Parallel Desynchronized Block Matching: A Feasible Scheduling Algorithm for the Input-Buffered Wavelength-Routed Switch

P. Pavon-Mariño[1], J. Garcia-Haro[1], A. Jajszczyk[2]
[1]*Department of Information Technologies & Communications, Polytechnic University of Cartagena, E-30202, Spain. Tel: +34 968 325952, Fax: +34 968 32 59 73*
Email: {pablo.pavon, joang.haro}@upct.es
[2]*Department of Telecommunications, AGH University of Science and Technology, Kraków, Poland*
Email: jajszczyk@kt.agh.edu.pl

Abstract

The Input-Buffered Wavelength-Routed (IBWR) switch is a promising switching architecture for slotted Optical Packet Switching (OPS) networks. The benefits of the IBWR fabric are a better scalability and lower hardware cost, when compared to output buffered OPS proposals. A previous work characterized the scheduling problem of this architecture as a type of matching problem in bipartite graphs. This characterization establishes an interesting relation between the IBWR scheduling and the scheduling of electronic Virtual Output Queuing switches. In this paper, this relation is further explored, for the design of feasible IBWR scheduling algorithms, in terms of hardware implementation and execution time. As a result, the Parallel Desynchronized Block Matching (PDBM) algorithm is proposed. The evaluation results presented reveal that IBWR switch performance using the PDBM algorithm is close to the performance bound given by OPS output buffered architectures. The performance gap is especially small for Dense Wavelength Division Multiplexing (DWDM) architectures.

Keywords: Optical Packet Switching, scheduling algorithms, performance evaluation.

1. Introduction

The Optical Packet Switching paradigm is similar to electronic packet switching, except that the packet payload is switched and buffered in the optical domain, while the packet header is processed electronically. In slotted OPS networks, packets are of a fixed size and are aligned at the inputs of the switching node. Slotted OPS, with packet duration on the order of 1 μs, has been quoted by the DAVID project as an envisaged switching alternative for the Wavelength Division Multiplexing (WDM) backbone network [1]. However, the optical switching and buffering of the packets implies a significant hardware cost with the current state-of-the-art of the photonic technology. For this reason, a commercial deployment of an OPS network is not envisioned in short term.

In the OPS backbone network, traffic flows are provisioned to follow a fixed sequence of hops from ingress to egress nodes. The Scattered Wavelength Path (SCWP) operational mode [2] means that the packet transmission wavelength in each hop is not fixed. Therefore, when a packet arrives at a switching node, its destination fiber is given by the information stored in the packet header; but the packet output wavelength is undetermined and has to be chosen dynamically. Consequently, a degree of freedom exists for the SCWP switch schedulers to take a joint decision on the packet delay and packet output

wavelength. This joint decision augments the statistical multiplexing effect, yielding lower delay and lower buffer requirements. By nature, this effect is particularly relevant in the DWDM scenario: the higher the number of wavelengths per fiber is, the higher the multiplexing gain.

The Input-Buffered Wavelength-Routed (IBWR) switch is an OPS switching architecture proposed in [3] and also studied in [4]. Figure 1 displays the design of an IBWR architecture adapted to operate in a WDM network, with N input/output fibers and n wavelengths per fiber. This means nN input and output ports to the switch. The IBWR switch consists of a buffering section connected to a non-blocking switching section. The buffering section contains M delay lines, of sizes from 0 to M–1 slots, nN Tunable Wavelength Converters (TWC) [5] with a tuning range of $\lambda_0...\lambda_{K-1}$, K=max (nN,M) and two $K\times K$ Arrayed-Waveguide-Gratings (AWG) [6]. The routing properties of AWG devices make packets from input port i leave the buffering section at the i-th output port, independently of the wavelength conversion applied. Wavelength conversion is then used to determine the delay line the packet will go through. The switching section consists of a set of nN TWCs, followed by a $nN\times nN$ AWG. The AWG device routes the packets to the appropriate output port of the switch, given by the combination of the packet output fiber and the packet output wavelength.

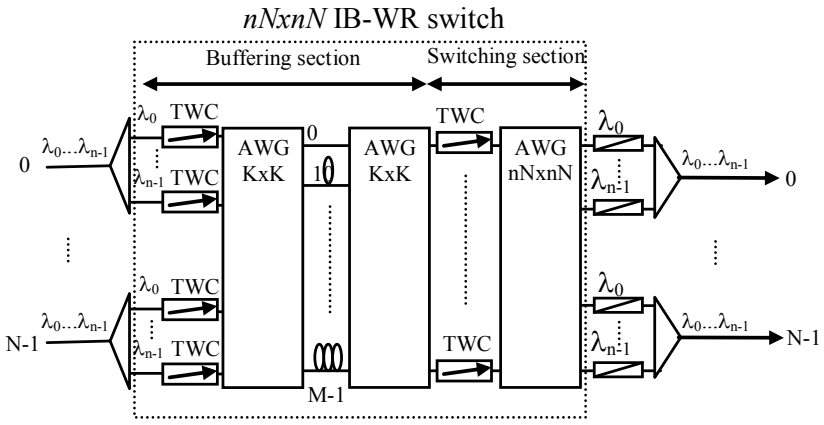


Fig. 1. Adaptation of the Input-Buffered Wavelength-Routed switch (IBWR) to the WDM networking environment.

The IBWR switch scheduler should decide on packet delay and packet output wavelength. Both decisions are independent and can be implemented by two separated processes.

Packet delay assignment: Packet delay allocation is performed within the time slot the packet enters the switch and cannot be changed. It is constrained by the following two conditions:

(i) *Output fiber contention*: At most n packets can be scheduled to leave the switch by the same output fiber and time slot, such that each of them can be assigned a different output wavelength.

(ii) *Input port contention:* Two packets arriving at the same input port in different time slots cannot be scheduled to leave the switch in the same time slot. This is because they would collide in the TWC of the switching section, since TWC devices can operate with only one packet at a time.

Therefore, the set of eligible delays for each input packet is the intersection of the allowed delays, according to both the input port contention and the output fiber contention. If no delay fulfills both conditions, the incoming packet is discarded.

Output wavelength assignment: The switch scheduler should spread the (at most) n simultaneous output packets per output fiber, among the n transmission wavelengths. This decision is taken independently of packet delay assignment, within the time slot the packet leaves the switch. In this paper,

packets transmitted in each output fiber are assumed to receive an output wavelength according to a pure round-robin scheme, as described in [7]. This methodology guarantees an equal average utilization of all wavelengths in the fibers.

The major benefits and interest of the IBWR architecture lay in the lower cost and better scalability of its hardware, when compared to OPS architectures capable of emulating output buffering [3][8][9]. These are switch fabrics for which delay allocation is only constrained by the (unavoidable) output fiber contention. A previous work evaluated the buffering requirements of output buffered fabrics in a SCWP network [10]. For example, results show that a buffer depth of only two delay lines is required to provide a packet loss probability of 10^{-9} in an output buffered OPS switch of 32 wavelengths per fiber, under a Bernoulli input traffic load of ρ=0.8. For a slot time of 1 μs, this can be implemented by two delay lines of lengths 0 m (cut-through) and 200 m. When we compare the IBWR and output buffered alternatives, the following trade-off arises. On the one hand, the extra set of input port contention constraints, present in the IBWR delay assignment problem (set of constraints (ii) above), must imply some performance degradation. On the other hand, the hardware complexity of the IBWR architecture is much lower than the one for the output buffered proposals.

Table I illustrates the cost side of this complexity vs. performance trade-off, for three prominent OPS output buffered architectures: the KEOPS switch [9], the Output-Buffered Wavelength-Routed switch [3] and the space switch [8]. The comparison involves switch fabrics of N input and output fibers, n wavelengths per fiber and M buffer positions. Interestingly, IBWR architecture hardware is not based on optical gates, and the number of wavelength converters it requires grows linearly with the switch size.

The aforementioned cost vs. performance trade-off, raises our interest in the following question: Which are the design criteria for feasible delay assignment algorithms in IBWR switches to achieve an acceptable performance? This paper is intended to answer this question. Our research is based on the work in [11], where the IBWR delay assignment scheduling was characterized as a matching problem in bipartite graphs. We focus on the interesting relation established between the IBWR scheduling and the scheduling of the Virtual Output Queuing (VOQ) high-performance electronic packet switches. The analysis of the similarities and differences between both optimization problems is the basis of the proposal of the Parallel Desynchronized Block Matching (PDBM) algorithm. As far as the authors know, this algorithm is the first attempt to design a *feasible* scheduling algorithm for delay assignment in an IBWR switch. This means an algorithm which allows a practical electronic implementation that provides an execution time bounded at most by the slot time.

The rest of the paper is organized as follows. Section 2 analyzes the optimization problem to solve for the delay assignment in IBWR switches. Section 3 presents the PDBM algorithm. Comparative evaluation results are provided in Section 4. Section 5 concludes.

TABLE I
COST COMPARISON OF OPS SWITCHING ARCHITECTURES

Switch	FWC	TWC	TWC TUNING RANGE	OPTICAL GATES
KEOPS switch	$2nN$	0	---	$MnN+n^2N^2$
OBWR switch	nN	nN	$\max(nN,M)$	n^2N^2
Space switch	0	nN	n	nN^2M
IBWR switch	nN	$2nN$	$\max(nN,M)$	0

FWC: Fixed Wavelength Converters
TWC: Tunable Wavelength Converters

2. Delay assignment in the IBWR switch

2.1 Problem definition

This section extends the work presented in [11], characterizing the delay assignment problem that the IBWR scheduler has to solve, *within one time slot.* Section 2.2 deals with the dynamics of this problem in consecutive time slots. From now on, we concentrate on an IBWR switch fabric with N input and N output fibers, n wavelengths per fiber, and M delay lines, like the one shown in Figure 1. We denote T as the *current* time slot. For each time slot, the switch state information can be expressed by two sets of vectors:

- ***nN* Input contention vectors**

One vector input contention vector $\overline{X_i}$ exists per input port (i=0,…,nN-1), that is, one vector per input wavelength and fiber.

$$\overline{X_i} = \left(x_i(0),...,x_i(M-1)\right), i = 0,...,nN-1$$

Coordinate $x_i(t)$, t=0,…,M-1 takes the value of 1 if a packet is scheduled to leave the buffering section through the i-th port, in time slot $T+t$. It is 0 otherwise. A value of $x_i(t)$=1 implies that delay t is not eligible for assignment to a packet present at input port i.

- ***N* Output contention vectors**

One output contention vector $\overline{Y_j}$ exists per output fiber (j=0,…,N-1).

$$\overline{Y_j} = \left(y_j(0),...,y_j(M-1)\right), j = 0,...,N-1$$

Coordinate $y_j(t)$, t=0,…,M-1 symbolizes the number of packets destined to output fiber j, which have been scheduled to leave the switch in time slot $T+t$. Therefore, the number of packets destined to output fiber j, which can be assigned a delay t, is limited to $n-y_j(t) \in \{0,...,n\}$. We define this value as *availability of delay t, in output fiber j.*

For an arbitrary packet at the i-th input port destined to the j-th output fiber, the status of $\overline{X_i}$ and $\overline{Y_j}$ vectors should be inspected. The packet can be assigned a delay t only if $x_i(t) = 0$ and $y_j(t) < n$. Accordingly, a packet is discarded if none of the time slots t' without input port contention, matches the time slots without output contention for the destination fiber.

Once delay assignment in a time slot is finished, the state vector for the succeeding time slot should reflect the propagation of the packets across the fiber delay lines, and remove the packets that leave the buffering section. This can be performed by a simple coordinate shift in every $\overline{X_i}$ and $\overline{Y_j}$ vector (i=0,…,nN-1, j=0,…,N-1): (i) coordinate t of each state vector is stored in coordinate t–1, t=1,...,M–1, (ii) coordinate M–1 of each state vector is reset to 0.

The scheduling problem to be solved in one time slot can be expressed as a type matching problem in a bipartite graph [12]. The graph is constructed as follows:

- nN left side nodes, one for every input port.
- NM right side nodes, one for every delay of every output fiber.

- An arc between left side node i=0,...,nN–1 and right side node (j,t), j=0,...,N–1, t=0,...,M–1, if a packet is present at input port i, destined to output fiber j, and satisfies the input port contention constraint ($x_i(t) = 0$).

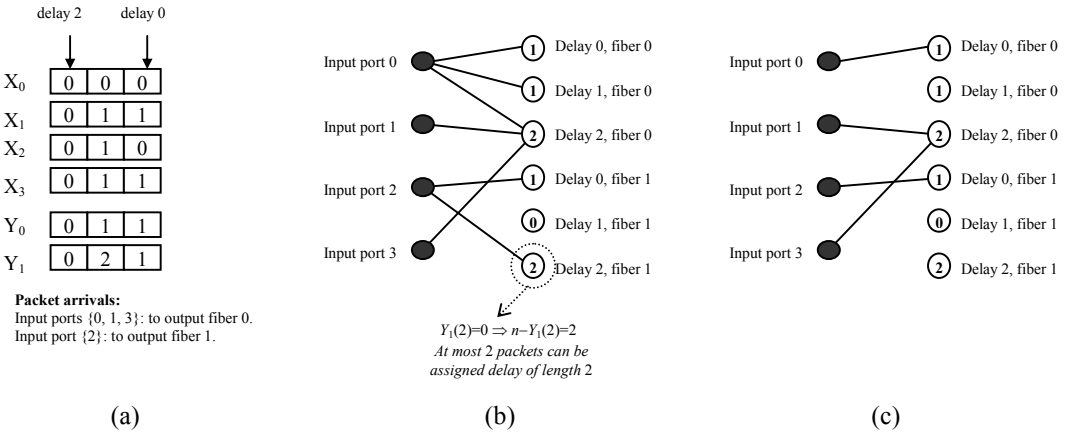


Fig. 2. (a) Example of state vectors and packet arrivals. (b) Bipartite graph depicting the assignment problem associated to switch scheduling. (c) availability matching to problem (b).

Feasible solutions to the scheduling problem given by a bipartite graph G are represented as a subgraph G' for which: (1) each left side node is connected with at most one arc, (2) each right side node t is connected with at most $n - y_j(t) \in \{0,...,n\}$ arcs (the delay availability). In this paper, we designate a subgraph representing a feasible solution to graph G, as an *availability matching* of graph G. Figure 2(b) exemplifies this. It shows the bipartite graph associated to the state vectors and packet arrivals depicted in Figure 2(a). To simplify the graphical representation of the assignment problem, the availability of each delay node t, $n - y_j(t)$, is drawn inside the node. A feasible solution to this assignment is represented by the graph of Figure 2(c). Note that any availability matching problem can be seen as a type of (1,k)-matching problem in bipartite graphs, where at most one arc can leave left side nodes, and the number of arcs k which can leave each right side node is equal to the node availability, which varies across right side nodes, and along time slots.

Note that for a given time slot T, delay assignment decisions for packets in different input ports destined to the same j-th output fiber share $\overline{Y_j}$ information, and are jointly affected. But decisions for packets destined to different output fibers are *independent*. In fact, the delay assignment in one time slot can be divided into N separated problems, one per output fiber. In the bipartite graph representation, this means that the graph G representing an IBWR delay assignment problem can be partitioned into N *unconnected* subgraphs $G_0,...,G_{N-1}$. Each subgraph G_i is composed by the arcs destined to the delays in output fiber i. For every G_i and G_j subgraphs, G_i and G_j are unconnected, as left side nodes (input ports) with an arc in G_i graph (an arriving packet destined to output fiber i) do not have an arc in G_j graph.

2.2. Switch scheduler design

We can define a *switch scheduler* as an algorithm executed every time slot, which selects among the possible availability matchings; the one which maximizes a specified objective function f. The desired

property for an objective function f is that every time slot should converge into the feasible assignment decision which provides the highest switch performance. The performance metrics considered are the packet loss probability and the average packet delay. The following issue arises: while switch scheduler takes an assignment decision every time slot, according to current switch state information, the switch performance is defined as a temporal average measure. But, which among the feasible assignments in a given time slot contributes to the highest average performance? This question is relevant, as optimum assignments in one time slot can lead to suboptimum assignments in future time slots and vice-versa.

A formal analysis to address this question leads to a dynamic integer programming problem, with stochastic variables and linear constraints. There are no general mathematical tools to solve this type of problems.

To obtain valuable design criteria for the searched objective function, the following considerations are issued:

- *Fact 1:* For each time slot, packets which are not assigned a delay are discarded.
- *Fact 2:* A packet which is assigned a delay D, contends with future input packets in the subsequent D time slots. The longer the delay assigned to a packet is, the longer the packet will occupy switching resources.

In this paper, minimization of instantaneous packet loss is the main concern. As a consequence, we design our objective function f as the one which (i) converges to the feasible solution of the maximum size, i.e., the maximum number of assignments, (ii) if more than one feasible solution of the maximum size exists, the objective function should give preference to the one which minimizes the average packet delay. The resulting optimization problem is as follows:

$$\text{Find assignment } D_{i,t} \text{ which maximizes } f = \sum_{i=0}^{nN-1}\sum_{t=0}^{M-1} D_{i,t},$$

$$\text{and after that minimizes } \sum_{i=0}^{nN-1}\sum_{t=0}^{M-1} t \cdot D_{i,t},$$

Constrained to

$$(i) \quad D_{i,t} \le R_{i,t}, \forall i = 0,...,nN-1, \forall t = 0,...,M-1 \tag{1}$$

$$(ii) \quad \sum_{t=0}^{M-1} D_{i,t} \le 1, \forall i = 0,...,nN-1$$

$$(iii) \quad \sum_{i=0, A_{i,j}=1}^{nN-1} D_{i,t} \le n - Y_j(t), \forall t = 0,...,M-1, j = 0...N-1$$

The newly introduced variables are explained below:

- Solution matrix $[D]_{nN\text{x}M}$, $D_{i,t}=\{0,1\}$,i=0,…,nN-1, t=0,…,M-1: $D_{i,t}$=1 if a packet in input port i is assigned a delay t. $D_{i,t}$=0 otherwise.
- Arrivals matrix $[A]_{nN\text{x}N}$, $A_{i,j}=\{0,1\}$, i=0,...,nN–1, j=0,...,N–1: $A_{i,j}$ includes the information on the packet arrivals at the switch in the current time slot. $A_{i,j}$=1 if a packet at input port i is destined to output fiber j. $A_{i,j}$=0 otherwise.
- Request matrix $[R]_{nN\text{x}M}$, $R_{i,t}=\{0,1\}$, i=0,...,nN–1, t=0,...,M–1: $R_{i,t}$=1 if a packet at input port i is

destined to output fiber j ($A_{i,j}$=1) and delay t is selectable for this packet, attending to input port contention ($x_i(t)$=0). $R_{i,t}$=0 otherwise.

In expression (1), the set (i) of nNM constraints means that an input port i can be assigned a delay, only if a packet is present in that port destined to output fiber j and also, if the delay is available due to the input port contention. The set (ii) of nN constraints states the fact that a packet can be assigned at most one delay. The set (iii) of NM constraints represents the output fiber contention and guarantees that, for every output fiber j, the number of packets scheduled to leave the switch in any time slot do not exceed the number of wavelengths n.

The global optimum to the matching problem defined by (1) can be achieved by implementing the modified Edmonds-Karp Maximum Size Matching (MSM) algorithm proposed in [13]. This algorithm obtains the matching of the maximum size, giving preference to the arcs destined to lower delay nodes, when performing path augmentation [12]. Unfortunately, the complexity of the most efficient algorithm known to date obtaining the optimum MSM solution of an $N \times N$ bipartite graph is $O(mlog(m))$, being m the number of edges [14]. Furthermore, MSM algorithms are very hard to implement in hardware and cannot operate at high speeds.

As a conclusion, in our particular scenario, the search for the global MSM optimum solution to the matching problem given should be avoided.

3. PDBM algorithm

3.1. Related work

In synchronous slotted OPS networks, the scheduling response time is limited by the optical packet duration, because one scheduling decision (involving all simultaneous arriving packets) should be taken every slot time (on the order of 1 μs). However, algorithm implementations with response time far below the slot time are of interest to simplify system integration. In the IBWR switch, the payload of input packets has to be delayed after header detection and before entering the buffering section, to assure that packet payload arrives at the TWC of the input port, when it is correctly tuned to determine the packet delay. For this purpose, a fiber delay line of a sufficient length is used at every input port. TWC devices with tuning times on the order of nanoseconds have been demonstrated [9]. Scheduling algorithms on the order of tens of nanoseconds would lead to short fiber delays (or even no delays at all).

As far as the authors know, the only proposal of the delay assignment algorithm for an IBWR architecture operating under the SCWP operational mode was presented in [11]. This algorithm was conceived for testing purposes, to make a first performance evaluation of the SCWP IBWR architecture. The algorithm completes a sequential check of all input ports, selecting for each input packet, the shortest delay which satisfies the input and output contention constraints. This sequential algorithm is useless in a practical electronic implementation. The sequential operation implies an algorithm response time depending on the switch size, which makes it impossible to fulfill the time constraints even for moderate switch sizes.

Fortunately, the description of the optimization problem to be solved as a matching problem in bipartite graphs, can facilitate the search for practical scheduling algorithms. The design of fast algorithms to *approximate* the maximum matching in bipartite graphs, by giving a *maximal* solution (i.e., a local maximum), has been a hot research topic in the last decade. Specifically, it has been the subject of study for the scheduling of high-performance Virtual Output Queuing (VOQ) *electronic* packet switches, first introduced by Tamir *et al.* in [15]. Several VOQ scheduling algorithms, which allow an implementation based on feedback combinational circuits and parallel architectures, have been proposed [16]-[23]. As a result, response times on the order of tens of nanoseconds are attained.

The differences between IBWR and VOQ scheduling are summarized as follows:

- The matching problem per time slot in IBWR schedulers can be decomposed into N independent problems, one per output fiber. This is not the case of VOQ scheduling.
- In VOQ bipartite graphs, one right side node exists per output port. All right side nodes may have the same importance to guarantee system fairness. In IBWR graphs, each right side node represents a delay and IBWR schedulers should prioritize right side nodes associated with lower delays.
- VOQ schedulers must guarantee that each output port is assigned to at most *one* packet. In SCWP IBWR matchings, the availability of a delay line limits the number of packets that can be assigned. Furthermore, delay availability varies for different delays and different time slots.
- In VOQ switches, the packets which do not find an output port remain stored in the queues. In IBWR switches, the packets which are not assigned a delay are discarded.
- In VOQ switches, the evolution of the bipartite graphs in subsequent time slots is highly correlated (i.e., VOQ queues with more than one packet remain occupied). Nevertheless, in the IBWR optimization problem, bipartite graphs corresponding to subsequent time slots are highly variable.

However, in the authors' opinion, the underlying design criteria in the basis of VOQ scheduling algorithms can be very useful for the design of feasible IBWR schedulers. Therefore, we can take benefit from the experiences in the VOQ scheduling field. The design of the Parallel Desynchronized Block Matching (PDBM) scheduling algorithm, presented in this section, is a sample of this. Specifically, PDBM adapts implementation concepts present in the *parallel-iterative* VOQ schedulers [18], [20] and especially [21].

3.2. Algorithm description

Figure 3 shows the electronic implementation proposed for the PDBM scheduler. It is based on an interconnection of nN input modules (one per input port) and NM output modules (one per delay of each output fiber). The control information required for the algorithm execution is distributed across the I/O modules.

- A register in input module i, i=0,...,nN–1, maintains the state vector $\overline{X_i}$ of length M bits. Every time slot, these registers are shifted to reflect the propagation of the packets along the delay lines.
- Output module (j,t), j=0,...,N–1, t=0,...,M–1, contains (i) the delay availability register $n–y_j(t)$, of length $log_2(n)$ bits, (ii) the *grant pointer* $G(j,t)$, of length $log_2(nN)$ bits, and (iii) a clockwise/counter-clockwise bit $CW(j,t)$. Note that every time slot, the availability register in module (j,t) must be transferred to module (j,t–1), j=0,...,N–1, t=1,...,M–1. Also, modules (j,M–1), j=0,...,N–1, reset the availability registers to the value of n.

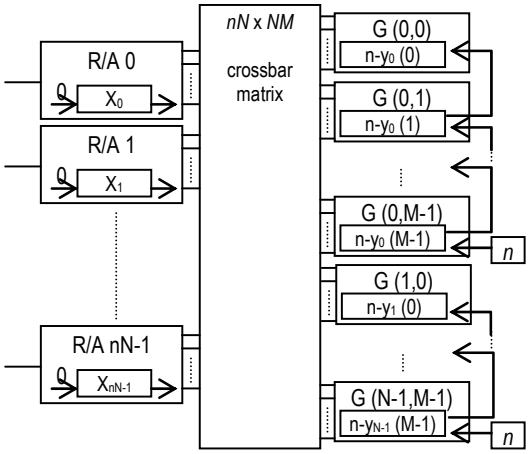


Fig. 3. Electronic implementation scheme for the PDBM scheduler.

3.2.1. Algorithm iteration

The PDBM algorithm is designed as an iterative algorithm. In every time slot, a bounded number of iterations of the algorithm are executed. Each algorithm iteration consists of three phases, namely: *request, grant,* and *accept.*

- *Step 1. Request:* Executed in parallel, in each of the *nN* input modules. For input module *i* with a packet destined to output fiber *j*, a request signal is sent to every delay of output fiber *j*, which does not violate the input contention constraint. That is, output modules (*j,t*) for which $x_i(t)=0$. Note that at most *M* request signals are created per input module.
- *Step 2. Grant*: Executed in parallel, in each of the *NM* output modules. For output module (*j,t*), request signals from input modules are scanned, starting by the input module directed by the grant pointer *G*(*j,t*). The scanning of the other input modules continues in a clockwise or counter-clockwise procedure, as indicated by the *CW*(*j,t*) bit. The first $n-y_j(t)$ scanned request signals are granted, and a grant signal is sent to the associated input module. Therefore, a *block of grants*, of a size limited by the delay availability, is produced.
- *Step 3. Accept:* Executed in parallel, in each of the *nN* input modules. Each input module *i* receives at most *M* grants, from the *M* delays associated with the destination output fiber. The grant associated with the shorter delay *t*, if any, is accepted. Packet present at that input port is assigned a delay *t*. An accept signal is sent to the accepted output module. Output modules update their availability state to reflect packet allocation.

Packets allocated in one iteration of a time slot are not involved in subsequent iterations of the same time slot. Therefore, the delay assignment process is incremental along subsequent iterations. If no assignments are produced in one iteration of a time slot *T*, no assignments are produced in the following iterations of *T*. We define PDBM convergence as the number of iterations where at least one packet is allocated and thus, assignment size is improved. The algorithm convergence will be studied in Section 4.4.

After the last iteration, for every time slot, *CW*(*j,t*) bits and *G*(*j,t*) grant pointers are updated as follows:

- $CW(j,t)$ bits, j=0,...,N–1, t=0,...,M–1, are inverted. Therefore, the request scanning direction changes every time slot, and it is the same during all the iterations of a given time slot.
- $G(j,t)$ grant pointers, j=0,...,N–1, t=0,...,M–1, are incremented by one, (modulo nN), every *two* time slots. This occurs independently of the number of grants produced. Consequently, the relative distance (modulo nN) among grant pointers is always kept.

3.2.2. System initialization

During equipment start-up, $\overline{X_i}$ and $\overline{Y_j}$ vectors (i=0,…,nN-1, j=0,…,M-1) that store delay lines occupancy are reset to 0. Clockwise flags CW of all output modules are also reset. Grant round-robin pointers are initialized as follows:

- M grant pointers associated with the same output fibers are initialized to point at different input ports. We denote this situation as *pointers desynchronization*. In our tests, the pointers are initialized to maximize the minimum distance between pointer positions, as given by (2).

$$\begin{aligned} &G(f,0) = 0 \quad \forall f = 0...N-1 \\ &G(f,t) = G(f,t-1) + \min\left(1, \left\lfloor \frac{nN}{M} \right\rfloor\right) \quad \forall t = 1...M-1 \end{aligned} \tag{2}$$

- Grant pointers associated with different output fibers can be initialized in any form. In our tests, pointers associated with different output fibers are equal $G(f_1,t)=G(f_2,t)\ \forall\ t, f_1, f_2$.

3.3. Algorithm justification

In the PDBM algorithm, the delay assignment behavior is determined by the actions performed during the grant phase. In this stage, each output module (j,t) receives a number of requests $r(j,t)$. If the number of requests is greater than the delay availability, only a subset of these requests is granted. The input modules granted by module (j,t) are those closer to the grant pointer $G(j,t)$, in accordance to the scanning direction. Input modules which are "far" from the grant pointer (j,t) may not receive a grant from this delay module.

In this situation, it is of interest that input modules which do not receive any grant from module (j,t), may receive a grant from any other output module (j,t'), $t' \neq t$. This depends on the positions of the grant pointers in the *other output modules associated with the same output fiber*. If grant pointers had the same position (synchronized), closer input modules would receive more than one grant, while far input modules would receive no grant in this iteration. We call this effect *grant block overlapping*.

The objective of PDBM is to minimize grant blocks overlapping. The employed method is an adaptation of the pointers desynchronizing scheme proposed in RDSRR (Rotating Double Static Round Robin) VOQ algorithm [21]: pointers are desynchronized during system start-up, and move periodically, independently of packet arrivals, allowing a simpler hardware implementation. In PDBM, grant pointers associated with the same output fiber are desynchronized during system start-up. After that, pointers desynchronizing is maintained by simultaneously incrementing (modulo nN) the positions of all pointers every two time slots. Therefore, all grant pointers maintain the same desynchronizing scheme determined during system start-up.

During initialization, the positions of the pointers associated with the same output fiber are spread

across the nN input modules, such that the minimum distance (modulo nN) between two nodes is maximized. The objective is to decrease the chances of block overlapping. Of course, actual overlapping depends on traffic conditions. Grant positions associated with a delay in different output fibers are not subject to grant block overlapping. For this reason, in our tests, they are set to equal values: $G(f_1,t)=G(f_2,t)$ $\forall\ t, f_1,f_2$.

The objective of the per-time-slot rotation in the scanning direction is to provide a fair operation when packet arrivals are not uniform across input fibers. Let us analyze the following example. We consider an IBWR switch with 4 input fibers, where packet arrivals occur only in the two first fibers f_0 and f_1. If the scanning direction is the same every time slot, input ports associated to f_0 would be prioritized to be assigned the lower delay 0, when the grant pointer directs empty input fibers f_2 and f_3 . On average, ports in f_0 would be given priority over ports in f_1, every 3 of 4 time slots. Therefore, the system performance perceived by traffic in input fibers f_0 and f_1 is different. By rotating the scanning direction every time slot, the fairness is improved.

3.4. Convergence of the algorithm

The PDBM algorithm searches a maximal (local maximum) solution to the optimization problem described in Section 2: (i) it searches a maximal size matching, (ii) and the result is also a local optimum to the minimum average delay, as lower delay grants are the ones accepted. The process is performed in parallel, by operating in the scheduling bipartite graph G.

Because of the incremental packet allocation, PDBM actually achieves a maximal solution if the algorithm executes enough iterations to converge, such that subsequent iterations do not increase the match. Obviously, the algorithm convergence also concerns algorithm response time and algorithm implementation. Fortunately, PDBM convergence can be guaranteed.

Property 1: The PDBM algorithm converges in at most min (nN, M) iterations, for an IBWR switch of N input fibers, n wavelengths per input fiber and M delay lines.

Proof: PDBM searches in parallel in a graph which can be partitioned into N unconnected bipartite graphs, one per each output fiber. Each partition has nN left side nodes and M right side nodes. In the worst case, one new arc is added to each partition per iteration. This implies a bound of min(nN, M) iterations to converge.

It should be noted that the algorithm convergence bound is *independent of the switch size* when the number of delay lines is lower than the switch size. This occurs for medium to large size switches, as shown later in Section 4.

4. Performance evaluation results

The performance of the PDBM algorithm has been evaluated by means of simulation (the Batch Means method [24], 99% confidence intervals, 1% tolerance, upper limit $5\cdot10^7$ time slots). In all cases, the results are compared to the performance bound provided by the OPS switches able to emulate output buffering (which we denote as OB architectures).

4.1. Traffic patterns in OPS SCWP networks

There is no specific research work regarding packet traffic patterns in SCWP OPS networks. Conventional sources of traffic, employed in the evaluation of electronic packet switches, cannot be

directly applied for the evaluation of SCWP OPS networks. This is because the packet transmission wavelength is decided in SCWP networks by a switch (or edge node) scheduler. The particular manner in which the traffic in a fiber is spread across the wavelengths is the source of a statistical correlation among traffic processes in different wavelengths *of the same fiber*. Previous evaluation research of OPS switching architectures did not consider this intrinsic correlation for the injected traffic. In most of the situations, independent Bernoulli sources were considered for each input port. In [8], the performance of the space switch (and thus, applicable to any OB switch) was evaluated, assuming that each input port receives an independent ON-OFF bursty source. Again, dependence among input ports associated with the same input fiber was not considered.

In this paper, we try to employ more realistic traffic sources. We create a *n-SCWP traffic source* to model the packet arrivals in an input fiber consisting of *n* wavelengths. As shown in Figure 4, we define a *n-SCWP* source, as the combination of (i) a feeding packet source, creating up to *n* packets per time slot, and (ii) a wavelength dispatcher, which distributes the generated packets across the *n* wavelengths in the fiber. The IBWR architecture is evaluated under two different types of traffic, namely:

- *n-SCWP uniform Bernoulli traffic*. A Bernoulli feeding source of parameter $\rho \leq 1$ is used. On average, $n\rho$ packets are created every time slot. Destination fiber of each packet is uniformly distributed.
- *n-SCWP uniform ON-OFF traffic*. An ON-OFF feeding source, modulated by a 2-state Markov chain is used, as the one described in [25]. Traffic load is given by parameter $\rho \leq 1$. On average, $n\rho$ packets are created every time slot. Average length of ON periods is given by β. The feeding source creates a packet with probability 1 during the ON periods. All packets in a burst have the same destination fiber, which is uniformly selected.

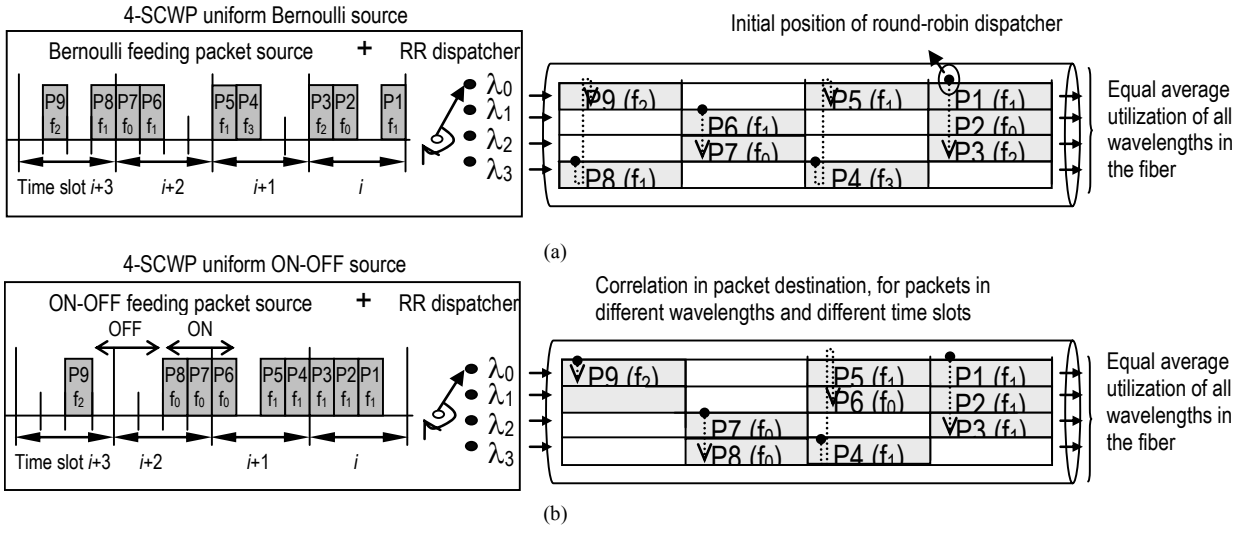


Fig. 4. Example of n-SCWP packet sources. (a) 4-SCWP uniform Bernoulli source, (b) 4-SCWP uniform ON-OFF source.

The employed wavelength dispatcher in all cases implements a round-robin distribution of packets [7]. Figure 4(a) illustrates the effect of this distribution in a *n*-SCWP source fed by a Bernoulli packet source.

The selected destination fibers (f_0,...,f_3 in the figure) are independent among generated packets. As a consequence, destination fibers of packets arriving at different wavelengths of the same fiber are also independent. Figure 4(b) exemplifies a *n*-SCWP ON-OFF source. Note that in this case, the correlation in the destination fibers, present in the feeding source, is transformed into a correlation in the destination fibers of simultaneous and non-simultaneous packets in different wavelengths.

4.2. Results

Figures 5(a) and 5(b) depict the average delay performance of the PDBM algorithm under *n*-SCWP Bernoulli traffic. Buffering is dimensioned for a negligible packet loss probability. Switch sizes considered are N={2,4} input and output fibers, and n={2,8,32,64} wavelengths per fiber. Results obtained for higher values of N={6,8}, not included in the paper, do not differ from the ones shown.

Results reveal that PDBM performance is very close to the OB architectures performance. In all our tests, average delay measured is below 2 time slots. Only two exceptions occur, for a 90% input load and n=2 wavelengths per fiber, where average delays obtained were 2.40 (N=2) and 4.44 (N=4) time slots. Results are especially encouraging in the DWDM environment. The average delay is below 1 time slot, even for high traffic loads, and approaches the OB bound.

Table II comparatively evaluates the buffering requirements to achieve a packet loss probability below 10^{-7} (simulation length is limited to 10^9 packets in this test). Results show that the performance gap between both IBWR and OB architectures is small, especially under the DWDM paradigm.

Figure 5(c,d) shows the average delay performance under ON-OFF *n*-SCWP traffic. Switch parameters are $N = 4$ input and output fibers, n = {2,8,32,64} wavelengths per fiber. Buffer sizes of the switches are the same for IBWR and OB architectures, M = {35,10,3,2}, for parameter n = {2,8,32,64}, respectively. This parameter was dimensioned using the performance of OB architectures under Bernoulli traffic as a reference. The selected buffering depth provides a packet loss probability below 10^{-9} under 90% Bernoulli load in OB architectures. Burst length parameters under test are $\beta = 16$ (Figure 5(c)) and $\beta = 64$ (Figure 5(d)). Figure 5(e) presents the packet loss probability obtained in both situations.

The performance degradation observed under bursty traffic is notable in all circumstances. Average delay grows, especially for low values of parameter n. Again, DWDM architectures achieve a much better performance, also diminishing the gap between the PDBM and OB architectures. The packet loss probability obtained is especially high because the correlation among input ports produces a large number of simultaneous packets with a common destination. The behavior of the IBWR switch in this situation is governed by the output contention. This is the reason to observe an almost equal performance of the IBWR and OB architectures. The critical deterioration in the observed packet loss performance is an interesting point to take into consideration. In the authors' opinion, it reveals the need of mechanisms to split traffic bursts in SCWP networks. This is because the peculiar spreading of traffic among fiber wavelengths of SCWP networks may produce a higher number of simultaneous arrivals of packets with a common destination.

Tables III and IV summarize the practical results obtained in the PDBM algorithm convergence tests. They describe the number of iterations K, such that the algorithm does not converge in at most 1 out of 10^6 time slots. Only results for 90% input loads are shown. Results are compared to the theoretical convergence bound min (nN, M). For Bernoulli traffic, practical algorithm convergence is reached in two or three iterations. For ON-OFF traffic, a higher number of iterations are required. This effect can be intuitively explained as follows. As shown, ON-OFF bursts of traffic are transformed into a set of consecutive input ports with a common output fiber. Because of the proximity of the input ports, there are a lot of chances for the grant block overlapping: a large set of input ports may fall into the same grant blocks, receiving grants from different delays, while other input ports receive no grants. In subsequent

iterations, assigned input ports are not involved, but no assigned ports remain close, maintaining the chances of block overlapping.

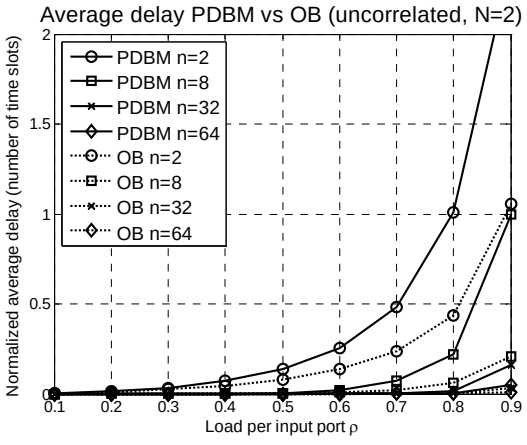
5. Conclusion and future work

The benefits of the IBWR switching architecture are its lower cost and better scalability when compared to output buffered proposals. The scheduling of this architecture is characterized as a type of matching problem in bipartite graphs. The study of the similarities and differences with the scheduling of VOQ switches allowed us to propose the PDBM scheduling algorithm. As far as we know, it is the first feasible scheduler proposed in the field, meeting the requirements related to scheduling response time and hardware complexity. The expected response time and hardware complexity of the PDBM scheduler are similar to that of the commercial iterative iSLIP-like VOQ schedulers.

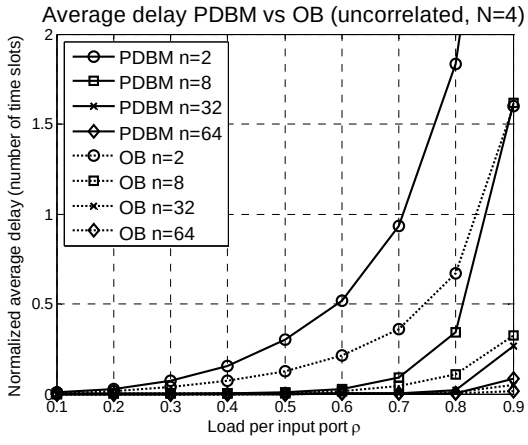
We have evaluated the performance of PDBM algorithm, comparing it to the OB proposals. For this purpose, we introduced more realistic sources of traffic for SCWP networks. Results are promising, as the performance gap is small, especially in the DWDM scenario.

Interestingly, the large-scale version of the IBWR architecture (also presented in [3]) defines exactly the same scheduling problem as the IBWR switch analyzed in this paper. Therefore, both versions of the switch can be governed by the same scheduler. This raises the interest in scheduling algorithms like PDBM, whose execution time can be independent of the switch size. The hardware scalability is expected to be good in the DWDM scenario. This is because, the number of grant arbiters required in the scheduler is equal to the number of output fibers, and the number of fiber delays, but is not affected by the number of wavelengths per fiber.

The work presented in this paper does not consider the packet sequence issue. The PDBM scheduler does not preserve the order between arriving packets. Currently, a comparative study is being carried out to evaluate a set of techniques that address the packet order problem in the IBWR architecture.



(a)



(b)

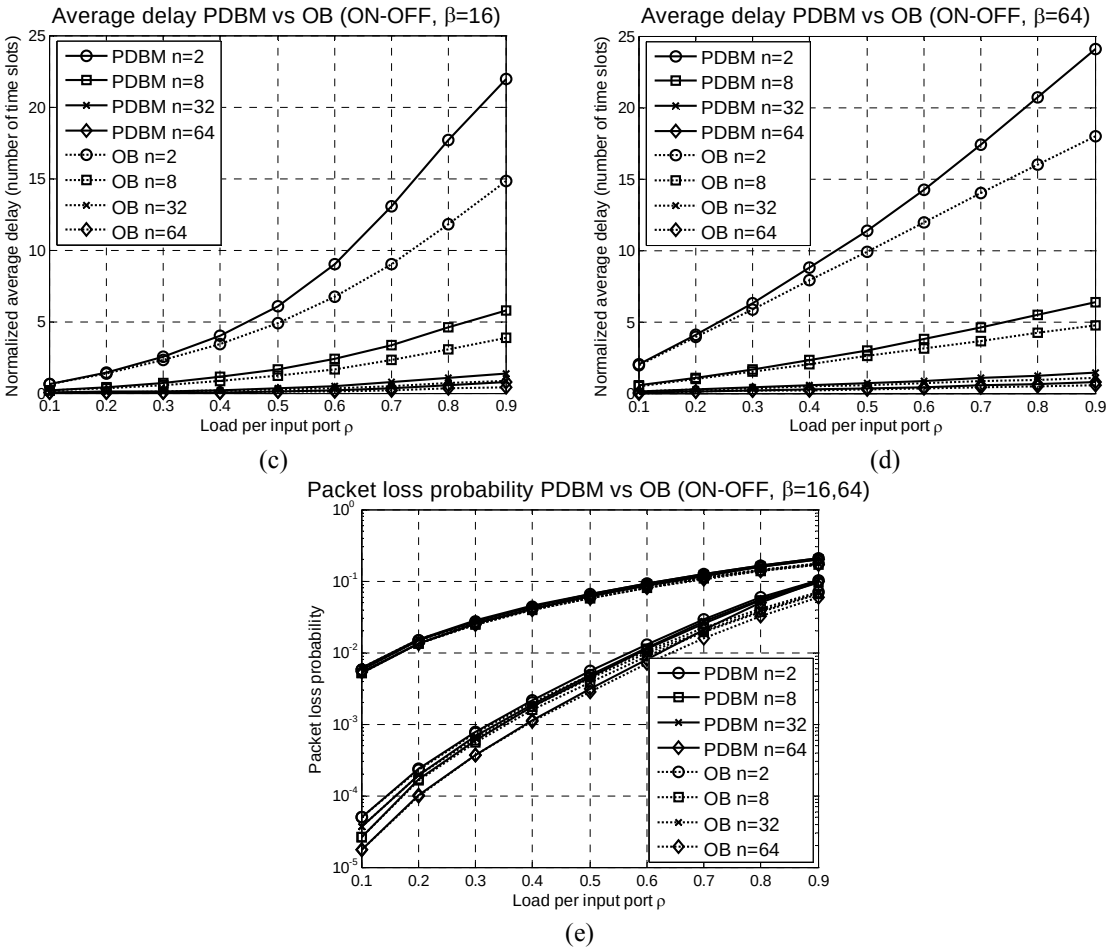


Figure 5. (a) and (b): Average delay performance under *n*-SCWP Bernoulli traffic. (a) *N*=2, (b) *N*=4. (c) and (d): Average delay performance under *n*-SCWP MMPP traffic (c) β=16, (d) β=64. (e) Packet loss probability (PLP) performance under *n*-SCWP MMPP traffic, β=16 (set of lines with lower PLP), β=64 (set of lines with lower PLP).

TABLE II
BUFFER REQUIREMENTS OF IBWR VS OUTPUT BUFFERED ARCHITECTURES. BERNOULLI INPUT TRAFFIC, 10^{-7} PACKET LOSS PROBABILITY

Switch size	ρ=0.1	ρ=0.2	ρ=0.3	ρ=0.4	ρ=0.5	ρ=0.6	ρ=0.7	ρ=0.8	ρ=0.9
N=2,n=2	4 / 2	4 / 3	4 / 3	5 / 4	6 / 5	7 / 5	8 / 7	11 / 10	20 / 18
N=2,n=8	1 / 1	3 / 2	3 / 2	4 / 2	4 / 2	5 / 2	6 / 3	7 / 3	9 / 6
N=2,n=32	1 / 1	1 / 1	1 / 1	1 / 1	3 / 2	3 / 2	4 / 2	4 / 2	5 / 2
N=2,n=64	1 / 1	1 / 1	1 / 1	1 / 1	1 / 1	1 / 1	3 / 2	3 / 2	4 / 2
N=4,n=2	5 / 3	5 / 3	6 / 4	7 / 5	8 / 6	10 / 7	13 / 9	19 / 14	30 / 26
N=4,n=8	1 / 1	3 / 2	3 / 2	3 / 2	4 / 2	4 / 3	5 / 3	8 / 4	13 / 8
N=4,n=32	1 / 1	1 / 1	1 / 1	1 / 1	3 / 2	3 / 2	4 / 2	4 / 2	5 / 3
N=4,n=64	1 /1	1 / 1	1 / 1	1 / 1	1 / 1	3 / 2	4 / 2	4 / 2	5 / 2

TABLE III
PRACTICAL NUMBER OF ITERATIONS TO CONVERGE VS. CONVERGENCE BOUND, FOR BERNOULLI TRAFFIC

ON-OFF ρ=0.9	n=2	n=8	n=32	n=64
N=4	1 / 4	2 / 9	2 / 5	2 / 4
N=2	2 / 8	3 / 13	2 / 5	2 / 5

TABLE IV
PRACTICAL NUMBER OF ITERATIONS TO CONVERGE VS. CONVERGENCE BOUND, FOR ON-OFF TRAFFIC

ON-OFF ρ=0.9, N=4	n=2	n=8	n=32	n=64
β=16	5 / 8	6 / 10	3 / 3	2 / 2
β=64	4 / 8	6 / 10	3 / 3	2 / 2

Acknowledgements

This research has been funded by the Spanish MCyT grant TEC2004-05622-C04-02/TCM (ARPaq). Authors would like to thank also the COST 291 action and the e-Photon/ONe+ European Network of Excellence.

References

[1] L. Dittman, *et al*., "The European IST Project DAVID: A Viable Approach Toward Optical Packet Switching", *IEEE Journal on Selected Areas in Communications,* vol. 21, no. 7, Sep. 2003, pp. 1026-1040.
[2] D. Hunter, *et al.*, "WASPNET: A Wavelength Switched Packet Network", *IEEE Communications Magazine,* vol. 37, no. 3, March 1999, pp. 120-129.
[3] W. Zhong and R. Tucker, "Wavelength routing-based photonic packet buffers and their applications in photonic packet switching systems", *IEEE Journal of Lightwave Technology*, vol. 16, no. 10, Oct. 1998, pp. 1737-1745.
[4] W. D. Zhong and R. Tucker, "A new wavelength-routed packet buffer combining traveling delay-lines with delay-line loops", *IEEE Journal of Lightwave Technology*, vol. 19, August 2001, pp. 1085-1092.
[5] I. White, *et al.*, "Wavelength Switching Components for Future Photonic Networks", *IEEE Communications Magazine*, vol. 40, no. 9, September 2002, pp. 74-81.
[6] H. Takahashi, S. Suzuki and K. Kato, I. Nishi, "Arrayed-waveguide grating for wavelength division multi/demultiplexer with nanometre resolution", *Electronic Letters*, vol. 26, 1990, pp. 87-88.
[7] P. Pavon-Mariño, F.J. Gonzalez-Castaño and J. Garcia-Haro, "Round-robin wavelength assignment: A new packet sequence criterion in Optical Packet Switching SCWP networks", *European Transactions on Telecommunications* (Wiley Publishers), vol. 17, no. 4, Jul/Aug 2006, pp. 451-459.
[8] S. L. Danielsen, C. Joergensen, B. Mikkelsen and K. Stubkjaer, "Analysis of a WDM packet switch with improved performance under bursty traffic conditions due to tunable wavelength converters", *IEEE Journal of Lightwave Technology*, vol. 16, no. 5, May 1998, pp. 729-735.
[9] C. Guillemot, *et al.*, "Transparent optical packet switching: the European ACTS KEOPS project approach", *IEEE Journal of Lightwave Technology*, vol. 16, no. 12, Dec. 1998, pp. 2117-2134.
[10] P. Pavon-Marino, J. Garcia-Haro, J. Malgosa-Sanahuja and F. Cerdan, "Scattered Versus Shared Wavelength Path Operation, Application to Output Buffered Optical Packet Switches. A comparative study", *SPIE/Kluwer Optical Networks Magazine,* vol. 4, no. 6, November/December 2003, pp. 134-145.
[11] P. Pavon-Marino, J. Garcia-Haro, J. Malgosa-Sanahuja and F. Cerdan, "Maximal Matching Characterization of Optical Packet Input-Buffered Wavelength Routed Switches", Proc. of 2003 IEEE Workshop on High Performance Switching and Routing (HPSR 2003), Torino (Italy), June 2003, pp. 55-60.
[12] S. Asratian *et al.*, *Bipartite graphs and their applications*". Cambridge Tracts in Mathematics, 1998.
[13] A. Mekkittikul and N. McKeown, "A Practical Scheduling Algorithm to Achieve 100% Throughput in Input-Queued Switches", *IEEE Infocom 98, vol. 2, pp. 792-799, April 1998, San Francisco.*

[14] N. Alon, “A simple algorithm for edge-coloring bipartite multigraphs”, *Information Processing Letters*, vol. 85, no. 6, March 2003, pp. 301-302.
[15] Y. Tamir and G. Frazier, “High performance multi-queue buffers for VLSI communication switches”, Proc. of 15th Ann. Symp. on Comp. Arch., pp. 343-354, June 1988.
[16] Y. Tamir and H. C. Chi, “Symmetric Crossbar Arbiters for VLSI Communication Switches”, *IEEE Transactions on Parallel and Distributed Systems*, vol. 4, no. 1, January 1993, pp. 13-27.
[17] R. O. LaMaire and D. N. Serpanos, “Two-dimmensional round-robin schedulers for packet switches with multiple input queues”, *IEEE/ACM Transactions on Networking,* vol. 2, no. 5, October 1993, pp. 471-482.
[18] T. Anderson, S. Owicki, J. Saxe and C. Thacker, “High speed switch scheduling for local area networks”, *ACM Transactions on Computer Systems,* pp. 319-352, November 1993.
[19] C. Lund, *et al.*, “Fair prioritized scheduling in an input-buffered switch”, Proceedings of the IFIP-IEEE Conference on Broadband Communications ‘96, Montreal, April 1996, pp. 358-369.
[20] N. McKeown, “iSLIP: A Scheduling Algorithm for Input-Queued Switches”, *IEEE/ACM Transactions on Networking,* vol. 7, no. 2, April 1999, pp. 188-201.
[21] Y. Jiang and M. Hamdi, “A fully Desynchronized Round-Robin Matching Scheduler for a VOQ Packet Switch Architecture”, IEEE Workshop on High Performance Switching and Routing, Dallas, 2001, pp. 407-411.
[22] C. Chang, D. Lee and Y. Jou, “Load balanced Birkhoff-von Neumann Switches, Part I: One stage buffering”, *Computer Communications,* vol. 25, no. 6, April 2002, pp. 611-622.
[23] R. Asorey, *et al.,*”On the behavior of PHM Distributed Schedulers for Input Buffered Packet Switches”, *IEEE Transactions on Communications,* vol. 51, no. 7, July 2003, pp. 1057-1060.
[24] A. M. Law and J. S. Carson, “A sequential procedure for determining the length of a steady-state simulation”, Operations Research, vol. 27, 1979, pp. 1011-1025.
[25] S. C. Liew, “Performance of Various Input-buffered and Output-buffered ATM Switch Design Principles under Bursty Traffic: Simulation Study”, *IEEE Transactions on Communications*, vol. 42, no. 2/3/4, April 1994, pp. 1371-1379.