begin{aligned}|Q_1(t-1)| &\le |Q_1(t_0-1)| + (t-t_0)\\ &= \sum_{j=1}^{N} |F_j(t_0-1)| + \sum_{j=1}^{N} |T_j(t_0-1)| + (t-t_0)\\ &\le ND_T + N(k-1) + (t-t_0)\\ &\le B_1 - 1 - k + (t-t_0)\\ &< B_1 - 1.\end{aligned}$$

We reach a contradiction to $|Q_1(t-1)| = B_1$.

Subcase (2d): No dump operation is performed and a retrieve operation is performed at time t_0. Since no dump operation is performed at time t_0, we have from Lemma 10(i) that

$$\sum_{j=1}^{N} |T_j(t_0 - 1)| \le N(k-1).$$

As there is a retrieve operation at time t_0, we have from Lemma 11(ii) that

$$\sum_{j=1}^{N} |F_j(t_0 - 1)| \le ND_T - k.$$

Since there are $t - t_0$ packets retrieved from level 2 to level 1 and there are at most $t - t_0$ arrivals in $[t_0, t-1]$

$$\begin{aligned}|Q_1(t-1)| &\le |Q_1(t_0-1)| + (t-t_0) + (t-t_0)\\ &= \sum_{j=1}^{N} |F_j(t_0-1)| + \sum_{j=1}^{N} |T_j(t_0-1)| + 2(t-t_0)\\ &\le ND_T - k + N(k-1) + 2(t-t_0)\\ &\le B_1 - 1 - 2k + 2(t-t_0) < B_1 - 1.\end{aligned}$$

We reach a contradiction to $|Q_1(t-1)| = B_1$.

(P4) FIFO: The FIFO property is guaranteed because we always choose the packet with the smallest departure index to depart from the RRQ in level 1 (see the read operation in (R2)). ∎

Appendix B
Proof of Theorem 9

In this appendix, we prove Theorem 9. In Lemma 13, we first show upper bounds for queues in level 1. We then show lower bounds for queues in level 1 in Lemma 14. The results of these two lemmas are then used to prove Theorem 9.

Lemma 13 *Suppose that t is the beginning time slot of a frame and U is a subset of N queues, i.e., $U \subseteq \{1, 2, \ldots, N\}$.*

(i) Suppose that no dump operation is performed at time t. If either the buffer of the N parallel LIFO queues in level 2 is not full at time $t-1$ or there is a retrieve operation at time t, then $\sum_{j \in U} |Q_{1,j}(t-1)| \le |U|D_T + |U|(k-1)$.

(ii) If the buffer of the N parallel LIFO queues in level 2 is not full at time $t-1$, then

$$\sum_{j \in U} |Q_{1,j}(t-1)| \le |U|D_T + N(k-1) + k. \tag{19}$$

Proof. This lemma holds trivially if U is an empty set, so in the following we assume that U is nonempty.

(i) Since we assume that either the buffer of the N parallel LIFO queues in level 2 is not full at time $t-1$ or there is a retrieve operation at time t, the only reason that no dump operation is performed at time t (see (LR4)) is because the number of packets in each queue in level 1 is less than $D_T + k$. The result then follows by summing all the packets in level 1 of the queues in set U.

(ii) We prove this lemma by induction on time. Since the network element is started from an empty system, for any nonempty subset U of N queues we have

$$\sum_{j \in U} |Q_{1,j}(0)| = 0 \le |U|D_T + N(k-1) + k.$$

Assume that the inequality in (19) holds at some beginning time slot t of a frame as the induction hypothesis. We would like to show that this is true at the beginning time slot $t+k$. There are two possible cases:

Case 1: No retrieve operation is performed on any queue in U at time t. In this case, we first show that

$$\sum_{j \in U} |Q_{1,j}(t-1)| \le |U|D_T + N(k-1) + d_U(t)k, \tag{20}$$

where $d_U(t)$ is an indicator variable with $d_U(t)=1$ if a dump operation is performed on some queue in U at time t and 0 otherwise.

Since there are at most k arriving packets in a frame (k time slots) and $d_U(t)k$ packets dumped from level 1 to level 2 in $[t, t+k-1]$, we have from (20) that

$$\sum_{j\in U}|Q_{1,j}(t+k-1)| \le \sum_{j\in U}|Q_{1,j}(t-1)| + k - d_U(t)k \le |U|D_T + N(k-1) + k. \quad (21)$$

There are four subcases for the inequality in (20).

Subcase (1a): A dump operation is performed at time t on some queue in U and the buffer of the N parallel LIFO queues in level 2 is not full at time $t-1$. To show (20), note that $d_U(t)=1$ in this subcase. It then follows from the induction hypothesis that

$$\sum_{j\in U}|Q_{1,j}(t-1)| \le |U|D_T + N(k-1) + k.$$

Subcase (1b): A dump operation is performed at time t on some other queue that is not in U and the buffer of the N parallel LIFO queues in level 2 is not full at time $t-1$. For this subcase, we have $d_U(t)=0$. Therefore, we need to show that

$$\sum_{j\in U}|Q_{1,j}(t-1)| \le |U|D_T + N(k-1). \quad (22)$$

Suppose that a dump operation is performed on queue $q\notin U$ at time t. From (LR4), we know

$$|Q_{1,q}(t-1)| \ge D_T + k. \quad (23)$$

Also from induction hypothesis, we have

$$\sum_{j\in U\cup\{q\}}|Q_{1,j}(t-1)| \le (|U|+1)D_T + N(k-1) + k. \quad (24)$$

It then follows from (23) and (24) that

$$\sum_{j\in U}|Q_{1,j}(t-1)| \le \sum_{j\in U\cup\{q\}}|Q_{1,j}(t-1)| - |Q_{1,q}(t-1)| \le (|U|+1)D_T + N(k-1) + k - D_T - k = |U|D_T + N(k-1).$$

Subcase (1c): No dump operation is performed at time t and the buffer of the N parallel LIFO queues in level 2 is not full at time $t-1$. For this subcase, we also show that (22) holds. It follows from (i) of this lemma that

$$\sum_{j\in U}|Q_{1,j}(t-1)| \le |U|D_T + |U|(k-1) \le |U|D_T + N(k-1).$$

Subcase (1d): The buffer of the N parallel LIFO queues in level 2 is full at time $t-1$. Since the buffer of the N parallel LIFO queues in level 2 is full at time $t-1$ and it is not full at time $t+k-1$, we know in this case that no dump operation is performed at time t. Using (i) of this lemma, we can show that (22) still holds as in Case (1c).

Case 2: A retrieve operation is performed at time t on some queue in U. Suppose that a retrieve operation is performed at time t on queue q in U. From (LR3), we know that

$$|Q_{1,q}(t-1)| \le R_T = D_T - k. \quad (25)$$

Let $\tilde{U} = U\backslash\{q\}$. If $\tilde{U}$ is empty, then $U=\{q\}$ and hence from (25) we have

$$\sum_{j\in U}|Q_{1,j}(t-1)| = |Q_{1,q}(t-1)| \le D_T - k \le |U|D_T + N(k-1) + k.$$

Therefore, in the following we assume that $\tilde{U}$ is not empty.

Since no retrieve operation is performed at time t for any queue in $\tilde{U}$, we have from (20) that

$$\sum_{j\in\tilde{U}}|Q_{1,j}(t-1)| \le |\tilde{U}|D_T + N(k-1) + d_{\tilde{U}}(t)k. \quad (26)$$

Therefore, from (25) and (26), we have

$$\sum_{j\in U}|Q_{1,j}(t-1)| = \sum_{j\in\tilde{U}}|Q_{1,j}(t-1)| + |Q_{1,q}(t-1)| \le |\tilde{U}|D_T + N(k-1) + d_{\tilde{U}}(t)k + D_T - k. \quad (27)$$

Moreover, we can show that $d_U(t) = d_{\tilde{U}}(t)$. As $\tilde{U} = U\backslash\{q\}$, the only case that $d_U(t)\ne d_{\tilde{U}}(t)$ is when a dump operation is performed on queue q. But this is not possible because of (25) and (LR4).

Now, for the set U, there are $d_U(t)k$ packets dumped from level 1 to level 2, k packets retrieved from level 2 to level 1, and at most k packets arriving in $[t, t+k-1]$. Thus, from (27)

$$\begin{aligned}\sum_{j\in U}|Q_{1,j}(t+k-1)| &\le \sum_{j\in U}|Q_{1,j}(t-1)| - d_U(t)k + k + k \\ &= \sum_{j\in U}|Q_{1,j}(t-1)| - d_{\tilde{U}}(t)k + k + k \\ &\le |\tilde{U}|D_T + N(k-1) + D_T + k \\ &= |U|D_T + N(k-1) + k,\end{aligned}$$

and the proof is completed. ■

Lemma 14 *Suppose that t is the beginning time slot of a frame. Let $R(t)$ be the set of queues that have packets in level 2 at the end of the $(t-1)^{th}$ time slot. If U is a nonempty subset of $R(t)$, i.e., $U\subseteq R(t)$ and $|U|>0$, then*

$$\sum_{j\in U}|Q_{1,j}(t-1)| \ge |U|\left(1 + k\sum_{\ell=1}^{|U|-1}\frac{1}{\ell}\right). \quad (28)$$

Proof. We prove this lemma by induction on time. Suppose the value of $|R(t)|$ changes from zero to one for the first time at time t_0 which is the beginning time slot of a frame. Therefore, a dump operation must have been performed for the first time at time t_0-k. From (LR4), we know that there is a queue, say the i^{th} queue, such that $|Q_{1,i}(t_0-k-1)| \geq D_T+k$. As there are k packets dumped from level 1 to level 2 and there are at most k packets that can depart during a frame,

$$\begin{aligned}\sum_{j\in R(t_0)} |Q_{1,j}(t_0-1)| &= |Q_{1,i}(t_0-1)| \\ &\geq |Q_{1,i}(t_0-k-1)| - k - k \\ &\geq D_T - k = R_T \geq 1. \end{aligned} \tag{29}$$

Since $|R(t_0)|=1$, the only nonempty subset of $R(t_0)$ is itself. Thus, the inequality in (28) follows trivially from (29).

Assume that the inequality in (28) holds at some beginning time slot $t_1 \geq t_0$ of a frame as the induction hypothesis. We would like to show that this is true at the beginning time slot t_1+k. Let U be a nonempty subset of $R(t_1+k)$. We need to consider the following three cases.

Case 1: $U \subseteq R(t_1)$ and no dump operation is performed at time t_1 on any queue in U. In this case, U is a nonempty subset of $R(t_1) \cap R(t_1+k)$. We first show that

$$\sum_{j\in U} |Q_{1,j}(t_1-1)| \geq |U| \left(1+k\sum_{\ell=1}^{|U|-1}\frac{1}{\ell}\right) + k - r_U(t_1)k, \tag{30}$$

where $r_U(t_1)$ is an indicator variable with $r_U(t_1)=1$ if a retrieve operation is performed on some queue in U at time t_1 and 0 otherwise.

Since there are at most k packets that can depart in a frame (k time slots) and $r_U(t_1)k$ packets retrieved from level 2 to level 1 in $[t_1, t_1+k-1]$, we then have from (30) that

$$\begin{aligned}\sum_{j\in U} |Q_{1,j}(t_1+k-1)| &\geq \sum_{j\in U} |Q_{1,j}(t_1-1)| + r_U(t_1)k - k \\ &\geq |U| \left(1+k\sum_{\ell=1}^{|U|-1}\frac{1}{\ell}\right). \end{aligned} \tag{31}$$

There are three subcases for the inequality in (30).

Subcase (1a): A retrieve operation is performed at time t_1 on some queue in U. To show (30), note that $r_U(t_1)=1$. It then follows from the induction hypothesis that

$$\sum_{j\in U} |Q_{1,j}(t_1-1)| \geq |U| \left(1+k\sum_{\ell=1}^{|U|-1}\frac{1}{\ell}\right).$$

Subcase (1b): A retrieve operation is performed at time t_1 on some queue in $R(t_1)\backslash U$. For this subcase, we have $r_U(t_1)=0$. Therefore, we need to show that

$$\sum_{j\in U} |Q_{1,j}(t_1-1)| \geq |U| \left(1+k\sum_{\ell=1}^{|U|-1}\frac{1}{\ell}\right) + k. \tag{32}$$

To show (32), suppose that a retrieve operation is performed at time t_1 on queue q in $R(t_1)\backslash U$. From (LR3), we know that at time t_1 the number of packets in the q^{th} queue is not greater than that of any other queue in $R(t_1)$, i.e., $|Q_{1,q}(t_1-1)| \leq |Q_{1,i}(t_1-1)|$ for all $i \in R(t_1)$ and $i \neq q$. Thus, if $|Q_{1,q}(t_1-1)| \geq 1+k\sum_{\ell=1}^{|U|}\frac{1}{\ell}$, then

$$\begin{aligned}\sum_{j\in U} |Q_{1,j}(t_1-1)| &\geq |U||Q_{1,q}(t_1-1)| \\ &\geq |U| \left(1+k\sum_{\ell=1}^{|U|}\frac{1}{\ell}\right) \\ &= |U| \left(1+k\sum_{\ell=1}^{|U|-1}\frac{1}{\ell}\right) + k. \end{aligned}$$

On the other hand, if $|Q_{1,q}(t_1-1)| < 1+k\sum_{\ell=1}^{|U|}\frac{1}{\ell}$, we then have from the induction hypothesis that

$$\begin{aligned}&\sum_{j\in U} |Q_{1,j}(t_1-1)| \\ &= \sum_{j\in U\cup\{q\}} |Q_{1,j}(t_1-1)| - |Q_{1,q}(t_1-1)| \\ &\geq (|U|+1)\left(1+k\sum_{\ell=1}^{|U|}\frac{1}{\ell}\right) - |Q_{1,q}(t_1-1)| \\ &= |U|\left(1+k\sum_{\ell=1}^{|U|}\frac{1}{\ell}\right) + \left(1+k\sum_{\ell=1}^{|U|}\frac{1}{\ell}\right) - |Q_{1,q}(t_1-1)| \\ &> |U|\left(1+k\sum_{\ell=1}^{|U|}\frac{1}{\ell}\right) = |U|\left(1+k\sum_{\ell=1}^{|U|-1}\frac{1}{\ell}\right) + k. \end{aligned}$$

Subcase (1c): No retrieve operation is performed at time t_1. For this subcase, we also show that (32) holds. To show (32), we note from (LR3) that the number of packets in every queue is more than the retrieving threshold R_T because no retrieve operation is performed at time t_1. This implies that

$$\begin{aligned}\sum_{j\in U} |Q_{1,j}(t_1-1)| &\geq |U|(R_T+1) \\ &\geq |U|\left(1+k\sum_{\ell=1}^{N}\frac{1}{\ell}\right) + |U| \\ &\geq |U|\left(1+k\sum_{\ell=1}^{|U|}\frac{1}{\ell}\right) \\ &= |U|\left(1+k\sum_{\ell=1}^{|U|-1}\frac{1}{\ell}\right) + k. \end{aligned}$$

Case 2: $U \subseteq R(t_1)$ and a dump operation is performed at time t_1 on some queue in U. Suppose that a dump operation is performed at time t_1 on some queue $q \in U$ From (LR4), we know

$$|Q_{1,q}(t_1-1)| \geq D_T + k. \tag{33}$$

Let $\tilde{U} = U\backslash\{q\}$. If $\tilde{U}$ is empty, then $U=\{q\}$. As there are k packets dumped from level 1 to level 2 and at most k packets that can depart during a frame, we have

$$\begin{aligned}\sum_{j\in U} |Q_{1,j}(t_1+k-1)| &= |Q_{1,q}(t_1+k-1)| \\ &\geq |Q_{1,q}(t_1-1)| - k - k \\ &\geq R_T \geq 1. \end{aligned}$$

So in the following, we assume that $\tilde{U}$ is not empty.

As no dump operation is performed at t_1 for any queue in $\tilde{U}$, we have from (30) that

$$\sum_{j\in\tilde{U}} |Q_{1,j}(t_1-1)| \geq |\tilde{U}| \left(1 + k\sum_{\ell=1}^{|\tilde{U}|-1} \frac{1}{\ell}\right) + k - r_{\tilde{U}}(t_1)k. \tag{34}$$

Using (33) and (34) yields

$$\begin{aligned}
&\sum_{j\in U} |Q_{1,j}(t_1-1)| \\
&= \sum_{j\in\tilde{U}} |Q_{1,j}(t_1-1)| + |Q_{1,q}(t_1-1)| \\
&\geq |\tilde{U}| \left(1 + k\sum_{\ell=1}^{|\tilde{U}|-1} \frac{1}{\ell}\right) + k - r_{\tilde{U}}(t_1)k + D_T + k \\
&= |\tilde{U}| \left(1 + k\sum_{\ell=1}^{|\tilde{U}|-1} \frac{1}{\ell}\right) + R_T + 3k - r_{\tilde{U}}(t_1)k, \qquad (35)
\end{aligned}$$

where we use $D_T = R_T + k$ in the last equality.

Moreover, we can show that $r_U(t_1) = r_{\tilde{U}}(t_1)$. As $\tilde{U} = U\backslash\{q\}$, the only case that $r_U(t_1) \neq r_{\tilde{U}}(t_1)$ is when a retrieve operation is performed on queue q. But this is not possible because of (33) and (LR3).

Now, for the set U, there are $r_U(t_1)k$ packets retrieved from level 2 to level 1, k packets dumped from level 1 to level 2, and at most k packets departing in $[t_1, t_1+k-1]$. Thus, we have from $r_U(t_1) = r_{\tilde{U}}(t_1)$ and (35) that

$$\begin{aligned}
\sum_{j\in U} |Q_{1,j}(t_1+k-1)| &\geq \sum_{j\in U} |Q_{1,j}(t_1-1)| + r_U(t_1)k - k - k \\
&\geq |\tilde{U}| \left(1 + k\sum_{\ell=1}^{|\tilde{U}|-1} \frac{1}{\ell}\right) + R_T + k \\
&= |\tilde{U}| \left(1 + k\sum_{\ell=1}^{|\tilde{U}|} \frac{1}{\ell}\right) + R_T \\
&\geq (|\tilde{U}|+1) \left(1 + k\sum_{\ell=1}^{|\tilde{U}|} \frac{1}{\ell}\right) \\
&= |U| \left(1 + k\sum_{\ell=1}^{|U|-1} \frac{1}{\ell}\right).
\end{aligned}$$

Case 3: $U \not\subseteq R(t_1)$. In this case, there is an element in U that is not in $R(t_1)$. Without loss of generality, assume that $q \in U$ and $q \notin R(t_1)$. Since $U \subseteq R(t_1+k)$, we know that $q \in R(t_1+k)$ and $q \notin R(t_1)$. Thus, a dump operation must have been performed on the q^{th} queue at time t_1 and

$$|Q_{1,q}(t_1-1)| \geq D_T + k. \tag{36}$$

Let $\tilde{U} = U\backslash\{q\}$. If $\tilde{U}$ is empty, then $U = \{q\}$. As there are k packets dumped from level 1 to level 2 and at most k packets that can depart during a frame, we have

$$\begin{aligned}
\sum_{j\in U} |Q_{1,j}(t_1+k-1)| &= |Q_{1,q}(t_1+k-1)| \\
&\geq |Q_{1,q}(t_1-1)| - k - k \\
&\geq R_T \geq 1.
\end{aligned}$$

So in the following we assume that $\tilde{U}$ is not empty. We first show $\tilde{U} \subseteq R(t_1)\cap R(t_1+k)$. Since a dump operation is already performed on the q^{th} queue at time t_1, no dump operation can be performed for any queue in $\tilde{U}$ at time t_1. As we assume that U is a nonempty subset of $R(t_1+k)$, every queue in U has packets in level 2 at t_1+k-1. Hence, we also know that every queue in $\tilde{U}$ has packets in level 2 at t_1-1 because no dump operation is performed for any queue in $\tilde{U}$ at time t_1. Thus, we have $\tilde{U} \subseteq R(t_1)\cap R(t_1+k)$.

Since no dump operation is performed for any queue in $\tilde{U}$ at time t_1, we still have (34). In view of (36) and (34), the rest of the proof in this case is the same as that in Case 2. ■

Proof. (Proof of Theorem 9) To prove that our construction indeed achieves the exact emulation of N parallel LIFO queues with a shared buffer B_1+kB_2, we need to show the following four properties.

(P1) Flow conservation: As discussed in the proof of N parallel FIFO queues, the only problem is whether there is always a buffer space in level 1 for a packet retrieved from the N parallel LIFO queues in level 2. To show this, suppose that a retrieve operation is performed on the i^{th} queue at the beginning time slot t of the m^{th} frame, i.e., $t = k(m-1)+1$. Consider the following two cases:

Case 1: A dump operation is also performed at time t. As there is a packet dumped from level 1 to level 2 during each time slot of the m^{th} frame, there is always a buffer space for a packet retrieved from the i^{th} queue in level 2.

Case 2: No dump operation is performed at time t. Since a retrieve operation is performed on the i^{th} queue, it follows from (LR3) that

$$|Q_{1,i}(t-1)| \leq R_T = D_T - k.$$

Also by Lemma 13(i),

$$\sum_{j\in\{1,2,\cdots,N\}\backslash\{i\}} |Q_{1,j}(t-1)| \leq (N-1)D_T + (N-1)(k-1).$$

Thus,

$$\begin{aligned}
&\sum_{j=1}^{N} |Q_{1,j}(t-1)| \\
&= \sum_{j\in\{1,2,\cdots,N\}\backslash\{i\}} |Q_{1,j}(t-1)| + |Q_{1,i}(t-1)| \\
&\leq ND_T + (N-1)(k-1) - k \\
&= ND_T + N(k-1) - (k-1) - k \\
&\leq B_1 - 3k,
\end{aligned}$$

where we use the fact that $B_1 \geq ND_T + N(k-1) + k + 1$. Since there are at most k arriving packets during the m^{th} frame and there are at least $3k$ unoccupied buffer spaces in level 1 at time $t-1$, we conclude that there is always one buffer space in level 1 for every packet retrieved from the i^{th} queue in level 2 during the m^{th} frame.

(P2) Non-idling: We prove this property by contradiction. Suppose the non-idling property is violated for the first time at time t for some queue i. Without loss of generality, assume that t is within the m^{th} frame, i.e., $k(m-1)+1 \le t < km+1$ for some $m \in \mathbb{N}$. Let $t_0 = k(m-1)+1$ be the beginning time slot of the m^{th} frame. When this happens, we know that there are packets of queue i in level 2 at time $t-1$ and queue i in level 1 is empty at time $t-1$. This implies that $i \in R(t_0)$ and $|Q_{1,i}(t-1)| = 0$. As there is at most one departure in a time slot and $t - t_0 < k$, we also know that

$$|Q_{1,i}(t_0-1)| < k. \tag{37}$$

From Lemma 14, it follows that $|Q_{1,i}(t_0-1)| \ge 1$ and thus $t \ne t_0$. As such, we have $t_0 < t < t_0 + k$.

Consider the following three cases.

Case 1: A retrieve operation is performed on queue i at time t_0. In this case, there is a packet retrieved from queue i in level 2 to its queue in level 1 from t_0 to t. As there is at most one packet departure in a time slot and $|Q_{1,i}(t_0-1)| \ge 1$, it follows that $|Q_{1,i}(t-1)| \ge 1$. This contradicts to $|Q_{1,i}(t-1)| = 0$.

Case 2: A retrieve operation is performed on some queue $j \ne i$ at time t_0. According to (LR3), we know that $j \in R(t_0)$ and $|Q_{1,j}(t_0-1)| \le |Q_{1,i}(t_0-1)|$. Since $|Q_{1,i}(t_0-1)| < k$ in (37), we have

$$|Q_{1,j}(t_0-1)| + |Q_{1,i}(t_0-1)| \le 2|Q_{1,i}(t_0-1)| < 2k.$$

On the other hand, we know from Lemma 14 that $|Q_{1,j}(t_0-1)| + |Q_{1,i}(t_0-1)| \ge 2(1+k)$. Thus, we reach a contradiction.

Case 3: No retrieve operation is performed at time t_0. From (LR3), we know that at time t_0-1 the number of packets in every queue in $R(t_0)$ is not less than or equal to R_T. Thus,

$$|Q_{1,i}(t_0-1)| \ge R_T + 1 \ge k+1.$$

This contradicts to (37).

(P3) Maximum buffer usage: We prove this property by contradiction. Suppose the property of maximum buffer usage is violated for the first time at time t. Without loss of generality, assume that t is within the m^{th} frame, i.e., $k(m-1)+1 \le t < km+1$ for some $m \in \mathbb{N}$. Let $t_0 = k(m-1)+1$ be the beginning time slot of the m^{th} frame. When this happens, no read operation is performed at time t and no packet departs at time t. Moreover, we know that the buffer in the RRQ in level 1 is full at time $t-1$, i.e.,

$$|Q_1(t-1)| = \sum_{j=1}^{N} |Q_{1,j}(t-1)| = B_1,$$

and the buffer in the system of N parallel LIFO queues in level 2 is not full at time $t-1$, i.e.,

$$|Q_2(t-1)| = \sum_{j=1}^{N} |Q_{2,j}(t-1)| < B_2 k.$$

Consider the following two cases.

Case 1: The buffer in the system of N parallel LIFO queues in level 2 is full at time t_0-1. Since the buffer in the system of N parallel LIFO queues in level 2 is not full at time $t-1$, we know in this case that a retrieve operation is performed at time t_0 on some queue q and no dump operation is performed at time t_0. We have from Lemma 13(i) that

$$\sum_{j\in\{1,2,\cdots,N\}\backslash\{q\}} |Q_{1,j}(t_0-1)| \le (N-1)D_T + (N-1)(k-1). \tag{38}$$

Also, since a retrieve operation is performed at time t_0 on queue q, we have from (LR3) that

$$|Q_{1,q}(t_0-1)| \le R_T = D_T - k. \tag{39}$$

From (38) and (39), it follows that

$$\begin{aligned}
&|Q_1(t_0-1)| \\
&= \sum_{j\in\{1,2,\cdots,N\}\backslash\{q\}} |Q_{1,j}(t_0-1)| + |Q_{1,q}(t_0-1)| \\
&\le ND_T + N(k-1) - 2k + 1 \le B_1 - 3k.
\end{aligned}$$

Since the number of packets in the dual-port RRQ in level 1 can be increased by at most 2 packets in a time slot and $t - t_0 < k$,

$$\begin{aligned}
|Q_1(t-1)| &\le |Q_1(t_0-1)| + 2(t-t_0) \\
&\le B_1 - 3k + 2(t-t_0) \\
&< B_1 - k \le B_1 - 1.
\end{aligned}$$

We reach a contradiction to $|Q_1(t-1)| = B_1$.

Case 2: The buffer in the system of N parallel LIFO queues in level 2 is not full at time t_0-1. There are four subcases in this case.

Subcase(2a): A dump operation is performed and no retrieve operation is performed at time t_0. By Lemma 13(ii),

$$\sum_{j=1}^{N} |Q_{1,j}(t_0-1)| \le ND_T + N(k-1) + k.$$

As there are $t-t_0$ packets dumped from level 1 to level 2 and there are at most $t-t_0$ arrivals in $[t_0, t-1]$

$$\begin{aligned}
|Q_1(t-1)| &\le |Q_1(t_0-1)| - (t-t_0) + (t-t_0) \\
&\le ND_T + N(k-1) + k \le B_1 - 1.
\end{aligned}$$

We reach a contradiction to $|Q_1(t-1)| = B_1$.

Subcase(2b): A dump operation is performed and a retrieve operation is performed at time t_0. Suppose that a retrieve operation is performed at time t_0 on queue q. By Lemma 13(ii)

$$\sum_{j\in\{1,2,\cdots,N\}\backslash\{q\}} |Q_{1,j}(t_0-1)| \le (N-1)D_T + N(k-1) + k.$$

As a retrieve operation is performed on queue q at time t_0, we have from (LR3) that

$$|Q_{1,q}(t_0-1)| \le R_T = D_T - k.$$

Now there are $t-t_0$ packets dumped from level 1 to level 2, $t-t_0$ packets retrieved from level 2 to level 1, and at most

$t-t_0$ packets arriving in $[t_0, t-1]$. Thus,

$$\begin{aligned}
&|Q_1(t-1)| \\
&\le |Q_1(t_0-1)| - (t-t_0) + (t-t_0) + (t-t_0) \\
&= \sum_{j\in\{1,2,\cdots,N\}\setminus\{q\}} |Q_{1,j}(t_0-1)| + |Q_{1,q}(t_0-1)| + (t-t_0) \\
&\le (N-1)D_T + N(k-1) + k + D_T - k + (t-t_0) \\
&= ND_T + N(k-1) + (t-t_0) \\
&\le B_1 - 1 - k + (t-t_0) < B_1 - 1.
\end{aligned}$$

We reach a contradiction to $|Q_1(t-1)| = B_1$.

Subcase(2c): No dump operation is performed and no retrieve operation is performed at time t_0. Since no dump operation is performed at time t_0, we have from Lemma 13(i) that

$$\sum_{j=1}^{N} |Q_{1,j}(t_0-1)| \le ND_T + N(k-1).$$

Since there are at most $t-t_0$ arrivals in $[t_0, t-1]$

$$\begin{aligned}
|Q_1(t-1)| &\le |Q_1(t_0-1)| + (t-t_0) \\
&\le ND_T + N(k-1) + (t-t_0) \\
&\le B_1 - 1 - k + (t-t_0) < B_1 - 1.
\end{aligned}$$

We reach a contradiction to $|Q_1(t-1)| = B_1$.

Subcase(2d): No dump operation is performed and a retrieve operation is performed at time t_0. Suppose that a retrieve operation is performed at time t_0 on queue q. Since no dump operation is performed at time t_0, we have from Lemma 13(i) that

$$\sum_{j\in\{1,2,\cdots,N\}\setminus\{q\}} |Q_{1,j}(t_0-1)| \le (N-1)D_T + (N-1)(k-1).$$

As a retrieve operation is performed at time t_0 on queue q, we have from (LR3) that

$$|Q_{1,q}(t_0-1)| \le R_T = D_T - k.$$

Since there are $t-t_0$ packets retrieved from level 2 to level 1 and there are at most $t-t_0$ arrivals in $[t_0, t-1]$

$$\begin{aligned}
&|Q_1(t-1)| \\
&\le |Q_1(t_0-1)| + (t-t_0) + (t-t_0) \\
&= \sum_{j\in\{1,2,\cdots,N\}\setminus\{q\}} |Q_{1,j}(t_0-1)| + |Q_{1,q}(t_0-1)| + 2(t-t_0) \\
&\le ND_T - k + (N-1)(k-1) + 2(t-t_0) \\
&\le B_1 - 3k + 2(t-t_0) < B_1 - k \le B_1 - 1.
\end{aligned}$$

We reach a contradiction to $|Q_1(t-1)| = B_1$.

(P4) LIFO: The LIFO property is guaranteed because we always choose the packet with the smallest departure index to depart from the RRQ in level 1 (see the read operation in (LR2)). ■

References

[1] M. J. Karol, "Shared-memory optical packet (ATM) switch," in *Proceedings SPIE vol. 2024: Multigigabit Fiber Communication Systems (1993)*, October 1993, pp. 212–222.

[2] I. Chlamtac, A. Fumagalli, L. G. Kazovsky, P. Melman, W. H. Nelson, P. Poggiolini, M. Cerisola, A. N. M. M. Choudhury, T. K. Fong, R. T. Hofmeister, C.-L. Lu, A. Mekkittikul, D. J. M. Sabido IX, C.-J. Suh, and E. W. M. Wong, "Cord: contention resolution by delay lines," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 1014–1029, June 1996.

[3] I. Chlamtac, A. Fumagalli, and C.-J. Suh, "Multibuffer delay line architectures for efficient contention resolution in optical switching nodes," *IEEE Transactions on Communications*, vol. 48, pp. 2089–2098, December 2000.

[4] J. T. Tsai, "COD: architectures for high speed time-based multiplexers and buffered packet switches," Ph.D. Dissertation, University of California, San Diego, CA, USA, 1995.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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," *IEEE Transactions on Information Theory*, vol. 52, pp. 4519–4531, October 2006.

[10] 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.

[11] 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.

[12] 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.

[13] 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," in *Proceedings of IEEE 25th Annual Conference on Computer Communications (INFOCOM'06)*, Barcelona, Spain, April 23–29, 2006.

[14] D. K. Hunter, W. D. Cornwell, T. H. Gilfedder, A. Franzen, and I. Andonovic, "SLOB: a switch with large optical buffers for packet switching," *IEEE Journal of Lightwave Technology*, vol. 16, pp. 1725–1736, October 1998.

[15] E. A. Varvarigos, "The "packing" and the "scheduling packet" switch architectures for almost all-optical lossless networks," *IEEE Journal of Lightwave Technology*, vol. 16, pp. 1757–1767, October 1998.

[16] N. McKeown, "Scheduling algorithms for input-queued cell switches," Ph.D. Dissertation, University of California at Berkeley, Berkeley, CA, USA, 1995.

[17] Y. Li, S. Panwar, and H. J. Chao, "On the performance of a dual round-robin switch," in *Proceedings IEEE 20th Annual Conference on Computer Communications (INFOCOM'01)*, Anchorage, AK, USA, April 22–26, 2001, pp. 1688–1697.

[18] 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, April 2002.

[19] I. Keslassy, S.-T. Chung, K. Yu, D. Miller, M. Horowitz, O. Slogaard, and N. McKeown, "Scaling Internet routers using optics," in *Proceedings ACM Special Interest Group on Data Communication (SIGCOMM'03)*, Karlsruhe, Germany, August 25–29, 2003.

[20] J. L. Hennessy, D. A. Patterson, *Computer Organization and Design*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997.

[21] H. R. Gail, G. Grover, R. Guerin, S. Hantler, Z. Rosberg , and M. Sidi, "Buffer size requirements under longest queue first," *Performance Evaluation*, vol. 18, pp. 133–140, September 1993.

[22] S. Iyer, R. R. Kompella, and N. McKeown, “Designing packet buffers for router linecards,” *Stanford University HPNG Technical Report* - TR02-HPNG-031001, Stanford, CA, Mar. 2002. also submitted to *IEEE/ACM Transactions on Networking*.

[23] A. Bouillard and C.-S. Chang, “Optical FIFO queues and non-overtaking delay lines: construction and control,” *Technical Report*, Institute of Communications Engineering, National Tsing Hua University, August 2006.

[24] C. J. Chang-Hasnain, P. C. Ku, J. Kim, and S. L. Chuang, “Variable optical buffer using slow light in semiconductor nanostructure,” *Proceedings of the IEEE*, vol. 9, pp. 1884–1897, November 2003.

[25] M. R. Fisher, S. Minin, and S. L. Chuang, “Tunable optical group delay in an active waveguide semiconductor resonator,” *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 11, pp. 197–203, January/February 2005.

[26] Y. Okawachi, M. S. Bigelow, J. E. Sharping, Z. Zhu, A. Schweinsberg, D. J. Gauthier, R.W. Boyd, and A. L. Gaeta, “Tunable all-optical delays via Brillouin slow light in an optical fiber,” *Physical Review Letters*, vol. 94, 153902, April 2005.

[27] D. Dahan and G. Eisenstein, “Tunable all optical delay via slow and fast light propagation in a Raman assisted fiber optical parametric amplifier: a route to all optical buffering,” *Optics Express*, vol. 13, pp. 6234–6249, August 2005.

[28] Z. Zhu, A. M. C. Dawes, D. J. Gauthier, L. Zhang, and A. E. Willner, “12-GHz-Bandwidth SBS Slow Light in Optical Fibers,” in *Proceedings International Conference on Optical Fiber Communications (OFC’06)*, Anaheim, CA, USA, March 5–10, 2006, PDP1.

[29] A. Dupas, L. Billes, J. C. Simon, B. Landousies, M. Henry, I. Valiente, F. Ratovelomanana, A. Enard, and N. Vodjdani, “2R all-optical regenerator assessment at 2.5 Gbit/s over 3600 km using only standard fibre,” *IEE Electronics Letters*, vol. 34, pp. 2424–2425, December 1998.

[30] G. Lakshminarayana, R.-M. Mu, H. Wang, B. Stefanov, B. Dave, and J. Sarathy, “Integrated performance monitoring, performance maintenance, and failure detection for photonic regenerators,” in *Proceedings 31st European Conference on Optical Communication (ECOC’05)*, Glasgow, Scotland, September 25–29, 2005, pp. 609–610.

[31] M. R. G. Leiria and A. V. T. Cartaxo, “On the optimization of regenerator parameters in a chain of 2R all-optical regenerators,” *IEEE Photonics Technology Letters*, vol. 18, pp. 1711–1713, August 2006.

[32] C. Dragone, “An $N \times N$ optical multiplexer using a planar arrangement of two star couplers,” *IEEE Photonics Technology Letters*, vol. 3, pp. 812–815, September 1991.

[33] W. Lin, H. Li, Y. J. Chen, M. Dagenais, and D. Stone, “Dual-channel-spacing phased-array waveguide grating multi/demultiplexers,” *IEEE Photonics Technology Letters*, vol. 8, pp. 1501–1503, November 1996.

Po-Kai Huang received the B.S. and M.S. degrees from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 2004 and 2006, respectively, all in Electrical Engineering. He is now doing his military service at the Coast Guard Administration of the Republic of China.

Cheng-Shang Chang (S’85-M’86-M’89-SM’93-F’04) received the B.S. degree from National Taiwan University, Taipei, Taiwan, R.O.C., in 1983, and the M.S. and Ph.D. degrees from Columbia University, New York, NY, USA, 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, NY, USA. Since 1993, he has been with the Department of Electrical Engineering at National Tsing Hua University, Hsinchu, Taiwan, R.O.C., where he is currently 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, R.O.C., 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.

Jay Cheng (S’00-M’03) received the B.S. and M.S. degrees from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1993 and 1995, respectively, and the Ph.D. degree from Cornell University, Ithaca, NY, USA, in 2003, all in Electrical Engineering. In August 2003, he joined the Department of Electrical Engineering at National Tsing Hua University, Hsinchu, Taiwan, R.O.C., as an Assistant Professor. Since October 2004, he has been with the Department of Electrical Engineering and the Institute of Communications Engineering at National Tsing Hua University, Hsinchu, Taiwan, R.O.C., where he is currently an Assistant Professor. His current research interests include communications theory, optical queueing theory, information theory, and quantum information theory.

Duan-Shin Lee (S’89-M’90-SM’98) received the B.S. degree from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1983, and the MS and Ph.D. degrees from Columbia University, New York, NY, USA, in 1987 and 1990, respectively, all in Electrical Engineering. He worked as a research staff member at the C&C Research Laboratory of NEC USA, Inc., Princeton, NJ, USA, from 1990 to 1998. He joined the Department of Computer Science at National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1998. Since August 2003, he has been a Professor. His research interests are high-speed switch and router design, wireless networks, performance analysis of communication networks and queueing theory.