Modelling and Characterisation of an Asynchronous Optical Packet Switch for Direct IP over WDM

Wim Vanderbauwhede [1], David Harle[2]
University of Strathclyde
Broadband and Optical Networks group
Institute of Communications and Signal Processing
Department of Electronic and Electrical Engineering
Royal College, George St, G1 1XW Glasgow, UK

[1] Email:w.vanderbauwhede@comms.eee.strath.ac.uk
Phone: ++141 548 2090 Fax: +44 141 552 4968

[2] Email:d.harle@comms.eee.strath.ac.uk
Phone: +44 141 548 2717 Fax: +44 141 552 4968

15th August 2003

Abstract

The prime objective of the EPSRC-funded OPSnet project is the design and demonstration of an asynchronous DWDM optical packet switch (OPS) capable of directly carrying IP packets over DWDM-based core networks at transport rates in the order of 100 Gb/s and above. To achieve such an objective demands a highly flexible and innovative core switch architecture. The operation and performance of such an architecture is the subject of this paper. The paper directly addresses the performance of the core OPS module and results obtained from simulation models show that the proposed asynchronous OPS architecture exhibits low latency and packet losses allied with relatively high throughput.

Keywords: Quality of Service, Simulation, Optical Networks, MPLS

1 Introduction

The ability to transport of IP packets directly over DWDM is attractive as the overhead of intermediate protocols is then eliminated. However, this implies that the IP packets must be switched in the optical domain, i.e. optical packet switching. IP packet streams are by nature asynchronous, with variable packet length and variable inter-arrival time. As a consequence, the optical packet switch must be designed to be able to handle this type of traffic. This is the goal of the OPSnet project: to design, model and demonstrate an asynchronous DWDM optical packet switch running at 40Gb/s and scalable to 100Gb/s and higher [1].

2 Asynchronous Optical Packet Switch Architecture

2.1 Requirements

An optical network with high data transport rates and QoS imposes a number of requirements on the optical packet switching node:

2.1.1 Quality of Service requirements

To be able to fulfill QoS requirements, the OPS must be GMPLS compliant. Generalized Multi-Protocol Label Switching [2, 3] is an extension or generalization of MPLS [4] that allows a label to be a wavelength, frequency, time slot or position in space.

The basic idea behind MPLS is to pre-establish paths along which the data will be forwarded. For an OPS, forwarding of a packet is based on three "labels": the input port of the OPS, the input wavelength, and the packet label. Furthermore, to guarantee a certain QoS level, it must be possible to prioritize the traffic, e.g. utilising the DiffServ classes [5]. The QoS requirements imply high throughput, low latency and low packet loss (although not all apply for every class of traffic). In general, it is desirable to conserve the packet order, because packet reordering increases latency at the destination.

2.1.2 Scalability

The OPS node must be suitable for DWDM and scalable. In addition to simple space switching, the node must be able to distinguish between different wavelengths and be able to switch datastreams from one wavelength to another. The number of wavelengths should not be limited by the design, although it may be limited by the state of the art for the technology. Such a scalability requirement has a major impact on the architecture and cannot be over emphasized.

2.1.3 High Data Transport Rates

The OPS node design must allow operation at high bitrates (40 Gb/s per datachannel, scalable to 100 Gb/s and higher) under high network load. As the data remains in the optical domain, the key issues lie with header processing which is required to be extremely fast.

2.1.4 Contention Resolution

The node must be able to handle packets of variable length, with variable inter-arrival times and asynchronous arrivals. To minimize packet losses, there must be contention resolution. This implies the need for optical buffering.

2.2 Design Solutions

The main issues for the system-level design are scalability, fast header processing and contention resolution.

- Scalability requires a modular architecture.

  The OPSnet architecture uses passive wavelength (de)multiplexers to separate the wavelength channels, wavelength translators and three single-wavelength OPS stages. The three single-wavelength stages are necessary to ensure the switch is strictly non-blocking [6] and this is a requirement for backward compatibility with circuit switched networks). This approach is very scalable because every OPS never has a large number of ports. Additionally, it means the OPS design itself is simplified because every OPS is essentially single-wavelength (although the employed technology, AWG+wavelength translation, uses multiple wavelengths for the actual switching [7]).

- Fast header processing with asynchronous logic.

  Fast header processing is obtained by using an asynchronous electronic circuit with content-addressable memory as lookup table. Asynchronous logic does not depend on an internal clock, but is event-driven and results very fast header processing. The lookup table is written by the management layer, and the writing is completely decoupled from the reading. This means the implementation of the management layer is independent of the OPS control layer implementation.

- Contention resolution via optical buffering.

  The OPS buffering strategy applies traditional statistical multiplexing but uses an innovative buffer architecture based around an in-line parallel per-packet recirculating buffer [**?**]. Here, the buffers circulate in parallel in order to maximise reinsertion probabilities. Such an architecture essentially replicates the action of electronic random-access memory using optical technology and offers low latency and high throughput.

3 Performance Modelling

3.1 Switch Modelling

The complete OPS system-level design (optical part and electronic control) was implemented in the Verilog hardware description language (HDL) using an object-oriented code generator [9]. The OPS control diagram is represented in Fig. 1. To model the performance of the OPS, the HDL description was ported to a high-level C++ model. The model is implemented as an asynchronous discrete-event simulator. The packets are modeled as length/label pairs, the buffers are modeled as 2-D arrays, the length reflecting the buffer depth. The parallel arrival of traffic streams at the input ports is simulated by continuously looping over all ports whilst keeping track of the arrival times of the packets. The signaling flows from the HDL model are simulated by passing variables between the different modules.

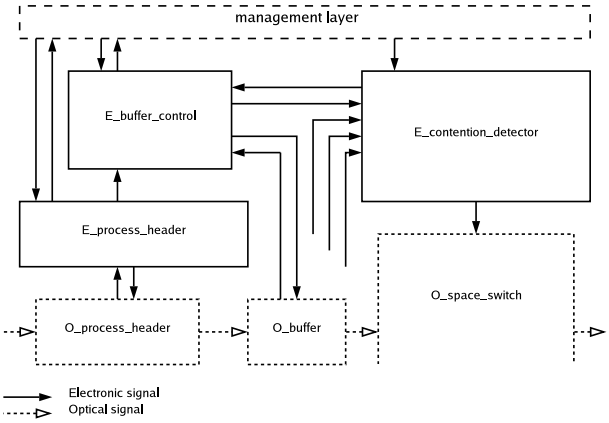

Figure 1: OPSnet OPS control diagram

3.2 Traffic Modelling

Switch performance is determined by running the switch model supporting a range of different traffic distributions and loads. The traffic models employed are an integral part of the switch simulation environment and are used to determine a range of system performance metrics, e.g. packet loss & latencies, buffer occupancies and traffic shaping characteristics. Thus the effect of different traffic types and loads can be investigated as a function of buffering strategies (type and depth), packet contention strategies (dropping and deflection)and packet ordering regimes.

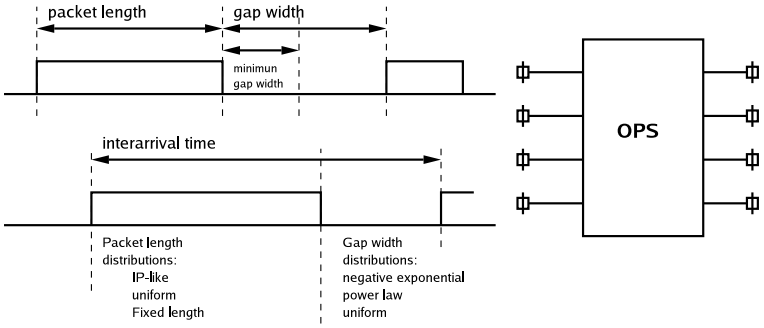

Figure 2: Traffic distribution model

The traffic models are based upon a 2-state model as shown in Fig. 2. Packet length distributions are taken from uniform, fixed and IP-like distributions [9] while the gaps between packets follow one of uniform, negative exponential (Poisson) and power-law (Pareto) distributions. To ensure fairness when comparing overall performance, all three packet gap distributions have common minimum and mean gap intervals.

The OPSNet OPS architecture has a very strong traffic shaping effect and thus it is necessary to model the steady-state core traffic distribution. This steady state was achieved by queuing the switched packets in lines with a length equal to the average link length, changing their destination labels and switching the streams back to the OPS using a random multiplexer (Fig. 3).

Such an approach is equivalent to connecting a number of OPS within a network topology where the average load per node is uniform over the network. By maintaining a high overall network/node load, such a configuration can be used to evaluate the performance of an OPS-based transport core network. The overall network performance can then be characterised by the aggregated losses and latencies of individual switches.

3.3 Simulation Automation

Performance modelling generally requires a large number of simulations to cover the complete parameter space. For this reason, a generic simulation automation tool has been developed in the frame of the OPSnet project. This tool makes it possible to run and process thousands of simulation, allowing for a much more in-depth characterisation of the model performance. It is written in Perl and available from the Comprehensive Perl Archive Network [10].

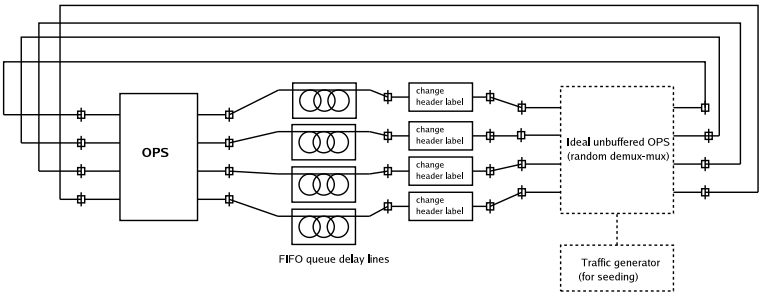

Figure 3: Core traffic simulation strategy

4 Results

The performance of the OPS network under a range of traffic conditions and switch configurations are now presented.

4.1 Buffer Depth

The first result shows the maximum sustainable load that can be achieved with a packet loss of less than 1 in 10^6as a function of the buffer depth. The packet length distribution is IP-like, the gap width distribution is either Pareto-distributed or obtained via the core traffic simulation method. The results also depend strongly on the type of recirculating buffer, in this case a multi-exit buffer with 16 exits. From Fig. 4a we can see that the buffer depth is moderate even for very high loads, and small for moderate loads. We also note that the performance under steady-state core traffic is much better than under Pareto traffic, which means that the traffic shaping done by the OPS improves the network performance.

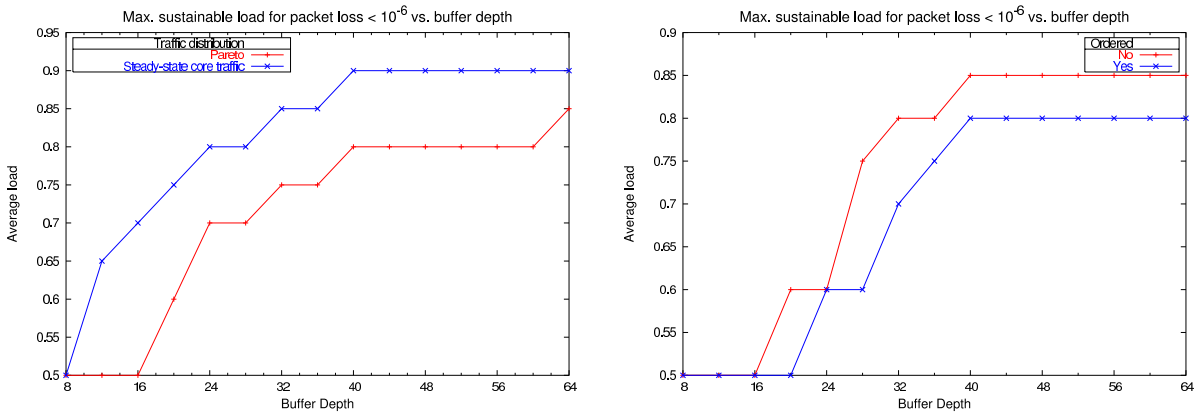

Figure 4: Influence of (a) traffic distribution and (b) conservation of packet order on required buffer depth

Another aspect is the impact of conserving the packet order. Fig. 4b shows that for moderate load, the effect is small. However, conserving the packet order reduces the maximum load, even for large buffer depths.

4.2 Buffer Type

As mentioned previously, the type of buffer has a strong impact on the buffer depth requirements. Fig. 5 compares the packet loss versus network load for two buffer types, the fixed buffer (essentially a simple circular delay line) and the multi-exit buffer, a new concept developed for the OPSnet switch. For both cases the impact of conserving the packet order is also shown. The buffer depth for this experiment is fixed at 32 buffers.

4.3 Buffer Occupancy

For a better understanding of the buffer depth requirements, a histogram of the number of occupied buffers (the ”fill state”) is very instructive. The histogram is constructed by monitoring the number of occupied buffers for every port, and store the values

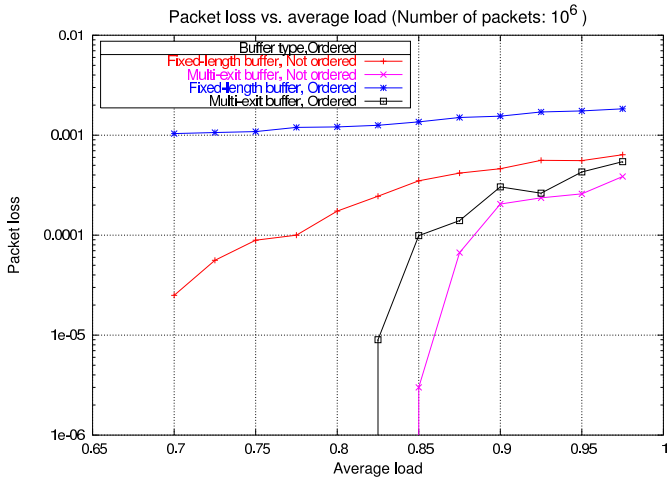

Figure 5: Influence of buffer type and packet order conservation on packet loss for variable load

every time a packet enters or leaves the buffer. A histogram of the number of occurrences of every state, i.e. the number of times a given occupancy occurred, is represented in Fig.6. The count is on a logarithmic scale, and the results are for core traffic with and without conservation of the packet order. The average occupancy is very low, for most of the time only one or two buffers are occupied. However, the buffer must be designed for the maximum occupancy to avoid buffer overflow. From Fig 6, the impact of the conservation of the packet order is very clear: the occupancy distribution acquires a much longer tail, which of course confirms the buffer depth requirements from 4.1.

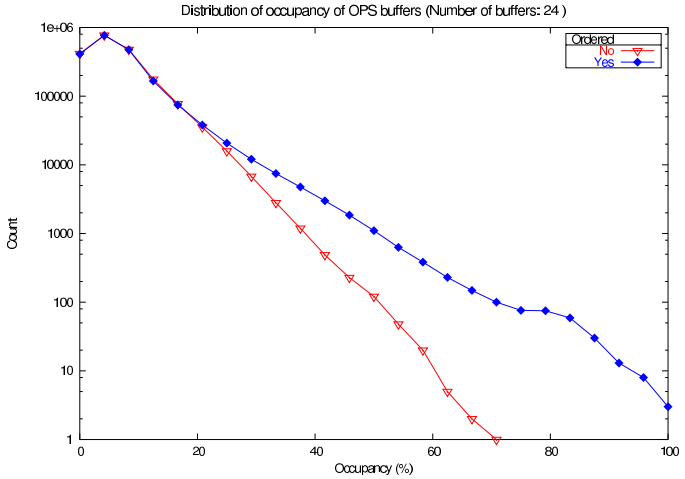

Figure 6: Influence of conservation of packet order on the buffer occupancy

4.4 Latency

An important performance indicator for an OPS is the latency introduced by the buffering process. The latency was simulated by monitoring the sojourn time of every packet in its buffer. Results for core traffic with and without conservation of the packet order are shown in Fig. 7.The maximum packet length was 1500 bytes (as typical for IP over Ethernet). At 100Gb/s, this corresponds to 120 ns, and this is the length of the recirculating buffer. The network load was 0.7, the buffer depth 24 and the buffer a multi-exit buffer with 8 exits. It is clear that the average latency is very small: 98.5% of all packets has a latency of less than 100ns, i.e. less than the maximum packet length (which is possible because the multi-exit buffer allows packets to leave before they have made a full loop). Even when the packet order is conserved, less than 1 packet per million has a latency of more than $5\mu s$ (at 100Gb/s).

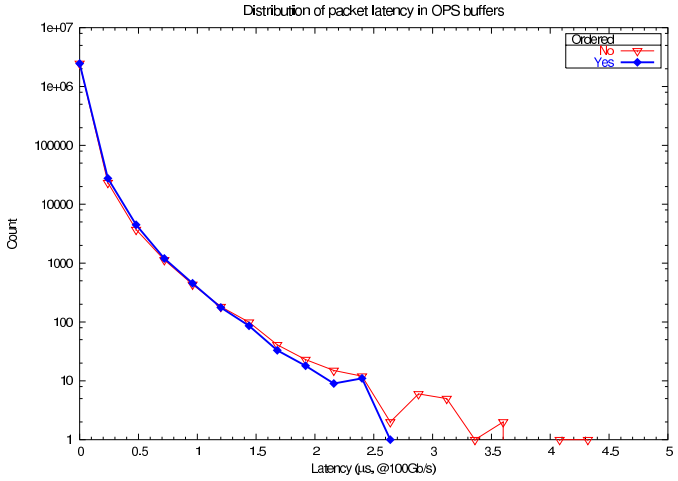

Figure 7: Packet latency (in multiples of the maximum packet length).

5 Conclusions

This paper represents a performance evaluation of the proposed OPSNet asynchronous (IP-like) Optical Packet Switching (OPS) architecture. The architecture is based upon a novel all-optical buffering scheme and supports QoS services, is DWDM-capable and fully scalable over a range of transmission rates. The architecture is flexible and allows packet order and priorities to be applied within the network if required. Results obtained from simulation models show, for a variety of different traffic loads and types, that the proposed design offers excellent throughput and latency characteristics. Furthermore, the novel parallel recirculating buffers at the core of the switch architecture requires only relatively small buffer depths to preserve the required latency and loss targets associated with a variety of different packet streams.

References

[1] W. Vanderbauwhede, D. Harle, ”Novel design for an asynchronous optical packet switch”, Proc. ONDM-2003, Feb 2003, pp737-754

[2] A. Banerjee, J. Drake, J. Lang, B.Turner, D. Awduche, L. Berger, K. Kompella, Y. Rekhter, “Generalized Multiprotocol Label Switching: An Overview of Signalling Enhancements and Recovery Techniques”, IEEE Comms. Mag., Jul 2001, pp144-151

[3] A. Banerjee, J. Drake, J. Lang, B.Turner, K. Kompella, Y. Rekhter, “Generalized Multiprotocol Label Switching: An Overview of Routing and Management Enhancements”, IEEE Comms. Mag., Jan 2001, p144-150

[4] F. Le Faucheur et al., “MPLS Support of Differentiated Services”, RFC 3270, May 2002, http://www.ietf.org/rfc/rfc3270.txt

[5] K. Nichols et al., “Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification”, RFC 3086, Apr 2001, http://www.ietf.org/rfc/rfc3086.txt

[6] M. Collier, T. Curran, "The strictly nonblocking condition in three-stage networks," Proc. ITC-14, 1994.

[7] 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, I. H. White: “WASPNET - a Wavelength Switched Packet Network”, IEEE Communications Magazine, March 1999, pp120-129

[8] K. J. Warbrick, P. R. Roorda, D. Pugh, “Performance and Scaling of a Recirculating Optical Buffer”, LCS 2000

[9] W. Vanderbauwhede, ”Object-oriented Verilog Code Generator”, http://search.cpan.org/author/WVDB/Verilog-CodeGen-0.9.2/

[10] K. Claffy, G. Miller, K. Thompson, “The nature of the beast: recent traffic measurements from an Internet backbone”,23 April 1998, INET 1998, http://www.caida.org/outreach/papers/1998/Inet98

[11] W. Vanderbauwhede, ”Generic Template-driven Simulation Automation Tool ”, http://search.cpan.org/author/WVDB/Simulation-Automate-0.9.4/