Multi-stage optical buffered switch for IP traffic

D. K. Hunter[1], I. Andonovic, M. C. Chia

University of Strathclyde, EEE Department, 204 George Street, Glasgow G1 1XW, UK

ABSTRACT

A novel architecture is proposed for future multi-terabit IP (internet protocol) routers, employing multiple cascaded stages of optical switching and buffering. WDM is used within the node to facilitate its operation. External synchronization is not required, and a void-filling algorithm is used to simplify hardware requirements. Packet priorities are not implemented in the current version of the switch, and the issue of header table lookup is not considered. Performance with respect to packet loss is studied by simulation, demonstrating that this multi-stage concept results in substantial hardware reduction.

Keywords: optical packet switching, internet protocol, optical switching, optical communications, asynchronous switching, void filling, multistage architectures, simulation

1. INTRODUCTION

Optical packet switching has been extensively discussed in the literature recently, with the intention of overcoming the anticipated problems inherent in constructing future very large electronic switch cores. Although many architectures for optical packet switches have been proposed in the last few years [1], these have almost exclusively concentrated on fixed-length packets.

This paper proposes an optical packet switched node architecture, which is suitable for use in IP (internet protocol) routers, since it switches and buffers variable-length packets optically. IP header address lookup will not be considered here, but although very high-speed IP address lookup has been demonstrated [2], it may be necessary to consider a simplified addressing scheme.

Although switching elements of any size can in principle be constructed, those with 16 inputs and outputs are studied, which could be implemented optically and could form larger nodes with say 128 or 256 inputs and outputs, by means of a Clos architecture.

2. PRINCIPLES OF NODE DESIGN

High hardware complexity is a potential difficulty when implementing optical IP routers. Three measures are implemented to combat this:

- Asynchronous operation is permitted on router inputs, so it is not necessary to synchronize packets to timeslot or byte boundaries, prior to entering the switch. Synchronizers represent a considerable extra hardware burden and are expensive [3]. Here, it is assumed that packet lengths are multiples of one byte, while packet inter-arrival times are continuously distributed.

- If the lowest increment of delay permitted is less than one byte, packets may be directed in a FIFO manner to the appropriate outputs. However, the hardware complexity inherent in this approach may be avoided by using a technique,

[1] Correspondence: Email: d.hunter@eee.strath.ac.uk; WWW: http://voyager.eee.strath.ac.uk/~dhunter; Telephone: +44 141 548 2527; Fax: +44 141 553 1955

known as void filling, proposed by Tancevski [4]. Here, the delay-line increment may be much larger than the minimum packet length; as a result, unused gaps or "voids" are created between adjacent packets on any given output (Figure 1). Rather than scheduling a new packet under consideration for a particular output to be transmitted immediately after the last one, it may be scheduled to "fill" a void on the correct output in a non-FIFO manner. If necessary, it is still possible to maintain FIFO operation for each input-output pair by stipulating that a packet cannot be scheduled for transmission before the previous packet between that particular input and output.

- By use of multiple buffer stages in cascade, the high hardware costs inherent in implementing only a single stage may be avoided. This approach has already been proposed for fixed-length optical packets in the Switch with Large Optical Buffers [5].

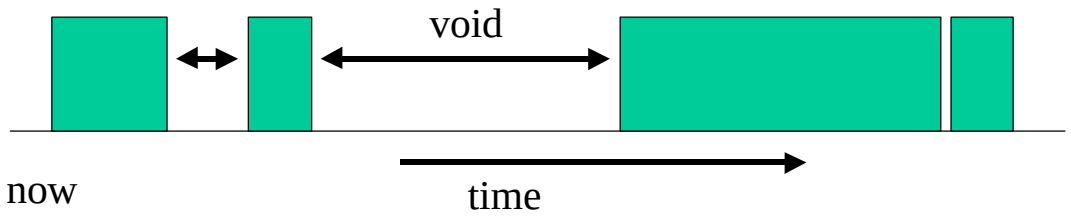

Figure 1: Formation of voids in the packet stream scheduled to pass from a given output. The void may be filled by scheduling another packet to be transmitted during it.

3. NODE ARCHITECTURE

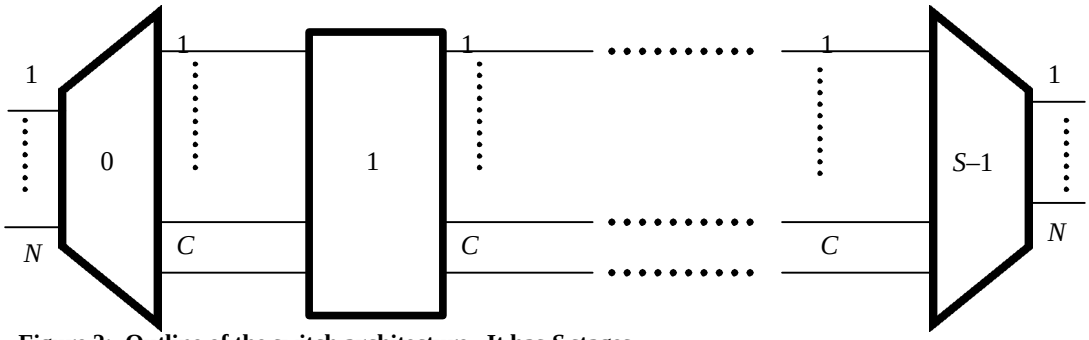

Figure 2: Outline of the switch architecture. It has *S* stages.

The architecture is composed of a series of *S* stages, numbered 0 to *S*-1 (Figure 2). Each stage has the following number of inputs and outputs:

- Stage 0 (first stage): *N* inputs and *C* outputs.
- Stage *S*-1 (final stage): *C* inputs and *N* outputs.
- All other stages: *C* inputs and *C* outputs.

Throughout, $C \geq N$. Each stage can subject each packet to a delay in $\{0, \delta N^{S-1-i}, 2\delta N^{S-1-i}, ..., (D-1)\delta N^{S-1-i}\}$, with δ being the smallest unit of delay, known as the delay granularity. i is the stage number, with $i = 0$ representing the leftmost stage, N is the number of inputs and outputs, and D is the number of possible delays in each stage which may vary between architectures, including delays of length zero.

These delay-line values are not the only ones that may be tried, and others may be used, but for convenience, these values have been used in the simulations here.

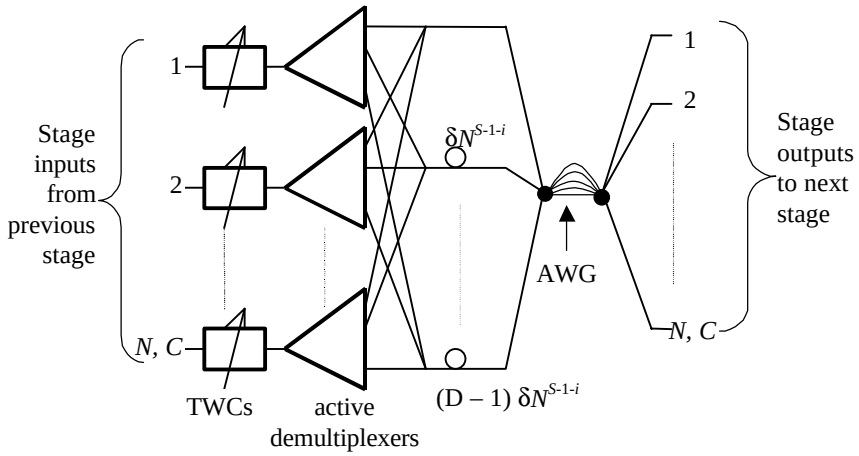

Figure 3: One stage of the architecture. There are either *N* or *C* inputs or outputs, depending on the stage number.

The fundamental function of a stage is to send each incoming packet to any free stage output, having passed through one of the *D* delay-lines. While there are undoubtedly many ways of implementing this, one approach involves using the OASIS switch proposed under the RACE ATMOS program (Figure 3 [6]). The delay lines within each stage are of length 0, δN^{S-1-i}, $2\delta N^{S-1-i}$,…,$(D-1)\,\delta N^{S-1-i}$, where i is the stage number (= 0, 1,…,$S-1$). Each stage can delay each incoming packet by any of these delays, and direct it to any of its outputs, subject to the constraint that no more than one packet can be directed to a particular output at once. There is a constraint C on the number of packets that can pass through a stage simultaneously, which is equal to the number of links between each pair of consecutive OASIS switches. There are no delay-lines between stages since they are provided as part of the OASIS architecture.

4. CONTROL ALGORITHM

The router controller implements a modified form of the original void-filling algorithm, making a sequential search of the available routes. Tancevski proposed void filling originally [4]. Packet priorities have not been examined, although this feature could be added.

The algorithm is recursive; each time the algorithm is called to route a packet from a certain stage to the output, it calls itself to route the same packet from the next stage to the output, unless the stage is the final stage. The control algorithm maintains a list of the packets that are scheduled to pass through each point in the architecture in the future. The algorithms work by determining if a packet can take a particular path through the architecture, depending on whether there is contention with one or more packets that have been already scheduled, with the timing dictated by the delays in that path. If the path is free, then these lists are amended to reflect the allocation the packet to the route through the architecture. Thus if there are any gaps (or "voids") between future scheduled packets, the packet being scheduled may be transmitted during this void, hence the name "void filling".

The algorithm follows, routing a packet from a particular stage in the architecture:

```
IF stage is final stage
    /* try to route it to the correct output. */
    FOR index = 0 TO D – 1
        IF packet can be accommodated on required output d_stage, index seconds later
            reserve space and return successfully
        ENDIF
    NEXT index
    return unsuccessfully /* since the required output cannot be accessed on any delay */
```

```
ELSE /* i.e. stage is not final stage */
        /* try to route it to any link in the next stage in OASIS architecture */
        FOR index = 0 TO D – 1
                avail = 0
                /* try to find free link for this delay ... */
                FOR linknum = 0 TO C – 1
                        IF link linknum to next stage free after delay d_stage, index
                                avail = 1
                                GOTO X
                        ENDIF
                NEXT linknum
                /* the following line calls this function recursively */
X:              IF avail = 1 and packet can be routed from link linknum to output
                        reserve space and return successfully
                ENDIF
        NEXT index
        return unsuccessfully
ENDIF
```

When the simulator itself calls the algorithm, routing takes place from the first stage of the architecture. A VLSI implementation of this algorithm is anticipated.

5. SIMULATION METHODOLOGY

5.1 Traffic generation

Traffic with negative exponentially distributed inter-packet gaps and geometrically distributed packets is studied. Throughout, all packet lengths are measured in units of bytes, and U is a uniformly distributed random variable between 0 and 1.

The negative exponential inter-packet gaps are generated by:

$$G = -g \log U$$

where g is the mean gap length, and the geometrically distributed packet lengths are generated by:

$$P = \left\lfloor 1 + \frac{\log U}{\log(1 - 1/p)} \right\rfloor$$

where p is the mean packet length. A high mean packet length was chosen for the simulations, so that the distribution approximates to a truncated negative exponential. This is because $1 - 1/p \approx e^{-1/p}$ for large p. The packets have a minimum length of one byte.

5.2 Generation of results

A simulator was written in C, and was exhaustively tested against existing results in the literature [4,7,8,9,10].

Each simulation took place for 3.2×10^7 packets, to determine packet loss above 10^{-6}; for the purposes of this paper, a packet loss less than this is considered to be acceptable. Delay was also determined from the simulations. A load of 0.8 was chosen, and the other options chosen were:

1. N (no. inputs/outputs) = 16

2. *S* (no. stages) = 1, 2 or 3

3. *D* (no. fiber delay lines per stage) = 16, 20, 24, … , or 64 for 1 stage
   D = 4, 5, 6, …, 16 for 2 stages.
   D = 3, 4, 5, …, 16 for 3 stages.

4. *C* (packet constraint for each stage) = 16, 20, 24, 28 or 32 for 2 stages
   C = 16 for 3 stages

5. *p* (mean packet length) = 400 bytes

6. δ (fundamental delay line unit) = 200, 400, 1000, 1750, 2500, 3250 or 4000 bytes.

6. RESULTS AND DISCUSSION

6.1 Single-stage architectures

For comparison purposes, Figure 4 shows the packet loss probability for a single-stage architecture. The delay-line granularity has a profound effect on the packet loss, which exhibits a marked improvement as the granularity is increased up to 2500 bytes, although there is no appreciable improvement thereafter. As the granularity is increased, the amount of buffer storage available increases proportionately, for the same number of delay-lines. Void filling permits more efficient use to be made of this storage capacity, even although every storage location within it is not immediately addressable.

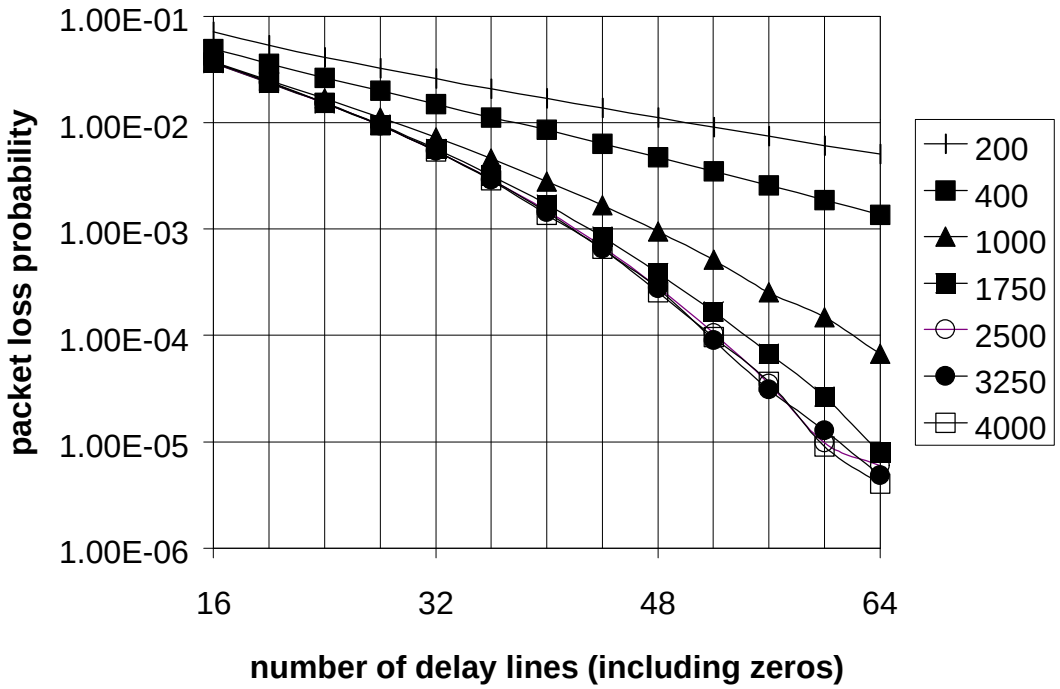

Figure 4: Packet loss probability for various delay-line granularities (shown in the box on the right), with 16 inputs and outputs, a mean packet length of 400, and a traffic intensity of 0.8.

For the traffic being simulated here, a delay-line granularity of 2500 bytes would be selected. However, this only applies to this traffic, and different granularities would suit other traffic types. A worthwhile goal would be to produce a switch that

would operate well for all traffic statistics, regardless of packet length, burstiness or intensity. By permitting a much larger selection of feasible delays, it is conceivable that multi-stage architectures could carry out such a task.

In order to obtain a packet loss below 10^{-5}, over 60 delay-lines are required which is hardware intensive. It will now be shown that this problem can be overcome by using a multi-stage architecture. (Due to the statistics of the simulation, some of the lines cross over for low values of packet loss probability. 90% confidence intervals were calculated for all data points using a Student's t-test, and it was found, for example, that the result was $4.03\times10^{-6} \pm 1.57\times10^{-6}$ for a granularity of 4000 bytes with 64 delay lines. This result is typical for low values of packet loss, indicating that the crossing of the lines is not significant.)

6.2 Two-stage architectures

Figure 5 shows the packet loss versus number of delay-lines for a router with C = 16 links between stages, while Figure 6 shows the same for C = 32. (The spurious point for D = 11 and δ = 400 is not statistically significant). An absent data point indicates a packet loss of less than 10^{-6}. Again, there is a profound difference in performance depending on the delay-line granularity, although above a granularity δ of 1750 there is no substantial improvement. For C = 32, feasible packet losses of less than 10^{-6} can be obtained with only 7 delay lines per stage i.e. a total of only 14 delay lines compared to over 64 that would be required with a single stage. The issue here is not so much the number of delay-lines as the amount of supporting hardware such as switches to direct each packet to the correct delay-line.

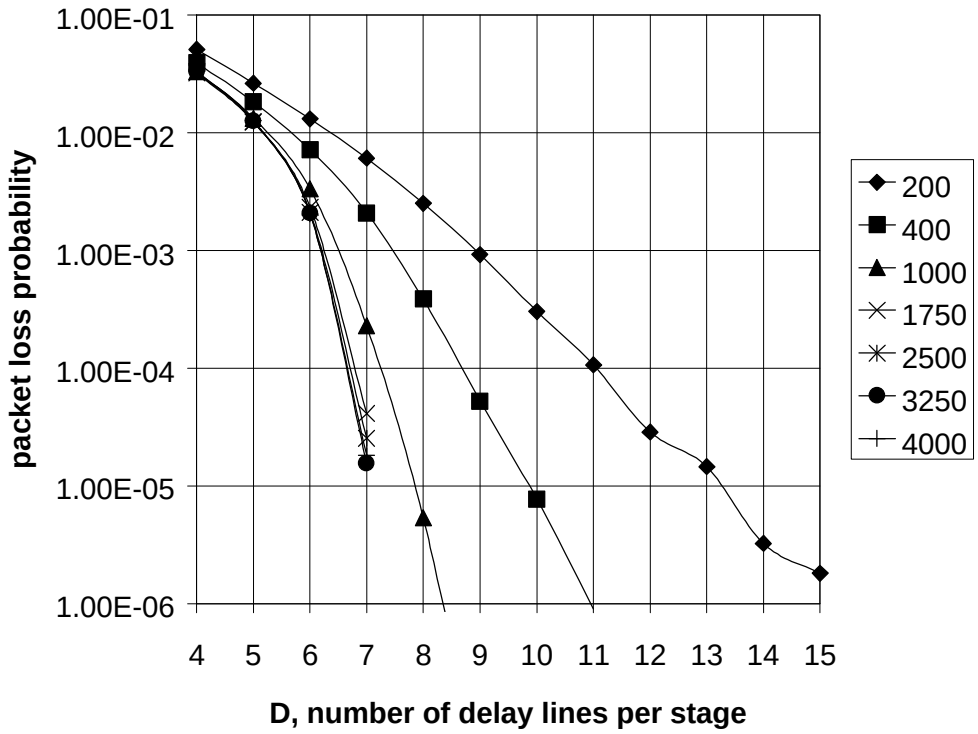

Figure 5: Simulation results for a 2-stage architecture with C = 16 interconnections between stages.

From Figure 5 and Figure 6, there is a reduction in buffer depth requirements for C = 32; 7 delay-lines per stage are required for acceptable packet loss compared to 8. This is explored in more detail in Figure 7, where D is fixed to 6. As C is increased beyond 20, there is no improvement in packet loss performance. Clearly there is a trade-off between C and D for a given level of packet loss performance. Whether it is worthwhile to increase C to reduce D or vice-versa remains to be determined.

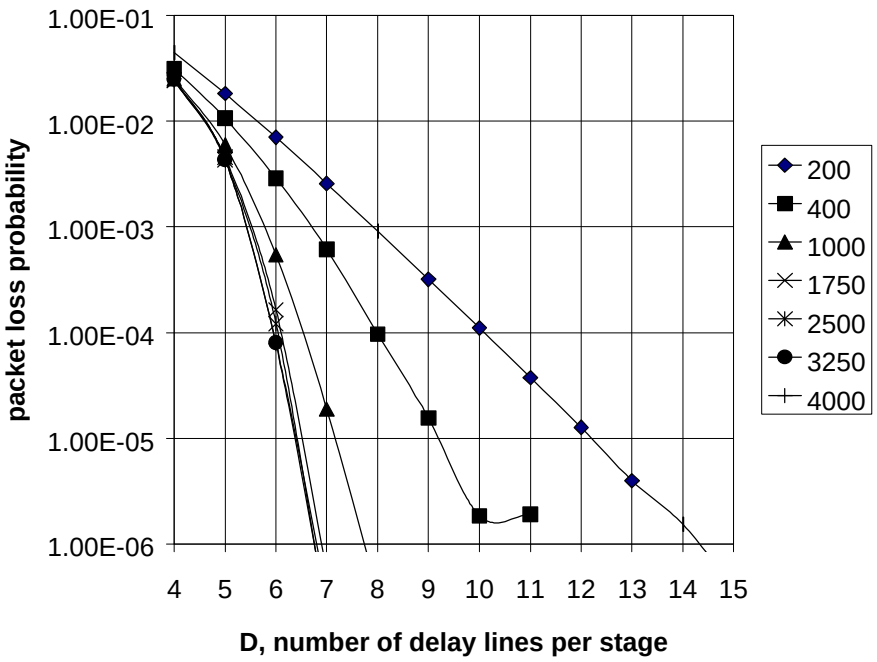

Figure 6: Simulation results for a 2-stage architecture with C = 32 interconnections between stages.

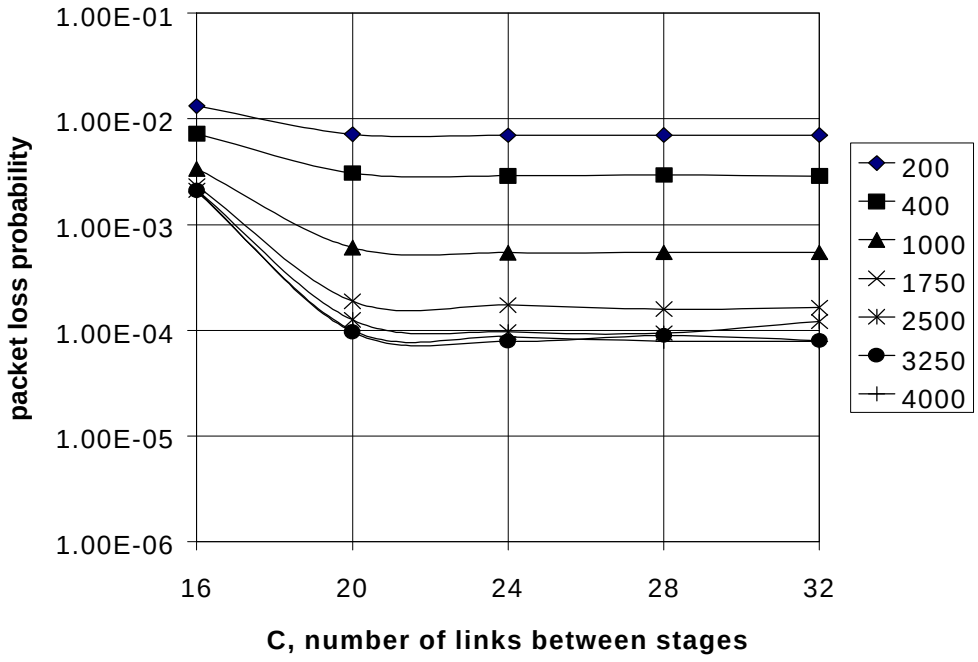

Figure 7: Simulation results for a 2-stage architecture with *D* = 6 delay-lines within each stage.

6.3 Three-stage architectures

Due to the time taken to compute the results, fewer data points were collected for three-stage architectures than for those with one or two stages. The results are presented in Table 1.

	$D = 3$	$D = 4$	$D = 5$
δ = 200 bytes	1.843×10^{-2}	8.031×10^{-6}	$<10^{-6}$
δ = 400bytes	1.458×10^{-2}	$<10^{-6}$	$<10^{-6}$
δ = 1000 bytes	1.205×10^{-2}	$<10^{-6}$	$<10^{-6}$
δ = 1750 bytes	1.134×10^{-2}	$<10^{-6}$	$<10^{-6}$

Table 1: Packet loss for a 3-stage router with N = 16 inputs and outputs, C = 16 links between stages and a load of 0.8. Throughout, D is the number of delay-lines per stage (including those of length zero) and δ is the fundamental delay-line unit.

As before, the packet loss probability decreases as the delay-line granularity is increased. Except for a delay-line granularity of 200 bytes, four delay-lines per stage is sufficient to provide an acceptable packet loss of less than 10^{-6}. Hence a total of twelve delay-lines are required, compared to over 64 in the single-stage case, with an associated simplification of the switching hardware required to switch packets into the correct delay-lines.

6.4 Queuing delay

Thus far, the discussion has concentrated on packet loss. Delay is also an important factor, which was studied during the simulation, for a bitrate of 40Gb/s per wavelength channel. Three roughly comparable scenarios are compared with respect to mean delay in Table 2, for different numbers of stages.

The delays increase with the number of stages, as is to be expected, since the longest delay-line in the 3-stage architecture is very much longer than the longest delay-line with a single stage. The mean packet delays vary by a factor of roughly 20. The confidence intervals on these delays were all several orders of magnitude smaller than the results themselves. Therefore a price to be paid for increased hardware economy is higher delay.

δ = 1750, 1 stage, D = 64	mean delay = 5.51 μs
δ = 1750, 2 stages, D = 7, C = 32	mean delay = 9.58 μs
δ = 1750, 3 stages, D = 4, C = 16	mean delay = 0.119 ms

Table 2 : Mean switch delays for various numbers of stages.

7. CONCLUSIONS

A new optical architecture has been presented for routing variable-length optical packets (e.g. IP packets), based upon asynchronous operation, void filling and a multi-stage architectural concept. The architecture was simulated under traffic that approximated to being negative exponentially distributed. The following conclusions can be drawn from this work:

- With only 2 or 3 stages, the architectures here produce a useful packet loss of under 10^{-6}.
- For these packet losses, extending the number of stages reduces the total number of optical delay-lines that are required, with a concomitant reduction in optical switching hardware within the router.

- The performance of the architecture is improved by making the delay-line granularity much greater than the mean packet length. With the traffic simulated, after increasing the granularity beyond a certain point, there is no further improvement in performance.

- There is a trade-off between *C*, the number of links between stages, and *D*, the number of delay-lines per stage i.e. increasing one involves decreasing the other, for a given packet loss rate. The possible cost and optical performance gains that might be made using this trade-off have yet to be quantified.

This paper has not considered the optical performance of the architecture with respect to factors such as crosstalk and noise. However, previous studies have shown that several stages similar to those proposed here can be cascaded [5], and experimental studies have demonstrated the validity of cascading up to 40 such stages with the aid of all-optical regeneration [11]. Hence it is clear that the architectures described in this paper are practical with regard to optical noise and crosstalk performance.

ACKNOWLEDGMENTS

The authors thank Ljubisa Tancevski of Alcatel Corporate Research Center, Richardson, for encouragement and useful discussions. David Hunter is funded by the EPSRC in the form of an Advanced Fellowship and Meow Chia acknowledges support from the Fujitsu Europe Telecom R&D Centre.

REFERENCES

1. D. K. Hunter, M. C. Chia, I. Andonovic: "Buffering in Optical Packet Switches", *IEEE/OSA Journal of Lightwave Technology*, vol. 16, no. 12, December 1998, pp2081-2094
2. V. Srinivasan, G. Varghese: "Fast Address Lookups using Controlled Prefix Expansion", *ACM Transactions on Computer Systems*, 1999, vol. 17, no. 1, pp1-40
3. A. Franzen, D. K. Hunter, I. Andonovic: "Low Loss Optical Packet Synchronizer Architecture", *All-Optical Networking: Architecture, Control and Management Issues*. SPIE International Symposium on Voice, Video and Data Communications, 1-5 November 1998, Boston, USA, paper 3531-40
4. L. Tancevski, A. Ge, G. Castanon, L. Tamil: "A New Scheduling Algorithm for Asynchronous, Variable Length IP Traffic Incorporating Void Filling", *OFC '99*, San Diego, CA, February 1999
5. D. K. Hunter, W. D. Cornwell, T. H. Gilfedder, A. Franzen, I. Andonovic: "SLOB: a Switch with Large Optical Buffers for Packet Switching", *IEEE/OSA Journal of Lightwave Technology*, vol. 16, no. 10, October 1998, pp1725-1736
6. J. M. Gabriagues, J. B. Jacob: "OASIS: A High-Speed Photonic ATM Switch – Results and Perspectives", *15th International Switching Symposium*, Berlin, vol. 2, paper C8.4, pp457-461, April 1995
7. M. G. Karol, M. J. Hluchyj, S. Morgan: "Input versus Output Queueing on a Space-Division Packet Switch", *IEEE Transactions on Communications*, vol. 35, December 1987, pp1347-1356
8. S. L. Danielsen, B. Mikkelsen, T. Durhuus, K. E. Stubkjaer: "WDM Packet Switch Architectures and Analysis of the Influence of Tuneable Wavelength Converters on the Performance", *IEEE/OSA Journal of Lightwave Technology*, vol. 15, no. 2, February 1997, pp219-227
9. S. L. Danielsen, C. Joergensen, B. Mikkelsen, K. E. Stubkjaer: "Analysis of a WDM packet Switch with Improved Performance Under Bursty Traffic Conditions Due to Tuneable Wavelength Converters", *IEEE/OSA Journal of Lightwave Technology*, vol. 16, no. 5, May 1998, pp729-735
10. L. Tancevski, L. Tamil, F. Callegati: "Non-Degenerate Buffers: An Approach for Building Large Optical Memories", *IEEE Photonics Technology Letters*, vol. 11, no. 8, August 1999
11. D. Chiaroni, B. Lavigne, L. Hamon, A. Jourdan, F. Dorgeuille, C. Janz, E. Grard, M. Renaud, R. Bauknecht, C. Graf, H. P. Schneibel, H. Melchior: "Experimental Validation of an All-Optical Network Based on 160Gbit/s Throughput Packet Switching Nodes", *24th European Conference on Optical Communications (ECOC '98)*, 20-24 September 1998, Madrid, Spain