# Architecture, Design, and Modeling of the OPSnet Asynchronous Optical Packet Switching Node

Wim A. Vanderbauwhede, *Member, IEEE*, and David A. Harle, *Member, IEEE*

***Abstract*—An all-optical packet-switched network supporting multiple services represents a long-term goal for network operators and service providers alike. The EPSRC-funded OPSnet project partnership addresses this issue from device through to network architecture perspectives with the key objective of the design, development, and demonstration of a fully operational asynchronous optical packet switch (OPS) suitable for 100 Gb/s dense-wavelength-division multiplexing (DWDM) operation. The OPS is built around a novel buffer and control architecture that has been shown to be highly flexible and to offer the promise of fair and consistent packet delivery at high load conditions with full support for quality of service (QoS) based on differentiated services over generalized multiprotocol label switching.**

***Index Terms*—Optical communication, optical fiber communication, optical switches, packet switching, photonic switching systems.**

## I. Introduction

MUCH of the current research and development in optical networks focuses on the implementation of a dynamically reconfigurable optical transport layer based on fast optical cross connects (OXCs) coupled with a suitable control and management architecture. Thus, even today, an optical transport network capable of supporting large numbers of high-capacity circuit-switched optical channels, with bit rates of 40–160 Gb/s, is being realized. Although, in such a scenario, it might seem that bandwidth is not an issue, this is not the case; economics will always demand that network resources are used efficiently. A major advantage of packet switching lies in its bandwidth efficiency and ability to support diverse services.

The OPSnet project [1] is an Engineering and Physical Sciences Research Council (EPSRC)-funded collaboration between the universities of Strathclyde, Essex, and Cambridge with participation from a number of industrial partners. Like its predecessor, WASPNET [2], the project aims to bring the packet switching concept into the optical domain. The WASPNET project demonstrated a prototype switch with a synchronous, synchronous digital hierarchy (SDH)-like architecture. Motivated by, in part, the increasing deployment of Gigabit Ethernet in access networks, OPSnet investigates the merits of direct asynchronous optical packet switching of variable length (Ethernet-like) packets. The objective of the OPSnet project is the design and implementation of a fast high-capacity optical packet switching node that can switch packets of variable length in an asynchronous fashion. Moreover, this design will explicitly support quality of service (QoS) requirements [3].

This paper intends to give an overview of the system-level design and architecture and simulated performance of the OPSnet optical packet switching node. Section II covers the design requirements. The node architecture is discussed in Section III, focusing on the need for a modular architecture. The design of the optical packet switching modules and the way QoS compliance is implemented is detailed in Section IV. This section also presents the results of performance simulations on the architecture. For details on the physical layer technology, the reader is referred to [4] and [5].

## II. Optical Packet Switch (OPS) Design Requirements and Economic Constraints

This section discusses the design requirements for an optical packet switching node. The physical layer (deployed technology), the networking layer, and the transport layer each impose specific requirements on the design. Performance requirements must be traded off against economic constraints.

### A. Physical-Layer Requirements

The target of the OPSnet project was to demonstrate optical packet switching at bit rates of 40 Gb/s, scalable to 160 Gb/s. To achieve this target, a technology for very fast switching is required. In circuit switching and burst switching, reconfiguration of the OXC is relatively infrequent, and the reconfiguration speed does not need to be very high. In contrast, for optical packet switching, the state of the switch changes with every arriving packet. Furthermore, the switching speed must be very high, because, while a part of the system is switching, no packets can be transiting that part of the node. This means that the available bandwidth is determined by the ratio of the switching time to the average packet length. However, the maximum packet length is limited by the buffer size; longer packets lead to longer buffering times, larger buffer depths, and more expensive buffers.

To create a transparent optical packet switching node, both port switching and wavelength translation are required (see Section II-B). For these reasons, the switching technology

Manuscript received January 17, 2005; revised March 30, 2005. The OPSnet project is funded by the Engineering and Physical Sciences Research Council (EPSRC) under the OSI grants scheme (R33427) [30].

W. A. Vanderbauwhede is with the Department of Computing Science, University of Glasgow, G12 8QQ Glasgow, U.K. (e-mail: wim@dcs.gla.ac.uk).

D. A. Harle is with the Department of Electronic and Electrical Engineering, University of Strathclyde, G1 1XW Glasgow, U.K. (e-mail: d.harle@eee.strath.ac.uk).

Digital Object Identifier 10.1109/JLT.2005.850023

must support very rapid space and wavelength switching. The OPSnet switching technology builds upon the know-how acquired during the WASPNET project [2]. The key switching element is a tunable wavelength converter (TWC) based on a semiconductor optical amplifier and a tunable pump laser. Where WASPNET proposed cross-gain modulation (XGM), OPSnet uses four-wave mixing, because XGM is not sufficiently fast. To achieve space switching, an arrayed-waveguide grating (AWG) multiplexer is used. This component essentially operates like a two-dimensional prism. Regardless of the ingress port, different wavelengths exit at different positions (egress ports). As illustrated in Fig. 1, the network wavelength is translated to the required internal wavelength using a TWC, sent through the AWG and translated back using a second TWC [1].

Because of the choice of technology, the OPSnet switch is inherently capable of wavelength translation. This is an advantage over other solutions, where the wavelength translation capacity is an add-on.

An important consequence of the choice of this technology is the need for a modular architecture. The number of wavelengths of the TWC and the number of ports on the AWG are both limited by technological and physical constraints. To create a monolithic OXC that can switch a circuit from any ingress port and wavelength to any egress port and wavelength, both the number of internal wavelengths and the number of AWG ports are the product of the number of ports and the number of wavelengths. This would obviously scale very poorly. The modular architecture is discussed in detail in Section III.

### B. Network and Transport Layer Requirements

Most requirements for an optical packet switching node are imposed by the network and transport layers of the OSI model. Essentially, they originate from the need for QoS.

*1) Contention Resolution Capability:* In a packet switch, contention occurs whenever a packet requests egress to a port that is occupied by another packet. Contention can be solved in the following three ways:

1) by dropping the contending packet;
2) by deflecting the contending packet to another port or wavelength;
3) by buffering the contending packet (using delay lines or using electronic memory) until the egress port is free.

Of these three, only buffering offers an immediate solution.

1) Port deflection is undesirable, because it is, in general, not compatible with end-to-end requirements as implemented in, for example, generalized multiprotocol label switching (GMPLS) [6]. The exception would be if the egress port were a fiber bundle, and the position of the fiber in the bundle would not be part of the GMPLS label. In that case, all fibers in the bundle would be equivalent, and port deflection within the bundle would be a valid way of solving contention. This is, however, generally not the case.

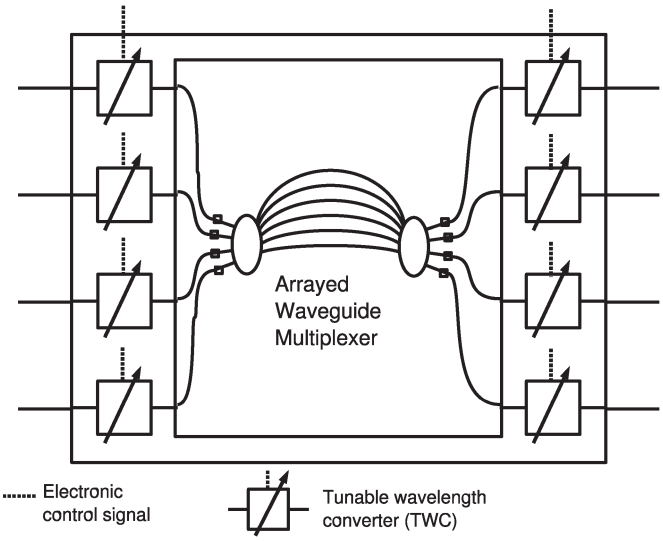


Fig. 1. Optical port switching using an AWG multiplexer and TWCs.

2) Wavelength deflection is, in general, undesirable for the same reasons: In GMPLS, the wavelength determines the label-switched path (LSP), so in general, wavelength deflection would switch a packet to a different LSP. It is possible to preserve the LSP while allowing wavelength deflection, but only if the GMPLS LSP would itself consist of a set of wavelengths, and if the position of the wavelength in the set would not be part of the GMPLS label. In such a case, all wavelengths in the set would be equivalent, and wavelength deflection within the set would be a valid way of solving contention. However, this cannot be a general assumption, as operators may prefer to control the bandwidth at the finest granularity, and therefore make use of the wavelength label.
3) Dropping the contending packet is obviously the poorest strategy, as it relies on the higher level protocols to retransmit the packet, causing round-trip delays and reordering issues.
4) Buffering does not have any of the abovementioned drawbacks and, moreover, enables the prioritization of traffic, based on, e.g., the differentiated services (DiffServ) specifications [7]. The drawback of buffering is the introduction of additional delay and jitter in the network. However, as we demonstrate in this paper, it is possible to design the optical buffers in such a way that the delay and jitter are negligible with respect to the end-to-end delay.

For these reasons, the OPSnet project has adopted buffering as the most appropriate approach to contention resolution in a packet-switched optical network that relies on GMPLS for QoS.

*2) Circuit-Switching Compatibility:* Current optical networks use wavelength-division multiplexing (WDM) circuit switching, and therefore the OPSnet node must be compatible with circuit switching. This requires that the OXC node is nonblocking, i.e., all combinations of circuits connecting ingress ports and egress ports must be possible, either for all possible configurations (strictly nonblocking) or by rearranging

established connections (rearrangeable nonblocking). The OPSnet node architecture, discussed in detail in Section III, is strictly nonblocking.

*3) Generalized Multiprotocol Label Switching Support:* GMPLS [8], [9] is a generalization of multiprotocol label switching (MPLS) [10], [11], which extends the concept of a label to, among others, a wavelength, frequency, time slot, or position in space. The basic idea behind (G)MPLS is to forward data along preestablished path. For every so-called LSP, bandwidth is reserved in advance. This connection-oriented approach with guaranteed bandwidth is an essential requirement for QoS.

In optical packet switching, forwarding of a packet is based on three "labels" (Fig. 2): the input port label of the OPS (which, in its turn, could consist of the label of a fiber bundle link and the number of the fiber in the bundle), the input wavelength, and the packet label. To be GMPLS compliant, the OPS control system must support the hierarchical label structure.

*4) Differentiated Services Support:* Traditional Internet protocol (IP) packet networks cannot guarantee that packets will not be delayed or even dropped, regardless of how important the packets are. With the convergence of voice, data, and video networks, the protocols deployed in the core networks must evolve. To accommodate the significant differences in the applications operating over large networks, QoS is critical. To guarantee a certain QoS level, it must be possible to prioritize the traffic. One of the emerging standards for traffic classes is DiffServ [7].

The DiffServ standard [12], [13] was developed by the Internet Engineering Task Force to provide a common methodology for implementing priority-based QoS. The standard defines three main traffic classes: expedited forwarding (EF), assured forwarding (AF), and best effort (BE). The AF group is further divided into four independent AF classes. Within each AF class, every packet is assigned one of three different levels of drop precedence. As the names indicate, the EF class has the highest priority, the BE class the lowest. The four AF classes have identical priorities, i.e., a packet from one AF will not be prioritized with respect to another AF class.

To support DiffServ, the OPS control system must implement the DiffServ per-hop behavior (PHB), which requires the ability to prioritize the packet loss and delay and conserve the packet ordering. This means that an OPS that cannot conserve packet ordering cannot be DiffServ compliant.

### C. Economic Constraints

Compared with the level of integration achieved in electronics, today's optical systems are still at the component level, comparable to building electronic circuits from discrete components such as transistors and resistors. There is no doubt that a high level of integration in optics would result in huge gains in performance and cost.

Although the technological requirements are very similar, the economic situation for optics is quite different compared with electronics. To achieve large-scale integration, the process

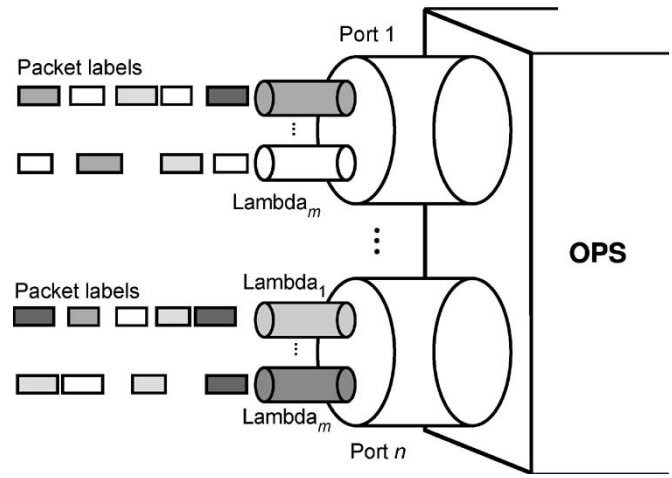


Fig. 2. GMPLS generalized labels for optical packet switching.

control must be very tight, and the production environment must be very clean. Building such a production facility is a huge investment and will only be considered if the return is guaranteed. Unfortunately, such major investments have not yet materialized.

Consequently, a complex OPS such as the OPSnet switch will have to be built from discrete components. This imposes serious constraints on the design, as the component count must be kept as low as possible. To illustrate the predicament, consider an optical buffering system. This system consists of a number of delay lines and switches. In discrete components, this means fiber loops and packaged AWGs and TWCs. Particularly the TWCs are very expensive, as the processing and packaging yield is very low. Furthermore, the system will require extra optical amplifiers as a result of losses incurred at the connectors. If this system would be integrated, this would eliminate the packaging cost and the connection losses. In the end, it would probably cost as much one of the discrete components.

However, as it is unrealistic to assume that large-scale integration for optics will be achieved in the near future, in this paper, we assume that the switch consists of discrete components. We will demonstrate how apparently more complex designs can lead to a lower component count.

## III. Optical Packet Switching Node Architecture

In this section, we present the architecture of the OPSnet optical packet switching node. After presenting the layer model underlying the architecture, we discuss the need for scalability and the modular OPSnet architecture that addresses this need.

### A. Layer Model for the Optical Packet Switching Node

The OPSnet switch architecture consists of three layers, which are shown in Fig. 3.

The data remains in the optical layer and does not require any optical–electrical/electrical–optical (OE/EO) conversion on traversal of the switch. Only the header information is extracted and processed electronically. The node control layer consists of

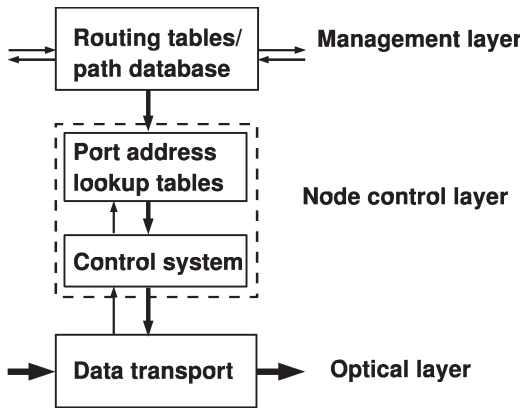


Fig. 3. OPSnet node layer model.

a dedicated event-driven asynchronous logic system to control the buffering and switching systems, and a set of content-addressable lookup tables. The choice of an asynchronous control system is motivated by the asynchronous nature and high bit rate of the traffic. The speed of synchronous logic is limited by clock skew, and the sampling of the incoming signal by the clock introduces additional uncertainty on the timing of the events. The lookup tables contain the egress port and wavelength and the new packet label. A content-addressable memory [14] allows lookup of this information based on the header label in a single operation (equivalent to a clock cycle on a synchronous system). The management layer corresponds roughly to the conventional router control system. It is responsible for updating the lookup tables based on the LSP allocation.

### B. Scalable Nonblocking Architecture

*1) Scalable Modular Architecture:* Any proposed OPS node must both be suitable for dense-wavelength-division multiplexing (DWDM) and be scalable in terms of the number of ports and wavelengths as well as in terms of bit rate. The number of ports and wavelengths should not be limited by the design, although the state of the art for the technology may impose its limitations. Such a scalability requirement has a major impact on the architecture and cannot be overemphasized.

The OPSnet modular OPS architecture is schematically represented in Fig. 4. It uses passive wavelength (de)multiplexers to separate the wavelength channels, wavelength translators, and three OPS stages forming a Clos network [15]. Every OPS module has the same number of ingress and egress ports. For the two outer OPS stages, this number is determined by the number of external wavelengths; for the middle stage, it is determined by the number of ports. Three stages are necessary to ensure that the switch is nonblocking, a requirement for backward compatibility with circuit-switched networks. A two-stage architecture would be sufficient for packet switching. For this reason, the middle stage does not have to be an OPS and a simple OXC (which is essentially a bufferless OPS) is sufficient. This is explained in more detail in Section III-B-2.

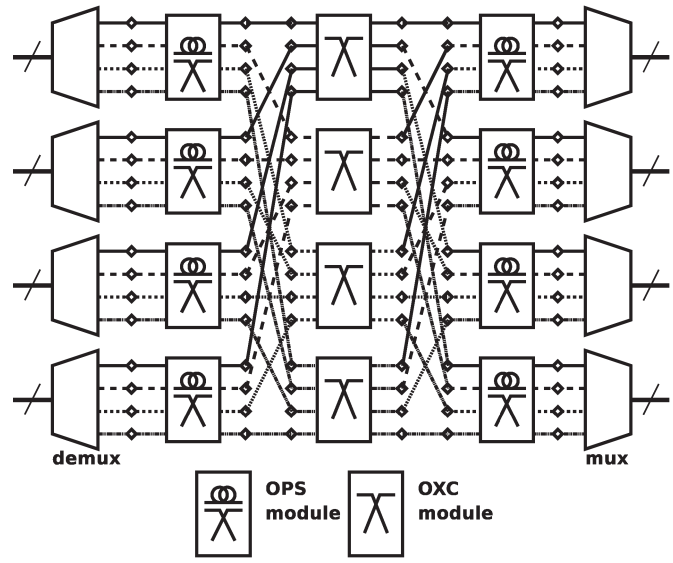


Fig. 4. Scalable modular OPS architecture.

This approach scales very well, because each individual OPS module does not require a large number of ports. In addition, the OPS design itself is simplified, because, from a system design point of view, every OPS module is essentially a single-wavelength switch. The employed technology for the OXC, arrayed-waveguide multiplexers with wavelength translation, uses multiple wavelengths for the actual switching [4]. For that reason, the actual implementation requires fewer wavelength converters than would be the case if the OXC adopted a different technology.

To compare the modular OPSnet node architecture with a nonmodular ("monolithic") node, Tables I and II show the required number of the key components in the architecture, as well as their wavelength range. Only the components for which the counts are different are shown. As a function of the number of ingress/egress ports and the number of wavelengths, the tables list the required number of components. The monolithic architecture requires one buffer module per wavelength per port; the modular architecture requires two, because there are two OPS stages. The monolithic case uses a single AWG. The number of internal channels required for this AWG is the product of the number of ports and external wavelengths (number of channels = number of ports × number of wavelengths). As the table shows, this number grows very rapidly. The number of OPS modules required for the modular architecture is twice the number of ports, but every module requires only as many internal channels as there are external wavelengths (number of channels = number of wavelengths). The modular architecture requires an additional OXC stage for circuit-switching compatibility. The component counts for this stage are listed in Table II(b).

The comparison shows clearly why the OPSnet architecture adopts the modular approach: The monolithic architecture requires a very large number of internal wavelength channels. This severely restricts the maximum number of ports and wavelengths: With the current technology, the number of

TABLE I
COMPONENT COUNT FOR THE MONOLITHIC ARCHITECTURE WITH CIRCUIT-SWITCHING CAPABILITY

| OPS node | | Buffer module | OPS node AWG | |
|---|---|---|---|---|
| #ports | #wvl | #comp | #comp | #channels |
| 4 | 4 | 16 | 1 | 16 |
| 8 | 8 | 64 | 1 | 64 |
| 16 | 16 | 256 | 1 | 256 |
| 32 | 32 | 1024 | 1 | 1024 |
| 64 | 64 | 4096 | 1 | 4096 |
| 128 | 128 | 16384 | 1 | 16384 |

TABLE II
COMPONENT COUNT FOR THE MODULAR ARCHITECTURE WITH CIRCUIT-SWITCHING CAPABILITY. (a) OPS COMPONENTS. (b) OXC COMPONENTS

(a)

| OPS node | | Buffer module | OPS module AWG | |
|---|---|---|---|---|
| #ports | #wvl | #comp | #comp | #channels |
| 4 | 4 | 32 | 8 | 4 |
| 8 | 8 | 128 | 16 | 8 |
| 16 | 16 | 512 | 32 | 16 |
| 32 | 32 | 2048 | 64 | 32 |
| 64 | 64 | 8192 | 128 | 64 |
| 128 | 128 | 32768 | 256 | 128 |

(b)

| OPS node | | OXC module AWG | | TWC (static) | |
|---|---|---|---|---|---|
| #ports | #wvl | #comp | #channels | #comp | #wvl |
| 4 | 4 | 4 | 4 | 16 | 4 |
| 8 | 8 | 8 | 8 | 64 | 8 |
| 16 | 16 | 16 | 16 | 256 | 16 |
| 32 | 32 | 32 | 32 | 1024 | 32 |
| 64 | 64 | 64 | 64 | 4096 | 64 |
| 128 | 128 | 128 | 128 | 16384 | 128 |

internal wavelength channels, and thus the maximum *ports* × *wavelengths* product, would be about 128.

*2) Nonblocking Architecture for Circuit-Switching Compatibility:* The OPSnet design uses the same number of internal and external wavelengths. This results in a nonblocking but not strictly nonblocking circuit switch. The switch is rearrangeable nonblocking as $m = n$ and the condition for a rearrangeable architecture is $m \geq n$ [16].

However, the architecture is not strictly nonblocking: Clos' theorem [15] states that a two-sided three-stage Clos network $\nu(m, n, r)$ is nonblocking in the strict sense if and only if $m \geq 2n - 1$, with $m$ as the number of internal wavelengths. In the OPSnet architecture, the number of internal wavelengths is equal to the number of external wavelengths, so the strict condition is not satisfied.

Although it is possible to make the architecture strictly nonblocking, this has actually a negative effect on the packet switching performance: The required buffer depth for the first stage decreases with the number of internal wavelengths, but this decrease is completely outweighed by the increase in the last stage. Consequently, using the same number of internal and external wavelengths is the best option for the packet-switched modular architecture. This is confirmed by the results of discrete-event simulations of the OPS node, presented in Fig. 5 for the cases of 8 and 12 external wavelengths. The traffic distribution is IP like [17] for the packet length while

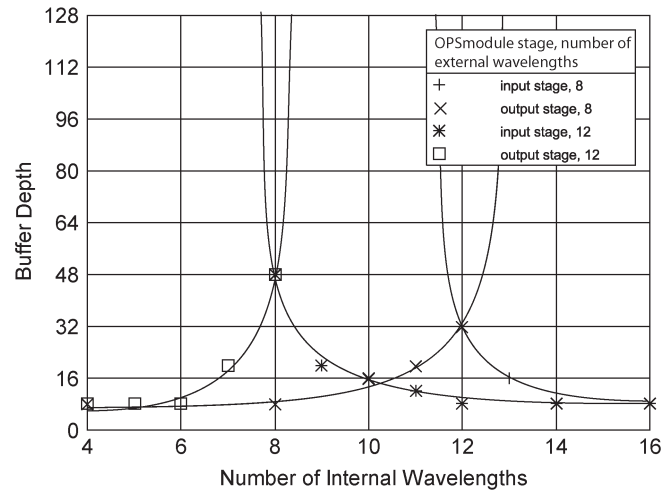


Fig. 5. Required buffer depth for packet loss $< 10^{-6}$ versus number of internal wavelengths (for 8 and 12 external wavelengths).

the interarrival times have a negative exponential distribution. The traffic load was 0.7.

The graph shows the buffer depth required for an acceptable packet loss ($10^{-6}$), for both the input stage and the output stage of the OPS. The total required buffer depth is the sum of the required buffer depths for the input and output stage. We find that when the number of internal wavelengths exceeds the number of external wavelengths, no solution is found: however large the buffer depth, the packet loss never drops to the acceptable level. Both curves intersect where the number of internal wavelengths equals the number of external wavelengths, and therefore the total required buffer depth is minimal when the number of internal wavelengths equals the number of external wavelengths.

It is important to note that (assuming $m = n$), the middle stage is actually redundant for optical packet switching. As soon as $n \geq r$ (the number of wavelengths is larger than or equal to the number of ports), complete connectivity is guaranteed. This means that the OXCs could be replaced by a static internal connection scheme. This is illustrated in Fig. 6. In practice, the OXCs will be present (for backward compatibility with circuit switching, as explained above), but the TWCs between the OPS module in the first stage and the OXC module in the middle stage do not have to change the tuning wavelength on a per-packet basis. The only reason to change the internal wavelength would be to achieve a nonblocking configuration when the node is being used for circuit switching. Consequently, these TWCs have relaxed switching speed requirements, and as such are relatively inexpensive. This is equally the case for the wavelength converters at the output ports of the last stage, as they translate any incoming wavelength to a fixed network wavelength.

## IV. OPTICAL PACKET SWITCH DESIGN FOR QUALITY OF SERVICE PROVISION

One of the most important properties of a future packet switching node in the core network is the ability to provide

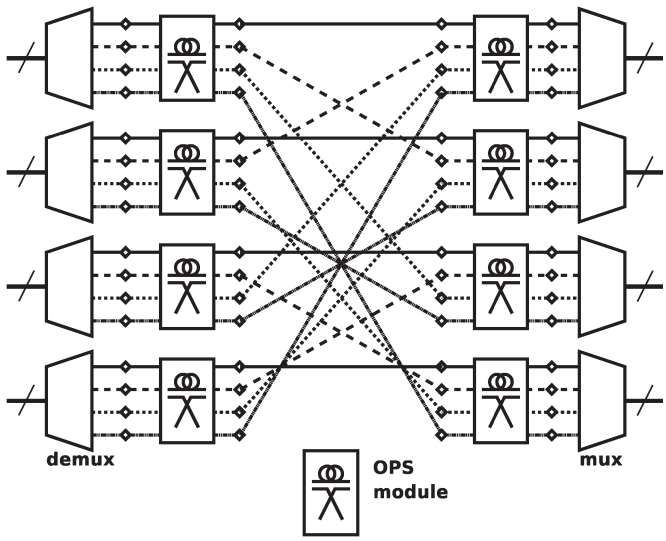


Fig. 6. Scalable modular OPS architecture with static internal connections.

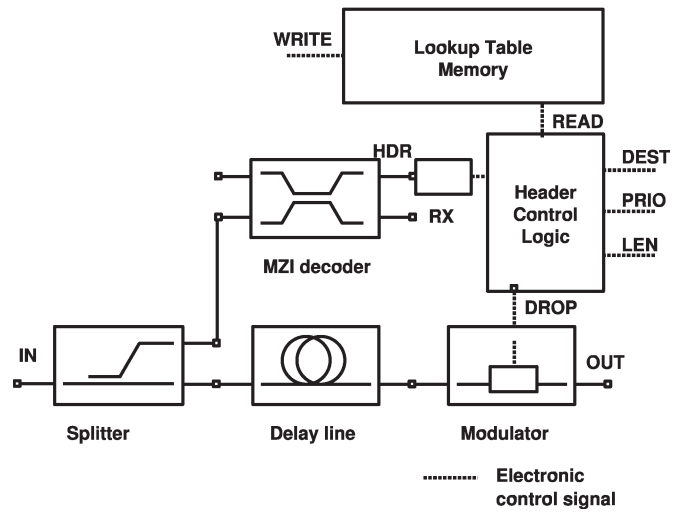


Fig. 7. Header processing module.

QoS. Asserting QoS entails that throughput, loss, delay, jitter, etc., are guaranteed to stay within preset limits. One of the solutions proposed to guarantee QoS in optical networks is to use DiffServ over GMPLS [3]. This approach was adopted for OPSnet as it is a generic solution based on emerging standards. The end-to-end requirements are covered by GMPLS, while DiffServ provides the traffic prioritization. This section explains the OPSnet design solutions which enable the provision of explicit QoS based on such a scheme.

### A. Low-Loss Low-Latency Optical Packet Switch Design

The rationale behind the OPSnet design was to create a scalable OPS architecture with low packet loss and low latency, while at the same time offering full control over packet ordering and traffic prioritization.

*1) Low-Latency Header Processing:* The OPSnet switch leaves the packet in the optical domain and does not change the packet payload. The optical signal is monitored and only the header is processed electronically. The header processing module is schematically depicted in Fig. 7.

To facilitate header processing, the OPSnet project adopted a DPSK header encoding technique [18]. Using a phase modulator, the header information is encoded as a discrete differential phase shift on the payload signal. Demodulation is achieved without disturbing the payload, simply by sending the signal through a Mach–Zehnder interferometer (MZI). In principle, this technique enables encoding the header in parallel with the payload signal; however, experiments show that phase jitter causes deterioration of the payload signal when passed through the MZI. For this reason, the OPSnet header is encoded on a sequence of high bits preceding the actual payload. This sequence is rewritten at every hop without modifying the payload. This approach combines the power of simple header encoding and decoding offered by differential phase shift keying (DPSK) with superior payload bit error rates (BERs). The overhead caused by the serial header encoding is small as the OPSnet header format is simple, requiring less than 50 b [19]. Such a short header has the further benefit of ensuring short processing times, in particular the time required for conversion from the serial optical signal to parallel electronic signal used by the electronic header processing unit.

The electronic header processing unit first performs an error check on the header data (in real time) and drops any packet with a corrupted header. If the header is intact, the GMPLS packet label, packet length, and DiffServ packet priority are extracted; a label-based lookup retrieves the destination port and internal wavelength (for modules in the first stage) or the new header label (for modules in the last stage). Every input port of every OPS module has a dedicated lookup table. This approach keeps the lookup tables small which reduces the lookup times. The OPSnet header encoding and processing scheme adds a latency of less than 5 ns (at 100 Gb/s). The serial-to-parallel conversion takes 2 ns, because the DPSK header signal is encoded at a quarter of the payload bit rate. Any remaining latency is caused mainly by the lookup process which, with current technology, takes less than 1 ns.

*2) Low-Loss Low-Latency Optical Buffer Architecture:* As explained in Section II-B, the OPSnet switch uses optical buffering for contention resolution. In practice, optical packets can only be buffered using delay lines, as there is no practical optical equivalent to electronic memory. In synchronous OPSs, packets are generally buffered in series in delay lines with lengths equal to an integer number of time slots [2]. This approach is possible, because contention occurs on a per-slot basis and can therefore be solved by delaying the packets for a fixed number of slots.

In asynchronous packet switching, this is not possible, because contention occurs as soon as there is partial overlap between two packets, and because the status of the ports can change over intervals much shorter than the packet length, i.e., the gaps between the packets are not equal to the packet length. For this reason, a fixed-length buffer is not an option. Variable-length buffers could solve this problem, as it is in principle

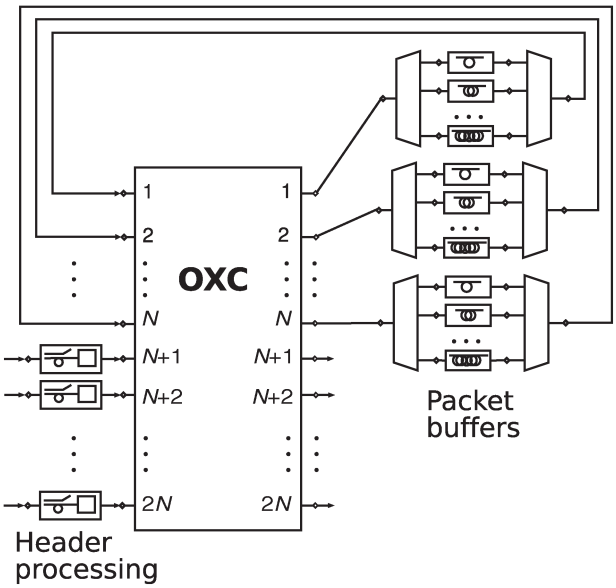


Fig. 8. Multiplexed system of recirculating serial buffers.

possible to calculate, based on the length of the packets, how long it will take before the port is free. However, this would lead to a very complex buffer design. A possible alternative would be to use a multiplexed system of recirculating serial buffers, as proposed in WASPNET [2]. The recirculating buffers in the WASPNET design (Fig. 8) each have a fixed length, equal to 1, 2, 4, ... slots. This is possible, because even if the packets would have variable length, the timeslot length is fixed. To be most efficient, a corresponding design for the OPSnet switch would require variable-length recirculating buffers to accommodate a variable-length series of variable-length packets with unknown interarrival times, which would be extremely complex.

Apart from the complexity of such a design, two important issues would still remain.

1) The sojourn time of packets in a serial buffer can become very long, leading to unacceptable packet latency. This is because the time windows during which the destination port is free will in general not coincide with the moment at which the packet reaches the exit point of the recirculating buffer. As a consequence, the packets in the buffer will be forced to make many iterations, leading to very long sojourn times. This in its turn would lead to a high buffer overflow probability, as the buffer would empty very slowly. Simply increasing the serial buffer capacity is not an appropriate solution, as the sojourn time is proportional to the length of the buffer delay line.
2) A serial buffer does not easily facilitate packet management in terms of drop precedence and latency. Once a packet is in the serial buffer, it cannot be dropped until it reaches the exit. Consequently, it is not possible to free up buffer space for incoming packets by dropping buffered packets with a lower priority. For the same reason, it is also not possible to assert packet priority.

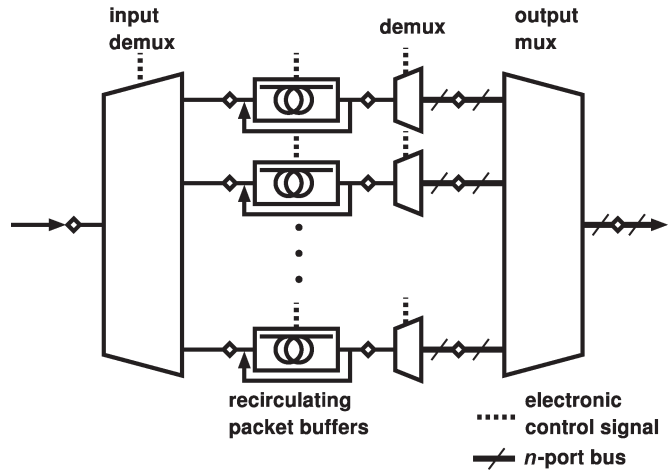


Fig. 9. Parallel recirculating packet buffer array.

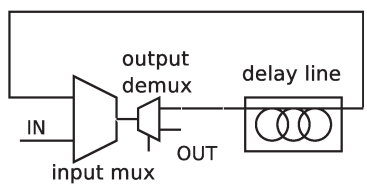


Fig. 10. Fixed-length recirculating buffer.

As a result of the above analysis, the OPSnet switch does not use a serial buffering system but employs a parallel per-packet buffer architecture as schematically illustrated in Fig. 9.

In such an architecture, every packet is stored in an individual recirculating buffer, which eliminates head-of-line blocking. The input demultiplexer control system keeps track of the empty buffers and decides, in the case of buffer overflow, which packet to drop. The output multiplexer is a passive star coupler. On leaving their respective buffers, the packets are switched to the appropriate internal wavelength required for the AWG prior to entering the output multiplexer, a process necessary in order to avoid contention at the inputs of the AWG. This parallel architecture with transparent output multiplexer, combined with a special design for the recirculating buffer, results in very low packet loss for moderate buffer depths, as illustrated in Section IV-B.

*3) Recirculating Packet Buffer Design:* A recirculating buffer is the closest optical equivalent to an electronic memory element: The optical packet circulates in the buffer loop until it can leave, just as an electronic packet would be stored in an electronic memory until it can leave. Such buffer architectures have extensively been reported [20]–[22].

A recirculating buffer is essentially a closed-loop delay line with an input multiplexer and an output demultiplexer. When considering a recirculating buffer for a single packet, a number of designs are possible. The following schematics serve to illustrate the principle, but are not meant to represent the actual implementation.

In the simplest case (Fig. 10), the length of the delay line is fixed.

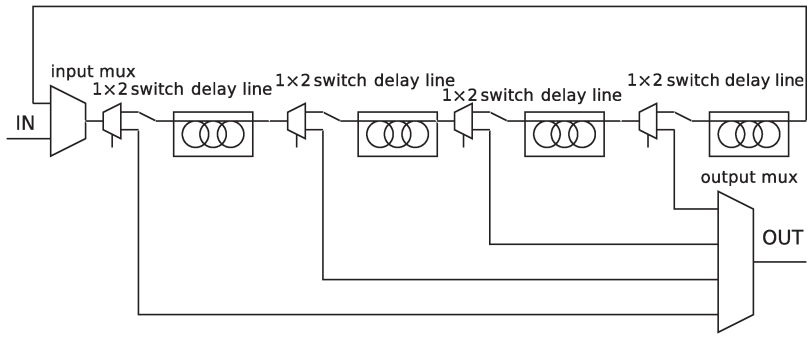


Fig. 11. Multi-exit recirculating buffer.

In this design, the request frequency (i.e., the frequency at which a packet can request to leave the buffer) is determined by the length of the delay line. As this length must be at least equal to the length of the longest possible packet, the egress probability will be low. To increase the egress probability, the OPSnet switch uses a multiexit recirculating buffer. In this design (Fig. 11), the delay line has multiple exits located at regular intervals along the loop. This means that the request frequency is determined by the distance between the exits, rather than by the length of the loop. The total loop delay equals the maximum packet length.

Ideally, if the delay between two subsequent exits is smaller than or equal to the delay between the end of a packet and the start of the next packet (the minimum gap width), then every time a gap in the traffic occurs, it is guaranteed to coincide with a packet egress request. However, as can be seen from the simulation results shown in Fig. 12, even a relatively small number of exits lead to a dramatic improvement in performance. The figure shows the behavior of a four-port OPS with a buffer depth of 16; the traffic distribution is IP-like for the packet length, while the interarrival times have a negative exponential distribution.

From Fig. 12, it is very clear that the multiexit buffer requires a much smaller buffer depth for the same load. Furthermore, it can be observed that for more than eight exits, increasing the number of exits has a relatively small impact. Thus, the increased performance (lower loss and lower latency) offered by the multiexit architecture far outweighs the increased complexity of the design and the resulting higher cost per unit. In itself, the results presented are major justification for the choice of architecture and the use of multi-exit buffers. A proposed implementation of the multiexit buffer architecture and a feasibility analysis can be found in [23].

*4) OPSnet Optical Packet Switching Module Architecture:* The complete OPSnet optical packet switching module architecture is illustrated in Fig. 13. A passive multiplexer (AWG) combines the paths from all inputs. The additional processing depends on the position of the OPS module: In the first stage, the egress wavelength will be translated to the internal wavelength required for switching the packet through the intermediate OXC. As discussed in Section III-B, this is a static translation. In the last stage, the egress wavelength will be translated to a fixed external wavelength depending on the egress port. The GMPLS packet label swapping module is discussed in Section IV-C.

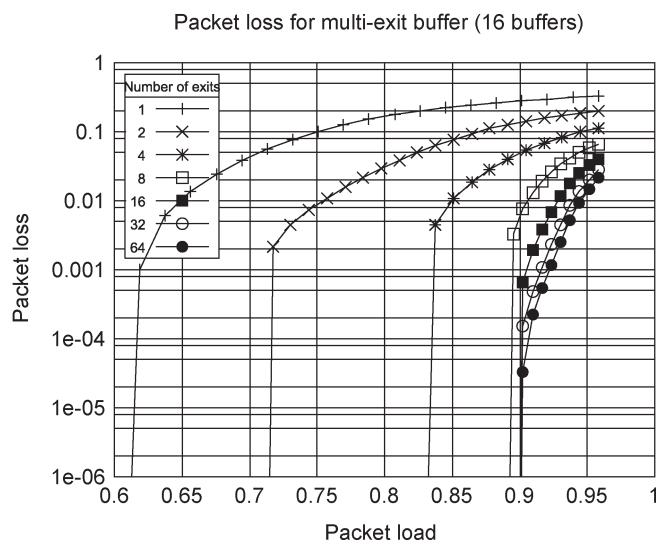


Fig. 12. Performance of the multi-exit buffer for varying number of exits, compared with the fixed-length buffer.

The buffer architecture is conceptually simple. The complexity resides mainly in the electronic control system, which is responsible for scheduling, prioritization, ordering, preemptive drop, etc. The control logic is asynchronous and event driven; most subsystems are operating independently and communicating via signals when necessary. This approach ensures that the design remains manageable and scalable. Shifting the complexity from the optical to the electronic domain is key to reducing the cost of the switch, as contrary to optical circuitry, the cost of electronic integrated circuits (ICs) does depend on their complexity. Furthermore, although synchronous logic is prevalent, asynchronous logic is well established for specific applications [24].

## B. Performance of the OPSnet Optical Packet Switch

This section illustrates the performance of the OPSnet design in terms of packet loss and latency. In particular, the influence of traffic shaping and traffic aggregation is discussed.

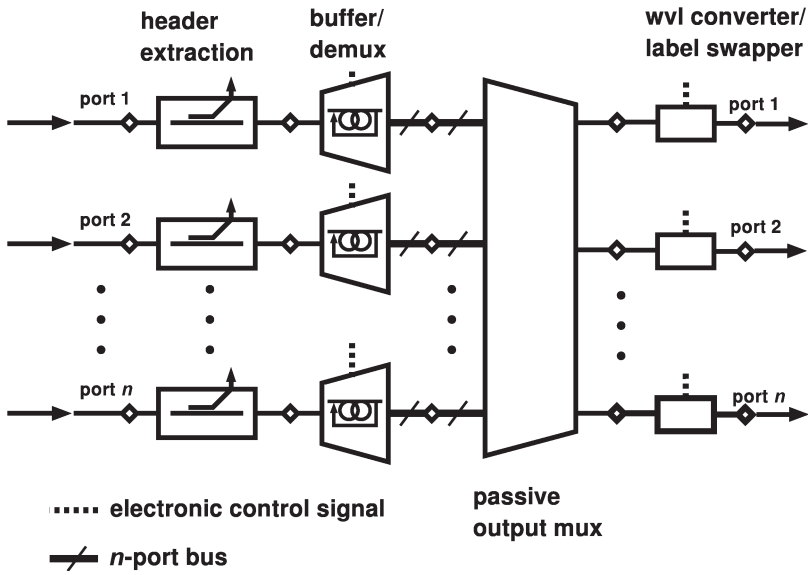


Fig. 13. Architecture of the OPSnet optical packet switching module.

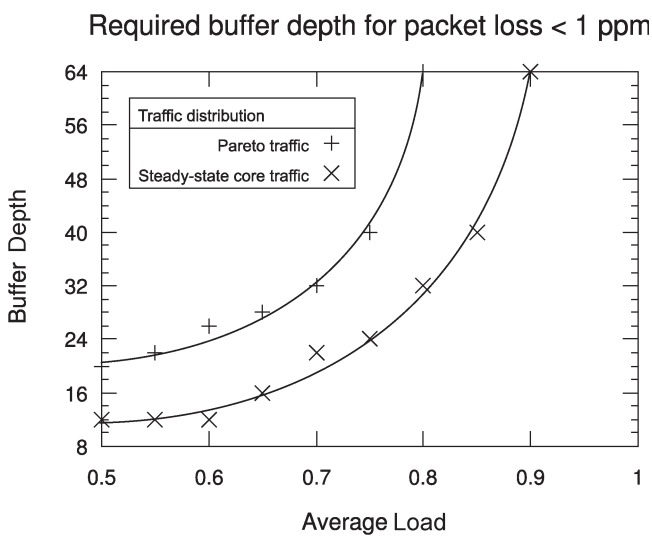


Fig. 14. Influence of steady-state core traffic on maximum sustainable load.

*1) Traffic Shaping Improves Network Performance:* The OPSnet OPS architecture has a very strong traffic shaping effect. Fig. 14 shows a comparison of the performance under Pareto ingress traffic and steady-state core traffic [25]. Pareto traffic is defined as traffic with a Pareto (power law) distribution of the interarrival times. Steady-state core traffic refers to the distribution of the interarrival times of packets which have circulated in the core network until a steady state is obtained, assuming no new traffic enters the network. As a measure of the performance of the switch, we use the required buffer depth to achieve a packet loss of less than 1 ppm (packet per million). As can be observed from Fig. 14, the performance under steady-state core traffic is much better than under Pareto traffic, which indicates that the traffic shaping caused by the switch improves the network performance.

The strong traffic shaping effect is caused by the combination of statistical multiplexing and the absence of head-of-line blocking (as the packets are buffered in parallel). This can be most easily understood if one assumes a filled buffer with an infinite buffer depth. In that case, the egress traffic distribution is governed by the probability that a packet can leave the buffer at a given time. Assuming a fixed packet length, it is obvious that this is a Poisson process, as every buffer has the same egress probability. As the actual packet length distribution consists of a discrete set of fixed packet lengths (IP over Ethernet), the actual resulting distribution will still be a negative exponential (Poisson) distribution.

In practice, if the buffer is dimensioned to have negligible packet loss, the approximation of an infinite buffer depth still holds. The assumption of a filled buffer will only hold if the load is high. Consequently, the traffic shaping will be more pronounced for high loads. The traffic shaping creates a more Poisson-like distribution, which results in improved switch performance. This is illustrated in Fig. 15, which shows the packet loss as a function of the load for a given buffer depth, with Poisson ingress traffic and Pareto ingress traffic.

*2) Packet Aggregation Improves Network Performance:* The packet payload in the OPSnet network consists of aggregated IP packets. We assume that the access networks use some type of Gigabit Ethernet, which is likely considering the growing deployment of this technology. Consequently, the IP packets will arrive encapsulated in Ethernet frames, which determines the minimum and maximum packet length (46/1500 B).

The OPSnet edge routers (developed as part of the OPORON project [26]) perform a degree of class-based traffic aggregation, mainly to reduce burstiness (self-similarity). It has been shown [27] that a significant reduction in burstiness can be achieved with minimal impact on the end-to-end delay. The aggregation is, in simple terms, the combination of packets with

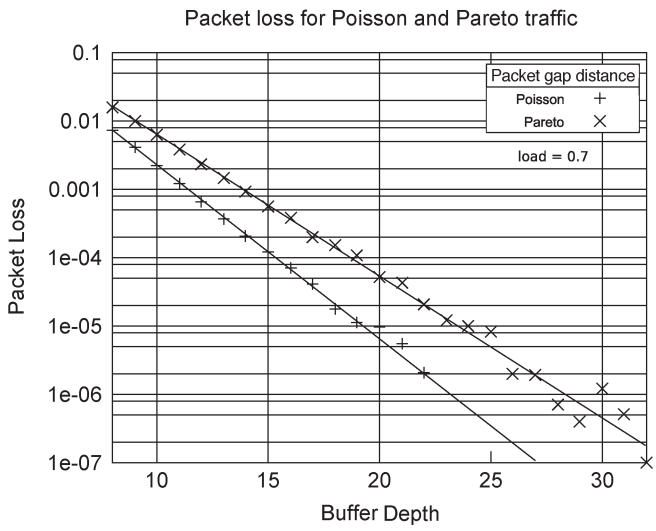


Fig. 15. Switch performance for Poisson and Pareto traffic.

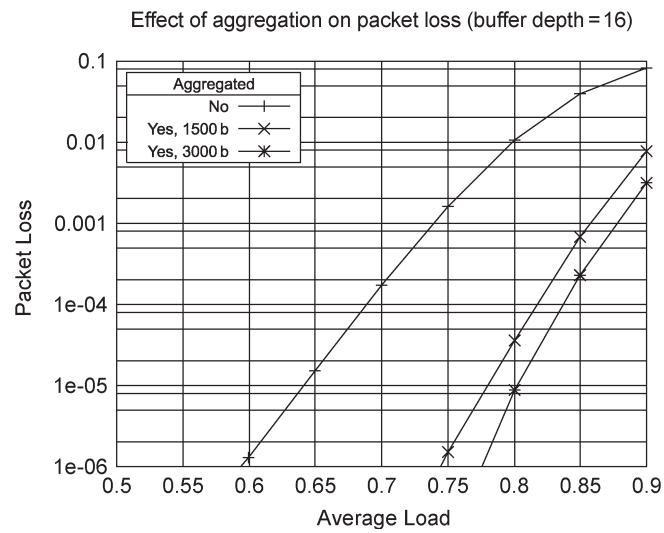


Fig. 16. Effect of traffic aggregation on packet loss.

a common label into a single extended packet. As a result, the switch has to deal with fewer, longer packets. As illustrated in Fig. 16, this again significantly reduces the overall packet loss. The main reason for this is that the switch performance is relatively insensitive to the packet length. This is a result of the multi-exit buffer architecture: Once the packet is in the buffer, the length of the packet is irrelevant. Consequently, traffic aggregation will not increase the packet latency induced by the switch. However, the interarrival time of the ingress traffic is a critical parameter; the traffic aggregation leads to a longer average interarrival time for the same load compared to non-aggregated traffic.

*3) Buffer Design Results in Low Packet Latency:* The OPSnet OPS buffers are designed to have the lowest possible latency (considering an asynchronous switch for packets with variable length). To keep the latency, i.e., the sojourn time of the packet in the switch, as low as possible, the recirculating buffers adopt a novel design which maximizes the egress probability—discussed in Section IV-A-3.

The simulation results show that, for typical IP-type traffic, less than 1 ppm has a latency of more than 4 $\mu$s (at 100 Gb/s) (Fig. 17). As the typical latency specifications for applications such as Voice over IP (VoIP) or video are less than 100 ms (ITU-T Recommendation G.114 [28]), the latency of the OPSnet OPS can be considered as negligible.

It is interesting to note that a low-latency design reduces the probability for buffer overflow and thus the packet loss. In the case of a constant load with no variance, the latency does not affect the packet loss. However, if the traffic distribution and, in particular, the distribution of the interarrival times, has a high variance, then temporarily the arrival rate of the packets can be much higher than the departure rate. In this case, if the sojourn time of the packet in the buffer is long, the buffer depth required to prevent buffer overflow will be larger than for short sojourn times. This is the main reason why the performance of the multiexit buffer design is more than proportionally better when compared to the fixed-length buffer.

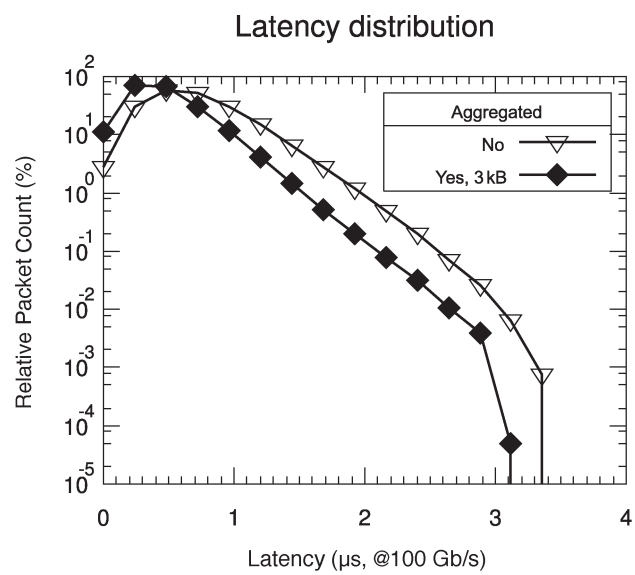


Fig. 17. Packet latency.

## C. GMPLS Compliance

As represented by the layer model shown in Section III, the OPSnet switch interacts with the management layer exclusively via a set of shared lookup tables. The management layer is responsible for building and updating the tables; the switch control system has read-only access to the tables and indicates in which intervals updates can occur. The switch control system is actually network agnostic: The information in the lookup tables pertains only to the local ports and wavelengths. Such a loosely coupled system has the advantage that any change in management protocols will not entail modifications to the underlying system. Nevertheless, to support a given protocol, a system architecture must have certain capabilities. To support GMPLS, the switch must support the concepts of generalized hierarchical labels and label swapping [6], [11].

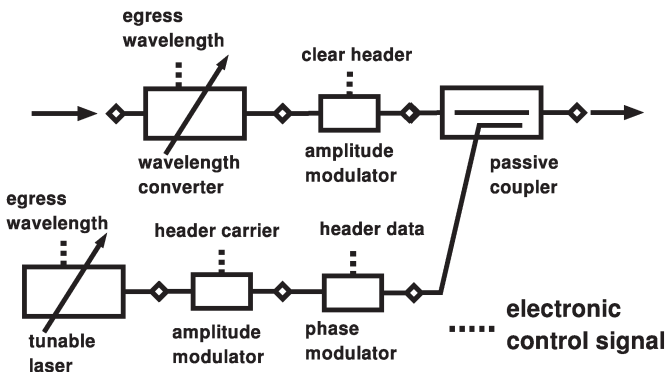


Fig. 18. OPSnet header swapping.

The OPSnet switch control uses label-indexed lookup tables to determine the outgoing port and wavelength and the new header label; the key being the label extracted from the packet header. As every input port and wavelength has its own lookup table, this approach, combined with the capability of rewriting the header at every hop, naturally supports generalized path labels.

The lookup tables are read by the switch control as soon as the header information has been extracted. At this point, the control will raise a flag to prevent the management layer from updating lookup table. The lookup typically takes less than 5 ns, whereas a typical packet duration at 100 Gb/s is about 100 ns. This means that the management layer has sufficient time to update the table between lookups.

The architecture supports header rewriting at every hop, as required by the (G)MPLS protocol [6], [11], because of the basic MPLS forwarding technique of label swapping, i.e., looking up an incoming label to determine the outgoing label.

As discussed under Section IV-A, the header information is encoded in parallel with the payload using a DPSK technique [18]. To implement label swapping in the OPSnet OPS, the original header sequence is suppressed using an amplitude modulator, and a new sequence is generated. At the same time, the header information is DPSK encoded on this carrier sequence using a phase modulator (Fig. 18).

### D. DiffServ Integration

As discussed in Section II-B-4, GMPLS compliance alone is not sufficient to guarantee QoS. Therefore, the OPSnet switch integrates the traffic prioritization mechanisms as proposed in the DiffServ specification [7].

To support the PHB as defined for the different DiffServ classes, the OPS must be able to 1) distinguish between the different DiffServ classes and 2) adapt the PHB according to each class, i.e., prioritize the traffic.

The traffic parameters that the OPS must control are as follows:

1) the packet latency;
2) the packet order;
3) the packet loss.

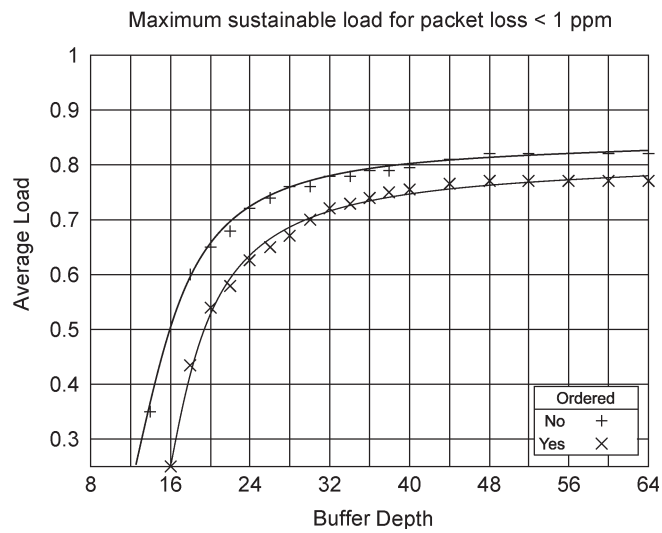


Fig. 19. Influence of conservation of packet order on the maximum sustainable load.

*1) Packet Latency Prioritization:* Because of the intrinsic low latency of the OPSnet switch (cf. Section IV-B-3), it is, in general, not beneficial to prioritize the packet latency. Nevertheless, it is possible to do so if required, because the buffer egress control system is DiffServ aware: The ordering first-in/first-outs (FIFOs) are structured to reflect the DiffServ classes.

*2) Packet Ordering:* According to RFCs describing the AF and EF PHB [12], [13], conserving the packet order is required for EF class traffic and for every individual AF class. The OPSnet switch design can conserve packet order if required as all packets are buffered by default. The buffer control keeps the packets in order by keeping track of the packet buffer addresses in FIFOs on a per-destination per-class basis. Fig. 19 shows the influence of conservation of packet order on packet loss for aggregated traffic. The penalty is relatively small, due to the use of multiexit buffers. When properly dimensioned, the multiexit buffer ensures that packets will be able to leave on the first free slot. As a result, the majority of packets will leave the buffer after the minimum sojourn time. Consequently, the probability that packets would leave out of order is small, which explains why conservation of packet order has only a small impact.

*3) Packet Loss Prioritization:* The OPS buffer control supports packet drop prioritization in accordance with the DiffServ classes. The packet loss requirements are most stringent for EF traffic and most relaxed for BE traffic. The four AF classes are further subdivided in three drop precedence levels. To prioritize the drop behavior of the OPS, a preemptive drop mechanism has been implemented (Fig. 20).

1) If the buffer is full when a packet arrives, check the packet class.
2) For non-BE packets, drop a packet from a lower class, starting with BE.
3) If no BE packets are available, conduct the following steps.

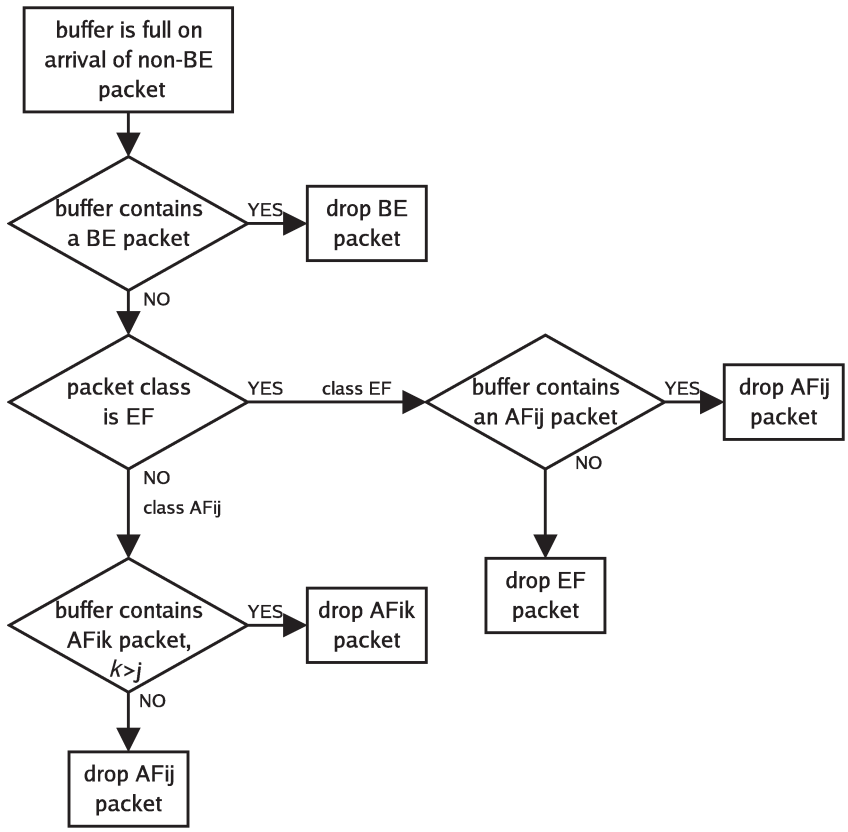


Fig. 20. Preemptive drop scheme.

a) If the arriving packet belongs to an AF subclass, drop a packet with a lower drop precedence, if any; otherwise, drop the packet itself.
b) If the arriving packet belongs to the EF class, drop the packet with the lowest drop precedence of any AF subclass, if any; otherwise, drop the packet itself.

4) BE packets are dropped immediately.

To implement such a mechanism, the buffer control system needs to keep track of the buffer addresses for all packets. This is implemented as a series of FIFOs (one per class/subclass/drop level).

Fig. 21 shows the effect of the preemptive drop scheme on EF class packet loss for varying network load. In this simulation, the EF fraction was fixed at 10% and the AF fractions at 15% for each AF class. As expected, the preemptive drop mechanism results in a dramatic reduction of nearly three orders of magnitude in EF class packet loss. The results illustrate clearly the effects of the preemptive drop mechanism. Because this mechanism results in such a strong packet loss reduction, it was necessary to increase the overall packet loss. This was simulated by using nonaggregated Pareto traffic and reducing the buffer depth. A detailed analysis of the packet drop probability as a function of the DiffServ class mix (the relative amounts of EF, AF, and BE traffic) can be found in [29].

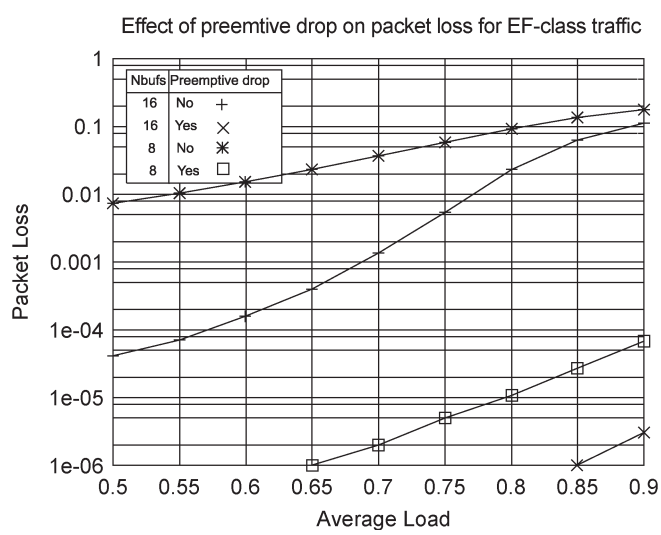


Fig. 21. Effect of preemptive drop on EF class packet loss.

## V. Conclusion

In this paper, an overview of the system design research work carried out in the frame of the Engineering and Physical Sciences Research Council (EPSRC) project OPSnet has been presented. The architecture of the OPSnet asynchronous optical packet switching node and the design that implements quality of service (QoS) support has been discussed. The switch has

a modular architecture that can be scaled to large numbers of ports and wavelengths, thus supporting dense-wavelength-division multiplexing (DWDM) and fiber bundles. The switch control architecture is generalized multiprotocol label switching (GMPLS) compliant and the design fully supports DiffServ traffic handling. Several techniques have been implemented to achieve low packet loss and low latency: a novel recirculating buffer design with very low latency, buffer occupancy equalization, traffic aggregation, and traffic shaping.

## ACKNOWLEDGMENT

The authors would like to thank in particular the input from D. Klonidis, R. Nejabati, and T. Politi from the University of Essex.

## REFERENCES

[1] W. Vanderbauwhede, D. Harle, and D. G. Smith, "Opsnet: An optical packet-switched network," in *Proc. PHOTON02*. [CD-ROM], Cardiff, U.K., Sep. 2002.

[2] D. K. Hunter, M. H. M. Nizam, K. M. Guild, J. D. Bainbridge, M. C. Chia, A. Tzanakaki, M. F. C. Stephens, R. V. Penty, M. J. O'Mahony, I. Andonovic, and I. H. White, "WASPNET—A wavelength switched packet network," *IEEE Commun. Mag.*, vol. 37, no. 3, pp. 120–129, Mar. 1999.

[3] X. Xiao, A. Hannan, B. Bailey, and L. Ni, "Traffic engineering with MPLS in the Internet," *IEEE Network*, vol. 14, no. 2, pp. 28–33, Mar./Apr. 2000.

[4] D. Klonidis, G. Zervas, C. T. Politi, D. Simeonidou, and M. J. O'Mahony, "Fast control circuit for a gcsr tunable laser for applications in optical packet switching," in *Proc. Networks Optical Communications (NOC)*, Darmstadt, Germany, Jun. 2002, pp. 394–400.

[5] D. Klonidis, C. Politi, M. J. O'Mahony, and D. Simeonidou, "Fast and widely tunable optical packet switching scheme based on tunable laser and dual-pump four-wave mixing," *IEEE Photon. Technol. Lett.*, vol. 16, no. 5, pp. 1412–1414, May 2004.

[6] L. Berger *et al.* (2003, Jan.). RFC 3471—Generalized multi-protocol label switching (GMPLS) signaling functional description. Internet Engineering Task Force (IETF) Proposed Standard. [Online]. Available: http://www.faqs.org/ftp/rfc/rfc3471.txt,ftp://ftp.rfceditor.org/in-notes/rfc3471.txt.

[7] K. Nichols *et al.* (2001, Apr.). RFC 3086—Definition of differentiated services per domain behaviors and rules for their specification. Internet Engineering Task Force (IETF) Proposed Standard. [Online]. Available: http://www.ietf.org/rfc/rfc3086.txt.

[8] A. Banerjee, J. Drake, J. Lang, B. Turner, D. Awduche, L. Berger, K. Kompella, and Y. Rekhter, "Generalized multiprotocol label switching: An overview of signalling enhancements and recovery techniques," *IEEE Commun. Mag.*, vol. 39, no. 7, pp. 144–151, Jul. 2001.

[9] A. Banerjee, J. Drake, J. Lang, B. Turner, K. Kompella, and Y. Rekhter, "Generalized multiprotocol label switching: An overview of routingand management enhancements," *IEEE Commun. Mag.*, vol. 39, no. 1, pp. 144–150, Jan. 2001.

[10] F. L. Faucheur *et al.* (2002, May). RFC 3270—Multi-protocol label switching (MPLS) support of differentiated services. Internet Engineering Task Force (IETF) Proposed Standard. [Online]. Available: http://www.ietf.org/rfc/rfc3270.txt,ftp://ftp.rfc-editor.org/innotes/rfc3270.txt.

[11] E. C. Rosen *et al.* (2001, Jan.) RFC 3031—Multiprotocol label switching architecture. Internet Engineering Task Force (IETF) Proposed-Standard. [Online]. Available: http://www.faqs.org/rfcs/rfc3031.txt,ftp://ftp.rfc-editor.org/innotes/rfc3031.txt.

[12] V. Jacobson *et al.* (2002, Mar.) RFC 3246—An expedited forwarding PHB (per-hop behavior). Internet Engineering Task Force (IETF) Proposed Standard. [Online]. Available: http://www.faqs.org/rfcs/rfc3246.txt,ftp://ftp.rfc-editor.org/innotes/rfc3246.txt.

[13] J. Heinanen *et al.* (1999, Jun.) RFC 2597—Assured forwarding PHB Group. Internet Engineering Task Force (IETF) Proposed Standard. [Online]. Available: http://www.faqs.org/rfcs/rfc2597.txt,ftp://ftp.rfc-editor.org/innotes/rfc2597.txt.

[14] K. Pagiamtzis and A. Sheikholeslami, "Pipelined match-lines and hierarchical search-lines for low-power content-addressable memories," in *IEEE Custom Integrated Circuits Conf. (CICC) Dig. Tech. Papers*, Sep. 2003, pp. 383–386.

[15] C. Clos, "A study of non-blocking switching networks," *Bell Syst. Tech. J.*, vol. 32, no. 2, pp. 406–424, Mar. 1953.

[16] A. Jajszczyk, "Nonblocking, repackable, and rearrangeable clos networks: Fifty years of the theory evolution," *IEEE Commun. Mag.*, vol. 41, no. 10, pp. 28–33, Oct. 2003.

[17] S. McCreary and K. Claffy, "Trends in wide area ip traffic patterns—a view from ames internet exchange," in *ITC Specialist Seminar IP Traffic Modeling*, Monterey, CA, Sep. 2000, pp. 1–11.

[18] T. Koonen, J. Jennen, H. deWaardt, I. T. Monroy, Sulur, and G. Morthier, "An optical-label controlled packet router for IP-over-WDM networks," in *Proc. LEOS Benelux Symp.*, Brussels, Belgium, Dec. 2001, pp. 1–4.

[19] W. Vanderbauwhede, D. Harle, and F. Touvet, "Providing quality of service in an IP over optical packet switch network using GMPLS," presented at the IV Workshop in MPLS/GMPLS Networks, Girona, Spain, Apr. 2005.

[20] D. K. Hunter, D. Cotter, R. B. Ahmad, W. D. Cornwell, T. H. Gilfedder, P. J. Legg, and I. Andonovic, "Buffered switch fabrics for traffic routing, merging and shaping in photonic cell networks," *J. Lightw. Technol.*, vol. 15, no. 1, pp. 86–101, Jan. 1997.

[21] K. J. Warbrick, P. R. Roorda, and D. Pugh, "Performance and scaling of a recirculating optical buffer," in *Proc. London Communications Symp. 2000*, London, U.K., p. S10-2.

[22] Y. N. Singh and M. Naik, "Study of power variation of a buffered packet in optical loop buffer," in *Proc. Photonics*, Mumbai, India, Dec. 2002, p. NET-11.

[23] W. Vanderbauwhede and H. Novella, "A multi-exit recirculating optical packet buffer," *IEEE Photon. Technol. Lett.*, accepted for publication, 2005.

[24] *10th Int. Symp. Asynchronous Circuits and Systems* (ASYNC2004), Hersonissos, Crete, Greece, icsp. Apr. 2004 [Online]. Available: http://www.ics.forth.gr/async2004/.

[25] W. Vanderbauwhede and D. Harle, "Modelling and characterisation of an asynchronous optical packet switch for direct IP over WDM," in *Proc. International Network Optimization Conference (INOC)*, Paris-Evry, France, Oct. 2003, p. C3-2.

[26] R. Nejabati and D. Simeonidou, "Class-based aggregation in optical packet switched WDM networks," in *Proc. Trans-European Research and Education Networking Association (TERENA) Networking Conf.*, Zagreb, Croatia, May 2003, p. 5b2.

[27] T. Ferrari, "End-to-end performance analysis with traffic aggregation," in *Proc. Trans-European Research and Education Networking Association (TERENA) Networking Conf.*, Lisbon, Portugal, May 2000, p. 4b1.

[28] (2003, May). ITU-T Recommendation G.114—*One-Way Transmission Time*. International Telecommunication Union (ITU). [Online]. Available: http://www.itu.int/ITU-T/publications/recs.html.

[29] W. Vanderbauwhede and D. Harle, "Design and modeling of an asynchronous optical packet switch for DiffServ traffic," in *Proc. 8th Conf. Optical Network Design and Modelling*, Gent, Belgium, Feb. 2004, pp. 19–35.

[30] D. K. Hunter, *Optical Packet Switched Networks (OPSnet)*. [Online]. Available: http://gow.epsrc.ac.uk/ViewGrant.ASPx?Grant=GR/R33427/01.

**Wim A. Vanderbauwhede** (M'03) received the Ph.D. degree in Applied Sciences from the University of Ghent, Belgium, in 1996.

He was a Research Associate with the Broadband and Optical Networks Group, University of Strathclyde, Glasgow, U.K., where his research focused on the design and modeling of an asynchronous optical packet switch. He was an IC design and modeling engineer at Alcatel. He is an Advanced Research Fellow (EPSRC) and a Lecturer with the Department of Computing Science, University of Glasgow, U.K. His current research interests are in service-based architectures for systems-on-chip.

**David A. Harle** (M'90) received the Ph.D. degree in integrated telecommunications from the University of Strathclyde, Glasgow, U.K., during the 1990s.

He was a Research Assistant in Integrated Telecommunications with the University of Strathclyde. Currently, he is a Senior Lecturer with the Broadband and Optical Networks Group in the Department of Electronic and Electrical Engineering in the same university. His current research interests within the Broadband and Optical Networks group focus on performance evaluation and design and management issues associated with current and future broadband and optical communication systems. He is the author of more than 50 research papers and undergraduate texts.