A Modified Architecture for the "Staggering Switch"

Rajat Kumar Singh, Yatindra Nath Singh
Department of Electrical Engineering
IIT, Kanpur. India. 208016
{rajatks, ynsingh}@iitk.ac.in

Abstract

This paper proposes a modified architecture of the Staggering Switch: an electronically controlled optical packet switch. Results are compared and this modified version gives the better performance in terms of probability of loss of packets. Modification is done by using some recirculating delay lines of one packet delay at the Scheduling stage.

1. Introduction

The Staggering Switch [1] is an example of "almost-all" optical packet switch, where the data remains in the optical domain throughout the switch while the control operation of the switching is done electronically. Packet switched network are impossible without the presence of optical memory, which will be implemented by delay lines using different format [2]. Each format has its own advantages and disadvantages. The Staggering Switch does not consider recirculating loops. Here modification is done to improve the performance using some extra delay lines, each of only one packet delay, means these extra delay lines are not fully recirculating. Hence the basic principles are not changed by this modification.

2. Staggering Switch

The staggering switch implements an output-collision resolution scheme, which was based on a set of delay lines of unequal delay. Its architecture (Fig.1) consist of two stages: the scheduling stage (n x m) and the switching stage (m x n), where $m \geq n$. The scheduling stage is connected to the switching stage by m delay lines, d_i, ($1 \leq i \leq m$) providing delay of i packets. Each stage may be implemented as reconfigurably and rearrangeably nonblocking switch. The scheduling is done by retrieving the header information from all of the arriving packets and attempts to place as many of them as possible into the suitable delay line using an algorithm known as scheduling algorithm. This algorithm tries to allocate each and every incoming packet into the appropriate delay line (Fig.2), which will provide lowest delay, while considering the following two condition:

1. that no previous packet was inserted in the delay line in this time slot.
2. that no other packet to the same destination exists in the column in which the packet is to be inserted.

The whole switch is controlled electronically. The controller detects the destination of each incoming packets, and instructs the scheduling and switching stages to perform in such a way so that there will not be any collision at the switching stage.

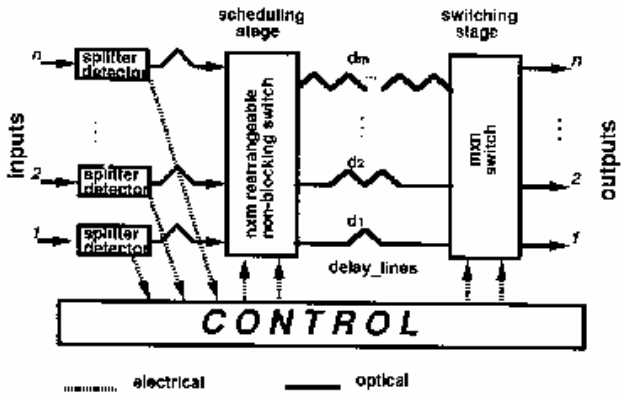

Fig.1. Architecture of the Staggering Switch.

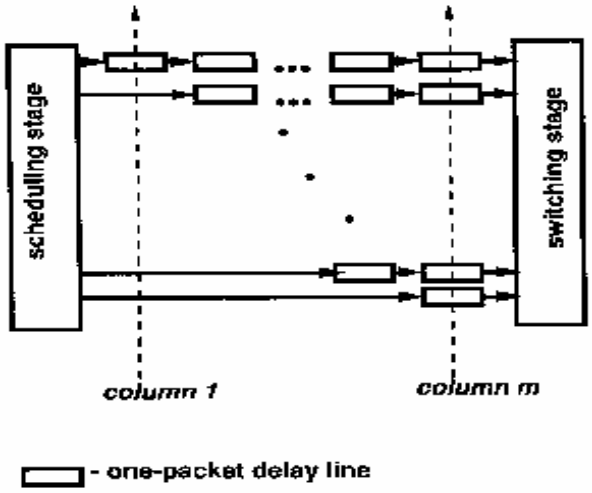

Fig.2. Definition of Scheduling algorithm.

3. Modified Architecture

Our main concern is to reduce the loss probability. For current staggering switch, the loss probability decreases when we will increase m beyond a fixed n. But here, each single increment of m increases the delay by one slot (proportional increment of delay). Means loss probability decreases on the cost of increased delay. In the modified architecture (Fig.3), we are increasing the dimension of scheduling switch to (n+D) x (m+D) using some D (>0) extra delay lines (loops), each of one packet (slot) duration, from output to input. Means whatever will be the value of D, these all extra delay lines will provide maximum of one packet delay.

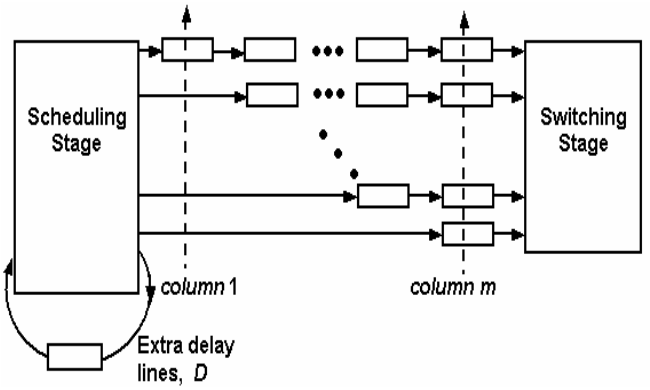

Fig.3. Modified Architecture

These extra delay lines are used to carry the packets, which are going to be lost in the current slot due to non availability of suitable delay line d_i, while considering the two conditions. In the next slot, the priority for choosing the appropriate line d_i will be given to these extra delay lines at the input of scheduling switch. The data in these extra lines will be deleted after every slot. Thus any packet will be lost permanently:

i. if such packet, after delayed by these extra lines, may not be absorbed in any of d_i delay lines in the next slot
ii. if such packet is not able to be placed in any of *m* delay lines and extra delay lines are full for that slot.

Loss probability is calculated for different values of *D* and compared to the case of *D*=0. Here, D=0 refer to the previous case (i.e. non modified).

4. Results

Simulation was done using matlab program for the scheduling algorithm and probability of loss is caculated for different no. of extra delay lines, keeping *m*=*n*. Results are compared for both cases of (NO and *D*) extra delay lines. As indicated in the plots (Fig.4-6), the loss probability decreases when *D* increases for different values of probability of arrival and a particular *n*.

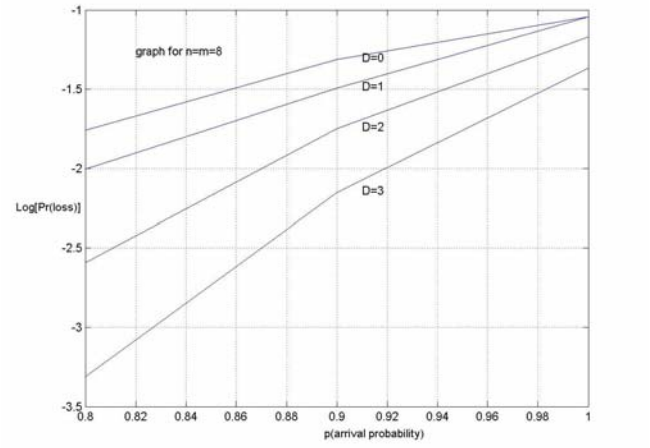

Fig.4 Loss probability Vs Arrival probability for n=m=8 and various D.

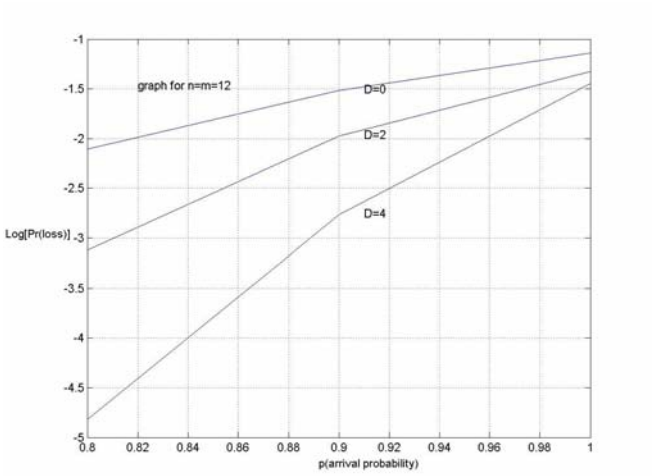

Fig.5 Loss probability Vs Arrival probability for n=m=12 and various D.

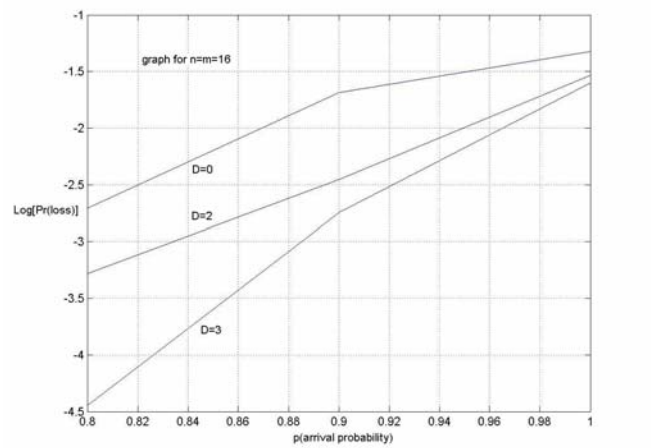

Fig.6 Loss probability Vs Arrival probability for n=m=16 and various D.

5. Conclusion

It is advantageous to add extra delay lines, instead of increasing the *m* beyond *n*, to reduce the loss probability. This will also results in the improvement of delay performance of the whole system.

Reference

[1] Z. Hass, "The Staggering Switch: An Electronically Controlled Optical Packet Switch", IEEE Journal of Lightwave Technology, pp 925-936, May/June 1993.

[2] D. K. Hunter et al., "Buffering in Optical Packet Switches", IEEE Journal of Lightwave Technology, pp 2081-2094, December 1998.