Architectures for optical TDM switching

D. K. Hunter, P. E. Barnsley*, I. Andonovic, B. Culshaw

Department of Electronic and Electrical Engineering, University of Strathclyde,
204 George Street, Glasgow G1 1XW, Scotland

*Optical Transport and Processing, BT Laboratories, B29 Room 138,
Martlesham Heath, Ipswich IP5 7RE, England

ABSTRACT

This paper describes some novel architectures for optical TDM switching, and an experimental system working at 720Mb/s; this includes a complete BER characterisation. The architectures are composed of 2x2 optical switches and delay lines. The trade-offs between the various factors affecting their design are explored, and a wide range of architectures are presented suitable for different applications.

1. INTRODUCTION

Multichannel time division multiplexed (TDM) optical systems will be important for future telecommunications networks for two reasons. TDM will be used for the transport of high speed continuous bitrate services such as high definition television (HDTV) and high quality video telephone, while also being used to carry asynchronous transfer mode (ATM). In addition, optical switching allows transparency to bitrate, coding format and wavelength that is not possible with electronics. Indeed, the bitrate through such a switching network is virtually unlimited. It is clear, therefore, that optical TDM switching is a highly relevant area of research.

In this paper we describe several new switch architectures for optical switching of TDM signals and present an initial experimental validation of one of the architectures. The theoretical results describe how to construct large nonblocking switching networks out of simple 2x2 switches and delay lines. Here, lithium niobate directional coupler switches and fibre delay lines are assumed since this represents the most mature technology available. Other technologies, such as silica delay lines[1], or all-fibre switches[2] could be used in the future.

The architectures are intended for use with block multiplexing[3], where the switches need only change state between each block of bits (i.e. every timeslot) rather than once every bit. Thus the electronic control bitrate can be much lower than the optical data bitrate.

The experimental system consists of three 2x2 directional coupler switches and two delay lines. It switches four 720Mb/s TDM channels. Besides examples of network operation, a complete bit error rate (BER) characterisation of the system is given.

The important aspects that characterise the performance of optical TDM switching systems are:

* *number of switches* - loosely correlated to cost,
* *attenuation* - if this is excessive, optical amplifiers must be incorporated into the network, with due regard for noise performance,
* *crosstalk performance* - dependent on both the architecture and the crosstalk performance of the individual switches,
* *control complexity* - related to the switch architecture and the number of processors used to control it; the amount of processing time available to set up each new call will depend on the application,
* *frame integrity* - a TDM switching system is said to have frame integrity if all the blocks entering on one frame leave on one frame; without frame integrity, the blocks may be spread over several output frames,
* *frame delay* - in a frame integrity system, this is the delay between an input frame entering the network and the corresponding output frame leaving the network,
* *integration onto substrates* - it is desirable to have as many switches as possible integrated onto one substrate, as this reduces the interconnection cost and also reduces the attenuation due to waveguide/fibre coupling, and
* *number of waveguide crossings* - this should be minimised, as waveguide crossings introduce unwanted attenuation and crosstalk.

All these issues are considered in the theoretical part of the paper, and the trade-offs between them are explored and discussed. For example, it is found that although the theoretical minimum number of switches may be reached, crosstalk or control complexity may be reduced at the expense of using more switches.

The architectures described here overcome the limitations of previous work, such as requiring a large number of expensive switches[4], providing a restricted number of inputs and outputs[5], having poor crosstalk performance[4,6], or having non-uniform attenuation[4,6]. Regarding the latter two, the improvement is due to the use of delays which always feed forward to the next stage of switches, rather than feeding back to the input of the same switch. This ensures that the attenuation is uniform, and also that blocks do not become attenuated down toward the crosstalk signal level.

The control of these networks can be accomplished by adapting standard control algorithms that were designed for Benes and Cantor networks[7,8,9]; this is discussed in section 5.

2. TYPE 1 ARCHITECTURES - WITH AND WITHOUT FRAME INTEGRITY

The switch architectures in this paper are called type 1, type 2 and type 3 networks, where the architectures of each type have certain common characteristics. In this section, type 1 architectures are described; they can use the theoretical minimum number of switches, but require a comparitively slow control algorithm.

Type 1 architectures are built up from Baseline networks[10], reverse Baseline networks[10] and Waksman networks[11], each of which may be fabricated on one substrate, giving the benefits discussed above. Baseline networks were chosen, rather than any of their equivalents[10], because their interconnection pattern implies the minimum number of waveguide crossovers, and the minimum worst-case number of crossings in a signal path[12]. This is desirable because waveguide crossovers introduce unwanted attenuation and crosstalk into the network. When synthesising Waksman networks, a similar interconnection pattern is chosen for the same reason. Let *B(m)*, *R(m)* and *W(m)* denote *mxm* Baseline, reverse Baseline and Waksman networks respectively. As is well known[10,11], these networks may be defined recursively as follows. The smallest possible networks, *B(2)*, *R(2)* and *W(2)* are simply single 2x2 switches. Larger networks may be constructed from smaller ones as shown in figs. 1, 2, and 3; this allows any size of network to be constructed, only limited by the size of the substrate.

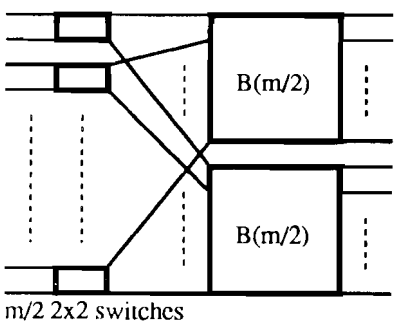


Fig. 1: Recursive definition of a Baseline network B(m)

The notation *T(m,n)* represents a type 1 switching network which can switch TDM data and is made up of the networks discussed in the previous paragraph. *T(m,n)* has *m* input and output links, and handles *n* timeslots per frame. Thus it can switch a total of *mn* TDM channels, and any of *(mn)!* mappings are possible from the input channels to the output channels. When creating a new type 1 network, one starts with *T(m,1)* - this is simply *W(m)* since if there is only one timeslot per frame, no interchange of blocks between timeslots

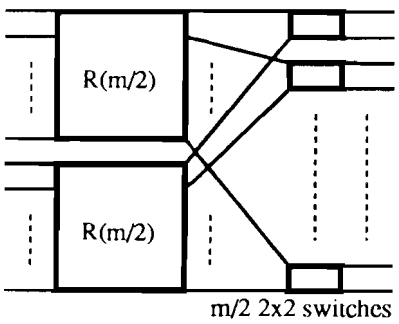


Fig. 2: Recursive definition of a reverse Baseline network R(m)

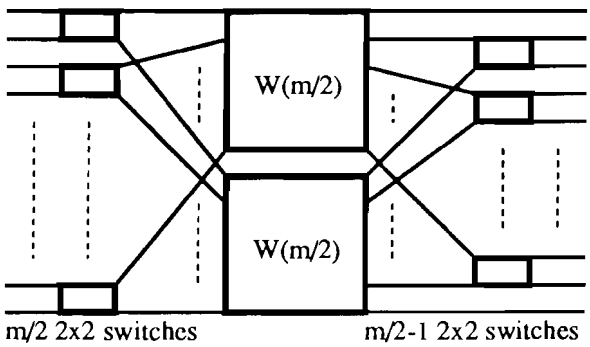


Fig. 3: Recursive definition of a Waksman network W(m)

is possible, and the network is equivalent to a time multiplexed space switch. Fig. 4 shows how to create a larger network *T(em, fn)* from smaller networks *T(m,n)* plus additional Baseline and reverse Baseline networks. To ensure correct operation, the connections should be made in exactly the order shown. The networks in the centre stage must have any delay lines in them multiplied in length by a factor of *f*. The meaning of the thick dashed lines will be discussed shortly. *e* may be thought of as the "space expansion factor", since the network *T(em,fn)* has *e* times as many inputs and outputs as the centre stage networks *T(m,n)*. Likewise, *f* may be thought of as the "time expansion factor". *e* and *f* must both be integral powers of 2.

There are two varieties of type 1 architecture - type 1A which does not have frame integrity and type 1B which does. To specify the delay line lengths correctly, it is necessary to use *r(i)*, the bit-reversal function, which operates on $\log_2 f$-digit binary numbers. That is, if the binary representation of i is $b_{k-1}b_{k-2}\ldots b_1 b_0$ where $k = \log_2 f$ then $r(b_{k-1}b_{k-2}\ldots b_1 b_0) = b_0 b_1 \ldots b_{k-2} b_{k-1}$. When creating a type 1A architecture (without frame integrity), each thick dashed line in fig. 4 represents a bundle of *f* connections, which have delays in them as shown in fig. 5. With frame integrity - i.e. a type 1B architecture - the substitution of fig. 6 must be used instead. *P(f)* is Lawrie's Omega network[13], with *f* inputs and *f* outputs, whose function is to carry out rotation operations - this is discussed in Appendix A. In practice, because a Baseline network has fewer waveguide crossings[12], *P(f)* would be replaced by a Baseline network[10], with the input connections suitably rearranged[10].

Sometimes it is necessary to have switch networks with a single input and output i.e. timeslot interchangers (TSIs). Using our notation, this would be expressed as *T(1,n)*. One way of creating such a network would be to take some network *T(2,n)* and simply leave one input and one output unused. However, a more economical method is shown in fig. 7 without frame integrity (type 1A) and in fig. 8 with frame integrity (type 1B).

Fig. 9 is an example of a type 1A architecture. It has 4 inputs and outputs, handles 16 timeslots per frame, and does not have frame integrity. Here *e*=1 in fig. 4. An equivalent type 1B architecture would have 9 substrates instead of 5 (each containing 4 switches except for the central one), due to the additional Omega (Baseline) networks required to align the frames, with a consequent increase in attenuation. Increasing *m* generally decreases the number of substrates for any given *n*; this is because

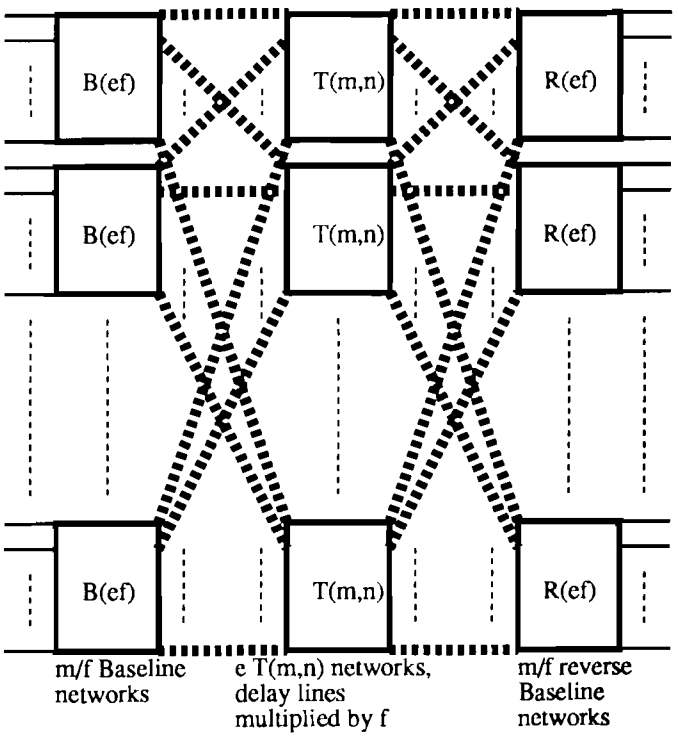


Fig. 4: Recursive definition of T(em,fn) with em inputs and outputs

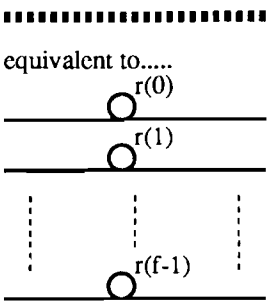


Fig. 5: Definition of interconnect without frame integrity

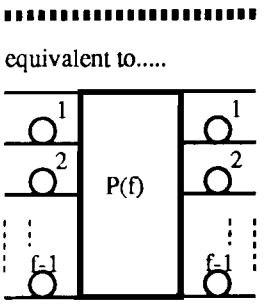


Fig. 6: Definition of interconnect with frame integrity

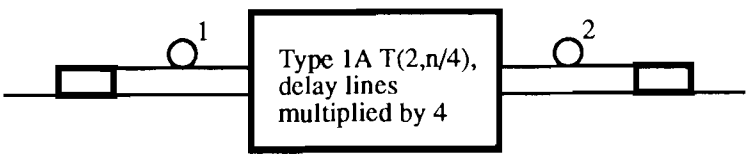


Fig. 7: A type 1A T(1,n) without frame integrity

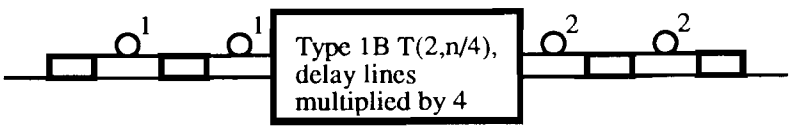


Fig 8: A type 1B T(1,n) with frame integrity

the maximum value of f in fig. 4 is equal to m. For example, a type 1A $T(16,16)$ only needs three substrates; the outer two having 32 switches instead of 4 as before. Because the number of crossovers in a path is variable on each substrate[12], it may be necessary to include dummy crossovers to provide more uniform attenuation.

The performance of type 1 networks will be discussed in section 7.

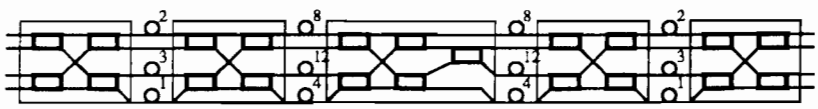


Fig. 9: A type 1A T(4,16)

Desired network	Restrictions	$A(m,n/2)$ replaced by:
$X(m,n)$	m=2 only	$X(m,n/2)$
$Y(m,n)$	m=2 only	$X(m,n/2)$
$T(m,n)$	none	$Y(m,n/2)$ or $X(m,n/2)$

Table 1: Allowable conditions for fig. 13

3. TYPE 2 ARCHITECTURES - WITH FRAME INTEGRITY

The inclusion of Omega networks which re-align frame boundaries in type 1B networks is one approach to providing frame integrity. Here, we discuss a second approach which has certain advantages and disadvantages which will be discussed more fully in section 7. For now, it is sufficient to say that although type 2 architectures use fewer switches than type 1B, they do not split up into substrates containing large numbers of switches, therefore the interconnection losses are greater. The control complexity for both architectures is the same.

Desired network	Restrictions	$B(m/2,n)$ replaced by:
$X(m,n)$	none	$X(m/2,n)$
$Y(m,n)$	none	$Y(m/2,n)$ or $X(m/2,n)$
$T(m,n)$	none	$T(m/2,n)$ or $X(m/2,n)$

Table 2: Allowable conditions for fig. 14

Desired n	Starting point
2	fig. 10
4	fig. 11
n >= 8	fig. 12

Table 3: Starting points for providing type 2 networks, depending on desired n

As intermediate steps in creating the desired type 2 network $T(m,n)$, two other types of network are produced - $X(m,n)$ and $Y(m,n)$. While a detailed discussion of these networks can be found elsewhere[14], they are essentially the same as the T-networks already considered (i.e. networks of the type $T(m,n)$ - this abbreviation will be used throughout). The only difference is that the frame boundaries are not necessarily aligned on the inputs and outputs. This misalignment is crucial to the operation of the network and its economical use of hardware.

As before, the principle is to build up larger networks from smaller ones until the desired size of network is produced. To start, one of the networks shown in figs. 10-12 is used. Observe that $Y(2,1)$ and $X(2,1)$ use more than the single switch required for $T(2,1)$; this is because the extra hardware is used to re-align frame boundaries.

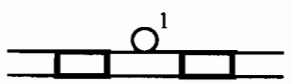


Fig. 10: Definition of Y(2,1)

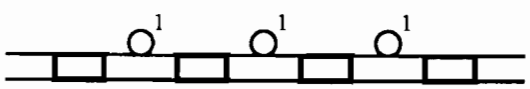


Fig. 11: Definition of Y(2,2)

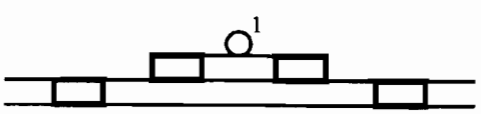


Fig. 12: Definition of X(2,1)

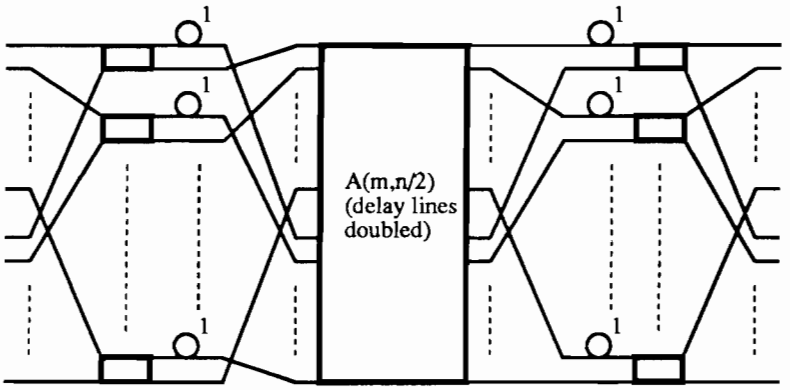


Fig. 13: Producing a larger network handling twice as many timeslots per frame from a smaller one

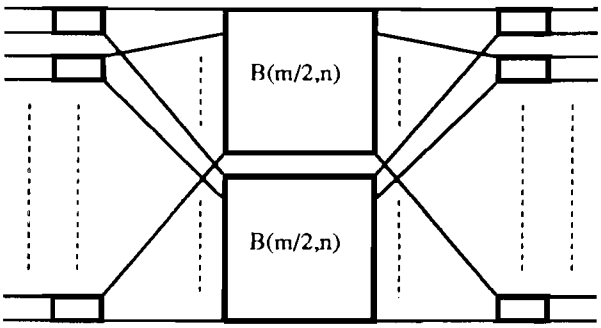


Fig. 14: Producing a larger network handling twice as many inputs and outputs from a smaller one

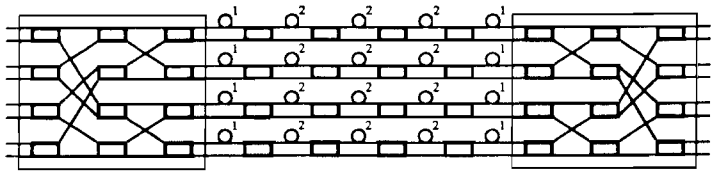


Fig. 15: A type 2 T(8,4)

Producing larger networks from smaller ones is more complex than with type 1 architectures. Larger architectures may be produced from smaller ones in either of two ways (figs. 13 & 14). Fig. 13 creates a larger network that handles twice as many timeslots per frame, while in fig. 14, the new network has twice as many inputs and outputs. In both cases, the new network has m inputs and outputs, and handles n timeslots per frame. There are certain restrictions on what networks may be substituted for $A(m,n/2)$ and $B(m/2,n)$ in the diagrams- for a full discussion see reference 14. Tables 1 and 2 define $A(m,n/2)$ and $B(m/2,n)$ by showing what restrictions exist for figs. 13 and 14 respectively. The choice of small network from figs. 10-12 that is used when starting to create a new type 2 network can be shown, as a consequence, to depend on the size of frame ultimately desired in the finished network (table 3).

It can be seen from table 1 that when creating an X-network or a Y-network, m is restricted in size to 2. It can be shown that as a consequence of this, when the finished type 2 network has been created, structures usually result which consist of several parallel strings of single switches and delay lines with substrates containing many switches on either side. This is illustrated by the example of fig. 15 (type 2 $T(8,4)$); here, the central strings of switches, each contain five delay lines, consisting of a $Y(2,2)$ network (fig. 11) with all its delay lines doubled in length, plus delay lines of length 1 timeslot on either side which were introduced in fig. 13.

4. TYPE 3 ARCHITECTURES - REDUCED CONTROL COMPLEXITY WITH AND WITHOUT FRAME INTEGRITY

Type 1 and type 2 architectures both require the same control algorithm as Benes networks[7,8]. Although this will be discussed in more detail in the next section, a problem with this control algorithm in some applications is that it requires a comparitively long time to run. Type 3 networks circumvent this problem by using a much faster control algorithm - the same algorithm that is used to control Cantor networks[9,15,16]. The price payed for this reduction in control complexity is that more switches are required to build the network.

Like type 2 networks, an intermediate network is involved in producing the desired network $T(m,n)$. It is called $I(m,n)$. Let $M=\log_2 mn$, where m and n represent the desired size of the finished T-network. When creating a new I-network, one starts with a network $I(1,1)$ which is a $1xM$ demultiplexer; fig. 16 can then be used to create larger I-networks from smaller ones. The thick dotted lines can be substituted as in section 2 to produce architectures with or without frame integrity - types 3B or 3A respectively. Mm must be a multiple of f, and it may be necessary to set f to, say, 1 initially in order to fulfil this condition. Alternatively, M could be rounded up to the nearest power of 2; when calculating attenuation in section 7, it will be assumed that this has been done, as this yields the lowest attenuation.

Once fig. 16 has been used to produce a network $I(m,n)$ of the desired size, $T(m,n)$ can be synthesised in the following way. First produce a new network called $J(m,n)$ by taking $I(m,n)$ and reflecting it about its vertical axis; when doing this, a demultiplexer becomes a multiplexer and a R-network becomes a B-network. Omega networks remain unchanged when the network is reflected. These Omega networks and R-networks will have been introduced when using fig. 16 to create the desired size of I-network. $I(m,n)$ and $J(m,n)$ are then joined to produce $T(m,n)$ (fig. 17). The number of switches can be reduced by carrying out a simple optimisation. Every output stage switch in $I(m,n)$ will have both its outputs connected to the same input stage switch in $J(m,n)$. Each such pair of switches may be joined together to form a single switch.

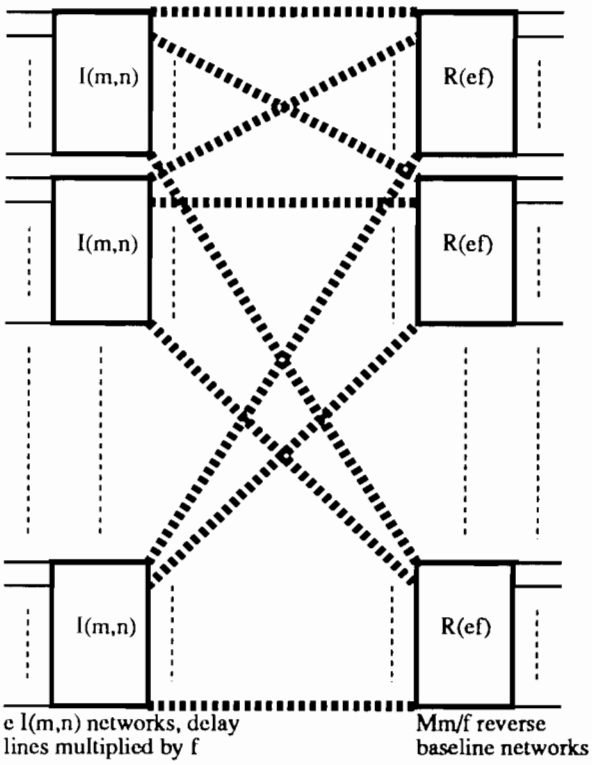


Fig. 16: Recursive definition of I(em,fn) with em inputs and eMm outputs

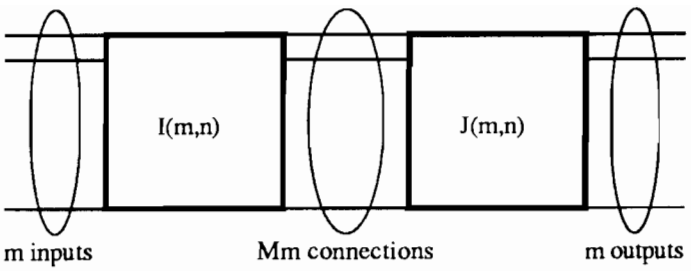


Fig. 17: Producing T(m,n) from I(m,n) and J(m,n)

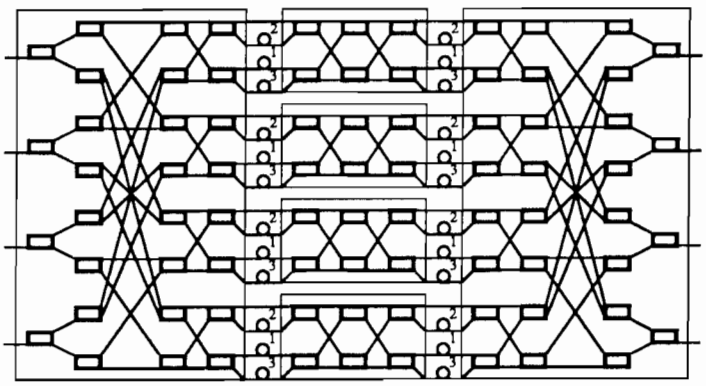


Fig. 18: A type 3A T(4,4)

Fig. 18 is an example of a type 3A *T(4,4)* network. It uses 80 switches, whereas the corresponding type 1A network uses only 13. As discussed above, the benefit obtained from the increased use of switches is the reduction in control complexity. The demultiplexers and multiplexers are made up of groups of three switches. 4x4 Benes networks have been formed in the centre stage by combining Baseline and reverse Baseline networks (fig. 17).

5. CONTROLLING THE NETWORKS

To implement one of the networks described above, it is necesarry to have some way of translating the desired connections between input and ouput channels into control signals for the switches; the latter may change state every timeslot. Fortunately, existing control algorithms may be adapted to carry out this task. A standard Benes network control algorithm is used for type 1 and type 2 architectures. This has a complexity of $O(mn\log mn)$ if one processor is used[7], or $O((\log mn)^2)$ for mn processors[8]. For type 3B networks, the Cantor network control algorithm could be used to set up new calls without re-routing those in progress[9]; the complexity is $O((\log mn)^2)$ for a single processor. Depth-first-search circuit hunting[16] or a similar algorithm would be used with type 3A networks.

The control algorithm computes the switch settings for a hypothetical space switch i.e. a switch composed solely of switches, with no delay lines. The space switch can be obtained from the real switch network by using a space-time mapping[14,17], and has mn inputs and outputs. Type 1 and 2 architectures map onto Benes networks[18,19] and Waksman networks[11]; type 3 architectures map onto Cantor networks[15]. The switch settings for the space switch are then fed in parallel into an array of shift registers which transmit them in series to the real switch fabric.

6. DILATION

Dilation improves the crosstalk performance of a switching network while roughly doubling the number of switches required. Although it was originally proposed for Benes and Omega networks[20], the concept will be generalised to be compatible with any of the switching networks discussed in this paper.

The dilation process involves taking a network and modifying it so that at most one block can travel through a switch at once. Clearly, this will drastically improve the crosstalk performance by eliminating "first order" crosstalk terms. (These are terms in the expression for crosstalk due to the extinction ratio rather than its higher powers.) Fig. 19 shows the first step in the procedure - each switch which has both inputs and outputs used is replaced by the four-switch structure shown. The condition now holds that two blocks cannot pass through a switch simultaneously. Pairs of "back to back" switches may now be joined (fig. 20) while still preserving this condition; if there is a delay line between them, it must be shifted to one side of the replacement switch.

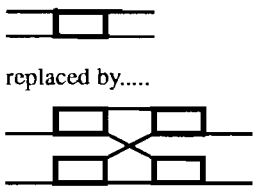


Fig. 19: First step in dilating a network

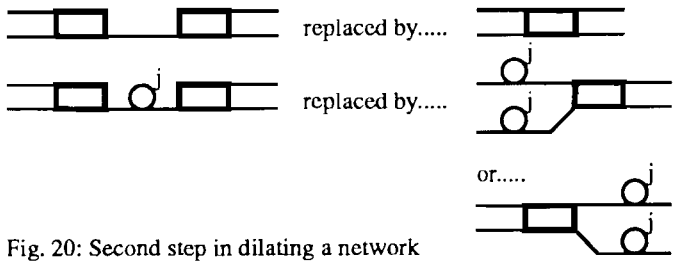


Fig. 20: Second step in dilating a network

The effect of dilation on crosstalk performance is discussed in the next section.

7. SWITCH NETWORK PERFORMANCE

Formulae characterising the performance of each network architecture are given in table 4. (SXR stands for signal to crosstalk ratio.) The variables are all in dB:

X = crosstalk in a single switch element (a negative number)
L = loss of a switch element (excluding fibre to substrate coupling loss)
W = fibre to substrate coupling loss

To ensure mathematical tractability, crosstalk was assumed to add to the signal linearly, and waveguide bends and crossovers were ignored in the calculations. The function "ceil" rounds up to the nearest integer.

	Type 1A	Type 1B	Type 2	Type 3A	Type 3B
Number of switches	$m\log_2 mn - m + 1$	$m\log_2 mn^2 - m + 1$	$m\log_2 m + m$ $(n=2)$ $m\log_2 m + 2m$ $(n=4)$ $m\log_2 mn + m$ $(n>=8)$	$Mm\log_2 mn + 3Mm/2 - 2m$	$Mm\log_2 mn^2 + 3Mm/2 - 2m$
Attenuation	$(4\text{ceil}(\log_2 n/\log_2 m)+2)W + (2\log_2 mn - 1)L$	$(8\text{ceil}(\log_2 n/\log_2 m)+2)W + (2\log_2 mn^2 - 1)L$	$(2\log_2 m+2)L+8W$ $(n=2)$ $(2\log_2 m+4)L + 12W$ $(n=4)$ $(2\log_2 mn+2)L + (4\log_2 n+4)W$ $(n>=8)$	$(4\text{ceil}(\log_2 n/\text{ceil}(\log_2 Mm))+2)W + (2\text{ceil}(\log_2 Mmn) - 1)L$	$(8\text{ceil}(\log_2 m/\text{ceil}(\log_2 Mm))+2)W + (2\text{ceil}(\log_2 Mmn^2) - 1)L$
SXR	$-X - 10\log_{10}(2\log_2 mn - 1)$	$-X - 10\log_{10}(2\log_2 mn^2 - 1)$	$-X - 10\log_{10}(2\log_2 mn)$	$-X - 10\log_{10}(2\log_2 mn - 1)$	$-X - 10\log_{10}(2\log_2 mn^2 - 1)$
Control complexity	$O(mn\log mn)$ one processor $O((\log mn)^2)$ mn processors			-	$O((\log mn)^2)$ one processor
Frame integrity?	no	yes	yes	no	yes
Frame delay	-	$2n-2$	$4(n-1)/3$ $(n=4^i)$ $2(2n-1)/3$ otherwise	-	$2n-2$

Table 4: Performance of network architectures

	Type 1A	Type 1B	Type 2	Type 3A	Type 3B
Atten., m=2	n=<512/4	n=<16/2	n=<256/2	n=<16777216/8	n=<1024/-
Atten., m=16	n<=1048576/8	n=<2048/-	n=<256/-	n=<1.34x10^8/-	n=<4096/-
Crosstalk	mn=<8.51x10^{37}/16	mn^2=<8.51x10^{37}/16	mn=<4.25x10^{37}/8	mn=<8.51x10^{37}/16	mn^2=<8.51x10^{37}/16

Table 5: Limits on network size due to attenuation and crosstalk: best case/worst case

Table 5 shows the limits on all the architectures due to crosstalk and attenuation. The figures were calculated for both "best case" switches, with good performance[21] (X=-35dB, L=0.25dB, W=0.5dB) and "worst case" switches with relatively poor performance[22] (X=-20dB, L=1dB, W=2dB). Attenuation of less than 25dB, and a signal-to-crosstalk ratio (SXR) of over 11dB are assumed throughout. A signal-to-noise ratio of 11dB implies a bit error rate of 10^{-9}, but the system is more tolerant to crosstalk than noise since, unlike noise, crosstalk has an upper bound. The crosstalk calculations for type 3 architectures yield the most pessimistic possible figures since it is assumed that a block will meet another block in every switch it passes through. A dash in the table indicates that no architecture is possible for the given conditions.

The effect of dilation is to double the SXR and add 3dB[14]. For example, $- X - 10\log_{10}(2\log_2 mn - 1)$ for type 1A becomes $- 2X - 20\log_{10}(2\log_2 mn - 1) + 3$ with dilation.

Several interesting points emerge from the tables:

* although it is possible to attain optimum use of hardware (type 1A), component count may be traded off to improve control complexity (type 3) or crosstalk performance (dilation),
* frame integrity networks use more switches than their non frame integrity counterparts, and have poorer attenuation and crosstalk performance,
* with type 1 and 3 architectures and best-case switch elements, increasing m increases the maximum n that is allowed; this is because the substrate size increases with m,
* besides using fewer switches than type 1B, type 2 architectures have better crosstalk performance, and
* type 3 architectures have identical crosstalk performance to type 1; this is because each switch in the multiplexers and demultiplexers only carry one block at once and hence can be ignored.

8. EXPERIMENT

The experimental work involved building a $T(1,4)$ type 1A architecture - see fig. 21. All components were connectorised so that the experiment can be extended to cope with larger architectures in the future. The fibres had to be cut to within 1cm to ensure accurate synchronisation of the system. The precision required depends on the block rate. The loss of each directional coupler switch was typically 6-7dB, with an extinction ratio of 15dB. The spare output on the last switch was used as a "drop" output, so that a block could either be sent to the main upper output or "dropped" to the lower output.

Three erbium doped fibre amplifiers (EDFAs) were required to overcome the switch loss; one on the input to the switch fabric, and two on the outputs (see fig. 21). When carrying out BER measurements, the output amplifiers were not used. Since the switches were polarisation sensitive, polarisation controllers were used on all switch inputs. The 720Mb/s pattern generator produced a repeating sequence of four 16-bit blocks which were fed via a DFB laser into the switch network. The 45Mb/s pattern generator, running at the block rate, generated the control signals for the switches which changed state only between blocks i.e. every 16 bits. A guard band of 2.75ns (2 bits) was included between blocks to allow the switches time to change state. Rise and fall times of 2 to 2.5ns were measured for the switches, this being limited by the pattern generator; the switches had a bandwidth of 4GHz. A microcomputer ran a control algorithm which calculated the switch control signals from the desired mapping of input channels onto output channels.

Fig. 22 shows the output pulse trains for two configurations of the switch (the timeslots containing the 16-bit blocks are numbered form 0 to 3). The arrows indicate the mapping of the input line onto the main and "drop" outputs. In each case, timeslot 3 of the output frame is sent to the lower "drop" channel. Consider the left-hand configuration:

* the block arriving on timeslot 0 leaves on timeslot 2,
* the block arriving on timeslot 1 leaves on timeslot 1, too,

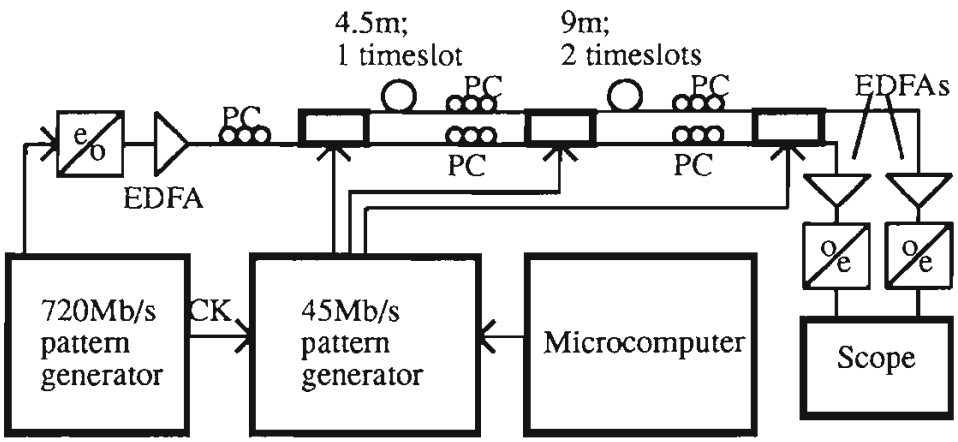
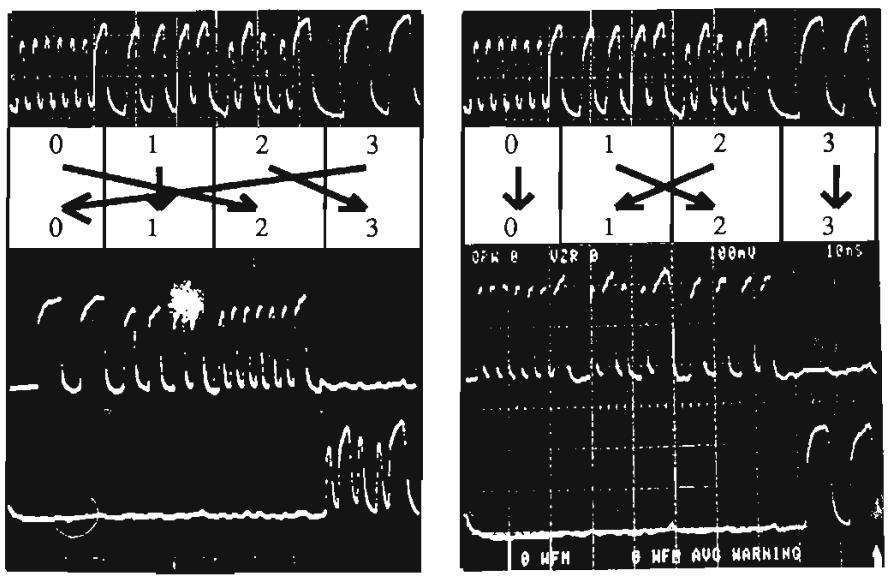


Fig. 21: Experimental arrangement

* the block arriving on timeslot 2 leaves on timeslot 3, but on the "drop" channel, and
* the block arriving on timeslot 3 leaves on timeslot 0.

Bit error ratio (BER) measurements for this system are shown in fig. 23. The optical preamplifiers were removed for these measurements, which were made on the dropped channel, and a variable attenuator was inserted between the final switch and the receiver. A non-optimal optical receiver (approximately -30dBm at 622Mb/s) was used. In fig. 23, "through switch, best case" indicates that no other block travelled through the centre switch when the dropped block did, while "through switch, worst case" means that there was another block there to provide crosstalk. What is perhaps surprising is that best case crosstalk gives a better result than "no switches, 1 timeslot full", where the source was connected directly to the detector (i.e. a "back-to-back" measurement), and one timeslot was full in each frame to mimic the output from the dropped channel. This is presumably because the switch network reduces the amplified spontaneous emission (ASE) produced by the transmitter EDFA, replacing it by crosstalk in the empty timeslots. Since, unlike noise, crosstalk has an upper bound, an improvement in performance results. When a back-to-back measurement was done without the switch fabric, and with four full timeslots per frame ("no switches, 4 timeslots full"), the difference from the 1-timeslot case was 5.2dB, not 6dB as expected; this was due to the contrast ratio of the optical data being the same for both cases.



0 -> 2
1 -> 1
2 -> 3 (drop)
3 -> 0

0 -> 0
1 -> 2
2 -> 1
3 -> 3 (drop)

Fig. 22: Two trial runs of the system

9. CONCLUSIONS

We have proposed a new class of optical TDM switching networks which have a wide variety of characteristics, making them suitable for many different applications. By constructing a 720Mb/s timeslot interchanger with added "drop" function, we have shown that this approach is completely valid. We plan to look at more complex configurations; also, issues such as the removal of guard bands, synchronisation within a system and equalisation of attenuation in large architectures will be investigated in future work.

Networks both with and without frame integrity were proposed; without frame integrity they can use the minimum possible number of switches, and the number required for frame integrity is only slightly more. All of the networks may be dilated, reducing crosstalk at the expense of using more switches. The networks are designed so that they can be constructed out of large substrates, making large networks possible with low attenuation and low interconnection cost. The control of these networks can be accomplished by adapting standard control algorithms that were designed for circuit switched networks. Some of the architectures feature low control complexity, making them suitable for applications which require fast connection of new calls.

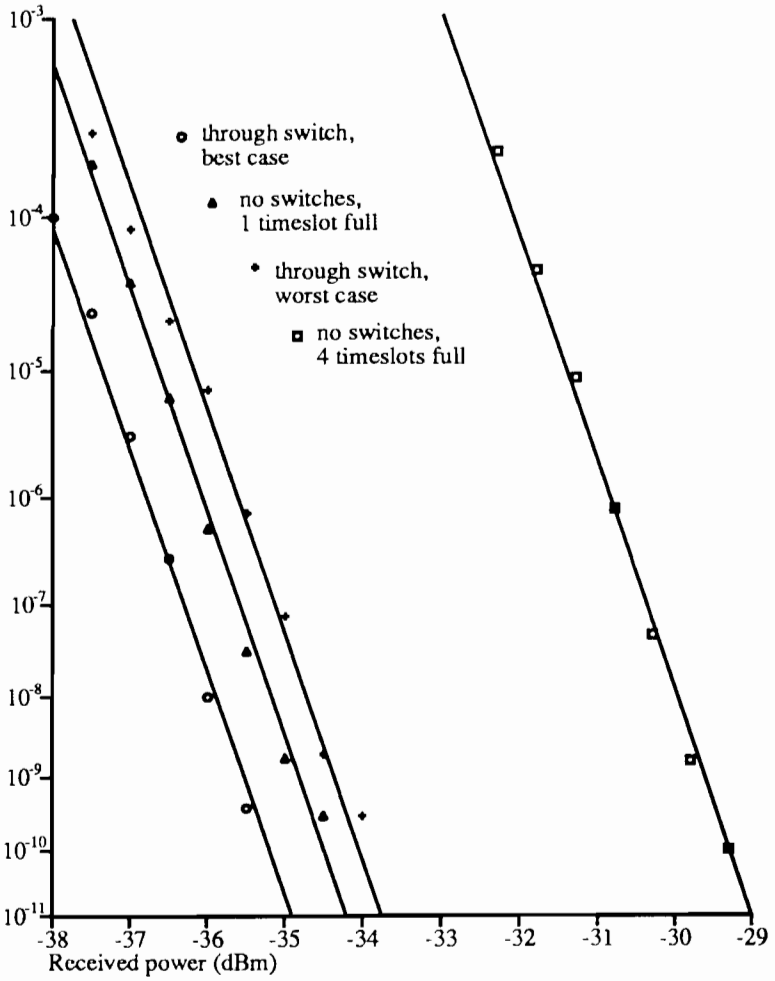


Fig. 23: Bit error ratio (BER) measurements for the experimental system

Many existing optical TSI designs[4] are based on electronic architectures; the number of switches is linearly related to the number of timeslots per frame. The architectures in this paper exhibit a logarithmic relationship, since they exploit the characteristics of delay lines, and allow economical use to be made of the hardware.

10. ACKNOWLEDGEMENTS

The authors wish to thank A. McGuire and A. Thomas for supplying the devices. We also wish to thank D. G. Smith and M. Shabeer for many useful discussions.

APPENDIX A: SETTING OF OMEGA NETWORKS IN TYPE 1B AND 3B ARCHITECTURES

Consider one of the stages of Omega networks in the frame integrity version of fig.4 or fig. 16. Define the variable t (=0,...fn-1) as being zero on the first timeslot that blocks from a new frame pass through the networks. On all other timeslots, t is one greater than on the previous timeslot. i and j denote the input and output terminals on the network. They are numbered from 0 to f-1, starting at the top. The mappings carried out by the networks - which change every timeslot - can now be defined by $j=i+fn-1-t \bmod f$. On any timeslot t, input terminal i on each network is connected to output terminal j.

When the T-networks in fig. 4 or the I-networks in fig. 16 are modified by multiplying their delay lines by f, the old frame of size n becomes a new frame of size fn. Let t_1 $(=0,...,n-1)$ and t_2 $(=0,...fn-1)$ be defined in a similar way to t so that they represent the timeslot numbers in the original network and the modified network respectively. Then on timeslot t_2 in the modified network, the networks are set to the same state as they were at timeslot $t_1=\text{floor}(t_2/f)$ originally. The function "floor" rounds down to the nearest integer.

REFERENCES

1. C. J. Beaumont, S. A. Cassidy, D. Welbourn, M. Nield, A. Thurlow, D. M. Spirit, "Integrated Silica Optical Delay Line for Signal Processing", *British Telecom Technology Journal*, vol. 9, no. 4, pp30-35, October 1991

2. K. McCallion, W. Johnstone, G. Thursby, "Investigation of Optical Fibre Switch Using Electro-Optic Overlays", *Electronics Letters*, vol. 28, no. 4, pp410-411, 13 February 1992

3. H. S. Hinton, "Photonic Time-Division Switching Systems", *IEEE Circuits and Devices Magazine*, vol. 5, no. 4, pp39-43, July 1989

4. R. A. Thompson, "Architectures with Improved Signal-to-Noise Ratio in Photonic Systems with Fiber-Loop Delay Lines", *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 7, pp1096-1106, August 1988

5. S. Andresen, S. R. Harrison, "Toward a General Class of Time-Division-Multiplexed Connecting Networks", *IEEE Transactions on Communications*, vol. 20, no. 5, pp836-846, October 1972

6. S. V. Ramanan, H. F. Jordan, J. R. Sauer, "A New Time Domain, Multistage Permutation Algorithm", *IEEE Transactions on Information Theory*, vol. 36, no. 1, pp171-173, January 1990

7. D. C. Opferman, N. T. Tsao-Wu, "On a Class of Rearrangeable Switching Networks Part I: Control Algorithm", *Bell System Technical Journal*, vol. 50, no. 5, pp1579-1600, May-June 1971

8. G. F. Lev, N. Pippenger, L. G. Valiant: "A Fast Parallel Algorithm for Routing in Permutation Networks", *IEEE Transactions on Computers*, vol. 30, no. 2, pp93-100, February 1981

9. N. Pippenger, "The Complexity Theory of Switching Networks", *Massachusetts Institute of Technology Research Laboratory of Electronics Technical Report 487*, 19 December 1973

10. C.-L. Wu, T.-Y. Feng, "On a Class of Multistage Interconnection Networks", *IEEE Transactions on Computers*, vol. 29, no. 8, pp694-702, August 1980

11. A. Waksman, "A Permutation Network", *Journal of the Association for Computing Machinery*, vol. 15, no. 1, pp159-163, January 1968

12. C.-T. Lea, "Crossover Minimization in Directional-Coupler-Based Photonic Switching Systems", *IEEE Transactions on Communications*, vol. 36, no. 3, March 1988

13. D. H. Lawrie, "Access and Alignment of Data in an Array Processor", *IEEE Transactions on Computers*, vol. 24, no. 12, pp1145-1155, December 1975

14. D. K. Hunter, "Optical Switching in Ultrafast Communications Networks", *University of Strathclyde Ph.D. Thesis*, Glasgow, September 1991

15. D. G. Cantor, "On Construction of Nonblocking Switch Networks", *Symposium on Computer-Communications Networks and Teletraffic*, Polytechnic Institute of Brooklyn, 4-6 April 1972

16. J. Y. Hui, *Switching and Traffic Theory for Integrated Broadband Networks*, Kluwer Academic Publishers, 1990

17. M. J. Marcus, "Space-Time Equivalents in Connecting Networks", *Proc. Int. Conf. Communications*, pp 35.25-35.31, 1970

18. V. E. Benes, *Mathematical Theory of Connecting Networks and Telephone Traffic*, Academic Press, 1965

19. F. K. Hwang, "On Benes Rearrangeable Networks", *Bell System Technical Journal*, vol. 50, no. 1, pp201-207, January 1971

20. K. Padmanabhan, A. N. Netravali: "Dilated Networks for Photonic Switching", *IEEE Transactions on Communications*, vol. 35, no. 12, pp1357-1365, December 1987

21. A. Selvarajan, J. E. Midwinter, "Photonic Switches and Switch Arrays on $LiNbO_3$", *Optical and Quantum Electronics*, vol. 21, no. 1, pp1-15, 1989

22. R. A. Spanke, "Architectures for Guided Wave Optical Space Switching Systems", *IEEE Communications Magazine*, vol. 27, no. 5, pp42-48, May 1987