Multistage Constructions of Linear Compressors, Non-overtaking Delay Lines, and Flexible Delay Lines

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

***Abstract*— Queueing theory is generally known as the theory to study the performance of queues. In this paper, we are interested in another aspect of queueing theory, the theory to construct queues via switched delay lines. We consider three types of discrete-time queues: linear compressors, non-overtaking delay lines and flexible delay lines. These three types of queues correspond to certain conditional nonblocking switches and (strict sense) nonblocking switches in switching theory. Analogous to their counterparts in switching theory, there exist multistage constructions for these three types of queues. Specifically, we develop a two-stage construction of a linear compressor and a three-stage construction of a non-overtaking delay line. Similarly, there is a three-stage construction of a flexible delay line. Moreover, a flexible delay line can also be constructed by a layered Cantor network.**

I. Introduction

Queueing theory and switching theory are closely related. Queueing theory is generally known as the theory to study the performance of queues. On the other hand, switching theory is regarded as the theory to construct switches. Up to present, both queueing theory and switching theory have been studied extensively in the literature. In fact, queueing theory has been constantly applied to switching theory for analyzing the performance of switches.

In this paper, we are interested in another aspect of queueing theory, the theory to construct queues. We will show how switching theory can be applied to this aspect of queueing theory. Intuitively, a sample path of a discrete-time queue with a single input link and a single output link can be viewed as a mapping from the arrivals (at the input link) to the departures (at the output link). This is similar to the mapping that connects the inputs of a switch to the outputs of a switch. The difference is that the arrivals at a queue is progressive in time and can be infinite. On the other hand, the inputs of a switch is usually finite and they can be wrapped around (or permuted in any manner).

Despite the difference, switching theory is quite useful in constructing queues considered in this paper. There are three types of discrete-time queues in this paper: linear compressors, non-overtaking delay lines and flexible delay lines. In the queueing context, a linear compressor is a First In First Out (FIFO) queue with vacations. A non-overtaking delay line is a FIFO queue with known departure times upon arrivals. A flexible delay line is a queue with an infinite number of servers. In the switching context, a linear compressor corresponds to a conditional nonblocking switch that satisfies a certain monotone and consecutive condition. In fact, such a conditional nonblocking switch is also called a linear compressor in switching theory (see e.g., the book by Li [23]). A non-overtaking delay line also corresponds to a conditional nonblocking switch that satisfies a monotone condition. Such a switch is called a UU (Unimodal-Unimodal) nonblocking switch in [23]. A flexible delay line corresponds to a (strict sense) nonblocking switch.

Our approaches for constructing these queues are inspired by the early works in [24], [28], [21], [27] that map rearrangeable nonblocking switches to time slot interchanges. As the two-stage construction for a linear compressor in switching theory, we show that there is also a two-stage construction of a linear compressor in our setting. Moreover, via recursive expansion of the two-stage construction, one can then construct a linear compressor via a banyan type of network. Analogous to the three-stage construction of a UU nonblocking switch, we show that there is a three-stage construction of a non-overtaking delay line. This is a generalization of the two-stage construction of a non-overtaking line in our previous paper [8]. Similarly, there is a three-stage construction of a flexible delay line. Such a result is analogous to the three-stage Clos network for a nonblocking switch. It is known that the Cantor network [5] can be used for constructing a nonblocking switch with less complexity than that by the Clos network. We show that this is also the case by modifying the Cantor network to the queueing setting.

Our interest in constructing queues originates from optical packet switching. For optical packet switching, one has to build optical queues that store optical packets. 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 the optical packets come out at the right place and at the right time.

There are already several papers in the literature that addressed the issues of optical queues via switched delay lines. It was first demonstrated by Karol [22] that SDL elements could be used as a buffer for a shared-memory optical packet switch. A huge project (see [6], [7]), called CORD (contention resolution by delay lines), was started by Chlamtac et al at Boston University. Cruz and Tsai [12] constructed a 2-to-1 FIFO multiplexer with SDL elements. Hunter, Chia and Andonovic [16] constructed a 2-to-1 FIFO multiplexer with less complexity. In [18], SLOB (Switch with Large Optical Buffers) was proposed for the extension of optical buffered switches with N input/output ports $(N \geq 2)$. Varvarigos [31] proposed a "packing" and "scheduling" optical switch. Recently, Sarwate and Anantharam [29] addressed the complexity of a priority queue. For additional references of optical packet switches, we refer to the review papers [17], [15], [33]. For the introduction of switching theory, we refer to the books by Benes [3], Hui [19], Schwartz [30], Hwang [20], Li [23], and references therein.

The paper is organized as follows. In Section II-A, we introduce basic definitions and prior works for linear compressors, non-overtaking delay lines, and flexible delay lines. We develop the two-stage construction for a linear compressor in Section III. We then move on to the three-stage construction of a non-overtaking delay line in Section IV. For the construction of flexible delay lines, we show the three-stage construction in Section V and the construction by the Cantor network in Section VI. The paper is concluded in Section VII, where we summarize our results and address some problems for future research.

II. Basic definitions and prior works

A. SDL elements

In this paper, we consider fixed size packets over optical links. Assume that time in all our optical links is slotted and synchronized so that a packet can be transmitted within a time slot. Our constructions will be based on network elements that are built by optical crossbar switches and fiber delay lines. Such network elements are called Switched Delay Line (SDL) elements in the literature (see e.g., [22], [6], [12], [16], [10]). As addressed in our previous paper [8], a typical example of an SDL element is an optical memory cell in Figure 1. An optical memory cell is constructed by a 2×2 optical crossbar switch and a fiber delay line with one time slot (unit) of delay. To write a packet to the memory cell, set the 2×2 crossbar switch to the "cross" state so that the packet can be directed to the fiber delay line with one time slot of delay. Once the write operation is completed, the crossbar switch is then set to the "bar" state so that the packet directed into the fiber delay line keeps circulating through the fiber delay line. To read out the information from the memory cell, set the crossbar switch to the "cross" state so that the packet in the fiber delay line can be directed to the output link.

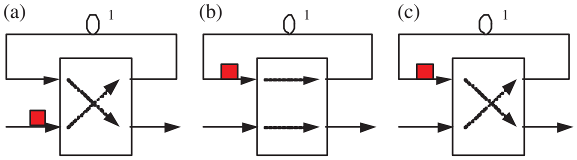


Fig. 1. An optical memory cell: (a) writing information (b) circulating information (c) reading information

One of the most important properties of SDL elements is the following time interleaving property for scaled SDL elements in [10].

Definition 1 (Scaled SDL element [10]) *A scaled SDL element is said to be with scaling factor m if the delay in every delay line is m times of that in the original (unscaled) SDL element.*

Proposition 2 (Time interleaving property [10]) *A scaled SDL element with scaling factor m can be operated as time interleaving of m SDL elements.*

The intuition of the time interleaving property, as explained in [8], can be easily understood by considering the scaled optical memory cell with scaling factor 2 in Figure 2. Note that the length of the delay line in Figure 2 is twice of that in the original optical memory cell in Figure 1. As such, every packet directed into the delay line takes two units of time to circulate back to the 2×2 optical crossbar switch. To operate the scaled optical memory cell with scaling factor 2, one first partitions time into even and odd numbered time slots. For the even numbered time slots, one can set the connection patterns of the 2×2 optical crossbar switch in the scaled SDL element according to the read/write operation for one memory cell. Similarly, for the odd numbered time slots, one can set the connection patterns of the 2×2 optical crossbar switch in the scaled SDL element according to the read/write operation for another memory cell.

B. Flexible and non-overtaking delay lines

In this section, we introduce the concepts of flexible delay lines and non-overtaking delay lines in [8].

Definition 3 (Flexible delay line [8]) *A flexible delay line is a network element with one input link and one output link. Let $\tau^a(n)$ be the arrival time of the n^{th} packet at the input link and $\tau^d(n)$ be the departure time of the n^{th} packet at the output link. Suppose that the departure time of a packet is known upon its arrival. A* flexible *delay line with the range of delay $[d_1, d_2]$ realizes the set of mappings (or sample paths) that satisfy*

$$\tau^a(n) + d_1 \leq \tau^d(n) \leq \tau^a(n) + d_2, \quad \textit{for all } n, \qquad (1)$$

and

$$\tau^d(m) \neq \tau^d(n), \quad \textit{for all } m \neq n. \qquad (2)$$

In particular, if $d_1 = 0$, then it is called a flexible delay line with maximum delay d_2.

A non-overtaking delay line (defined below) is a special case of a flexible delay line that do not allow a packet to overtake its previous packet.

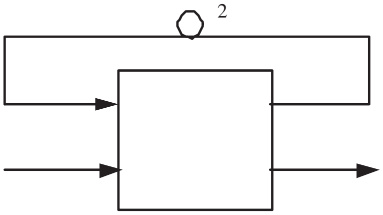


Fig. 2. An optical memory cell with scaling factor 2

Definition 4 (Non-overtaking delay line [8]) *A non-overtaking delay line is a network element with one input link and one output link. Suppose that the departure time of a packet is known upon its arrival. A* non-overtaking *delay line with the range of delay $[d_1, d_2]$ realizes the set of mappings that satisfy (1) and*

$$\tau^d(n) < \tau^d(n+1) \quad \textit{for all } n. \qquad (3)$$

In particular, if $d_1 = 0$, then it is called a non-overtaking delay line with maximum delay d_2.

A linear compressor (defined below) is a special case of a non-overtaking delay line that satisfies a certain monotone and consecutive condition.

Definition 5 (Linear compressor [8]) *Suppose that the departure time of a packet is known upon its arrival. A network element with a single input link and a single output link is called a linear compressor with the range of delay $[d_1, d_2]$ if it realizes the set of mappings that satisfy (1) and the following monotone and consecutive condition: $\tau^d(n) = \tau^d(n-1) + 1$ whenever $\tau^a(n) \leq \tau^d(n-1)$. In particular, if $d_1 = 0$, then it is called a linear compressor with maximum delay d_2.*

As pointed in [8], the name, linear compressor, originates from its counterpart for space switches (see e.g., [19], [23]). The condition $\tau^a(n) \leq \tau^d(n-1)$ means that the n^{th} packet arrives before the $n-1^{th}$ packet departs. If one defines a busy period of a linear compressor as the period of time that there are packets in the linear compressor, then the monotone and consecutive condition implies that the departures in a busy period are monotone and consecutive. Note that the packet that initiates a busy period can have an arbitrary delay (as long as its delay is not greater than the maximum delay).

Definition 6 (Mirror image [8]) *The mirror image of an SDL element is an SDL element that reverses the direction of every link in the original SDL element. By so doing, the inputs (resp. outputs) of the original SDL element become the outputs (resp. inputs) of its mirror image.*

One key property of the mirror image is the following proposition in [8].

Proposition 7 *If a sample path can be realized by an SDL element, then its time reversed sample path can also be realized by the mirror image of the SDL element.*

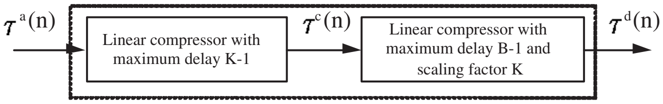
The mirror image of a linear compressor is known as a linear decompressor defined below.

Definition 8 (Linear decompressor [8]) *Suppose that the departure time of a packet is known upon its arrival. A network element with a single input link and a single output link is called a linear decompressor with the range of delay $[d_1, d_2]$ if it realizes the set of mappings that satisfy (1), (3) and the following* inverse *monotone and consecutive condition: $\tau^a(n) = \tau^a(n-1) + 1$ whenever $\tau^a(n) \leq \tau^d(n-1)$. In particular, if $d_1 = 0$, then it is called a linear decompressor with maximum delay d_2.*

It is shown in [8] that a self-routing linear compressor with maximum delay $2^k - 1$ can be constructed by using the 2-to-1 multiplexer with buffer $2^k - 1$ in [10]. As a linear decompressor is the mirror image of a linear compressor, a self-routing linear decompressor can be obtained by mirroring the 2-to-1 multiplexer in [10]. Furthermore, it is shown in [8] that a non-overtaking delay line with maximum delay d can be constructed by a concatenation of a linear compressor with maximum delay d and a linear decompressor with maximum delay d. As such, a self-routing non-overtaking delay line with maximum delay $2^k - 1$ can be constructed by the 2-to-1 multiplexer with buffer $2^k - 1$ in [10] and its mirror image. In the following two sections, we will generalize the results in [8] by developing multistage constructions of linear compressors and non-overtaking delay lines. We will show how they can be constructed by optical memory cells.

III. A TWO-STAGE CONSTRUCTION OF A LINEAR COMPRESSOR

In Figure 3, we consider a two-stage construction of a linear compressor with maximum delay $BK - 1$. The first stage is a linear compressor with maximum delay $K - 1$. The second stage is a scaled linear compressor with maximum delay $B - 1$ and scaling factor K. From the time interleaving property in Proposition 2, we note that the scaled linear compressor at the second stage can be operated as K time interleaved linear compressors. As shown in Figure 4, these K time interleaved linear compressors are connected to the output link of the linear compressor at the first stage and the output link of the network element *periodically* with period K. Moreover, as the delay in each delay line of the scaled linear compressor at the second stage is K times of that in the original linear compressors, these K time interleaved linear compressors, when connected, only allow delay that is an integer multiple of K, i.e., $d = 0, K, 2K, \ldots, (B-1)K$.



$\tau^a(n)$
Linear compressor with maximum delay K-1
$\tau^c(n)$
Linear compressor with maximum delay B-1 and scaling factor K
$\tau^d(n)$

Fig. 3. A two-stage construction of a linear compressor

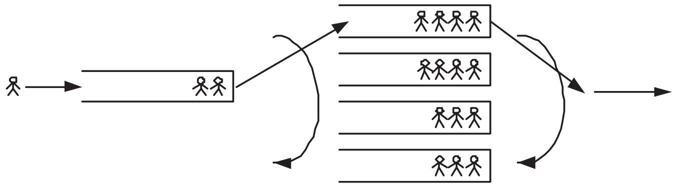


Fig. 4. An illustration of periodic connections in the two-stage construction

Now we specify the operation rule for the two-stage construction. As shown in Figure 3, let $\tau^a(n)$ be the arrival time of the n^{th} packet, $\tau^c(n)$ be its departure time from the linear compressor at the first stage, and $\tau^d(n)$ be its departure time.

(R1) Initially, we set

$$\tau^c(1) = \tau^a(1) + \Big((\tau^d(1) - \tau^a(1)) \bmod K\Big).$$

If $\tau^c(n-1) < \tau^a(n)$, then we set

$$\tau^c(n) = \tau^a(n) + \Big((\tau^d(n) - \tau^a(n)) \bmod K\Big).$$

Otherwise, we set $\tau^c(n) = \tau^c(n-1) + 1$.

Theorem 9 *If the network element in Figure 3 is started from an empty system, then under (R1) the two-stage construction is a linear compressor with maximum delay $BK - 1$.*

Proof. According to Definition 5 for a linear compressor, we need to show that the network element in Figure 3 can realize all the mappings (or sample paths) that satisfy

$$\tau^a(n) \le \tau^d(n) \le \tau^a(n) + BK - 1, \tag{4}$$

$$\tau^d(n) = \tau^d(n-1) + 1, \quad \text{whenever } \tau^a(n) \le \tau^d(n-1). \tag{5}$$

In other words, if (4) and (5) hold for all n, then under the assignment rule in (R1)

$$\tau^a(n) \le \tau^c(n) \le \tau^a(n) + K - 1, \tag{6}$$

$$\tau^c(n) = \tau^c(n-1) + 1, \quad \text{whenever } \tau^a(n) \le \tau^c(n-1), \tag{7}$$

$$\tau^c(n) \le \tau^d(n) \le \tau^c(n) + (B-1)K, \tag{8}$$

$$(\tau^d(n) - \tau^c(n)) \bmod K = 0, \tag{9}$$

$$\tau^d(n) = \tau^d(n^*) + K, \quad \text{whenever } \tau^a(n) \le \tau^d(n^*), \tag{10}$$

where n^* is the last packet that departs before the n^{th} packet from the same time interleaved linear compressor at the second stage.

We will prove this by induction. For $n = 1$, we have

$$\tau^c(1) = \tau^a(1) + \Big((\tau^d(1) - \tau^a(1)) \bmod K\Big).$$

Thus,

$$\tau^a(1) \le \tau^c(1) \le \tau^a(1) + K - 1$$

and

$$\begin{aligned} \tau^d(1) - \tau^c(1) &= K\lfloor(\tau^d(1) - \tau^a(1))/K\rfloor \\ &\le (B-1)K. \end{aligned}$$

It is easy to see that (6), (8) and (9) are satisfied. As the network element is started from an empty system, there is no need to check (7) and (10).

Now suppose that the induction hypotheses in (6)-(10) hold up to $n-1$. For the n^{th} packet, we need to consider the following cases.

Case 1 $\tau^a(n) > \tau^d(n-1)$:

In this case, the n^{th} packet sees an empty system. The proof is the same as the case for $n = 1$.

Case 2 $\tau^a(n) \le \tau^d(n-1)$ and $\tau^a(n) > \tau^c(n-1)$:

Since $\tau^a(n) > \tau^c(n-1)$, we also have from (R1) that

$$\tau^c(n) = \tau^a(n) + ((\tau^d(n) - \tau^a(n)) \bmod K). \tag{11}$$

As argued for the case $n = 1$, it is easy to see that (6), (8) and (9) are satisfied. Since $\tau^a(n) > \tau^c(n-1)$, there is no need to verify (7).

To prove (10), let $n_0 = \sup\{m < n : \tau^a(m) > \tau^d(m-1)\}$ be the index of the packet that initiates the busy period containing the n^{th} packet. From (5), it follows that for all $n_0 < m \le n$

$$\tau^d(m) = \tau^d(m-1) + 1, \tag{12}$$

As illustrated in Figure 4, these K time interleaved linear compressors at the second stage are connected to the output link of the linear compressor at the first stage periodically with period K. If $n - K \ge n_0$, then we have from (12) that $n^* = n - K$ is the last packet that departs before the n^{th} packet from the same time interleaved linear compressor at the second stage. As such, it follows from (12) that

$$\tau^d(n^*) + K = \tau^d(n-K) + K = \tau^d(n). \tag{13}$$

On the other hand, if $n - K < n_0$, then the n^{th} packet arrives at an empty linear compressor at the second stage and there is no need to check (10).

Case 3 $\tau^a(n) \le \tau^d(n-1)$ and $\tau^a(n) \le \tau^c(n-1)$:

Since $\tau^a(n) \le \tau^c(n-1)$, we also have from (R1) that

$$\tau^c(n) = \tau^c(n-1) + 1. \tag{14}$$

Thus, (7) is satisfied. Moreover,

$$\tau^a(n) \le \tau^c(n-1) = \tau^c(n) - 1 \le \tau^c(n). \tag{15}$$

Using the induction hypothesis for $n-1$ in (6) and $\tau^a(n-1) < \tau^a(n)$ yields

$$\begin{aligned}\tau^c(n) = \tau^c(n-1)+1 &\leq \tau^a(n-1)+K-1+1 \\ &\leq \tau^a(n)+K-1. \end{aligned} \tag{16}$$

Thus, (6) is satisfied.

Since $\tau^a(n) \leq \tau^d(n-1)$, we have from (5) that

$$\tau^d(n) = \tau^d(n-1)+1, \tag{17}$$

From (17) and (14), it follows that

$$\tau^d(n) - \tau^c(n) = \tau^d(n-1) - \tau^c(n-1).$$

Thus, (8) and (9) follow from the induction hypotheses for $n-1$.

The argument for (10) is the same as that in Case 2. ■

Now we show how one constructs self-routing linear compressors by using optical memory cells. First, we show that an optical memory cell can be used as a linear compressor with maximum delay 1. When there is a packet stored in the optical memory cell, the 2×2 switch is always set to the "cross" state. When a packet arrives at the empty optical memory cell, the 2×2 switch is set to the "bar" state if its delay is 0 and the "cross" state if its delay is 1. As the maximum delay is 1, every packet passes through the fiber delay line with one unit of delay at most once.

With $B = K = 2$, we can apply Theorem 9 to construct a linear compressor with maximum delay 3 via a concatenation of an optical memory cell (at the first stage) and a scaled optical memory cell with the scaling factor 2 (at the second stage). By recursively expanding the two-stage construction in Theorem 9 with $K = 2$, one can construct a linear compressor with maximum delay $2^k - 1$ by using a series of k scaled optical memory cells with scaling factors $2^0, 2^1, 2^2, \ldots, 2^{k-1}$ (see Figure 5(a)). As a packet passes through every fiber delay line in Figure 5(a) at most once, one can simply use the binary representation of packet delay for self-routing. Since a linear decompressor is the mirror image of a linear compressor, one can also construct a self-routing linear decompressor with maximum delay $2^k - 1$ by using a series of k scaled optical memory cells with scaling factors $2^{k-1}, 2^{k-2}, \ldots, 2^1, 2^0$ (see Figure 5(b)).

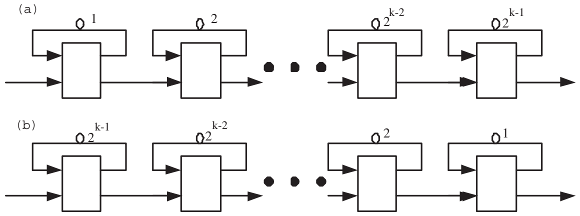


Fig. 5. (a) A self-routing linear compressor with maximum delay $2^k - 1$, (b) a self-routing linear decompressor with maximum delay $2^k - 1$.

IV. A THREE-STAGE CONSTRUCTION OF A NON-OVERTAKING DELAY LINE

In Figure 6, we consider a three-stage construction of a non-overtaking delay line with maximum delay $BK - 1$. The first stage is a linear compressor with maximum delay $K - 1$, the second stage is a scaled non-overtaking delay line with maximum delay $B - 1$ and scaling factor K, and the third stage is a linear decompressor with maximum delay $K - 1$.

As shown in Figure 6, let $\tau^a(n)$ be the arrival time of the n^{th} packet, $\tau^c(n)$ be the departure time of the n^{th} packet from the linear compressor, $\tau^b(n)$ be the arrival time of the n^{th} packet at the linear decompressor, and $\tau^d(n)$ be the departure time of the n^{th} packet. In order to show that the three-stage construction in Figure 6 can be operated as a non-overtaking delay line, we need to specify $\tau^c(n)$ and $\tau^b(n)$ for every n.

(R2) Initially, we set

$$\tau^c(1) = \tau^a(1) + \Big((\tau^d(1) - \tau^a(1)) \bmod K\Big), \tag{18}$$

and

$$\tau^b(1) = \tau^c(1) + \lfloor \frac{\tau^d(1) - \tau^c(1)}{K} \rfloor \times K. \tag{19}$$

Note from (18) that

$$\tau^d(1) - \tau^c(1) = \lfloor \frac{\tau^d(1) - \tau^a(1)}{K} \rfloor \times K. \tag{20}$$

Thus, it follows from (19) and (18) that

$$\tau^b(1) = \tau^d(1). \tag{21}$$

If $\tau^c(n-1) < \tau^a(n)$, then we set

$$\tau^c(n) = \tau^a(n) + \Big((\tau^d(n) - \tau^a(n)) \bmod K\Big), \tag{22}$$

and

$$\tau^b(n) = \tau^c(n) + \lfloor \frac{\tau^d(n) - \tau^c(n)}{K} \rfloor \times K. \tag{23}$$

By so doing, we also have

$$\tau^b(n) = \tau^d(n). \tag{24}$$

Otherwise, we set

$$\tau^c(n) = \tau^c(n-1) + 1, \tag{25}$$

and

$$\tau^b(n) = \tau^c(n) + \lfloor \frac{\tau^d(n) - \tau^c(n)}{K} \rfloor \times K. \tag{26}$$

Note from (19),(23) and (26) that we always have

$$\tau^b(n) = \tau^c(n) + \lfloor \frac{\tau^d(n) - \tau^c(n)}{K} \rfloor \times K \tag{27}$$

for all n. As such, $\tau^b(n) - \tau^c(n)$ is an integer multiple of K and it can be carried by the scaled non-overtaking delay line with scaling factor K in the second stage.

Theorem 10 *If the network element in Figure 6 is started from an empty system, then under (R2) the three-stage construction is a non-overtaking delay line with maximum delay* $BK - 1$.

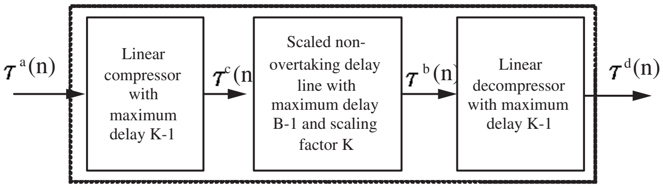

τ^a(n)
Linear compressor with maximum delay K-1
τ^c(n
Scaled non-overtaking delay line with maximum delay B-1 and scaling factor K
τ^b(n)
Linear decompressor with maximum delay K-1
τ^d(n)

Fig. 6. A three-stage construction of a non-overtaking delay line

Proof. We need to show that all the mappings that satisfy (1) and (3) can be realized by a concatenation of a linear compressor with maximum delay $K-1$ (at the first stage), a scaled non-overtaking delay line with maximum delay $B-1$ and scaling factor K (at the second stage), and a linear decompressor with maximum delay $K-1$ (at the third stage). According to Definition 4 for a non-overtaking delay line, Definition 5 for a linear compressor and Definition 8 for a linear decompressor, we need to show that if for all n

$$\tau^a(n) \leq \tau^d(n) \leq \tau^a(n) + BK - 1, \tag{28}$$

$$\tau^d(n) > \tau^d(n-1), \tag{29}$$

then under the assignment rule for $\tau^c(n)$ and $\tau^b(n)$ in (R2) that

$$\tau^a(n) \leq \tau^c(n) \leq \tau^a(n) + K - 1, \tag{30}$$

$$\tau^c(n) = \tau^c(n-1) + 1, \quad \text{whenever } \tau^a(n) \leq \tau^c(n-1), \tag{31}$$

$$\tau^c(n) \leq \tau^b(n) \leq \tau^c(n) + (B-1)K \tag{32}$$

$$(\tau^b(n) - \tau^c(n)) \bmod K = 0, \tag{33}$$

$$\tau^b(n) > \tau^b(n-1), \tag{34}$$

$$\tau^b(n) \leq \tau^d(n) \leq \tau^b(n) + K - 1, \tag{35}$$

$$\tau^b(n) = \tau^b(n-1) + 1, \quad \text{whenever } \tau^b(n) \leq \tau^d(n-1). \tag{36}$$

We first show that $\tau^a(n) \leq \tau^c(n)$ for all n. If $n = 1$ or $\tau^c(n-1) < \tau^a(n)$, then we have $\tau^a(n) \leq \tau^c(n)$ from (18) and (22). On the other hand, if $\tau^c(n-1) \geq \tau^a(n)$, then we have from (25) that

$$\tau^a(n) < \tau^c(n-1) + 1 = \tau^c(n).$$

Next, we show by induction that $\tau^c(n) \leq \tau^d(n)$ for all n. Since $\tau^a(1) \leq \tau^d(1)$, this holds trivially for $n = 1$ from (18). Now suppose that $\tau^c(n-1) \leq \tau^d(n-1)$. If $\tau^c(n-1) < \tau^a(n)$, then we also have from (22) and $\tau^a(n) \leq \tau^d(n)$ that $\tau^c(n) \leq \tau^d(n)$. On the other hand, if $\tau^c(n-1) \geq \tau^a(n)$, then we have from (25) that $\tau^c(n) = \tau^c(n-1) + 1$. It then follows from the induction hypothesis and $\tau^d(n) > \tau^d(n-1)$ that

$$\tau^c(n) = \tau^c(n-1) + 1 \leq \tau^d(n-1) + 1 \leq \tau^d(n).$$

Now we use $\tau^c(n) \leq \tau^d(n)$ for all n to show that (32),(33) and (35) hold for all n. Note from (27) that

$$\tau^b(n) - \tau^c(n) = K\lfloor \frac{\tau^d(n) - \tau^c(n)}{K} \rfloor.$$

Thus, (33) is satisfied for all n. Moreover, we have from $\tau^a(n) \leq \tau^c(n)$ for all n and (28) that

$$\begin{aligned}
&\tau^b(n) - \tau^c(n) = K\lfloor \frac{\tau^d(n) - \tau^c(n)}{K} \rfloor \\
&\leq K\lfloor \frac{\tau^d(n) - \tau^a(n)}{K} \rfloor \leq K\lfloor \frac{(BK-1)}{K} \rfloor \\
&= (B-1)K.
\end{aligned} \tag{37}$$

Thus, (32) is also satisfied for all n. Furthermore,

$$\begin{aligned}
&\tau^d(n) - \tau^b(n) = \tau^d(n) - \tau^c(n) - K\lfloor \frac{\tau^d(n) - \tau^c(n)}{K} \rfloor \\
&= ((\tau^d(n) - \tau^c(n)) \bmod K.
\end{aligned} \tag{38}$$

This shows that (35) is also satisfied for all n.

We will prove the other four conditions, i.e., (30), (31), (34) and (36), by induction on n. For $n = 1$, it follows from (18) and (28) that (30) holds trivially. As this is the first packet, there is no need to verify (31), (34) and (36).

Now suppose that the induction hypotheses in (30), (31), (34) and (36) hold up to $n-1$. For the n^{th} packet, we need to consider the following two cases.

Case 1 $\tau^a(n) > \tau^c(n-1)$:

In this case, the assignments for $\tau^c(n)$ and $\tau^b(n)$ are the same as those for $n = 1$. As such, (30) holds accordingly. Since $\tau^a(n) > \tau^c(n-1)$, there is no need to verify (31). Also, in view of $\tau^b(n) = \tau^d(n)$ in (24) and $\tau^d(n) > \tau^d(n-1)$ in (29), we have $\tau^b(n) > \tau^d(n-1)$. Thus, there is no need to verify (36). In conjunction with $\tau^d(n-1) \geq \tau^b(n-1)$ in (35), we derive $\tau^b(n) > \tau^b(n-1)$ for (34).

Case 2 $\tau^a(n) \leq \tau^c(n-1)$:

In this case, we have from (25) that $\tau^c(n) = \tau^c(n-1) + 1$. Thus, (31) is satisfied. Moreover, using the induction hypothesis for $n-1$ in (30) and the fact that $\tau^a(n) \geq \tau^a(n-1) + 1$ yields

$$\begin{aligned}
&\tau^c(n) = \tau^c(n-1) + 1 \leq \tau^a(n-1) + K - 1 + 1 \\
&\leq \tau^a(n) + K - 1.
\end{aligned}$$

This shows that $\tau^c(n) \leq \tau^a(n) + K - 1$ in (30).

To see (34), note from (25) and (29) that

$$\tau^d(n) - \tau^d(n-1) \geq 1 = \tau^c(n) - \tau^c(n-1).$$

It then follows from (27) and (25) that

$$\begin{aligned}
&\tau^b(n-1) \\
&= \tau^c(n-1) + \lfloor \frac{\tau^d(n-1) - \tau^c(n-1)}{K} \rfloor \times K \\
&\leq \tau^c(n-1) + \lfloor \frac{\tau^d(n) - \tau^c(n)}{K} \rfloor \times K \\
&< \tau^c(n) + \lfloor \frac{\tau^d(n) - \tau^c(n)}{K} \rfloor \times K \\
&= \tau^b(n).
\end{aligned}$$

It remains to verify (36). If $\tau^b(n) \leq \tau^d(n-1)$, then it follows from (35) (for $n-1$) that

$$\tau^b(n) \leq \tau^d(n-1) \leq \tau^b(n-1) + K - 1.$$

Since $(\tau^b(n) - \tau^c(n)) \bmod K = 0$ for all n in (33), it then follows from (25) that

$$\Big((\tau^b(n) - \tau^b(n-1)) \bmod K\Big) = \Big((\tau^c(n) - \tau^c(n-1)) \bmod K\Big) = 1.$$

Thus, we have $\tau^b(n) = \tau^b(n-1) + 1$. ■

For the special case that $B = 1$, the second stage can be omitted. Thus, the three-stage construction of a non-overtaking delay line in Theorem 10 is a generalization of the two-stage construction in [8]. As we have shown how one constructs self-routing linear compressors and decompressors by optical memory cells in Section III, it is quite straightforward to apply Theorem 10 to construct a self-routing non-overtaking delay line by a concatenation of scaled optical memory cells. As pointed out in [8], one can further construct buffered multiplexers in [10], [9] by non-overtaking delay lines.

V. A THREE-STAGE CONSTRUCTION OF A FLEXIBLE DELAY LINE

In Figure 7, we show a construction of a flexible delay line with the range of delay $[K-1, BK-1]$. It is a combination of three parallel three-stage constructions. In each three-stage construction, there is a flexible delay line with maximum delay $K-1$ (at the first stage), a scaled flexible delay line with maximum delay $B-1$ and scaling factor K (at the second stage), and a flexible delay line with maximum delay $K-1$ (at the third stage). These three parallel three-stage constructions are joined by a 1×3 switch at the beginning and a 3×1 switch at the end. The 1×3 switch acts as a 1-to-3 demultiplexer so that an arriving packet can choose its path from one of the three three-stage constructions. The departures from these three three-stage constructions are then multiplexed by the 3×1 switch at the end.

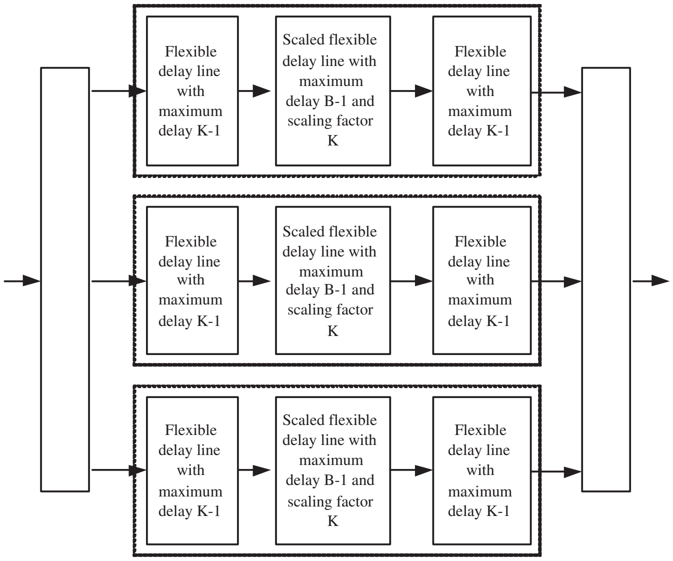

Flexible delay line with maximum delay K-1
Scaled flexible delay line with maximum delay B-1 and scaling factor K
Flexible delay line with maximum delay K-1
Flexible delay line with maximum delay K-1
Scaled flexible delay line with maximum delay B-1 and scaling factor K
Flexible delay line with maximum delay K-1
Flexible delay line with maximum delay K-1
Scaled flexible delay line with maximum delay B-1 and scaling factor K
Flexible delay line with maximum delay K-1

Fig. 7. A three-stage construction of a flexible delay line with the range of delay $[K-1, BK-1]$

Theorem 11 *The three-stage construction in Figure 7 is a flexible delay line with the range of delay* $[K-1, BK-1]$.

Proof. The proof for Theorem 11 is quite similar to that for the classical three-stage nonblocking Clos networks [11]. We will show for each arriving packet, there is a non-conflicting path from the input to the output. Consider the n^{th} packet. Let $\tau^a(n)$ be the arrival time of the n^{th} packet and $\tau^d(n)$ be the departure time of the n^{th} packet. Suppose that the delay of the n^{th} packet, denoted by d, is in the range $[K-1, BK-1]$, i.e., $K-1 \le d \le BK-1$. Then we claim that the number of feasible paths for the n^{th} packet to go through one of the three three-stage constructions is K. To see this, note that one can choose $0 \le d_1 \le K-1$ to be the delay at the flexible delay line with maximum delay $K-1$ at the first stage. Once d_1 is chosen, there is a unique way to determine the delay at the second stage and the delay at the third stage (as the delay at the second stage must be an integer multiple of K). Specifically, the delay at the third stage is $((d - d_1) \bmod K)$ and the delay at the second stage is $\lfloor (d - d_1)/K \rfloor \times K$.

As there are three parallel constructions, the total number of feasible paths for the n^{th} packet to go through the network element is $3K$. The only places that the n^{th} packet might collide with others are the three output links of the first stage and the three input links of the third stage. As the first stage (in all the three-stage constructions) is a flexible delay line with maximum delay $K-1$, those packets that might collide with the n^{th} packet at the output links of the first stage must arrive during $[\tau^a(n) - (K-1), \tau^a(n) - 1]$. These packets will use at most $K-1$ paths among the $3K$ feasible paths for the n^{th} packet. On the other hand, those packets that might collide with the n^{th} packet at the three input links of the third stage must depart during $[\tau^d(n) - (K-1), \tau^d(n) - 1]$ and $[\tau^d(n) + 1, \tau^d(n) + K - 1]$. Similarly, these packets use at most another $2(K-1)$ paths among the $3K$ feasible paths for the n^{th} packet. Thus, there is at least one non-conflicting path for the n^{th} packet. ■

Unlike the classical nonblocking Clos networks, the construction in Figure 7 cannot accommodate for the packets with delay smaller than $K-1$. This is because there are not enough feasible paths for those packets with short delay. For instance, for a packet with delay 0, there are only three feasible paths.

To construct a flexible delay line with maximum delay $BK-1$, one can simply add a flexible delay line with maximum delay $K-1$ parallel to the three three-stage constructions (see Figure 8). By so doing, those packets with delay smaller than $K-1$ can be routed through the added flexible delay line with maximum delay $K-1$.

We note that one may construct a flexible delay line with maximum delay $K-1$ by a concatenation of $K-1$ optical memory cells (all with one unit of delay). This is because an optical memory cell can be used for storing one packet.

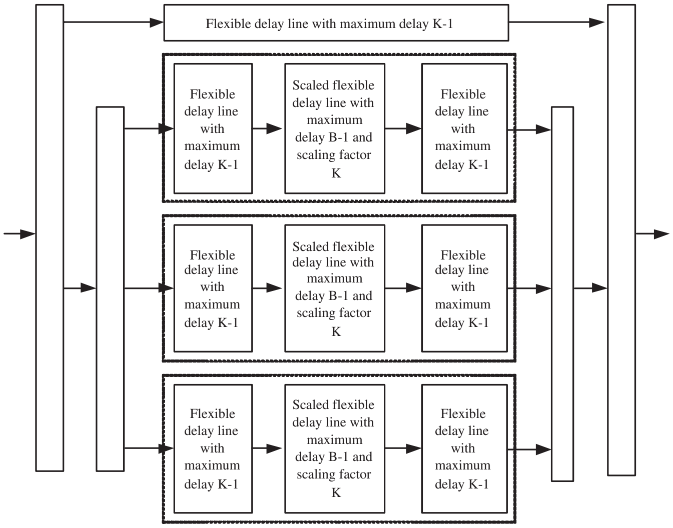

Flexible delay line with maximum delay K-1
Flexible delay line with maximum delay K-1
Scaled flexible delay line with maximum delay B-1 and scaling factor K
Flexible delay line with maximum delay K-1
Flexible delay line with maximum delay K-1
Scaled flexible delay line with maximum delay B-1 and scaling factor K
Flexible delay line with maximum delay K-1
Flexible delay line with maximum delay K-1
Scaled flexible delay line with maximum delay B-1 and scaling factor K
Flexible delay line with maximum delay K-1

Fig. 8. A flexible delay line with maximum delay $BK-1$

When a packet arrives, we may store that packet in an optical memory cell until its departure time. As the maximum delay is $K-1$, there are at most $K-1$ packets that need to be stored at the same time.

To compute the construction complexity, let $H(B)$ be the number of 2×2 switches needed for a flexible delay line with maximum delay B using the construction in Figure 8. Note that a 1×3 switch (and a 3×1 switch) can be implemented by two 2×2 switches. Clearly, we have the following recursive equation:

$$H(BK-1) = 7H(K-1) + 3H(B-1) + 6. \qquad (39)$$

Letting $B=K$ yields

$$H(B^2-1) = 10H(B-1) + 6. \qquad (40)$$

If we use $B-1$ optical memory cells to construct all the ten (scaled or unscaled) flexible delay lines with maximum delay $B-1$, we can construct a flexible delay line with maximum delay B^2-1 by using $10B-4$ 2×2 switches. By recursive expansion of (40), one can construct a flexible delay line with maximum delay B with $O((\log B)^\gamma)$ 2×2 switches, where $\gamma = \log_2 10 \approx 3.321928$.

VI. Constructions of Flexible Delay Lines by Cantor Networks

Instead of using only three parallel three-stage constructions in the previous section, one may consider a combination of multiple multistage constructions as in the Cantor Network [5] for a nonblocking switch. First, we consider a multistage network element constructed by a concatenation of $2k-1$ scaled optical memory cells. The scaling factor at the j^{th} stage is 2^{j-1} for $j=1,2,\ldots,k$ and the scaling factor at the j^{th} stage is 2^{2k-1-j} for $j=k+1,\ldots,2k-1$. Such a network element is called a Benes time slot interchange as it can be used for realizing a $2^k\times 2^k$ Benes time slot interchange (see e.g., [24], [27], [8]). Now we combine m $2^k\times 2^k$ Benes time slot interchanges by adding a $1\times m$ switch in the front and an $m\times 1$ switch at the end. Such a network element is called the (m,k)-Cantor network (as it is closely related to the nonblocking Cantor network). In Figure 9, we depict the $(6,4)$-Cantor network.

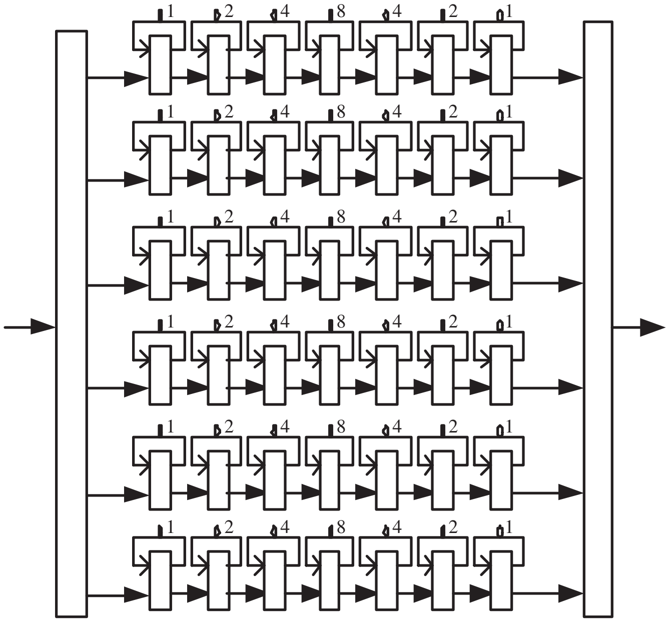


Fig. 9. The $(6,4)$-Cantor network

Theorem 12 *Suppose that every delay line in the (m,k)-Cantor network can be traversed by a packet at most once. If $m \geq \frac{3}{2}k$, then the (m,k)-Cantor network is a flexible delay line with the range of delay $[2^{k-1}-1, 2^k-1]$.*

Proof. The proof for Theorem 12 is quite similar to the proof for showing that Cantor networks are non-blocking switches (see e.g., [19]). Consider the n^{th} packet. Let $\tau^a(n)$ be its arrival time and $\tau^d(n)$ be its departure time. Suppose that the delay of the n^{th} packet, denoted by d, is in the range $[2^{k-1}-1, 2^k-1]$, i.e.,

$$2^{k-1}-1 \leq d \leq 2^k-1.$$

First, we claim that the total number of feasible paths for the n^{th} packet is $m2^{k-1}$. To show this, it suffices to argue that the total number of feasible paths through a particular Benes time slot interchange is 2^{k-1}. Let d_1 be the delay to the *input link* of the k^{th} stage of a particular Benes time slot interchange. As each delay line can be traversed at most once, we have $0 \leq d_1 \leq 2^{k-1}-1$. Once d_1 is chosen, the path to the input link of the k^{th} stage is uniquely determined by the binary representation of d_1. On the other hand, since $2^{k-1}-1 \leq d \leq 2^k-1$, we have $0 \leq d-d_1 \leq 2^k-1$. In view of the constraint that each delay line can be traversed at most once, the path from the input link of the k^{th} stage to the output of the Benes time slot interchange is also uniquely determined by the binary representation of $d-d_1$. Since there are 2^{k-1}

choices of d_1, there are 2^{k-1} feasible paths through a Benes time slot interchange.

Let S_1 be the set of *feasible* paths for the n^{th} packet from the input of the Cantor network to the m input links of the optical memory cells at the k^{th} stage. As argued in the previous paragraph, each path in S_1 corresponds to a feasible path from the input to the output. Thus, we have $|S_1| = m2^{k-1}$. Moreover, the delay of a path in S_1 is between 0 and $2^{k-1}-1$. As such, these paths in S_1 might be in conflict with those packets that arrive during $\tau^a(n)-1, \tau^a(n)-2, \ldots, \tau^a(n)-(2^{k-1}-1)$.

Now we claim that the maximum number of paths in S_1 that are in conflict with the packet arriving at $\tau^a(n)-1$ is at most 2^{k-2}. First, note that the packet that arrives at $\tau^a(n)-1$ might be in conflict with the n^{th} packet at various stages (from 2 to k). However, the worst case that results in the maximum number of conflicting paths is to have a conflict at the earliest stage, i.e., the second stage. To achieve this, suppose that the packet that arrives at $\tau^a(n)-1$ is delayed by 1 to the input link of the k^{th} stage in one of the m Benes time slot interchanges. In this case, if the n^{th} packet would still like to use the same Benes time slot interchange, it cannot be delayed by 0 at the first stage. As the number of paths for that Benes time slot interchange with delay 0 at the first stage is 2^{k-2}, the number of conflicting paths in this case is 2^{k-2}.

In general, the maximum number of paths in S_1 that are in conflict with the packet arriving at $\tau^a(n)-s$ for some $2^{j-2} \leq s \leq 2^{j-1}-1$ and $2 \leq j \leq k$, is 2^{k-j}. This is because the earliest conflict can only occur at the input link of the j^{th} stage of a Benes time slot interchange (as the maximum delay for the n^{th} packet to be at the input link of the $(j-1)^{th}$ stage is $2^{j-2}-1$). As such, the maximum number of paths in S_1 that are in conflict with those packets arriving during $[\tau^a(n)-2^{j-2}, \tau^a(n)-(2^{j-1}-1)]$ is $2^{j-2}2^{k-j}$. Hence, the maximum number of paths that are in conflict with those packets arrive during $[\tau^a(n)-1, \tau^a(n)-(2^{k-1}-1)]$ is $(k-1)2^{k-2}$.

Define a *reachable* path as a *feasible* path that is not conflicting with other packets (ahead of the n^{th} packet). Let $\tilde{S}_1$ be the set of *reachable* paths for the n^{th} packet from the input link of the Cantor network to the m input links of the optical memory cells at the k^{th} stage. Clearly, we have

$$|\tilde{S}_1| \geq |S_1| - (k-1)2^{k-2} = m2^{k-1} - (k-1)2^{k-2}. \quad (41)$$

Similarly, one can define S_2 (resp. $\tilde{S}_2$) to be the set of *feasible* (resp. *reachable*) paths for the n^{th} packet from the output of the Cantor network to the m output links of the k^{th} stage. As the Cantor network is symmetric (its mirror image is itself), we also know that $|S_2| = m2^{k-1}$. Moreover, these paths in S_2 might be in conflict with those packets that depart during $[\tau^d(n)-1, \tau^d(n)-(2^{k-1}-1)]$ and $[\tau^d(n)+1, \tau^d(n)+(2^{k-1}-1)]$. Analogous to the argument used in the previous paragraph, one can show that the maximum number of paths that are in conflict with those packets departing during $[\tau^d(n)-1, \tau^d(n)-(2^{k-1}-1)]$ is $(k-1)2^{k-2}$. Similarly, the maximum number of paths that are in conflict with those packets departing during $[\tau^d(n)+1, \tau^d(n)+(2^{k-1}-1)]$ is also $(k-1)2^{k-2}$. Thus,

$$|\tilde{S}_2| \geq m2^{k-1} - 2(k-1)2^{k-2}. \quad (42)$$

If $m \geq \frac{3}{2}k$, it then follows from (41) and (42) that

$$|\tilde{S}_1| + |\tilde{S}_2| > m2^{k-1}. \quad (43)$$

Since the total number of feasible paths for the n^{th} packet is $m2^{k-1}$, it follows from (43) that there must be at least one feasible path that is in the intersection of $\tilde{S}_1$ and $\tilde{S}_2$. Thus, there must be a reachable path from the input of the Cantor network to the output of the Cantor network. ■

As discussed in the previous section, there are not enough feasible paths for packets with short delay, especially in the middle stages of the Cantor network. To construct a flexible delay line with maximum delay 2^k-1, one may simply add a flexible delay line with maximum delay $2^{k-1}-1$ parallel to the $(\lceil \frac{3}{2}k \rceil, k)$-Cantor network as in the previous section. Since the number of 2×2 switches in the $(\lceil \frac{3}{2}k \rceil, k)$-Cantor network is $O(k^2)$, recursively expanding such a construction yields a flexible delay line with maximum delay 2^k-1 that needs $O(k^3)$ 2×2 switches.

Here we propose a better alternative to solve the problem for packets with short delay. The idea is to add redundancy to each Benes time slot interchange so that packets with short delay can bypass the middle stages. As illustrated in Figure 10, we consider a "layered" (m,k)-Cantor network by inserting a 1×2 switch before the optical memory cell at the j^{th} stage for $j = 1, 2, \ldots, k-1$ and a 2×1 switch after the optical memory cell at the j^{th} stage for $j = k+1, k+2, \ldots, 2k-1$. In addition to these, we add another scaled optical memory cell with scaling factor 2^{j-1} at the j^{th} stage, $j = 1, 2, \ldots, k-1$. For the newly added optical memory cell at the j^{th} stage, its input link is connected to the upper output link of the 1×2 switch before the optical memory cell at the j^{th} stage, and its output link is connected to the upper input link of the 2×1 switch after the optical memory cell at the $2k-j^{th}$ stage. For a packet with delay between $2^{j-1}-1$ and 2^j-1, $j = 1, 2, \ldots, k-1$, it is then routed through the newly added optical memory cell at the j^{th} stage. This is equivalent to embedding an (m,j)-Cantor network, $j = 1, 2, \ldots, k-1$, inside the layered (m,k)-Cantor network. By so doing, a packet with the delay ranging between $2^{j-1}-1$ and 2^j-1 can be routed by the embedded (m,j)-Cantor network. This is stated in the following corollary.

Corollary 13 *Suppose that every delay line in the layered (m,k)-Cantor network can be traversed by a packet at most once. If $m \geq \frac{3}{2}k$, then the layered (m,k)-Cantor network (see Figure 10) is a flexible delay line with maximum delay 2^k-1.*

Proof. It suffices to show that a packet with the delay ranging between $2^{j-1}-1$ and 2^j-1 can be routed by the embedded (m,j)-Cantor network. The argument for this is exactly the same as that in the proof of Theorem 12. ■

Note that the number of 2×2 switches in the layered $(\lceil \frac{3}{2}k \rceil, k)$-Cantor network is still $O(k^2)$. This implies that one can construct a flexible delay line with maximum delay B by using $O((\log B)^2)$ 2×2 switches.

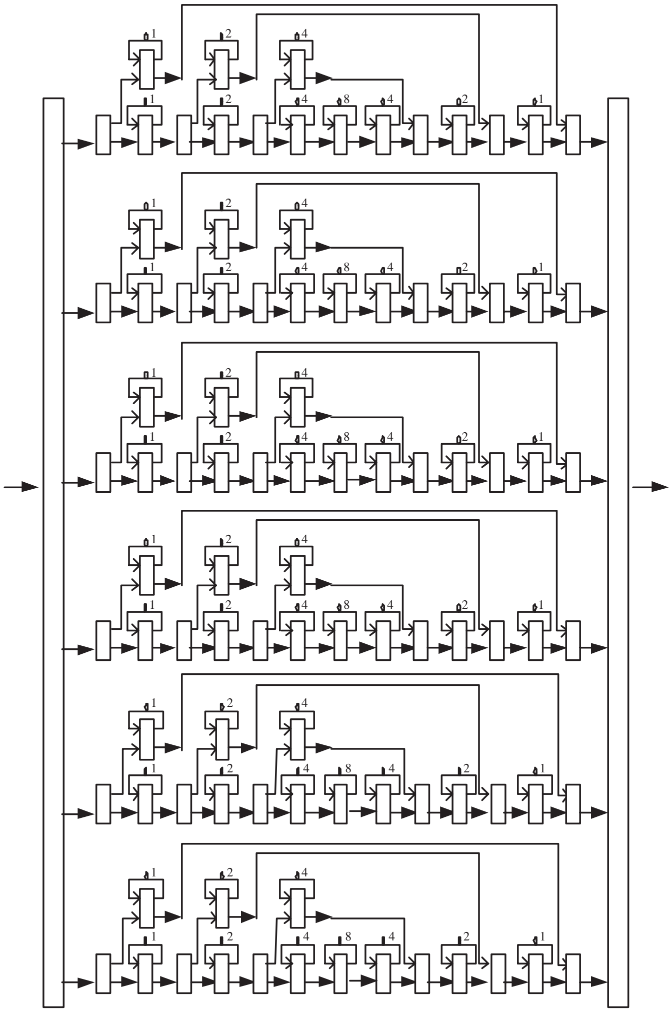


Fig. 10. The layered $(6, 4)$-Cantor network

VII. Conclusions

With the help from the classical switching theory, in this paper we developed mathematical theory for multistage constructions of linear compressors, non-overtaking delay lines and flexible delay lines. In Section III, we showed that there is a two-stage construction of a linear compressor. Via recursive expansion of the two-stage construction, one can then construct a linear compressor via a series of scaled optical memory cells. In Section IV, we showed that there is a three-stage construction of a non-overtaking delay line. This is a generalization of the two-stage construction for a non-overtaking line in [8]. In Section V, we showed that there is a three-stage construction of a flexible delay line. Such a result is analogous to the three-stage Clos network for a nonblocking switch. We used the Cantor network in Section VI to construct a flexible delay line. The complexity for constructing a flexible delay line by the Cantor network is less than that by the three-stage construction.

There are some research problems that need further investigation. We have shown that one can construct a flexible delay line with maximum delay B by using $O((\log B)^2)$ 2×2 switches. Is it possible to construct a flexible delay line with $O(\log B)$ complexity? For this, one might look into the expanders in [26], [25], [1], [13], [27]. In the classical switching theory, one might construct a nonblocking switch by sorting, e.g., the Batcher-banyan networks [2]. How can we construct a "sorting line" so that overtaking can be avoided? If this is feasible, then a flexible delay line can be constructed by a concatenation of a "sorting line" and a non-overtaking delay line.

Acknowledgment

This research is 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, and the Program for Promoting Academic Excellence of Universities NSC 94-2752-E-007-002-PAE.

References

[1] L. A. Bassalygo and M. S. Pinsker, "Complexity of optimal non-blocking switching networks without rearrangement," *Problems of Information Translations*, New York, Plenum, Vol. 9, pp. 64-66, 1974.

[2] K. E. Batcher, "Sorting networks and their applications," *Proc. 1968 Spring Joint Comput. Conf.*, Vol. 32, pp. 307-314, 1968.

[3] V. E. Benes. *Mathematical Theory of Connecting Networks and Telephone Traffic*. New York: Academic Press, 1965.

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

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

[6] 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.

[7] 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, 2000.

[8] C.-S. Chang, Y.-T. Chen, and D.-S. Lee, "Construction of optical FIFO queues and non-overtaking delay lines," *Technical Report*, 2005.

[9] C.-S. Chang, D.-S. Lee, and C.-K. Tu, "Using switched delay lines for exact emulation of FIFO multiplexers with variable length bursts," to appear in *IEEE Journal on Selected Areas in Communications*. Conference version in *Proceedings of IEEE INFOCOM*, 2003.

[10] 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.

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

[12] 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, 1996.

[13] O. Gabber and Z. Galil, "Explicit construction of linear size superconcentrators," *J. Comput. Syst. Sci.,* Vol. 22, pp. 407-420, 1981.

[14] A. Huang and S. Knauer, "Starlite: a wideband digital switch," *Proc. Globecom*, 1984.

[15] D. K. Hunter and I. Andonovic, "Approaches to optical Internet packet switching," *IEEE Communications Magazine*, Vol. 38, pp. 116-122, 2000.

[16] 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.

[17] D. K. Hunter, M. C. Chia, and I. Andonovic, “Buffering in optical packet switches,” *IEEE Journal of Lightwave Technology*, Vol. 16, pp. 2081-2094, 1998.

[18] 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.

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

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

[21] H. F. Jordan, D. Lee, K. Y. Lee, and S. V. Ramanan, “Serial array time slot interchangers and optical implementations,” *IEEE Transactions on Computers*, ,Vol. 43 , No. 11 , pp. 1309-1318, 1994.

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

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

[24] M. J. Marcus, “Designs for time slot interchangers,” *Proc. National Electronics Conf.*, Vol. 26, pp. 812-817, 1970.

[25] G. A. Marguis, “Explicit constructions of concentrations,” *Problems Peredachi Informatsii*, Vol. 9, No. 4, 1973. (In English: Problems of Information Translations, New York, Plenum, 1975).

[26] M. S. Pinsker, “On the complexity of a concentrator,” *Proceedings of the Seventh International Teletraffic Congress*, Stockholm, 1973.

[27] N. Pippenger, “Juggling networks,” *Canada-France Conference on Parallel and Distributed Computing*, pp. 1-12, 1994.

[28] S. V. Ramanan, H. F. Jordan, and J. R. A. Sauer, “A new time domain, multistage permutation algorithm,” *IEEE Transactions on Information Theory*, Vol. 36 , No. 1 , pp. 171-173, 1990.

[29] 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*.

[30] M. Schwartz, *Broadband Integrated Networks*. New Jersey: Prentice Hall, 1996.

[31] 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.

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

[33] S. Yao, B. Mukherjee, and S. Dixit, “Advances in photonic packet switching: An overview,” *IEEE Communications Magazine*, Vol. 38, pp. 84-94, 2000.

[34] M. Yoo, C. Qiao, and S. Dixit, “QoS performance of optical burst switching in IP-over-ATM networks,” *IEEE Journal on Selected Areas in Communications*, Vol. 18, pp. 2062-2071, 2000.