Available online at www.sciencedirect.com

SCIENCE DIRECT®

Optical Switching and Networking 2 (2005) 137–147

Optical Switching and Networking

www.elsevier.com/locate/osn

Keeping the packet sequence in optical packet-switched networks

F. Callegati[a,1], D. Careglio[b,2], W. Cerroni[a,*], G. Muretto[a,1], C. Raffaelli[a,1],
J. Solé Pareta[b,2], P. Zaffoni[a,1]

[a] *Dipartimento di Elettronica, Informatica e Sistemistica, University of Bologna, Viale Risorgimento, 2, 40136 Bologna, Italy*
[b] *Advanced Broadband Communications Center (CCABA), Universitat Politècnica de Catalunya, Jordi Girona, 1-3, Modul D6, 08034 Barcelona, Spain*

Received 14 January 2005; accepted 23 September 2005
Available online 7 November 2005

Abstract

This paper deals with optical packet switches with limited buffer capabilities, subject to asynchronous, variable-length packets and connection-oriented operation. The focus is put on buffer scheduling policies and queuing performance evaluation. In particular a combined use of the wavelength and time domain is exploited in order to obtain contention resolution algorithms that guarantee the sequence preservation of packets belonging to the same connection. Four simple algorithms for strict and loose packet sequence preservation are proposed. Their performance is studied and compared with previously proposed algorithms. Simulation results suggest that by accepting some additional processing effort it is possible to guarantee very low packet loss probabilities while avoiding the out-of-sequence delivery.

Keywords: Optical packet switching; Asynchronous and variable-length packets; Connection-oriented operation; DWDM; Packet sequence

1. Introduction

Within the research framework on the next generation Internet, a widely discussed issue is the integration of IP with optical networks [1]. Mainly, the proposals focus on the use of the wavelength as a circuit and therefore operate with a very coarse bandwidth granularity [2]. In a longer term perspective optical packet switching (OPS) should guarantee a much finer and flexible access to the optical bandwidth [3]. At the same time, the introduction of the DWDM technology allows us to exploit the huge capacity of optical fibers and provides effective alternatives in switch design optimization. OPS has usually been studied with reference to fixed length synchronous packets, resulting in easier switching matrix design [4]. In this case, the drawbacks are both the increase in hardware complexity necessary to manage optical synchronization and the limited inter-working skill with network protocols which mainly employ variable-length packets, such as IP. For these reasons, solutions adopting asynchronous and variable-length packets have been investigated recently [5], showing that in this case the issue of congestion resolution becomes fundamental to achieving an acceptable level of performance.

[*] Corresponding author. Tel.: +39 051 2093089; fax: +39 051 2093053.

E-mail addresses: fcallegati@deis.unibo.it (F. Callegati), careglio@ac.upc.es (D. Careglio), wcerroni@deis.unibo.it (W. Cerroni), gmuretto@deis.unibo.it (G. Muretto), craffaelli@deis.unibo.it (C. Raffaelli), pareta@ac.upc.es (J. Solé Pareta), pzaffoni@deis.unibo.it (P. Zaffoni).

[1] Tel.: +39 051 2093089; fax: +39 051 2093053.

[2] Tel.: +34 93 4016985; fax: +34 93 4017055.

1573-4277/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.osn.2005.09.001

Contention resolution may be achieved in the time domain by means of optical queuing and in the wavelength domain by means of suitable wavelength multiplexing. Optical queuing is realized by Fiber Delay Lines (FDLs) that are used to delay packets contending for the same output fiber in case all wavelengths are busy. In [6] it has been shown that, by using suitable contention resolution algorithms able to combine the use of the time and the wavelength domain, it is possible to improve the performance up to an acceptable level, with a limited number of FDLs. We refer to them as *Wavelength and Delay Selection* (WDS) algorithms. Moreover these concepts may be effectively extended to a connection-oriented network scenario, for instance based on MPLS. In this case, a suitable design of dynamic allocation WDS algorithms permits us to obtain fairly good performance, by exploiting queuing behaviors related to the connection-oriented nature of the traffic, but with significant savings in terms of processing effort for the switch control with respect to the connectionless case [7].

The main drawback of previously proposed WDS algorithms is that *out-of-sequence delivery* of packets belonging to the same traffic flow cannot be avoided. The occurrence of out of order delivery raises performance problems for the end-to-end transport protocols and/or issues of implementation complexity if re-ordering at the edges of the optical network should be implemented. This will be further discussed in the following and is the basic motivation of this paper, where we address the issue and propose new WDS algorithms that guarantee the packet sequence preservation. The results presented in the paper show that, at the expense of some additional processing effort, the new algorithms improve the overall performance with respect to algorithms proposed in [7].

The paper is organized as follows. In Section 2 the networking scenario is introduced and the general task of integration between MPLS and OPS is addressed. In Section 3 the WDS problem is discussed in detail. In Section 4 the out-of-order problem is defined and discussed with special reference to its effects on the end-to-end transport control protocols. Section 5 is devoted to the description of the new proposed algorithms. In Section 6 numerical results are presented and conclusions are drawn in Section 7.

2. Networking scenario: MPLS/OPS

The Multi-Protocol Label Switching (MPLS) architecture [8] is based on a partition of the network layer functions into *control* and *forwarding*. The control component uses standard routing protocols to build up and maintain the forwarding table, while the forwarding component examines the headers of incoming packets and takes the forwarding decisions. Packets coming from client layers are classified into a finite number of subsets, called *Forwarding Equivalent Classes* (FECs), based on identification address and quality of service requirements. Each FEC is identified by an additional *label* added to the packets. Unidirectional connections throughout the network, called *Label Switched Paths* (LSPs), are set up and packets belonging to the same FEC are forwarded along these LSPs according to their labels. On each core node, simple label matching and swapping operations are performed on a precomputed LSP forwarding table, thus simplifying and speeding up the forwarding function.

On the other hand, optical packet switching (OPS), in order to be feasible and effective, requires a further partitioning of the forwarding component into *forwarding algorithm* and *switching* [9]. The former corresponds to the label matching that determines the next hop destination and the latter is the physical action of transferring a datagram to the proper output interface. The main goal of this separation is to limit the bottleneck of electro-optical conversions: the header is converted from optical to electrical and the execution of the forwarding algorithm is performed in electronics, while the payload is optically switched without electrical conversion.

This paper considers a DWDM network integrating MPLS and OPS, which relies on optical routers that exploit the best of both electronics and optics. Standard routing protocols are used as the (non-critical) routing component, MPLS labels in the forwarding algorithm (where strict performance limits are present) and, finally, optical technologies are used in switching and transmission, providing very high data rate and throughput.

In order to avoid scalability problems, we assume that each LSP represents a top-level explicitly routed path, formed by an aggregation of lower-level connections including several traffic flows [10], and that the number of LSPs managed by a single optical core router is not so high as to affect the correct label processing.

We also assume the availability of an optical switching matrix able to switch variable-length packets [11]. This paper is not supposed to deal with implementation issues. Therefore, a generic non-blocking architecture for an OPS node is assumed, which provides full wavelength conversion.

The switch is assumed to use a feed-forward optical buffering configuration [12], realized by means of *B*

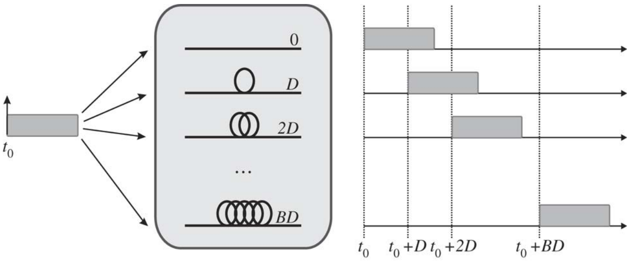

Fig. 1. Degenerate buffer structure.

fiber delay lines. Basically, an output queuing approach is assumed: each wavelength on each output fiber has its own buffer. This results in a logical queue per output wavelength. In principle, a set of delay lines per output wavelength could be deployed, leading to a fairly large amount of fiber coils. However, a single pool of FDLs may be used in WDM for all the wavelengths on a given fiber or for the whole switch. The delays provided are linearly increasing with a basic delay unit D, i.e. $D_j = k_j D$, with k_j an integer number and $j = 1, 2, \ldots, B$. For the sake of simplicity we assume a *degenerate buffer* [12] with $k_j = j - 1$ (see Fig. 1), but the ideas presented here are valid also for *non-degenerate buffers*, i.e. with different arrangements of k_j.

3. The wavelength and delay selection problem

The electronic Switch Control Logic (SCL) takes all the decisions regarding the configuration of the hardware to realize the proper switching actions. Once the forwarding component has decided to which output fiber the packet should be sent, determining also the network path, the SCL:

- chooses which wavelength of the output fiber will be used to transmit the packet, in order to properly control the output interface;
- decides whether the packet has to be delayed by using the FDLs or it has to be dropped since the required queuing resource is full.

These decisions are also routing independent and all the wavelengths of a given output fiber are equivalent for routing purposes but not from the contention resolution point of view. The choices of wavelength and delay are actually correlated: since each wavelength has its own logical output buffer, choosing a particular wavelength is equivalent to assigning the packet one of the available delays on the corresponding buffer. This is what we call the *Wavelength and Delay Selection* (WDS) problem. Here we assume that, once the wavelength has been chosen using a particular policy, always the smallest delay available after the last queued packet on the corresponding buffer is assigned. The smallest delay available on a given wavelength can be easily computed using the smallest integer greater than or equal to the difference between the time when the wavelength will be available again and the packet arrival time, divided by the buffer delay unit. This operation provides also the gap between the current and the previous queued packet.

The choice of the wavelength can be implemented by following different policies, producing different processing loads at the SCL and different resource utilizations:

- **Static**. The LSP is assigned to a wavelength at LSP setup and this assignment is kept constant over all the LSP lifetime. Therefore packets belonging to the same LSP are always carried by the same couple of input/output wavelengths. Contention on the output wavelength can only be solved in time domain by using delay lines. In this case the WDS algorithm is trivial and requires minimum control complexity. On the other hand resource utilization is not optimized since it is possible to have a wavelength of an output fiber congested even if the other wavelengths are idle.
- **Dynamic**. The LSP is assigned to a wavelength at LSP setup but the wavelength may change during the LSP lifetime. Two approaches are possible:
  - **per-packet**. The wavelength is selected on a per-packet basis, similarly to the connectionless case, choosing the most effective wavelength in the perspective of optimizing resource usage. It requires some processing effort on a per-packet basis, therefore this alternative is fairly demanding in terms of processing load on the SCL, which must be carefully dimensioned;
  - **per-LSP**. When heavy congestion arises on the assigned wavelength, i.e. when the time domain is not enough to solve contention due to the lack of buffering space, the LSP is temporary moved to another wavelength. When congestion disappears, the LSP is switched back

to the original wavelength. This alternative stays somewhat in between, aiming at realizing a trade-off between control complexity and performance

It is obvious that the per-packet alternative is the most flexible. In [5], the author observes that, since the FDL buffers are only able to provide discrete delays, this creates gaps between queued packets that can be considered equivalent to an increase of the packet service time, meaning an artificial increase in the traffic load (*excess load*). It has been demonstrated in [13] that a void filling algorithm that aims at minimizing those gaps gives best performance with respect to other policies. Nonetheless, the computational complexity of such an algorithm is high, since it requires the knowledge of the length and duration of every gap in the queues. Therefore, in order to keep the packet scheduling process as simple as possible, the gaps between queued packets are not considered as possible buffer places and a simplified approach is adopted here: the so-called MINGAP algorithm, proposed in [6]. It is a per-packet strategy selecting the wavelength such that the corresponding smallest delay available provides also the smallest gap after the packet previously scheduled.

On the other hand, in [7] it has been originally observed that by adopting a per-LSP strategy the switch performance in general depends on the configuration of the LSP forwarding table. The basic observation is that packets following LSPs incoming on the same input wavelength cannot overlap, because of the serial nature of the transmission line. Therefore such packets contend for output resources only with packets incoming on different wavelengths. As a consequence, if LSPs incoming on the same input wavelength are the only ones forwarded to the same output wavelength, contention will never arise (*optimal allocation*). This is just a trivial example but obviously a whole spectrum of combinations is possible. More generally this effect produces a negative correlation in the traffic profile, which is smoothed and therefore is less aggressive in terms of queuing requirements because it decreases the likelihood of congestion. Based on these considerations, in [7] some WDS algorithms are proposed that try to exploit this negative correlation. The best performing of these algorithms is called *Empty Queue Wavelength Selection* (EQWS): it is a per-LSP strategy trying to exploit the queuing space available on output wavelengths in the optimal allocation condition. In normal conditions, packets following a given LSP are transmitted on the wavelength assigned to that LSP. However, as soon as a packet finds a congested wavelength (i.e. a wavelength with no delays available), the algorithm looks for a wavelength in optimal allocation, which has an empty queue. If such a wavelength is found, the LSP is switched to it for as long as needed, that is until congestion arises also in the new wavelength or congestion ends on the previous one. In this case the LSP is switched back to the original wavelength. In case a wavelength in optimal allocation cannot be found, the LSP is re-assigned to the wavelength with the smallest delay available.

4. Problems due to out-of-order packet delivery

As already outlined in the introduction, it is well known that packet losses as well as out-of-order packet deliveries and delay variations affect end-to-end protocols behavior and may cause throughput impairments [14,15]. In particular, the problem of packet re-sequencing is not negligible in optical packet-switched networks, especially when optical packet flows carrying traffic related to emerging, bandwidth-demanding, sequence-sensitive services, such as grid applications and storage services, are considered.

When considering TCP-based traffic these phenomena influence the typical congestion control mechanisms adopted by the protocol and may result in a reduction of the transmission window size with consequent bandwidth under-utilization. In particular the TCP congestion control is very affected by the loss or the out-of-order delivery of bursts of segments. This is exactly what may happen in the OPS network where traffic is typically groomed and several IP datagrams (and therefore TCP segments) are multiplexed in an optical packet, because optical packets must satisfy a minimum length requirement to guarantee a reasonable switching efficiency. Therefore out-of-order or delayed delivery of just one optical packet may result in out-of-order or delayed delivery of several TCP segments triggering (multiple duplicate ACKS and/or timeouts that expire) congestion control mechanisms and causing unnecessary reduction of the window size.

Another example of how out-of-sequence packets may affect application performance is the case of delay-sensitive UDP-based traffic, such as real-time traffic. In fact unordered packets may arrive too late and/or the delay required to reorder several out-of-sequence packets may be too high with respect to the timing requirements of the application.

These brief and simple examples make evident the need to limit the number of unordered packets. In general out-of-order delivery is caused by the fact that packets belonging to the same flow of information can take different paths through the network and then can experience different delays [16]. In traditional connection-oriented networks, packet reordering is not an issue since packets belonging to the same connection are supposed to follow the same virtual network path and therefore are delivered in the correct sequence, unless packet loss occurs.

In an OPS network, using the wavelength domain for contention resolution (i.e. using dynamic policies), this may not be the case. Optical packets are transmitted on a given wavelength depending on the flow (i.e. LSP) they belong to. When a static WDS policy is adopted, a given flow is always transmitted on the same wavelength, so the order of packets cannot be changed, because a FIFO queuing discipline is assumed. On the other hand, when a dynamic WDS policy is used, packets from a given flow may be transmitted on different wavelengths, experience different delays and, eventually, be delivered not in the correct sequence.

To check the likelihood of out-of-sequence packet delivery when the WDS algorithms mentioned above are implemented, we set up the two hops network scenario shown in Fig. 2. Every switch is identically set up with 16 wavelengths per link, an optical buffer of $B = 4$ FDLs and a granularity equal to the average packet length. The inter-arrival packet generation follows a Poisson model while the packet size is exponentially distributed with an average value corresponding to a transmission time of the order of 1 μs, a typical value for optical packet switching technologies. We focus on packet size only in terms of duration because the simulators used here are built to be independent from the packet size in terms of bytes and from the bit-rate. Since one of the benefits of optical switching is the transparency to the bit-rate, what really matters is indeed the average packet duration and the inter-arrival time distribution. Obviously, once the optical packet duration is set, the higher the bit-rate, the higher the average packet size in bits, leading to the need for traffic grooming.

The input load on each wavelength is fixed to 0.8 and the traffic distribution is uniform. The traffic input at the edge switches is ideal in the sense that packets belonging to the same LSP arrive in order on the same wavelength. At each switch, packets are processed by the selected WDS algorithm, sent to the next hop and the resulting amount of out-of-sequence packets is also evaluated.

Table 1 shows the packet reordering distribution for the static, MINGAP, and EQWS algorithms. These results, even though related to a simple network architecture, are meaningful to show that MINGAP and EQWS algorithms are not able to avoid sequence breaking. The percentage of packets out-of-sequence of three or more locations is already not null at the input of the core switch. By assuming n switch in series along a path this percentage is expected to increase accordingly. Previous studies [14,15] confirm that just a small percentage of out-of-sequence (such as that caused by the EQWS algorithm) may impact harmfully on the network performance.

Table 1
Out-of-order percentages at the input and output ports of the core switch, comparing the static, MINGAP, and EQWS algorithms

Algorithm	Input	Output
Static	0	0
MINGAP	3.6498	6.9948
EQWS	1.6798	5.4200

A possible solution could be to assume that this problem is solved at the egress edge nodes that should take care of re-sequencing the various packet flows. This assumption in our view is not very realistic. It can be feasible for some flow of high value traffic, but it is unlikely that it will happen for all the flows of best effort traffic, because of the amount of memory and processing effort that would be necessary. Therefore we argue that it becomes fundamental to control out-of-order delivery of packets directly in the OPS network nodes.

5. WDS algorithms preserving the packet sequence

The way to overcome the aforementioned problem is to design WDS algorithms able to preserve the packet sequence from the beginning. When facing this problem in a WDM network, it is easy to realize that the term “out-of-sequence delivery” must be defined in more detail.

In this paper we use two possible definitions of ordered packets:

- **Strict sequence preservation**. Given a stream of ordered packets at the switch input, packet n is considered to be out-of-order when the first bit of packet n leaves the switch before the *last* bit of packet $n - 1$.
- **Loose sequence preservation**. Given a stream of ordered packets at the switch input, packet n is considered to be out-of-order when the first bit of packet n leaves the switch before the *first* bit of packet $n - 1$.

Indeed the strict sequence preservation is the ideal case. The loose sequence preservation is more difficult to control, especially when considering a cascade of

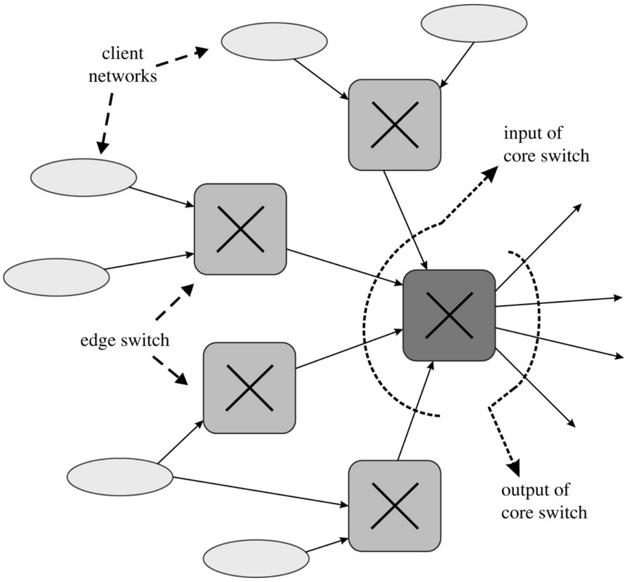

Fig. 2. Network scenario to evaluate the amount of out-of-order packets.

switches and, most of all, may still cause limited out-of-sequence arrivals at the edge nodes. This may be due to the fact that an optical packet is, in general, an aggregate of several IP datagrams and hence the relative position of subsequent IP packets included in two subsequent optical packets cannot be controlled if some overlapping is permitted. Nonetheless looser sequence preservation imposes weaker constraints on the WDS algorithm, which is expected to perform better. In the following subsections we describe the proposal of four simple algorithms for strict and loose packet sequence preservation.

To describe the algorithms, let us introduce the following notations:

- t_n: the arrival time of the generic packet n;
- d_n: the delay assigned to the generic packet n;
- l_n: the duration of the generic packet n;
- a_n: the time at which the first bit of the generic packet n is scheduled to leave the switch;
- b_n: the time at which the last bit of the generic packet n is scheduled to leave the switch. It is obvious that $b_n = a_n + l_n$ and $a_n = t_n + d_n$.

5.1. *Strict packet sequence*

The WDS algorithms designed to keep a strict packet sequence are called *Strict Packet Sequence with Minimum Length wavelength selection* (SPS-ML) and *Strict Packet Sequence with Minimum Gap wavelength selection* (SPS-MG). These algorithms have been originally proposed in [17]. Basically, they adopt the following per-LSP strategy: packets following a given LSP are transmitted on the wavelength assigned to that LSP as long as the corresponding buffer is not full. When congestion arises, the algorithm looks for a wavelength that can provide a delay such that the packet sequence is not broken. In case multiple choices are available, the options are to choose the wavelength with the smallest delay available (SPS-ML) or providing the smallest gap after the previous queued packet (SPS-MG).

In particular, when packet n arrives at time t_n, in order to preserve the packet sequence, the algorithm selects a queue depending on the time b_{n-1} at which the last bit of packet $n-1$, following the same LSP, is scheduled to leave the node. Assuming ordered packet streams arriving at the switch input, it is obvious that $t_n \geq t_{n-1} + l_{n-1}$, but if packet $n-1$ is queued, then $d_{n-1} > 0$ and there are two possible alternatives:

1. $t_n < b_{n-1}$: packet $n-1$ was queued and packet n overlaps with it. There is the chance to break the packet sequence.
2. $t_n \geq b_{n-1}$: packet n arrives when or after packet $n-1$ has completely left the node. The packet sequence is always guaranteed.

In the first case, packet n has to be delayed by an amount of time at least as long as the residual transmission time of packet $n-1$. Due to the discrete number of delays provided by the FDLs, which are multiples of the basic unit D, the minimum delay that may be assigned to packet *n* is:

$$D_{\min} = \left\lceil \frac{b_{n-1} - t_n}{D} \right\rceil D. \tag{1}$$

On the other hand, the minimum delay for the second case is $D_{\min} = 0$ because there is no possibility to break the correct packet sequence.

Once $D_{\min}$ is determined, the algorithm verifies if the wavelength assigned to the LSP is able to provide this delay. If not, it searches for another wavelength. This search may give multiple results, meaning that there are several wavelengths on which the packet may be transmitted with the required delay. When this happens the WDS algorithm must choose one among these queues. This choice is independent of the sequence issue that has already been solved by choosing the proper delay. Therefore the choice of the queue to send the packet can be made based on general optimization procedures similar to what has already been studied for the pure connectionless case [6]. Here we follow the same approach and two algorithms are then considered that adopt different choices:

- **SPS-MG** selects the queue among those not full which introduces the minimum gap between subsequent queued packets. If two or more queues provide the same gap, the shortest one is chosen, i.e. the queue providing the smallest delay greater than $D_{\min}$.
- **SPS-ML** selects the shortest queue. If two or more queues have the same minimum length, the one with the smallest gap is chosen.

It has to be underlined that to execute the SPS algorithms the SCL needs to store the last value of b_n for each LSP in the forwarding table. Nonetheless, with respect to previous algorithms, the extra cost only stays in the memory requirements, because whatever algorithm is used, the SCL must always calculate the value of b_n for each queued packet to know the buffer's occupancy.

Fig. 3 illustrates an example of the behavior of these algorithms. In Fig. 3(a), packet *n* arrives at the node and packet $n - 1$ on the same LSP is still in the queue. Therefore, packet *n* overlaps with it and the minimum assignable delay is $D_{\min} = 3$. Both algorithms select the second queue because it provides this delay. In Fig. 3(b), packet *n* arrives when packet $n - 1$ has completely left the node. In this case, the algorithms behave differently: with SPS-MG, packet *n* is put in the second queue (minimum gap) with delay $2D$; while with SPS-ML, the third queue (shortest one) and delay *D* is selected.

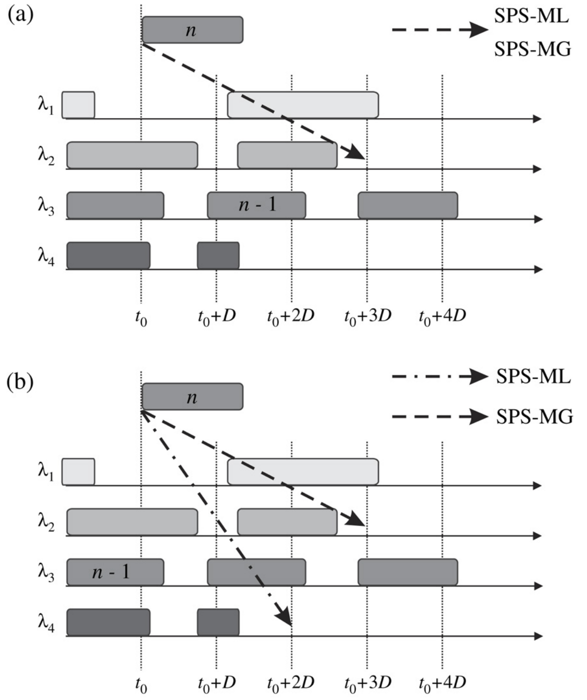

Fig. 3. Example of the SPS-ML and SPS-MG algorithms' behavior. (a) Packet $n - 1$ is still in the queue. (b) Packet $n - 1$ has already left the queue.

5.2. *Loose packet sequence*

The algorithms proposed for this case are called *Loose Packet Sequence with Minimum Length wavelength selection* (LPS-ML) and *Loose Packet Sequence with Minimum Gap wavelength selection* (LPS-MG) and have been originally proposed in [18]. Their behavior is the same as SPS policies, with the only difference that the delay requirements for not breaking the sequence are less strict and allow partial packet overlapping. LPS-ML and LPS-MG are therefore similar to SPS-ML and SPS-MG.

In particular, by considering the stream of packets belonging to the same LSP, now packet *n* is considered to be out-of-sequence when its first bit is sent on the output wavelength before the first bit of packet $n - 1$. This definition allows a partial overlapping between two consecutive packets. In practice if the first bit of packet $n - 1$ has been already sent when the first bit of packet *n* arrives, then the SCL schedules packet *n* without bothering about the sequence problem.

Following the same notation of the previous section, again we have two possibilities:

- $t_n < a_{n-1}$: then there is the chance to break the packet sequence definition.

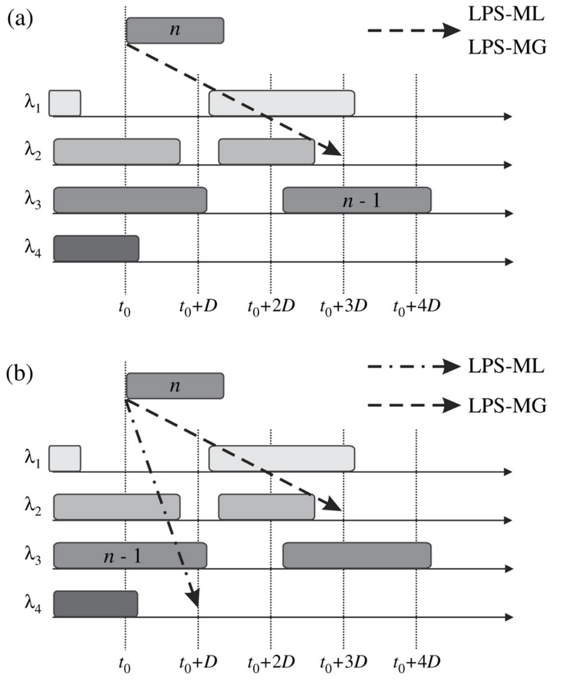

Fig. 4. Example of the LPS-ML and LPS-MG algorithms' behavior. (a) Packet $n - 1$ is still in the queue. (b) Packet $n - 1$ is leaving the queue.

- $t_n \geq a_{n-1}$: packet n arrives when packet $n - 1$ is leaving or has already left the node. Then the packet sequence is guaranteed.

In the first case, the amount of time needed to delay packet n in order to keep the sequence is:

$$D_{\min} = \left\lceil \frac{a_{n-1} - t_n}{D} \right\rceil D. \tag{2}$$

Fig. 4 shows how LPS-ML and LPS-MG work. In the first case (Fig. 4(a)) packet $n-1$ is still in the queue and packet n has to be delayed by three granularities to be brought back in order. Only the second wavelength can provide this delay and both algorithms behave similarly. In the second case, the first bit of packet $n - 1$ is leaving the switch. In this case the algorithms do not have constraints apart from that of choosing a feasible delay. For this choice LPS-MG behaves like SPS-MG and looks for the wavelength that minimizes the gap while LPS-ML resembles SPS-ML and selects the wavelength that minimizes the length of the queue.

6. Numerical results

The results presented here have been obtained by means of an ad hoc event-driven simulator, which has been widely tested in the past with different WDS algorithms. The simulator implements a 4 × 4 switch with 16 wavelengths per fiber, which results in a 64 × 64 optical switching matrix. Each input wavelength is supposed to carry 3 different LSPs, for a total of 192 incoming LSPs. As already outlined in Section 4, the simulator is bit-rate transparent and works with packet transmission and inter-arrival times only, making it scalable to different bit-rates and packet sizes.

The input traffic is generated according to the appropriate statistics on a per wavelength basis. Two types of input traffic are considered: a classical uniform traffic with Poisson distribution of the inter-arrival times and a more realistic self-similar model implemented with 32 multiplexed point arrival processes having Pareto distribution [19]. We assume asynchronous, variable-length optical packets, with an exponential distributed packet size with average value corresponding to a transmission time of the order of 1 μs. The input load on each wavelength is 0.8, equally divided among the LSPs and the traffic distribution to the outputs of the switching matrix is uniform.

In this simulation study we compare the performance of the new algorithms oriented to the packet sequence with the static, the EQWS [7], and the MINGAP algorithms [6]. It is reasonable to forecast that the new algorithms will show intermediate performance between the connectionless case (i.e. MINGAP) and the worst case (i.e. static). This is because the static algorithm does not use the wavelength domain for statistical multiplexing of contending packets at all, while the new ones do, but with less freedom than MINGAP, due to the constraints on the packet sequence preservation.

Fig. 5 plots the packet loss probability (PLP) for SPS-ML, SPS-MG and other WDS policies as a function of D normalized to the average packet duration, with $B = 4$ FDLs. Confidence intervals are not shown for all the curves for readability reasons. However, simulations have been carried on long enough in order to typically generate a billion packets, which makes loss measures up to 10^{-6} quite accurate. To prove this, the curve related to SPS-ML has been plotted with confidence intervals. It should be noticed that curve fluctuations are not caused by insufficient statistical data, but are most probably due to the mismatch between FDL buffer granularity and average packet size.

As expected the PLP shows a typical concave behavior as a function of D, with an optimal value that depends on the algorithm. It can be seen that both SPS algorithms significantly improve the performance of the static allocation, which is the only other algorithm

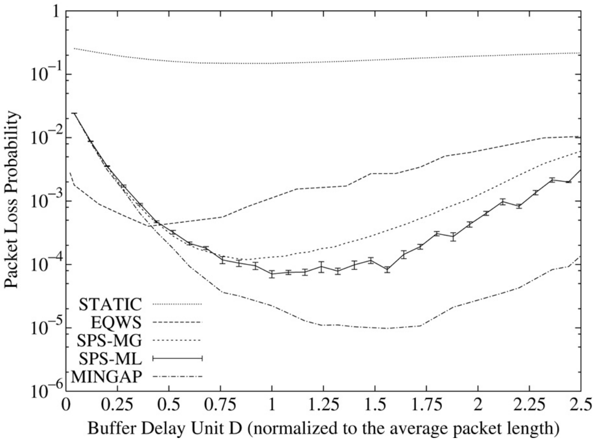

Fig. 5. Packet loss probability as a function of D, comparing the SPS algorithms with other algorithms under Poisson traffic model.

that avoids the out-of-sequence delivery. Moreover, they also improve the overall performance with respect to the EQWS, being closer to MINGAP. While SPS-MG and SPS-ML show the same behavior for small values of D, SPS-ML performs slightly better than SPS-MG when D is large. This is because by choosing the queue with the smallest gap, SPS-MG may buffer a packet in a queue that will become empty later (higher $D_{\min}$) than using SPS-ML (as in the case of Fig. 3). This means that the following packet belonging to the same LSP will have less queuing resources available due to the preservation of the packet sequence constraint. On the other hand SPS-ML, by choosing the shortest queue, provides a better utilization of the buffering space since it tries to keep the queues as short as possible, thus leaving more room for following packets.

Fig. 6 compares the performance of the strict sequence algorithms (SPS-ML and SPS-MG) with that of the loose packet sequence ones (LPS-ML and LPS-MG). Again the graph shows the performance as a function of the granularity D. We see that by allowing a partial overlapping between the packets a little further improvement can be obtained. Nonetheless such improvement is rather small, especially when weighted against the possible drawbacks of this solution. Hence we conclude that the loose packet sequence algorithms are less appealing than SPS-ML and SPS-MG. For this reason we focus the following results on these two algorithms only.

Fig. 7 shows that a significant improvement of the performance may be obtained with a small increase of the number FDLs in the buffer. For instance, for SPS algorithms adding 2–4 FDLs, the PLP decreases by two orders of magnitude. In this figure the values of D are those providing the lowest PLP for each algorithm. It is interesting to see from this graph that, as expected, the SPS algorithms outperform the other connection-oriented algorithms in spite of providing more functionality. The only case that provides a better PLP is the MINGAP case that, due to its connectionless behavior, has more freedom to exploit the wavelength domain for statistical multiplexing.

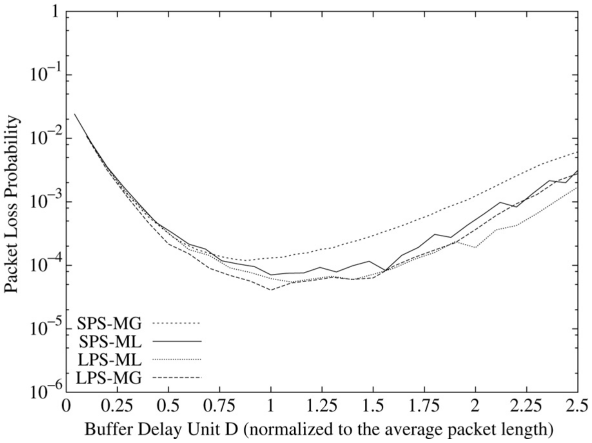

Fig. 6. Packet loss probability as a function of D, comparing the strict sequence definition (SPS-ML and SPS-MG) with the loose sequence definition (LPS-ML and LPS-MG) under Poisson traffic model.

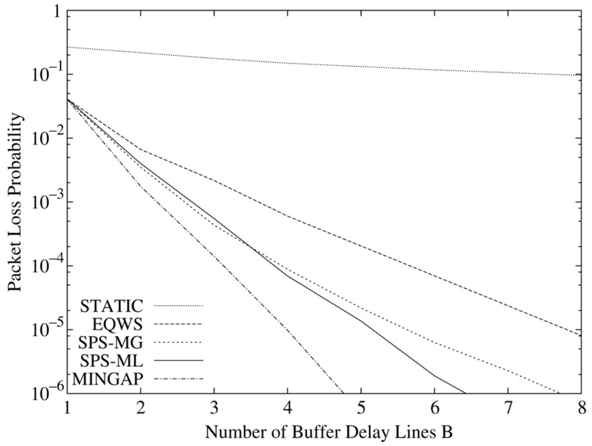

Fig. 7. Packet loss probability as a function of B, comparing the SPS algorithms with other algorithms under Poisson traffic model.

Table 2 compares the algorithms in terms of the percentage of strict out-of-sequence packets. SPS algorithms guarantee the correct sequence delivery as well as the static allocation, while both MINGAP and EQWS are not able to avoid the sequence breaking. Moreover, here we have considered a single switch scenario; by assuming n switches in series along a path, the percentages can increase accordingly.

The price to pay to keep the correct sequence delivery is the additional processing effort needed

Table 2
Out-of-sequence percentages

Algorithm	Out-of-sequence (%)
Static	0
EQWS	3.347
MINGAP	4.018
SPS-ML	0
SPS-MG	0

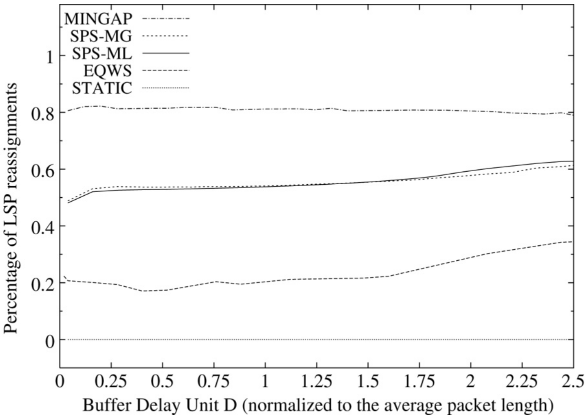

Fig. 8. Percentage of LSP-to-wavelength reassignments as a function of *D*, comparing the SPS algorithms with other algorithms under Poisson traffic model.

by SPS algorithms that reallocate the LSPs fairly more often than EQWS. In this sense, the percentage of LSP-to-wavelength reallocations is a measure of the processing effort that the switch control logic must deal with when applying a given WDS policy. Fig. 8 shows that the static algorithm does not require any reassignments, while the SPS algorithms present intermediate values between MINGAP (a per-packet strategy) and EQWS.

Finally, in Fig. 9 the PLP as a function of D with $B = 4$ is plotted considering a self-similar traffic model with Hurst parameter $H = 0.9$. This figure compares the SPS algorithms with the best performing algorithm (MINGAP) and the EQWS algorithm which needs the lowest processing effort. As expected, all algorithms perform worse under self-similar traffic than using the Poisson model; the PLP increases by an order of magnitude. Nevertheless, the relation among the algorithms remains the same as in Fig. 5.

7. Conclusions

In this paper we have analyzed the contention resolution problem in optical packet switches with MPLS-like connection-oriented operation. In particular, we have pointed out that previously proposed wavelength and delay selection algorithms (such as EQWS) do not maintain the sequence of the packet flows neither at a global switch level nor at the single traffic flow level, except for the static allocation solution.

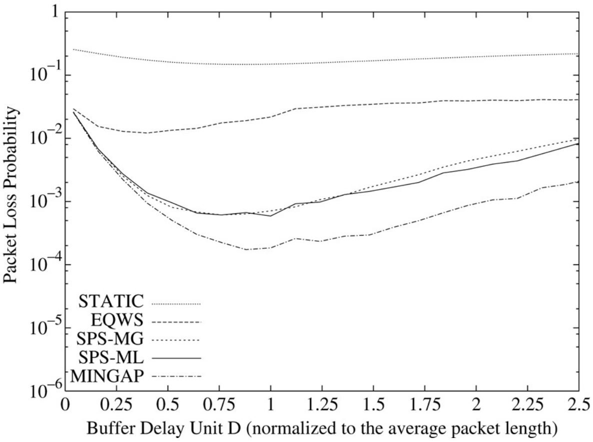

Fig. 9. Packet loss probability as a function of *D*, comparing the SPS algorithms with other algorithms under self-similar traffic model.

Four novel dynamic algorithms (SPS-ML, SPS-MG, LPS-ML, and LPS-MG) that maintain the packet sequence within the same LSP have been proposed. Two different strategies have been pursued: strict packet sequence where packets following the same LSP cannot overlap among themselves, and loose packet sequence where partial overlapping is allowed. The performance of these algorithms has been studied and compared with respect to previously proposed algorithms. Simulation results show that by accepting some additional processing effort it is possible to guarantee very low packet loss probabilities while avoiding the undesired out-of-sequence delivery. The results also show that the new algorithms improve the performance compared to EQWS.

In this paper we have considered a single switch scenario, letting the case of evaluating the algorithms in a whole network context for future studies.

Acknowledgments

This work was partially funded by the Commission of the European Union through the IST-FP6 Network of Excellence e-Photon/ONe and by MCYT (Spanish Ministry of Science and Technology) under contract FEDER-TIC2002-04344-C02-02.

References

[1] N. Ghani, S. Dixit, T.-S. Wang, On IP-over-WDM integration, IEEE Communications Magazine 38 (3) (2000) 72–84.

[2] E. Mannie (Ed.), Generalized Multi-Protocol Label Switching (GMPLS) Architecture, IETF, RFC3945, October 2004.

[3] T.S. El-Bawab, J.-D. Shin, Optical packet switching in core networks: between vision and reality, IEEE Communication Magazine 40 (9) (2002) 60–65.

[4] P. Gambini et al., Transparent optical packet switching: network architecture and demonstrators in the KEOPS project, IEEE Journal on Selected Areas in Communications 16 (7) (1998) 1245–1259.

[5] F. Callegati, Optical buffers for variable length packets, IEEE Communications Letters 4 (9) (2000) 292–294.

[6] F. Callegati, W. Cerroni, G. Corazza, Optimization of wavelength allocation in WDM optical buffers, Optical Networks Magazine 2 (6) (2001) 66–72.

[7] F. Callegati, W. Cerroni, C. Raffaelli, P. Zaffoni, Dynamic wavelength assignment in MPLS optical packet switches, Optical Network Magazine 5 (5) (2003) 41–51.

[8] E. Rosen, A. Viswanathan, R. Callon, Multiprotocol Label Switching Architecture, IETF, RFC 3031, January 2001.

[9] C. Guillemot et al., Transparent optical packet switching: the European ACTS KEOPS project approach, IEEE/OSA Journal of Lightwave Technology 16 (12) (1998) 2117–2134.

[10] K. Kompella, Y. Rekhter, LSP Hierarchy with Generalized MPLS TE, draft-ietf-mpls-lsp-hierarchy-08.txt, IETF draft, September 2002.

[11] D. Chiaroni et al., First demonstration of an asynchronous optical packet switching matrix prototype for multiterabit-class routers/switches, in: Proc. of 27th European Conference on Optical Communications, ECOC 2001, October 2001, Amsterdam, The Netherlands, vol. 6, 2001, pp. 60–61.

[12] D.K. Hunter, M.C. Chia, I. Andonovic, Buffering in optical packet switches, IEEE/OSA Journal of Lightwave Technology 16 (12) (1998) 2081–2094.

[13] L. Tancěvski, S. Yegnanarayanan, G. Castanon, L. Tamil, F. Masetti, T. McDermott, Optical routing of asynchronous, variable length packets, IEEE Journal on Selected Areas in Communications 18 (10) (2000) 2084–2093.

[14] S. Jaiswal, G. Iannacone, C. Diot, J. Kurose, D. Towsley, Measurement and classification of out-of-sequence packets in a tier-1 IP backbone, in: Proc. of INFOCOM 2003, March 2003, San Francisco, CA, vol. 2, 2003, pp. 1199–1209.

[15] M. Laor, L. Gendel, The effect of packet reordering in a backbone link on application throughput, IEEE Network 16 (5) (2002) 28–36.

[16] J.C.R. Bennett, C. Patridge, Packet reordering is not a pathological network behavior, IEEE/ACM Transactions on Networking 7 (6) (1999) 789–798.

[17] F. Callegati, D. Careglio, W. Cerroni, J. Solé-Pareta, C. Raffaelli, P. Zaffoni, Keeping the packet sequence in optical packet-switched networks, in: Proc. of 9th European Conference on Networks and Optical Communications, NOC 2004, June 2004, Eindhoven, The Netherlands, 2004, pp. 443–450.

[18] G. Muretto, C. Raffaelli, P. Zaffoni, Evaluation of packet reordering in optical packet networks with dynamic algorithms, in: Proc. of Photonics in Switching, PS 2003, September 2003, Paris, France, 2003, pp. 77–79.

[19] J.J. Gordon, Long range correlation in multiplexed Pareto traffic, in: Proc. of International IFIP-IEEE Conference on Broadband Communications, April 1996, Montreal, Canada, 1996, pp. 28–39.