

An Alternative Implementation Perspective for the Scheduling Switch Architecture

George Theophilopoulos, Marios Kalyvas, Konstantinos Yiannopoulos, Kyriakos Vlachos, Emmanouel Varvarigos and Hercules Avramopoulos

Abstract— In this paper we propose a novel configuration for the implementation of an almost all-optical switch architecture, called the “Scheduling Switch”, which when combined with appropriate wait-for-reservation or tell-and-go connection and flow control protocols provides lossless communication for traffic that satisfies certain smoothness properties. An all-optical 2x2 exchange/bypass switch based on the nonlinear operation of a Semiconductor Optical Amplifier (SOA) is considered as the basic building block of the Scheduling Switch as opposed to active, SOA-based space switches that use injection current to switch between 'ON' and 'OFF' states. The experimental demonstration of the optically addressable 2x2 Exchange/Bypass, which is summarized for 10 Gbps data packets as well as SDH/STM-64 data frames, ensures the feasibility of the proposed configuration at high speeds, with low switching energy and low losses during the scheduling process. Additionally, it provides reduction of the number of required components for the construction of the Scheduling Switch, which is calculated to be 50% in the number of active elements and 33% in the fiber length.

Index Terms— All-optical signal processing, all-optical packet switching, scheduling switch architecture, exchange-bypass switch, ultrafast nonlinear interferometer, semiconductor optical amplifier, lossless communication.

I. INTRODUCTION

In the quest towards high capacity data networks, all optical packet switching is set to provide a path for the deployment of more efficient transport networks, offering a variety of optical services in an affordable way [1]-[3]. For optical packet switching, the optical layer must be transformed from a static transmission medium to a dynamically reconfigurable facility. This should possess the ability of changing the connectivity between nodes during the time scale of a packet and possibly to allow for some limited bit-level processing [4]-[6]. To this end, Semiconductor Optical Amplifier (SOA) based switch modules have been demonstrated at data rates up to 100 Gbps, operating with low switching energies (< 100 fJ)

and having the potential to be integrated in single chips [7]. Thus, the optical-to-electronic (O/E) and electronic-to-optical (E/O) conversion of the data signal at intermediate switches can be eliminated. However, O/E conversion is still required for header processing and consequently for the control of the switch fabric [8]. To efficiently perform packet switching in the so called (almost) all-optical packet switches [9], where the data remains in the optical domain and the header is processed electronically, the architecture of the switch should provide lossless communication, efficient capacity utilization, packet arrival in the correct order and design modularity. The Scheduling Switch architecture, which was first proposed in [9], meets the aforementioned requirements when combined with appropriate connection and flow control protocols. It is the purpose of this paper to show that the Scheduling Switch can be implemented with the experimentally demonstrated, SOA-based 2x2 exchange/bypass switch [11], offering the advantages of all-optical operation, high speed (given the potential of operation of the switch at 40 Gbps [12]), low energy consumption, as well as major reduction in the number of active and passive optical components required for the construction of the switching fabric.

The remainder of the paper is organized as follows. In Section II, we revise the principle of operation of the Scheduling Switch architecture. A typical and the proposed configuration for the realization of the Scheduling Switch is described in Section III, while in Section IV the experimental implementation of the 2x2 optically addressable exchange/bypass switch is summarized. Finally, in Section V we analyze, component-wise, the cost for the construction of the Scheduling Switch using both the initial and the proposed configuration.

II. THE SCHEDULING SWITCH

The Scheduling Switch is designed to provide lossless communication for sessions that have certain smoothness properties or can be transformed to sessions with such properties, tolerating the corresponding delay. The time axis on a link is divided into frames of length equal to T slots, where we assume that all packets have the same length and require one slot for transmission. A session is said to have the (n, T) -burstiness property [9], [13] at a node if at most n packets of the session arrive at that node during a frame of

Manuscript submitted August 25, 2003.

G. Theophilopoulos, M. Kalyvas, K. Yiannopoulos and H. Avramopoulos are with the Department of Electrical and Computer Engineering, National Technical University of Athens, 15773 Athens, Greece (e-mail: gtheof@cc.ece.ntua.gr, phone +30 210 7722057, fax +30 210 7722077).

M. Varvarigos and K. Vlachos are with the Department of Computer Engineering & Informatics, University of Patras, 26500 Patras, Greece.

size T . A session can easily be made to have the (n, T) – burstiness property at a source, and the property is automatically preserved throughout a network consisting of scheduling switches, since such switches maintain frame integrity.

We let n_{ij} be the number of packets that arrive during a frame over incoming link i and have to be transmitted on link j , and N the number of incoming (and outgoing) links of a node. If the connection and flow control protocols used guarantee that the number of packets which require the same outgoing link j in a frame is less than or equal to the frame size T , i.e.,

$$\sum_{i=1}^N n_{i,j} \leq T \quad (1)$$

then all of the incoming packets can be assigned slots in the required outgoing links so that no packets will have to be dropped. Both wait-for-reservation and tell-and-go protocols can be used to ensure that this property is met, as described in [14]-[16].

The scheduling switch consists of a Scheduler with N input and N output ports, and a $N \times N$ non-blocking space switch, as shown in fig. 1. The purpose of the Scheduler is to rearrange the incoming packets, so that packets appearing during the same slot at its output request different outgoing links of the space switch. If this procedure is done successfully, there will be no collisions at any of the space switch output ports.

The function of the Scheduler can be described through a

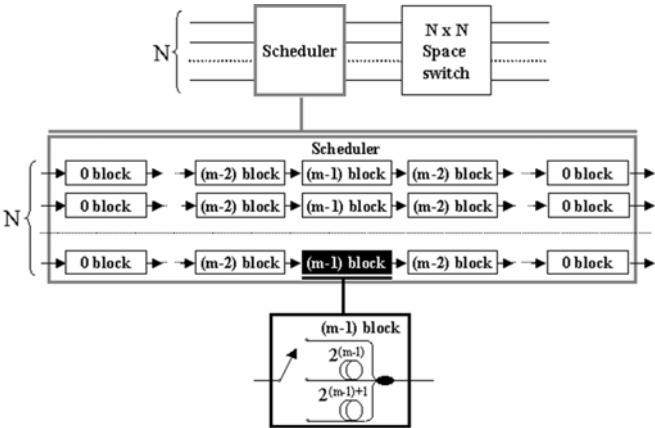


Fig. 1: The Scheduling Switch architecture, consisting of the Scheduler (N inputs) and a $N \times N$ space switch. The Scheduler comprises of N branches, each of which consists of $2\log_2 T - 1$ delay blocks. The i -th delay block consists of one three-state switch and three delay lines of length 0 , 2^i and 2^{i+1} packet slots.

frame arrival matrix (N), defined as the $N \times N$ matrix, whose (i, j) component is equal to the number of packets that arrive during a given frame $F(i)$ of the incoming link i and require the same frame $F(j)$ of outgoing link j . By defining the *permutation matrix* as a $N \times N$ matrix with the property that each line has at most one non-zero element, indeed equal to '1', the *frame matrix* can be written as the sum of at most T *permutation matrices* P_s , $s=1,2,\dots,T$, when condition (1) is satisfied. The matrix P_s can be used to determine the packet (if any) that will appear during slot s at each of the output ports of the scheduler. In particular, if the (i, j) element of this

matrix is equal to '1', then a packet arriving over link i and departing over link j is assigned to outgoing slot s of the scheduler.

The Scheduler comprises of N parallel branches, one for each input, where each branch delays the packets arriving on an incoming link until their assigned slots on their desired outgoing links. This is equivalent to a time-slot interchanger and is implemented using $(2\log_2 T - 1)$ three-state delay blocks (fig. 1), where we have assumed T is a power of 2. The i^{th} block [in fig. 1, we illustrate the details for block $i = (m-1)$] consists of a three-state switch and three fiber delay paths, corresponding to delays equal to 0 , 2^i and 2^{i+1} packet slots. To ensure that the packets in the incoming frame can be assigned to any slot in the outgoing frame, the outgoing frame must start at least $(3T)/2 - 2$ after the incoming frame begins [10].

Using the approach of [17], each branch of delay blocks can

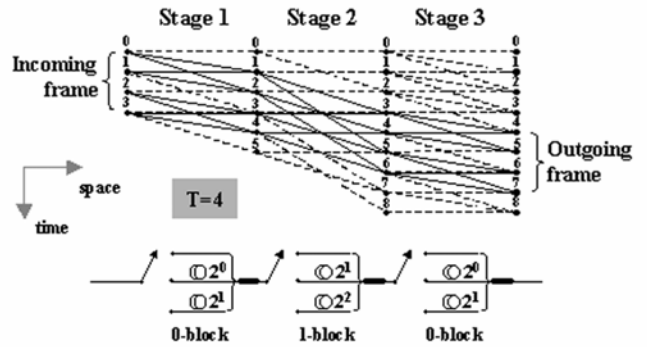


Fig. 2: Space-time representation of a branch of the scheduler when $T = 4$. Each line (solid or dashed) represents a feasible state transition in the time-space domain. The solid lines shows that the time-space representation has a Beneš subgraph.

be expanded into a corresponding graph, using a space-time representation, as shown in fig. 2 for $T = 4$ and accordingly for $(2\log_2 4 - 1) = 3$ delay blocks. By using the above concept, the problem of scheduling packets through a branch of delay blocks to avoid collisions, becomes a problem of routing every packet of the incoming frame through the space-time graph to the appropriate slot of the outgoing frame, where node-disjoint paths in the graph correspond to collision-free transmission through the delay blocks of the Scheduler. In fig. 2 the incoming frame corresponds to time slots 0–3, while the outgoing frame corresponds to time slots 4–7. Each line, either solid or dashed, represents a feasible state transition in the time-space domain; movement along a horizontal line between nodes corresponds to passing through a delay block without being delayed, while movement along a cross line between nodes corresponds to passing through a delay block and being delayed. Packet collisions can be avoided within a branch of delay blocks if the paths followed by the packets are node disjoint. Among the feasible state transitions in the space-time representation of fig. 2, we observe a Beneš structure (solid lines), which is known to be rearrangeably non-blocking [18]. This means that it is possible to schedule the packets in a collision-free manner through a branch of delay blocks using only the transitions corresponding to the solid lines in the space-time graph. As mentioned earlier, for

the Beneš structure to stand, the outgoing frame must start (3T)/2-2 packet slots after the incoming frame begins [10].

III. TYPICAL AND ALTERNATIVE CONFIGURATION OF THE SCHEDULING SWITCH ARCHITECTURE

We propose a novel configuration for the implementation of the aforementioned “scheduling switch” architecture that is based on the use of an experimentally demonstrated, all-optical 2x2 exchange/bypass (E/B) switch as the basic building block. The 2x2 switch, which is described in detail in Section VI, ensures the feasibility of the proposed configuration at high speeds, with low switching energy and low losses during the scheduling process. Additionally, it provides 50% reduction in the required active elements (SOAs) components and 33% in the fiber length, if compared to a typical SOA-based implementation. From now on, we will refer to the proposed configuration as the “E/B implementation”, while to the typical configuration as the “SOA-based implementation”.

A. Typical implementation of the scheduling switch

One of the most common techniques applied for optical switching is the use of SOAs as active space switches [19]-[22]. Current injection into semiconductor pn-junctions generates free carriers, and this carrier modulation varies the loss and/or gain characteristics. Employing these characteristics, switchable semiconductor optical amplifiers (SOAs) can be realized and many different configurations for different applications have already been demonstrated [23]. The modulation speed that has been achieved with this method is in the order of 1 ns.

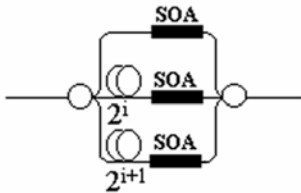


Fig. 3: Implementation of the i th delay block of the scheduler, using three switchable SOAs.

A typical implementation of the i th block of the Scheduler using the aforementioned technique is shown in fig. 3. Each of the three SOAs can be operated in “transparent” (‘ON’) or in “block” (‘OFF’) mode, allowing or preventing respectively the corresponding signal to pass through. In this way, the incoming packet may experience zero delay, delay equal to 2^i packet slots or delay equal to 2^{i+1} packet slots, depending on which of the three SOAs is ‘ON’. In every packet slot, only one of the three SOAs is ‘ON’ during the slot, while the other two are ‘OFF’, depending on the delay that is required to be inserted.

Following a similar rationale, the $N \times N$ space switch can be implemented using N^2 SOAs, as shown in fig. 4. A fully connected shuffle network between input and output ports is realized using adequate power splitters and combiners. Each

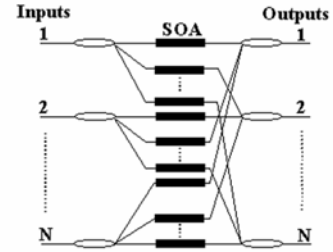


Fig. 4: A common implementation of the $N \times N$ space switch using N^2 SOAs.

connecting path can be switched ‘ON’ or ‘OFF’ using the corresponding SOA. In particular, only one of these SOAs shall be ‘ON’ during a slot, depending on the desirable outgoing link, while the rest $(N-1)$ shall be ‘OFF’. Such a switching matrix generates splitting losses that may be important for large switching arrays. To compensate for the losses, additional booster amplifiers can be included with the drawback of noise accumulation. The configuration shown in fig. 4 forms a strictly nonblocking space switch with broadcast capabilities.

B. Alternative implementation of the scheduling switch

In this sub-section we describe the proposed E/B implementation for the Scheduler and the $N \times N$ space switch that follows it, using appropriate number of 2x2 exchange/bypass switches. A 2x2 switch has two operating states: the ‘BAR’ state, where the two input signals pass through unaffected to the corresponding output ports and ‘CROSS’, where the two input signals are interchanged at the output ports.

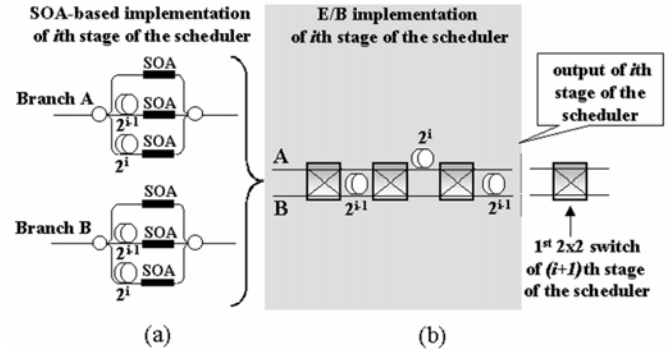


Fig. 5: (a) SOA-based implementation and (b) E/B implementation of the i th stage of the Scheduler.

Fig. 5 (a) shows the SOA-based implementation of the i th stage for two branches of the Scheduler, while in fig. 5 (b) the E/B implementation is depicted. We will show that the latter implementation is functionally equivalent to the former. In the new implementation, every two branches of the Scheduler (corresponding to a pair of input ports) are integrated in the two ports of a 2x2 switch and the whole stage uses three such switches. The fourth 2x2 switch shown on the right of fig. 5 (b) is the corresponding first switch of the next stage. Each switch changes its state in every packet slot. In the following paragraphs we prove that the space-time representation of the configuration of fig. 5 (a) can be emulated in a collision-free manner by the space-time representation of the configuration of fig. 5 (b). After showing this, it will become evident that we can replace, in the implementation of the delay blocks of

the Scheduler, the configuration of fig. 5 (a) with the configuration of fig. 5 (b), without affecting the essential functionality and properties of the scheduling switch, except for some additional fixed delay that is introduced. The new implementation has several performance and cost advantages over the previous implementation, as will be shown later on.

In the SOA-based implementation, collisions may occur at the output of stage i of the Scheduler only between packets that arrive at its input abstaining 2^{i-1} time slots. We consider two such packets, and we assume without loss of generality that they appear in time slots '0' and ' 2^{i-1} '. In Fig. 6 we show the time space location of two such packets at the inputs of the i^{th} stage for branches A and B of the Scheduler, and the possible time space locations of the two packets at the outputs of stage i . Packets that arrive at the input of the i^{th} stage but do

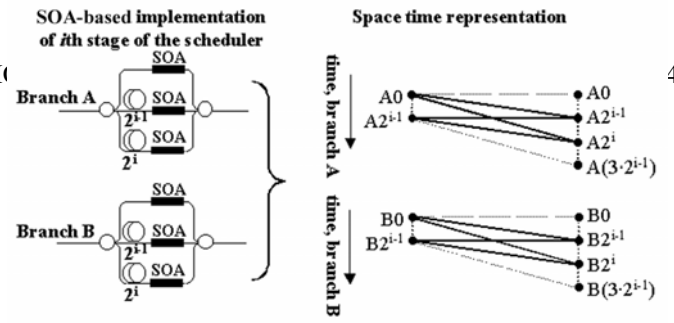


Fig. 6: Time-space representation of the i^{th} stage of two branches A and B of the SOA-based implementation of the scheduler. We only illustrate time-space locations of packets (nodes) that are 2^{i-1} slots apart from each other, because they are the only ones that could generate collisions at the outputs of stage i . The solid lines correspond to delay-block states that may be used.

the inputs of stage i of the Scheduler on slots ' 0 ' and ' 2^{i-1} ' for branches A and B. Packets that arrive on slots that do not abstain 2^{i-1} slots, do not collide to each other (since all delays introduced in stage i are multiples of 2^{i-1} slots) and are not

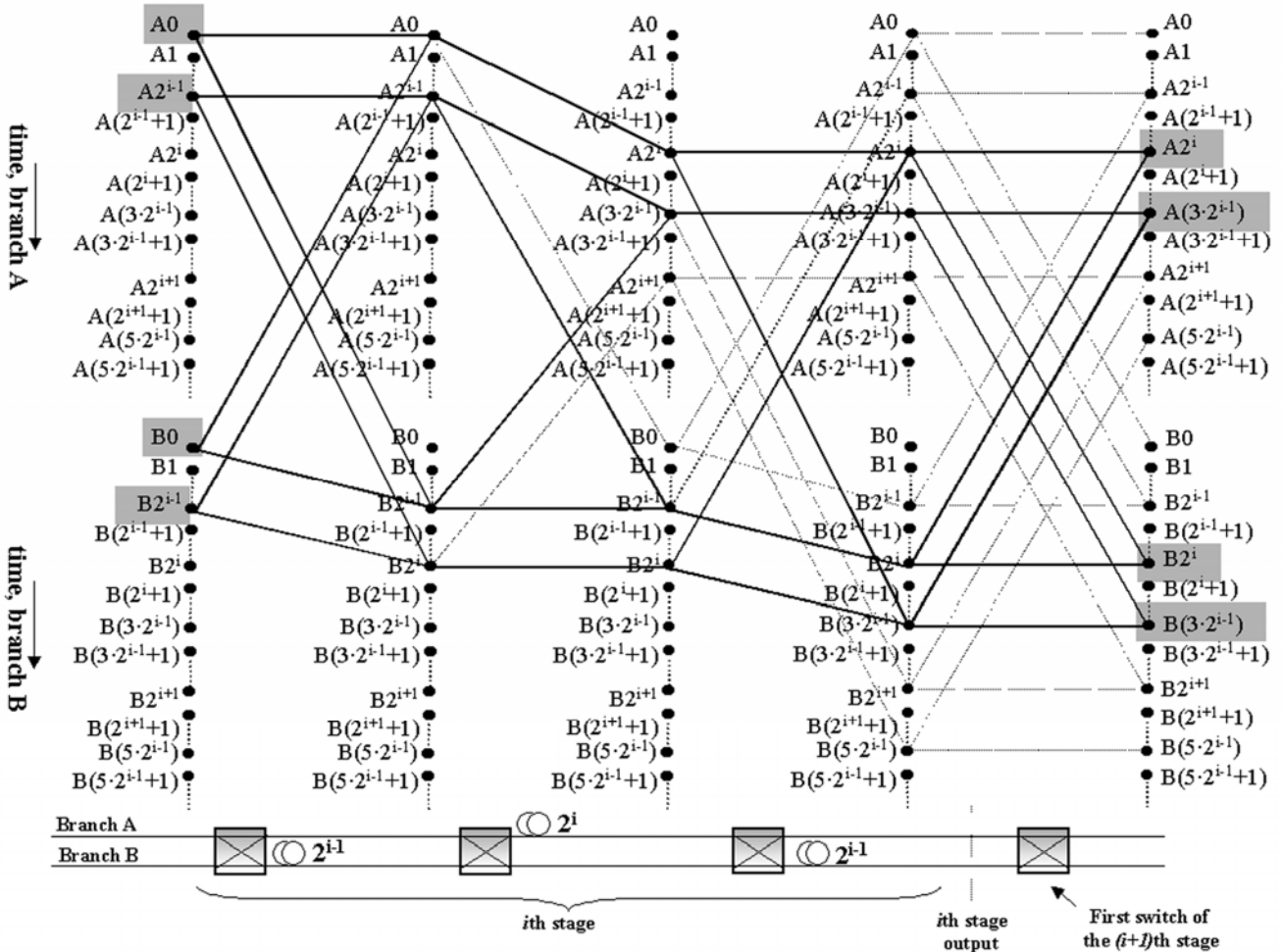


Fig. 7: Space-time representation for the i^{th} stage of the E/B implementation of the scheduler architecture for packets abstaining 2^{i-1} slots. The lines (dashed and solid) represent all the feasible state transitions in the time-space domain. The solid lines correspond to delay-block states that may be used.

not abstain 2^{i-1} slots, have no possibility of colliding at the outputs of the i^{th} stage and do not need to be considered.

In fig. 7 we give the space-time representation of the i^{th} stage of the E/B implementation. Node A_n (or B_n) of a given sub-stage of the time space representation of stage i corresponds to the n^{th} slot of the upper (or respectively lower) input of the 2×2 switch of that sub-stage. We only show the time space paths that could be followed by packets arriving at

depicted in detail. Additionally, packets on different branches are not related to the packets on branches A and B since they follow space-disjoint paths. Note that the incoming time-space slots $A_0, A_{2^{i-1}}$ are routed to time-space slots $A_{2^{i-1}}, A_{2^i}$ at the output of stage i of the SOA-based implementation (fig. 6), while they are routed to time-space slots $A_{2^i}, A_{3 \cdot 2^{i-1}}$ at the output of stage i of the E/B implementation (fig. 7). This additional delay of 2^{i-1} slots, introduced by stage i in the E/B

implementation, results in the imposition of an extra total delay to the outgoing frame, in relation to the incoming frame. This fixed additional delay does not play any role in the switch ability to schedule packets in a collision-free way and is calculated later on in this Section [Eq. 2].

In order to make the principle of operation of the E/B implementation and the role of the 2x2 switches more clear, we mention the following example, referring to fig. 7; the first packet in branch A (which is initially placed in slot '0') can: **(a)** experience zero delay (with the first exchange/bypass switch operating in "BAR" state) and remain in slot '0' (in particular in 'A0', since it is placed in the upper line of the delay block) or **(b)** experience delay equal to 2^{i-1} time slots (if the first exchange/bypass switch operates in "CROSS" state) and be carried to slot ' 2^{i-1} ' (in particular in 'B 2^{i-1} ', since it is now placed in the lower line of the delay block). At the same time, the first packet in branch B (initially placed in slot '0') will: in case **(a)** experience delay equal to 2^{i-1} time slots and be carried to slot ' 2^{i-1} ' or, in case **(b)**, experience zero delay and remain in slot '0' but in the upper branch (slot 'A0').

In fig. 7 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 does not follow a known structure (e.g. in fig. 2 a Beneš structure was formed) and it is not thereby profound that packets can be routed in a collision-free manner, equivalent to that shown in fig. 6.

However, if we rearrange the solid lines of fig. 7, they form

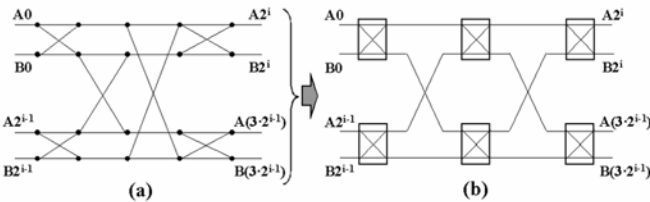


Fig. 8: (a) Network formed by the solid lines of fig. 7. (b) Equivalent topology, which is a Beneš switch, i.e. in every case, node-disjoint paths exist for all of the four packets.

the space-time representation graph shown in fig. 8(a). If we substitute the paths with properly connected 2x2 elementary switches, a Beneš switch is formed, as the one shown in fig. 8(b). Since the Beneš graph is rearrangeably non-blocking, every input can be routed to any output using a node-disjoint path. This means that the input packets at branches A and B may come out at any desired slot in a collision-free manner by using the proposed architecture.

By expanding this conclusion, we can claim that when we can find collision free paths to reschedule the packets in any way allowed by the original architecture in fig 5(a), then we can always find collision free paths to reschedule the packets in the same way using the proposed architecture, shown in fig. 5(b). The only difference is that packets that enter in slots 0 and 2^{i-1} , instead of appearing in slots 2^{i-1} and 2^i , appear in slots 2^i and $3 \cdot 2^{i-1}$. This means that by using the new architecture, an extra time delay is inserted. In order to calculate the minimum

number of packet slots that the incoming and the outgoing frame must abstain when using the E/B implementation, we calculate the number of packet slots between the first packet of the incoming frame (slot '0') and the first packet of the outgoing frame. This number corresponds to the delay of 2^i packets slots through each one of the i blocks, where $i=1, 2, \dots, \log_2 T-1, \log_2 T, \log_2 T-1, \dots, 2, 1$. By summing all these delays, we get:

$$L_{branch} = 2 \cdot \sum_{i=1}^{\log_2 T-1} 2^i + T = 2^{\log_2 T+1} - 4 + T = 3T - 4 \quad (2)$$

This means that the outgoing frame must start at least $3T - 4$ packet slots after the incoming frame begins. Consequently, we need double the time in order to re-arrange the packets in the proposed scheme (in the original architecture, this number was $[3T]/2 - 2$).

It is easy to show that all the other packets of the two frames (entering in branches A and B) can be rearranged in a collision-free manner as well, since in each stage of fig. 7, two standard time slots are used for the two packets of each frame that abstain 2^{i-1} . This observation ensures that no internal collisions occur inside the scheduler.

One more remark concerning fig. 7, is the role of the fourth exchange/bypass switch (1st 2x2 switch of the $(i+1)$ th stage of the scheduler). At the input of this switch all the packets have been assigned to the desired time slot but it is possible to be in the wrong line of the delay block (branch A or B). For example, a packet from frame A may be in the lower line, assigned however to the right time slot. The line that a packet is (upper or lower) does not have any meaning, since it can be corrected from the exchange/bypass switch of the next stage. Yet, at the output of the scheduler, an extra exchange/bypass switch is obligatory in order to assign all the packets from frame A to frame A and all the packets from frame B to frame B. This can be done in such a way that frame A occurs in the upper line at the input of the space switch and frame B at the lower or vice-versa, accomplishing this way elementary switching. The last observation has impact in the design of the NxN space switch too, as will be shown later on.

Consequently, each stage i of the SOA-based implementation (two branches) can be emulated by the E/B implementation, at the cost of a fixed extra delay that is inserted between the incoming and the outgoing frame. If all the stages of the SOA-based implementation are replaced by the E/B implementation, then the whole scheduler is emulated, except that packets in the input frame are routed to a frame that starts $(2T-4)$ packet slots after the end of the input frame, instead of $(T/2-2)$ when using the original configuration.

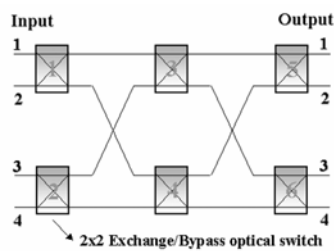


Fig. 9: A 4x4 rearrangeably non-blocking Beneš structure for the space switch, using 6 '2x2 exchange/bypass switches'.

By using the 2x2 exchange/bypass switch it is possible to simplify the space switch as well, and implement it not with a SOA-based, crossbar configuration (see fig. 4), but with a different one that must be at least rearrangeably non-blocking (e.g. Beneš). Fig. 9 shows a 4x4 rearrangeably non-blocking Beneš structure for the space switch. The corresponding crossbar switch according to the SOA-based implementation would use $N^2 = 16$ active elements, while this one uses only 6. It is known that for any N , a rearrangeably non-blocking $N \times N$ Beneš structure consists of three stages and requires $(N/2)$ 2x2 switches at its input, $(N/2)$ 2x2 switches at its output and $2(N/2) \times (N/2)$ Beneš switches at its middle stage. The 2x2 exchange/bypass switches at the output of the scheduler branches (i.e. before the input ports of the space switch) can be used as the first stage of the Beneš configuration, reducing thus the required switches in the space switch.

IV. EXPERIMENTAL DEMONSTRATION OF THE 2X2 ALL-OPTICAL EXCHANGE-BYPASS SWITCH

For the operation of the optically addressable 2x2 exchange/bypass switch three optical signals are needed, two data signals and one control signal, as shown in fig. 10. Data signals enter the switch from the input ports 1 and 2. If there is no control signal the switch is in the 'BAR' state and both data signals pass straight through to the output ports 1 and 2. If the control signal is present the switch is in the 'CROSS' state and the two data streams are interchanged at its output. The length of the bit sequence that is interchanged through the switch is determined by the length of the control signal and

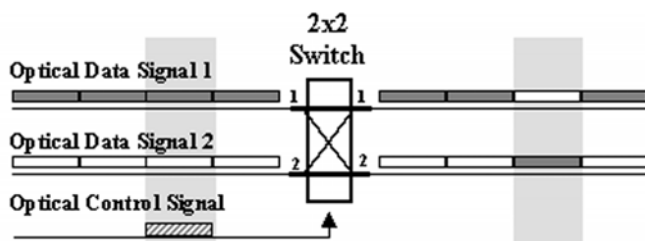


Fig. 10: Principle of operation of the 2x2 Exchange/Bypass Switch.

may be arbitrarily long or short depending on the length of the incoming packet.

The 2x2 exchange/bypass switch is constructed with an Ultrafast Nonlinear Interferometer-UNI based optical gate [24] properly configured to provide two input, two output and one control port, as shown in fig. 11. Data signals 1 and 2 enter through the input ports A and B, while the control signal

through port CON. If there is no control signal then data signal 1 exits through port X, while data signal 2 exits through port Y. In the presence of a control pulse, the phase of the two synchronized, counter-propagating polarization components is changed simultaneously, so that when the components of each data signal recombine at the corresponding Polarization Beam Splitter (PBS), their polarization states are rotated by 90° . In this way, data signal 1 exits through port Y, while data signal 2 through port X. In our implementation the nonlinear element in the UNI gate was a 1.5 mm bulk InGaAsP/InP ridge waveguide SOA with small signal gain of 30 dB at 1560 nm with 700 mA drive current.

Initially, the performance of the switch was investigated at the data packet level. Specifically, the data signals were PRBS and PRBS packets, while the control signal consisted of packets containing 10 GHz clock pulses at a different wavelength than the data. Fig. 12 shows the two output ports of the switch monitored simultaneously on a sampling oscilloscope. Specifically, fig. 12 (a) shows the input data signals 1 and 2 into the switch, fig. 12 (b) and (c) the corresponding output signals for the 'BAR' and the 'CROSS' state respectively and fig. 12 (d) shows the control signal. In the 'BAR' state, the data packets from signals 1 and 2 cross the switch unchanged. While the optical control signal is present, the switch is in the 'CROSS' state and the packets are interchanged in the output ports. The crosstalk of the switch in the BAR and in the CROSS state was -12 dB and -10 dB respectively. In the presence of the control signal there was also a 1 dB drop in the switched signals due to additional SOA gain saturation, that may be mitigated using a gain transparent arrangement [25].

It is important to note that if this exchange-bypass unit is used for optical packet switching, it relaxes the requirement for guardbands between the packets, since the switch changes state within the bit period. In avoiding guardbands, the improvement in throughput becomes more pronounced as the packet length decreases.

The error performance of the switch was evaluated in static configuration with input data of SDH/STM 64 format. In particular, both data signals were generated with the SDH/STM-64 Network Analyzer by Acterna, which produced SDH packets containing a $2^{31}-1$ maximal length PRBS at 9.95328 Gb/s. The control signal was a continuous clock at the same rate and at a different wavelength. The data input

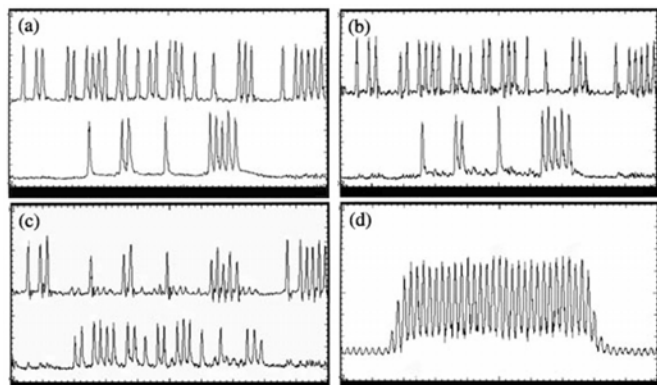


Fig. 12: BAR and CROSS state: (a) input signals, (b) output data streams with the control signal off (BAR state), (c) output data streams with the control signal on (CROSS state), (d) control signal. The time base is 500 ps/div.

streams were decorrelated in the switch by using different optical delays between them, while the control stream could be turned on or off to assess the switch states. The error rate was statistically calculated from the network analyzer by checking appropriate control bits at each SDH/STM-64 packet and was less than 10^{-11} for both data signals in both switch states.

V. COST ANALYSIS

A basic cost analysis, concerning the SOA-based and the E/B implementation of the scheduler and the space switch, is presented in this section. As measures of cost we will compare the total number of elementary components and the insertion losses of the original and of the proposed architecture.

The SOA-based scheduling switch architecture is composed of a scheduler and a $N \times N$ space switch. The scheduler has N parallel branches, one for each input port of the space switch. Consequently, for the whole scheduler,

$$K_{SOAs}^{Sched.} = 3N(2 \log_2 T - 1) \quad (2)$$

active elements (SOAs) are required, as well as $[2 \cdot N \cdot (2 \log_2 T - 1)]$ 3:1 couplers. In addition, for each delay block (the i^{th}), we need length of fiber that inserts $(2^i + 2^{i+1})$ packet slots delay. Accordingly, for every branch of the scheduler we need

$$L_{branch} = 2 \cdot \sum_{i=0}^{\log_2 T - 2} (2^i + 2^{i+1}) + (T/2 + T) \quad (3)$$

total length of fiber, measured in packet slots. The summation term in the above equation corresponds to all the symmetrical

TABLE I
COST ANALYSIS FOR THE SOA-BASED AND THE E/B IMPLEMENTATION OF THE SCHEDULING SWITCH ARCHITECTURE IN TERMS OF ELEMENTARY COMPONENTS COST.

	Components	SOA-based implementation	E/B implementation
N-branch Scheduler	Delay blocks	$2 \cdot N \log_2 T - N$	$3 \cdot N \log_2 T - 3 \cdot N/2$
	Fiber length	$9 \cdot N \cdot T/2 - 6 \cdot N$	$3 \cdot N \cdot T - 4 \cdot N$
	Active Elements (SOAs)	$3 \cdot N \cdot (2 \log_2 T - 1)$	$3 \cdot N \cdot (\log_2 T - 1/3)$
	3:1 couplers	$2 \cdot N \cdot (2 \log_2 T - 1)$	-
$N \times N$ Space Switch	Active Elements (SOAs)	N^2	$N \cdot (\log_2 N - 1)$

blocks of the scheduler, while the last term to the middle delay block (the $[\log_2 T - 1]$ th). For all the branches the required fiber is calculated to be

$$\begin{aligned} L &= N \cdot L_{branch} = 2 \cdot N \cdot \sum_{i=0}^{\log_2 T - 2} (2^i + 2^{i+1}) + N \cdot (T/2 + T) = \\ &= 6 \cdot N \cdot \sum_{i=0}^{\log_2 T - 2} (2^i) + 3N \cdot T/2 = 9 \cdot N \cdot T/2 - 6N \end{aligned} \quad (4)$$

packet slots.

Concerning the $N \times N$ space switch in the SOA-based implementation (see fig. 5), the number of active elements (SOAs) required is:

$$K_{SOAs}^{NxN} = N^2 \quad (5)$$

In the E/B implementation, the number of active elements in the scheduler is calculated as following: The scheduler has $(N/2)$ 2-port-branches, each consisting of $(2 \log_2 T - 1)$ blocks, while each block has 3 exchange/bypass switches. Consequently, $[3 \cdot (N/2) \cdot (2 \log_2 T - 1)] = [3N \cdot (\log_2 T - 1/2)]$ active elements (SOAs) are required for the scheduler. At the output of the scheduler (and at the input of the space switch), another $(N/2)$ exchange/bypass switches are needed and these will be subtracted from the active elements that the space switch needs. Consequently, the number of the active elements for the scheduler is:

$$K_{SOAs}^{Sched.} = 3N(\log_2 T - 1/2) + N/2 = 3N(\log_2 T - 1/3) \quad (6)$$

The total length of fiber used in the delay stages of this case is calculated by following the same rationale as in the SOA-based implementation and is given by:

$$\begin{aligned} L &= (N/2) \cdot \left[2 \cdot \sum_{i=0}^{\log_2 T - 2} (2 \cdot 2^i + 2^{i+1}) + (T/2 + T + T/2) \right] = \\ &= N \cdot \sum_{i=0}^{\log_2 T - 2} (2^{i+2}) + N \cdot T = 4 \cdot N \cdot (T/2 - 1) + N \cdot T = \\ &= 3 \cdot N \cdot T - 4 \cdot N \end{aligned} \quad (7)$$

Concerning the $N \times N$ space switch, it is known that in general, a rearrangeably non-blocking $N \times N$ Beneš structure requires

$$S_N = N \cdot \log_2 N - N/2 \quad (8)$$

2×2 switches, N being a power of 2. In our case, this number is further reduced by $N/2$ because of the respective extra number of exchange/bypass switches that the scheduler uses at its output. Consequently, the number that the $N \times N$ space switch uses in the rearrangeably non-blocking Beneš structure is

$$K_{SOAs}^{NxN} = N \cdot (\log_2 N - 1) \quad (9)$$

Table I summarizes the above cost analysis for the original and the alternative switch architecture, in terms of elementary components cost.

Concerning the scheduler, the proposed technique consists of more delay blocks, since two blocks of the original scheduler equals three serially connected exchange/bypass switches. However, the total length of fiber used in the alternative technique is 33% less, while the number of the active elements (SOAs) as well as the corresponding electronic circuitry is reduced by almost 50%. In addition no 3:1 couplers are required, limiting thus the circuit losses. As far as the space switch concerns, the active elements used in the proposed rearrangeably non-blocking Beneš architecture are reduced by $(\log_2 N - 1)/N$. Since the 2×2 exchange/bypass switch exhibits 4 dB gain, these switches perform very well when cascaded, dispensing the need of amplification between them. This does not occur in the SOA-based configuration, since the use of two 3:1 splitters in each stage of the scheduler

leads to a minimum of 9,5 dB losses per stage. Finally, the switching energy for the demonstrated 2x2 switch is in the order of fJ/pulse, orders of magnitude lower than the power that the switchable SOAs demand.

VI. CONCLUSIONS

We have proposed a novel configuration for the implementation of an almost all-optical switch architecture, called the “scheduling switch” architecture by using an experimentally demonstrated, all-optical 2x2 exchange/bypass switch as the basic building block, as opposed to active SOA-based space switches that use injection current to switch between ‘ON’ and ‘OFF’ states. The 2x2 switch ensures the feasibility of the proposed configuration at high speeds, with low switching energy and low losses during the scheduling process. Additionally, it provides 50% reduction in the required active elements (SOAs) components and 33% in the fiber length.

REFERENCES

- [1] M. Renaud, F. Masetti, C. Guillemot and B. Bostica, “Network and system concepts for optical packet switching,” *IEEE Commun. Mag.*, vol. 35, no. 4, pp. 96-102, Apr. 2001.
- [2] M. J. O’Mahony, D. Simeonidou, D. K. Hunter and A. Tzanakaki, “The application of optical packet switching in future communication networks,” *IEEE Commun. Mag.*, vol. 39, no. 3, pp. 128-135, Mar. 2001.
- [3] D. Benjamin, R. Trudel, S. Shew, and E. Kus, “Optical Services over the Intelligent Optical network,” *IEEE Commun. Mag.*, vol. 39, no. 9, pp. 73-78, Sep. 2001.
- [4] V. W. S. Chan, K. L. Hall, E. Modiano and K. A. Rauschenbach, “Architectures and technologies for high-speed optical data networks,” *IEEE J. Lightwave Technol.*, vol. 16, no. 12, pp. 2146–2168, Dec. 1998.
- [5] K. E. Stubkjaer, “Semiconductor optical amplifier-based all-optical gates for high-speed optical processing,” *IEEE J. Select. Topics Quantum Electron.*, vol. 6, no. 6, pp. 1428-1435, 2000.
- [6] H. Avramopoulos, “TDM Devices and their applications,” in *Proc. Optical Fiber Communication Conference (OFC) 2001, WE (Tutorial Sessions)*.
- [7] S. A. Hamilton, B. S. Robinson, T. E. Murphy, S. J. Savage and E. P. Ippen, “100 Gb/s optical time-division multiplexed networks,” *IEEE J. Lightwave Technol.*, vol. 20, no. 12, pp. 2086-2100, Dec. 2002.
- [8] R. L. Cruz and J.-T. Tsai, “COD: Alternative architectures for high speed packet switching,” *IEEE/ACM Trans. Networking*, vol. 4, no. 2, 1996.
- [9] E. A. Varvarigos, “The “Packing” and the “Scheduling” packet switch architectures for almost all-optical lossless networks”, *J. Lightwave Technol.*, vol. 16, pp. 1725-36, Oct. 1998.
- [10] J. P. Lang, E. A. Varvarigos and D. J. Blumenthal, “The λ -Scheduler: A Multiwavelength Scheduling Switch”, *J. Lightwave Technol.*, vol. 18, pp. 1049-1063, Aug. 2000.
- [11] G. Theophilopoulos et al., “Optically Addressable 2x2 Exchange Bypass Packet Switch”, *IEEE Photonics Technology Letters*, Vol. 14, pp. 998-1000, July 2002.
- [12] N. S. Patel, K. A. Rauschenbach and K. L. Hall, “40-Gb/s demultiplexing using an ultrafast nonlinear interferometer (UNI),” *IEEE Photon. Technol. Lett.*, vol. 8, no. 12, pp. 1695-1697, Dec. 1996.
- [13] S. J. Golestani, “Congestion-free communication in high-speed packet networks”, *IEEE Trans. Commun.*, vol. 39, pp. 1802–12, 1991.
- [14] E.A. Varvarigos and V. Sharma, “An efficient reservation connection control protocol for gigabit networks”, *Computer Networks*, Vol. 1998, No. 12, pp. 1135-1156, 1998.
- [15] E.A. Varvarigos, “Control Protocols for Multigigabit-per-Second Networks”, *IEICE Transactions on Communications*, Vol. 1998, No. 2, pp. 440-448, 1998, ISSN: 0916-8516.

- [16] E.A. Varvarigos and J.P. Lang, “A virtual circuit deflection protocol”, *IEEE/ACM Transactions on Networking*, vol.7, No.3, pp. 335-349, 1999.
- [17] D. Hunter and D. Smith, “An architecture for frame integrity optical TDM switching,” *J. Lightwave Technol.*, Vol. 11, pp. 914–924, 1993.
- [18] F. Leighton, “Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes”, San Mateo, CA: Morgan Kaufmann, 1992.
- [19] C. Guillemot et al, “Transparent optical packet switching: The European acts KEOPS project approach,” *IEEE J. Lightwave Tech.*, vol. 16, pp. 2117-2134, 1998.
- [20] D. Chiaroni et al, “First demonstration of an asynchronous optical packet switching matrix prototype for MultiTerabit class routers/switches,” in *Proc. ECOC 01*, vol. 6, pp. 60-61.
- [21] 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. 219–227, 1997.
- [22] M. Renaud, M. Bachmann, and M. Erman, “Semiconductor Optical Space Switches”, *IEEE J. Select. Topics in Quantum electron.*, vol. 2, no. 2, pp. 277-287, June 1996.
- [23] P. Doussi re, “Recent advances in conventional and gain clamped semiconductor optical amplifiers,” presented at *Conf. Optical Amplifiers and Their Applications (OAA)*, 1996, invited paper.
- [24] K. Tajima, S. Nakamura, Y. Sugimoto, “Ultrafast polarization discriminating Mach-Zehnder all optical switch”, *Applied Physics Letters*, Vol. 67, No. 25, pp. 3709-3711, 1995.
- [25] S. Diez, R. Ludwig, H.G. Weber, “Gain-Transparent SOA-Switch for High-Bitrate OTDM Add/Drop Multiplexing,” *IEEE Photonics Technology Letters*, Vol. 11, pp. 60-62, 1999.



and semiconductor switching technologies. He is a member of IEEE and author or co-author of 20 publications.



the design and implementation of high speed (up to 40 Gbps) all-optical shift registers, memories, samplers, 2x2 switches, ultrafast nonlinear interferometric switches (AND and XOR gates) and on their functional integration. Dr. Kalyvas has worked for 2 months at Corning Inc. in the theoretical and experimental study of crosstalk in SCM WDM-CATV networks. He is the author or co-author of 20 publications.



modeling of linear shift registers, theoretical and experimental measurements

Dr. George Theophilopoulos was born in Athens, Greece in 1976. He obtained his Diploma of Electrical and Computer Engineering from the National Technical University of Athens (NTUA), Greece, in 1999 and his Ph.D degree in Electrical and Computer Engineering from the Photonics Communications Research Laboratory (PCRL) of NTUA in 2003. Since then, he is a senior research associate in PCRL. His research interests include all-optical logic, all-optical switches, optical networks

Dr. Marios Kalyvas was born in Ioannina, Greece, on 1976. He obtained the Diploma of Electrical Engineering & Computer Science from the Department of Electrical Engineering & Computer Science of the National Technical University of Athens, Greece, with specialization in Data Networks and Telecommunications and his Ph.D degree in Electrical and Computer Engineering from the Photonics Communications Research Laboratory (PCRL) of NTUA in 2003. Since then, he is a senior research associate in PCRL. His work is focused on

Konstantinos Yiannopoulos was born in Tripoli, Greece, on December 1977. He obtained the Diploma of Electrical Engineering & Computer Science with specialization in Telecommunications in 2000, graduating among the top 1% of the department. Mr Yiannopoulos is a third year graduate student working toward his PhD. His laboratory related work experience includes all optical high-speed logic, modeling and fabrication of low noise figure erbium doped fiber amplifiers,

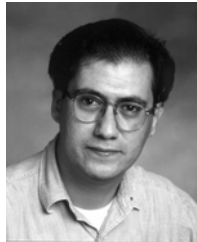
on amplitude and phase jittering of ring lasers. He is currently working on novel all optical clock recovery methods. As an undergraduate student Mr Yiannopoulos was awarded two grants from the National Scholarships Foundation and one grant from the National Technical University of Athens; he was also eligible for a four-year private institution scholarship.



Dr. Kyriakos Vlachos (SM'98–M'02) was born in Athens, Greece. He received his Dipl.-Ing. Degree in Electrical and Computer Engineering, from National University of Athens (NTUA), Greece, in 1998 and his Ph.D in Electrical and Computer Engineering also from National Technical University of Athens (NTUA), in 2001.

From 1997 to 2001, he has been a senior research associate in Photonics Communications Research Laboratory, in the National Technical University of Athens, Greece. From April 2001 to January 2003 he has been member of the technical staff of Bell Laboratories, Lucent Technologies, The Netherlands, where he conducted research on high-speed, short-pulse communication systems, optical packet switching, optical labeling techniques and ultrafast optical signal processing. From January 2003 till now he is a senior research associate in Computer Engineering and Informatics Department in the University of Patras, Greece. Dr. Vlachos has participated in various European and National R&D Programmes.

Prof. Emmanouel (Manos) Varvarigos was born in Athens, Greece, in 1965. He received a Diploma in Electrical and Computer Engineering from the National Technical University of Athens in 1988, and the M.S. and Ph.D. degrees in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology, Cambridge, MA, in 1990 and 1992, respectively. In 1990 he was a researcher at Bell Communications Research, Morristown, NJ. From 1992 to 1998 he was an Assistant and later an Associate Professor at the



department of Electrical and Computer Engineering at the University of California, Santa Barbara. In 1998-1999 he was an Associate Professor at the Electrical Engineering department at Delft University of Technology, the Netherlands. In 1999 he became a Professor at the department of Computer Engineering and Informatics at the University of Patras, where he is director of the Hardware and Computer Architecture Division. He has received an NSF research initiation award and the 1st Prize in the national competition in Mathematics. He was the organizer of the 1998 Workshop on Communication networks and was in the program committee of several international conferences. His research activities are in the areas of protocols and algorithms for high-speed networks, all-optical networks, high-performance switch architectures, parallel and distributed computing, interconnection networks, VLSI layout design, performance evaluation, and ad-hoc networks.



Prof. Hercules Avramopoulos is currently heading the Photonics Communications Research Laboratory of the National Technical University of Athens (NTUA). He received his PhD from Imperial College, London University and from 1989 to 1993 he worked for AT&T Bell Laboratories, Holmdel, NJ, USA.

His primary research interest is in the demonstration and application of novel concepts in photonic technologies for telecommunications. He has worked on pulse generation, amplification and transmission in optical fibers as well as on a large number of laser and amplifier systems for a variety of applications. For the past 15 years he has been working on ultra-high speed, bitwise, all-optical, logic circuits and he has been interested to demonstrate their feasibility as a commercially viable technology for the telecommunications industry. Prof. Avramopoulos has been awarded 4 international patents in ultra high speed switching systems and has more than 100 archival journal publications and presentations in the major international conferences of the field.