# Performance modeling of an asynchronous Optical Packet Switch for direct IP over WDM

W. Vanderbauwhede, D. Harle

8th September 2004

Broadband and Optical Networks group
Institute of Communications and Signal Processing
Department of Electronic and Electrical Engineering
University of Strathclyde, Glasgow

The EPSRC project OPSnet addresses the design, modeling and implementation of an asynchronous optical packet switch (OPS) suited for 100Gb/s DWDM operation. This novel switch architecture allows direct IP transport over DWDM-based core networks. The paper reports on the system-level design and modeling of the core OPS module, with a focus on performance modeling aspects like simulation automation and results processing. The modeling results show that high throughput and low latency and very low packet loss can be achieved with this type of OPS.

**Topics:**

- Switch Architectures for Heterogeneous Networks
- Other Performance Modelling Applications to Computer and Parallel Systems, Distributed Systems, Transportation Networks and Production Systems

## 1 Introduction

Direct IP over WDM allows IP packets to remain in the optical layer throughout transmission between source and destination, and eliminates the overhead of intermediate protocols. To make this possible, the IP packets must be switched in the optical domain, i.e. optical packet switching. As IP packets have variable length and variable inter-arrival times, and the arrivals are not synchronised, there is a need for an asynchronous optical packet switch. The EPSRC project OPSnet addresses the design, performance modeling and implementation of an asynchronous optical packet switch suited for 100Gb/s DWDM operation [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:

- Quality of Service requirements

  To be able to fulfill QoS requirements, the OPS must be GMPLS compliant. Generalized Multi-Protocol Label Switching [3, 2] 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. according to 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.

- 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.

- 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 main issues are with the header processing, which must be very fast.

- 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 (three stages are necessary to ensure the switch is strictly non-blocking [7]. 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 [6]).

- 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. This 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 buffering strategy is statistical multiplexing, but the main innovation in the OPSnet project is the buffer architecture, which is an in-line parallel per-packet recirculating buffer. This means all packets are buffered by default, and every packet has its own buffer. All buffers circulate in parallel, thus maximising the reinsertion probability. The basic idea was to come as close to an electronic random-access memory as possible with optical technology. This buffer architecture has low latency and high throughput.

# 3 Performance modeling

## 3.1 Switch modeling

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 [10]. 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.

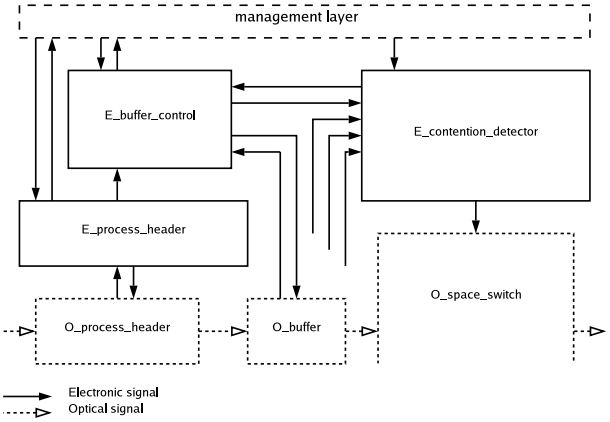


Figure 1: OPSnet OPS control diagram

## 3.2 Traffic modeling

The performance of the switch is evaluated by generating traffic with different distributions and load, and simulating the packet loss, latency, buffer fill state and traffic shaping effects as a function of a number of parameters like the type of buffer, the buffer depth, the use of deflection routing, whether the switch preserves the packet order or not etc.

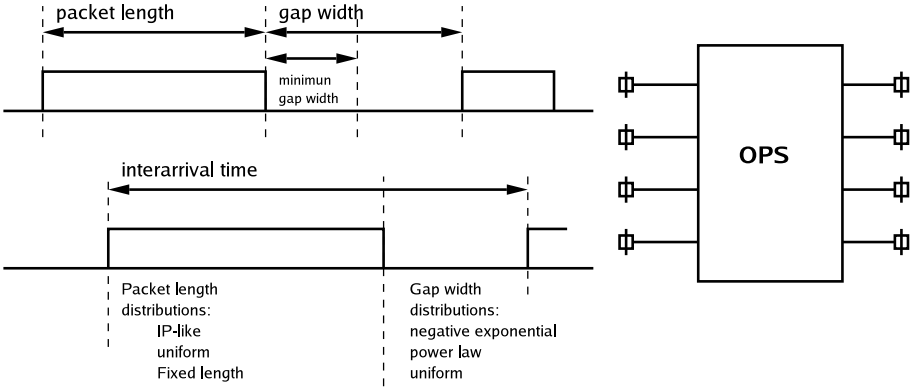


Figure 2: Traffic distribution model

The traffic was modeled using a 2-state model represented in Fig. 2. The packet length distribution can be uniform, fixed length or IP-like [9]. The gaps between the packets are modeled using three types of distributions: uniform, negative exponential (“Poisson”) and power law (“Pareto”). All three distributions have the same minimum and mean gap width.

Because the OPSnet OPS architecture has a very strong traffic-shaping effect, it is important to model the steady-state core traffic distribution. This was simulated 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)

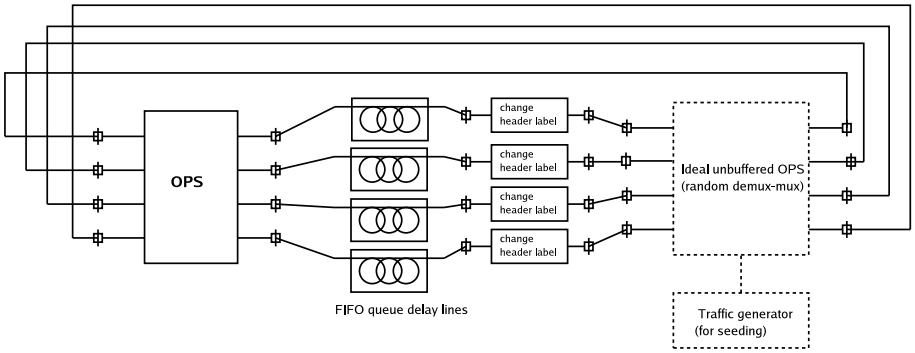


Figure 3: Core traffic simulation strategy

This approach is equivalent to connecting a number of OPS in a network topology, assuming that the average load is the same in all branches. By keeping the overall load high, this method can be used to evaluate the performance for a core network, because the traffic distribution is determined by the switches, and the performance of the network will be determined by the packet loss and latency caused by the switches.

### 3.3 Simulation automation

Performance modeling 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 [11].

## 4 Results

The performance of the OPSnet design has been evaluated for a large range of conditions. To illustrate the capability of the simulation tool and the performance of the OPS, we present some results.

### 4.1 Buffer depth

The first result shows the the required buffer depth for a packet loss of less than 1 in $10^6$as a function of the network load. 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. 4 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 core traffic is much better than under Pareto traffic, which means that the traffic shaping done by the OPS improves the network performance.

Another aspect is the impact of conserving the packet order. Fig. 5 shows that for moderate load, the effect is small, but for high loads, the buffer depth increases with a factor of 2 for the same loss.

### 4.2 Buffer type

As mentioned previously, the type of buffer has a strong impact on the buffer depth requirements. Fig. 6 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.

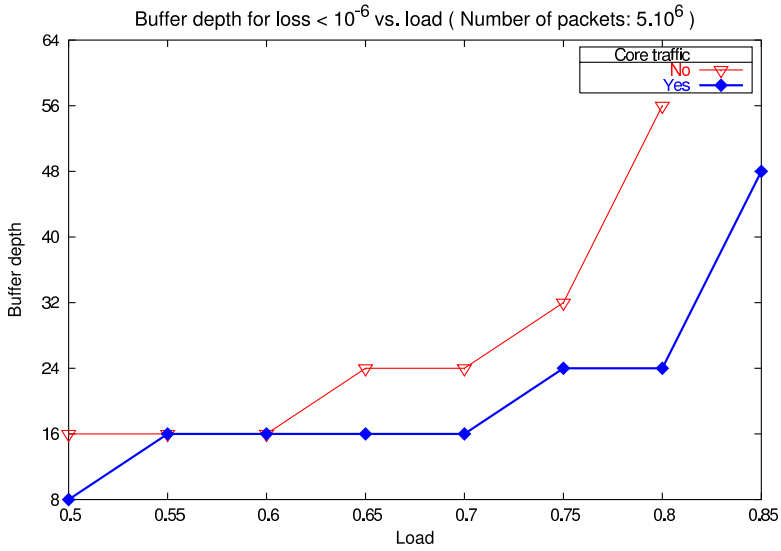


Figure 4: Influence of traffic distribution on required buffer depth for packet loss <1ppm

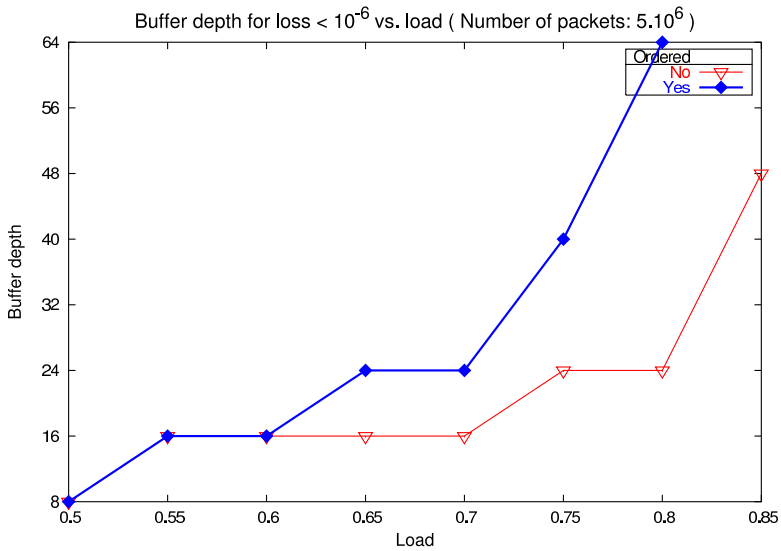


Figure 5: Impact of conserving the packet order on required buffer depth for packet loss <1ppm

### 4.3 Buffer fill state

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 every time a packet enters or leaves the buffer. A histogram of the number of occurrences of every state is represented in Fig.7. The count is on a logarithmic scale, and the results are for core traffic with and without conservation of the packet order. The average fill state is very low, for most of the time only one or two buffers are occupied. However, the buffer must be designed for the maximum fill state to avoid buffer overflow. From Fig 7, the impact of the conservation of the packet order is very clear: the fill state distribution acquires a much longer tail, which of course confirms the buffer depth requirements from 4.1.

### 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

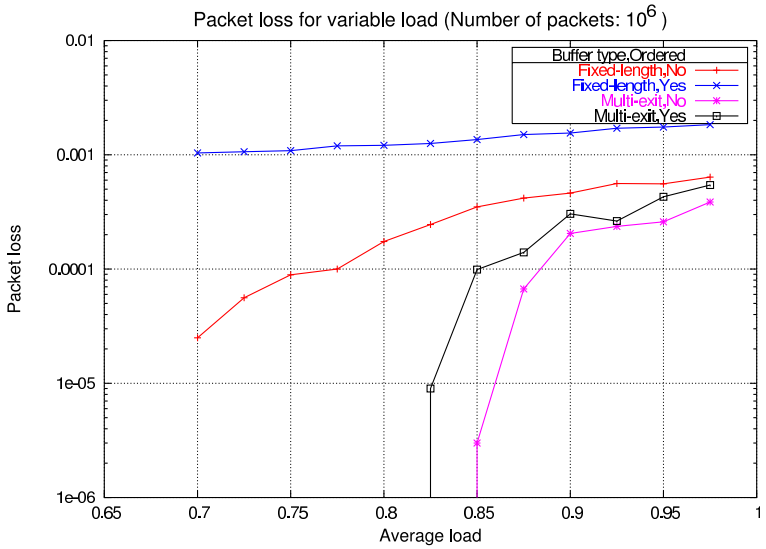


Figure 6: Influence of buffer type and packet order conservation on packet loss for variable load

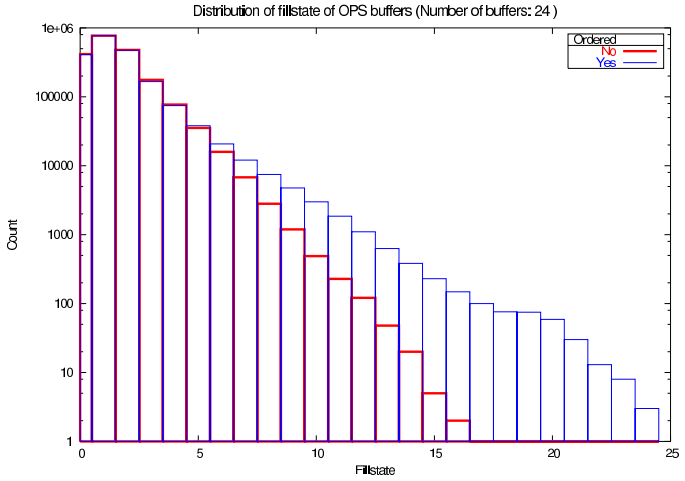


Figure 7: Influence of conservation of packet order on the buffer fill state

without conservation of the packet order are shown in Fig. 8.

The unit is maximum packet length, which was 1500 bytes (as typical for IP over Ethernet). At 100Gb/s, this corresponds to 120 ns. 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 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 40 ( $5\mu s$ at 100Gb/s).

## 5 Conclusion

In this paper, performance modeling of a new design for an asynchronous optical packet switch with a novel optical buffer architecture has been presented. The architecture has support for QoS,is DWDM-capable and fully scalable. The OPS can be configured to conserve the packet order and prioritize the traffic. The OPS has been modeled using a custom discrete-event simulator which has the capability of simulating the core traffic distributions which result from the traffic shaping properties of the switch. A simulation automation tool has been developed to facilitate in-depth characterisation of this architecture. The modeling results show that the

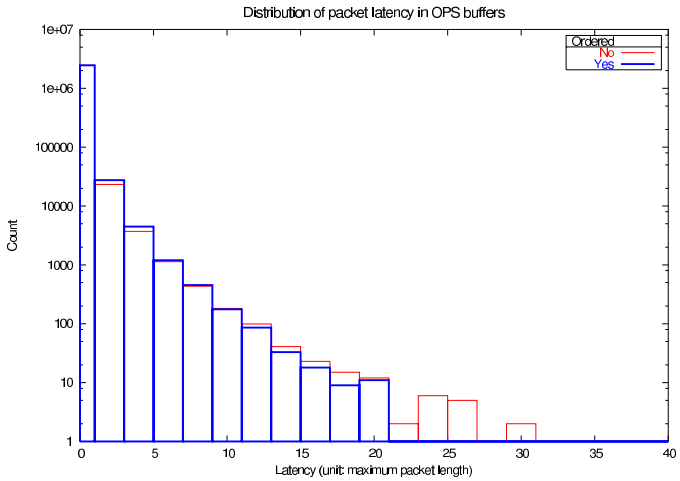


Figure 8: Packet latency (in multiples of the maximum packet length).

design has excellent throughput and latency even under high network load, while requiring only moderate buffer depths.

## 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] 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

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

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

[9] 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

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

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