

# Knockout Packet Loss Probability Analysis of SCWP Optical Packet Switching Wavelength Distributed Knockout Architecture

<sup>1</sup>P. Pavón-Mariño, <sup>1</sup>C. López-Bravo, <sup>1</sup>J. García-Haro, <sup>2</sup>F. J. González-Castaño

<sup>1</sup>*Department of Information Technologies and Communications. Polytechnic University of Cartagena. Campus Muralla del Mar s/n, 30202, Cartagena, Spain. Phone: + 34 968325952  
Fax: +34 968325973*

<sup>2</sup>*Department of Telematics. University of Vigo. Campus Lagoas-Marcosende, 36310, Vigo, Spain. Phone: +34 986813788. Fax: +34 986812116.*

*E-mail: {pablo.pavon, cristina.lopez, joang.haro}@upct.es, javier@det.uvigo.es*

## Abstract

*The deployment of Optical Packet Switching (OPS) in Dense Wavelength Division Multiplexing (DWDM) backbone networks is perceived as a medium term promising alternative. Scalability restrictions imply that conventional switching architectures are unfeasible in this large-scale scenario. In a previous paper, the wavelength-distributed knockout architecture was proposed as a cost-effective scaling strategy for OPS switching fabrics. In this paper, this growable architecture is applied to OPS switching fabrics able to emulate output buffering. We also propose an scheduling algorithm which provides optimum performance if knockout packet losses are made negligible. The mathematical analysis to evaluate the knockout packet loss probability of this architecture is obtained, under uniform and non-uniform traffic patterns. To complement the switch dimensioning process, an upper bound assuring 0-knockout packet losses is compared with the exact analytical results.*

## 1. Introduction

The Optical Packet Switching (OPS) paradigm in Wavelength Division Multiplexing (WDM) networks is similar to traditional electronic packet switching, except that packet payload transparently remains in the optical domain, while headers are processed electronically. OPS offers high flexibility and bandwidth efficiency, since it operates on packet granularity. However, it is well-known that fast packet-by-packet switching operation and optical buffering impose the highest constraints to the photonic switching function, incurring in high hardware costs under the state-of-the-art

technology. For this reason, although OPS is envisaged as a conclusive solution for WDM networks, the deployment of an OPS backbone network is not foreseen in the near future.

Optical packet length in OPS networks is a current topic of discussion. In this paper, we focus on synchronous slotted OPS. This strategy defines 1) a fixed packet length, equal to the slot time, 2) the alignment of optical packets to time slot boundaries, at switch input ports. Packet alignment requires the design of optical synchronizing stages. Although synchronization stages increase equipment cost, the associated performance improvement resulting from a better contention behavior has favored the study of this alternative. The European DAVID project [1] quoted synchronous slotted OPS, with a time slot in the order of  $1 \mu\text{s}$ , as the most promising option for the WDM backbone network.

When applying OPS to WDM networks, the *networking operational mode* establishes the way permanent higher layer connections (Optical Packet Paths, OPPs) between ingress and egress edge nodes are mapped onto appropriate wavelengths in the links. This issue has been addressed by the WASPNET project [2], where two possible methodologies were proposed: Shared Wavelength Path (SHWP) and Scattered Wavelength Path (SCWP). In SHWP, packets from the same OPP follow a fixed sequence of hops to the egress node, such that transmission fiber and wavelength are fixed for each hop. In the SCWP operational mode, an optical path has a fixed sequence of transmission fibers, but transmission wavelength in each hop is undetermined. Incoming packets to a switch node demand destination output fibers, but the switch may dynamically decide output wavelengths for each packet.

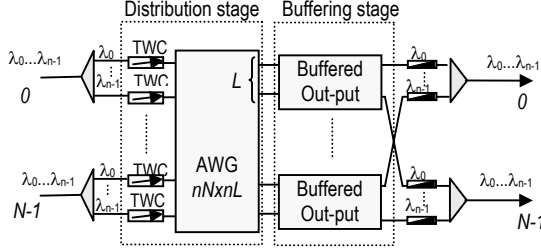


Figure 1. Wavelength-distributed OPS knockout architecture.

This is accomplished by the SCWP scheduling algorithm implemented in each node. The extra degree of freedom available for switch schedulers allows a joint decision on packet delay and packet output wavelength, boosting the statistical multiplexing effect. Therefore, SCWP allows higher throughput and lower packet delays in OPS switching architectures than SHWP operation, yielding simpler architectures according to optical buffering needs [3][4]. The authors consider SCWP as the logical operational mode in future OPS networks.

A previous paper of the authors reviewed several OPS switching architectures, able to emulate output buffering operation [3]. A SCWP scheduling algorithm for these switch fabrics was proposed with optimum throughput/delay performance. The results in [3] showed very low buffering requirements for dense WDM (DWDM) switching architectures, with a high number of wavelengths per fiber. This is because the statistical multiplexing effect caused by the freedom to select output wavelength increases with the number of output wavelengths. As an example, for 32 wavelengths per fiber, a buffer depth of only two positions yields a packet loss probability below  $10^{-9}$ , under Bernoulli uniform input traffic of average load 0.8. This only means 200 m of fiber delay lines, for a packet length of 1  $\mu$ s. However, DWDM OPS requires switching architectures with a large count of input and output ports. In this large-scale scenario, scalability constraints related to the photonic devices arise. The proposed OPS architectures are unfeasible even for 32x32 or 64x64 port switch sizes. Therefore, aggregation of optical switching elements into growable/scalable connection fabrics is mandatory. The interest in this research field is powered by the fact that Dense Wavelength Division Multiplexing is the conceived paradigm for optical backbone networks in the medium term.

In [5] the wavelength-distributed knockout (WDK) architecture was proposed as a growth strategy for OPS switch fabrics. This architecture, shown in figure 1, is based on the connection of a memoryless distribution stage to a buffered output stage, with one buffered module per output wavelength. The distribution stage

consists of a set of  $nN$  Tunable Wavelength Converters (TWC) and an Arrayed-Waveguide Grating (AWG) device. Wavelength conversion of input packets in the distribution stage is employed to select the output port of the AWG routing device, among the  $L$  inlets which lead to the assigned output wavelength buffered module (see figure 1).

Therefore, knockout packet losses [6] arise if more than  $L$  packets are destined to the same output module (i.e. are assigned the same output wavelength) in one time slot. Different OPS switch fabric architectures found in the open literature can be used as the buffered switching modules. Buffer overflow may cause packet losses in this stage.

Cost comparison results in [5] show that the wavelength-distributed knockout architecture is a very competitive large-scale switching strategy. In this paper we focus on the application of the WDK concept to OPS switch fabrics able to emulate output buffering. There are several output buffered OPS switch fabrics in the literature (see [3] for references). The results in this paper hold regardless of the OPS switch fabric alternative chosen for the output stage buffering module. In the WDK architecture, knockout losses depend on how the SCWP scheduling algorithm direct incoming packets to the output modules. In this paper, the *knockout uniform SCWP scheduler* is presented and applied to the WDK architecture with output buffered output modules. This scheduler is an adaptation of the *uniform SCWP scheduler* proposed in [7] for conventional OPS output buffered architectures. A relevant contribution of this paper is the mathematical analysis to obtain the *exact* knockout packet loss probability for the *knockout uniform SCWP scheduler*. The dimensioning process of this method is further employed to evaluate the WDK architecture under uniform and hot-spot traffic models.

The rest of the paper is organized as follows. Section 2 describes the scheduling algorithm applied. Section 3 explains the mathematical analysis developed to evaluate knockout losses. In section 4, the dimensioning process to obtain parameter  $L$  is applied in an evaluation of the WDK architecture. Section 5 concludes the paper.

## 2. Scheduler design

This section presents the *knockout uniform SCWP scheduler* for the knockout wavelength distributed architecture, with output buffered OPS switch fabrics as output stage modules. The objective of this algorithm is to select and assign for incoming packets a delay line in

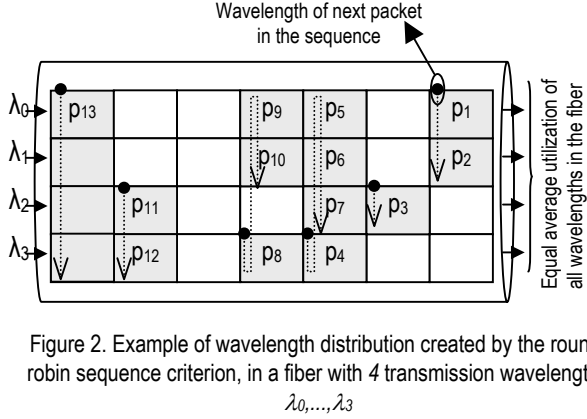


Figure 2. Example of wavelength distribution created by the round-robin sequence criterion, in a fiber with 4 transmission wavelengths  $\lambda_0, \dots, \lambda_3$

a buffered output module (figure 1) and an output wavelength. Output wavelength assignment determines the output stage module the packet is switched to. Delay assignment is the selection of delay lines in the output-buffered modules. Our goal is to optimize switch performance, maintaining the packet sequence for every OPP. In an OPS backbone network, it is necessary to preserve end-to-end packet sequence to avoid reordering cost at the egress node. Electronic re-sequencing stages would require very large memories due to the high speed of the optical links. Assuming this, the ingress nodes and the interconnection nodes must enforce packet sequence *in each hop* across the network. Thus, packet order information should be available for switching nodes.

In [7], the authors proposed the *round-robin packet sequence criterion* for SCWP OPS networks, which preserves end-to-end packet sequence. The method employs packet arrival time and packet arrival wavelength to deduce packet order. Interestingly, it does not require a specialized sequence field in packet headers. Therefore, it does not degrade performance due to header growth. An additional byproduct of the proposed sequencing criterion is that it guarantees a uniform wavelength utilization in all network fibers, for any traffic pattern. In [7] it was shown that this balanced wavelength usage was not achieved with previous packet-sequence methodologies, which incurred in performance impairments.

Let  $pck_i$  and  $pck_{i+1}$  be ordered and consecutive packets, transmitted in a WDM link in time slots  $t(pck_i)$  and  $t(pck_{i+1})$ , and wavelengths  $\lambda(pck_i)$  and  $\lambda(pck_{i+1})$ , respectively. The round-robin ordering criterion specifies that: 1)  $t(pck_{i+1}) \geq t(pck_i)$ , and 2)  $\lambda(pck_{i+1}) = (\lambda(pck_i) + 1) \bmod n$ , where  $n$  is the number of wavelengths in the fiber under consideration. Operation  $(a \bmod b)$  produces the remainder of  $a/b$  for any two integers  $a$  and  $b$ . From the switching node point of view, the criterion requires each node to “remember”

the wavelength of the last packet received/transmitted in the sequence across consecutive time slots.

As a consequence, implementations of this functionality require two sets of round-robin pointers to track packet sequence:

- 1) One round robin pointer per input fiber, tracking the wavelength of the next packet in the input traffic sequence. When a new packet appears in this wavelength, the pointer is incremented in a round-robin fashion.
- 2) One round robin pointer per output fiber, determining the output wavelength of the next packet to be transmitted. When a new packet is transmitted, the pointer is also round-robin incremented.

The method also requires pointer synchronization during equipment start-up: The pointer of each input fiber of a node should be synchronized with the pointer of the output fiber of the previous node. The result of this methodology, as shown in figure 2, is an exact round-robin packet spread across the wavelengths, for any traffic pattern in the fiber.

The pseudocode of the *knockout uniform SCWP scheduler* is shown below.

#### Knockout Uniform SCWP scheduler

```

/* RR_in [f_in] = R-R pointer for input fiber f_in */
/* RR_out [f_out] = R-R pointer for output fiber f_in */
/* delay [f_out] = active delay for packets destined to
output fiber f_out */
/* lastDelayOccup [f_out] = number of packets which
received the active delay of output fiber f_out */
/* ko [λ_out] = number of packets assigned this time
slot to output module λ_out */

1. for fiberCounter = 0 to N-1 do
2.   f_in = (f_0 + fiberCounter) mod N
3.   for wavCounter = 0 to n-1 do
4.     λ_in = RR_in [f_in] /* next wavelength to check */
5.     if (packet p in input (f_in, λ_in)) then
6.       /* if a packet is received, the input RR
pointer is incremented */
7.       RR_in [f_in] = (RR_in [f_in] + 1) mod n
8.       f_out = output fiber p ( = opp (p) )
9.       λ_out = RR_out [f_out]
10.      if (delay [f_out] < M and ko [λ_out] < L) then
11.        associate module λ_out to packet p
12.        associate delay [f_out] to packet p
13.        lastDelayOccup [f_out] ++
14.        if (lastDelayOccup [f_out] == n) then
15.          lastDelayOccup [f_out] = 0
16.          delay [f_out] ++
17.        endif
18.        ko [λ_out] ++
19.        RR_out [f_out] = (RR_out [f_out] + 1) mod n
20.      endif
21.    else
22.      /* for each input fiber, wavelength indexes
are selected until an "empty" wavelength
is found, or n wavelengths have been
selected */
23.      break;
24.    end
25.  endfor

```

```

26. endfor
27. /* Process performed at the end of every
   switching slot to guarantee input
   scanning fairness */
28. f0 = (f0 + 1) mod N
29. for fout = 0 to to N-1 do
30.   if (delay [fout] == 0)
31.     lastDelayOccup [fout] = 0;
32.   else
33.     delay [fout] --;
34.   endif
35. endfor
36. for λout = 0 to to n-1 do
37.   ko [λout] = 0 /* reset knockout counters every
   time slot */
38. endfor

```

This scheduler is an adaptation of the *uniform SCWP scheduler* proposed in [7] for conventional output buffered architectures. The new algorithm has the following properties:

- 1) If the knockout packet losses are made negligible, the overall switch operates as a pure output-buffered OPS switch. In these circumstances, the algorithm optimally minimizes packet losses and average packet delay. The markovian analysis for delay and packet loss evaluation presented in [3] can then be assumed.
- 2) The algorithm preserves packet sequence according to the *round-robin packet sequence criterion* proposed in [7].
- 3) The algorithm imposes a bound to the maximum number of packets that can be destined to a particular output module in the same time slot. If the parameter  $L$  is set to this bound,  $0$ -knockout packet loss can be assured for any input traffic pattern. The results in section 4 illustrate the high accuracy of this bound.

#### Algorithm Description

- *Input ports scanning*: To maintain packet order, each node requires a round-robin wavelength scanning pointer per input fiber (RR<sub>in</sub> [f<sub>in</sub>]) to track packet sequence. To fairly consider all traffic sources (i.e. all input fibers), the algorithm rotates the index of the first input fiber checked each time slot (f<sub>0</sub>).
- *Delay and output wavelength assignment*: Variable delay [f<sub>out</sub>] tracks what we call *active delay for output fiber f<sub>out</sub>*. The delay that is currently being assigned to packets destined to f<sub>out</sub>. Variable lastDelayOccup stores the number of packets being assigned the active delay. Optimum performance is achieved since *i*) any packet gets the shortest delay available (active delay, pseudocode line 12), *ii*) when n<sub>out</sub> packets are assigned, the algorithm uses the next delay (lines 14-17), *iii*) as a consequence, a packet destined to

output fiber f<sub>out</sub> is lost (condition in line 10 not true) when all  $M$  delays for that output fiber have n<sub>out</sub> packets. Consecutive packets transmitted through the same output fiber get output wavelengths from the round-robin pointer RR<sub>out</sub>[f<sub>out</sub>] (line 11). Thereby, the sequence criterion is fulfilled.

- *Inlet assignment*: The set of variables ko [λ<sub>out</sub>] store the number of packets assigned in the current time slot to each output module. If a module has no free inlets, the packet is discarded. No other output modules with free inlets are checked. This is because output wavelength of each packet should be assigned attending to the round-robin criterion. If the packet was assigned to other output wavelength with free inlets (1) the packet sequence would not be preserved, (2) the output wavelength-delay assignment coordination, required for output-buffered equivalent performance, would be lost.

### 3. Knockout packet loss probability analysis

This section describes the analysis proposed to calculate the knockout packet losses of the aforementioned OPS knockout architecture, when *the knockout uniform SCWP scheduler is in use*. Input traffic is assumed to be composed of  $nN$  independent sources, one per input port, of average load  $\rho$ . Under this assumption, input traffic correlation across successive time slots is not an issue to compute average knockout losses [6].

Let us define random variable  $A$ , which ranges from 0 to  $A_{MAX}$ , as the number of packet arrivals destined to a tagged output wavelength in a given time slot. Applying the Generalized Knockout Principle [6], the packet loss probability caused by the knockout effect ( $P_{KO-loss}$ ) is expressed by (1).

$$P_{KO-loss} = \lim_{T \rightarrow \infty} \frac{\# \text{ packets lost}}{\# \text{ packet arrivals}} = \frac{E(\text{packets lost in one slot})}{E(\text{packets arriving in one slot})} = \frac{\sum_{k=L+1}^{A_{MAX}} P(A=k) \cdot (k-L)}{\sum_{k=1}^{A_{MAX}} P(A=k) \cdot k} \quad (1)$$

Thus, in the sequel, our objective is to obtain the discrete probability density function of arrivals,  $A^{(k)} = [P(A=k), k=0..A_{MAX}]$ . First, the following random variables should be defined:

-  $\bar{a} = (a_0, \dots, a_{N-1})$  is the vector of arrivals in a given time slot, where  $a_i, i=0, \dots, N-1$ , is defined as the number

of packet arrivals destined to output fiber  $i$ . We denote  $E$  as the number of input ports with no incoming packet. Note that  $a_0 + \dots + a_{N-1} + E = nN$ .

-  $\vec{p} = (p_0, \dots, p_{N-1})$  is the vector of round-robin distances in the beginning of a time slot, before applying the scheduling algorithm. Each random variable  $p_i$  can take values from 0 to  $n-1$ . The value of  $p_i$  indicates the distance from the round-robin pointer associated to output of fiber  $i$  (RR\_out [i] in the pseudocode) to the tagged output module. A value of  $p_i=0$  means that the pointer points to our tagged module. This implies, as the knockout uniform SCWP scheduler states, that the first packet destined to output fiber  $i$  will be switched to this module. A value of  $p_i = p$  indicates that the first  $p$  packets destined to output fiber  $i$  will be addressed to other output modules.

-  $A_i, i=0, \dots, N-1$ , denotes the number of packets destined to output fiber  $i$ , which were addressed by the algorithm to the tagged output module. In a given time slot  $A=A_0+\dots+A_{N-1}$ . The knockout SCWP scheduling algorithm determines that the random variable  $A_i$  is specified by (2), using the *ceiling function*:

$$A_i = \left\lceil \frac{a_i - p_i}{n} \right\rceil, i = 0, \dots, N-1 \quad (2)$$

That is, the first  $p_i$  of all packets destined to output fiber  $i$  will be sent to other output modules. From the remaining  $(a_i - p_i)$  packets -if any-, one out of every  $n$  will be sent to the tagged output module.

Initially, the probability density function of  $A$  is given by (3):

$$A^{(K)} = P[A = k] = \sum_D P[\vec{a} = \vec{a}_d, \vec{p} = \vec{p}_d] \quad (3)$$

$$D = \left\{ \vec{a}_d, \vec{p}_d \mid A = \sum_{i=1}^N \left\lceil \frac{a_i - p_i}{n} \right\rceil = k \right\}$$

Consequently, the summatory in expression (3) requires checking all the combinations of  $\vec{a}$  and  $\vec{p}$ , summing the probabilities of those vector pairs  $(\vec{a}, \vec{p})$

fulfilling that  $A = \sum_{i=1}^N \left\lceil \frac{a_i - p_i}{n} \right\rceil = k$ . Set  $D$  symbolizes these combinations.

The complexity, in terms of the number of elements of set  $D$ , grows as a hard exponential function of the switch size. Thus, a brute force solution of equation (3) is unfeasible even for moderate switch sizes. In this paper, we propose a numerical method which takes benefit of the statistical relationship between the

random variables involved. This allows us to calculate the *exact* solution of equation (3), keeping complexity under reasonable levels. Our analysis has been validated by simulation, although we do not include the results in this paper since the method is exact, not an approximation.

Our analysis is based on the following properties of the random variables involved:

I) Random variables  $p_i$  depend on the arrivals of packets destined to output fiber  $i$ , in previous time slots. As a consequence, random variables  $p_i$  are jointly independent, as well as independent from the arrival process  $\vec{a}$ . The probabilities  $P[p_i=k]$  are uniformly distributed, due to the round-robin operation of the pointers across all output wavelengths.

$$P_k = P[p_i = k] = \frac{1}{n}, k = 0..n-1 \quad (4)$$

II) Random variables  $a_i, i=0..N-1$  in vector  $\vec{a}$  cannot be considered jointly independent. For instance, the number of arrivals to different output fibers and the number of empty ports are related by the equation  $a_0 + \dots + a_{N-1} + E = nN$ . When independent sources are considered, the joint probability of coordinates  $a_i$  is given by the multinomial distribution. If the output fiber demanded by each incoming packet is uniformly distributed, the associated multinomial expression is denoted by (5a).

$$P[\vec{a} = (a_0, \dots, a_{N-1})] = \quad (5a)$$

$$= \frac{(nN)!}{a_0! \dots a_{N-1}! E!} \left( \frac{\rho}{N} \right)^{a_0} \dots \left( \frac{\rho}{N} \right)^{a_{N-1}} (1-\rho)^E =$$

$$= \frac{(nN)!}{a_0! \dots a_{N-1}! E!} \left( \frac{\rho}{N} \right)^{(nN-E)} (1-\rho)^E$$

Equation (5b) shows the multinomial expression for non-uniform *hot-spot* traffic. In this example, an incoming packet is destined to output fiber 0 with probability  $0 \leq S \leq 1$ , and destined to output fibers  $1..N-1$  with equal probability  $\frac{1-S}{N-1}$ .

$$P[\vec{a} = (a_0, \dots, a_{N-1})] = \quad (5b)$$

$$= \frac{(nN)!}{a_0! \dots a_{N-1}! E!} (\rho S)^{a_0} \left( \frac{\rho(1-S)}{N-1} \right)^{a_1} \dots \left( \frac{\rho(1-S)}{N-1} \right)^{a_{N-1}} (1-\rho)^E =$$

$$= \frac{(nN)!}{a_0! \dots a_{N-1}! E!} (\rho S)^{a_0} \left( \frac{\rho(1-S)}{N-1} \right)^{(nN-a_0-E)} (1-\rho)^E$$

Our purpose is to obtain a simplified version of summation (3), by using the statistical relations (I) and (II). Let's define random variable  $A' = A_0' + \dots + A_{N-1}'$  as

the number of arrivals to the tagged output module, *conditioned* to a given packet arrivals vector. These conditional random variables are still a function of random vector  $\vec{p}_d$ . However, distribution probabilities of pointers positions are known (4). Therefore, it is possible to calculate the exact distribution of conditional random variable  $A'$ , from the known distributions in (4). This is performed in two steps:

- a) The distribution of variables  $A'_i$ ,  $i=0, \dots, N-1$  is calculated.  $A'_i$  is the number of packets destined to output fiber  $i$ , which are assigned the tagged output module, conditional to a given arrivals vector  $\vec{a}$ . Random variable  $A'_i$  depends on the number of packet arrivals destined to fiber  $i$  ( $a_i$ ) and the initial position of pointer  $i$  ( $p_i$ ). The uniform distribution of pointer positions (4) yields equation (6), where  $frac(x)$  stands for the fractional part of real number  $x$ .

$$A'_i = (A_i | \vec{a} = \vec{a}_d) = \begin{cases} \left\lfloor \frac{a_i - 0}{n} \right\rfloor \text{ with prob } \frac{1}{n} \\ \dots \\ \left\lfloor \frac{a_i - (n-1)}{n} \right\rfloor \text{ with prob } \frac{1}{n} \\ \left\lfloor \frac{a_i}{n} \right\rfloor \text{ with prob } 1 - frac\left(\frac{a_i}{n}\right) \\ 1 + \left\lfloor \frac{a_i}{n} \right\rfloor \text{ with prob } frac\left(\frac{a_i}{n}\right) \end{cases} \quad (6)$$

- b) The convolution theorem is applied to the calculus of the probability function  $A'$ , since  $A'_i^{(k)}$  conditional distributions are independent.

$$A' = A'_1 + \dots + A'_N \Rightarrow A'^{(k)} = A'_1^{(k)} * \dots * A'_N^{(k)} \quad (7)$$

In this stage of the analysis, we can obtain the exact distribution of conditional random variables  $A'$ . Applying the total probabilities theorem, the objective distribution  $A^{(k)}$  can be expressed as a function of these conditional probabilities. This produces expression (8). Note that, so far, a summatory with one addend per each possible combination of vector  $\vec{a}$  is required.

$$A^{(k)} = \sum_{\forall \vec{a}_d} P[A = k | \vec{a} = \vec{a}_d] P[\vec{a} = \vec{a}_d] = \sum_{\forall \vec{a}_d} P[A' = k] P[\vec{a} = \vec{a}_d] \quad (8)$$

The number of addends in the formula can be further decreased. Considering expression (8) for different vectors  $\vec{a}_d$ , it holds that two  $\vec{a}_d$  vectors with permuted values in their coordinates  $a_0$  to  $a_{N-1}$  yield the same values of  $P[A' = k]$  and the same values of  $P[\vec{a} = \vec{a}_d]$ . This is true for uniformly distributed traffic, when multinomial expression (5a) is applied. In this case, an additional simplification can be achieved by rewriting equation (8) as follows:

$$A^{(k)} = P[A = k] = \sum_{\forall \vec{a}_d} P[A' = k] P[\vec{a} = \vec{a}_d] = \sum_{A_D} perm(a_0, \dots, a_{N-1}) P[A' = k] P[\vec{a} = \vec{a}_d] \quad (9)$$

$$A_D = \left\{ \begin{array}{l} \text{each } \vec{a}_d \in A_D \text{ can not} \\ \text{be expressed as a} \\ \text{permutation of the} \\ \text{coordinates } a_0, \dots, a_{N-1} \text{ of} \\ \text{other } \vec{a}_d \in A_D \end{array} \right\}$$

Where:

- $P[\vec{a} = \vec{a}_d]$  is given by equation (5a).
- $P[A' = k]$  is calculated from equation (8).
- Function *perm* provides the number of different permutations of values  $a_0 \dots a_{N-1}$ , expressed by (10). Value  $v$  denotes the number of components with different values of  $a_i$ , and  $r_i$  ( $i=1 \dots v$ ) is the number of times a coordinate value is repeated.

$$perm(a_0, \dots, a_{N-1}) = \frac{N!}{r_1! \dots r_v!} \quad (10)$$

For hot-spot traffic, a similar simplification can be obtained. In multinomial formula (5b), vectors  $\vec{a}_d$  with the same value in coordinate  $a_0$ , and permuted values in their coordinates  $a_1$  to  $a_{N-1}$  yield the same probability values.

### 3.1 Implementation of the method

Expression (9) summarizes the proposed method. For uniform traffic, one addend in expression (9) covers  $n^N \cdot perm(a_0, \dots, a_{N-1})$  iterations of formula (3). For hot-spot traffic each addend includes  $n^N \cdot perm(a_1, \dots, a_{N-1})$  iterations. This is a measure of the complexity reduction achieved. The method requires an enumeration process to obtain the set of vectors  $A_D$  of expression (9). This is computationally simple. For instance, it can be implemented by updating an ordered coordinates vector.

Table I. Number of inlets per output module (parameter  $L$ ) required for assuring a knockout packet loss probability less than  $10^{-9}$ . Values displayed for uniform traffic / hot-spot traffic (parameter  $S=0.8$ ). Input load  $\rho=\{0.1, \dots, 0.9\}$ , number of wavelengths per fiber  $n=\{2, 4, 8, 16, 32, 64, 128\}$ . (a)  $N=2$ , (b)  $N=4$ .

$N=2$		$\rho=0.1$	$\rho=0.2$	$\rho=0.3$	$\rho=0.4$	$\rho=0.5$	$\rho=0.6$	$\rho=0.7$	$\rho=0.8$	$\rho=0.9$	$A_{MAX}$
(a)	$L(n=2)$	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3
	$L(n=4)$	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3
	$L(n=8)$	2/2	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3
	$L(n=16)$	2/2	2/2	3/3	3/3	3/3	3/3	3/3	3/3	3/3	3
	$L(n=32)$	2/2	2/2	2/2	2/3	2/3	3/3	3/3	3/3	3/3	3
	$L(n=64)$	2/2	2/2	2/2	2/2	2/3	2/3	3/3	3/3	3/3	3
	$L(n=128)$	2/2	2/2	2/2	2/2	2/3	2/3	2/3	3/3	3/3	3
$N=4$		$\rho=0.1$	$\rho=0.2$	$\rho=0.3$	$\rho=0.4$	$\rho=0.5$	$\rho=0.6$	$\rho=0.7$	$\rho=0.8$	$\rho=0.9$	$A_{MAX}$
(b)	$L(n=2)$	4/4	5/5	6/5	6/6	6/6	6/6	6/6	6/6	6/6	6
	$L(n=4)$	4/4	5/5	6/5	6/6	6/6	6/6	7/7	7/7	7/7	7
	$L(n=8)$	4/4	4/5	5/5	5/6	6/6	6/6	6/7	7/7	7/7	7
	$L(n=16)$	4/4	4/4	4/5	5/6	5/6	5/6	6/6	6/7	7/7	7
	$L(n=32)$	4/4	4/4	4/5	4/5	5/6	5/6	5/6	6/7	6/7	7
	$L(n=64)$	4/4	4/4	4/5	4/5	4/5	5/6	5/6	5/6	6/7	7
	$L(n=128)$	4/4	4/4	4/5	4/5	4/5	4/6	4/6	5/6	6/7	7

For each vector produced, the convolutions needed to compute  $A^{(k)}$  can be accelerated by storing previous convolution results along the enumeration process. As an example, the method we implemented produces the results presented in table I in a processing time ranging from seconds (for up to 64 input and output ports architectures) to hours (for the largest architectures in the table) in an AlphaServer HPC160 with 32 Gigaflops of theoretical computing capacity. An implementation of the method is also accessible at: <http://labit301.upct.es/ops/evaluation/wdk>. This web interface offers the possibility to dynamically obtain output wavelength distribution and packet knockout loss probabilities for low-to-medium scale switches.

#### 4. Evaluation results

The analysis described in the previous section can be employed to dimension parameter  $L$  in output buffered WDK switches. In this section, representative results have been calculated, which allow us to extract some interesting conclusions. The traffic per input fiber is supposed to be the aggregation of  $n$  independent Bernoulli sources of parameter  $\rho$ , one per input wavelength. Destination fiber selections of each input packet are also independent. Packet arrivals are supposed to be distributed across input ports according to the round-robin sequence criterion. Under these assumptions, the memoryless switch stage makes this particular distribution of packets irrelevant. The input traffic pattern is then equivalent to a set of

$nN$  independent sources, and the numerical method described in section 3 can be applied.

Table I shows the number of inlets (parameter  $L$ ) required to obtain a knockout packet loss probability below  $10^{-9}$ , when the destination fiber of input packets is uniformly distributed, compared to the evaluation under hot-spot sources with parameter  $S=0.8$ . That means that 80% of arriving packets are bound to output fiber 0. In both cases, input load  $\rho$  ranges from 0.1 to 0.9, the number of wavelengths  $n$  takes values in  $\{2, 4, 8, 16, 32, 64, 128\}$ , and the number of fibers are  $N=2$  and  $N=4$ . This low number of input/output fibers is typical in the backbone network, where the number of neighbours each node has is usually 5 or less. Column  $A_{MAX}$  shows the maximum number of packets which can be destined by the scheduling algorithm to any output module. The maximum number of arrivals  $A_{MAX}$  is given by expression (11).

$$A_{MAX} = \min\left(nN, N + \left\lceil \frac{nN - N - n + 1}{n} \right\rceil\right) \quad (11)$$

This maximum is achieved when:

- All pointers initially point to the tagged output module ( $p_i = 0 \quad \forall i$ ).
- There are  $nN$  packet arrivals,  $N$  of which are destined to  $N$  different output fibers, and thus routed to the tagged output module.
- All remaining  $(nN - N)$  packets are destined to the same output fiber, and, as a consequence,

$$\left\lceil \frac{nN - N - n + 1}{n} \right\rceil \leq N \text{ packets arrive to the tagged}$$

output module.

Therefore, if parameter  $L$  is dimensioned as  $L=A_{MAX}$ , a zero knockout packet loss probability is assured for any input traffic pattern. The results show that the upper-bound dimensioning method with 0-knockout packet loss yields practically the same switch sizes as those obtained after the complete analysis for medium and high input traffic loads. Its accuracy makes this upper bound specially interesting for input traffic patterns that do not admit our analysis. For instance, bursty sources in SCWP networks: The round-robin packet distribution across wavelengths transforms any correlation between consecutive time slots (present in bursty sources) into a correlation among input ports in the same input fiber.

Table I shows that a skewed distribution in output fiber selection, has practically no effect in the switch dimensioning process. This is because an unbalanced output fiber selection is corrected by the round-robin spread of packets across output wavelengths (modules).

## 5. Conclusions

This paper studies the usefulness of the wavelength-distributed knockout architecture as a scaling strategy for OPS output-buffered switch fabrics. The knockout uniform SCWP scheduler is presented, and applied to this architecture. The scheduler provides optimum throughput/delay performance if the knockout losses are made negligible. Packet sequence is also kept, according to the round-robin sequence criterion proposed in [7]. In this framework, a numerical method is derived which allows the exact evaluation of knockout packet losses, assuming independent traffic sources, for uniform and non-uniform output fiber selection. If the input traffic pattern does not satisfy the independent condition, an upper bound can be used for switch dimensioning. The results show the high accuracy of this upper bound.

## Acknowledgements

This research has been funded by Spanish MCyT grants TEC2004-05622-C04-01/TCM (CAPITAL) and TEC2004-05622-C04-02/TCM (ARPaq) and *Xunta de Galicia* grant PGIDIT04TIC322003PR.

## References

- [1] Dittman L., Develder C., Chiaroni F., Neri F., Callegati F., Koerber W., Stavdas A., Renaud M., Rafel A., Sole-Pareta J., Cerroni W., Leligou N., Dembeck L., Mortensen B., Pickavet M., Le Sauze N., Mahony M., Berde B., Eilenberger G., "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] Hunter D., Nizam M., Chia M., Andonovic I., Guild K., Tzanakaki A., O'Mahony J., Bainbridge J., Stephens M., Penty R., White I., "WASPNET: A Wavelength Switched Packet Network", *IEEE Communications Magazine*, vol. 37, no. 3, pp. 120-129, March 1999.
- [3] Pavon-Mariño P., Garcia-Haro J., Malgosa-Sanahuja J., Cerdan F., "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.
- [4] Pavon-Mariño P., Garcia-Haro J., Malgosa-Sanahuja J., Cerdan F., "Maximal Matching Characterization of Optical Packet Input-Buffered Wavelength Routed Switches", *Proc. 2003 IEEE Workshop on High Performance Switching and Routing (HPSR 2003)*, Torino (Italy), June 2003, pp. 55-60.
- [5] P. Pavon-Mariño J., Garcia-Haro J., Malgosa-Sanahuja J., Cerdan F., "A Performance Study of a Knockout Large-Scale Optical Packet Switching Architecture Under SCWP Operational Mode", in *Proc. 2003 IEEE PACRIM 2003*, vol. 1, Victoria (Canada), August 2003, pp. 478-481.
- [6] Yeh Y.S., Hluchyj M.G., Acampora A.S., "The knockout switch: A simple, modular architecture for High Performance Packet Switching", *IEEE Journal on Selected Areas in Comm.*, vol. SAC-5, no. 8, pp. 223-231, October 1987.
- [7] Pavon-Mariño P., Gonzalez-Castaño F.J., Garcia-Haro J., "Round-Robin wavelength assignment: A new packet sequence criterion in Optical Packet Switching SCWP networks", submitted to *European Transactions on Telecommunications*.