Optical TDM switching architectures with reduced control complexity

D.K. Hunter
D.G. Smith

Indexing terms: Optical TDM switching, Cantor networks, Attenuation and crosstalk performance

Abstract: Architectures for optical TDM switching with low control complexity, which would be suitable for implementation with lithium niobate switches and fibre delay lines, are discussed. The networks are mathematically equivalent to strict-sense nonblocking Cantor networks, and use the same control algorithm. The penalty paid for low control complexity is a large number of components. All such networks use 'feed-forward' delay lines which provide superior crosstalk performance and more uniform attenuation than existing designs. Architectures, both with and without frame integrity, are considered; performance is characterised in terms of attenuation and crosstalk.

1 Introduction

Lithium niobate optical switching is the most mature technology presently available for future broadband communications switching systems. This paper considers architectures for optical TDM switching which have low control complexity, and would be useful in applications such as high definition television (HDTV) and videophone.

The advantages of lithium niobate technology are well known: it is well-established, and offers transparency to data format, bitrate and wavelength. For high-speed systems, the electronic control circuitry may restrict the speed of the system. Existing work by the present authors [1, 2] discusses TDM switching architectures that need the same control algorithms as Beneš and Waksman networks. Although these networks use few components, the control complexity is high, and for applications requiring frequent reallocation of the connections between channels, a switch network with simplified control complexity would be advantageous. This paper describes such a network architecture, wherein the decrease in control complexity is made at the expense of an increased number of switches.

As before [1, 2], the architectures consist of 2×2 switches and fibre delay lines. Other technologies will be possible in the future. The use of delay line memories allows block multiplexing to be used [3], where each timeslot can contain many bits, and each timeslot may carry a different bitrate. There is a guard band between blocks to allow the switches time to change state. The use of block multiplexing implies that the optical data rate may be many times greater than that of the electrical control signals.

Architectures by other authors, requiring a large number of switches [4, 5], have uneven attenuation [6] and exhibit unsatisfactory crosstalk performance. The architectures described here use 'feed-forward' delay lines, thus solving these problems. In addition to describing the architectures themselves, attenuation and crosstalk are also discussed. Architectures both with and without frame integrity are considered. (A network is said to have frame integrity if all the blocks entering on one frame leave on the same frame.)

2 Definitions, notation and discussion

2.1 Notation

A network which is functionally equivalent to a TST network is represented by $T(m, n)$. m is the number of input and output TDM links, and n is the number of timeslots per frame. $T(m, n)$ represents the network *function* only, and not its internal structure; the notation represents any of a variety of networks with differing characteristics regarding hardware requirement, crosstalk, attenuation and frame integrity. If the network has frame integrity, μ is the frame delay, i.e. the delay in timeslots between the start of an input frame and the start of the corresponding output frame.

In general, if Θ represents any capital letter, a network $\Theta(a_1, a_2, \ldots, a_i)$ will be called a Θ-network. For example, $T(4, 8)$ is a T-network.

There are four other types of network which will be used; these are $H(m, n, h)$, $I(m, n, h)$, $E(j)$ and $W(m, n, h, j_s)$. They will be explained later, and are intermediate steps in the creation of T-networks, which are the desired end result. They do not necessarily represent useful systems when used in isolation. The order in which input terminals or output terminals are connected up on any network, except $E(j)$ or $W(m, n, h, j_s)$, is not important.

A 'rule' is a statement that shows how to obtain a larger network from one or more smaller ones, called 'subnetworks', with the addition of any necessary switches and delay lines. To apply such a rule, one takes an existing subnetwork, duplicates it if necessary, and adds extra switches and delay lines to produce a bigger network, as described by the rule.

Each rule is defined by a diagram which shows explicitly how to build a larger network from its constituent subnetwork(s). In rules where there are two subnetworks, it is assumed for convenience that they are identical (i.e. one is an exact replica of the other).This need not be the

Paper 9459I (E7, E13), first received 18th March 1992 and in revised form 5th January 1993

The authors are with the Department of Electronic and Electrical Engineering, University of Strathclyde, 204 George Street, Glasgow G1 1XW, United Kingdom

case in general, although the functions they perform are always the same.

To produce the desired $T(m, n)$ network, one begins with a small subnetwork called the 'starting point', and applies a series of rules until the desired network is produced. The application of such a series of rules is called a 'derivation'. A 'space expansion' involves applying a rule where the resulting network has more inputs and outputs than the subnetworks. The application of any other rule is called a 'time expansion'. A 'time interconnect' is an interconnect between two adjacent stages which contains delay lines; any other interconnect is a 'space interconnect'.

2.2 Switch network control

Electronic control signals drive the directional coupler switches in all the switching networks in this paper. The arrangement required to provide these signals is shown in Fig. 1. To appreciate how the control arrangement

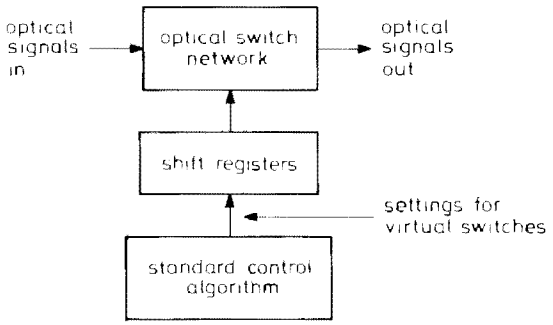

Fig. 1 *Arrangement for controlling a T-network*

works, it is necessary to consider a space switch which is related in structure to the optical switch network being controlled. This relationship is defined by a 'space-time mapping' which is discussed elsewhere [1]. The switches in this space switch are called 'virtual switches'. Under the space–time mapping, each virtual switch represents a switch in the optical switching network on one timeslot [1]. In this paper, the space switch is always a Cantor network [7], which is strict-sense nonblocking.

To control the optical switch network, standard control algorithms [8, 9] are used to find a free path through the space switch for each new call, without rearranging existing calls. The control algorithm thus computes the settings on the virtual switches. Each shift register accepts the virtual switch settings in parallel, and outputs them cyclically to the optical switch network. The shift registers are the only part of the electronics that needs to operate at the timeslot rate.

The control algorithm needs only to recompute the virtual switch settings when the assignment is changed; the complexity of the control algorithm and the speed of the hardware dictate how often this is possible. Other networks we have reported required the Beneš network control algorithm [1, 2] which had control complexity of $O(mn\log_2 mn)$. Although the architectures in this paper require more switches, a control algorithm with control complexity of $O((\log_2 mn)^2)$ may be used [8]. Thus, there is a trade-off between control complexity and hardware utilisation.

2.3 Bitwise logical operators

It is necessary to describe the timeslots on which the blocks can enter and leave the switch network, and bitwise logical operators provide a convenient mathematical language for doing this. Three operators are used which all operate on integers in the range 0 to $n-1$ inclusive. n is the desired number of timeslots per frame in the finished T-network and must be an integral power of 2.

To show how the operators work, the numbers they operate on must be thought of as $\log_2 n$-digit binary numbers. The AND operator, '$\wedge$' indicates that if

$$c = a \wedge b$$

then a bit in c is set to '1' if, and only if, *both* the bits in the same position in a and b are set to '1'. Similarly, '$\vee$' represents OR and '$\otimes$' represents EXCLUSIVE–OR.

2.4 H-Networks and I-networks

Two types of network, known as $H(m, n, h)$ and $I(m, n, h)$, are used during the derivation of the required T-network. As usual, m is the number of inputs and outputs; n is the number of timeslots per frame. h keeps track of the timeslots between which the blocks may be interchanged. It is defined thus: a block entering a network on timeslot p can leave it on timeslot q, if, and only if,

$$p \vee h = q \vee h \tag{1}$$

That is, two bits in the same position in p and q may only differ if the corresponding bit in h is '1'. This relationship between p and q is an equivalence relation, and it is convenient to represent it in terms of a partition of equivalence classes. Let $N = \{0, 1, \ldots, n-1\}$ be the set of timeslots, and let $U = \{U_1, U_2, \ldots, U_e\}$ be a partition of N defined by

$$p \in U_i, q \in U_i \Leftrightarrow p \vee h = q \vee h$$

A channel can enter an H or I-network on timeslot p and leave on timeslot q if, and only if, p and q are both in the same equivalence class. There are e equivalence classes in the partition; each one contains n/e timeslots. Define 'during U_i' as meaning 'on any timeslot p such that $p \in U_i$'.

Now consider $h = 0$. According to eqn. 1, a block entering on timeslot p can only leave on timeslot q if $p \vee h = q \vee h$. In this case, $p \vee 0 = q \vee 0$, or $p = q$. So no interchange of blocks between timeslots is possible, and $H(m, n, 0)$ and $I(m, n, 0)$ are both different types of time-multiplexed space switches. Such time-multiplexed space switches are important not in their own right, but as fundamental building blocks in larger architectures. This will be described in the following Section.

The opposite extreme is $h = n-1$. $n-1$ is, in binary, all '1's since n is an integral power of 2, so $p \vee (n-1) = q \vee (n-1)$ irrespective of p and q. $I(m, n, n-1)$ and $H(m, n, n-1)$ therefore both allow full interchange of blocks between timeslots, and it will be shown that they can be used to make the final T-network.

2.5 Optical switch model

The crosstalk of a single 2×2 switch is X dB, and its loss is L dB. X is always negative. Thus the signal is attenuated by L dB over the correct route through the switch, and $L - X$ dB over the incorrect route. The loss at each fibre/substrate or substrate/fibre interface is W dB. SXR is the signal-to-crosstalk ratio of the whole switching network, while the total insertion loss is A. It is assumed that X and L do not vary for different devices in the same network. To ensure mathematical tractability, interference effects in the switches are ignored; crosstalk is assumed to add to the signal linearly.

Two sets of values for X, L and W are used in calculations. The 'best case' values represent the best devices

available [10]; $X = -35$ dB, $L = 0.25$ dB, $W = 0.5$ dB. The 'worst-case' values represent relatively poor performance [11]; $X = -20$ dB, $L = 1$ dB, $W = 2$ dB.

The attenuation calculations only take account of the loss due to the directional coupler and fibre/substrate interfaces. No attempt is made to consider waveguide losses, or losses in waveguide bends and crossovers, as the complexity introduced would make the calculations intractable. The equations for attenuation should therefore only be taken as a rough guide.

The permissible values of SXR and A are application dependent, so figures have been chosen merely to give some indication of the characteristics of the switching systems. The minimum value of SXR is taken as 11 dB. For a BER of 10^{-9} at the input to a network, this represents a BER of 1.6×10^{-9} at the output for an undilated network made from worst case switches [12]. For best case devices, or a dilated network, the SXR at the output is better than 1.1×10^{-9}, limited by the accuracy of the calculation procedure used. This indicates that the BER is much less sensitive to crosstalk than it is to noise, due to its fundamentally different statistical properties. A generous maximum attenuation of $A = 25$ dB has been chosen; very few practical systems would be able to accommodate a greater loss.

3 Network architectures

This Section describes the new network architectures, which have lower control complexity than the networks described in previous papers [1, 2]. The networks are mathematically equivalent to Cantor networks [7], and require the same control algorithms [8, 9], as discussed above.

3.1 Discussion

It is necessary to extend the definitions given in the previous Section. Since these definitions and results were considered in detail elsewhere [1], they will only be outlined here.

The time expansions described here all use a type of network which will be called an E-network. These networks are used, along with subnetworks and 2×2 switches to create a larger network where the equivalence classes U_i are twice as large as in the original subnetworks. They have two inputs and outputs, and are defined in Figs. 2 and 3. There are two varieties; E_{NI}-networks do not have frame integrity, and E_I-networks do. Associated with each E-network is a mapping π which is $\pi(p) = p + j \bmod n$ for E_{NI}-networks, and $\pi(p) = p \oplus j$ for E_N-networks. Assuming a maximal assignment, two blocks will enter the network on timeslot p. One emerges on timeslot p and the other emerges on timeslot $\pi(p)$. With the E_{NI}-network, there is a delay of j timeslots between the start of a frame on the input to the network and the corresponding output frame. Using the notation in the previous Section, it can be shown that if $U_i \in U$ then $\pi(U_i) \in U$ and $\pi(\pi(U_i)) = U_i$. For E_{NI}-networks (i.e. without frame integrity), $\pi(\pi(U_i)) = U_i$ only if j, the length of delay line, is half the length of the shortest delay line in the subnetwork. Also, it can be shown that $V = \{V_k \mid V_k = U_i \cup \pi(U_i)\}$ is a partition of the set of timeslots, where $p \in V_k$, $q \in V_k \Leftrightarrow p \vee (h \vee j) = q \vee (h \vee j)$.

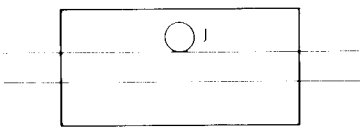

Fig. 2 *Definition of $E_{NI}(j)$*

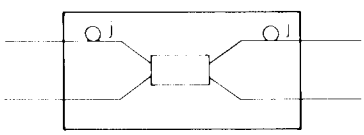

Fig. 3 *Definition of $E_I(j)$*

$H(m, n, h)$ is a network with m inputs and n timeslots per frame, where a block may enter on timeslot p and leave on timeslot q only if $p \vee h = q \vee h$. There are mn calls in a maximal assignment. There are Mm outputs; the meaning of the constant M will be explained shortly. $I(m, n, h)$ is the mirror image of $H(m, n, h)$; it has Mm inputs and m outputs. $c(x)$ is the number of output channels that a free input can reach in an H-network in the presence of $mn - 1$ other calls without rearranging any existing calls; in an I-network it is the number of input channels that a free output can reach. x is the number of expansions, both in space and time, that have taken place in the derivation so far. It will be shown that $c(x)$ is not dependent on the order in which these expansions were applied.

The proof is similar in concept to that used for the Cantor network [7]; here, the objective is to show that, in the presence of $mn - 1$ existing calls, the remaining idle input and idle output channel can each access more than half the virtual switches on the centre stage without rearranging the existing calls. If they can, there must be at least one virtual switch in common, and the network is therefore strictly nonblocking; it can accommodate any new call without re-routing existing calls.

Only the proof for H-networks will be given below, since the proof for I-networks is very similar.

3.2 Definition of starting points

The networks $H(1, n, 0)$ and $I(1, n, 0)$ are simply a $1 \times M$ demultiplexer and a $M \times 1$ multiplexer, respectively (Fig. 4). A block on the input of $H(1, n, 0)$ can reach all M outputs on the same timeslot, since there are no other calls, so $c(0) = M$. The value of M will be determined later.

Fig. 4 *Definition of $H(1, n, 0)$ (left) and $I(1, n, 0)$ (right)*

3.3 Space expansion of H and I-networks

Rules 5 and 6 represent space expansions of H and I-networks respectively. Consider the calls in one equivalence class U_i in Fig. 5. Since a block must leave a subnetwork on the same equivalence class that it entered on, this condition must also apply to the whole network since there are no delay lines of any kind outside the subnetworks. Suppose, without loss of generality, that the idle input channel is on the lower subnetwork; all the other $mn - 1$ channels carry a call. Let x be the number of space and time expansions used to produce each of the subnetworks. The free input can reach $c(x)$ of the output virtual switches, and in the absence of the calls in the upper subnetwork, could reach $2c(x)$ outputs. But there are $m|U_i|/2$ calls in the upper subnetwork. So, in the worst case, $c(x + 1) = 2c(x) - m|U_i|/2$. The number of

inputs on a subnetwork, $m/2$, is equal to two to the power of the number of space expansions so far, since each space expansion doubles m. Similarly, $|U_i|$ is equal

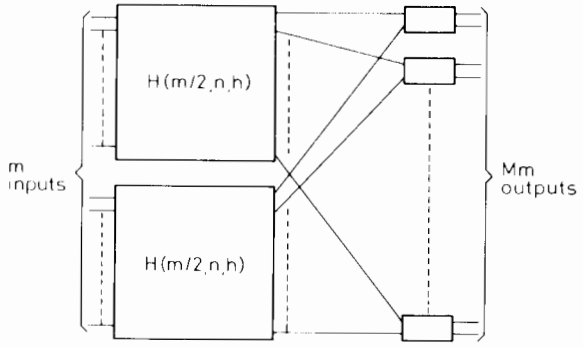

Fig. 5 *Space expansion to obtain H(m, n, h)*

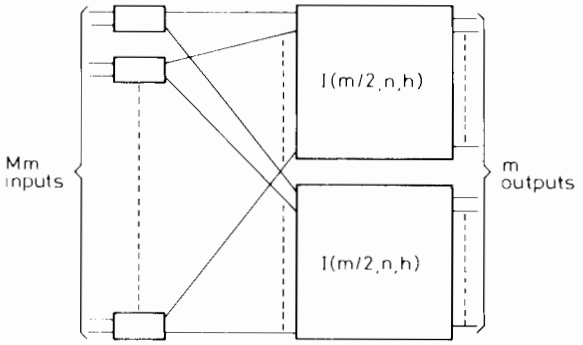

Fig. 6 *Space expansion to obtain I(m, n, h)*

to two to the power of the number of time expansions so far because after each time expansion, the size of the equivalence classes has doubled. It follows that $m|U_i|/2 = 2^x$, so

$$c(x+1) = 2c(x) - 2^x \tag{2}$$

3.4 Time expansion of H and I-networks

Figs. 7 and 8 show how time expansions can be applied to H and I-networks. In both cases, Mm must be a multiple of 2. To ensure that this is true even if M is not a multiple of 2, one can make a space expansion the first expansion in the derivation. If $m = 1$ in the finished

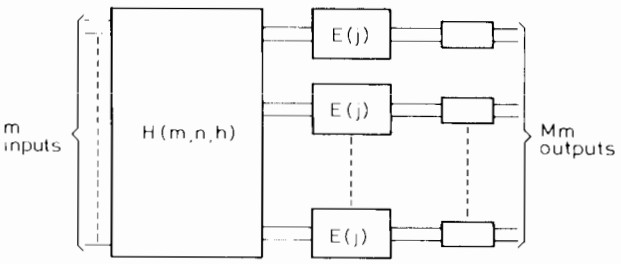

Fig. 7 *Time expansion to obtain H(m, n, h∨j)*

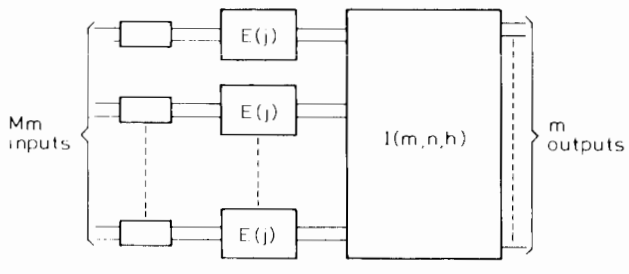

Fig. 8 *Time expansion to obtain I(m, n, h∨j)*

network, no space expansions are required, so M must be rounded up to the closest multiple of 2. This will not be taken into account in the following calculations, in the interest of clarity. The switches need not be connected to the E-networks precisely as shown. The important point is that in Fig. 7, one input of each output switch is connected to the upper output of any E-network, and the other input is connected to the lower output of any E-network. Similarly, in Fig. 8, one output of each input switch must be connected to the upper input of any E-network, and the other output is connected to the lower input of any E-network.

Suppose that there are $|V_k| - 1$ calls in progress on V_k and assume, without loss of generality, that the idle input channel is in U_i. This idle input can reach $c(x)$ output virtual switches without disturbing existing calls, where x is the total number of space and time expansions required to produce the subnetwork. This is because the idle input can reach a total of $c(x)$ channels on the inputs to the E-networks, and each of these input channels maps onto a different output virtual switch. If there were no calls in $\pi(U_i)$, then the idle input channel could reach $2c(x)$ output channels. But since $\pi(U_i)$ contains $m|U_i|$ channels, $c(x+1) = 2c(x) - m|U_i|$ in the worst case, which reduces to

$$c(x+1) = 2c(x) - 2^x \tag{3}$$

This is the same as eqn. 2, demonstrating that $c(x)$ depends only on the value of x, and not on the order in which space and time expansions take place.

3.5 Calculation of M

From the discussions above, $c(0) = M$ and $c(x+1) = 2c(x) - 2^x$. The solution is

$$c(x) = 2^x M - x2^{x-1}$$

Now suppose that a network $T(m, n)$ is formed by taking a network $H(m, n, n-1)$, and connecting each output to an input on a network $I(m, n, n-1)$, so that both outputs of any final-stage switch in $H(m, n, n-1)$ are connected to the same first stage switch in $I(m, n, n-1)$. Two switches which are connected in this way can be joined into a single switch. Each such pair of switches becomes one of the $Mm/2$ switches in the centre stage of the T-network. The worst case number of virtual switches that can be reached from either the idle input or idle output is $c(\log_2 mn - 1)$, since $\log_2 m$ space expansions and $\log_2 n$ time expansions are necessary to form each of the H-networks and I-networks. The number of centre-stage virtual switches is $Mmn/2$, so the condition for the network to be strict-sense nonblocking is

$$c(\log_2 mn - 1) > \frac{Mmn}{4}$$

i.e.

$$2^{\log_2 mn - 1} M - (\log_2 mn - 1)2^{\log_2 mn - 2} > \frac{Mmn}{4}$$

This reduces to

$$M > \log_2 mn - 1$$

so the smallest value of M which will give a strict-sense nonblocking network is

$$M = \log_2 mn \tag{4}$$

If $m = 1$, and M is not a multiple of 2, M must be rounded up, so it becomes $M = \log_2 mn + 1 = \log_2 n + 1$.

This was discussed in the previous Section. For the sake of clarity, this will not be considered in the following calculations.

3.6 Frame delay with frame integrity

On each time expansion of an H or I-network, a stage of $E_I(j)$ networks is added, where j is a different power of 2 between 1 and $n/2$ on each time expansion. There are $\log_2 n$ time expansions on both H and I-networks. Since the frame delay of $E_I(j)$ is j, the cumulative frame delay for the whole T-network is:

$$\mu = 2 \sum_{i=0}^{\log_2 n - 1} 2^i = 2n - 2$$

3.7 Examples

Figs. 9–12 show four examples of small T-networks. Figs. 9 and 10 do not have frame integrity; Figs. 11 and 12 do have frame integrity. Fig. 10 shows that these networks need not be symmetrical; the space and time expansions used to create $H(m, n, n-1)$ and $I(m, n, n-1)$ need not be applied in the same order. In Fig. 9, the largest delay lines are at either end of the network, and the smallest delay lines (of length 1 timeslot) are in the centre. This is true for all the T-networks without frame integrity which are described in this paper.

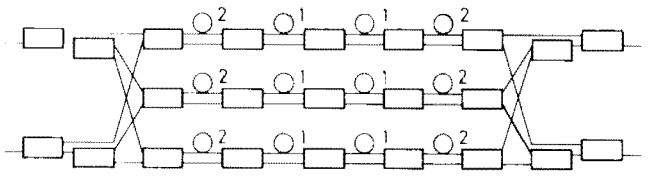

Fig. 9 *T(2,4) without frame integrity*

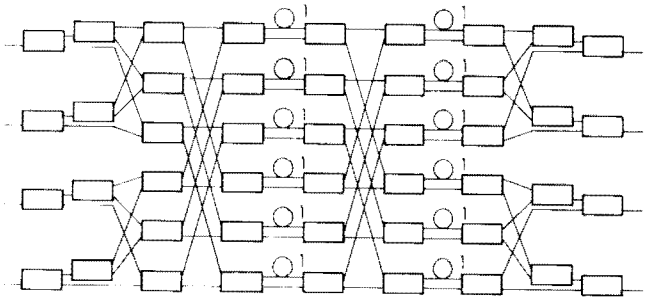

Fig. 10 *T(4,2) without frame integrity*

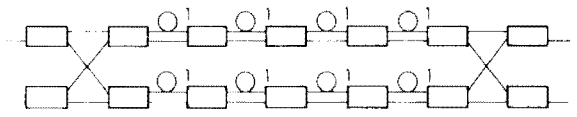

Fig. 11 *T(2,2) with frame integrity*

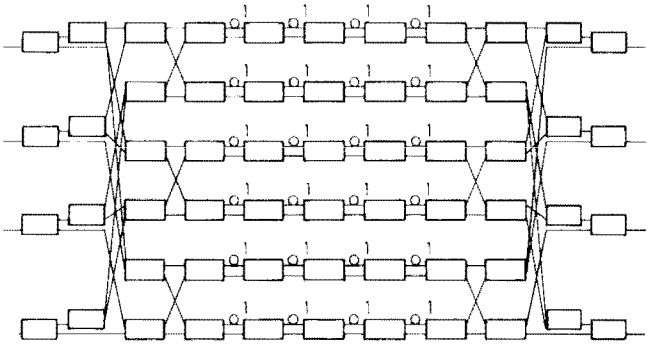

Fig. 12 *T(4,2) with frame integrity*

3.8 Number of switches

Ignoring the demultiplexers and multiplexers, the finished network consists of $2\log_2 mn - 1$ stages without frame integrity and $2\log_2 mn^2 - 1$ stages with frame integrity. This is because each space or time expansion adds one stage to both the H and I-network, except for a time expansion with frame integrity which adds two stages. $\log_2 mn$ expansions are required, hence the above expressions. The '-1' term is there because one stage of switches is lost when the H and I-networks are joined together to from a T-network.

There are $Mm/2$ switches in each stage. Hence the number of switches used in this part of the network is $(Mm/2)(2\log_2 mn - 1)$ without frame integrity, and $(Mm/2)(2\log_2 mn^2 - 1)$ with it. There are also m demultiplexers and m multiplexers, each requiring $M - 1$ switches. This brings the total up to

$$S = \frac{Mm}{2}(2\log_2 mn - 1) + 2m(M - 1)$$

$$= Mm\log_2 mn + \frac{3}{2}Mm - 2m$$

without frame integrity, and

$$S = Mm\log_2 mn^2 + \frac{3}{2}Mm - 2m$$

with frame integrity. Comparing these results with Reference 1 reveals that the penalty paid for the decreased control complexity is that over M times as many switches are required.

3.9 Attenuation

The maximum number of switches that a signal must pass through in a multiplexer or demultiplexer is $\lceil \log_2 M \rceil$; the minimum is $\lfloor \log_2 M \rfloor$. The signal must also pass through $2\log_2 mn - 1$ stages of switches without frame integrity or $2\log_2 mn^2 - 1$ with frame integrity. Making the usual assumption that fibre/substrate transitions only occur due to time interconnects and input/output fibres, the number of transitions is $4\log_2 n + 2$ without frame integrity, and $8\log_2 n + 2$ with it. The maximum total attenuation is, without frame integrity,

$$A = (2\log_2 mn - 1 + 2[\log_2 \log_2 mn])L + (4\log_2 n + 2)W$$

With frame integrity, it is:

$$A = (2\log_2 mn^2 - 1 + 2[\log_2 \log_2 mn])L + (8\log_2 n + 2)W$$

In both cases, the minimum loss is $2L$ dB less, except if M is an integral power of 2, when the minimum loss is the same. This difference arises because if M is not an integral power of 2, the attenuation in a multiplexer or demultiplexer is not uniform. In practice, the difference of $2L$ would be equalised to some extent by waveguide losses.

The restrictions that attenuation imposes on the size of the T-network, for a maximum acceptable attenuation of 25 dB, are summarised in Table 1. Fixed values of m have been used to make the calculations manageable. The attenuation with worst-case devices is unsatisfactory, and optical amplification would have to be used in any real system constructed using these devices, with due regard to noise performance [13, 14].

Table 1: Summary of permissible sizes of T-network for maximum attenuation of 25 dB

Best case devices	Worst case devices
No frame integrity	
$n \leq 256$ if $m = 8$	Only $n = 2$ possible if $m = 2$
Frame integrity	
$n \leq 16$ if $m = 8$	Not possible for any n

3.10 Crosstalk and dilation

The worst-case SXR occurs when a signal meets another signal in each switch it passes through. This excludes switches which are part of a multiplexer or demultiplexer, since each switch only carries one signal at once. The worst case SXR is [1]:

$$SXR \approx -X - 10 \log_{10}(2 \log_2 mn - 1)$$

without frame integrity, and

$$SXR \approx -X - 10 \log_{10}(2 \log_2 mn^2 - 1)$$

with frame integrity. Although it is very unlikely that this worst case would occur, it is nonetheless possible, and must be taken into account.

The network may be dilated. Dilation is a procedure for reducing the crosstalk in the network, at the expense of using more switches. It is a generalisation of the dilated Beneš networks of Padmanabhan and Netravali [15]. Although it is explained fully in Reference 1, it consists of modifying the original network according to the following procedure. First, a four-switch structure is substituted for each switch in the original network (Fig. 13). It is clear that in the modified network, only one block can pass through each switch at once; this drastically reduces crosstalk. This property can be preserved, while reducing the number of switches, by concatenating switches that are joined together as shown in Fig. 14. It is not necessary to carry out the substitution of Fig. 13 on the multiplexers and demultiplexers since only one signal passes through each of their switches at once anyway.

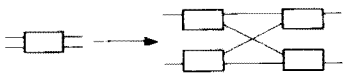

Fig. 13 *First step in dilating a network*

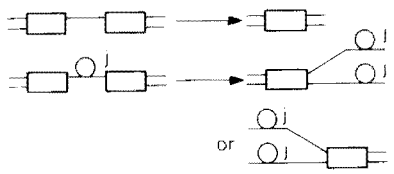

Fig. 14 *Second step in dilating a network*

When calculating the SXR, the crosstalk in the output multiplexers may be ignored since a stray signal must experience attenuation of at least x^3 before it arrives on the wrong output. This makes it negligible in comparison to the other crosstalk signals. The SXR is [1]:

$$SXR \approx -2X - 20 \log_{10}(2 \log_2 mn - 1) + 3$$

without frame integrity, and

$$SXR \approx -2X - 20 \log_{10}(2 \log_2 mn^2 - 1) + 3$$

with frame integrity.

The restrictions on the size of the T-network due to crosstalk are listed in Table 2, for best and worst-case components. They show that dilation would be required to build any non-trivial networks out of worst-case components.

Table 2: Restrictions on network size, assuming worst-case crosstalk for this architecture

Best case components	Worst case components
No frame integrity, undilated	
$mn \leq 8.51 \times 10^{37}$	$mn \leq 16$
Frame integrity, undilated	
$mn^2 \leq 8.51 \times 10^{37}$	$mn^2 \leq 16$
No frame integrity, dilated	
$mn \leq 2.23 \times 10^{189}$	$mn \leq 1048576$
Frame integrity, dilated	
$mn^2 \leq 2.23 \times 10^{189}$	$mn^2 \leq 1048576$

4 Reducing attenuation

4.1 Procedure for reducing attenuation: introduction

This Section describes a procedure for reducing the attenuation in the architectures considered earlier. Since the attenuation reduction procedure in Reference 1 is similar, the details of the proofs are omitted here. The most dramatic reduction occurs with non frame-integrity networks, although it can be used with some success on those with frame integrity also. The procedure is based on two very simple ideas. First, a switch with two delay lines of the same length connected to its inputs is equivalent to a similar switch with the two delay lines connected to its outputs (Fig. 15). That is, the left-hand part of Fig. 15 may be substituted for the right-hand part. The control signal for the switch would have to be delayed in time by j timeslots. The procedure may also be reversed, so that the right-hand part of Fig. 15 is substituted for the left-hand part, and the control signal is advanced by an additional j timeslots. The second idea is even more straightforward and concerns the concatenation of delay lines. This is illustrated in Fig. 16, which should be self-explanatory.

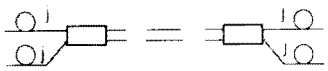

Fig. 15 *Moving delay lines from one side of a switch to another*

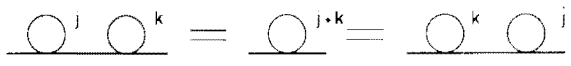

Fig. 16 *Concatenation of delay lines*

Figs. 15 and 16 will be referred to as rule 15 and rule 16, respectively, since, in common with the other rules, they show how one network may be produced from another. Neither rule has any effect on crosstalk, number of switches or control complexity. Rule 15 may be applied to networks which have already been discussed.

4.2 Networks without frame integrity

A general approach to the use of rules 15 and 16 for attenuation reduction is now developed. Here, networks without frame integrity are considered; those with frame integrity are treated later. A new pair of rules is described which may be used to derive H-networks without frame integrity. They correspond to special cases of rules 5 and 7, and are contrived so that the attenuation in the finished network may be minimised by applying rules 15 and 16. A separate set of rules for I-networks will not be given, since this type of network may be produced

by taking any network $H(m, n, n-1)$ and reflecting it about its vertical axis. Before discussing the networks themselves, some terminology and notation must be introduced.

Bitwise logical operators will be used to describe the architecture. As usual, m is the number of inputs, so there are $Mm/2$ 2×2 switches in each stage. Each switch is numbered; the top M switches in a stage are numbered 0, the next M are numbered 1, down to the lowest M which are designated $m/2 - 1$. Excluding the multiplexers and demultiplexers, there are $2 \log_2 mn - 1$ stages in the network, so there are $\log_2 mn$ such stages in the H-network. Let $s = \log_2 mn$, and let these stages be numbered starting from 1 at the leftmost stage, and increasing to the right up to s for the centre stage. $\sigma(y, i)$ denotes any switch numbered i in stage y. Interconnect y is between stages y and $y + 1$.

The type of interconnect used throughout the network is now defined. An (a, j)-interconnect lies between stages y and $y + 1$ ($1 \leqslant y \leqslant s - 1$) if, for all i such that $0 \leqslant i \leqslant m/2 - 1$, one input of each $\sigma(y + 1, i)$ is connected directly to any switch $\sigma(y, \min(i, i \oplus a))$, while the other input is connected to any $\sigma(y, \max(i, i \oplus a))$ via a delay line of length j. a and j are always both integral powers of two or zero; a is called the 'characteristic' of the interconnect. A delay line of length zero is simply a link, so an $(a, 0)$-interconnect is a space interconnect; all others are time interconnects. Fig. 17 shows three examples of these interconnects.

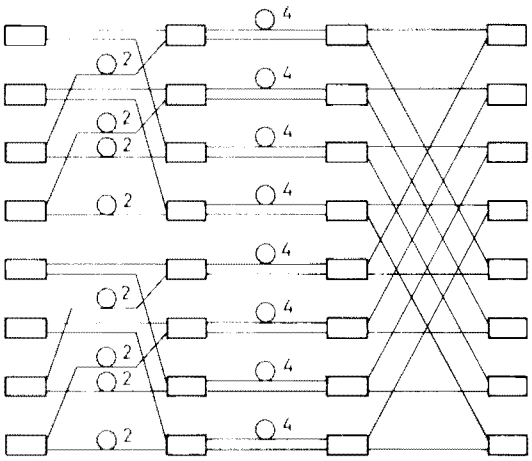

Fig. 17 *From left to right: a (1,2)-interconnect; a (0,4)-interconnect; and a (2,0)-interconnect, with M = 2*

We now state a theorem which states what delay lines may be moved around the network; for details of the proof, see Reference 1. Adding a delay line at an input of a switch means disconnecting the input, and inserting the delay line between the switch input and the line to which it was formerly connected.

Theorem: Let interconnect y be an (a, j)-interconnect, where $a \neq 0$. z is a stage to the left of this interconnect; that is, $1 \leqslant z \leqslant y$. Suppose that no interconnect between interconnect y and stage z has characteristic a. Then interconnect y may be replaced by an $(a, 0)$-interconnect, and delay lines of length j may be added to both inputs of every $\sigma(z, i)$, such that $i \wedge a = a$. ■

This theorem can be used to show how a network with minimum attenuation can be constructed. In the course of the derivation, certain interconnects will be designated 'final time interconnects'. The network is contrived so that after the derivation is over, all the delay lines in the rest of the network may be moved into these final time interconnects, making them the only time interconnects in the finished network.

During the derivation, the network consists of m/m_s H-networks $H(m_s, n, n - 2j_s)$. m_s is the number of inputs on each network; each uses $Mm_s/2$ contiguous switches in each stage. That is, the top network uses switches numbered $0, \ldots, m_s/2 - 1$, the second uses $m_s/2, \ldots, m_s - 1$, and so on. m is the number of inputs on the finished network; j_s is the value of delay line to be used in the next time expansion. These variables are updated as the derivation proceeds. W is a set containing the permitted characteristics that may be used in time interconnects before the next final time interconnect is added. This is chosen so that all the delay lines in any interconnect can be moved to the nearest final time interconnect to its left, and its use will be explained in detail later. The network is complete when $m = m_s$ and $j_s = 1/2$. It starts with the network of Fig. 18; this consists of $m/2$ smaller networks, each representing $H(2, n, 0)$ i.e. $m_s = 2$ and $j_s = n/2$. This will be the centre stage in the finished network.

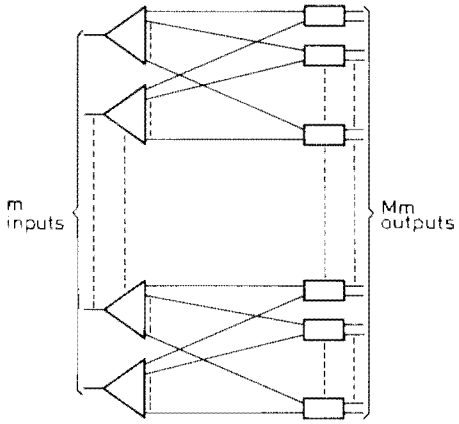

Fig. 18 *Starting point with one space expansion; $m_s = 2, j_s = n/2$*

For a space expansion, a $(m_s/2, 0)$-interconnect is added to the right of the existing network, followed by a stage of $m/2$ 2×2 switches. The value of m_s should now be doubled, ready for the next expansion. If the new value of m_s is m, no further space expansions are possible or necessary. It should be apparent that the interconnect in the first space expansion has characteristic 1; the next one has characteristic 2, and so on, until the final space expansion, which has characteristic $m/4$. Although one could allow for them being applied in any order, this would complicate the explanation, and is not necessary since it is equivalent to merely re-drawing the finished network.

For a time expansion, add an (a, j_s)-interconnect to the right of the existing network, followed by a stage of $m/2$ 2×2 switches. (The possible values of the characteristic a will be discussed shortly.) After the time expansion, j_s must be halved.

Consider what values the characteristic a may have. If $W = \varnothing$, a can be any integral power of 2, providing $a < m_s/2$; it may also be zero. (If $W \neq \varnothing$, the new time interconnect may be forced to be a final time interconnect simply by setting $W = \varnothing$ before connecting up the new stage). After adding the new final time interconnect and switching stage, W should be set to contain all integral powers of two between 1 and $m_s/4$ inclusive, unless $m_s = 2$, when $W = \varnothing$. This is the set of characteristics of all space interconnects currently in the network. Every subsequent time expansion when $W \neq \varnothing$ should produce

a time interconnect whose characteristic a is in W ($a \in W$); a is then removed from W. This continues until another final time interconnect is produced, or the derivation ends.

From the theorem, it can be shown that the delay lines in an interconnect with characteristic a may be shifted to the first final time interconnect on its left.

To minimise attenuation, it is best to maximise the distance between final time interconnects. This is done by doing all $\log_2 m - 1$ space expansions first. Then each final time interconnect will contain the delay lines from up to $\log_2 m$ time interconnects, including itself. The number of final time interconnects in the H-network is therefore $[\log_2 n/\log_2 m]$. The total attenuation in the finished T-network, making the usual assumptions is:

$$A = \left(4\left\lceil \frac{\log_2 n}{\log_2 m} \right\rceil + 2\right)W + (2\log_2 mn - 1 + 2\lceil \log_2 \log_2 mn \rceil)L \quad (5)$$

For the purposes of illustration, assume $m = 16$. Then the network size is restricted to $n \leqslant 524288$ for best case components, although it is not feasible for the worst case. This is a dramatic improvement over the figures quoted in subsection 3.9, since without applying this procedure, and with $m = 8$, the only values of n allowed are $n \leqslant 256$. The crosstalk figures are unaltered by this procedure.

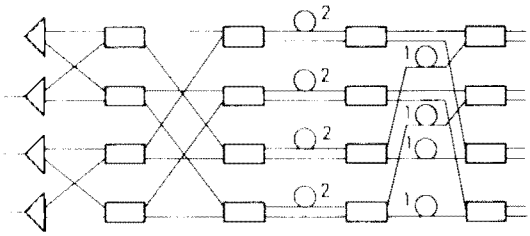

Fig. 19 *H(4,4,3) network (but with $M = 2$) prior to moving all the delay lines into the final time interconnect, which is interconnect 2*

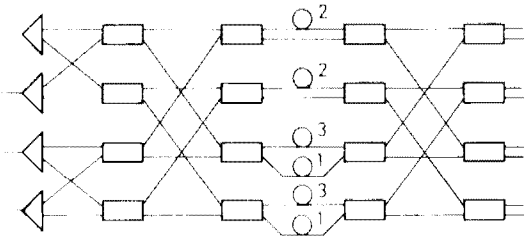

Fig. 20 *H(4, 4, 3) network after all the delay lines have been moved into the final time interconnect*

Stages 3 and 4 may now be fabricated on one substrate

Figs. 19 and 20 represent an example of this procedure, with four inputs, and four timeslots per frame. In a real network the value of M would be 4, but here $M = 2$ for clarity.

4.3 Frame integrity networks

Rules 15 and 16 may also be used to reduce attenuation in those networks which have frame integrity. Consider the frame integrity E_I-network of Fig. 3; one input and one output of the switch are connected to delay lines, while the other input and output are not. Hence rule 15 cannot be applied to the switch in an E_I-network. This is why the attenuation saving possible in a frame integrity network is much less than that for one without frame integrity.

A new type of network, $W(m, n, h, j_s)$, is now defined. It is similar to a $H(m, n, h)$ network with frame integrity, but it differs because the frame boundaries on the outputs do not all coincide. On the outputs, the frame boundaries on the lower $Mm/2$ lines occur j_s timeslots before the corresponding frame boundaries on the upper $Mm/2$ lines. Within either the upper or lower set of $Mm/2$ connections, order is not important, but exchanging of connections between these sets is forbidden. The exception is when $j_s = 0$, and all Mm connections may be rearranged in any order since they are all equivalent. It follows that $H(m, n, h) = W(m, n, h, 0)$.

The starting point for the derivation of a T-network is $W(1, n, 0, 0)$, which is a $1 \times M$ demultiplexer (Fig. 21). The space and time expansions of rules 22 and 23, respectively, may then be used to derive $H(m, n, n - 1) = W(m, n, n - 1, 0)$. Rule 22 is a modification of the space expansion, rule 5. Due to the way the input and output stages are connected, each 2×2 switch always handles two frames which are aligned, while preserving the misalignment between the upper and lower sets of $Mm/2$ connections. Rule 23 is a time expansion for W-networks. Imagine that $j' = 0$ and $j_s = 0$; rule 23 is then equivalent to rule 7 with the E_I-networks drawn out in full.

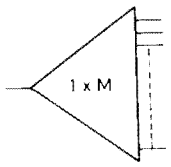

Fig. 21 *Definition of $W(1, n, 0, 0)$*

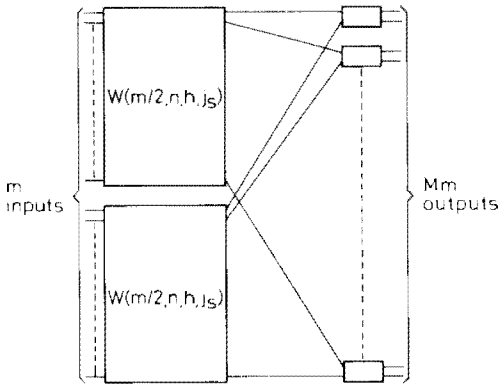

Fig. 22 *Space expansion to produce $W(m, n, h, j_s)$*

As usual, the output stage need not be connected up exactly as shown in Fig. 23. The dotted boxes each enclose what would be an $E_I(j)$-network were $j' = 0$. The contents of these boxes will be called E_I-networks for the sake of convenience, irrespective of the value of j'. Also, the delay lines of length j_s must always remain connected

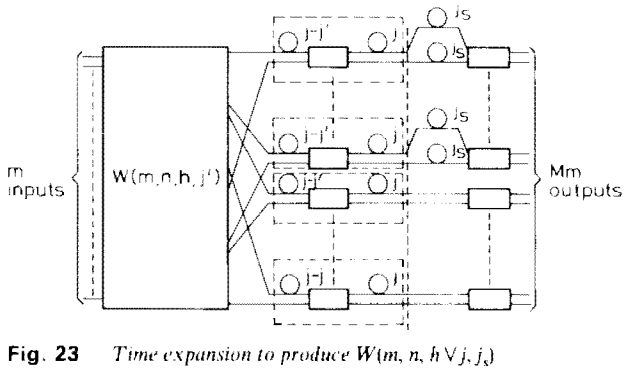

Fig. 23 *Time expansion to produce $W(m, n, h \vee j, j_s)$*

to the same output stage switches. Output stage switches must have one input connected to the upper output of any E_I-network, and the other input connected to the lower output of any E_I-network.

The upper $Mm/2$ terminals on the output of the subnetwork have frame boundaries j' timeslots after the others. This is compensated for by subtracting j' from each delay line connecting to these outputs. The extra delay lines of value j_s are added so that the lower $Mm/2$ outputs of the whole network all have frame boundaries j_s timeslots before the other connections.

To reduce the number of time interconnects, it is possible to set the delay lines on the outputs of the subnetwork to length zero i.e. eliminate them entirely. This is done by ensuring that $j = j'$; it is not possible on the first time expansion since $j' = 0$ always at that point. On all time expansions except the last one, j_s should be set to the value of j that will be used in the next time expansion. This ensures that when this next time expansion takes place, $j = j'$, and the delay lines of length $j - j'$ vanish. On the last time expansion, j_s should be zero, since $W(m, n, n - 1, 0) = H(m, nn - 1)$ with $j_s = 0$, the desired network. Before the output stage, delay lines of value j_s and j which are adjacent may be joined to produce delay lines of length $j_s + j$, by rule 16. Finally, this removal of time interconnects only works if $Mm \geqslant 4$, and Mm is an integer multiple of 4 during each time expansion. Hence it may be necessary to carry out one or two space expansions before any time expansions take place. $I(m, n, n - 1)$ may now be produced by taking a network $H(m, n, n - 1)$ and reflecting it about its vertical axis. $T(m, n)$ may then be obtained in the normal way.

This is equivalent to taking an ordinary frame-integrity network and then applying rules 15 and 16. Therefore, crosstalk, number of switches and control complexity are not affected. Each time expansion introduces two new delay lines in a signal's path, except for the first one, which introduces four. There are two additional fibre/substrate interfaces: one at the input and one at the output. There are also $2 \log_2 mn^2 - 1 + 2\lceil \log_2 \log_2 mn \rceil$ switches in the path. The worst case attenuation is, consequently

$$A = (4 \log_2 n + 6)W + (2 \log_2 mn^2 - 1 + 2\lceil \log_2 \log_2 mn \rceil)L$$

This imposes a limitation on network size that if $m = 16$ then $n \leqslant 64$ for the best case; this is an improvement over the original network, though it is not as dramatic an improvement as for networks without frame integrity. For worst case devices, there is no value of n which will give loss of less than 25 dB.

5 Conclusions

New architectures have been proposed for optical TDM switching which have low control complexity and use feed-forward delay lines to overcome the problems associated with feed-back architectures. The price paid for the low control complexity is that more switches are required than would otherwise be necessary. The number of inputs and outputs is not restricted as in earlier architectures, the size of the network only being restricted by crosstalk and attenuation. Networks with and without frame integrity were proposed. It was found that networks with frame integrity required more switches, and had poorer attenuation and crosstalk performance.

It was shown that the worst-case crosstalk can be reduced by dilating the network; this is a generalisation of the dilated Beneš networks of Padmanabhan and Netravali [15]. Dilating the network increases the number of switches that are required. Only the *worst case* crosstalk was considered; it may be possible to improve the crosstalk in the network without dilating it fully; this is a topic for further study.

Reduction in attenuation was also considered; this involved rearranging the network so that it could fit onto fewer substrates, thus reducing the loss due to fibre/substrate and substrate/fibre couplings.

This paper has discussed the networks theoretically, based on idealised models of the switches and delay lines. While we believe we have demonstrated the validity of this approach, these networks require experimental investigation, especially concerning fibre delay line temperature drift [16], synchronisation, crosstalk and timing.

6 References

1 HUNTER, D.K., and SMITH, D.G.: 'New architectures for optical TDM switching', *J. Lightwave Technol.*, 1993, **11**, (3)

2 HUNTER, D.K., and SMITH, D.G.: 'An architecture for frame integrity TDM switching', *J. Lightwave Technol.*, 1993, **11**, (5)

3 HINTON, H.S.: 'Photonic time-division switching systems', *IEEE Circuits Devices Mag.*, July 1989, **5**, (4), pp. 39–43

4 THOMPSON, R.A.: 'Optimizing photonic variable-integer-delay circuits'. Proceedings of the first topical meeting on *Photonic switching*, Incline Village, NV, 18–20 March 1987

5 THOMPSON, R.A.: 'Architectures with improved signal-to-noise ratio in photonic systems with fibre-loop delay lines', *IEEE J. Sel. Areas Commun.*, Aug. 1988, **6**, (7), pp. 1096–1106

6 RAMANAN, S.V., JORDAN, H.F., and SAUER, J.R.: 'A new time domain, multistage permutation algorithm', *IEEE Trans. Inf. Theory*, Jan. 1990, **36**, (1), pp. 171–173

7 CANTOR, D.G.: 'On construction of nonblocking switch networks'. Symposium on Computer-Communications Networks and Teletraffic, Polytechnic Institute of Brooklyn, 4–6 April 1972

8 PIPPENGER, N.: 'The complexity theory of switching networks'. Technical Report 487, Research Laboratory of Electronics, MIT, MA, 19 Dec. 1973

9 HUI, J.Y.: 'Switching and traffic theory for integrated broadband networks' (Kluwer Academic Publishers, Boston, 1990)

10 SELVARAJAN, A., and MIDWINTER, J.E.: 'Photonic switches and switch arrays on $LiNbO_3$', *Opt. Quantum Electron.*, 1989, **21**, pp. 1–15

11 SPANKE, R.A.: 'Architectures for guided-wave optical space switching systems', *IEEE Commun. Mag.*, May 1987, **27**, (5), pp. 42–48

12 SAXTOFT, C.: 'Photonic switching networks — component characteristics versus network requirements'. PhD Thesis, Technical University of Denmark, March 1992

13 URQUHART, P.: 'Review of rare earth doped fibre lasers and amplifiers', *IEE Proc. J*, Dec. 1988, **135**, (6), pp. 385–407

14 EISENSTEIN, G.: 'Semiconductor optical amplifiers', *IEEE Circuits Devices Mag.*, July 1989, **5**, (4), pp. 25–30

15 PADMANABHAN, K., and NETRAVALI, A.N.: 'Dilated networks for photonic switching', *IEEE Trans. Commun.*, Dec. 1987, **35**, (12), pp. 1357–1365

16 SARRAZIN, D.B., JORDAN, H.F., and HEURING, V.E.: 'Fiber optic delay line memory', *Appl. Optics*, 10 Feb. 1990, **29**, (5), pp. 627–637