Constructions of Multicast Flexible Delay Lines and Optical Multicast Switches with 100% Throughput

Tsz-Hsuan Chao, Cheng-Shang Chang, Duan-Shin Lee and Jay Cheng
Institute of Communications Engineering
National Tsing Hua University
Hsinchu 300, Taiwan, R.O.C.
Email: thchao@gibbs.ee.nthu.edu.tw
cschang@ee.nthu.edu.tw
lds@cs.nthu.edu.tw
jcheng@ee.nthu.edu.tw

***Abstract*— Optical queues, usually constructed by optical Switches and fiber Delay Lines (SDL), are the key elements for conflict resolution in optical packet switching. It is recently shown in [3] that several optical queues constructed by SDL elements are indeed infinite dimensional switches in *time* and they can be constructed by many classical constructions in the switching theory. In particular, a (unicast) flexible delay line in [3] is a discrete-time infinite-server queue that corresponds to the *nonblocking* switch in the switching theory, and it can be constructed either by the three-stage Clos network or the Cantor network. In this paper, we propose two new constructions for *multicast* flexible delay lines that use the unicast flexible delay lines as the basic construction elements. The first one is constructed by using parallel unicast flexible delay lines. It is shown that a multicast flexible delay line with maximum delay d can be constructed by using $O(\sqrt{d})$ unicast flexible delay lines with maximum delay d. Our second construction is a recursive construction. We show that a multicast flexible delay line with maximum delay $2d-1$ can be constructed by two unicast flexible delay lines with maximum delay $d-1$ and a multicast flexible delay line with maximum delay $d-1$. As an application, we show that multicast flexible delay lines can be used for the constructions of optical multicast switches with 100% throughput.**

I. Introduction

One of the fundamental problems of all optical packet switching is the lack of inexpensive optical buffers needed for confliction resolution. The only known way to store optical packets without converting them into other media is to direct optical packets via a set of Switches and fiber Delay Lines (SDL) so that optical packets can then come out at the right place and at the right time. By so doing, a SDL element behaves as if it were an optical queue. Such an approach for constructing optical queues was originally proposed for experimental optical networks in [23], [13], and it has received a great deal of attention recently. See e.g., [16], [19], [6], [7] for FIFO multiplexers, [2], [27] for FIFO queues, [29], [11], [12] for priority queues, [17] for LIFO queues, [30] for input-buffered switches, and [20], [10] for output-buffered switches.

As the state space for a SDL element is in general very large, the most challenging problem for the SDL approach (despite its other technical difficulties for practical implementation, e.g., crosstalks and power losses) is to find the appropriate control of the switches in a SDL element. Recent advances in this area (see e.g., [2], [3]) have found that many discrete-time queues are indeed infinite dimensional switches in time, and the classical switching theory (see e.g., [18], [21], [26], [8]) can be used for the constructions of various optical queues with SDL elements. In particular, a (unicast) flexible delay line in [3] is a discrete-time infinite-server queue that allows an incoming packet to be delayed by an arbitrary number of time slots as long as it does not collide with any other packets at the output. Analogous to the classical switching theory, a flexible delay line may be viewed as a *nonblocking* switch in *time*, and can then be constructed either by the three-stage Clos network [15] or by the (modified) Cantor network [1]. Moreover, it was shown in [3] that flexible delay lines can further be used as *buffers* in the constructions of output-buffered switches. As such, the construction complexity of an output-buffered switch depends on how efficiently flexible delay lines are constructed. For a flexible delay line with maximum delay d, it was shown in [3] that the construction complexity of the three-stage Clos network (in terms of the number of 2×2 switches) is $O((\log d)^\gamma)$, where $\gamma = \log_2 10 \approx 3.321928$. On the other hand, the construction complexity for the modified Cantor network is $O((\log d)^2)$.

In this paper, we are interested in extending the construction of *unicast* flexible delay lines to multicast flexible delay lines so that they can in turn be used for the construction of multicast switches. As in the most SDL constructions in the literature, we consider the discrete-time setting, i.e., $t = 0, 1, 2, \ldots$ This is usually done assuming that time in every (optical) link is slotted and synchronized. Also, we assume that packets are of the same size and a packet can be transmitted within a time slot in a link. In the following, we first generalize the definition of a *unicast* flexible delay line in [3] to a *multicast* flexible delay line.

This research was supported in part by the National Science Council, Taiwan, R.O.C., under Contract NSC-94-2213-E-007-046, Contract NSC-95-2221-E-007-039, and the Program for Promoting Academic Excellence of Universities NSC 94-2752-E-007-002-PAE.

Definition 1 (Multicast flexible delay line) *A multicast flexible delay line is a network element with one input link and one output link. Let c_n be the number of multicast copies of the n^{th} packet, $\tau^a(n)$ be the arrival time of the n^{th} packet at the input link, and $\tau^d(n,i)$, $i = 1, 2, \ldots, c_n$, be the departure time of the i^{th} copy of the n^{th} packet at the output link. Suppose that the departure time of each copy of a packet is known upon its arrival. A* multicast *flexible delay line with the maximum delay d realizes the set of mappings (or sample paths) that satisfy*

$$\tau^a(n) \leq \tau^d(n,i) \leq \tau^a(n) + d, \text{ for all } n \text{ and } i \leq c_n, \quad (1)$$

$$\tau^d(m,i) \neq \tau^d(n,j), \quad \text{for all } m \neq n \text{ or } i \neq j, \quad (2)$$

and

$$\tau^d(n,i) < \tau^d(n,i+1), \quad i \leq c_n. \quad (3)$$

The first condition in (1) requires that the delay of every packet is bounded above by d. The second condition in (2) is usually known as the external nonblocking condition (for space switches) that requires the departure times of all packets to be distinct. The third condition in (3) shows that the copies of the same packet should depart in the order of their indices. Note that when $c_n = 1$ for all n, the condition in (3) is not needed and the definition for a multicast flexible delay line is reduced to that for a unicast flexible delay line in [3]. Analogous to the classical switching theory, one may view a multicast flexible delay line as a nonblocking switch with multicasting in time. As long as the "input/output" condition in (1) and the external nonblocking conditions in (2) and (3) are satisfied, there is always a "path" from an "input" to an "output."

In Section II and Section III, we will propose two constructions of multicast flexible delay lines that use the unicast flexible delay lines as the basic construction elements. In Section IV, we show that multicast flexible delay lines can be used for the constructions of optical multicast switches with 100% throughput. The paper is concluded in Section V by addressing some research problems that require further study.

II. A CONSTRUCTION BY PARALLEL UNICAST FLEXIBLE DELAY LINES

Our first construction for a multicast flexible delay line with maximum delay d is to use parallel unicast flexible delay lines with maximum delay d. In Figure 1, we consider a network element that consists of m parallel unicast flexible delay lines with maximum delay d. When a packet arrives, it is routed to one of the m parallel unicast flexible delay lines through the $1 \times m$ switch (demultiplexer) in front of the network element. In front of each unicast flexible delay line, there is a copy element. For an arriving packet with c copies, it is duplicated in a copy element for c consecutive time slots (with a copy in each time slot). Specifically, if the packet arrives at time t, then there is a copy of the packet departing from the copy element at time $t, t+1, \ldots, t+c-1$. These c copies of the packet are then fed contiguously to a unicast flexible delay line with maximum delay d so that they can depart from the unicast flexible delay line at their designated departure times. The m outputs of the m unicast flexible delay lines are finally multiplexed by an $m \times 1$ switch at the end of the network element. As the departure times of all packets are required to be distinct in (2) for a multicast flexible delay line, there is no collision at the output.

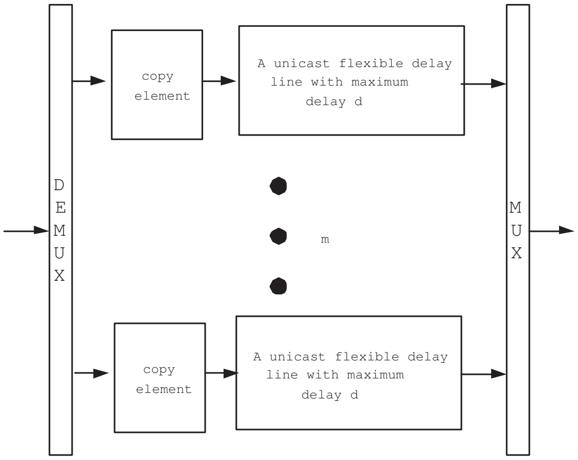
D
E
M
U
X
copy element
A unicast flexible delay line with maximum delay d
m
copy element
A unicast flexible delay line with maximum delay d
M
U
X

Fig. 1. A construction by parallel unicast flexible delay lines

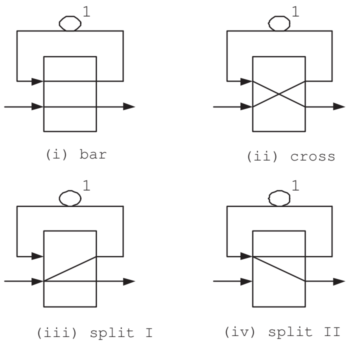

Fig. 2. A copy element and its four connection patterns

We note that a copy element can be easily constructed by a 2×2 multicast switch and a fiber delay line with one unit of delay (see Figure 2). The 2×2 multicast switch has four connection patterns: bar, cross, split I (from the lower input) and split II (from the upper input). To see how it works, consider a packet that arrives at a copy element in Figure 2 at time t. If the packet is a unicast packet, then the 2×2 switch is set to the "bar" state at time t. On the other hand, if the packet needs c copies (for some $c > 1$), the 2×2 switch is set to the "split I" state at time t so that one copy of the packet is routed directly to the output and the other copy is routed to the fiber delay line with one unit of delay. From time $t+1$ to time $t+c-2$, the 2×2 switch is set to the "split II" state so that one copy is routed to the output and the other copy is routed back to the fiber delay line with one unit of delay. At time $t+c-1$, the 2×2 switch is set to the "cross" state and the packet in the fiber delay line is routed to the output.

Once a copy element generates the number of copies needed for an arriving packet, it becomes free and can be used by another packet. For the network element to be nonblocking, it suffices to guarantee that there is always at least one free copy element when a packet arrives. The problem can be mapped to the queueing context by viewing each copy element as a "server" and each arriving packet as a "customer." The service time of a customer is the number of copies of the corresponding packet. The question is then to find the minimum number of servers so that there is no blocking in such a queue with parallel servers. Intuitively, if the number of copy elements is large enough, then every arriving packet can be routed to one of the free copy elements. In particular, if the number of copies is bounded, i.e., $c_n \leq c$ for some constant c, then one only needs c copy elements. However, if the bound c is the same as the maximum delay d, then one might expect that the number of copy elements is $O(d)$. In the following, we will show that one only needs $O(\sqrt{d})$ copy elements. As such, one only needs $O(\sqrt{d})$ unicast flexible delay lines.

Theorem 2 *Suppose that the network element in Figure 1 is started from an empty system. If*

$$m > \frac{\sqrt{8d+1}-1}{2}, \tag{4}$$

then it is a multicast flexible delay line with maximum delay d.

Proof. We prove this by contradiction. Suppose that an arriving packet is blocked at time t_0, i.e., all the m copy elements are being used at time t_0. Let $t_i, i = 1, \ldots, m$, be the arrival times of the m packets in the m copy elements. As there is at most one arrival in a time slot, the arrival times of these m packets and the packet arriving at t_0 must be distinct. Without loss of generality, we may simply assume that

$$t_m < t_{m-1} < \ldots < t_1 < t_0. \tag{5}$$

For the packet that arrives at t_i, $i = 1, 2, \ldots, m$, there are at least $t_0 - t_i + 1$ copies of that packet as the copy element generates a copy every time slot and it is not free at time t_0. From (1) in the definition of a multicast flexible delay line with maximum delay d, these packets must depart in the interval $[t_i, t_i + d]$. In view of (5), we also know that $[t_i, t_i + d]$ is a subinterval of $[t_m, t_0 + d]$. Thus, the total number of packets that must depart in the interval $[t_m, t_0 + d]$ is at least

$$1 + \sum_{i=1}^{m} (t_0 - t_i + 1),$$

adding the packet that arrives at time t_0. Since the departure time of every packet is required to be distinct in (2), the total number of packets that can depart in the interval $[t_m, t_0 + d]$ is at most $t_0 + d - t_m + 1$. This then leads to

$$1 + \sum_{i=1}^{m} (t_0 - t_i + 1) \leq t_0 + d - t_m + 1, \tag{6}$$

or equivalently

$$\sum_{i=0}^{m-1} (t_0 - t_i + 1) \leq d, \tag{7}$$

Note from (5) that $t_0 - t_i \geq i$. Using this in (7) yields

$$1 + 2 + \ldots + m \leq d. \tag{8}$$

This implies that $m \leq \frac{\sqrt{8d+1}-1}{2}$ and we reach a contradiction to (4). ■

III. A Recursive Construction

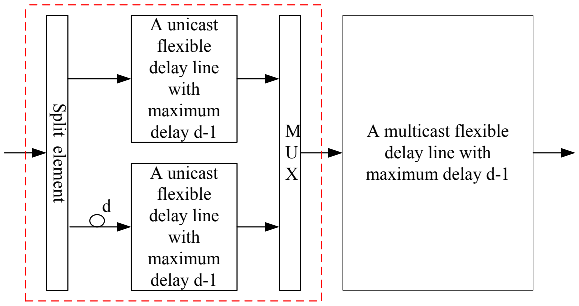
Split element
A unicast flexible delay line with maximum delay d-1
d
A unicast flexible delay line with maximum delay d-1
M U X
A multicast flexible delay line with maximum delay d-1

Fig. 3. A recursive construction of a multicast flexible delay line with maximum delay $2d - 1$

Our second construction is a recursive construction. In Figure 3, we show that a multicast flexible delay line with maximum delay $2d - 1$ can be constructed by two unicast flexible delay lines with maximum delay $d - 1$ and a multicast flexible delay line with maximum delay $d - 1$. The idea is to decompose the multicast copies of a packet into two disjoint sets: one with delay less than d and the other with delay larger than or equal to d. The latter after passing through a fixed fiber delay line with delay d also contains a multicast set with delay less than d. Thus, both of them can then be realized by a multicast flexible delay line with maximum delay $d - 1$. The only problem left is to make sure that there is no collision at the input of the multicast flexible delay line with maximum delay $d - 1$. For this problem, our trick is to use the property that the departure times of all packets are distinct in (2).

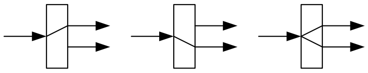

Fig. 4. The three connection patterns of the split element

To see how the construction in Figure 3 works, consider the n^{th} packet that satisfies

$$\tau^a(n) \leq \tau^d(n, i) \leq \tau^a(n) + 2d - 1, \text{ for all } n \text{ and } i \leq c_n, \tag{9}$$

and the two conditions in (2) and (3). Let z_n be the number of multicast copies of the n^{th} packet that has delay less than d. If $z_n = 0$, then the set of multicast copies with delay less than d is an empty set. On the other hand, if $z_n = c_n$, then the set of multicast copies with delay larger than or equal to d is an empty set. When the n^{th} packet arrives at $\tau^a(n)$, it is first sent to a split element. A split element is a 1×2 switch that has the three connection patterns as shown in Figure 4. If $z_n = c_n$, then the leftmost connection in Figure 4 is used as the set of multicast copies with delay larger than or equal to d is an empty set. If $z_n = 0$, then the middle connection in Figure 4 is used as the set of multicast copies with delay less than d is an empty set. Otherwise, the rightmost connection is used and the multicast set of the n^{th} packet is decomposed into two *nonempty* sets. By so doing, we guarantee that if there is a packet at one of the two outputs of the split element at $\tau^a(n)$, then that packet contains a *nonempty* multicast subset of the n^{th} packet.

Suppose that there is a packet at the *upper* output of the split element at $\tau^a(n)$. As discussed in the last paragraph, it contains z_n ($z_n > 0$) copies of the n^{th} packet with delay less than d. Note from (3) that $\tau^d(n,1)$ is the earliest departure time among the z_n copies of the n^{th} packet. We then send this packet through a unicast flexible delay lines with maximum delay $d-1$ with the designated departure time $\tau^d(n,1)$. After this, the z_n copies of the n^{th} packet have the remaining delays $\tau^d(n,i) - \tau^d(n,1)$, $i = 1, 2, \ldots, z_n$. As the remaining delays of these z_n copies of the n^{th} packet are still less than d, they can be realized by a multicast flexible delay line with maximum delay $d-1$ at the last stage of Figure 3. Also, the packet that "carries" these z_n copies of the n^{th} packet arrives at the multicast flexible delay line with maximum delay $d-1$ at $\tau^d(n,1)$ and it will not collide with any other packets as the departure times of all the packets are distinct in (2).

On the other hand, suppose that there is a packet at the *lower* output of the split element at $\tau^a(n)$. It contains $c_n - z_n$ ($c_n - z_n > 0$) copies of the n^{th} packet with delay greater than or equal to d. Note from (3) and (9) that $\tau^d(n, z_n+1)$ is the earliest departure time among the $c_n - z_n$ copies of the n^{th} packet, and that

$$\tau^a(n) + d \leq \tau^d(n, z_n+1) \leq \tau^a(n) + 2d - 1. \qquad (10)$$

We then send this packet through a fixed delay line with delay d, and then a unicast flexible delay lines with maximum delay $d-1$ with the designated departure time $\tau^d(n, z_n+1)$. This is feasible as we have from (10) that

$$0 \leq \tau^d(n, z_n+1) - \tau^a(n) - d \leq d - 1.$$

After this, the $c_n - z_n$ copies of the n^{th} packet have the remaining delays $\tau^d(n,i) - \tau^d(n, z_n+1)$, $i = z_n+1, \ldots, c_n$. As $\tau^d(n, z_n+1) \geq \tau^a(n) + d$ in (10) and $\tau^d(n,i) \leq \tau^a(n) + 2d - 1$ in (9), we still have $\tau^d(n,i) - \tau^d(n, z_n+1) \leq d-1$. Thus, these $c_n - z_n$ copies of the n^{th} packet can be realized by a multicast flexible delay line with maximum delay $d-1$ at the last stage of Figure 3. Also, the packet that "carries" these $c_n - z_n$ copies of the n^{th} packet arrives at the multicast flexible delay line with maximum delay $d-1$ at $\tau^d(n, z_n+1)$ and it will not collide with any other packets as the departure times of all the packets are distinct in (2).

To compute the construction complexity, let $H_1(d)$ be the number of 2×2 switches needed for a unicast flexible delay line with maximum delay d and $H(d)$ be the number of 2×2 switches needed for a multicast flexible delay line with maximum delay d using the construction in Figure 3. Then we have from Figure 3 that

$$H(2d-1) = 2H_1(d-1) + H(d-1) + 2. \qquad (11)$$

If we use the modified Cantor network [3] to construct all the unicast flexible delay lines, then we have $H_1(d) = O((\log d)^2)$. It then follows from (11) that $H(d) = O((\log d)^3)$.

IV. Constructions of optical multicast switches with 100% throughput

In this section, we show that multicast flexible delay lines can be used for the constructions of optical multicast switches with 100% throughput (as long as the delay d can be made arbitrarily large). It is known in [28] that input-buffered switches cannot achieve 100% throughput with multicasting traffic. To construct an optical multicast switch with 100% throughput, our idea is to use the load-balanced Birkhoff-von Neumann switch (see e.g., [4], [5], [24], [9], [22]). The generic load balanced Birkhoff-von Neumann switch in [4] consists of two crossbar switches and buffers between them (see Figure 5). The first stage performs load balancing and the second stage performs switching. The connection patterns of the two crossbar switches are set up according to a predetermined periodic schedule (typically generated by a one-cycle permutation matrix in [4], [5]). The buffers between the two switches are virtual output queues (VOQ). When a packet arrives at an input port of the first stage, it is switched instantly to the buffer connected to that input port. To achieve multicasting, fan-out splitting is performed at that buffer. Then each copy of the arriving packet is stored at the corresponding VOQ. Packets stored in the VOQs of that buffer are then sent to their outputs when the buffer is connected to their outputs. It is shown in [5] that 100% throughput can be achieved for such a multicast switch as long as the buffer size in the middle can be made arbitrarily large.

The key observation is that each buffer in Figure 5 can be replaced by a multicast flexible delay line (with some maximum delay d). This is because each buffer is made of VOQs that has at most one arrival and one departure in a time slot. Moreover, as the connection patterns of the second switch is periodic, the delay of each copy of an arriving packet can be easily computed and thus known when a packet arrives (see e.g., [31] for a detailed formula). Thus, the conditions (1), (2) and (3) in Definition 1 are satisfied as long as the maximum delay d is large enough.

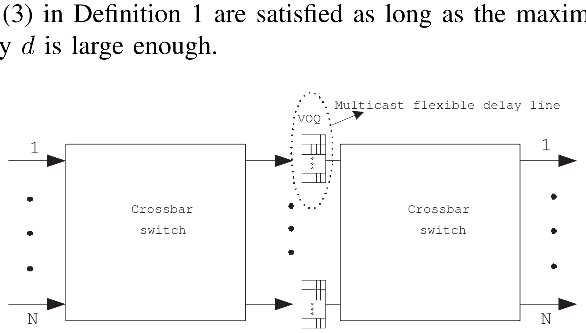

Fig. 5. The load-balanced Birkhoff-von Neumann switch

V. Conclusions

In this paper, we extended the constructions of unicast flexible delay lines in [3] to multicast flexible delay lines. Two constructions were proposed in this paper: (i) a construction by parallel unicast flexible delay lines in Section II and (ii) a recursive construction in Section III. Though the construction complexity by the first approach is much higher than that by the second approach, the control of the switches in the first approach is much easier than that in the second approach. As an application, we then showed that mulicast flexible delay lines can be used for the constructions of optical multicast switches with 100% throughput. The idea was based on the load-balanced Birkhoff-von Neumann switch.

To conclude the paper, we would like to point out some research problems that require further study.

(i) Complexity issues: here we showed that a multicast flexible delay line with maximum delay d can be constructed by using $O((\log d)^3)$ 2×2 switches. Is this the best one can do? See e.g., [25] for a recent work along this line. Also, finding the control of the 2×2 switches in such a construction may be a hard problem.

(ii) Practical limitations: there are crosstalks and power losses in optical switches and fiber delay lines. These put hard constraints on the number of 2×2 switches that can cascaded in each construction. Understanding these practical limitations are crucial for practical implementations of our constructions.

(iii) Performance tradeoffs: in the construction of an optical multicast switch, the maximum delay d needs to be arbitrarily large to achieve 100% throughput. In practice, one only needs to have a low packet loss probability. What would be the most appropriate value of d for a given packet loss probability? Are there other constructions that can achieve the same packet loss probability with a lower construction complexity?

References

[1] D. G. Cantor, “On nonblocking switching networks,” *Networks*, Vol. 1, pp. 367-377, 1971.

[2] C.-S. Chang, Y.-T. Chen, and D.-S. Lee, “Construction of optical FIFO queues,” *IEEE Transactions on Information Theory*, Vol. 52, No. 6, pp.2838-2843, 2006.

[3] C.-S. Chang, Y.-T. Chen, and D.-S. Lee, ”Multistage constructions of linear compressors, non-overtaking delay lines, and flexible delay lines,” *Proceedings of IEEE INFOCOM 2006*.

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

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

[6] 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, 2004.

[7] 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, No. 4, pp. 108-117, 2006.

[8] H. J. Chao, C. H. Lam and E. Oki. *Broadband Packet Switching Technologies: A Practical Guide to ATM Switches and IP Routers*. John Wiley & Sons, Inc., 2001.

[9] H. J. Chao, J. Song, N. S. Artan, G. Hu, and S. Jiang, “Byte-focal: a practical load-balanced switch,” *IEEE High Performance Switching and Routing*, 2005.

[10] Y.-T. Chen, C.-S. Chang, J. Cheng, D.-S. Lee and C.-C. Huang “Feed-forward SDL constructions of output-buffered multiplexers and switches with variable length bursts,” accepted by *IEEE INFOCOM 2007*.

[11] 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.

[12] H.-C. Chiu, C.-S. Chang, J. Cheng, and D.-S. Lee, “Using a single switch with O(M) inputs/outputs for the construction of an optical priority queue with $O(M^3)$ buffer,” accepted by *IEEE INFOCOM minisymposium* 2007.

[13] 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, 1996.

[14] 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, 2006.

[15] C. Clos, “A study of nonblocking switching networks,” *BSTJ*, Vol. 32, pp. 406-424, 1953.

[16] R. L. Cruz and J. T. Tsai, “COD: alternative architectures for high speed packet switching,” *IEEE/ACM Transactions on Networking*, Vol. 4, pp. 11-20, February 1996.

[17] P.-K. Huang, C.-S. Chang, J. Cheng and D.-S. Lee, “Recursive constructions of parallel FIFO and LIFO queues with switched delay lines,” accepted by *IEEE Transactions on Information Theory*.

[18] J. Hui, *Switching and Traffic Theory for Integrated Broadband Networks.* Boston: Kluwer Academic Publishers, 1990.

[19] 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, 1997.

[20] 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, 1998.

[21] F. K. Hwang. *The Mathematical Theory of Nonblocking Switching Networks*, Singapore: World Scientific Publishing Co., 1998.

[22] J.-J. Jaramillo, F. Milan, R. Srikant, ”Padded frames: a novel algorithm for stable scheduling in load-balanced switches.” Technical report from University of Illinois at Urbana-Champaign.

[23] M. J. Karol, “Shared-memory optical packet (ATM) switch,” SPIE Vol. 2024 Multigigabit Fiber Communications Systems, pp. 212-222, 1993.

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

[25] H. Kogan and I. Keslassy, “Optimal-complexity optical router,” accepted by IEEE INFOCOM 2007.

[26] S.-Y. R. Li. *Algebraic Switching Theory and Broadband Applications*. Academic Press, 2001.

[27] S.-Y. R. Li and J. Tan, “Fiber memory,” available from http://www.ie.cuhk.edu.hk/~bobli/fiber_memory.pdf.

[28] M. A. Marsan, A. Bianco, P. Giaccone, E. Leonardi, F. Neri, ”Multicast traffic in input-queued switches: optimal scheduling and maximum throughput,” *IEEE/ACM Trans. on Networking*, vol. 11, pp. 465-477, 2003.

[29] A. D. Sarwate and V. Anantharam, “Exact emulation of a priority queue with a switch and delay lines,” *Queueing Systems: Theory and Applications*, Vol. 53, pp. 115–125, July 2006.

[30] E. A. Varvarigos, “The ’Packing’ and ’Scheduling’ switch architectures for almost-all optical lossless networks,” *IEEE Journal of Lightwave Technologies*, vol. 16 (no. 10), pp. 1757-67, Oct. 1998.

[31] C.-L. Yu, C.-S. Chang, and D.-S. Lee, ”CR switch: a load-balanced switch with contention and reservation,” accepted by IEEE INFOCOM 2007.