SLOB: A Switch with Large Optical Buffers for Packet Switching

David K. Hunter, *Member, IEEE,* W. David Cornwell, Tim H. Gilfedder,
André Franzen, Ivan Andonovic, *Senior Member, IEEE, Member, OSA*

***Abstract*—Recently, optical packet switch architectures, composed of devices such as optical switches, fiber delay lines, and passive couplers, have been proposed to overcome the electromagnetic interference (EMI), pinout and interconnection problems that would be encountered in future large electronic switch cores. However, attaining the buffer size (buffer depth) in optical packet switches required in practice is a major problem; in this paper, a new solution is presented. An architectural concept is discussed and justified mathematically that relies on cascading many small switches to form a bigger switch with a larger buffer depth. The number of cascaded switches is proportional to the *logarithm* of the buffer depth, providing an economical and feasible hardware solution. Packet loss performance, control and buffer dimensioning are considered. The optical performance is also modeled, demonstrating the feasibility of buffer depths of several thousand, as required for bursty traffic.**

***Index Terms*—Buffer memories, communication switching, optical fiber communication, optical fiber delay lines, modeling, packet switching, photonic switching systems.**

I. Introduction

OPTICAL packet switches, in which packets are switched and buffered optically [1]–[9], have been proposed worldwide as a way of overcoming the envisaged future limitations of electronic packet switching technology. These limitations include susceptibility to electromagnetic interference (EMI) between interconnections, and other problems involved in interconnecting a large switch core, particularly with respect to providing sufficient pinout on each chip. There are two principal functions performed by such a packet switch; firstly, each packet must be directed to the correct output. Second, since packets arrive asynchronously, two or more may wish to go to a particular output at once. To overcome this, buffers are necessary to resolve such packet contention. This paper proposes a novel solution to a major problem in designing such optical packet switches, namely providing sufficiently large buffers, in order to resolve packet contention effectively without undue buffer overflow.

For example, in an output buffered packet switch [5]–[12] with 16 inputs and outputs subject to a uniform, nonbursty load of 0.8, attaining a packet loss of 10^{-11} requires a buffer depth of 55 packets [13], [14]; this doubles for a load of 0.9. For this type of traffic, the required buffer depth depends on the load, desired packet loss and number of inputs and outputs. For bursty traffic, which is more realistic, the buffer depth must be increased significantly, with figures of several thousand packets being quoted [10]. Current optical packet switch demonstrators exhibit much lower buffer depths (e.g., 23 packets [5]) limited either by the low number of possible recirculations in a loop [5], or by the excessive hardware required for larger buffer depths [8], [9], [12]. To resolve this mismatch, the switch with large optical buffers (SLOB) is introduced in this paper; it cascades many small switches, forming a larger switch with a greater buffer depth. The OASIS switch [8] is chosen as the basic element, although other switches could be used.

The packet loss performance of output-buffered switches was introduced above; how SLOB emulates one is described in Section II. The architectural concepts are described in Section III, and control is covered in Section IV. Section V considers the optical performance of the architecture, showing that buffer depths of many thousands are possible with present-day components, and the conclusions are drawn in Section VI. Detailed proofs are given in the Appendixes. The SLOB architecture may be regarded as a generalization of the 2 × 2 architecture described in [25]; a discussion of the relationship between them can be found in Appendix B.

II. Output-Buffered Switch Emulation

SLOB has m inputs and m outputs, and can delay each incoming packet by between zero and q timeslots. (A timeslot is an interval that may hold one packet—there is no frame structure since SLOB is a slotted packet switch.) It can direct each incoming packet to any output, subject to at most one packet leaving each output per timeslot. Its packet loss and delay performance are identical to the output-buffered switch it emulates; both exhibit optimal delay/throughput performance. Fig. 1 shows an output-buffered packet switch.

When emulating an output-buffered switch, m counters, one associated with each SLOB output, calculate the packet delays (Fig. 2). Each counter holds the number of packets in an imaginary SLOB FIFO (first in first out) output buffer, and

Manuscript received March 17, 1997; revised April 21, 1998 and July 6, 1998. This work was supported in part by the ACTS KEOPS project. The work of D. Hunter was supported by the EPSRC in the form of an Advanced Fellowship.

D. K. Hunter, A. Franzen, and I. Andonovic are with the Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow G1 1XW U.K.

W. D. Cornwell was with the Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow G1 1XW U.K. He is now with BICC Cables Ltd., Technology Centre, Helsby, Warrington, Cheshire WA6 0DJ U.K.

T. H. Gilfedder was with the Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow G1 1XW U.K. He is now with BT Labs, Martlesham Heath, Ipswich IP5 3RE U.K.

Publisher Item Identifier S 0733-8724(98)07414-3.

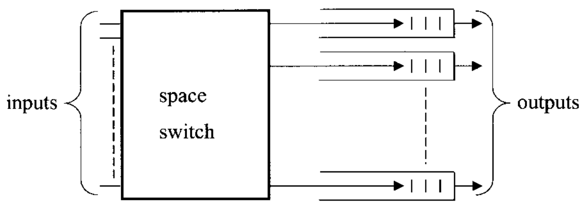

Fig. 1. An output-buffered packet switch; several packets, coming from different inputs, may be placed in an output buffer during the same timeslot.

during each timeslot is decremented by one when a packet leaves its output, then incremented by one for each arriving packet destined for its output. When packets arrive at an empty imaginary SLOB buffer (i.e., the counter is zero), one packet passes straight through, experiencing zero delay, while the counter is incremented once for each other arriving packet. No packet leaves an output on a particular timeslot if the corresponding counter is already zero (when the imaginary SLOB buffer is empty) and no packets arrive. If multiple packets arrive for an output on the same timeslot, they are assigned successive values of delay, corresponding to them being put in the imaginary SLOB output buffer one after another. The total SLOB delay (in timeslots) that a packet experiences is the value of the designated output counter upon its arrival. When a counter is already set to q, arriving packets are discarded before entering SLOB since buffer overflow has occurred. By delaying packets in this way, SLOB effectively buffers them, since the same delay would be experienced in a real output buffered switch.

It will be shown below how the m buffers that SLOB emulates are implemented by stages of switches and delay lines. It is not possible to isolate a particular SLOB buffer in the architecture because each component forming SLOB (such as a delay line) may emulate part of one or more SLOB output buffers. Hence, SLOB may be said to implement a type of shared buffering.

III. Architecture

Throughout this paper, the stage under consideration is stage i $(i = 0, \cdots, k)$ unless stated otherwise.

A. Architectural Options

In the SLOB architecture (Fig. 3), assume provisionally that stages zero to k (each denoted as "PSE" in the diagram) are $m \times m$ nonblocking space switches—initially ignore the lines coming from under them leading to the switches of size $(k+1) \times 1$ at the outputs. This architecture offers each packet any delay between zero and the SLOB buffer depth of $q = m^k - 1$ timeslots due to the delay line configuration between stages. However, contention occurs whenever several packets try to enter the same delay-line at once (Fig. 4). One solution involves making each stage an $m \times m$ output buffered switch, resolving the contention. Unfortunately, additional delay would be introduced by the buffers in each stage, upsetting the required target delay, while the throughput would be as low as $1/m$, since the zero delay line is often used by all packets in later stages. Also, each stage may well require a buffer depth equal to or greater than the SLOB itself, defeating the original objective.

To solve these problems, packets must be distributed more evenly over the delay lines by temporarily introducing an additional fictitious space switch before each set of delay lines. Each stage consists of an $m \times m$ output buffered switch (called a primitive switching element, or PSE) followed by an $m \times m$ fictitious space switch, then the delay lines (Fig. 5). These fictitious space switches are used as a conceptual aid and are eventually discarded—this will be explained later—since their function can be absorbed by their neighboring PSE's. The PSE output buffers at stage i work in conjunction with the fictitious space switch to try to ensure the programmed stage delay $d_i m^i$ for each transiting packet. Each PSE has m first-in first-out (FIFO) output buffers (each holding up to $m-1$ packets); it is important to distinguish between these and the SLOB output buffers that the SLOB itself is emulating, as a whole. Each PSE buffer is connected by the fictitious switch to the following stage delay line so as to try to implement the correct stage delay. The following Sections will specify the fictitious switch behavior and PSE buffer timing.

B. Fictitious Space Switch Operation Within SLOB and Packet Delay Implementation

"$a \rightarrow b$" means "on the stage i fictitious space switch, input a is connected to output b." Each $m \times m$ fictitious space switch is controlled so that on the first m^i timeslots, i.e., timeslots $t \in \{0, \cdots, m^i - 1\}, [1 \rightarrow 1, 2 \rightarrow 2, \cdots, m \rightarrow m]$. For m^i subsequent timeslots, $[1 \rightarrow m, 2 \rightarrow 1, 3 \rightarrow 2, \cdots, m \rightarrow m-1]$, and for m^i further timeslots after that, $[1 \rightarrow m-1, 2 \rightarrow m, 3 \rightarrow 1, 4 \rightarrow 2, \cdots, m \rightarrow m-2]$ and so on until $[1 \rightarrow 2, 2 \rightarrow 3, 3 \rightarrow 4, \cdots, m-1 \rightarrow m, m \rightarrow 1]$; then the cycle, which is in total m^{i+1} timeslots long, repeats itself. In general, $j \rightarrow ((j - 1 - \lfloor t/m^i \rfloor) \bmod m) + 1$, where $\lfloor x \rfloor$ is the largest integer less than or equal to x.

Fig. 6 shows the operation of a fictitious space switch with $m = 4$, starting at timeslot t, and proceeding at intervals of m^i timeslots. At timeslot $t = 4m^i$, the switch will be back in the "straight through" state, as at timeslot t. Loosely speaking, one can say that the switch shifts its signals further toward its upper outputs as time progresses. After an input is connected to the top output, it "wraps round" to the lower output. The reader may wish to verify that this conforms to the formula given in the previous paragraph.

There is a fictitious space switch after every PSE except the final one (Fig. 5). Just before PSE i, F_i is the remaining delay to be experienced in SLOB by a packet. If $d_i m^i$ is the delay to be experienced by the packet in question within stage $i, F_i = \Sigma_{j=i}^{k-1} d_j m^j$ hence $d_i m^i = F_i \bmod m^{i+1} (0 \leq d_i \leq m-1)$. The packet is routed to the PSE output buffer presently connected to the delay line of that value (i.e., $d_i m^i$) by the fictitious space switch. Due to the shifting action of the fictitious space switch, the total delay in stage i is always either $d_i m^i$ or $d_i m^i + m^{i+1}$, regardless of the PSE buffer delay.

To show this, let the PSE buffer contain b_i packets just before the packet is queued; for the system to operate correctly, this must result in the packet being delayed in the PSE buffer

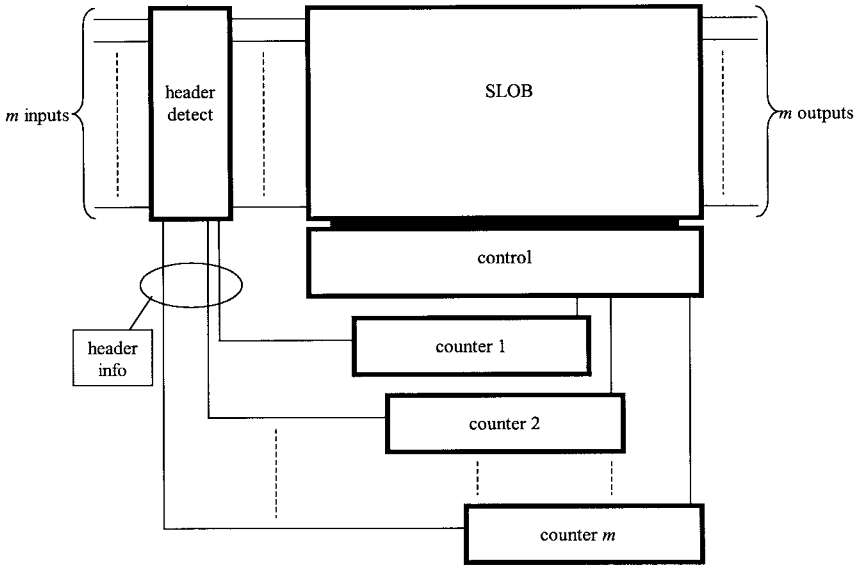

Fig. 2. Interconnection of SLOB, its controllers, and the counters representing the number of cells in each emulated output buffer.

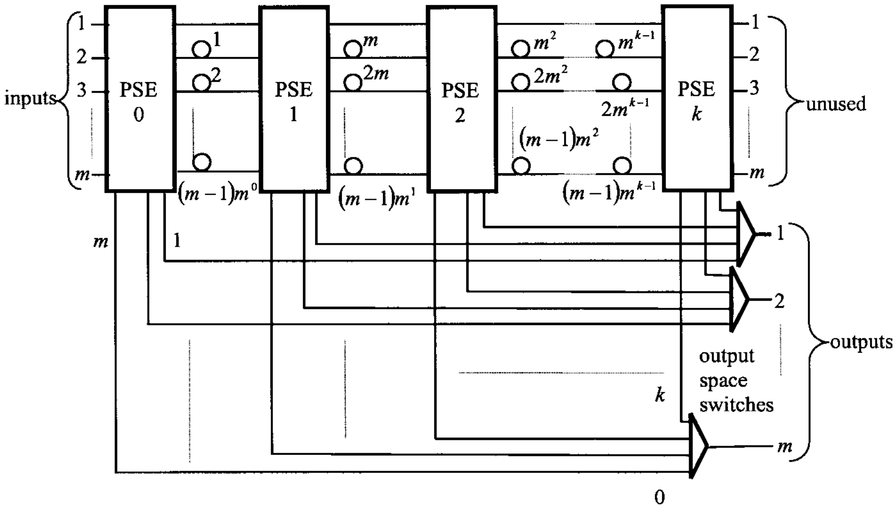

Fig. 3. SLOB architecture; each stage is a primitive switching element (PSE) in the final design.

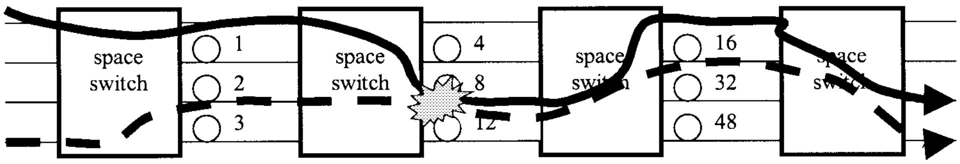

Fig. 4. A packet wishing to experience a delay of eight timeslots and a packet (dotted line) wishing to experience a delay of 26 timeslots undergoing contention at the eight-timeslot delay line.

by $b_i m^i$ timeslots. Then $b_i m^i$ timeslots later when it leaves the PSE buffer, the fictitious space switch has reconfigured b_i times. If $b_i \leq d_i$, the packet traverses a delay line $(d_i - b_i)m^i$ timeslots long, yielding a total stage delay of $(b_i + (d_i - b_i))m^i = dm^i$, as required. If $b_i > d_i$, the output of the PSE buffer, via the space switch, has "wrapped around" from delay line zero to delay line $(m-1)m^i$ while the packet was in the PSE buffer; this happens no more than once since it will be shown later that the PSE buffer is only $m-1$ packets deep. Hence, the delay line is $(d_i - b_i + m)m^i$ timeslots long, and the total stage delay is $(b_i + (d_i - b_i + m))m^i = d_i m^i + m^{i+1}$, m^{i+1} greater than before. This additional delay is compensated for in subsequent stages by reducing F_{i+1}, the delay experienced there, by m^{i+1} timeslots. The packet may therefore already have experienced more than the remaining desired delay F_i, this problem being addressed shortly.

Again, for $m = 4$, i.e., four SLOB inputs and outputs, Fig. 7 illustrates an example of this process. Each part of the diagram

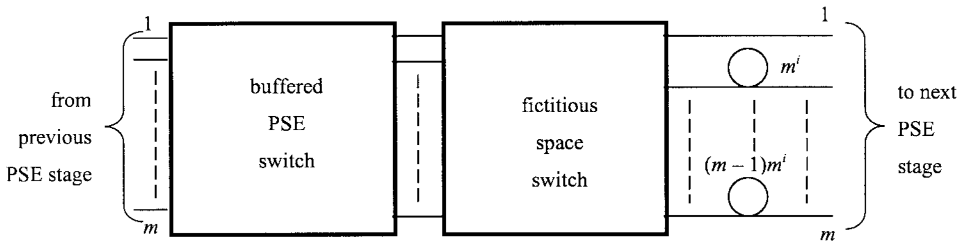

Fig. 5. Interconnection of a PSE and a fictitious space switch in stage i; each has m inputs and outputs. The fictitious space switch is followed by a set of delay lines.

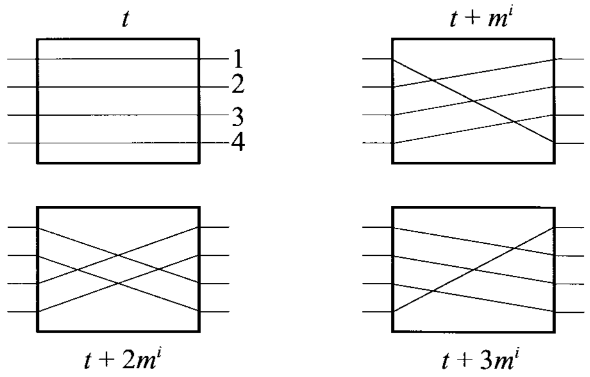

Fig. 6. Operation of a fictitious space switch, showing configuration for $m = 4$ at four instants m^i timeslots apart.

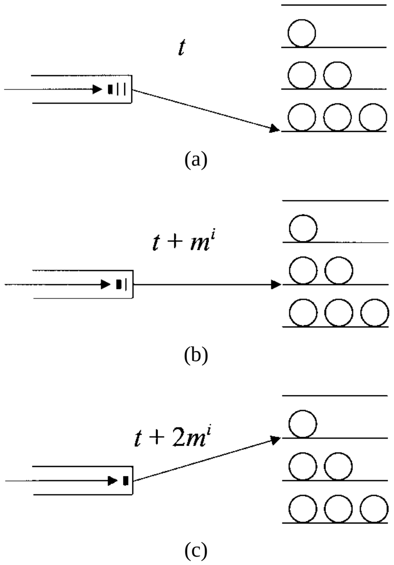

Fig. 7. Interaction of a PSE buffer with the fictitious space switch and the delay lines, at intervals m^i timeslots apart. Each delay loop is m^i timeslots long. In (a), the packet in question (thick line) has two other packets ahead of it in the queue. (b) The PSE output buffer shown is connected by the fictitious space switch to the delay line of the desired length (i.e., $3m^i$) when the packet arrives. By the time the packet of interest reaches the head of the queue in (c), it is connected to the delay line of length m^i, thus compensating for the delay in the queue.

represents one PSE buffer, one input of the fictitious space switch and the array of delay lines feeding into the next stage. When the packet arrives at the PSE [Fig. 7(a)], desiring a delay of $3m^i$, it is put into the buffer (the one shown) which connects to the delay line of that length. There are two packets ahead of it in the PSE buffer. By the time it has moved to the head of the buffer [Fig. 7(c)], $2m^i$ timeslots have elapsed, the fictitious space switch has reconfigured twice, and the packet is put into the delay line of length m^i. Hence, the total delay is the delay in the buffer plus the delay in the delay line, i.e., $2m^i + m^i = 3m^i$ timeslots, which is what was required.

When several packets in one slot arrive at stage i wishing the same delay $d_i m^i$, they must all be stored in the same PSE output buffer.

C. PSE Buffer Timing

To permit their correct operation with the fictitious space switches, each PSE buffer must schedule a packet entering it on timeslot t (with reference to some arbitrary origin $t = 0$) to leave on a timeslot in

$$E = \{t, t + m^i, t + 2m^i, \cdots, t + (m-1)m^i\}. \tag{1}$$

The packet leaves the PSE buffer on the first timeslot in during which another packet has not already been scheduled to leave. It is shown in Section III-D that this PSE buffer timing is easily and naturally implemented in optical delay-line systems by making all delay line lengths multiples of m^i timeslots. From the above expression for E, each PSE buffer has size $m - 1$ packets; it is shown in Appendix A that this is a requirement for zero PSE packet loss, regardless of loading level or traffic statistics. Although this may appear counter-intuitive, within SLOB a highly artificial traffic scenario is experienced by the PSE's due to the way packets are scheduled. Since there is no PSE buffer overflow, all packet loss is due to overflow of the buffers that SLOB emulates, i.e., SLOB implements an output-buffered switch. Both buffer types are quite distinct and must not be confused.

D. Stage Design

The stage design (Fig. 8) initially consists of a PSE (based on the OASIS switch [8]) followed by the fictitious space switch discussed above. The tunable wavelength converters wavelength encode each packet, sending it via the $2m \times 2m$ arrayed waveguide (AWG) filter[1] [15] to the correct output at B. The active demultiplexers (wavelength insensitive $1 \times m$ space switches) carry at most one packet per timeslot, presenting each packet to the appropriate delay line at A, from zero to $m^i(m-1)$ timeslots in length in steps of m^i, implementing the PSE buffer timing of Section III-C. The parallel bank of delay lines at A implements all m PSE buffers, with each

[1] The authors are grateful to Philip Lin of Tellabs Research Center for suggesting the use of an AWG device in this context.

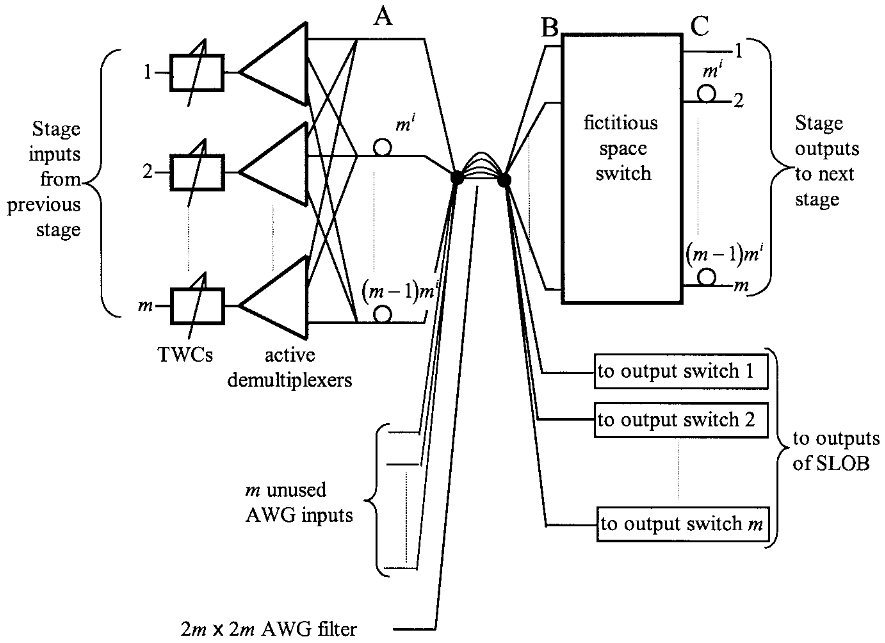

Fig. 8. SLOB stage i. TWC = tunable wavelength converter. The primitive switching element is everything to the left of the space switch. The delay lines at C are also shown in Fig. 3.

packet being sent to the correct output by encoding it with the correct wavelength.

When the stage delay is greater than the remaining desired delay F_i, it must be true that $d_i m^i + m^{i+1} > F_i$, i.e., $F_i \bmod m^{i+1} + m^{i+1} > F_i$, which is only possible if $F_i < m^{i+1}$. Packets having a remaining delay $F_i < m^{i+1}$ are wavelength encoded so they are sent to the correct output switch via the lower m AWG outputs; the packets are then delayed by F_i at A (Fig. 8). They are gathered by the additional lines extending from the lower m sides of each stage in Fig. 3 and selected by the output switches to form the m SLOB outputs. There is no conflict when this happens, for two reasons. First, there is never more than one packet scheduled to go to a SLOB output at once since it is emulating an output-buffered switch. Hence, no contention can arise in or around the output space switches. Second, the wavelengths of all packets going directly to the SLOB outputs are distinct from those packets going forward to B in Fig. 8. Hence, each packet leaves the chain once $F_i < m^{i+1}$. Since unwanted crosstalk signals from each PSE accumulate at each output if passive multiplexers are used there, active switches are employed instead to gate out this unwanted crosstalk. In the penultimate PSE (stage $i = k-1$), $F_i < m^{i+1} = m^k$ always, hence, no packets traverse the fictitious space switch or the delay lines at C; these are all omitted. Consequently, no packets ever reach stage k, which is also omitted.

(Alternatively, a SLOB delay of $\delta_0 = \Sigma_{i=0}^{k-2} m^{i+1} = (m^k - m)/(m-1)$ may be used to represent an empty SLOB buffer, so $F_i \geq m^{i+1}$ in all but the final stage, stage $k-1$; δ_o is added to the counter values to yield the SLOB delay assigned to packets, and the effective SLOB buffer depth reduces to $m^k - 1 - \delta_0$. This removes the need for output switches and the lines feeding them from stages $0, \cdots, k-2$ since all packets now leave from the final stage, $k-1$; furthermore, $m \times m$ AWG's may now be used throughout. This does not affect the conclusions of Appendix A.)

All fictitious space switches are redundant, since each PSE can divert each incoming packet to any output by converting it to the appropriate wavelength; the functionality of each fictitious space switch is incorporated into the neighboring PSE, further facilitating the architecture's implementation. Fictitious space switches were only introduced to facilitate the discussion. For example, in Fig. 5, suppose the hardware is controlled so that a certain packet enters the buffered PSE switch on input 2, and suffers a delay of five timeslots, before exiting it on output 6. It then enters the fictitious space switch on input 6, and emerges on output 3. The equivalent functionality of both the units may be provided by the PSE alone, with the packet entering on input 2, suffering a delay of five timeslots and exiting on output 3.

IV. Control

A parallel electronic fabric, similar in structure to the optical fabric, controls SLOB, as previously proposed for optical space switches [16]. For each optical packet, only a small message is transmitted electronically, representing the remaining delay over the remaining PSE's ($\log_2\ m^{k-i-1}$ bits) and destination SLOB output number ($\log_2\ m$ bits). If $(k-i)\log_2\ m$ parallel planes are used, the controller operates at the timeslot rate. Each electronic controller's PSE output buffer (Fig. 9) has $m^i(m-1)$ message storage locations per output. On each timeslot, the switches at the input and output of each buffer array advance, emulating the PSE buffer timing of Section III-C. A speedup is not necessary, even although several cells may be placed in a PSE output buffer simultaneously. The fictitious space switches following each PSE are implemented in the electronic fabric. From their state and also the delay through each buffered switch within

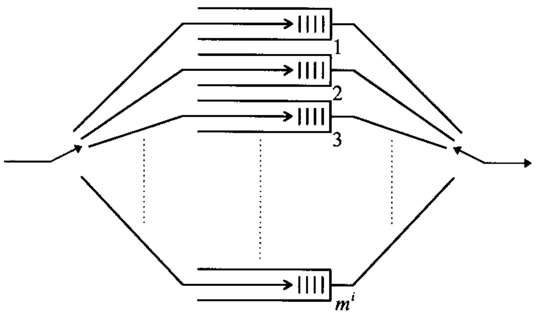

Fig. 9. Output buffer for electronic PSE.

PSE's, the electronic control logic determines the packet delays within each stage, and hence the control signals for the PSE tunable wavelength converters and active demultiplexers (Fig. 8). Synchronization of both fabrics would require careful consideration in a practical system.

To demonstrate the practicality of such a controller, its memory requirements are now evaluated. The number of controller PSE buffer memory bits B_b is

$$B_b = \left[k(m^k - 1) - (k-1)m^k + \frac{m^k - m}{m-1}\right] m \log_2 m. \tag{2}$$

For example, a SLOB with $m = 8$ inputs and outputs, and $k = 4$ stages (used as an example throughout, with SLOB buffer depth 4095) requires 112 224 bits of controller buffer memory. The number of memory bits B_d required to implement all the electronic delay-lines in the controller is

$$B_d = \left[k(m^{k-1} - 1) - (k-2)m^{k-1} + \frac{m^{k-1} - m}{m-1}\right] \cdot \frac{m \log_2 m}{2}. \tag{3}$$

In the example just given, 13 104 bits are required.

Each tunable wavelength converter produces one of $2m$ wavelengths, needing $\log_2 m + 1$ control points. Each active demultiplexer requires m control points, and each header processor produces $\log_2 m$ address signals. Hence, the total number of pins (excluding power, clocks, etc.) is $km(m + \log_2 m + 1) + m \log_2 m$. The present example requires 408 pins, well within the bounds of current technology.

With a 20-Gb/s optical bit rate and 53-byte packets, the packet rate is 47.17 MHz. Hence, for a single-chip controller, ≈126 Kbit static RAM with associated random logic is required, with ≈425 pins and a system speed of ≈50 MHz, since no speedup is required. This specification is completely feasible with today's full custom technology; chips operating at such speeds with similar or greater integration density are available. Furthermore, it is becoming increasingly feasible in semicustom mask programmable technology; mask programmable gate arrays (MPGA's) with over two million available gates have been reported [17], with RAM memory available as embedded macros [18]. Moreover, VHDL synthesis would be relatively straightforward due to the highly repetitive and regular nature of the control logic.

V. PSE Implementation and Cascadability Considerations

SLOB exhibits a power penalty due to noise and crosstalk in its components. Here, a simplified optical model is presented to demonstrate that large SLOB switches are feasible, exhibiting a reasonable power penalty. The analysis is intended merely to demonstrate the feasibility of the SLOB concept; for this reason, and for reasons of space, this section is kept brief.

A. Transmission Delay Variations and Signal Power Loss

Dispersion and temperature variations cause jitter, which is compensated for by a guard band on either side of each packet. The packets are retimed at the SLOB output electronically. In the example above, the maximum total fiber length a packet can traverse is 17.45 km (how often all this is traversed depends on the packet arrival rate). This corresponds to a total delay of 4095 packets with a delay each of 21.8 ns. Dispersion of -15 ps/km · nm and a 3.5 nm wavelength spread (0.5 nm spacing) yield a 1.844 ns delay variation; a guard band of 0.922 ns on either side of each packet. Alternatively, dispersion shifted fiber could be employed, virtually eliminating delay line dispersion. Temperature variation of $\pm 1\,^\circ$C corresponds to an optical path length change of $\approx \pm 6 \times 10^{-6}$, making the guard bands 0.525 ns larger; this is the total delay (4095 × 21.8 ns) multiplied by 6×10^{-6}. A total guard band size of 1.447 ns leaves $((53 \times 8)/20) - 2 \times 1.447 =$ 18.306 ns for the data, yielding an instantaneous packet bitrate of 23.16 Gb/s, i.e., a mean bitrate of 20 Gb/s. The guard band requirement may be relaxed by introducing extra wavelength conversion in the last stage prior to the delay lines, so that all packets traverse them at the same wavelength. With a fiber attenuation of 0.2 dB/km, 17.45 km of fiber has a loss of 3.5 dB, not significantly affecting the model. Connection losses are ignored as they unnecessarily complicate the analysis.

B. The Model

To evaluate the PSE cascadability when forming a SLOB, an idealized model was developed, representing one signal path through a 20 Gb/s PSE (Fig. 10). This bitrate was chosen because, at the time of writing, it represents the highest speed at which interferometric wavelength converters have been demonstrated. Physical device limitations, such as noise and crosstalk, limit the PSE cascadability and thus also the SLOB buffer depth; with ideal devices, arbitrarily large SLOB buffer depths could be attained. Power loss calculations showed that one semiconductor optical amplifier (SOA) booster was required in each PSE, in addition to that forming part of an active splitter.

In each PSE, the tunable wavelength converter (TWC) is first in the signal path (Fig. 10); the signal then enters a $1 \times m$ splitter, forming part of the active splitter, and having a $10 \log_{10} m + 2$ dB total loss; the 2 dB refers to the excess loss. Thus there is a direct correlation between the number of inputs m and the splitter loss. A further 3 dB loss represents the coupling loss in and out of the amplifier. This switches the packets to the correct delay lines, and boosts the signal for the next device, but with the introduction of noise [19], [20]. It

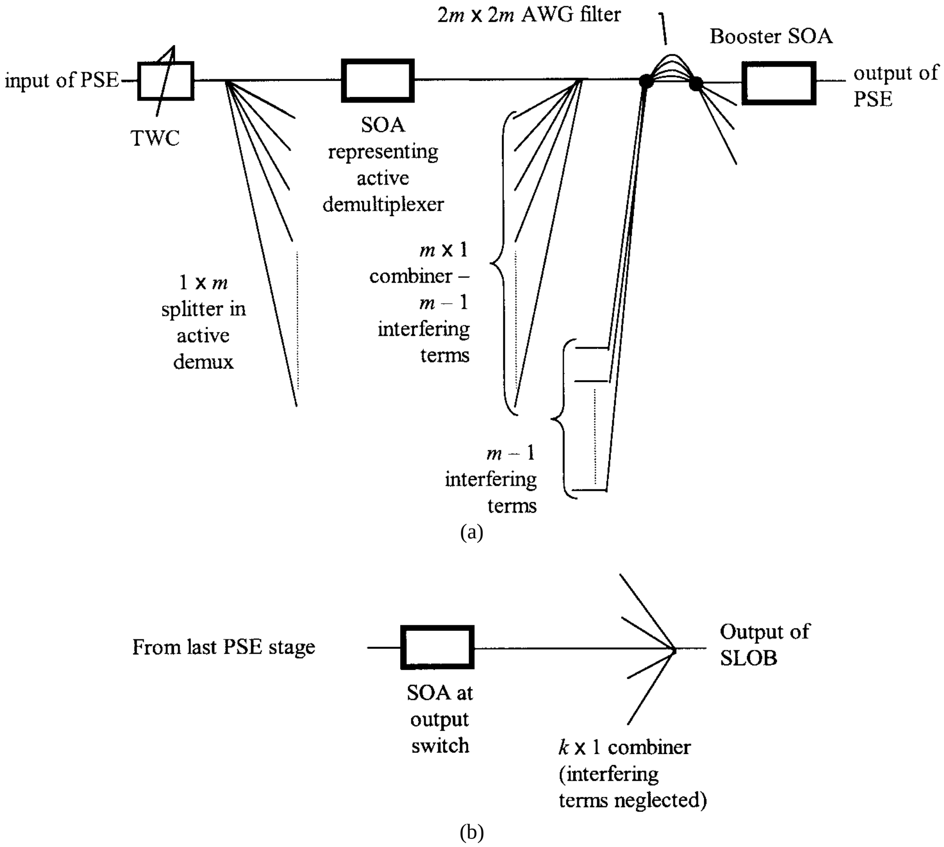
2m x 2m AWG filter
Booster SOA
input of PSE
output of PSE
TWC
SOA representing active demultiplexer
1 x m splitter in active demux
m x 1 combiner – m – 1 interfering terms
m – 1 interfering terms
(a)
From last PSE stage
Output of SLOB
SOA at output switch
k x 1 combiner (interfering terms neglected)
(b)

Fig. 10. The signal path used for the simplified optical model.

has an extinction ratio of 60 dB [21] and a noise figure of 6 dB. The signal passes through an $m \times 1$ combiner (having the same loss as the splitter above) before traversing an arrayed waveguide filter with an isolation of −37.73 dB, and a loss of 2 dB, the latter being achievable in current devices. It then enters a second SOA, which is used as a booster, and is identical in all respects to the first, the gain of both devices being set so that the each stage's overall gain is 0 dB. After the final stage of the simulation, representing the SLOB output, a single SOA represents the output $k \times 1$ space switch; here, crosstalk is neglected since each term is some 37.73 dB + 60 dB = 97.73 dB down. The output of this device is clamped to 7 dBm.

These figures for loss, extinction ratio, noise figure and filter isolation etc. are representative of devices obtainable in practice. While the AWG isolation of −37.73 dB is better than −35 dB, which is likely to be reached in future devices, the isolation quoted for real devices is always for the worst case, adjacent crosstalk term. Examining the characteristics of real devices [22] indicates that the total crosstalk power introduced by a real device with worst-case isolation −35 dB is comparable to that introduced by an idealized device, where every crosstalk term is −37.73 dB. This is because most crosstalk terms in the real device are much less than the worst case. This justifies the figure of −37.73 dB in the model.

C. Assumptions

The SOA's introduce additional detrimental spontaneous noise [19], [20], that is represented in the model and reduces the extinction ratio of the original signal. Noise power accumulates due to amplifier cascading, limiting the number of possible stages. The $m \times 1$ combiner and the AWG devices introduce unwanted crosstalk, which, for the worst case, are all assumed to be at the same wavelength and same linear polarization as the signal. In addition to Olsson's assumptions regarding spontaneous noise and its treatment [19], several further simplifying assumptions were made as follows:

- there is no SOA saturation due to spontaneous noise from previous SOA's, but there is gain saturation due to the signal—this derives from the observation that the signal power is much greater than the spontaneous noise;
- the tunable wavelength converter has zero loss—this is achievable in reality [23];
- the tunable wavelength converter does not add noise—this is justified since it has zero power penalty; although it does add noise, it improves the signal shape [23];
- the AWG filter profile is not important; it is modeled by the loss when a signal goes to the correct output (2 dB) and when it goes to a wrong output (37.73 dB + 2 dB = 39.73 dB);
- the lasers are assumed to be stable with respect to frequency;
- the delay lines have zero loss—as discussed above, delay line loss may be neglected.

D. Results and Discussion

Fig. 11 shows the power penalty in dB against the number of stages. When interpreting these figures, it is assumed that

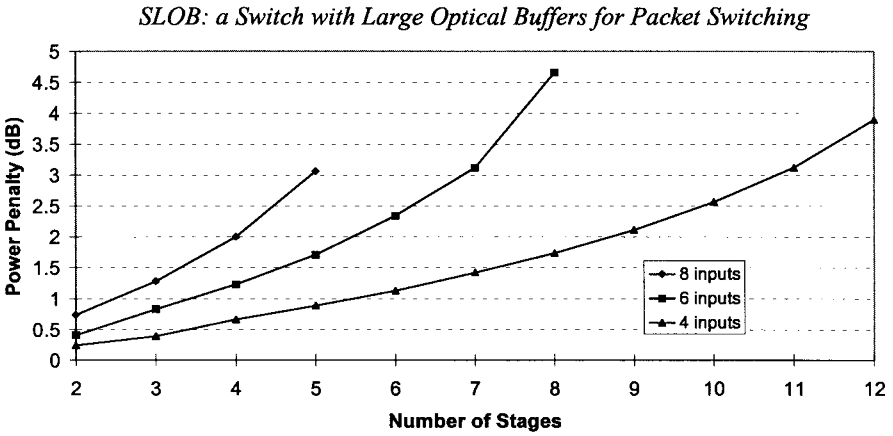

Fig. 11. PSE cascadability, according to the simplified theoretical model.

TABLE I
MAXIMUM POSSIBLE SLOB BUFFER SIZE AND NUMBER OF PSE STAGES, ACCORDING TO THE SIMPLIFIED THEORETICAL MODEL

Number of inputs and outputs	Maximum buffer depth	Number of stages
4	65535	8
6	7775	5
8	4095	4

signal regeneration occurs at the SLOB output; all packets are transmitted between network nodes on the same wavelength, overcoming link dispersion. SLOB is still justified if electronic signal regeneration is possible; a 20 Gb/s regenerator and retimer is much easier to build than an entire electronic switch architecture operating at that speed.

The maximum permissible power penalty is 2 dB, facilitating electronic signal regeneration. Remembering these assumptions, Table I shows the maximum possible SLOB buffer size for various numbers of inputs and outputs, with the corresponding number of stages. SLOB buffers as deep as 65 535 packets are possible for 4 × 4 architectures, reducing to 4095 for 8 × 8 since larger PSE's do not cascade so easily (see Fig. 11). This is because larger splitters and combiners introduce greater loss, and because there are more crosstalk terms per stage. To circumvent this problem, extra regeneration could be introduced between stages.

The model and accompanying figures are pessimistic since a worst case crosstalk scenario is assumed. Indeed, the feasibility of cascading 40 switches, similar to the PSE stages proposed here, has been demonstrated experimentally [24], by using all-optical regenerators. This further strengthens the case for the practicality of SLOB. Here, the bit error rate (BER) is the probability that a given bit in a given packet entering SLOB will emerge at the output with an error. This depends on how full the corresponding SLOB buffer is, and the worst case has been chosen corresponding to the traversal of all stages. While, for point-to-point transmission systems, this is equivalent to taking the time average over a large number of consecutive packets, it is not so here, and it allows the examination of the worst case, yielding a more rigorous definition of performance.

VI. CONCLUSIONS

A new architectural concept for optical packet switching with a large buffer depth has been introduced. This is necessary because existing optical packet switches do not possess a sufficiently high buffer depth to carry the bursty traffic that would be encountered in practice. An existing structure, the OASIS switch, is used as the basis of the fundamental building-block constituting the switch; this building-block itself only has a small buffer depth. Overall buffer depths of many thousands are possible, particularly when using optical regeneration, which has been demonstrated experimentally [24]. The performance is identical to a conventional output-buffered packet switch, hence the switch exhibits optimal delay/throughput performance for a given buffer depth. The switch control has been outlined, and the feasibility of implementing it using today's electronic components has been established.

A simplified theoretical optical performance model has demonstrated the practicality of implementing large optical buffers in this way, with buffer depths of thousands. The model is extremely pessimistic, and experimental work incorporating optical regeneration has demonstrated the cascading of up to 40 primitive switching elements [24]. Even without optical regeneration, electrical regeneration is possible, and SLOB is still justified since a high-speed regenerator and retimer is much easier to build than an entire high-speed electronic switch architecture.

APPENDIX A
PSE BUFFER DIMENSIONING

Referring to Fig. 8, it is shown that a PSE buffer depth of $m-1$ guarantees no packet loss there, regardless of traffic statistics or loading level. In this case, the PSE in stage s is examined, where this may be any stage in SLOB except the last one, i.e., $0 \leq s < k-1$. (The last stage contains no PSE buffers.) Clearly, if this limitation on PSE buffer depth can be shown to be valid for any stage s, then it is true for all stages having buffers, i.e., stages $0, \cdots, k-2$.

A. PSE Buffer Utilization

In this section, two results are derived which are used in the main proof of Appendix A-B. First, it is shown that each packet destined for a given SLOB output uses the PSE buffer adjacent to the previous packet going to that SLOB output, in a cyclic manner. Second, it is shown that to yield a worst case result for a stage s PSE's buffer depth, it should be assumed that all packets pass through this PSE and on to the next PSE. Throughout these proofs, it will be assumed that each PSE buffer is $m-1$ packets in length; it will be shown in Appendix A-B that this is compatible with zero packet loss in each PSE.

1) Time Interleaved PSE's: In PSE i $(0 \leq i \leq s$ throughout) and those to its right, each delay line is a multiple of m^i timeslots; a packet entering it on timeslot t must leave the SLOB on a timeslot in $\{t, t+m^i, t_2m^i, t+3m^i, \cdots, t+hm^i, \cdots\}$. Hence each PSE behaves as m^i independent Time Interleaved PSE's (TIP's), numbered $j = 0, \cdots, m^i - 1$; a packet cannot leave a PSE on a different TIP than that it entered on. TIP j represents PSE buffering over timeslots $j, j+m^i, j+2m^i, j+3m^i, \cdots$etc. where $0 \leq j < m^i$ (this is consistent with the PSE buffer timing of Section III-C).

2) Arrival by a Packet at PSE i Buffers: It is now shown that if a packet's SLOB delay is $m^{s+1}-1$ or more (i.e., there are $m^{s+1}-1$ or more packets in the SLOB buffer) it must reach PSE s. It is also shown that for the worst case, the packet can be regarded as always passing through a stage s PSE buffer, rather than arriving at stage s with $F_s < m^{s+1}$ and being diverted to the SLOB outputs before reaching stage $s+1$, as discussed in Section III-D. This is regardless of whether there is extra delay of m^{j+1} in any stage j, with $j < s$, as described in Section III-B.

To show this, consider the maximum possible delay in some stage j; from Fig. 8, this is the maximum delay at A plus the maximum delay at C, i.e., a delay of $(m-1)m^j+(m-1)m^j = 2(m-1)m^j$. Then the maximum possible delay prior to PSE s is $\Sigma_{j=0}^{s=1} 2(m-1)m^j = 2m^s - 2$. This is always less than $m^{s+1}-1$ hence, a packet cannot experience sufficient delay prior to stage s, which it must enter to do so.

Each packet whose total SLOB delay is $m^{s+1}-1$ is sent directly to the SLOB output by stage s, but this may also happen to packets experiencing a greater total SLOB delay because there have been additional delays in previous stages. It is now shown that a worse (greater) result is always obtained for the PSE buffer depth if it is assumed that the packet had instead proceeded to stage $s+1$. Suppose that such a packet is diverted to PSE $s+1$ instead of being directly being sent to the output; it may be put in any PSE buffer. The PSE buffers utilized by other packets (those going to the next PSE anyway) are determined solely by their arrival time at stage s and delay F_s, and this buffer assignment is not altered by the presence of the diverted packets. Hence the PSE buffers will each contain at least as many packets as they would have done if the packets in question had not been diverted to stage $s+1$ and sent to the output instead.

Therefore, to obtain a worst-case result for PSE buffer dimensioning, it is safe to assume that if the SLOB buffer depth (i.e., the packet delay for that buffer) is $m^{s+1}-1$ or greater, all the packets pass through PSE s into PSE $s+1$.

3) Justification of Worst Case SLOB Buffer Depth Assumption: For the purposes of determining the stage s PSE buffer depth, packets entering SLOB may be regarded as being of two types:

1) the packet has a delay of $m^{s+1}-1$ packets or more—it traverses stage s PSE buffers, even if it would normally go to the correct SLOB output from the PSE in stage s or
2) the packet has a delay of less than $m^{s+1}-1$; it is sent to the correct SLOB output by PSE s or one of the preceding PSE's.

The basis of the argument in Appendix A-B is to contrive a cell arrival scenario which gives the worst possible result for PSE buffer depth. This is done without regard to any correlation that may exist between the packet streams for different SLOB buffers; such a correlation may exist by virtue of packet interactions in previous stages. Hence, any such correlation can be disregarded and it can be said that in the worst case, all SLOB buffers should be regarded as having at least $m^{s+1}-1$ packets in them. This maximizes the packet flow through the stage PSE s buffers, giving the greatest opportunity to fill them with packets.

Hence, the assumption of at least $m^{s+1}-1$ packets per SLOB buffer yields a worst-case result as assumed below. The required PSE buffer depth with each SLOB buffer containing at least $m^{i+1}-1$ packets serves as an upper bound for the general case with no lower limit on the number of packets in each SLOB buffer. Furthermore, assuming that all packets that would have gone to the SLOB output from PSE s will proceed to stage $s+1$ instead produces a still worse result, which is also an upper bound.

4) Cyclic Utilization of PSE Buffers and Packet Ordering Within SLOB: Two assumptions are made here as follows:

- a continuous stream of packets is leaving the SLOB output (this is because it was shown in Appendix A-A3) that it is valid, as a worst case, to assume that the depth of each SLOB buffer is at least $m^{s+1}-1$);
- all packets are passing through this PSE and on to the next one [this assumption was justified in Appendixes A-A2 and A-A3].

Throughout, $0 \leq i \leq s$. It is shown that if packets entering stage s are in order, then they cycle round the PSE buffers in a manner that will be described. It is then shown that if the packets in stage i cycle round the PSE buffers, then they are in order in stage $i+1$. This forms the basis of an inductive proof, for the packets entering stage 0 (i.e., the input of the SLOB) are in order, hence it is true for all stages up to and including PSE s.

A packet enters input n on the stage fictitious space switch at timeslot $t, n = 1, \cdots, m$. From Section III-B, it emerges on output $((n-1-\lfloor t/m^i \rfloor) \bmod m)+1$ and hence enters a delay line of length $L = m^i((n-1-\lfloor t/m^i \rfloor) \bmod m)$, arriving at the next stage on timeslot $t+L$, i.e., TIP number:

$$\left(t + m^i\left(\left((n-1) - \left\lfloor \frac{t}{m^i} \right\rfloor\right) \bmod m\right)\right) \bmod m^{i+1} = (t \bmod m^i + m^i(n-1)) \bmod m^{i+1}. \qquad (4)$$

Suppose a packet enters PSE buffer n $(n = 1, \cdots, m)$ on timeslot t; it then enters fictitious space switch input n, undergoing a delay of L in the stage. It also undergoes a possible additional delay of m^{i+1} (as described in Section III-B) but it still goes to the same TIP in the next stage as otherwise, since timeslots a distance of m^{i+1} apart are all on the same TIP in stage $i+1$. In (4), t mod m^i is constant for a particular TIP, since there, every m^ith timeslot is involved. Hence to select the TIP buffer for the next packet in the sequence going to a particular SLOB output (i.e., increase the above expression by m^i, mod m^{i+1}) the PSE "buffer pointer" n is increased by 1 (set to 1 if it is equal to n).

If $i = s$, i.e., stage s is being considered, those packets that would have otherwise been diverted to the SLOB outputs at stage s [Appendix A-A3] also follow this sequence. They are sent to PSE $s+1$ but may incur an extra delay of m^{i+1} in stage s, i.e., they will suffer a larger delay than desired, since F_{s+1} remains unaltered and is not now adjusted by m^{s+1}. However, they still go to the correct PSE on stage $s+1$, hence they still cycle round the PSE buffers in stage s as described above for the other packets. What happens to these packets after they leave PSE s is of no concern here.

Since the packets are in order, and because, as shown above, n is increased by one for each new packet going to a given SLOB output, those arriving at a TIP and destined for a particular SLOB output cycle around the PSE output buffers. The first packet to arrive goes to, say, PSE buffer 1, the next arriving on the same TIP goes to PSE buffer 2, and so on. If several packets arrive at once, they go to several consecutive PSE buffers simultaneously. This observation is crucial to the proof in Appendix A-B.

Continuing the assumption that the packets are in order in stage i, consider the SLOB output buffer feeding some SLOB output z. Since the SLOB buffer is assumed never to be empty, packets leave SLOB output z on consecutive timeslots. Since the delay lines in stage i and those stages to its right are multiples of m^i, every m^ith packet in this sequence going to output z passes through the same stage i TIP. Likewise, every m^{i+1}th packet in the sequence going to SLOB output z must pass through the same stage $i+1$ TIP. Therefore every mth packet in the sequence going through a TIP in stage i must pass through the same TIP in stage $i+1$.

It can be seen from (4) that each stage $i+1$ TIP is accessed via a unique stage i TIP buffer, so all packets transmitted to a stage $i+1$ TIP pass through the same stage i PSE buffer. Hence, they cannot become unordered due to the FIFO PSE buffer discipline, so must therefore pass through stage $i+1$ TIP in the same order they leave SLOB. Thus the hypothesis is true for stage $i+1$, completing the proof by induction.

B. PSE Buffer Depth

$b(T)$ is the number of packets in a designated stage s TIP buffer (say the top one), exactly Tm^s timeslots after $T = 0$. T is set to zero just before the first TIP timeslot after every time the TIP buffer is empty; for $T \geq 1$ it is nonempty. $c_j(T)$ is the number of packets that have arrived at the TIP destined for SLOB output j, during timeslots $0, m^s, 2m^s, \cdots, (T-1)m^s$. Since the designated TIP buffer is nonempty for $T \geq 1$, one packet leaves it during every timeslot under analysis.

$\Sigma_{j=1}^{m} c_j(T) = u \leq mT; u$ is the total number of packets transmitted through the TIP from timeslot 0 up to (but not including) timeslot Tm^s. Assuming the worst case (i.e., the first packet to arrive for any SLOB output always goes to the designated TIP buffer), and remembering that one packet leaves the designated TIP buffer every timeslot

$$b(T) = \sum_{j=1}^{m} \left\lceil \frac{c_j(T)}{m} \right\rceil - T \tag{5}$$

(where $\lceil x \rceil$ is the smallest integer larger than or equal to x). Since the packets arriving at the TIP and destined for a particular SLOB output cycle round the PSE output buffers (Appendix A-A4), every mth of these packets arrives in the designated PSE buffer, justifying the division by m above.

To maximize the answer $b(T)$, set $c_1(T) = u - m + 1$ and each of the remainder to 1. While certain other values give equally large results, none give a larger value for $b(T)$

$$b(T) = (m-1) \left\lceil \frac{1}{m} \right\rceil + \left\lceil \frac{u-m+1}{m} \right\rceil - T. \tag{6}$$

$u = mT$ in the worst case, yielding a maximum required PSE buffer depth of $m-1$. On the first timeslot, one packet arrives at the TIP for each SLOB output (m packets in total, with $m-1$ remaining in the PSE buffer). On subsequent timeslots m packets arrive for SLOB output 1 and none for the other outputs, so one packet arrives at the TIP buffer; since one leaves, $m-1$ remain. No assumptions were made about traffic distribution or loading; no PSE packet loss ever occurs if the PSE buffers are of this depth.

APPENDIX B
2 × 2 BUFFERED OPTICAL PACKET SWITCHING MODULE

It is now shown that SLOB is a general case of a previously reported 2 × 2 buffered optical packet switching module [25]. According to Appendix A-B, a SLOB with two inputs and outputs requires TIP buffers of depth 1. By assuming the switch traffic has a density of one—a packet occupies every timeslot—it will be shown that when $m = 2$ the required PSE buffer depth is zero, i.e., each PSE is simply a 2 × 2 space switch. Such loading makes the switch unstable, but it will be shown that it may be used with smaller loadings while still retaining this assumption concerning the PSE buffers.

The switch operates with both SLOB buffer lengths summing to a constant $q = 2^k - 1$, and when an empty timeslot arrives it is put in the emptier SLOB buffer. Empty timeslots are treated as packets since they are each put into a SLOB buffer and each use up a SLOB buffer position. Because of this, the SLOB delay experienced by real packets is not generally optimal due to the empty timeslots ahead of them in each buffer; the delay is greater than in an output buffered switch. Placing empty timeslots into the emptier SLOB buffer always drives the SLOB toward the state where both SLOB buffers have an equal number of packets, preventing it from being unstable. The PSE "sees" a load of one because the empty packets are being put in the SLOB buffers as well.

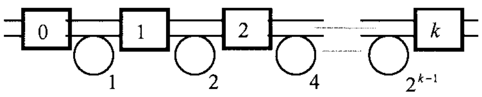

Fig. 12. 2 × 2 SLOB with zero PSE buffering.

(Empty timeslots are not treated this way in the main part of the paper because this procedure for reducing the PSE buffer depth to zero is not employed there.) Retaining the notation of Appendix A

$$c_1(T) + c_2(T) = 2T. \tag{7}$$

The TIP's are initialized at $T = 0$ so that the last packet going to SLOB output 1 was put into a different TIP buffer from the last packet going to SLOB output 2. Hence, either

$$b(T) = \left\lceil \frac{c_1(T)}{2} \right\rceil + \left\lfloor \frac{c_2(T)}{2} \right\rfloor - T \tag{8}$$

or

$$b(T) = \left\lfloor \frac{c_1(T)}{2} \right\rfloor + \left\lceil \frac{c_2(T)}{2} \right\rceil - T. \tag{9}$$

In the first case, there are two possibilities; since they sum to an even number, either $c_1(T)$ and $c_2(T)$ are both odd or they are both even. If they are both odd

$$\begin{aligned} b(T) &= \frac{c_1(T)+1}{2} + \frac{c_2(T)-1}{2} - T \\ &= \frac{c_1(T)+c_2(T)}{2} - T \\ &= \frac{2T}{2} - T = 0. \end{aligned} \tag{10}$$

The same result is obtained when they are both even, and also in the second case referred to above. Thus, no buffering is required in the PSE's here, and the architecture is identical to that of [25] (Fig. 12). Off-loading of packets directly to the output is not required, as there are no buffers hence no additional SLOB delay of m^{i+1} timeslots which would have been due to the interaction of the PSE buffers with the fictitious space switches and the delay lines (Section III-B).

Acknowledgment

The authors acknowledge useful discussions with P. Lysaght and G. McGregor of the University of Strathclyde, U.K., on control implementation technologies. They also thank R. Gardner of the University of Strathclyde, D. Cotter of BT Labs., and P. Lin of Tellabs Research Center for their helpful comments on previous versions of the manuscript. They are grateful to the anonymous referees, whose relevant and helpful comments greatly improved this paper. The authors would also like to thank B. Buchold of Dortmund University for her assistance in supplying information on AWG devices.

References

[1] Y. Choi, H. Tode, H. Okada, and H. Ikeda, "A large capacity photonic ATM switch based on wavelength division multiplexing technology," *IEICE Trans. Commun.*, vol. E79-B, no. 4, pp. 560–568, Apr. 4, 1996.

[2] D. Chiaroni, C. Chauzat, M. Sotom, D. De Bouard, F. Masetti, J. C. Jacquinot, J. L. Moncelet, M. Bachmann, P. Doussiere, M. Goix, and E. Grard, "Feasibility issues of a high speed photonic packet switching fabric based on WDM subnanosecond optical gates," in *Proc. ECOC '96*, Oslo, Norway, pp. 127–130.

[3] Y. Nakahira, H. Inoue, and Y. Shiraishi, "Evaluation of photonic ATM switch architecture—Proposal of a new switch architecture," in *Proc. 15th Int. Switching Symp. (ISS'95)*, vol. 2, Apr. 1995.

[4] Z. Haas, "The staggering switch: An electronically controlled optical packet switch," *J. Lightwave Technol.*, vol. 11, pp. 925–936, May/June 1993.

[5] W. Pieper, M. Eiselt, A. Ehrhardt, G. Großkopf, R. Langenhorst, and H. G. Weber, "Cross talk in a fiber-loop optical buffer," in *Proc. OFC '94*, San Jose, CA, Feb. 20–25, 1994, paper ThD1, pp. 194–195.

[6] B. Bostica, P. Cinato, A. de Bosio, E. Garetti, and E. Vezzoni, "Electro-optical ATM digital cross-connect system based on cell aggregation and compression," in *Proc. 14th Int. Switching Symp.*, Yokohama, Japan, Oct. 25–30, 1992.

[7] M. Calzavara, P. Gambini, M. Puleo, B. Bostica, P. Cinato, and E. Vezzoni, "Simultaneous buffering of ATM packets in a multiwavelength optical-fiber-loop memory," in *Proc. OFC '94*, Feb. 20–25, 1994, pp. 195–196.

[8] J. M. Gabriagues and J. B. Jacob, "OASIS: A high-speed photonic ATM switch—Results and perspectives," in *Proc. 15th Int. Switching Symp.*, Berlin, Germany, Apr. 1995, vol. 2, paper C8.4, pp. 457–461.

[9] A. Misawa, Y. Yamada, M. Tsukada, K. Habara, and T. Matsunaga, "20 Gbit/s photonic ATM switching with differential packet receiver," *Electron. Lett.*, vol. 32, no. 19, pp. 1798–1799, Sept. 12, 1996.

[10] J. Dupraz, "ATM: Current status and perspectives of evolution," in *Proc. ECOC '94*, Firenze, Italy, Sept. 26–29, 1994, pp. 555–562.

[11] D. Chiaroni, P. Gavignet-Morin, J. B. Jacob, J. M. Gabriagues, J. Jacquet, D. De Bouard, G. Da Loura, and C. Chauzat, "Feasibility demonstration of a 2.5 Gbit/s 16 × 16 ATM photonic switching matrix," in *Proc. OFC '93*, San Jose, CA, Paper WD2, Feb. 21–26, 1993, pp. 93–94.

[12] A. Misawa and T. Matsunaga, "Factor measurements of photonic ATM switch with four-channel FDM output buffer at 2.5 and 10 Gbit/s," *Electron. Lett.*, vol. 32, no. 17, pp. 1604–1605, Aug. 15, 1996.

[13] M. G. Hluchyj and M. J. Karol, "Queueing in high-performance packet switching," *IEEE J. Select. Areas Communications*, vol. 6, pp. 1587–1597, Dec. 1988.

[14] M. J. Karol and M. G. Hluchyj, "Input versus output queueing on a space-division packet switch," *IEEE Trans. Commun.*, vol. 35, pp. 1347–1356, Dec. 1987.

[15] C. Dragone, C. A. Edwards, and R. C. Kistler, "Integrated multiplexer on silicon," *IEEE Photon. Technol. Lett.*, vol. 3, pp. 896–899, Oct. 1991.

[16] P. Cinato and A. de Bosio, "Optical technology applications to fast packet switching," in *Proc. OSA Topic. Meeting on Photon. Switching*, Salt Lake City, UT, OSA, LEOS/IEEE, Mar. 1–3, 1989.

[17] Fujitsu Limited, *CE61 Series Data Sheet*, Edition 1.0, Feb. 1996.

[18] Y. S. Yung, "A complete 0.5 μm high performance array family," in *Proc. 8th Annu. IEEE Int. ASIC Conf. Exhibit*, TX, Sept. 18–22, 1995, pp. 53–56.

[19] A. Olsson, "Lightwave systems with optical amplifiers," *J. Lightwave Technol.*, vol. 7, pp. 1071–1082, July 1989.

[20] J. Yao, "Impact of input signals on BER performance of semiconductor laser amplifier based switch systems," *Inst. Elect. Eng. Proc.—Optoelectron.*, vol. 141, no. 4, pp. 265–270, Aug. 1994.

[21] A. Kelly, personal communication, BT Labs, Feb. 1997.

[22] H. Yamada, K. Takada, Y. Inoue, Y. Ohmori, and S. Mitachi, "Statically-phase-compensated 10 GHz-spaced arrayed-waveguide grating," *Electron. Lett.*, vol. 32, no. 17, pp. 1580–1582, Aug. 15, 1996.

[23] B. Hansen, "Cascadability of broadcast and select switch blocks with interferometric wavelength converters," in *Proc. ACTS KEOPS Meeting*, Bologna, Italy, Sept. 1996.

[24] D. Chiaroni, B. Lavigne, L. Hamon, A. Jourdan, F. Dorgeuille, C. Janz, E. Grard, M. Renaud, R. Bauknecht, C. Graf, H. P. Schneibel, and H. Melchior, "Experimental validation of an all-optical network based on 160 Gbit/s throughput packet switching nodes," in *Proc. ECOC '98*, Madrid, Spain, Sept. 20–24 1998.

[25] D. K. Hunter, D. Cotter, R. B. Ahmad, W. D. Cornwell, T. H. Gilfedder, P. J. Legg, and I. Andonovic, "Buffered switch fabrics for traffic routing, merging and shaping in photonic cell networks," *J. Lightwave Technol.*, vol. 15, pp. 86–101, Jan. 1997.

David K. Hunter (S'88–M'90) was born in Glasgow, Scotland, U.K. He received the B.Sc. degree in electronics and microprocessor engineering with first class honors from the University of Strathclyde, Scotland, U.K., in 1987 and the Ph.D. degree from the same university in 1991 for research into switch architectures for optical TDM switching.

Since 1991, he has been a Research Fellow, then Senior Research Fellow at the University of Strathclyde. From 1992 to 1993, he was a Visiting Researcher at BT Labs, Ipswich, U.K., and in 1994, he was a Visiting Scholar at the University of Pittsburgh, PA. Since 1995, he has held a five-year EPSRC Advanced Fellowship. His research interests include optical switch architectures, telecommunications switch modeling, network routing and network survivability.

Dr. Hunter is an associate member of the Institution of Electrcial Engineers.

W. David Cornwell was born in Glasgow, Scotland, U.K. He received the B.Eng. degree in electrical and electronic engineering with first class honors from the University of Strathclyde, U.K., in 1994. He received the Ph.D. degree from the University of Stathclyde in optical interferometric noise models and their experimental verification, used subsequently in the determination of the limitations imposed by this noise on the performance of optical networks.

He is presently at BICC plc, Helsby, U.K., employed as a Research Engineer responsible for the design and test of optical networks.

Dr. Cornwell is a student member of the Institution of Electrical Engineers.

Tim H. Gilfedder received the Bachelor's degree in theoretical physics from Exeter University, U.K., in 1989. He received the Master's degree in optical electronics from the University of Strathclyde, U.K., in 1993. He is currently working towards the Ph.D. degree.

He then spent a number of years working as a Software Developer with a naval defense contractor in Portsmouth, U.K. After receiving his Master's degree, he then continued his research into physical layer modeling of optical networks at Strathclyde as a postgraduate research student, spending nine months at BT labs, Ipswich, U.K., and a further six months at BICC Cables Research Centre, Helsby, U.K. He joined BT Labs in April 1997 as part of the Network Structures group within the Transport, Design and Performance Unit. He is currently part of the WDM features team that is heavily involved with the introduction of WDM into BT's network. He is a graduate member of the Institute of Physics.

André Franzen received the Dipl.-Ing. degree in electronic and electrical engineering from the Ruhr-Universität, Bochum, Germany, in 1993. After that he worked in the area of logistics. Since 1996, he has been studying at the University of Strathclyde, Scotland, U.K., for the Ph.D. degree in optical networking with a focus on synchronization.

Ivan Andonovic (S'79–SM'97) has been with the Department of Electronic and Electrical Engineering, the University of Strathclyde, since 1985 and is currently a Professor of Broadband Networks. Previously, he was a Research Scientist for three years at Barr and Stroud, where his responsibilities included design, manufacture, and test of guided wave devices for a variety of applications. His main interests center on the development of guided wave architectures for implementing optical signal processing, optical switching and routing as applied in next generation optical networks. He held a two-year Royal Society Industrial Fellowship in collaboration with BT Laboratories during which time he was tasked with investigating novel approaches to optical networking. He has edited two books and authored/coauthored five chapters in books and over 140 journal and conference papers. He has been Chairman of the IEE Professional Group E13, has held a BT Laboratories Short-term Fellowship, and is Editor of the *International Journal of Optoelectronics*.

Dr. Andonovic is a fellow of the Institution of Electrical Engineers (IEE) and a member of the Optical Society of America (OSA).