

The Odd-Even Input-Queueing ATM Switch: Performance Evaluation [†]

Christos Koliass and Leonard Kleinrock
{koliass, lk}@cs.ucla.edu
Department of Computer Science
University of California at Los Angeles
Los Angeles, CA 90024
tel.: (310) 825-1563 fax : (310) 825-2273

Abstract

This paper introduces and studies the performance of an $N \times N$ space-division, single-stage ATM switch with dual input-queueing. Each input port has two separate FIFO queues, an “odd” and an “even” queue. An incoming cell is stored at the input at either of two FIFOs according to its output port destination (output ports are also labeled as “odd” or “even”). Hence we call this scheme the Odd-Even switch. We compare the Odd-Even switch to an ordinary input-buffered switch and we find that it can achieve a considerably higher throughput. This is due to the fact that the Head-of-Line effect is less problematic under the Odd-Even scheme. We present results for various traffic models. Finally, we compare the Odd-Even strategy to the Look-ahead (input “window”) scheme.

1 Introduction

Asynchronous Transfer Mode (ATM) wide area networks are currently being deployed by telecommunications companies to support the forthcoming B-ISDN services. Various blocking and non-blocking, time-division and space-division, single stage and multi-stage ATM switches have been proposed and studied extensively, in order to provide high performance packet switching for integrated ATM transport. Some of these have been developed and tested under real-time traffic requirements and have become available as commercial products.

Switch architectures are classified [4] according to their buffering methods (i.e. input buffers, output buffers), according to their switch fabric (i.e. crossbar, Batcher-Banyan, Starlite) and according to their internal switching (non-blocking, blocking).

In this paper we study an input-buffered, non-blocking, crossbar, ATM switch with N input ports and N output

ports. Cells arriving at the input links are simultaneously and independently routed to the appropriate output ports. If there are no paths available to the output ports because of contention, incoming cells are stored at dedicated buffers at the input trunks, according to their output port destination.

The model we present for this switch ignores any hardware design and implementation details and we are only concerned with the performance evaluation of the switching system. Achieved throughput and delay are the two main performance metrics of interest. In this way, we can compare it in terms of performance with other switches that have different characteristics (i.e. arbitration policy) but with the same philosophy (i.e. input buffering).

The paper is organized as follows. Section 2 describes the Odd-Even switch* along with our modelling assumptions. In section 3, we present some performance evaluation results for the Odd-Even switch while we compare it to an ordinary input buffered switch, under various traffic conditions. Section 4 extends the comparison of the Odd-Even model to include the Look-ahead scheme. Concluding remarks are provided in section 5.

2 The Odd-Even Switch

2.1 Switch Architecture

The switch under consideration uses input-queueing to buffer a cell before it accesses the switch fabric in the case where there is no available connection to the output port and there are buffers only at the inputs. Cell arbitration among the input buffers is needed in order to control the switch and initiates the contention resolution cycle. The switch has N inputs and N outputs, therefore the switch is of size N . What is novel about this switch (figure 1)

[†]Supported by ARPA/CSTO under grant MDA-972-91-J-1011, Advanced Networking and Distributed Systems.

*We would like to thank Larry Roberts, president of *ATM Systems* for introducing the Odd-Even model to us.

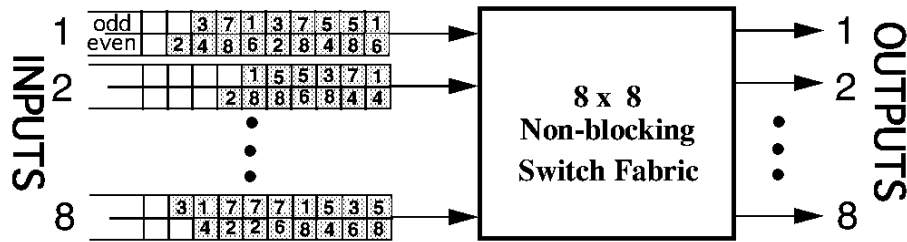


Figure 1: An 8×8 Odd-Even ATM switch (two FIFO queues per input port).

is the fact that each input is split (expanded) into two FIFO queues which we call *Odd* and *Even*. By convention, we refer to outputs numbered as 1,3,5,... as odd, and to 2,4,6,... as even ports. An incoming cell destined to an odd (even) numbered output port joins the odd (even) queue of the input port to which it was fed and awaits its turn to be switched to its output. When more than one cell is present at the heads of the queues contending for the same output address only one can access that output, and the others remain blocked until they are finally released during future time slots. This blocking phenomenon is known as *head-of-line (HOL)* blocking [2], a problem from which all input-buffered packet switches suffer. A queued cell that moves up to the head of its queue, after the previous HOL is switched, is called a *fresh* cell. Time is slotted and since cells are packets of fixed size, service time is deterministic and equal to a time slot (which is determined by the actual speed of the switch fabric) and which we assume here to be one time unit.

As mentioned above, cell arbitration is employed in order to resolve any potential contention among the input buffers. The full contention resolution process occurs at the beginning of each time slot, immediately following the cell arrivals, and consists of two (very short) rounds. Arbitration during the first round involves the HOL cells at the even input queues. In the second round cells at the HOL of the odd input queues contend for the odd output addresses. However, those input ports whose even queues could not access an output port in the first round - because they either lost the contention (and therefore were blocked) or simply did not have any HOL cells present - are allowed to participate in a contention among their odd queues in the subsequent second round. Consequently, in each time slot, an input port always gets a chance to route a cell either from an even or an odd queue, but not more than one cell is allowed to be cleared in both arbitration phases during a time slot from any given input port.

In order to maintain fairness we allow arbitration cy-

cles to interchange order in every time slot (if it is even-odd in one time slot it is odd-even in the next one), otherwise it would be as if we introduced some kind of priorities. We assume a random HOL selection policy[†] Under a balanced traffic assumption, this arbitration strategy can guarantee stability and fairness. Other possible policies are the “longest queue selection” and the “longest waiting HOL cell selection” - if there is a tie, the lowest numbered input queue is the winner.

Comparing this switch architecture to that of an ordinary simple input-buffered space-division switch (which we refer to as the “regular” switch), we recognize that the former requires additional hardware and complexity (such as an input controller and a more complex arbitrator). Clearly, the proposed architecture has no speed-up features; however, as we demonstrate in the next sections, it can increase rather significantly the output trunk utilization and decrease delays by introducing some degree of parallelism. We now proceed with the assumptions supporting our model.

2.2 Modelling Approach

The proposed switch is modelled as a multi-queue, multi-server, discrete-time queueing system. Cells arrive at the input ports at the beginning of the time slot and are queued accordingly in the Odd-Even FIFO queues. Departures are assumed to occur in a concurrent, independent fashion.

The model ignores any flow or congestion control mechanism by assuming that such control is accounted for in other parts of the network. We assume that the traffic management is efficient enough so that there is no cell loss inside the switch (though it may occur elsewhere as part of a policing mechanism) and that input buffers are infinitely large. Of course, the model can be adjusted in order to account for and study the effects of cell loss in buffer management by assuming finite size buffers. Buffers are considered as FIFO queues, thus

[†]When k cells contend for the same output at the beginning of a time slot one of the k is randomly chosen to be switched.

preventing any resequencing of cells. It is clear from the arbitration policy, that the throughput of the Odd-Even model is at least as high as that of the regular switch. A packet switch where the input traffic intensity is the same for every input link and the output address of each incoming cell is assigned with equal probability to any output port is said to be characterized as a homogeneous system.

The throughput of the switch, which is defined as the average number of successful (i.e. not blocked) cells switched per unit of time (i.e. timeslot), will be used as the main performance measure to evaluate the proposed switch. Switch throughput is then obtained as the average utilization of the N servers, where each server's utilization is obtained as the percentage of time the server is busy. As an example, in a homogeneous system, a throughput of $\gamma = 0.5$ means that on average only half of the input ports route cells in a typical time slot and that throughput gives the probability that a cell at an HOL position gets served during a time slot, provided of course that all HOL positions are occupied, i.e. a heavy traffic situation. Another interpretation is that an HOL cell has to wait, again on the average, for two time slots, to be switched, assuming again a heavy traffic case. Delay is obtained as the total time from when a cell is fed into an input port until it is switched out from an output port. Thus, delay consists of a queueing delay (waiting time) and a fixed time (equal to a time slot) in order to be switched, once it is selected. Queueing delay includes the time spent at the HOL position.

Throughput results presented as a function of N (on a logarithmic scale) refer to saturation situations. More precisely, we consider the case where input queues become saturated (when total delay becomes very large - practically infinite). If such a situation occurs, we assume that the switch operates at its attainable limit yielding a maximum throughput. Obviously, in the various cases we study below, saturation points are exhibited at different input load levels. For the balanced traffic case we expect all input queues, including odd and even queues in our model, to become saturated at the same input load. Since analysis in most cases is rather intractable, we present simulation results in order to evaluate the Odd-Even model. In our simulations we studied various switch sizes, namely, $N = 1, 2, 4, 8, 16, 32, 64, 128, 256$ and 512 , while also different traffic load situations (traffic intensities) were considered (but not included in this paper).

3 Simulation Results

3.1 Non-bursty traffic

For a quantitative evaluation and comparison of the Odd-Even switch we consider a homogeneous system. Cell arrivals to each of the N input ports are independent (thus we consider non-bursty traffic) and identically distributed according to a Bernoulli process with parameter ρ . A cell arrival at any input port occurs, with probability ρ , at the beginning of a time slot.

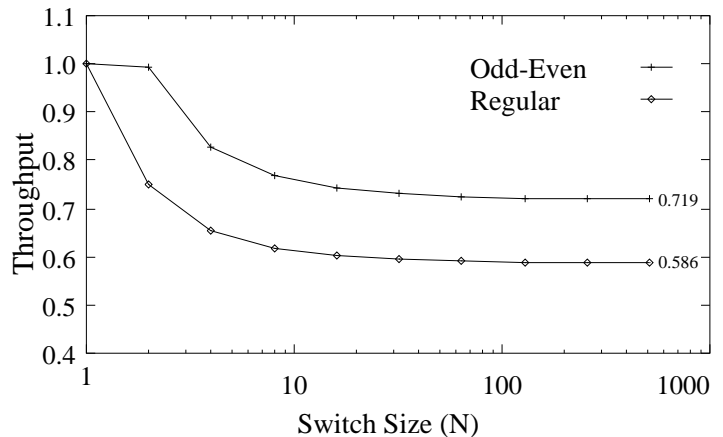


Figure 2: Odd-Even switch vs. Regular switch with Bernoulli arrivals (simulation).

Figure 2 shows the throughput of the Odd-Even switch, whose limit (for large N) is 0.719, a sizeable increase compared to that of the ordinary input buffered switch, which is known to be $2 - \sqrt{2} \approx 0.586$ [1]. In [3] we show that an approximate mathematical analysis results in a switch throughput of 0.705 for the Odd-Even model, which is a good approximation to 0.719. We see that the Odd-Even scheme outperforms considerably the regular switch (by more than 20%). Since we consider a saturation situation any input traffic where cell arrivals are i.i.d. (e.g. Poisson) gives the same throughput.

The mean waiting time for the Odd-Even model, as compared to the ordinary input-queueing switch, is shown in figure 3, as a function of ρ and for the values of $N = 2, 32$ and 512 , under a random HOL selection policy. Again, we see a significant improvement using the Odd-Even switch over the regular one.

3.2 Interrupted Bernoulli Process (IBP) traffic

As Variable Bit Rate (VBR) applications characterize a significant portion of the traffic (i.e. packetized voice and video and large data files) carried by ATM networks, we

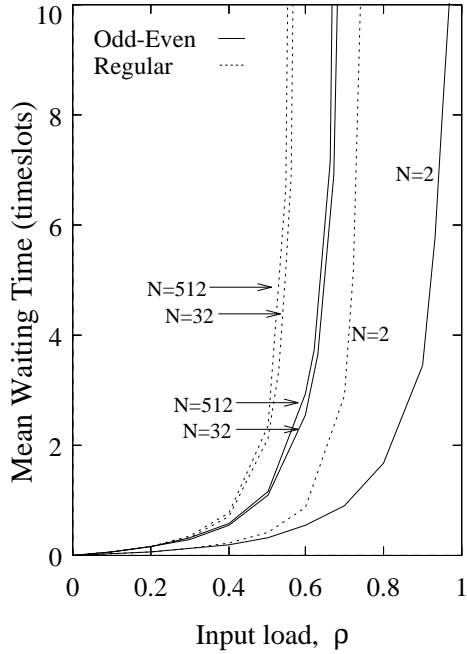


Figure 3: Mean waiting times - Bernoulli arrivals (simulation).

evaluate our model also under a bursty traffic pattern. In this model, input traffic alternates between active (busy) and silent (idle) periods, while output port addresses are still uniformly distributed. In particular, each of the inputs is described by the same On-Off model where both busy and idle periods have durations that are geometrically distributed. Cells within the same burst are destined to the same output port (i.e. cells belong to the same fragmented packet). Basically, each time slot is governed by a Bernoulli process, where given an input is in an On (busy) state, it will remain in that state (i.e. during the next time slot) with probability $1 - p$, while it will switch to the Off (idle) state with probability p . An On state corresponds to a burst being transmitted on an input link while the Off state corresponds to a silence period (figure 4).

The probability of a burst consisting of k time slots is given by:

$$B(k) = p(1 - p)^{k-1}, \quad k \geq 1 \quad (1)$$

If the input is in the Off (idle) state, where no cells arrive, it will stay in the same state with probability $1 - q$ and then the probability that an idle period lasts k time slots is simply:

$$I(k) = q(1 - q)^k, \quad k \geq 0 \quad (2)$$

which also takes into account the case where a burst can be followed immediately by another burst (i.e. two dif-

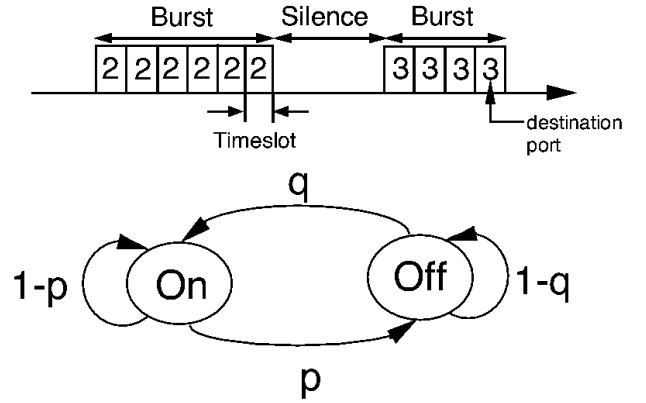


Figure 4: On(Burst)-Off(Silence) traffic model.

ferent streams being multiplexed) in which case there is no idle period ($k = 0$). The offered load can then be calculated as:

$$\rho = \frac{E[\text{busy period}]}{E[\text{busy per.}] + E[\text{idle per.}]} = \frac{q}{q + p - pq} \quad (3)$$

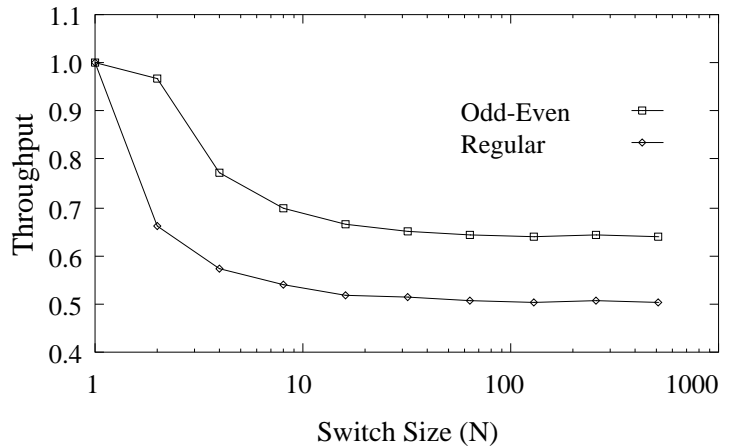


Figure 5: Odd-Even switch vs. Regular switch, IBP traffic, $p=0.05$ (simulation).

Figure 5 illustrates the saturation throughput results for $p = 0.05$, while figure 6 gives the maximum throughput for different values of p for a 512×512 switch. Saturation is obtained if we consider a heavy load situation, for instance, $\rho = 1$ (in general saturation occurs whenever input load exceeds the switch's maximum achieved throughput) and by using (3), then $q = 1$ (there are no silence periods), regardless of the value of p ($p \neq 0$). If now $p = 1$ (bursts consist of only one cell) then $q = \rho$ which then has the same effect, in terms of throughput, as the Bernoulli(ρ) case in the previous paragraph. We also notice in figure 6, which gives the asymptotic behaviour,

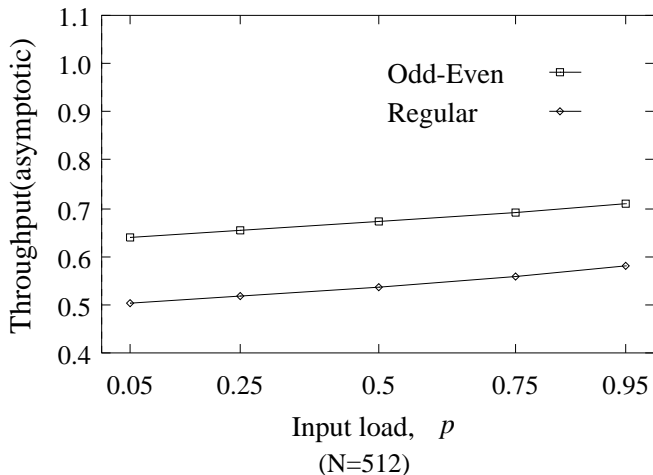


Figure 6: Odd-Even switch vs. Regular switch, IBP traffic (simulation).

that the absolute difference in throughput between the two switches remains almost constant for any value of p . This means that the percentage gain in throughput with the Odd-Even switch becomes larger as p decreases (hence traffic is burstier).

3.3 Non-uniformly distributed destinations

So far we have assumed uniformity for the output port addresses. In this section we evaluate the switch's performance under the assumption of a non-uniform, non-bursty traffic pattern. The output port mapping for each incoming cell is determined by a binomial distribution [5], where for $i = 2, \dots, N - 1$:

$$\begin{aligned}
 a_{i,j} &= Pr[\text{a cell arriving at the } i\text{-th input is} \\
 &\quad \text{destined to the } j\text{-th output}] \\
 &= \binom{N}{j} r_i^j (1 - r_i)^{N-j}, \quad j = 1, \dots, N \quad (4)
 \end{aligned}$$

and r_i is a probability associated with input i , $2 < i < N$. Note that for this binomial distribution the maximum probability occurs for $j = \lfloor N r_i \rfloor$. Now, if we choose $N - i$ to be the output for which the highest percentage of traffic arriving on i goes to (therefore it has the largest probability associated with input i), then we have

$$r_i = 1 - \frac{i}{N}, \quad i = 2, \dots, N - 1. \quad (5)$$

For inputs 1 and N the output address for $j = 1, \dots, N$ is given by a normalized Poisson-like distribution with rate r ,

$$a_{1,j} = \frac{\frac{r^{N-j}}{(N-j)!}}{\sum_{\forall j} \frac{r^{N-j}}{(N-j)!}}, \quad (6)$$

$$a_{N,j} = \frac{\frac{r^j}{j!}}{\sum_{\forall j} \frac{r^j}{j!}} \quad (7)$$

One of the advantages of using a binomial distribution for inputs $i = 2, \dots, N - 1$, is that we still retain the balanced traffic effect for the Odd and Even input queues in each input port. However, this observation does not apply for inputs 1 and N , where, depending on the value of r , there is some bias introduced against the even outputs, but as N gets very large this bias becomes negligible. Another reason for using this kind of binomial distribution is that we can have a substantial number of hot spots (instead of one or two), which is more realistic as the number of switch ports increases (and thus the number of potential connections among hosts).

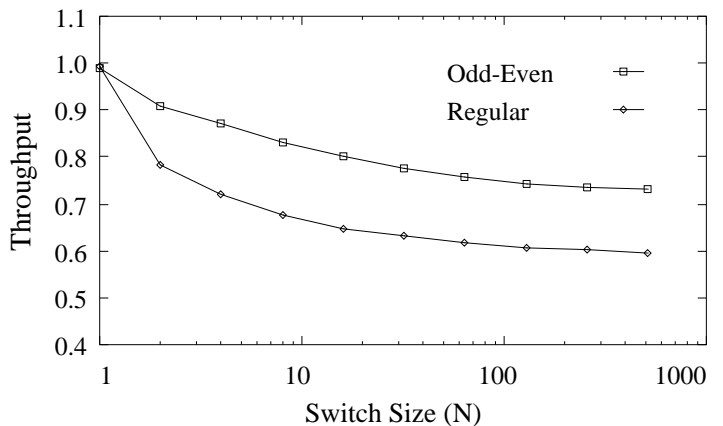


Figure 7: Odd-Even switch vs. Regular switch, non-uniform traffic (simulation).

In our simulations, we chose a value of $r=0.4$; however r has no significant relevance on estimating the throughput of the system. Using the same input-to-output address assignment functions (4)-(7) for both the Odd-Even and the Regular switch, figure 7 shows the throughput of each switch respectively.

4 The Odd-Even and the Look-ahead model

Lastly, we compare the Look-ahead scheme [1] (an input "window" policy for tackling the HOL problem) to the Odd-Even switch. In the Look-ahead strategy (also known as the bypass discipline) the arbitration and cell selection process employs w separate contention resolution rounds. During the first one all the HOL cells at the input queues contend for an output port. Those queues that lost the contention during that first round can participate in a new arbitration by having the cells right

behind the HOL cells (that were blocked) compete for outputs that are still idle (i.e. they have not received any cells during the current time slot). In the same fashion a contention cycle that is comprised of a total of w rounds, can have during the i -th round the i -th (deep in the queue) cells contend for a free output. It should be made clear that only one cell is allowed to be switched from a single input and only one cell can be served by an output port during a time slot. For $w=1$, we have the regular pure input-buffered switch.

In our comparison we consider a Look-ahead scheme with a window of $w = 2$. We feel that a window of two makes it comparable to the Odd-Even model where contention resolution also consists of two rounds (one for polling the odd and one for polling the even queues). Also, note that in both strategies only one cell can be cleared from a single input port (there is not any speed-up feature).

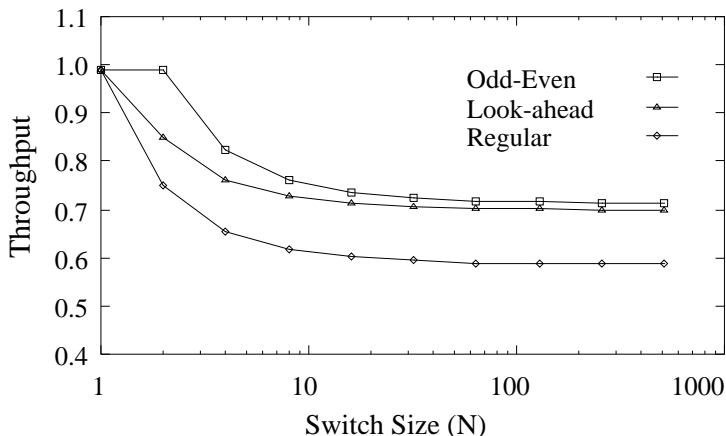


Figure 8: Odd-Even switch vs. Look-ahead($w=2$) and Regular switches, Bernoulli arrivals (simulation).

Figure 8 illustrates how the three models, namely the Odd-Even, the Look-ahead and the regular one compare, assuming Bernoulli arrivals. We see that the throughputs of the Odd-Even and the Look-ahead are very close. However as traffic becomes burstier this is not any longer true, and the Odd-Even performs much better, since a window of $w = 2$ for Look-ahead has a rather decreasing advantage (see figure 9).

5 Conclusion

In this paper we have described a new crossbar, input-buffered ATM switch, the *Odd-Even* switch, which allows incoming cells to be stored in either of the two FIFOs placed in every input port, according to their destination, namely odd or even. Its performance has been ex-

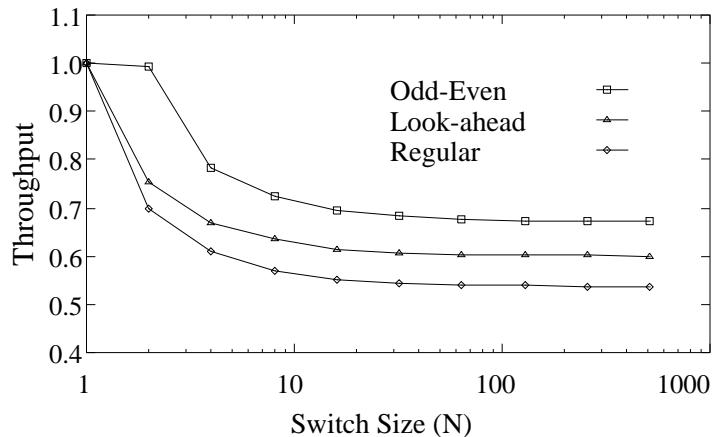


Figure 9: Odd-Even switch vs. Look-ahead($w=2$) and Regular switches, IBP traffic - $p=0.5$ (simulation).

amined rather extensively via simulation and under various conditions and assumptions i.e. balanced, bursty, non-uniform traffic and compared to the ordinary input buffered switch. Moreover, we compared the Odd-Even model against the Look-ahead scheme for which we assumed a window of $w = 2$, in order to make it comparable to the Odd-Even. In all cases, it has been demonstrated that the Odd-Even can achieve a gain of more than 20% in throughput over the regular input buffered switch, while the average delay is decreased accordingly.

References

- [1] M. G. Hluchyj and M. J. Karol, "Queueing in High-Performance Packet Switching", *IEEE J. on Selected Areas in Commun.*, vol. 6, no. 9, pp. 1587-1597, Dec. 1988.
- [2] J. Y. Hui, *Switching and Traffic Theory for Integrated Broadband Networks*, Kluwer Academic Publishers, Norwell, MA, 1991.
- [3] C. Koliass, L. Kleinrock, "Throughput Analysis of Multiple Input-Queueing in ATM Switches", *Broadband Communications '96*, Montreal, Quebec, Canada, April 1996.
- [4] R. Onvural, *Asynchronous Transfer Mode Networks: Performance Issues*, Artech House, Norwood, MA, 1994.
- [5] S.Q. Li, "Nonuniform Traffic Analysis on a Non-blocking Space-Division Packet Switch", *IEEE Trans. on Commun.*, vol. 38, no. 7, pp. 1085-1096, July 1990.