The λ-Scheduler: A Multiwavelength Scheduling Switch

Jonathan P. Lang, *Member, IEEE*, Emmanouel A. Varvarigos, *Member, IEEE*, and Daniel J. Blumenthal, *Senior Member, IEEE*

***Abstract*—We propose a new multiwavelength almost all-optical switch architecture called the λ-scheduler that uses wavelength division multiplexing (WDM) internally to fold the switch architecture in both the space and time domains to reduce the hardware complexity and to improve the signal characteristics through the switch. The λ-scheduler preserves the packet order for a given input–output pair, is consistent with virtual circuit switching, and when combined with appropriate connection and flow control protocols, provides lossless communication for *bursty* (or nonconstant rate) traffic, provided the traffic satisfies certain smoothness properties. The λ-scheduler uses novel scheduling and wavelength assignment algorithms, in conjunction with a series of feed-forward delay blocks, to avoid packet collisions within the switch or at the switch outputs. We present two implementations of the λ-scheduler when the number of internal wavelengths k equal the number of inputs (and outputs) N to the switch. In the *compressed* λ-scheduler, the N internal wavelengths are used to fold the architecture in the time domain, which reduces the total number of delay blocks for the switch by $2N \log N$. In the *collapsed* λ-scheduler, the N internal wavelengths are used to fold the architecture in the space domain, which reduces the number of delay blocks and total fiber length used for delays by a factor of N. We examine the insertion loss for both λ-scheduler implementations and discuss the tradeoffs between the reduction in overall component count and the improvement in the signal characteristics.**

***Index Terms*—Architecture, optical delay lines, optical switches, packet switching, photonic switching systems, scheduling.**

I. Introduction

FIBER-OPTIC communication links provide an extremely large (multiterahertz) bandwidth potential with very low loss. Optical networks composed of *almost-all* optical switches, where the data packets remain in the optical domain and only the packet header or control information is processed electronically, can offer large bandwidth gains with extremely fast switching speeds while maintaining data transparency. This is because almost all-optical switches eliminate the need for optical to electronic to optical (O–E–O) conversion of the data. Wavelength division multiplexing (WDM) can further exploit the large bandwidth potential of optical fiber by dividing the total fiber bandwidth into a number of narrower channels. As we will see, the flexibility provided by WDM can be used to fold switch architectures in the space and time domains, leading to a reduction in overall switch complexity and an improvement in the signal-to-noise ratio (SNR).

The design of almost-all optical switches has traditionally been based on emulating electronic switches, for which there are two basic components: the *space switch*, which connects the input ports to the output ports, and the *buffering strategy*, which is used to temporarily store data packets if contention occurs for a common resource (e.g., if multiple packets require the same output port). Depending on its design, the space switch can be either *blocking*, where certain permutations of input–output connections cannot be made, or *nonblocking*, where all permutations of input–output connections can be made. The buffering strategy can be categorized into one of five basic designs: *input buffering*, where there is a separate buffer for each input; *scheduling* (or *input smoothing* [1]), where a frame of T packets is examined at each input before being launched into the switching fabric; *output buffering*, where there is a separate buffer for each output; *shared buffering*, where the buffers are shared among multiple inputs or outputs; and *no buffering*, where contention is resolved by either dropping or misrouting (deflecting) packets. Furthermore, almost-all optical switches can be either single-wavelength systems or multiwavelength (WDM) systems. In the past, many researchers have argued that WDM-based systems were impractical for packet switching applications because wavelength tunable devices were too slow, or their filter characteristics were too broad, or they were simply too expensive. As the technology for wavelength tunable devices is improving, these arguments are losing strength, and researchers are beginning to reexamine WDM as an important facet of optical packet switching.

The *Starlite* [2] shared-buffer switch was one of the first architectures to address the high-bandwidth and buffering issues of optical networks. The Starlite uses a self-routing space switch to route packets from inputs to outputs and recirculating loops (from the outputs of the space switch back to the inputs) to resolve contention. This increases the switch-block complexity since a switch with N inputs and outputs and L buffers requires an $(N+L) \times (N+L)$ space switch instead of an $N \times N$ space switch and preserving the packet order requires complex scheduling algorithms. To maintain a reasonable *packet-loss probability* (PLP), many loops (and multiple recirculations per loop) are required. Recirculating loops, however, require amplification which can introduce amplified spontaneous emission (ASE) noise into the loop. Despite promising work using solutions and loop delays [3], feed-forward delays are generally preferred

Manuscript received August 9, 1999; revised March 8, 2000. This work was supported by the Defence Advanced Research Projects Agency (DARPA) under the MOST Project. The work of E. A. Varvarigos was performed while he was a Visiting Associate Professor at TU Delft.

J. P. Lang is with the Calient Networks, Goleta, CA 93117 USA (e-mail: jplang@calient.net).

E. A. Varvarigos and D. J. Blumenthal are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA (e-mail: manos@ece.ucsb.edu; danb@ece.ucsb.edu).

Publisher Item Identifier S 0733-8724(00)07376-X.

since they have superior crosstalk performance and more uniform attenuation [4]. In *Blazenet* [5], buffering is avoided by using *source routing*, where the sequence of hops is determined at the source, and *loop deflection*, where blocked packets are looped back to the previous node in the route. Using deflections in this manner is equivalent to backpressure-based schemes with a threshold value of a single packet which can waste capacity when the loads are all but minimal (see [6] for a discussion of backpressure-based protocols for multigigabit networks and some of the difficulties that arise in their implementation). Blumenthal *et al.* [7] demonstrated the first photonic deflection routing switch, where blocked packets are deflected to an alternate output (not necessarily leading to the previous node), and packets are routed on a hop-by-hop basis. This is more efficient than loop deflections since the routing is allowed to be adaptive, however, as with packet deflection routing in general, the resequencing of thousands of packets may be required at the destinations due to deflections. A disadvantage of the aforementioned architectures is that they do not preserve the packet order. This is inconsistent with virtual-circuit switching and ATM and may cause long delays since packet delay is not guaranteed. In addition, the resequencing of packets at the destination requires maintaining very large buffers and may cause packet loss due to buffer overflow.

The *staggering switch* [8] was the first optical switch design to truly emulate an output-buffered switch. In the staggering switch, two space switching stages are connected by a parallel set of delay lines of unequal delay. Haas investigated a number of variations in the architecture to address the packet resequencing problem, scalability issues, and the use of WDM internally to share buffers. The design, however, is *wasting*, in that packets may be unnecessarily delayed even if there is available capacity on their desired outgoing link. Furthermore, the architecture does not scale well, as the number of delay lines grows linearly with the buffer depth, and it is not modular, as increasing the number of inputs, outputs, or buffer depth requires redesigning one or both of the space switches. Renaud *et al.* [9] detail two WDM shared output-buffered packet-switching architectures, called the wavelength routing switch (WRS) and the broadcast-and-select switch (BSS), that were developed through the ACTS KEOPS (Keys to Optical Packet Switching) project. The WRS is a two-stage switch that first buffers conflicting packets and then routes them to their desired output, where a tunable wavelength converter (WC_t) is used before each stage to route the packets to the appropriate delay line and output port, respectively. Although the WRS is an improvement over the staggering switch by being nonwasting, it still suffers the same scalability and modularity issues as the staggering switch. In the BSS architecture, an internal wavelength is assigned to each input and a series of passive splitters/combiners are used to broadcast the input packets to the output links, where fast SOA gates are used to select the appropriate packets at each output for each timeslot. Although the BSS architecture is nonwasting and one of the few proposed architectures that can easily provide multicasting, the architecture is also neither modular nor scalable. (In fact, since the output power of a 1 : N passive splitter/combiner is reduced by a factor of $1/N$, the insertion loss through the passive splitters/combiners seriously limits the scalability of the architecture.) Danielson *et al.* [10] present several variations of an output-buffered switch architecture that use a WC_t for each input wavelength when WDM is used externally in the network. They examine some of the tradeoffs between the size of the space switch, the complexity of the switching blocks, and the number of internal wavelengths, and conclude that tunable wavelength converters are essential to reduce complexity and size of a switch.

All of the above switch architectures use a single-stage of delay lines for implementing the buffers; a delay line is selected from a parallel set of delay lines and the buffer size is linearly related to the number of delay lines. It has been projected [11] that for *bursty* traffic, buffer depths of thousands of packets may be required, and therefore, multi-stage architectures are more appropriate for bursty traffic. The *switch with large optical buffers* (SLOB) [12] is a multistage switch architecture that cascades many small output-buffered switches to form a switch with larger buffer depth. SLOB uses WDM interally and the hardware growth is logarithmic with the buffer depth, which is appealing for bursty traffic. The *packing* and *scheduling switches* [13] are multistage switches that are designed to provide lossless communication for bursty traffic, provided the traffic satisfies certain smoothness properties. This is fundamentally different from the other architectures that only provide packet-loss probabilities for regular traffic because it is designed in a coordinated fashion with the network control protocols. (Packet-loss probabilities depend very much on the arrival process, and estimates obtained by assuming independent Bernoulli arrivals may not be appropriate for bursty traffic.) The packing switch uses a simple scheme to assign output slots to the incoming packets so as to minimize the processing requirements, while preventing internal packet collisions within the switch. The scheduling switch is comprised of a scheduler followed by a nonblocking space switch, where the purpose of the scheduler is to rearrange the incoming packets so that packets appearing during the same slot at the outputs of the scheduler require different outgoing links of the space switch. Both the packing and scheduling switch architectures are single-wavelength switches that perform the buffering function using a small number of feed-forward optical delay lines arranged in a multistage fashion.

In this paper, we propose a new switch design, called the λ-scheduler, which uses WDM internally to fold the switch architecture in both the space and time domains to reduce the number of elementary components and the total fiber length required to implement delays. The λ-scheduler is based on the single-wavelength scheduling switch architecture of [13], and uses novel scheduling and wavelength assignment algorithms, in conjunction with a series of feed-forward delay blocks, to avoid packet collisions within the switch or at the switch output, and to maintain the desirable properties of the original scheduling switch architecture. Unlike most of the previously proposed switch architectures that provide packet-loss probabilities for regular traffic with a given load, the λ-scheduler provides lossless communication for bursty traffic (provided it satisfies certain smoothness properties) and packet delay guarantees (delay is at most $2T-1$ at each switch); has a modular design so that it can be easily expanded to accommodate more

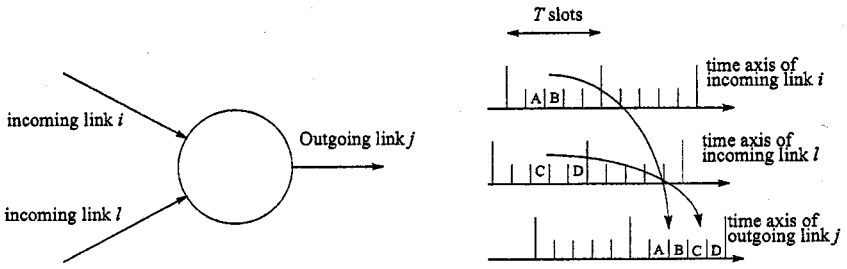

Fig. 1. We illustrate the incoming and outgoing frames at a node. Packets not intended for outgoing link j are not shown. Packets arriving in a particular frame of an incoming link that want to use the same outgoing link are sent over the same frame of the outgoing link.

burstiness in the traffic (in a way similar to the way electronic buffers can be expanded in a conventional electronic switch); and preserves the packet order when combined with appropriate scheduling algorithms. Two implementations of the λ-scheduler are presented when the number of internal wavelengths k is equal to the number of inputs (and outputs) N to the switch. In the *Compressed* λ-scheduler, WDM is used to compress each branch of delay blocks, reducing the total number of delay blocks by $2N \log N$ compared to the scheduling switch. In the *Collapsed* λ-scheduler, WDM is used to collapse the N parallel branches of delay stages to a single branch, which reduces the number of delay blocks by a factor of N compared to the scheduling switch. We perform a first-order loss analysis to examine the signal attenuation through each switch design, and we discuss the tradeoffs between compressing each branch of delay blocks and collapsing the parallel branches.

The organization of the remainder of the paper is as follows. In Section II, we describe our objectives for the switch architecture and our assumptions on the traffic. In Section III, we describe the single-wavelength scheduling switch architecture. In Section IV, we present the λ-scheduler architecture and show how WDM can be used internally to fold the architecture in both the space and time domains. In Section V, we describe a new component called the fast tunable grating reflector (FTGR) that provides the required flexibility to simultaneously switch multiple packets on different wavelengths independently. We present a cost analysis in Section VI that compares two specific implementations of the λ-scheduler, and we discuss the benefits of using WDM to fold the switch architecture. Finally, in Section VII, we conclude the paper.

II. Traffic Assumptions

The λ-scheduler is designed to provide lossless communication for sessions that have certain smoothness properties, or sessions that can tolerate the delay induced when transforming them into smooth sessions. Following the discussion in [14], we view the time axis on a link as being divided into frames of length equal to T slots (see Fig. 1), where we assume that all packets have the same length and require one slot for transmission. In practice, T will be determined by the embedded granularity of the system. For example, $T = 1024$ could correspond to a 16 Gb/s line rate with multiplexed 16 Mb/s signals; in a WDM system, the granularity could be the number of wavelengths per fiber (e.g., 16, 40, 80, 160, ...). A session is said to have the (n, T)-*smoothness property* at a node if at most n packets ($n \in \{1, 2, \ldots, T\}$) of the session arrive at that node during a frame of size T. By using a leaky bucket scheme [15] to shape traffic at the source, and by ensuring frame integrity at the intermediate nodes, a session can be made to have the (n, T)-smoothness property throughout the network. This requires that the frame boundaries are maintained between the transmitting and receiving nodes, and can be done by setting a bit in the header to be either 0 or 1, alternating values between frames. It is important to note that this is not circuit switched data, but instead, this is packet switching with built-in flow control to ensure lossless transmission; packets from a particular virtual circuit do not arrive in the same slot and the number of packets that arrive per frame is not constant, but is bounded by n.

The frame size T is an important parameter and can be viewed as a measure of the traffic burstiness we allow. The larger T is, the larger the flexibility in assigning rates to sessions. To see this, consider a network where each link has capacity C, and a session $\mathcal{S}$ has the (n, T)-smoothness property throughout the network. Session $\mathcal{S}$ will have an average rate of at most $R = nC/T$, and since capacity can only be allocated in discrete multiples of C/T, T determines the flexibility in assigning the rates.

We let N be the number of incoming (and outgoing) links of a node, and $n_{i,j}$, be the number of packets that arrive during a frame over incoming link i and have to be transmitted on the same outgoing frame of link j. Assuming unicast communication, we always have

$$\sum_{j=1}^{N} n_{i,j} \leq T, \quad \text{for all } i \in \{1, 2, \ldots, N\}. \tag{1}$$

If the connection and flow control protocols used guarantee that the number of packets requiring the same outgoing link in a frame is less than or equal to the frame size T, i.e.

$$\sum_{i=1}^{N} n_{i,j} \leq T, \quad \text{for all } j \in \{1, 2, \ldots, N\} \tag{2}$$

then all of the incoming packets can be assigned slots in the required outgoing links such that no packets will have to be dropped. Reservation based protocols, such as either *wait-for-reservation* (WR) protocols or *tell-and-go reservation* (TGR) protocols can be used to ensure lossless communication. (For a detailed discussion and comparison of WR and TG protocols,

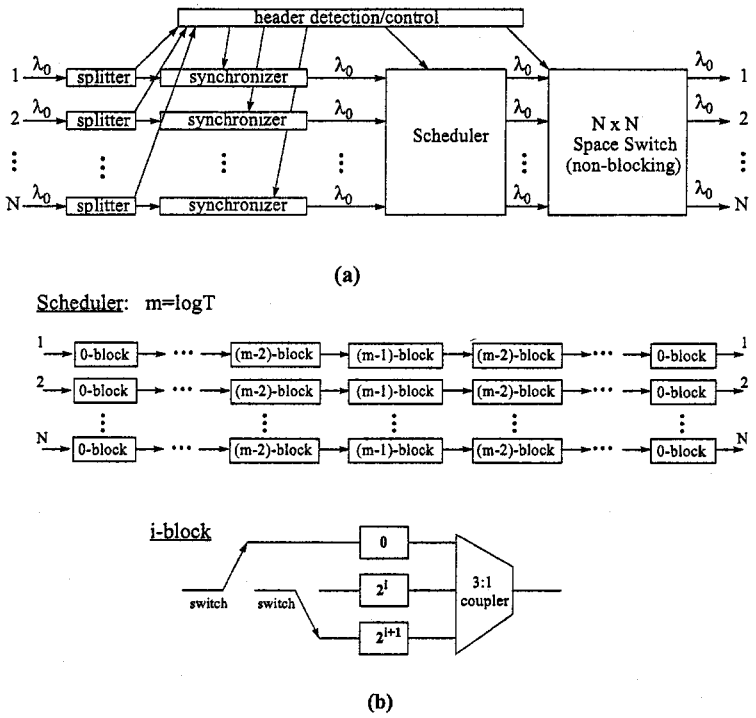

Fig. 2. (a) The scheduling switch architecture operates using a single wavelength λ_0, and is composed of two main blocks: the scheduler rearranges the incoming packets so that the packets appearing during the same slot at the outputs of the scheduler require different outgoing links, and the space switch is an $N \times N$ nonblocking switch. (b) The scheduler is implemented used N parallel branches, one for each input, where each branch consists of $2 \log T - 1$ delay blocks.

see [16].) In practice, we expect the frame size T to be considerably larger than the number of inputs N. For example, if we have 16 Gb/s links and we want the basic rate of the sessions that are switched to be 16 Mb/s, T will have to equal 1024.

III. The Scheduling Switch

In this section, we describe the scheduling switch architecture, first proposed in [13], which is used as the basic structure for the λ-scheduler. The scheduling switch was designed to operate as a single-wavelength system (i.e., WDM was not considered), and is composed of a scheduler followed by an $N \times N$ nonblocking space switch as shown in Fig. 2. At each input, a splitter is used for header detection, and packets are synchronized to a local clock so that the scheduling and switching operate synchronously. The purpose of the scheduler is to rearrange the incoming packets so that packets appearing during the same slot at the outputs of the scheduler require different outgoing links of the space switch. If this property is satisfied by the scheduler, then the space switch will be able to route each packet to its desired outgoing link without any collisions or internal buffering.

An important data structure that is useful to describe the operation of the scheduler is the *frame matrix*, defined as the $N \times N$ matrix $\mathcal{N} = \{n_{i,j}\}$, whose (i, j)th component is equal to the number $n_{i,j}, i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, N$, of packets that arrive during a given frame $F(i)$ of incoming link i and require the same frame $F(j)$ of outgoing link j.

Definition 1: A *permutation matrix* $\mathcal{P}$ is any matrix with entries equal to "0" or "1" with the property that each line of the matrix has at most one nonzero entry.

The following lemma is found in [13], and also used in [17] in a different context.

Lemma 1: A frame matrix $\mathcal{N}$ can be written as the sum

$$\mathcal{N} = \sum_{s=1}^{T} \mathcal{P}_s$$

of at most T permutation matrices, when condition (1) and (2) are satisfied.

Matrix $\mathcal{P}_s$ is used to determine the packet (if any) that will appear during slot s at each of the outputs of the scheduler. In particular, if the (i, j)th entry of the matrix $\mathcal{P}_s$ is equal to one, then a packet arriving over link i and departing over link j is assigned to outgoing slot s of the scheduler. Since $\mathcal{P}_s$ is a permutation matrix, this assignment guarantees that no packets arriving over the same incoming link or requesting the same outgoing link of the switch appear during the same outgoing slot s of the scheduler. The first property ensures that there will be no collisions at the output of the scheduler, while the second property ensures that there will be no collisions at the outputs of the space switch. Since the decomposition of the frame matrix into the T permutation matrices is not unique, we have the freedom to choose the order in which the packets are assigned to the outgoing slots of the scheduler. In particular, this assignment can be chosen to preserve the order of packets arriving over the same input and requesting the same output.

The scheduler is composed of N parallel branches, one for each input, where each branch delays the packets arriving on an incoming link until their assigned time slots are available on their desired outgoing links. This is equivalent to a time-slot interchanger and is implemented using $2 \log T - 1$ three-state

delay blocks, where we have assumed T is a power of 2. The ith delay block can be implemented using two 2-state switches, one $3:1$ coupler, and fiber delay lines corresponding to delays of 2^i and 2^{i+1} packet slots [see Fig. 2(b)]. To ensure that the packets in the incoming frame can be assigned to any slot in the outgoing frame, the outgoing frame must start after the incoming frame ends (or equivalently, at least T packet slots after the incoming frame begins). To ensure that collisions are avoided in the scheduling switch, we will show that the outgoing frame must start at least $(3T)/2-2$ packet slots after the incoming frame begins.

Using the approach of [18], we can expand each branch of delay blocks into an equivalent network using a space-time representation, as shown in Fig. 3. The problem of scheduling packets through a branch of delay blocks to avoid collisions then becomes a problem of routing a permutation between inputs and outputs in the equivalent network, where nonoverlapping paths in the network correspond to collision-free transmission through the delay blocks. Fig. 3 illustrates the case $T=4$ (the incoming frame corresponds to time slots 0–3, and the outgoing frame corresponds to time slots 4–7). Each row in the equivalent network is a replication of the branch of delay blocks for a unit of time, where time increases from top to bottom. Each line, either solid or dashed, represents a feasible state transition in the space-time domain; movement along a horizontal line between nodes corresponds to passing through a delay block without being delayed, while movement along a diagonal line between nodes corresponds to passing through a delay block and being delayed. For example, looking at Fig. 3, a packet q_1 entering the first delay block at time 1 and passing through the fiber delay line of delay 2 corresponds to a transition along the dotted line from input 1 (node 1 at stage 1) to node 3 at stage 2 (corresponding to time slot 3). Similarly, a packet q_2 entering the first delay block at time 2 and passing through the fiber delay line of delay 1 corresponds to a transition along the solid line from input 2 (node 2 at stage 1) to node 3 at stage 2. In this example, both q_1 and q_2 will arrive at the output of the first delay block at the same time, resulting in a packet collision; this can be seen in the equivalent network since both paths share a common node.

Packet collisions can be avoided within a branch of delay blocks if the paths followed by the packets are node disjoint. This can be realized if the equivalent network is *nonblocking*, where every interconnection pattern between the inputs and the outputs can be achieved, and multiple connections do not share common nodes. In the space-time structure of Fig. 3, we have limited the transitions between nodes to a subset of transitions: the solid lines correspond to delay-block states that may be used, while the dashed lines correspond to delay-block states that will not be used. The resulting structure of solid lines in the space-time representation follows the Beneš structure, and is therefore rearrangeably nonblocking [19]. This means that it is possible to schedule the packets in a collision-free manner through a branch of delay blocks using only the states corresponding to the solid lines in the equivalent network. A result of limiting the delay-block states in the equivalent network so that it follows a Beneš structure is that the outgoing frame must start at least $(3T)/2-2$ packet slots after the incoming frame begins. To see this, note that the path that includes the uppermost

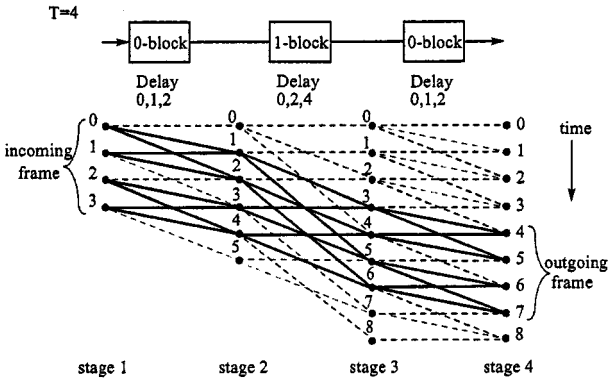

Fig. 3. We expand a branch of delay blocks using a space-time representation when $T=4$. The incoming frame corresponds to time slots 0–3, and the outgoing frame corresponds to time slots 4–7. Each row corresponds to a replication of the branch of delay blocks for a unit of time, where time increases from top to bottom. The solid lines correspond to delay-block states that may be used, while the dashed lines correspond to delay-block states that do not have to be used. The resulting structure of solid lines is a Beneš structure, and the minimum and maximum delay through the structure is $T/2-1$ and $(5T)/2-3$, respectively.

solid line in each stage of the Beneš structure connects the first slot on the incoming frame to the first slot of the outgoing frame. This path corresponds to a delay of 2^i packet slots through each of the i-blocks, where $i \in 0, 1, \ldots, \log T-1$. Summing all such delays through a branch of delay blocks gives a total delay equal to $(3T)/2-2$ packet slots. Therefore a packet arriving on link p in slot i, $i=0, 1, \ldots, T-1$, of an incoming frame, and departing on an outgoing link in slot j, $j=0, 1, \ldots, T-1$, of the outgoing frame will have to be delayed by

$$d_{i,j} = \frac{3T}{2} - 2 + j - i \tag{3}$$

packet slots, where $d_{i,j}$ takes on integer values ranging from $T/2-1$ (corresponding to $i=T-1$ and $j=0$) to $(5T)/2-3$ (corresponding to $i=0$ and $j=T-1$).

The scheduling switch is a single-wavelength system that does not use WDM. In the next section, we present the λ-scheduler, which is a scheduling architecture that uses WDM internally to reduce hardware complexity and improve the signal characteristics.

IV. The λ-Scheduler

In this section, we describe the λ-scheduler, and show how we use the wavelength flexibility of WDM to fold the scheduling switch architecture by collapsing the N parallel branches of delay blocks into fewer branches (or even a single branch) and/or by compressing each branch of delay blocks to reduce the number of delay blocks per branch. Collapsing several parallel branches to a single physical branch is done by wavelength multiplexing packets from multiple inputs onto a single fiber where each input uses a different wavelength. Compressing each branch of delay stages reduces the number of delay blocks per branch by using the internal wavelengths to realize different groups of delays. As we will see, folding the scheduling switch architecture in the space and time domains (by collapsing and

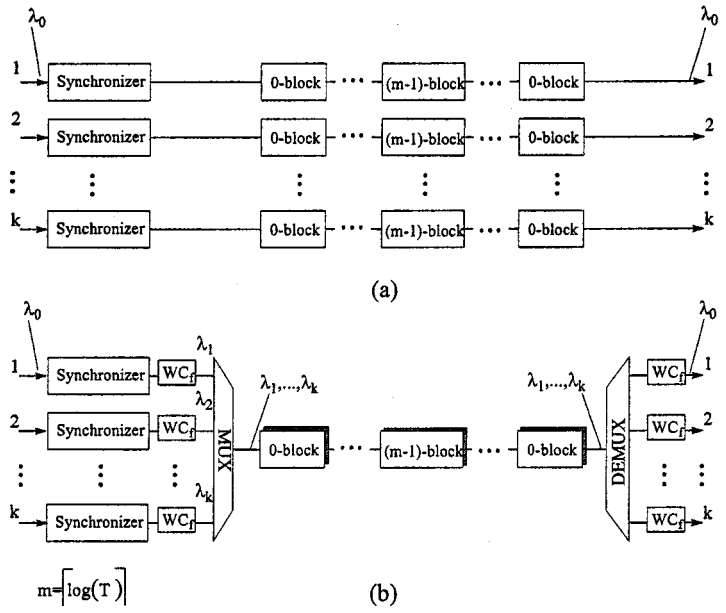

Fig. 4. (a) The delay branches implemented as k parallel branches. (b) WDM has been used to collapse the k parallel branches to single common branch using k wavelengths $\lambda_1, \lambda_2, \ldots, \lambda_k$. The number of delay blocks per branch remains the same as the single-wavelength system in (a), however, the simple two-state switches of each delay block are now implemented using multiwavelength selectable devices.

compressing the delay branches) reduces the number of components required within the switch and can improve the signal characteristics, while preserving the original desirable characteristics of the architecture. To achieve this improvement, however, the delay blocks in each branch of the λ-scheduler will now have to be wavelength selective. In Section IV-A, we show how k internal wavelengths can be used to collapse k parallel branches of delay blocks to a single common branch, and in Section IV-B, we show how k internal wavelengths can be used to compress each branch of delay blocks to reduce the number of delay blocks per branch. In Section IV-C, we present a general implementation of the λ-scheduler with k internal wavelengths, where w, $w \leq k$, internal wavelengths are used to collapse parallel branches, and the remaining $k - w$ internal wavelengths are used to compress each branch of delay blocks.

A. Collapsing Parallel Branches

We first show how k internal wavelengths $\lambda_1, \lambda_2, \ldots, \lambda_k$ can be used to collapse a group of k parallel branches of delay blocks to a single common branch by assigning a different wavelength to each input (see also Fig. 4). Packets arriving on input i, $i = 1, 2, \ldots, k$, using wavelength λ_0 are first converted to wavelength λ_i using a fixed wavelength converter (WC_f). The packets are then multiplexed onto a single WDM fiber and transmitted along a single physical branch. In the original scheduling switch, the i-blocks in each of the k parallel branches are implemented using two simple 2-state switches that are independently set for each packet, followed by fiber delay lines with delays corresponding to 2^i and 2^{i+1} packet slots, and a $3:1$ coupler. To obtain the same functionality using a single common branch and WDM, the simple two-state switches of each delay block must now be implemented using multiwavelength selectable devices (e.g., FTGRs, to be discussed in Section V) that can simultaneously switch packets on multiple wavelengths to different outputs, independently of each other. In other words, the ith delay block of Fig. 4(b) will introduce a delay of 0, 2^i, or 2^{i+1} packet slots depending on the wavelength on which each packet is carried and the desired delay for the packet. Once the packets have passed through all of the delay blocks on the single common fiber, they are demultiplexed back onto k parallel branches so that they can be switched in the space switch. Note, however, that if the demultiplexer is dynamic (in other words, if packets using different wavelengths can be arbitrarily routed to different outputs), then the dynamic demultiplexer can replace the space switch altogether.

B. Compressing Delay Branches

We now show how k internal wavelengths $\lambda_1, \lambda_2, \ldots, \lambda_k$ can be used to compress each branch of delay blocks, which reduces the number of delay blocks per branch by $2\lceil \log k \rceil$, where $\lceil x \rceil$ is the smallest integer greater than or equal to x (see also Fig. 5). The idea here is that each wavelength will be responsible for a smaller range of delays, which can be realized using fewer delay blocks. To see how this is done, we focus on an input link p and define the corresponding $T \times T$ *delay matrix* $\mathcal{D} = \{d_{i,j}\}$, where $d_{i,j}$ is the delay in packet slots that is required for a packet arriving on input link p in slot i, $i = 0, 1, \ldots, T-1$, of an incoming frame, and departing on an outgoing link in slot j, $j = 0, 1, \ldots, T-1$, of the outgoing frame, and is given by (3). [Recall from Section III that $d_{i,j}$ can take on $2T-1$ different integer values ranging from $T/2-1$ to $(5T)/2-3$, and it is these values of $d_{i,j}$ that must be realized using the delay blocks.] Therefore, to reduce the number of delay blocks, the range of delays that have to be realized for each wavelength must be reduced.

To compress each branch of delay blocks, we map each element $d_{i,j}$ of $\mathcal{D}$ to a wavelength λ_q, $q = 1, 2, \ldots, k$, such that $q = 1 + (\lfloor i/(T/k) \rfloor - \lfloor j/(T/k) \rfloor) \bmod k$, where $\lfloor x \rfloor$ is

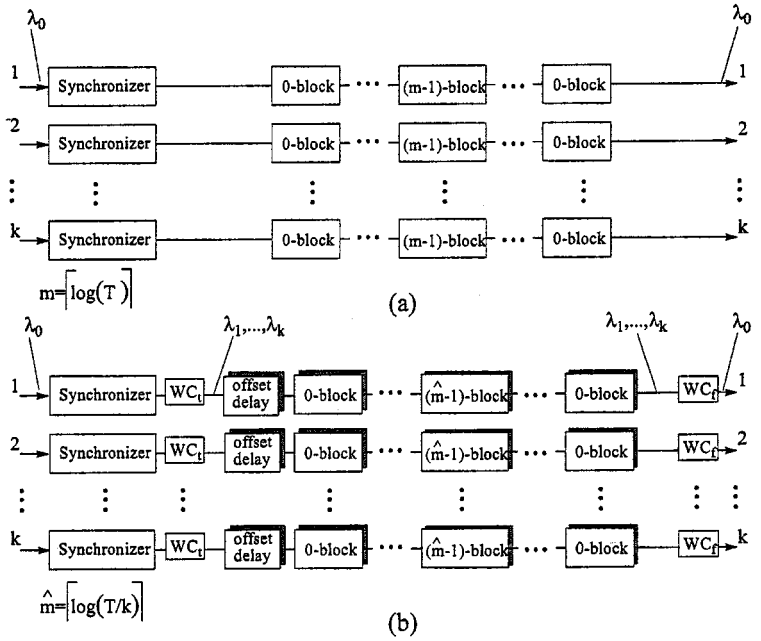

Fig. 5. (a) The delay branches of the scheduling switch implemented as k parallel branches. (b) WDM has been used to compress each branch of delays and reduce the total number of delay blocks per branch by $2\lceil \log k \rceil$. As in Fig. 4, the simple two-state switches of each delay block are now implemented using multiwavelength selectable devices (e.g., FTGRs).

the largest integer less than or equal to x, and we have assumed that T is a multiple of k (in practice, T and k will be powers of 2). This is equivalent to partitioning matrix $\mathcal{D}$ into k^2 smaller $(T/k) \times (T/k)$ submatrices S_m, $m = 1, 2, \ldots, k^2$, (indexed from left to right and top to bottom, as shown in Fig. 7) such that the elements of each submatrix are mapped to a common wavelength, and the elements of adjacent submatrices are mapped to different wavelengths. Note that the values of $d_{i,j}$ in the submatrices with elements that are mapped to wavelength λ_q, $q = 1, 2, \ldots, k$, take integer values in the range

$$\frac{3T}{2} - \frac{qT}{k} - 1, \ldots, \frac{3T}{2} - \frac{qT}{k} - 3 + \frac{2T}{k}$$
$$\text{for } \left\lfloor \frac{i}{(T/k)} \right\rfloor \geq \left\lfloor \frac{j}{(T/k)} \right\rfloor \text{ and } q = 1, 2, \ldots, k$$

and

$$\frac{5T}{2} - \frac{qT}{k} - 1, \ldots, \frac{5T}{2} - \frac{qT}{k} - 3 + \frac{2T}{k}$$
$$\text{for } \left\lfloor \frac{i}{(T/k)} \right\rfloor < \left\lfloor \frac{j}{(T/k)} \right\rfloor \text{ and } q = 2, 3, \ldots, k$$

and the number of different values that the $d_{i,j}$'s can take in each submatrix is $2(T/k) - 1$. Note also that the submatrices along the diagonals correspond to the same wavelength and have the same values for their elements, and two submatrices that correspond to the same wavelength but are not along the same diagonal have elements that differ by T.

Since the range of different values that the $d_{i,j}$'s can take in each delay submatrix is $2(T/k) - 1$, the delays of each submatrix can be implemented without collisions using a single branch consisting of $2\log(T/k) - 1$ delay blocks. (This can be seen by expanding the compressed branch of delay blocks using a space-time representation that realizes a Beneš structure as discussed in Section III.) The delays for submatrices corresponding to the same wavelength can be implemented using a common branch of delay blocks because of the nonblocking structure of each branch. The delays for submatrices corresponding to different wavelengths can also be implemented using a common branch of delay blocks, provided that multiwavelength selective devices are used for switching. For example, the fast tunable grating reflector (FTGR) that is discussed in Section V is a two-state switch that can independently switch signals (packets) on multiple wavelengths to either output port. Therefore, the two-state switches in the i-blocks of Fig. 2(b) can be replaced with FTGR's as shown in Fig. 6(a). Even though the number of values that the d_{ij}'s can take on in each submatrix is the same, an additional *offset* delay is required for each submatrix to account for the different values of the $d_{i,j}$'s. The offset delay can be separated into a fixed delay and a wavelength dependent delay [see Fig. 6(b)]. The fixed delay consists of a delay of $T/2(1-(1/k))$ packet slots to account for the difference in minimum delays between a branch of $2\lceil \log T \rceil - 1$ delay blocks and a branch of $2\lceil \log(T/k) \rceil - 1$ delay blocks, and a delay of T packet slots to account for the difference in delays between submatrices corresponding to the same wavelength that are not along the same diagonal. The wavelength dependent delay of $(k-q)(T/k)$ packet slots accounts for the difference in delays between submatrices corresponding to different wavelengths λ_q, $q = 1, 2, \ldots, k$. The scheduling algorithm of Section III can be used in conjunction with the above wavelength assignment to switch packets to the outgoing slots without any packet collisions within the switch or at the switch outputs.

In Fig. 7, we show the delay matrix $\mathcal{D}$, and how it can be partitioned into k^2 submatrices when $T = 8$ and $k = 1$, 2, and 4. We also show the wavelength mapping of each submatrix, and the corresponding branch of delay blocks that is used to implement the delays. For example, when $k = 2$, $\mathcal{D}$ is partitioned into submatrices S_1, S_2, S_3, and S_4; the five delay blocks have been

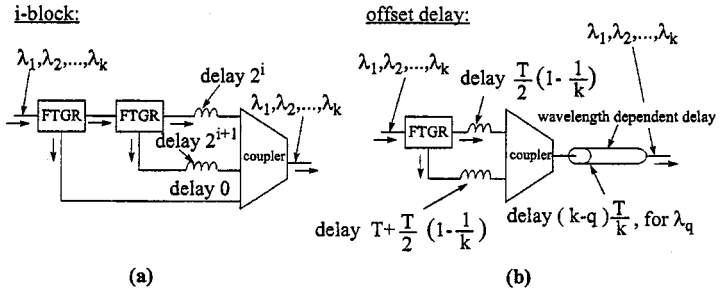

Fig. 6. (a) The i-block is composed of two multiwavelength switches or FTGRs (to be discussed in Section V), fibers corresponding to delays of 2^i and 2^{i+1} packet slots, and a $3:1$ coupler. (b) The offset delay block is composed of one FTGR, two fixed delay lines corresponding to delays of $T(1-1/k)/2$ and $T+T(1-1/k)/2$ packet slots, a $2:1$ coupler, and a wavelength dependent delay that delays packets using wavelength λ_q by $(k-q)T/k$ packet slots.

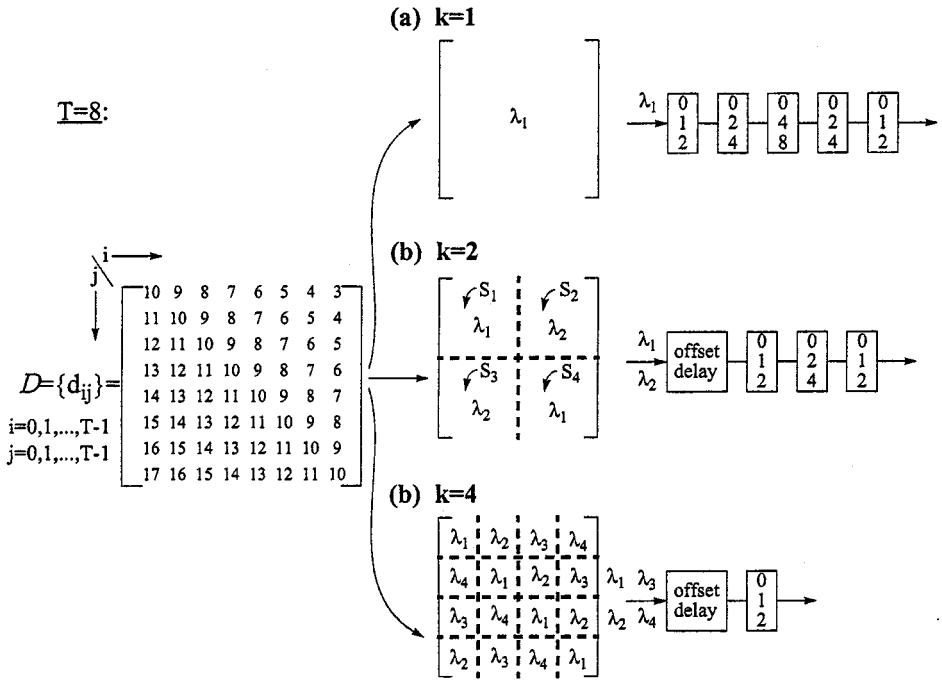

Fig. 7. We focus on an incoming link, and show how the delay matrix $\mathcal{D}$ can be partitioned into k^2 submatrices when the frame size $T=8$ and the number of internal wavelengths used for compression is $k=1,\ 2$, and 4. We also show the wavelength mapping of each submatrix, and the corresponding branch of delay blocks that is used to implement the delays. For $k=2$, the offset delays corresponding to submatrices $S_1,\ S_2,\ S_3$, and S_4 are 6, 2, 10, and 6 packet slots, respectively.

compressed to three delay blocks; and the offset delays corresponding to submatrices S_1, S_2, S_3, and S_4 are 6, 2, 10, and 6 packet slots, respectively.

C. General Implementation of λ-Scheduler

The λ-scheduler is a multiwavelength scheduling switch that is designed to rearrange the incoming packets so that packets appearing during the same slot at the outputs require different outgoing links. The key feature of the λ-scheduler is that the scheduling and space switching are no longer restricted to a single wavelength; WDM is used instead, with a tunable wavelength converter (WC_t) at each switch input, which can convert each packet to a different wavelength, and a fixed wavelength converter (WC_f) at each switch output, which converts packets back to the original wavelength.

Let N be the number of inputs (and outputs) to the switch, k be the number of internal wavelengths (or channels) available, and T be the frame size in packet slots. In Fig. 8, we show a general implementation of the λ-scheduler where w, $w \le k$, internal wavelengths are used to collapse groups of w parallel delay-block branches into single branches, for a total of $\lceil N/w \rceil$ parallel branches. The remaining $k-w$ internal wavelengths are used to compress each branch of delay blocks, reducing the number of delay blocks per branch to $2\hat{m}-1$, where $\hat{m} = \lceil \log(T/\hat{k}) \rceil$ and $\hat{k} = \lfloor k/w \rfloor - 1$ is the number of wavelengths that are used to compress each channel of each branch of delay blocks. For example, if $N=4$, $T=8$, and $k=6$, then $w=2$ wavelengths can be used to collapse the 4 parallel branches into two parallel branches. Furthermore, the remaining $k-w = 4$ wavelengths can be used within each branch to reduce the number of delay blocks per branch from five to three and to ensure packet collisions are avoided.

The ith delay block (or i-block) must now simultaneously switch packets using different wavelengths to one of three delay paths (corresponding to delays of 0, 2^i, and 2^{i+1} packet slots),

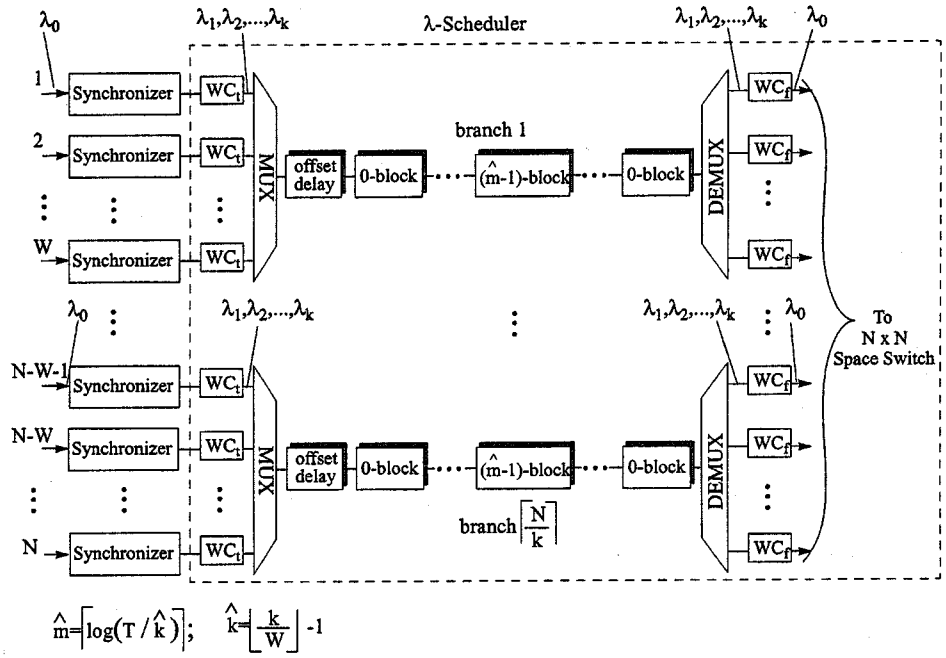

Fig. 8. The λ-scheduler uses WDM with a tunable wavelength converter (WC_t) at each switch input and a fixed wavelength converter (WC_f) at each switch output. We show an implementation with k internal wavelengths, where w, $w \leq k$, wavelengths are used to collapse parallel branches, and the remaining $k - w$ wavelengths are used to compress each branch of delay stages.

and is composed of two multiwavelength switches (FTGR's), fibers corresponding to delays of 2^i and 2^{i+1} packet slots, and one 3 : 1 coupler [see Fig. 6(a)]. An additional offset delay block [see Fig. 6(b)] is required so that the compressed branches of delay blocks can provide the same total delays as the original delay branches, as explained in Section IV-B. One possible implementation of the wavelength dependent delay is to use a highly dispersive fiber, which delays signals (packets) on different wavelengths by different (fixed) amounts. Another possible implementation could use a recirculator followed by a series of fiber delays with Bragg gratings tuned to different wavelengths interspersed between the delays. A wavelength multiplexer (mux) and demultiplexer (demux) are required to collapse the parallel branches. As we will see in Section VI, if $k = N$, that is if all available wavelengths are used to collapse the parallel branches, the $N \times N$ crossbar switch can be replaced by a dynamic wavelength router (dynamic demux) that can simultaneously route multiple WDM signals (packets) to different outputs.

A packet p that arrives to the switch on wavelength λ_0 during a given frame $F(i)$ of incoming link i is assigned an outgoing slot s of its desired outgoing link according to the algorithm given in Section III. Packet p is then assigned a wavelength λ_q based on the input link upon which it arrived and the time delay between the slot in which it arrived and the assigned outgoing slot s. The packet is then multiplexed onto a delay branch and routed through the delay blocks according to the required delay. In the ith delay block, the packet is switched using multiwavelength switches (e.g., FTGR's) to one of three delay paths, incurring a delay of either 0, 2^i, or 2^{i+1} packet slots. Once the packet p passes through the delay blocks, it is demultiplexed to an outgoing branch, and then converted back to the original wavelength λ_0 before entering the space switch. (It is also possible to wait until after passing through the space switch before converting back to the original wavelength so that WDM can also be used in the space switch implementation.)

V. FTGR

In this section, we describe a new multiwavelength component called the FTGR, which is used to implement the i-block and offset delays for the λ-scheduler [see Fig. 6(a) and (b), respectively]. Each WDM packet that arrives at the input of an i-block must be individually switched to one of three delay paths; similarly in the offset-delay block, each WDM packet must be individually switched to one of two delay fibers before passing through the wavelength dependent delay. The FTGR is functionally similar to the N-type building block of [20] (or to an acoustooptic tunable filter (AOTF) [21] with only one active input port), but is intended to operate at ns-switching speeds with a flat-top passband and a 3-dB bandwidth matched to the channel grid, and can be used to switch individual packets rather than slower circuit switched connections. Our design is based on recent work by Ozturk *et al.* [22], where a polymer semiconductor hybrid approach was used to fabricate a high speed tunable filter in the context of an add/drop multiplexer.

The FTGR is composed of a recirculator followed by a series of finely spaced tunable gratings (see Fig. 9). WDM signals that arrive to the FTGR are individually routed to one of the two outgoing ports by electrically tuning the gratings to either pass or reflect the signals on the corresponding wavelengths. For a system that is designed to operate using k wavelengths, k tunable gratings will be required (one for each wavelength), where

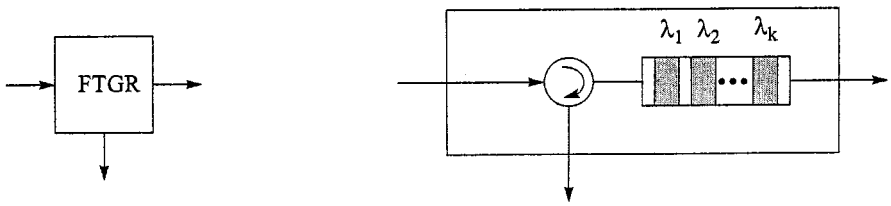

Fig. 9. The FTGR is a multiwavelength device that can pass or drop signals on multiple wavelengths independently. The FTGR is composed of a recirculator followed by a series of tunable gratings that can be electronically tuned at ns packet rates to either pass or reflect the signals.

the ith grating, $i = 1, 2, \ldots, k$, can be tuned either to wavelength λ_i or to a common *rest* wavelength λ_r, where $\lambda_r \neq \lambda_i$ for all i. This enables the FTGR to individually route signals arriving on k wavelengths simultaneously, and requires a total bandwidth equal to $k + 1$ channels. We use the FTGR in the λ-scheduler as a multiwavelength two-state switch that simultaneously switches multiple wavelengths independently.

VI. Cost Analysis

We now present a basic cost analysis and compare the costs of both implementations of the λ-scheduler. Our purpose here is to study the tradeoffs between using WDM internally to collapse parallel branches or to compress each branch of delay stages. For our analysis, we have chosen two measures of cost: the total number of elementary components that are required and the signal attenuation (or insertion loss) through the switch.

Recall that N is the number of inputs (and outputs) to the switch and T is the frame size in packet slots, and we assume that $k = N$ is the number of internal wavelengths available. (In the following analysis, we will assume T is a multiple of N, and both T and N are powers of 2.) We will analyze the cost of the two implementations of the λ-scheduler, using the scheduling switch as a reference (see Fig. 10). In the *compressed λ-scheduler* [see Fig. 10(b)], WDM has been used internally to compress each of the N branches of delay blocks, while in the *collapsed λ-scheduler* [see Fig. 10(c)], WDM has been used internally to collapse all of the N parallel branches of the scheduler to a single common branch. In all three switch designs, we have assumed that the synchronizer is implemented using wavelength conversion followed by a dispersive optical delay line as discussed in [23], the space switch (if required) is an $N \times N$ crossbar switch, and to simplify the figures, we have removed the splitter and header detection/control blocks, which are common to all three designs.

A. Component Cost

We first examine the number of elementary switching components, the number of wavelength converters, and the total fiber required to implement the delays (see also Table I).

The original scheduling switch architecture is composed of a synchronizer and a scheduler, followed by an $N \times N$ crossbar switch, where the synchronizer and scheduler are implemented using N parallel branches, one for each input. Each of the N parallel branches requires one tunable wavelength converter (WC_t), one fixed wavelength converter (WC_f), and $2\log T - 1$ delay blocks, where each delay block consists of two two-state switches and one $3:1$ coupler, and the ith delay block, $i = 0, 1, \ldots, \log T - 1$, requires fiber delay lines corresponding to delays of 2^i and 2^{i+1} packet slots. The scheduling switch also requires N^2 simple switches for the $N \times N$ crossbar switch.

In the compressed λ-scheduler, the N wavelengths are used to compress each branch of delay blocks, reducing the number of delay blocks to $2\log(T/N) - 1$ per branch, or equivalently, reducing the total number of delay blocks for all N branches by $2N \log N$. To achieve these reductions using WDM, an additional WC_t and an offset delay block are required for each of the branches, and the multiwavelength FTGR has replaced the simple two-state switch.

In the collapsed λ-scheduler, the N wavelengths are used to collapse the N parallel branches of the scheduling switch to a single common branch by assigning a different wavelength to each of the N inputs. This reduces the total number of delay blocks and the total fiber length by a factor of N, which is the best we could hope for. The $N \times N$ crossbar switch is no longer needed, and the space switching at the output can be done using a dynamic wavelength router (dynamic demux) that is able to arbitrarily route multiple WDM signals (packets) to different outputs. An additional static wavelength router (mux) that is able to route a fixed set of multiple WDM signals (packets) to different outputs is also needed at the input, and can be implemented, for example, using an arrayed wavelength grating (AWG). Since each input is assigned a unique wavelength, wavelength conversion can be done using only one additional WC_f per input. As with the compressed λ-scheduler, the multiwavelength FTGR has replaced the simple two-state switch.

Comparing the component cost of the λ-scheduler implementations, we find that using N internal wavelengths to collapse the parallel branches provides the best component count savings. This is because the number of delay blocks, $3:1$ couplers, and switching elements, as well as the total fiber length used for delays are all reduced by a factor of N (compared to using the N wavelengths to compress the delay stages, which results in only a $2N \log N$ reduction in component count). Furthermore, the $N \times N$ crossbar switch can be replaced by a simple demux, and an offset delay block is not required. The main drawbacks of using internal wavelengths to fold the switch architecture is that the internal wavelengths cannot be used externally in a WDM system (ie, a separate set of wavelengths must be used if the incoming/outgoing links also use WDM) and the simple switches must now be wavelength selective which is potentially more expensive.

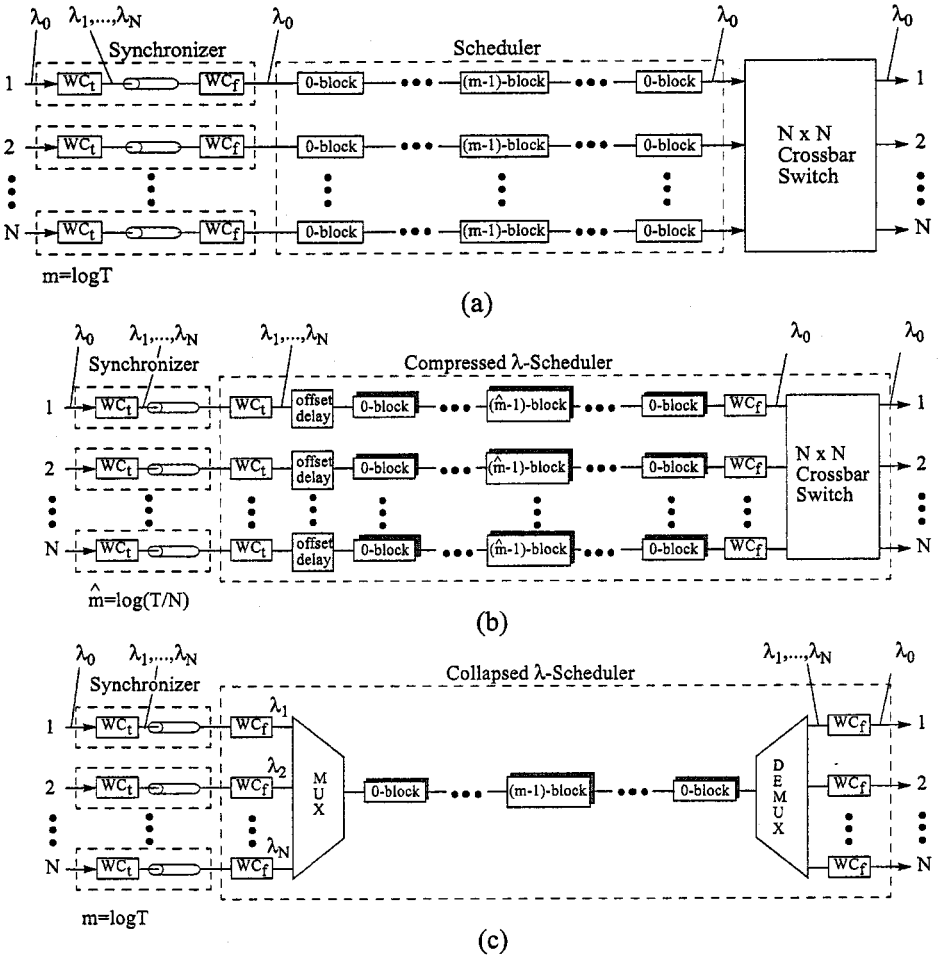

Fig. 10. We illustrate an implementation of the scheduling switch and two versions of the λ-scheduler where we assume the number of wavelengths k is equal to the number of inputs (and outputs) N to the switch. In all three switch designs, the synchronizer is implemented using wavelength conversion followed by a dispersive optical delay line as discussed in [23], the space switch (if required) is an $N \times N$ crossbar switch, and we have removed the splitter and header detection/control blocks, as they are common to all three switch designs. (a) The scheduling switch architecture. (b) The compressed λ-scheduler, where WDM is used to compress each of the N parallel branches of delay blocks. (c) The collapsed λ-scheduler, where WDM is used to collapse all of the N parallel branches to a single common branch. Note that the space switch has been replaced by a dynamic demultiplexer (as described in the text).

B. Loss Analysis

We now study the attenuation cost, or system insertion loss, for both implementations of the λ-scheduler. Since the insertion loss is different for different paths through the switch, we will examine the maximum insertion loss $L_{\max}$ for the path with the greatest loss, the minimum insertion loss $L_{\min}$ for the path with the smallest loss, and the differential loss $\Delta L = L_{\max} - L_{\min}$ between the paths with the greatest and smallest loss. The maximum insertion loss is important since the worst-case signal attenuation will limit the total number of switches that can be used by a packet. The differential loss, however, is often more important since it increases the dynamic range of the signals in the network. We will concentrate our analysis on the scheduling and switching sections of the architectures, and ignore the header detection and synchronization stages, as they are common to all three designs. It is important to note, however, that since we assume the fine synchronization is implemented using SOA-WCs, the synchronization stage may be used to amplify and reshape the signal, reducing the overall system insertion loss [24]. In our analysis, we assume the fiber attenuation is negligible, and the loss for each component includes fiber coupling losses when appropriate.

The scheduling switch is composed of N parallel branches of delay blocks, one for each input, followed by an $N \times N$ crossbar switch. Each incoming signal will pass through one branch consisting of $2\log T - 1$ delay blocks. The path through the delay blocks which has the minimum loss is the path corresponding to a delay of 0, and any path that goes through a fiber delay line at each delay block will have the maximum loss. Therefore, the system insertion loss parameters for the scheduling switch are as follows:

$$L_{\max} = (2\log T - 1)(L_{3C} + 2L_{2S}) + L_{NX}(\max)$$
$$L_{\min} = (2\log T - 1)(L_{3C} + L_{2S}) + L_{NX}(\min)$$
$$\Delta L = (2\log T - 1)L_{2S} + \Delta L_{NX}$$

where L_{3C} and L_{2S} represent the insertion loss through the $3:1$ coupler and the two-state switch, respectively, and $L_{NX}(\min)$, $L_{NX}(\max)$, and $\Delta L_{NX} = L_{NX}(\max) - L_{NX}(\min)$ represent the maximum, minimum, and differential insertion loss through the $N \times N$

TABLE I
ELEMENTARY COMPONENT COST COMPARISON BETWEEN THE SCHEDULING SWITCH, THE COMPRESSED λ-SCHEDULER, AND THE COLLAPSED λ-SCHEDULER WHERE N IS THE NUMBER OF INPUT/OUTPUTS TO THE SWITCH, T IS THE FRAME SIZE IN PACKET SLOTS, AND $k = N$ IS THE NUMBER OF INTERNAL WAVELENGTHS AVAILABLE. THE COST MEASURE FOR FIBER LENGTH IS GIVEN IN TERMS OF PACKET DURATION, AND THE OTHER COST MEASURES ARE GIVEN IN COMPONENT COUNTS.

Components	Scheduler	Compressed λ-Scheduler	Collapsed λ-Scheduler
Delay Blocks	$2N\log T - N$	$2N\log(T/N) - N$	$2\log T - 1$
Offset Blocks	—	N	—
Fiber Length (for delays)	$(9/2)NT - 6N$	$(13/2)T - 6N - T/N$	$(9/2)T - 6$
2-state switches	$4N\log T - 2N$	—	—
FTGR	—	$4N\log(T/N) - 2N + 1$	$4\log T - 2$
3:1 Coupler	$2N\log T - N$	$2N\log(T/N) - N$	$2\log T - 1$
2:1 Coupler	—	N	—
WC_t	N	2N	N
WC_f	N	N	2N
mux/demux	—	—	1
$N \times N$ crossbar	1	1	—

crossbar switch, respectively. If we assume the $3:1$ coupler is implemented using a passive $4:1$ coupler with one idle input, the loss can be estimated as $L_{3C} \approx -6$ dB. Furthermore, if we assume the two-state switches are $LiNbO_3$ electrooptic switches, the loss per switch can be estimated as $L_{2S} \approx -4$ dB, giving a maximum and minimum loss per delay block of -14 dB and -10 dB, respectively. If we integrate the two two-state switches of each delay block onto a single chip, the coupling losses between the switches can be eliminated, resulting in a 0 dB differential loss per delay block. For the crossbar switch, we use the design proposed in [25] which uses optically amplified suppressed modal interference (OASIS) switch elements at the crosspoints. Each OASIS switch element can either be in the *bar* state where signals pass straight through the crosspoint, or in the *cross* state where signals make a turn at the crosspoint. Estimates for the loss of the OASIS elements in the bar state are around -0.5 dB, while ideally there should be a 2–3 dB gain in the cross state. An additional 1–2 dB loss is also seen due to fiber coupling to the crossbar switch inputs and outputs. Therefore, the insertion loss through the crossbar switch is a function of the path through the switch, and $L_{NX}(\max) \approx (2N-1)(-0.5\text{ dB}) + 2\text{ dB} - 4\text{ dB}$, $L_{NX}(\min) \approx 2\text{ dB} - 4\text{ dB}$, and $\Delta L_{NX} \approx (2N-1)(-0.5\text{ dB})$.

For the compressed λ-scheduler, the N wavelengths are used to compress each branch of series delay blocks, reducing the total number of delay blocks per branch by $2\log N$. The system insertion loss parameters for the compressed λ-scheduler are

$$\begin{aligned} L_{\max} &= (2\log(T/N) - 1)(L_{3C} + 2L_{\text{FTGR}}) \\ &\quad + L_{\text{offset}} + L_{NX}(\max) \\ L_{\min} &= (2\log(T/N) - 1)(L_{3C} + L_{\text{FTGR}}) \\ &\quad + L_{\text{offset}} + L_{NX}(\min) \\ \Delta L &= (2\log(T/N) - 1)L_{\text{FTGR}} + \Delta L_{NX} \end{aligned}$$

where $L_{\text{offset}} = L_{2C} + L_{\text{FTGR}}$, and L_{2C} and L_{FTGR} represent the loss through the $2:1$ coupler and FTGR, respectively. In determining the loss for the FTGR, we have assumed that the insertion loss is the same for both output ports; preliminary results suggest this is true, and that a loss of -4 dB through the device is achievable. Assuming passive couplers are used, the maximum and minimum loss per delay block can be estimated as -14 dB and -10 dB, respectively, and the loss through the offset delay block can be estimated as -7 dB.

The collapsed λ-scheduler uses the N wavelengths to collapse the parallel branches, but does not reduce the number of delay blocks per branch. Therefore, the maximum and minimum loss through the delay blocks will be identical to that of the scheduling switch, except that L_{2S} is replaced by L_{FTGR}. The collapsed λ-scheduler also has an additional mux, and the crossbar switch is replaced by a dynamic demux. The system insertion loss parameters for the collapsed λ-scheduler are

$$\begin{aligned} L_{\max} &= (2\log T - 1)(L_{3C} + 2L_{\text{FTGR}}) + L_{\text{mux}} + L_{\text{demux}} \\ L_{\min} &= (2\log T - 1)(L_{3C} + L_{\text{FTGR}}) + L_{\text{mux}} + L_{\text{demux}} \\ \Delta L &= (2\log T - 1)L_{\text{FTGR}} \end{aligned}$$

where L_{mux} and L_{demux} represent the loss through the mux and demux, respectively. If we assume the mux is implemented using an arrayed waveguide grating (AWG), then $L_{\text{mux}} \approx -9$ dB. If the demux is used to replace the crossbar switch, then the demux cannot be built using a simple AWG; instead, it must be a dynamic wavelength router that is able to simultaneously route multiple WDM signals (packets) to different outputs. A simple (yet limited) design could be a passive $1:N$ splitter followed by an FTGR on each branch; this is limited by N, since the loss for the splitter goes as $-10\log_{10} N$, and an alternate design would be required for large N. The maximum and minimum loss per delay is the same as with the compressed λ-scheduler, and the

TABLE II
WE COMPARE THE LOSS PARAMETERS FOR THE SCHEDULING (DELAY BLOCK) STAGES OF THE SCHEDULING SWITCH, THE COMPRESSED λ-SCHEDULER, AND THE COLLAPSED λ-SCHEDULER USING THE NOMINAL VALUES FOR COMPONENT LOSS GIVEN IN THE TEXT. THE LOSSES FOR THE mux, $demux$, AND $N \times N$ CROSSBAR SWITCH ARE NOT IN THE TABLE

Loss (dB)	Scheduler	Compressed λ-Scheduler	Collapsed λ-Scheduler
L_{max}	$(2\log T - 1)(-14)$	$(2\log(T/N) - 1)(-14) - 7$	$(2\log T - 1)(-14)$
L_{min}	$(2\log T - 1)(-10)$	$(2\log(T/N) - 1)(-10) - 7$	$(2\log T - 1)(-10)$
ΔL	$(2\log T - 1)(-4)$	$(2\log(T/N) - 1)(-4)$	$(2\log T - 1)(-4)$

TABLE III
WE COMPARE THE λ-SCHEDULER WITH A NUMBER OF OTHER ALMOST ALL-OPTICAL SWITCH ARCHITECTURES WHEN THE NUMBER w OF INTERNAL WAVELENGTHS IS EQUAL TO THE NUMBER N OF INPUT (AND OUTPUTS) TO THE SWITCH, T IS THE BURST PARAMETER, AND b IS THE NUMBER OF INTERNAL WAVELENGTHS IS EQUAL TO THE NUMBER N OF INPUT (AND OUTPUTS) TO THE SWITCH, T IS THE BURST PARAMETER, AND b IS THE NUMBER OF PARALLEL DELAY BRANCHES IN THE SINGLE-STAGE SWITCH ARCHITECTURES

Architecture	Lossless	Bursty	Modular	WDM	Elementary Components
Staggering [8]	No	No	No	No	$N\log b + b\log N$
SMOP [26]	No	No	No	No	$(N+b)^2$
BSS [9]	No	No	No	Yes	$bN + N^2$ $(b + 2N + 1)$ $(1 \times N)$-splitters $(N + 1)$ $(1 \times b)$-splitters
OASIS [27]	No	No	No	Yes	$N\log b$ b $(1 \times N)$-splitters
SLOB [12]	No	Yes	Yes	Yes	$kN\log b$ kb $(1 \times N)$-splitters
Packing [13]	Yes	Yes	Yes	No	$N^2\log T + N\log N$
Scheduling [13]	Yes	Yes	Yes	No	$2N\log T + 2N\log N$
Compressed λ-Scheduler	Yes	Yes	Yes	Yes	$4N\log T$ FTGR
Collapsed λ-Scheduler	Yes	Yes	Yes	Yes	$N + 4\log T$ FTGR

loss through the mux and demux can be estimated as -9 dB and $(-4 - 10\log_{10} N)$ dB, respectively.

In Table II, we compare the loss parameters for the scheduling stage of the three switch designs using the nominal values for component loss given above (the losses for the mux, demux, and $N \times N$ crossbar switch are not included). We find that using the N wavelengths to compress each branch of series stages gives the best maximum signal attenuation characteristics for the scheduling stage.

Comparing the loss parameters for the scheduling stage of the two λ-scheduler implementations, we find that using the N wavelengths to compress each branch of series delay stages gives the smallest insertion loss. This result is contrary to the results for the component count of Section VI-A. Comparing the results for the attenuation cost with those of the component count of Section VI-A give contrary results when N is large; to reduce signal attenuation, WDM should be used internally to compress each branch of delay stages, but to reduce the component cost, WDM should be used internally to collapse the parallel branches of delay stages. Depending on the size N of the switch, the size T of the frames, and the number k of internal wavelengths (or channels) available, the λ-scheduler should be designed to take advantage of both cost reductions by using a portion of the wavelengths to compress each branch of delay blocks, and the remaining wavelengths to collapse the parallel branches of delay blocks. As mentioned earlier, the SOA-WCs provide additional gain not considered in our analysis, and optical amplifiers could also be used to compensate for the losses through the delay blocks or in the crossbar switch.

VII. Conclusion

We have proposed a new multiwavelength almost all-optical switch architecture called the λ-scheduler that uses WDM internally to compress each branch of delay stages and/or collapse the number of parallel branches required in the switch. The λ-scheduler preserves the packet order for a given input–output pair, is consistent with virtual circuit switching, and when combined with appropriate connection and flow control protocols, provides lossless communication for traffic that satisfies certain smoothness properties. The λ-scheduler compares favorably with other almost all-optical switch architectures (see also Table III) because it can provide lossless communication for bursty traffic with a hardware cost that grows logarithmically with the burst parameter T; it has a modular design so that it can be easily expanded to accommodate more burstiness in the traffic in a similar way to how electronic buffers can be expanded in conventional switches; and WDM is used internally to share resources and reduce the number of required components. We have compared the cost (as measured by the number of elementary components and the signal attenuation) of two implementations of the λ-scheduler, and studied the tradeoffs between using WDM to compress each branch of delay stages and to collapse the number of parallel branches. When k wavelengths are used to compress each branch of delay blocks, we see a $2N \log k$ total reduction in the number of delay blocks. When k wavelengths are used to collapse the parallel branches of delay blocks, the number of delay blocks and total fiber length are reduced by a factor of k. We are currently studying other cost measures, and investigating implementations of the λ-scheduler for variable-length packets and when the smoothness property of the sessions is violated.

References

[1] M. G. Hluchyi and M. J. Karol, "Queueing in high-performance packet switching," *J. Select. Areas Commun.*, vol. 6, pp. 1587–97, Dec. 1988.

[2] A. Huang and S. Knauer, "Starlight: A wideband digital switch," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM'84)* Atlanta, GA, Nov. 1984, vol. 1, pp. 121–5.

[3] V. Chan, K. L. Hall, E. Modiano, and K. A. Rauschenbach, "Architectures and technologies for high-speed optical data networks," *J. Lightwave Technol.*, vol. 16, pp. 2146–68, Dec. 1998.

[4] D. Hunter and D. Smith, "An architecture for frame integrity optical TDM switching," *J. Lightwave Technol.*, vol. 11, pp. 914–24, May–June 1993.

[5] Z. Haas and D. R. Cheriton, "Blazenet: A packet-switched wide-area network with photonic data path," *IEEE Trans. Commun.*, vol. 38, pp. 818–29, June 1990.

[6] E. A. Varvarigos and V. Sharma, "The ready-to-go virtual circuit protocol: A loss-free protocol for multigigabit networks using FIFO buffers," *IEEE/ACM Trans. Networking*, vol. 5, pp. 705–18, Oct. 1997.

[7] D. J. Blumenthal, K. Y. Chen, J. Ma, F. R. J., and J. R. Sauer, "Demonstration of a deflection routing 2×2 photonic switch for computer interconnects," *IEEE Photon. Technol. Lett.*, vol. 4, pp. 169–73, Feb. 1992.

[8] Z. Haas, "The "Staggering switch:" An electronically controlled optical packet switch," *J. Lightwave Technol.*, vol. 11, pp. 925–36, May/June 1993.

[9] M. Renaud, F. Masetti, C. Guillemot, and B. Bostica, "Network and system concepts for optical packet switching," *IEEE Commun. Mag.*, vol. 35, pp. 96–102, Apr. 1997.

[10] S. L. Danielsen, B. Mikkelsen, C. Joergensen, T. Durhuus, and K. E. Stubkjaer, "WDM packet switch architectures and analysis of the influence of tunable wavelength converters on the performance," *J. Lightwave Technol.*, vol. 15, pp. 219–27, February 1997.

[11] J. Dupraz, "ATM: Current status and perspectives of evolution," in *Proc. European Conf. Optic. Commun. (ECOC'94)*, Firenze, Italy, Sept. 1994, pp. 555–62.

[12] D. Hunter, W. Cornwell, T. Gilfedder, and A. Franzen, *et al.*, "SLOB: A switch with large optical buffers for packet switching," *J. Lightwave Technol.*, vol. 16, pp. 1725–36, Oct. 1998.

[13] E. A. Varvarigos, "The "Packing" and the "Scheduling" packet switch architectures for almost all-optical lossless networks," *J. Lightwave Technol.*, vol. 16, pp. 1757–67, October 1998.

[14] S. J. Golestani, "Congestion-free communication in high-speed packet networks," *IEEE Trans. Commun.*, vol. 39, pp. 1802–12, December 1991.

[15] A. Eckberg, D. Luan, and D. Lucantoni, "An approach to controlling congestion in atm networks," *Proc. Int. J. Dig. Analog Commun. Syst.*, vol. 3, pp. 199–209, Apr.-June 1990.

[16] E. A. Varvarigos and J. P. Lang, "A virtual circuit deflection protocol," *IEEE/ACM Trans. Networking*, vol. 7, pp. 335–49, June 1999.

[17] G. Bongiovanni, D. Coppersmith, and C. W. Wong, "An optimum time slot assignment algorithm for an SS/TDMA system with variable number of transponders," *IEEE Trans. Commun.*, vol. COM-29, pp. 721–6, May 1981.

[18] D. Hunter and D. Smith, "New architectures for optical DM switching," *J. Lightwave Technol.*, vol. 11, pp. 495–511, March 1993.

[19] F. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. San Mateo, CA: Morgan Kaufmann, 1992.

[20] Y.-K. Chen and C.-C. Lee, "Fiber Bragg grating-based large nonblocking multiwave cross-connects," *J. Lightwave Technol.*, vol. 16, pp. 1746–56, October 1998.

[21] D. Smith, R. Chakravarthy, and Z. Bao, *et al.*, "Evolution of the acousto-optic wavelength routing switch," *J. Lightwave Technol.*, vol. 14, pp. 2044–51, September 1996.

[22] C. Ozturk, Y. Byun, S. Sakamoto, and N. Dagli, "Integrated polymer/semiconductor grating reflector for WDM applications," in *Proc. IEEE Lasers Electro-Optics Soc. (LEOS'98)* Orlando, FL, Dec. 1998, vol. 2, pp. 5–6.

[23] J. Lang, "Almost-all optical switching: Architectures, protocols, and analyzes," Ph.D. dissertation, University of California at Santa Barbara, 1999.

[24] D. Wolfson, S. L. Danielsen, H. N. Poulsen, P. B. Hansen, and K. E. Stubkjaer, "Experimental and theoretical investigation of the regenerative capabilities of electrooptic and all-optical interferometric wavelength converters," *IEEE Photon. Technol. Lett.*, vol. 10, pp. 1413–15, Oct. 1998.

[25] G. Fish, B. Mason, L. Coldren, and S. DenBaars, "Compact 4×4 InGaAsP–InP optical crossconnect with a scalable architecture," *IEEE Photon. Technol. Lett.*, vol. 10, pp. 1256–1258, Sept. 1998.

[26] M. Karol, "Shared-memory optical packet (ATM) switch," in *Proc. SPIE* San Diego, CA, July 1993, vol. 2024, pp. 212–22.

[27] J. Gabriagues and J. Jacob, "OASIS: A high-speed photonict ATM switch—Results and perspectives," in *Proc. Fifteenth Int. Switching Symp. (ISS'95)*, Berlin, Germany, Apr. 1995, pp. 457–61.

Jonathan P. Lang (M'89) received the B.S. degree in electrical and computer engineering from the University of California, San Diego, in 1993 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California, Santa Barbara, in 1995 and 1999, respectively.

During the spring and summer of 1995, he worked at Whitetree, Inc., Palo Alto, CA, where he developed rate allocation algorithms for ATM switches. In 1999, he joined Calient Networks, Goleta, CA, as a Systems Engineer, Network Architecture. He is actively involved in the IETF and OIF. His research interests are in all-optical networking, protocol design, and optical switch architectures.

Dr. Lang is a member of the Tau Beta Phi and Eta Kappa Nu honor societies.

Emmanouel A. Varvarigos (M'92) received the Dipl.Eng. degree from the National Technical University of Athens, Greece, in 1988, the M.S. degree in electrical engineering in 1991, and the Ph.D. degree in electrical engineering and computer science in 1992, both from the Massachusetts Institute of Technology, Cambridge.

In 1990, he was with Bell Communications Research, Morristown, NJ. In 1992, he joined the Department of Electrical and Computer Engineering, University of California at Santa Barbara, where he is currently an Associate Professor. In 1998–1999, during his sabbatical leave, he was a Visiting Associate Professor at the Technical University of Delft, The Netherlands, and the University of Patras, Greece. His research interests are in the areas of algorithms and protocols for data networks, performance evaluation, all-optical networks, parallel and distributed computation, and mobile communications.

Dr. Varvarigos received the First Panhellenic Prize in the Greek Mathematic Olympiad in 1982, the Technical Chamber of Greece Award four times (1984–1988), and the National Science Foundation Research Initiation Award in 1993. He was a co-organizer of the 1996 Workshop on Advanced Communication Systems and Networking.

Daniel J. Blumenthal (M'97–SM'99) received the B.S.E.E. degree from the University of Rochester, NY, in 1981, the M.S.E.E. degree from Columbia University, New York, NY, in 1988, and the Ph.D. degree from the University of Colorado, Boulder, in 1993.

From 1993 to 1997, he was Assistant Professor in the School of Electrical and Computer Engineering at the Georgia Institute of Technology. He is currently the Associate Director for the Center on Multidisciplinary Optical Switching Technology (MOST) and Associate Professor in the Department of Electrical and Computer Engineering at the University of California at Santa Barbara. He heads the Optical Communications and Photonic Networks (OCPN) Research. His current research areas are in optical communications, wavelength division multiplexing, photonic packet switched and all-optical networks, all-optical wavelength conversion, optical subcarrier multiplexing, and multispectral optical information processing.

Dr. Blumenthal is the recipient of a 1999 Presidential Early Career Award for Scientists and Engineers (PECASE) from the White House and the DoD, a 1994 NSF Young Investigator (NYI) Award, and a 1997 Office of Naval Research Young Investigator Program (YIP) Award. He is a member of the Optical Society of America (OSA) and the Lasers and Electrooptic Society (LEOS).