To appear in the Proceedings of the Fourth Annual Optical Networking
and Communications Conference (OptiComm 2003), Dallas, TX

Performance Analysis of WDM Optical Packet Switches with a Hybrid Buffering Architecture

Lin Li, Stephen D. Scott, and Jitender S. Deogun
Dept. of Computer Science and Engineering, University of Nebraska, Lincoln, NE 68588-0115, USA
email: {lili, sscott, deogun}@cse.unl.edu

ABSTRACT

For lack of optical random access memory, optical fiber delay line (FDL) is currently the only technology to implement optical buffering. Feed-forward and feedback are two types of FDL structures in optical buffering, both have advantages and disadvantages. In this paper, we present a novel architecture for WDM optical packet switches with an effective hybrid FDL buffering that combines the merits of both feed-forward and feedback schemes. The core of the switch architecture is the arrayed waveguide grating (AWG) and the tunable wavelength converter (TWC). It requires smaller optical device sizes and fewer wavelengths and has less noise than feedback architecture. At the same time, feed-forward architecture can only do non-preemptive priority routing while ours supports preemptive priority routing. Our empirical results show that the new switch architecture significantly reduces packet loss probability.

Keywords: Optical buffering, hybrid FDL architecture, AWG, TWC

1. INTRODUCTION

Compared to optical circuit switching, optical packet switching is a long-term strategy to support high-speed transmission, data transparency, and reconfigurability. The main functions of an optical packet switch include: *routing*, *switching* and *buffering*. Routing and switching ensure that the switch maintains the information of the network topology, process the packets and switch the packets to the correct output ports. Buffering is used to resolve contentions that occur whenever two packets are routed to the same output port in the same time slot. Because of the lack of an optical random access memory, currently an optical buffer can only be implemented using fiber delay lines (FDL). A fiber delay line is just a fixed-length fiber. Once a packet enters it, the packet will emerge from the other side after a fixed time. Much work has been done on optical packet switch designs based on various buffering schemes.

Previous work, such as Haas,[1] divided the packet switch into two stages: *scheduling stage* and *switching stage*. The scheduling stage is for contention resolution. The switching stage is for packet switching. Zhong & Tucker[2] described a feed-forward shared-buffering strategy based on arrayed waveguide grating (AWG) and tunable wavelength converter (TWC). But this switch suffers from head-of-line blocking.[3] Chia et al.[4] extended these results, discussing both feed-forward and feedback buffering approaches. Xu et al.[3] and Hunter et al.[5] compare different switch designs and point out the basic problems in designing the optical packet switch.

Since feed-forward buffering does not support priority routing and feedback buffering suffers from more signal attenuation, in previous work[6] we proposed an optical packet switch architecture with a hybrid FDL buffering scheme. Our objective is to combine the merits of both feed-forward and feedback buffering that leads to efficient FDL utilization, fewer wavelength requirements, smaller component size, and good signal quality. In this paper, we further extend the switch design to overcome the shortcomings of our previous design. By extending the single-wavelength input/output switch architecture to the WDM version with multiple wavelength input/output ports and multiple switch planes, we show that the WDM technology can be used more effectively and the number of long FDLs can be significantly reduced.

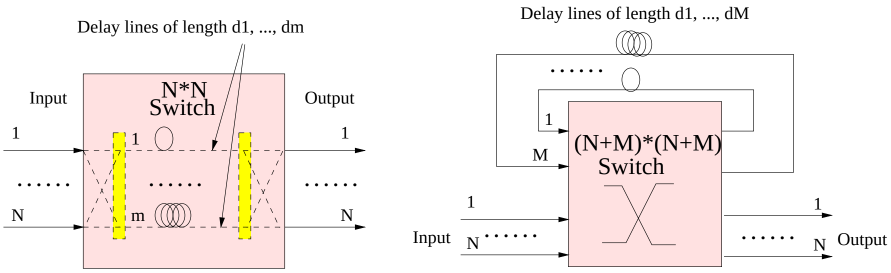
The rest of the paper is organized as follows: in Section 2, we review the characteristics of feed-forward and feedback buffering schemes. Then we describe our proposed switch architecture and present our scheduling algorithms. In Section 3, empirical results are analyzed and compared. Finally, in Section 4, we present conclusions and ideas for future work.

2. THE CORE SWITCH ARCHITECTURE

Throughout the paper, we assume the network is synchronized (slotted). The packet must be aligned to its time slot boundary before entering the switch. The packet header is processed electronically and and the payload stays in the optical domain. We use the following notation:
N: *number of incoming input/output ports of the switch*;
M: *number of feedback port of the switch*;
m: *size of the feed-forward buffer*;
n: *number of single wavelength input/output switch plane*;
ρ: *average traffic load rate*.

2.1. A Hybrid FDL Buffering Scheme

In general, we can categorize various designs of optical buffers into two classes: *feed-forward* and *feedback*, as shown in Figure 1. In the feed-forward method, the packets are fed into fiber delay lines of different lengths to resolve contention. Once a packet comes out of the FDL, it has to be switched out from the output port and has no chance to stay any longer inside the switch. In the feedback method, recirculation buffers are introduced for contention resolution. This architecture leads to larger switch fabric[*] and more crosstalk. Moreover, in the feedback method, a packet may recirculate in the switch several times when there is high contention for output ports. Because of this, the signal could suffer from significant power loss and noise. So a feed-forward architecture may be preferable in practice .[2, 21] However, feedback architecture allows packet priority routing since a lower-priority packet can be preempted by being sent into another loop. This feature is important for QoS in optical networks.



(a) Feed-forward buffering

(b) Feedback buffering

Figure 1. Feedback and feed-forward FDL architecture

Although the feed-forward architecture can also provide restricted priority routing, it cannot handle preemption. In order to retain the desirable features of the feed-forward architecture, we add a limited number of feedback FDLs to it to realize priority routing. It is expected that the feed-forward FDLs can handle most scheduling problems and the feedback buffer will resolve the remaining contentions and packet preemption. Our objective is to construct a feed-forward-like switch architecture to achieve feedback-like or better performance. As shown in Figure 2, the switch has a $(N + M) \times (N + M)$ fabric architecture ($M \ll N$). We employ the wavelength routing switch approach, in which TWC and AWG are the kernel parts, rather than the space switch approach since the latter generally suffers higher splitting/combining losses and more amplification noise with the increase of input/output number. Moreover, wavelength converters can help regenerate the signals such that wavelength routing switches can significantly reduce the noise.[2, 5] Through wavelength conversion, the complexity of the switching stage is also reduced greatly due to the static configuration of the AWG. We give each input/output port a set of FDLs in a way similar to the WASPNET switch. Although more FDLs are used, the scheduling is more flexible and its buffering ability is better. This switch architecture has the following features: (1) it supports priority routing; (2) compared to WASPNET switch, smaller AWGs are used which reduce crosstalk and noise; The required number of wavelengths is reduced ($N + M \ll 2N$), which saves system resources and cost. (3) From the empirical results 3, we can see that the packet loss probability can be significantly reduced even without sacrificing the delay performance. Finally, compared to WASPNET switch, if a packet has to be sent into a loop fiber, although the packet

---

[*] For example, a WASPNET switch[7] consists of a $2N \times 2N$ AWG, $4N$ TWCs and N sets of FDLs, each with m lines and requiring $2N$ wavelengths. Because only N input/output ports are for external signals, the resource utilization is 50%.

may pass the feed-forward FDLs first, it will not suffer from much more noise because the feedback buffers of the former have the same structure as the feed-forward part here, and the feedback buffers in this architecture are simple fiber delay lines. The only difference is the signals will pass one more AWG before sent out. Another arrangement of the feedback FDLs is to place the buffer between the two AWGs with the same stage as the feed-forward buffer part. Correspondingly, the length of the loop will be zero. This architecture has the same function as that shown in Figure 2 but there are minor differences in scheduling.

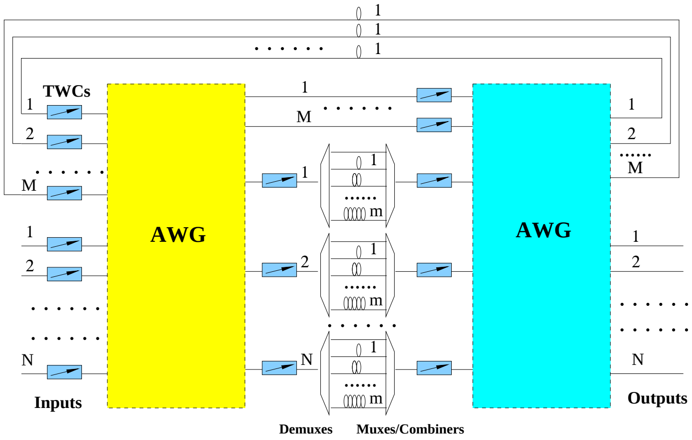


Figure 2. The hybrid FDL buffering switch architecture

2.2. The WDM Input Switch Architecture

The core switch architecture described in Section 2.1 has single-wavelength input/output ports. This architecture has two shortcomings: (1) *The WDM technology is only used inside the switch fabric. To exploit the huge bandwidth of optical networks, we should also use it in data transmission.* (2) *With only one switch plane as shown in Figure 2, we still need several long FDLs to achieve a low packet loss probability* (e.g. Li et al.[6] found that with $N = 16$, $\rho = 0.8$, $M = 4$, we can reach a packet loss probability of $10^{-3.9}$ if we set $m = 16$, which means each feed-forward FDL set has 16 fibers and the longest one can accommodate 16 packets). To overcome these shortcomings, we can upgrade the switch architecture to a WDM version by using multiplexers, combiners and multiple switch fabric planes similar to WASPNET ,[4] which is shown in Figure 3. Each plane contains a single wavelength optical packet switch and an $N \times N$ space switch. It has N input/output fibers and each fiber contains n wavelengths. With the space switch, this architecture also allows multiple optical packets to leave from the same output of a single plane on different wavelengths. The control unit will ensure there is no conflict in the combiners. From the experimental results in Section 3, we can see that by expanding the switch in wavelength domain, we can still achieve a good performance with much fewer and shorter FDL sets (e.g. with $N = 16$, $M = 4$, $\rho = 0.8$, $n = 4$, we can reach a packet loss probability of $10^{-3.58}$ with only $m = 4$).

Note that the hybrid FDL architecture proposed above has unit length feedback FDLs, which means once a packet is sent into a fiber loop, it will come back to some input port at the next time slot. We can instead use feedback FDLs with different lengths. This allows the feedback FDLs to accommodate more packets. To make the feedback FDLs more powerful, we can also give each output port a set of FDLs similar to the WASPNET switch. But this may introduce another problem: the signal will pass through more devices before coming back. This scheme would reduce the packet loss probability, but the data may suffer from more noise and system cost may be higher.

2.3. Scheduling Algorithm

In our switch architecture, the first AWG routes the packets to the appropriate FDL sets (the second stage of the TWCs) in order to prevent the head-of-line blocking problem. Each output port, therefore, logically has its own feed-forward buffer set and the feedback buffers are shared by all the inputs/outputs, as shown in Figure 4. The logically independent feed-forward buffer set of each output is actually a queuing system. At each time slot, the objective is to schedule the packet to the shortest idle time slot. If a packet is placed into the kth position of the queue, it can only be routed after k time slots. The feedback buffer, however, is not a single queue although there are M different positions. Since every feedback

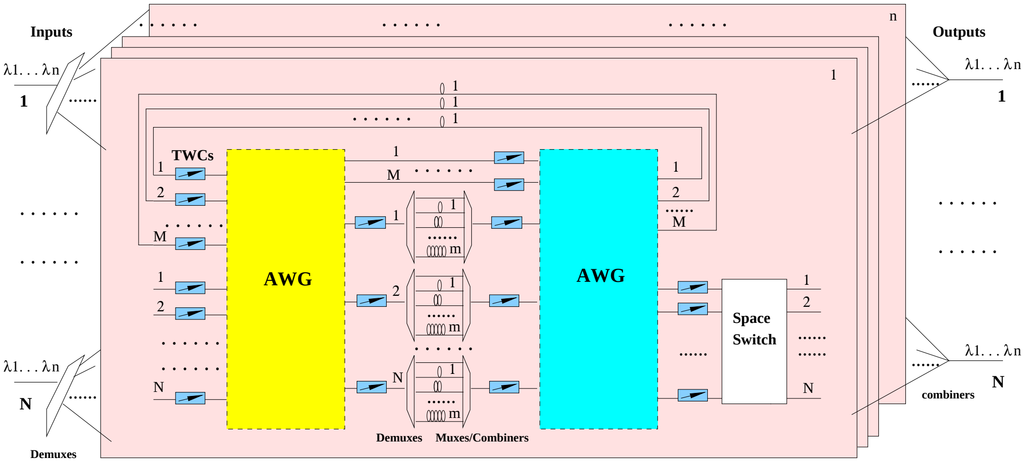


Figure 3. WDM version of the switch architecture

fiber is connected by an independent pair of input/output ports, all packets could be scheduled again at each time slot. The feedback buffers are like a waiting room system with a capacity of M. Once the queue of the destined output of a packet is full, the scheduler will check the waiting room for a free space and the packet will be dropped only if waiting room is also full.

Since the feedback buffers are employed, some strategy must be adopted to prevent a packet from looping in the switch indefinitely (i.e. that no packets stay in the waiting room indefinitely). The basic idea of the scheduling algorithm is: at each time slot, we first process the packets in the M feedback buffers. We start from buffer 1, then 2, etc. up to M. Then we process the packets from the N incoming inputs. When processing a packet p, we first attempt to route it to the shortest available feed-forward FDL of its specified output port. If no such a FDL exists, then p is routed to an available feedback buffer with the lowest index. If no feedback buffer is available, p is dropped.

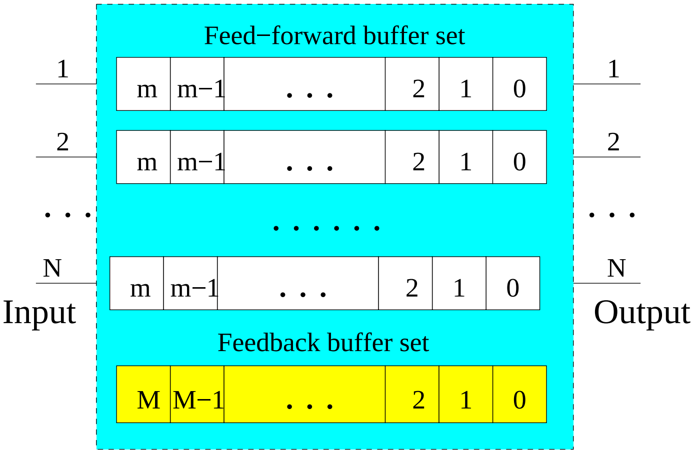


Figure 4. logically equivalent buffer architecture

At the beginning of a new time slot, all the packets in the feed-forward buffer have been shifted one slot forward, so at least the longest FDLs of each feed-forward buffer set will be free to store a new packet. Thus at least one packet in the feedback loops (from loop 1) can be stored in the feed-forward buffer sets and then sent out of the switch. This implies that at least feedback buffer 1 will be available to store a new packet. So a packet in feedback loop i in the current time slot will (if no feed-forward buffer can accommodate it because of contention) be sent to feedback loop j ($j \leq i - 1$) in the next time slot. Hence all packets that are sent into the feedback loops will be sent out after at Most M time slots. If non-priority routing is considered, we can also keep the packets in *first in first out* (FIFO) order in this way which is another nice feature of our switch.

```
Function Scheduling()
begin
    $i \leftarrow 0$
    while $i < M$ do
        if There is a packet at feedback input $i$.
            $j \leftarrow 0$
            while $j \leq m$ do
                find the shortest FDL $j$ in the feed-forward buffer which is free and
                no other packet to the same output after the same delay
                $j \leftarrow j + 1$
            end while
            if $j > m$ then
                find a free feedback buffer and drop the packet if no free buffer exists
                $lossNum \leftarrow lossNum + 1$
        $i \leftarrow i + 1$
    end while
    $i \leftarrow 0$
    while $i < N$ do
        if There is a packet at incoming input $i$.
            $j \leftarrow 0$
            while $j \leq m$ do
                find the shortest FDL $j$ in the feed-forward buffer which is free and
                no other packet to the same output after the same delay
                $j \leftarrow j + 1$
            end while
            if $j > m$ then
                find a free feedback buffer and drop the packet if no free buffer exists
                $lossNum \leftarrow lossNum + 1$
        $i \leftarrow i + 1$
    end while
    return lossNum
end Function()
```

Figure 5. The scheduling algorithm

If priority routing is considered, since preemption may happen, we give each packet a certain priority (e.g. between 1 and 5). Once the packet passes through the loop buffer, we increase its priority by one and we always switch out the packets with higher priority. Thus we process all switch inputs at the same time rather than processing all feedback loops before incoming inputs. It can be easily proved that with this mechanism, we can prevent a packet from getting stuck in the switch.

The pseudo code for the scheduling algorithm is shown in Figure 5.

Note that with the space switch, more than one packet could be routed out from the same output port of each switch plane if at most n packets in different wavelengths are routed to each switch output from all planes at each time slot. For example, at a time slot, if two packets are destined for a certain output k, while the remaining packets (less than $N-2$) are destined to the other $N-1$ output ports, then the switch could send these two packets out at the same time slot if during that slot, the number of all packets for output k from other planes is less than $n-2$. So for each time slot, there are two more scheduling constraints:

1. *At most n packets from all switch planes can be routed out to each switch output, and*
2. *At most N packets can be routed out from each switch plane.*

Although this strategy can use the system resources more efficiently, another potential problem, sequential biased routing (SBR) may arise because it gives higher priority to the lower numbered inputs by first routing the packet from input 1, then 2, etc. Due to the characteristics of a fiber delay line — once we send the signal into a fiber delay line, we can only process it again after it emerges from the other end of the fiber, this scheduling scheme may affect the overall performance of the switch, irrespective of whether switch architecture is based on feed-forward or feedback. An example is shown in Figure 6. Given the switch with three switch planes and m=3, the queuing statuses[†] of all planes are shown in Figure 6(a). Suppose now we have two more packets entering the second plane, which are heading toward output ports 2 and 3 separately, to the left of Figure 6(a), how shall we route them? Assuming the packet to output 3 comes from a lower numbered input, we route it first and put it into the shortest FDL in the second queue. After that we can only put packet to output 2 in the third position of queue 2 due to routing constraint 1. The result is shown in Figure 6(b). This will lead to a longer average latency of the packets because a better routing method is to route the packet to output 2 first, which is shown in Figure 6(c). Thus, even though we can make locally optimal scheduling decisions by allowing multiple packets (all destined output i) to be switched out from the same plane j at one time slot, this approach can adversely affect global performance. Since the problem is from the scheduling dependency of different packets of all planes, this trend could become worse when the traffic load is high and the number of switch planes increases, which we can see from the experimental results described in Section 3.

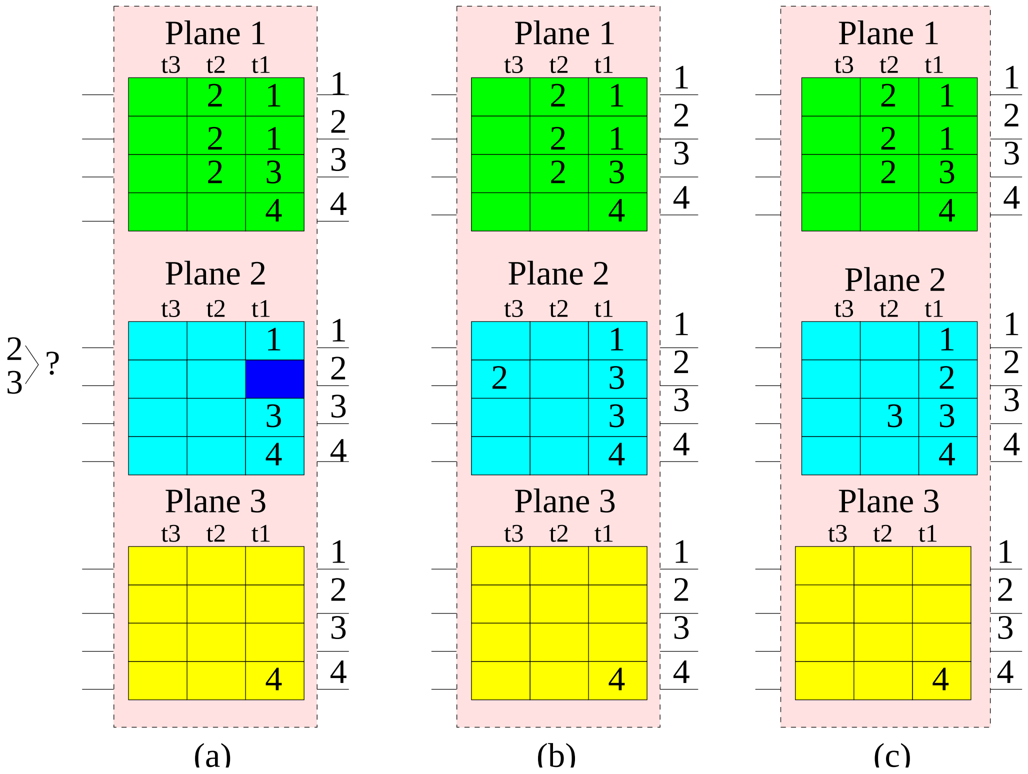


Figure 6. A sequential biased routing example

3. EMPIRICAL RESULTS

We evaluate our switch architecture and compare it to a feed-forward switch in terms of *packet loss probability* and *switch latency*. Obviously, the addition of feedback buffers will increase the switch complexity and cost (e.g. currently, the cost of the AWG and the wavelength requirement will increase linearly with M, and the crosstalk will also increase with the

---

[†] Recall that we allow multiple packets destined to the same output to be routed out from the same plane at each time slot with the space switch. So in Figure 6, there are two packets destined for output 1 at the first time slot in switch plane 1. One of the packets occupies the first buffer of the queue to output 2. This also occurs to the packets destined to output 2 in the second time slot.

size of the AWGs). However, we show that these increases are small in relation to the advantages gained by decreased packet loss probability. Note that the feed-forward switch architecture is actually an unfolded version of the WASPNET feedback geometry. It has the same packet loss probability as the feedback architecture except that it cannot support packet preemption.[4] Therefore, in our experiments we only compare our architecture to a feed-forward architecture.

Under some traffic statistics, the packet loss probability is closely related to ρ. The higher ρ, the higher loss probability the switch will have ($\rho = 0.8$ is usually regarded as a practical traffic load[13]). Sometimes deflection routing is combined with optical buffering which means if the switch can not buffer a packet, the packet may be sent out from another output port (the packet is not dropped). But since it does not take the expected path, in this paper, we still count it as a lost packet. Switch latency is calculated by averaging the time slots the packets stay in the switch.

Uniform traffic is the simplest traffic model used to analyze a switch architecture. Given that a traffic load ρ, is independent of previous time slot and other input ports, the probability of i packets arriving at a switch can be represented by a Binomial distribution: $P(i) = \binom{N}{i}(\rho)^i(1-\rho)^{(N-i)}$, where N is the number of inputs at a time slot for the switch. Although real Internet traffic is much more complicated (e.g. exponential or heavy tailed distributed), it can still provide important testing results for the switch. In the following experiments, we generate 10^8-10^{13} packets to test each set of parameters.

Figures 7 and 8 compare the simulation results of the hybrid switch with a 16 × 16 feed-forward section and a four input/output feedback section (i.e $N = 16$, $M = 4$, and $n = 4$) to a 16 × 16 feed-forward switch ($M = 0$) with different values of m under uniform traffic. From the figures, we can see that by adding the feedback buffers, the packet loss probability is greatly reduced, e.g. at a traffic load $\rho = 0.8$, without the loop buffers ($m = 4$), the probability is about $10^{-3.5}$, while for our design with four feedback buffers, the probability[‡] is less than 10^{-5}. When ρ is very high, e.g $\rho = 1$, the performance is similar for both switch architectures. This is because the switch buffers are always full and the few feedback buffers cannot help much. As indicated in Figure 8, the average latency of our switch is quite close to the feed-forward switch, especially for $\rho < 0.9$.

From Figure 7, we can also see that the packet loss probability of our switch ($M = 4$, $m = 4$) is quite similar to a feed-forward switch with $M = 0$ and $m = 6$, and the packet loss probability of our switch with $M = 4$, $m = 5$ is similar to a feed-forward switch with $M = 0$ and $m = 7$. This suggests that in terms of packet loss probability, adding four unit length feedback buffers is similar to increasing the number of longer fibers (of lengths 5 and 6 or 6 and 7) to each feed-forward FDL set. Although the AWGs in our architecture have larger sizes, the MUXs/DEMUXs will have smaller sizes and we avoid using longer fibers inside the switch, which will make the fabrication of the switch easier and cheaper. We are running more experiments to estimate how to find m when mapping from our architecture (with $M > 0$) to an equivalent (in terms of packet loss probability) feed-forward architecture. We believe that it will be a function of N, M, and m.

Figures 9 and 10 compare the simulation results of the hybrid buffering switch with different numbers of feedback loops ($N = 16$, $m = 4$) under uniform traffic. From these we can see that given a certain number of feedback buffers, we can significantly reduce the packet loss probability. Although we can improve this performance by increasing M, within some scope, there will be no significant change (e.g. the performance is similar for $M = 4$ and $M = 5$). This is more obvious in Figure 11. As stated before, because the cost of the switch is mainly determined by the size of its components, some tradeoff has to be made. Similarly, the average switch latency does not change much for different M.

Figures 11 and 12 give the results for other switch parameters, and they are similar to the results in Figures 9 and 10. The packet loss probability drops to nearly 10^{-8} with $m = 6$ and $M = 8$ when $\rho = 0.8$.

Figures 13 and 14 compare the perfomance improvement with increasing n — the number of switch planes, which is also the wavelength channels of each input/output port (so $n = 1$ is equivalent to the switch of Li. et al[6]). We can see that the larger the number of wavelengths per input/output, the lower the packet loss probability will be. E.g. with a switch of $N = 16$, $m = M = 4$, the packet loss probability drops from $10^{-3.03}$ at $n = 2$ to $10^{-6.78}$ at $n = 6$. In additon, the average latency also decreases with increasing n. Another item worth noting about Figure 14 is that when the traffic load is high ($\rho \approx 1$), the average latency increases much faster with increasing the number of switch planes. This is because the scheduling algorithm is sequential biased as described in Section 2.3. As the number of packets and switch planes increase, the higher is the probability of getting stuck in a local optimum.

[‡] Since the packet loss probability drops very fast—less than 10^{-10}, for $M = 4$ when ρ is less than 0.8 and $M = 0$ when ρ is less than 0.6, we do not extend the curves after that. This is the same for all the following figures.

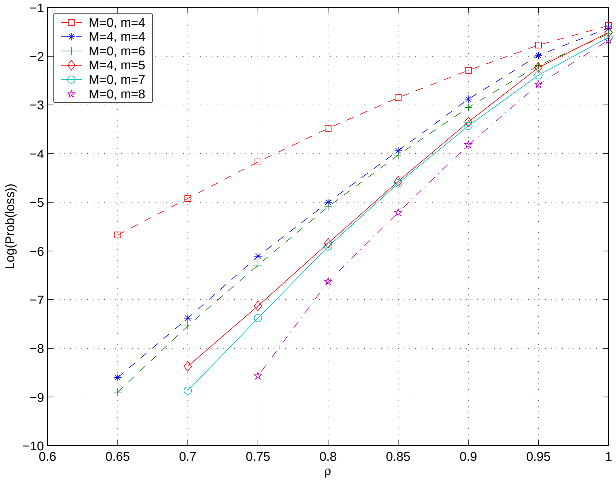


Figure 7. Packet loss probability, $N = 16, n = 4$.

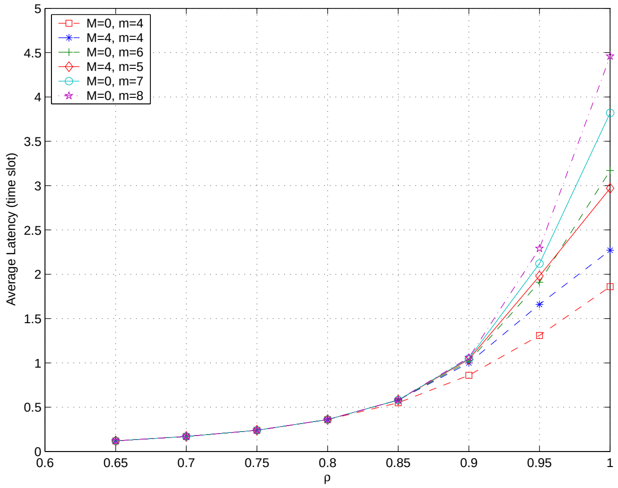


Figure 8. Average latency, $N = 16$, $n = 4$.

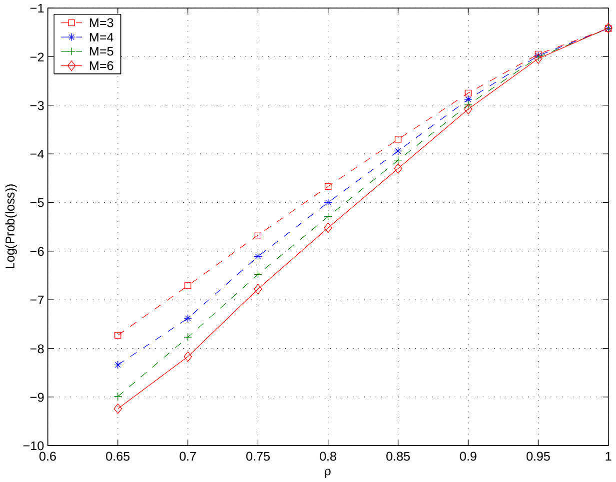


Figure 9. Packet loss probability with different M, $N = 16$, $m = n = 4$.

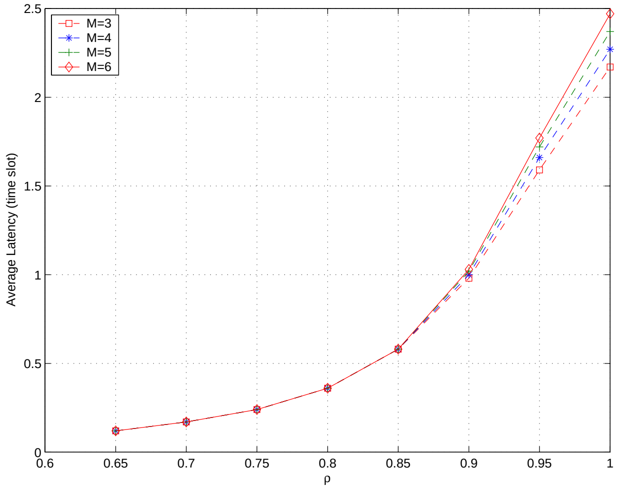


Figure 10. Average latency with different M, $N = 16$, $m = n = 4$.

Figures 15 and 16 compare the hybrid buffering switch architecture ($N = 16$, $M = 4$ and various m values) under the bursty traffic model. The inter-arrival time between the packets is exponentially distributed and the average burst length is 4. The traffic load is $\rho = 0.8$. We can see that although the hybrid switch still has better performance than the feed-forward, but the improvement is not as significant as it is under uniform traffic. However, the relative difference between $M = 4$ and $M = 0$ with different m is very stable, which implies that we can get a steady improvement with the extra feedback buffers.

In our last experiment, we evaluated the need for our priority-based scheduling algorithm versus our basic scheduler which has no priority control (both described in Section 2.3). We randomly assigned priorities to packets generated by our uniform traffic model and measured the fraction of packet drops that were handled incorrectly by our basic scheduler (i.e. when a higher-priority packet was dropped). Results are in Figure 17. From the large value in the figure, we see that the priority routing scheduling algorithm is necessary in this case.

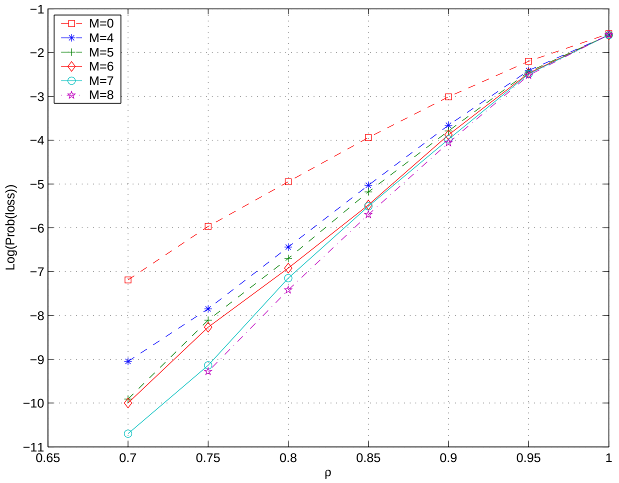


Figure 11. Packet loss probability with different M, $N = 32$, $m = 6$, $n = 4$.

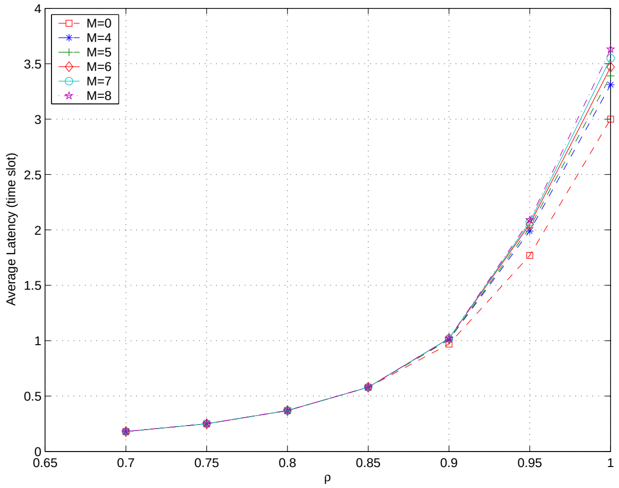


Figure 12. Average latency with different M, $N = 32$, $m = 6$, $n = 4$.

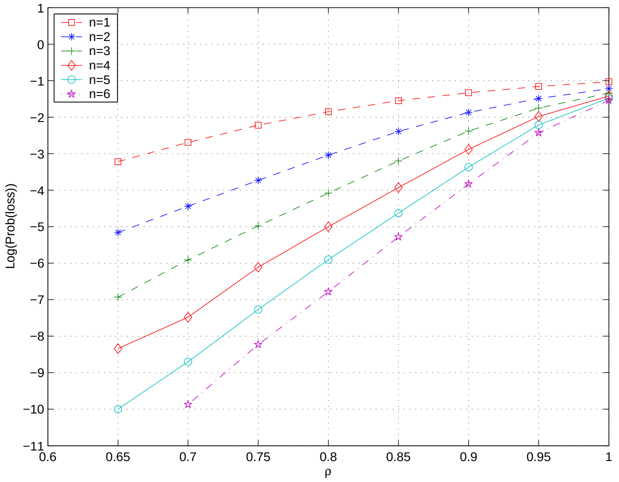


Figure 13. Packet loss probability with different n, $N = 16$, $m = M = 4$.

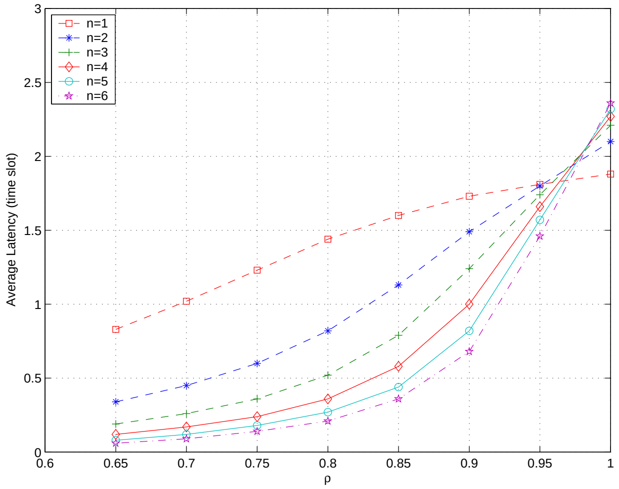


Figure 14. Average latency with different n, $N = 16$, $m = M = 4$.

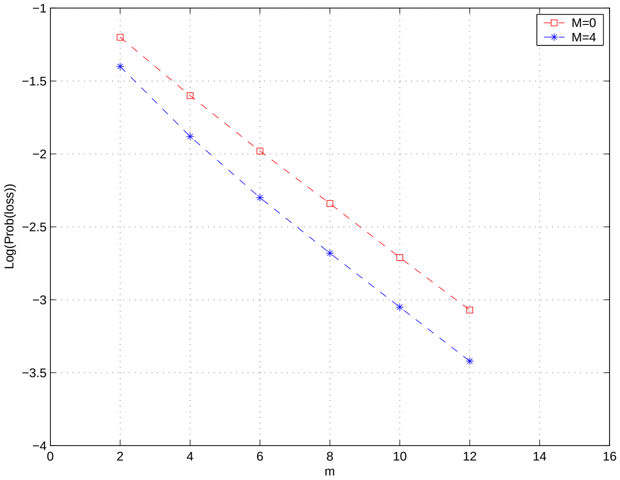


Figure 15. Packet loss probability with bursty traffic, $N = 16$, $n = 4$.

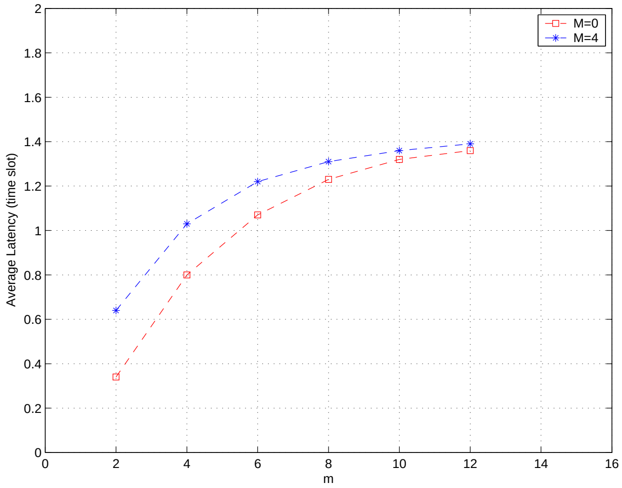


Figure 16. Average latency with bursty traffic, $N = 16$, $n = 4$.

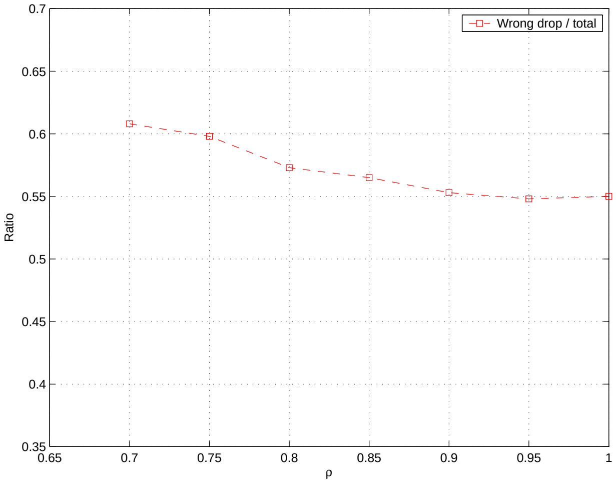


Figure 17. Ratio of packet loss of non-priority routing to priority routing, $N = 16, m = M = n = 4$.

4. CONCLUSIONS AND FUTURE WORK

We presented a novel architecture for WDM optical packet switches with hybrid buffering, which presents an extension of our previous work.[6] This switch architecture combines the merits of feedback and feed-forward switching schemes and requires smaller component sizes and fewer wavelengths. Moreover, it leads to good signal quality and can implement priority routing. The buffering scheme shows excellent performance in terms of packet loss probability without incurring significant increases in average latency or switch cost. Compared to our previous work, proposed architecture accepts data transmitted in multiple wavelengths and does not use many long FDLs in the feed-forward buffering section. Experimental results suggest that there exists a relationship in terms of packet loss probability between our proposed architecture and the feed-forward architecture. If such a relationship can be theoretically verified, it would make an analysis of our architecture straightforward, applying the results of Danielsen et al.[9] and Chia et al.[4] We are investigating this further.

Plans for future work include theoretical analysis and comparative performance evaluation of our switch and other two proposed switch architectures which have non-unit-length feedback buffer structures. We also plan to test our switch with other traffic patterns and develop new non-sequential biased scheduling algorithms.

Acknowledgments

This work was funded in part by NSF grants CCR-9877080 (with matching funds from CCIS) and CCR-0092761. The authors thank Dr. Byrav Ramamurthy and Dr. Meow C. Chia for their useful discussions and the anonymous reviewers for their helpful comments.

REFERENCES

1. Z. Haas, "The staggering switch: an electronically controlled optical packet switch," *IEEE/OSA Journal of Lightwave Technology* , pp. 925–936, May, 1993.
2. W. Zhong and R. Tucker, "Wavelength routing-based photonic packet buffers and their applications in photonic packet switching systems," *IEEE/OSA Journal of Lightwave Technology* , pp. 1737–1745, Oct, 1998.
3. L. Xu, H. Perros, and G. Rouskas, "Techniques for optical packet switching and optical burst switching," *IEEE Communications Magazine* , pp. 136–142, Jan, 2001.
4. M. C. Chia, D. Hunter, I. Andonovic, P. Ball, I. Wright, S. Ferguson, K. Guild, and M. O'Mahony, "Packet loss and delay performance of feedback and feed-forward arrayed-waveguide grating-based optical packet switches with WDM inputs-outputs," *IEEE/OSA Journal of Lightwave Technology* , pp. 1241–1253, Sept, 2001.
5. D. Hunter, M. Chia, and I. Andonovic, "Buffering in optical packet switches," *IEEE/OSA Journal of Lightwave Technology* , pp. 2081–2094, Dec, 1998.

6. L. Li, S. Scott, and J. Deogun, “A novel hybrid fiber delay line buffering architecture for optical packet switching,” *to appear in the Proc. of IEEE Globecom*, 2003.
7. D. K. Hunter, M. H. M. Nizam, M. C. Chia, I. Andonovic, K. M. Guild, A. Tzanakaki, M. O’Mahony, J. D. Bainbridge, M. F. C. Stephens, R. V. Penty, and I. White, “WASPNET: a wavelength switched packet network,” *IEEE Communications Magazine* , pp. 120–129, Mar, 1999.
8. D. Hunter and I. Andonvic, “Advances in photonic packet switching: an overview,” *IEEE Communications Magazine* , pp. 84–94, Feb, 2000.
9. S. L. Danielsen, B. Mikkelsen, C. Joergensen, T. Durhuus, and K. Stubjaer, “WDM packet switch architecture and analysis of the influence of tunable wavelength converters on the performance,” *IEEE/OSA Journal of Lightwave Technology* , pp. 219–226, Feb, 1997.
10. S. L. Dannielsen, C. Joergensen, B. Mikkelsen, and K. E. Stubjaer, “Analysis of a WDM packet switch with improved performance under bursty traffic conditions due to tunable wavelength converters,” *IEEE/OSA Journal of Lightwave Technology*, pp. 729-735, May, 1998.
11. C. Guillemot, M. Renaud, P. Gambini, C. Janz, I. Andonovic, R. Bauknecht, B. Bostica, M. Burzio, F. Callegati, M. Casoni, D. Chiaroni, F. Chlerot, S. Danielsen, F. Dorgeuille, A. Dupas, A. Franzen, P. B. Hansen, D. K. Hunter, A. Kloch, R. Krahenbuhl, B. Lavigne, A. L. Corre, C. Raffaelli, M. Schilling, J. Simon, and L. Zucchelli, “Transparent optical packet switching: the European ACTS KEOPS project approach,” *IEEE/OSA Journal of Lightwave Technology*, pp. 2117-2133, Dec, 1998.
12. T. El-Bawab and J. Shin, “Optical Packet Switching in Core Networks: Between Vision and Reality,” *IEEE Communications Magazine*, pp. 60-65, Sept, 2002.
13. D. K. Hunter and I. Andonvic, “Approaches to optical Internet packet switching,” *IEEE Communications Magazine*, pp. 116-122, Sept, 2000.
14. S. Yao and B. Mukherjee and S. Dixit, “Advances in photonic packet switching: an overview,” *IEEE Communications Magazine*, pp. 84-94, Feb, 2000.
15. S. Yao, B. Mukherjee and S. Dixit, “All-optical packet switched networks: a study of contentions resolution schemes in an irregular mesh networks with variable-sized packets,” *Proceedings of SPIE*, (OPTICOMM 2000), Oct, 2000.
16. F. Callegati, G. Corazza, and C. Raffaelli, “Exploitation of DWDM for optical packet switching with quality of service guarantees,” *IEEE Journal on Selected Areas in Communications*, pp. 190-201, Jan, 20002.
17. C. Rallaelli, “Architecture and performance of optical packet switching over WDM,” *Photonic Network Communications*, pp. 91-99, 2001.
18. J. Y. Hui, *Switching and traffic theory for integrated broadband networks*, Kluwer Academic Publishers, 1990.
19. W. Willinger and V. Paxson, “Where mathematics meets the Internet,” *Notices of American Mathematics Society*, pp. 961-970, Aug, 1998.
20. M. C. Chia, D. K. Hunter, I. Andonovic, P. Ball, and I. Wright, “Optical packet switches: a comparison of designs,” *Proc. of IEEE ICON International Conference on Networks*, pp. 365-369, 2000.
21. R. Ramaswami and K. N. Sivarajan, *Optical Networks: a practical perspective*, Morgan Kaufmann, 2001.