A Novel Fiber Delay Line Buffering Architecture for Optical Packet Switching

Lin Li
University of Nebraska-Lincoln, lili@cse.unl.edu

Stephen Scott
University of Nebraska-Lincoln, sscott2@unl.edu

Jitender S. Deogun
University of Nebraska-Lincoln, jdeogun1@unl.edu

IEEE Global Telecommunications Conference, 2003. GLOBECOM '03. Volume: 5
Digital Object Identifier: 10.1109/GLOCOM.2003.1258747
Publication Year: 2003 , Page(s): 2809 - 2813 vol.5

A Novel Fiber Delay Line Buffering Architecture for Optical Packet Switching

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— **Due to the lack of optical random access memory, optical fiber delay line (FDL) is currently the only way to implement optical buffering. Feed-forward and feedback are two kinds of FDL structures in optical buffering. Both have advantages and disadvantages. In this paper, we propose a more effective hybrid FDL architecture that combines the merits of both schemes. The core of this switch 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, it can facilitate preemptive priority routing which feed-forward architecture cannot support. Our numerical results show that the new switch architecture significantly reduces packet loss probability.**

I. 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, processes the packets and switches the packets to the correct output ports. Buffering is used for resolving 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 optical buffers can only be implemented using fiber delay lines. 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. Chia et al. [4] extended these results, discussing both feed-forward and feedback buffering approaches. Xu et al. [3] and Hunter et al. [5] compared different switch designs and pointed out the basic problems in designing optical packet switch.

Since feed-forward buffering does not support priority routing and feedback buffering suffers from more signal attenuation, we propose a novel 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 more efficient FDL utilization, fewer wavelength requirements, smaller component size, and good signal quality.

The rest of the paper is organized as follows: in Section II, we review the characteristics of feed-forward and feedback buffering schemes and describe the proposed switch architecture and present our scheduling algorithms. In Section III, numerical results are analyzed and compared. Finally, in Section IV, we give the conclusions of the paper and propose some future work.

II. The Proposed Hybrid FDL Architecture

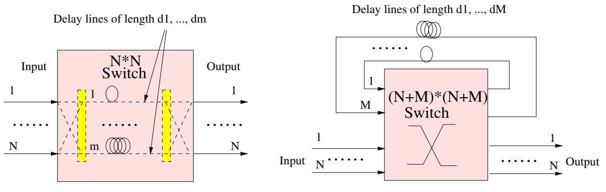
Throughout the paper, we assume the network is synchronized (slotted). A packet must be aligned to its time slot boundary before entering the switch. The packet header is processed electronically and the payload stays in optical domain. We use the following notation:
N: *number of incoming input/output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

In general, we can categorize various designs of optical buffers into two classes: *feed-forward* and *feedback*, as shown in Fig. 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 inside the switch any longer. In the feedback method, recirculation buffers are introduced for contention resolution. Because of this, the architecture leads to larger switch fabric[1] 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 preferred in practice [2], [16]. 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 to provide QoS in optical networks.

Although the feed-forward architecture can also provide some kind of priority routing (e.g. we can send the packets

[1] 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%.

0-7803-7974-8/03/$17.00 © 2003 IEEE



(a) Feed-forward buffering

(b) Feedback buffering

Fig. 1. Feed-forward and feedback FDL architecture

with lower priority to the longer FDLs and the packets with higher priority to the shorter FDLs), this architecture cannot handle the case if a packet has to be preempted. 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 Fig. 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 and so wavelength routing switches can significantly improve the noise performance [2], [5]. Through wavelength conversion, the complexity of the switching stage is also greatly reduced due to the static configuration of the AWG. We give each input/output port a set of FDLs as the WASPNET switch did. Although more FDLs are used, the scheduling will be more flexible and its buffering ability will be better. This switch architecture has the following features: (1) it supports priority routing; (2) compared to WASPNET switch, smaller AWGs are used which reduces crosstalk and noise; (3) the required number of wavelengths is reduced, which saves system resources and cost. Finally, compared to WASPNET switch, if a packet has to be sent into a loop fiber, although the packet may pass the feed-forward FDLs first, it will not suffer from 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 that the signals will pass one more AWG before being 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 Fig. 2 but there are minor differences in scheduling.

The switch fabric has single-wavelength input/output ports. We can upgrade it to a WDM version by using multiplexers, combiners and multiple of the switch fabric planes [6].

Note that the above hybrid FDL architecture 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 instead use feedback FDLs with different lengths to accommodate more packets. To make the feedback FDLs more powerful, we can also give each output port a set of FDLs as the WASPNET switch did. It 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 will be higher.

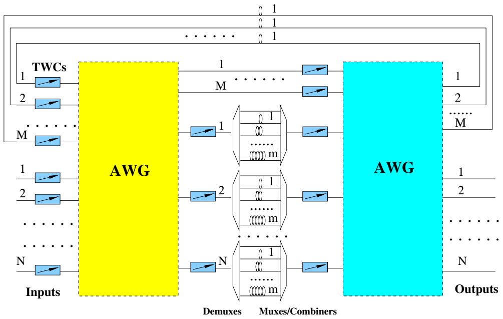


Fig. 2. The hybrid FDL buffering switch architecture

A. Scheduling Algorithms

In our switch architecture, each incoming input functionally has its own feed-forward buffer set but the feedback buffers are shared by all the inputs/outputs. Since the feedback buffers are employed, some strategy must be adopted to prevent a packet from looping in the switch 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. Again, we start from the first buffer 1, then 2, etc. up to N. 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.

After a 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, in this way, we can also keep the packets in first in first out (FIFO) order which is another nice feature of a switch. 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 the packet from getting stuck in the switch.

III. Numerical Results

We compared our switch architecture 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 N and M, and the crosstalk will also increase with the scale of the components). However, we will show that these increases are small with respect to the decrease in 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 a usually regarded as a practical traffic load [10]). 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 doesn't 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 the switch architecture. Given a ρ, the traffic load is independent of previous time slot and other input ports. So for a switch fabric with N inputs at a time slot, the probability of i packets arriving at the switch could be represented by a Binomial distribution: $P(i) = \binom{N}{i}(\rho)^i(1-\rho)^{(N-i)}$. 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 10^9-10^{10} packets to test each set of parameters.

Figs. 3 and 4 compare the simulation results of the hybrid switch with a 16×16 feed-forward section and a four input/output feedback section (i.e $N = m = 16$, $M = 4$) to a 16×16 feed-forward switch under uniform traffic. From the figures, we can see that the packet loss probability is greatly reduced, e.g. at a traffic load $\rho = 0.8$, without the loop buffers, the probability is $10^{-3.9}$, while for our design with four feedback buffers, the probability [2] is less than $10^{-4.7}$. When ρ is very high, e.g $\rho \geq 0.95$, 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 Fig. 4, our switch's average latency is quite close to the feed-forward switch, especially for $\rho < 0.9$.

Figs. 5 and 6 compare the simulation results of the hybrid buffering switch with different numbers of feedback loops ($N = m = 16$) under uniform traffic. From these we can see that given enough feedback buffers, we can significantly reduce the packet loss probability. Although we can improve this performance by increasing M, within some scope, there will not be significant change (e.g. the performance is similar for $M = 4$ and $M = 5$). This is more obvious in Fig. 9. 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 a lot for different M.

Figs. 7-9 give the results for other switch parameters, and they are similar to the results in Fig. 3-6. Another item worth noting about Fig. 7-9 is that the feed-forward buffer is the one that most controls the switch's performance. Indeed, the packet loss probability in in Fig. 7 ($m = 16$) ranged from $10^{-2.46}$ to $10^{-2.95}$ for $\rho = 0.9$, while in Fig. 9 ($m = 32$, similar values of M), packet loss probability ranged from $10^{-4.23}$ to $10^{-4.70}$. Actually, the logically independent feed-forward buffer set of each output is 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 buffers, however, are like a waiting room system with capacity of M. All packets could be scheduled again at each time slot. In terms of packet loss probability, adding more feedback buffers is similar to increasing the number of longer fibers to each feed-forward FDL set. We are still investigating the relationship between the effect of adding the two different buffers and we belive that it is a function of N, M, and m.

Figs. 10 and 11 compare the same hybrid buffering switch architecture ($N = m = 16$ and $M = 4$) as in Figs. 3 and 4, but we evaluated the performance under the bursty traffic model in [1] with a mean burst length of four. The model is a simple three-state (idle, from idle to burst and from burst to another burst) Markov chain. 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.

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 II-A). 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 Fig. 12. From the large value in the figure, we see that priority routing scheduling algorithm is necessary in this case.

IV. Conclusions and Future Work

We proposed a hybrid FDL buffering architecture for optical packet switching that combines the merits of feedback and feed-forward schemes. This switch architecture requires smaller component sizes and fewer wavelengths. It will lead to good signal quality and can implement priority routing. The buffering scheme shows good performance in terms of packet loss probability without incurring significant increases in average latency or switch cost.

[2] Since the packet loss probability drops very fast—less than 10^{-10}, for $M = 4$ when ρ is less than 0.6 and $M = 0$ when ρ is less than 0.5, we do not extend the curves after that. And this is the same for all the following figures.

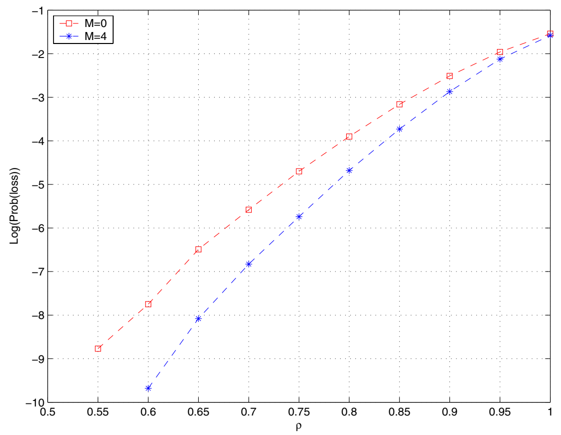


Fig. 3. Packet loss probability, $N = m = 16$.

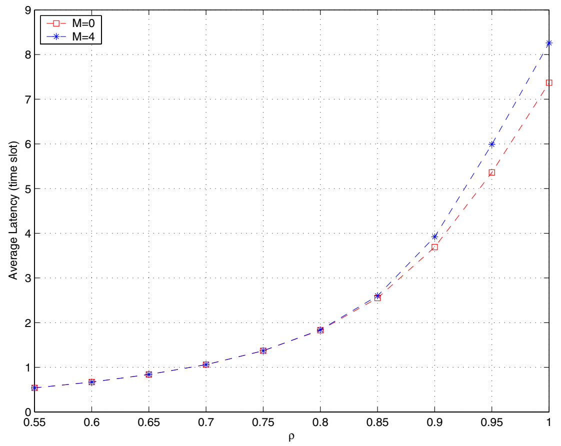


Fig. 4. Average latency, $N = m = 16$.

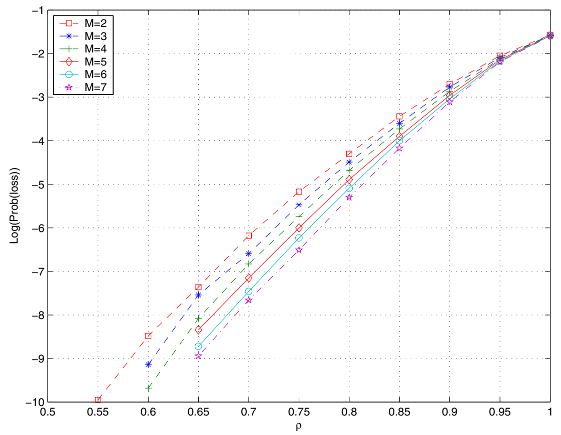


Fig. 5. Packet loss probability with different M, $N = m = 16$.

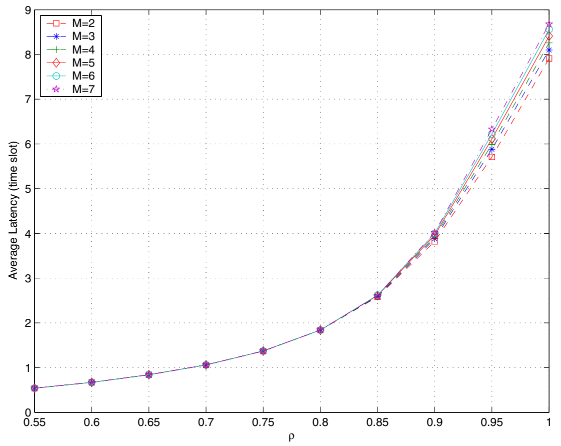


Fig. 6. Average latency with different M, $N = m = 16$.

Plans for future work include the theoretical analysis and performance evaluation of a WDM version of our switch and other two proposed switch architectures.

Acknowledgments

This work was funded in part by NSF grants CCR-9877080 (matching funds from CCIS) and CCR-0092761. The authors thank Byrav Ramamurthy and Qun Li 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 J. of Lightwave Technol.* , vol. 11, 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 J. of Lightwave Technol.* , vol. 16, pp. 1737–1745, Oct, 1998.

[3] L. Xu, H. Perros, and G. Rouskas, "Techniques for optical packet switching and optical burst switching," *IEEE Commun. Mag.* , vol. 39, pp. 136–142, Jan, 2001.

[4] M. C. Chia, D. K. 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 J. of Lightwave Technol.* , vol. 19, pp. 1241–1253, Sept, 2001.

[5] D. K. Hunter, M. C. Chia, and I. Andonovic, "Buffering in optical packet switches," *IEEE/OSA J. of Lightwave Technol.* , vol. 16, pp. 2081–2094, Dec, 1998.

[6] L. Li, S. Scott, and J. Deogun, "Performance Analysis of WDM Optical Packet Switches with a Hybrid Buffering Architecture," *to appear in Proc. of SPIE OptiComm*, Oct. 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 Commun. Mag.* , vol. 39, pp. 120–129, Mar, 1999.

[8] 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 J. of Lightwave Technol.* , vol. 15, pp. 219–226, Feb, 1997.

[9] T. El-Bawab and J. Shin, "Optical Packet Switching in Core Networks: Between Vision and Reality," *IEEE Commun. Mag.*, vol. 39, pp. 60-65, Sept, 2002.

[10] D. K. Hunter and I. Andonvic, "Approaches to optical Internet packet switching," *IEEE Commun. Mag.*, vol. 39, pp. 116-122, Sept, 2000.

[11] S. Yao and B. Mukherjee and S. Dixit, "Advances in photonic packet switching: an overview," *IEEE Commun. Mag.*, vol. 39, pp. 84-94, 2000.

[12] 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," *Proc. of SPIE OptiComm*, 2000.

[13] F. Callegati, G. Corazza, and C. Raffaelli, "Exploitation of DWDM for optical packet switching with quality of service guarantees," *IEEE J. Select. Areas Commun.*, vol. 20, pp. 190-201, Jan, 20002.

[14] C. Rallaelli, "Architecture and performance of optical packet switching over WDM," *Photonic Network Communications*, vol. 3, pp. 91-99, 2001.

[15] 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.

[16] R. Ramaswami and K. N. Sivarajan, *Optical Networks: a practical perspective*, Morgan Kaufmann, 2001.

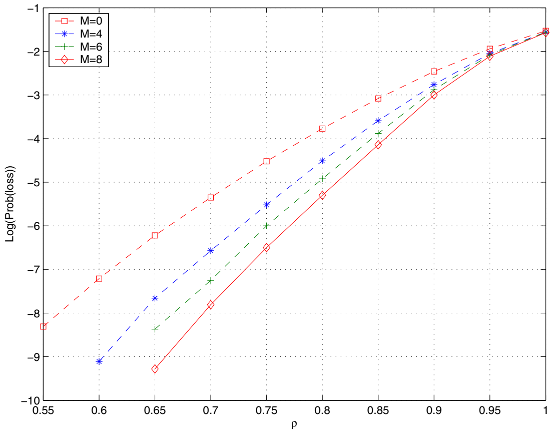


Fig. 7. Packet loss probability with different M, N=32, m =16.

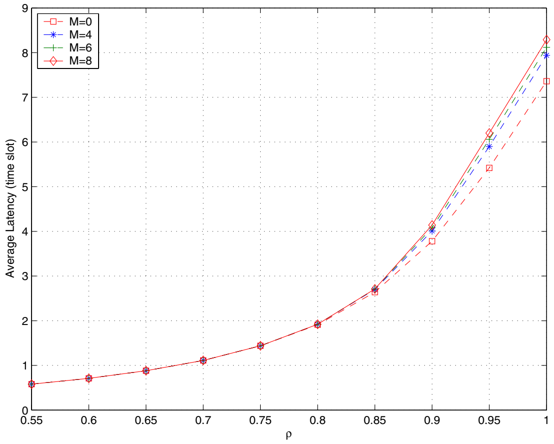


Fig. 8. Average latency with different M, N=32, m =16.

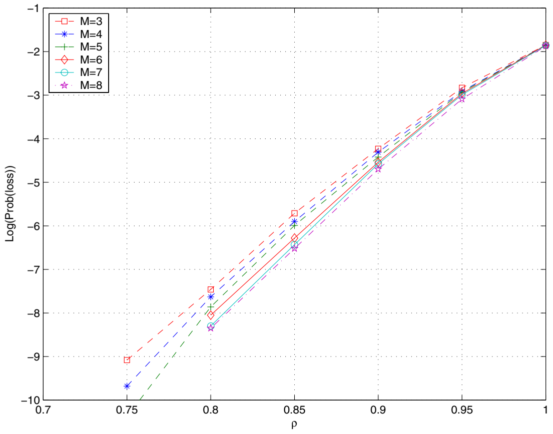


Fig. 9. Packet loss probability with different M, $N = m = 32$.

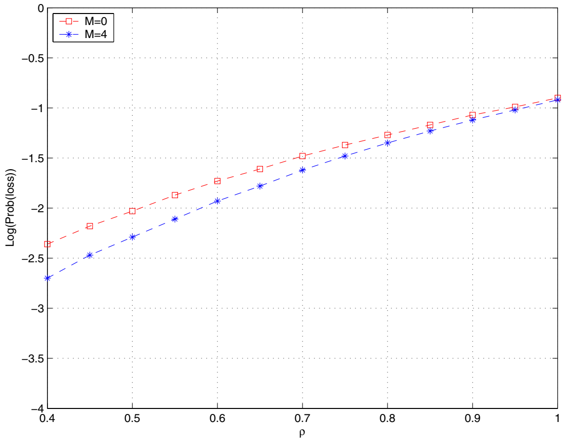


Fig. 10. Packet loss probability with bursty traffic, $N = m = 16$.

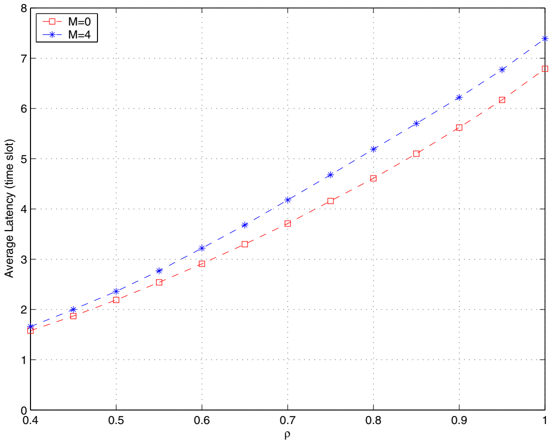


Fig. 11. Average latency with bursty traffic, $N = m = 16$.

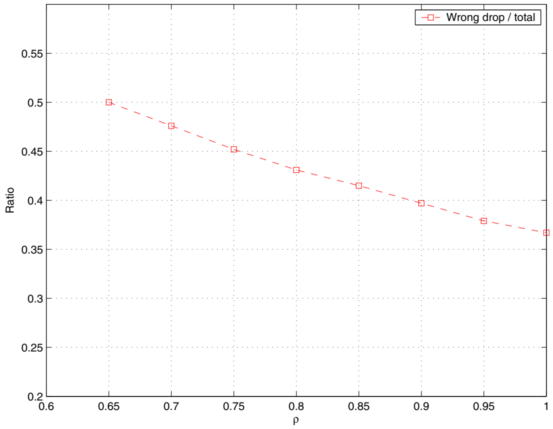


Fig. 12. Ratio of packet loss of non-priority routing to priority routing, $N = m = 16$, $M = 4$