# Frame-based Matching Algorithms for Optical Switches

Yihan Li, Shivendra Panwar and H. Jonathan Chao

Electrical and Computer Engineering Department, Polytechnic University, Brooklyn, NY 11201, USA
Email: yli@photon.poly.edu, panwar@catt.poly.edu, chao@poly.edu

**Abstract— Virtual Output Queuing is widely used by fixed-length high-speed electronic switches to overcome head-of-line blocking. This is done by means of matching algorithms. These matching algorithms have typically been cell-based. That is, in every time slot, a new matching set is calculated and the switch fabric is updated to connect matched inputs and outputs. Fabric reconfiguration in an optical switch is not as fast as in an electronic switch. During reconfiguration, no data can be transferred. Given this overhead, it is not efficient to update connections between inputs and outputs for every time slot. In this paper frame-based matching algorithms for optical packet switches are presented, so that connections can be updated less frequently to reduce the bandwidth loss. The implementation complexity and performance of these schemes are studied.**

***Index Terms*— optical switching, scheduling, Virtual Output Queueing, exhaustive service, polling systems.**

## I. INTRODUCTION

IN a high-speed network, it is desirable to use optical switches as core switches. An all-optical packet switch is not feasible today due to the lack of optical devices for packet storage. In a practical hybrid switch, packet storage, processing, and arbitration can be implemented electronically, while packet switching is done by an optical switch. Switches with capacities beyond a few terabits/sec require optical interconnections between interconnect line cards and the switch fabric system. Thus, Optical/Electronic (O/E) and Electronic/Optical (E/O) conversion is required for the lines connecting line cards to the switch fabric. These conversions can be avoided by using an optical switch fabric. Also, an optical switch fabric is compact and low power as compared to an electronic switch. Furthermore, optical switches are "transparent" to the line bit rate, i.e., the switch fabric does not have to be upgraded as the line rate is increased. However, the switch fabric reconfiguration speed of an optical switch is slow. It is important to achieve high performance by designing a suitable scheduling and arbitration scheme for an optical switch. We focus on this aspect of the design of a hybrid optical switch in this paper.

Switches based on Input Queuing (IQ) are desirable for high speed switching, since the internal operation speed is only slightly higher than the input line rate. However, an Input Queuing switch has a critical drawback [1,2]: the throughput is limited to 58.6% due to the head-of-line (HOL) blocking phenomena. Output Queuing (OQ) switches have optimal delay-throughput performance for all traffic distributions, but the N-times speed-up in the fabric limits the scalability of this architecture. Virtual Output Queuing (VOQ) is used to overcome this drawback. In a VOQ switch, each input maintains N queues, one for each output. By using VOQ, no additional speedup is required and HOL blocking can be eliminated. We will therefore restrict our subsequent discussion to input queued switches with VOQs.

Considerable work has been done on scheduling algorithms for VOQ electronic switches. These algorithms match input and output ports to maximize throughput. A maximum weight matching algorithm (MWM) finds the maximum weight matching and is proved to be stable [3,4,5]. But MWM is not practical to implement in hardware due to its complexity. Maximal matching algorithms [4,6], more practical than MWM, have been proved to be stable with a speedup of 2 [7,8]. Iterative algorithms such as PIM [8] iSLIP [5,9] and DRRM [10,11], use multiple iterations to converge on a maximal matching. Exhaustive service DRRM (EDRRM) [12,13], a variation of DRRM, improves switching performance under bursty and nonuniform traffic by using the exhaustive service discipline. Birkhoff-von Neumann switches [14] and its variation [15] use multiple stages to resolve the scheduling problem. A randomized scheduling algorithm presented by Tassiulas et al [16] guarantee 100% throughput with low complexity but high delay. These matching algorithms, with the exception of EDRRM, have typically been cell-based. That is, in every time slot, a new matching set is calculated and the switch fabric is updated to connect matched inputs and outputs.

In an optical switch, switch fabric reconfiguration is much slower than in an electronic switch. Among the technologies under consideration, micro-electro-mechanical-systems (MEMS) is the most promising technology for optical cross-connect switches [17]. The reconfiguration time of a MEMS switch is currently several milliseconds, and is expected to drop to the microsecond range. This contrasts with a packet transmission time of 32ns at 10Gb/s when the packet length is 40bytes. During the reconfiguration time, no data can be transferred. Given this overhead, it is not efficient to update connections between inputs and outputs at every time slot. Some work has been done to reduce the frequency of connection updates [18,19,20]. In [18], every $K$ time slots are grouped into a frame. Contentions are resolved and a set of matchings is generated at each frame boundary for use over the next frame. In [19] a switch without overhead is emulated by accumulating a batch of configuration requests and generating a corresponding schedule for a switch with overhead. Speedup is required to compensate the overhead of switch configuration and time slots left empty by the

This work is supported in part by the New York State Center for Advanced Technology in Telecommunications (CATT), and also in part by the National Science Foundation under grants ANI0081527 and ANI0081357.

scheduling algorithm. In [20], a new method of using large frame sizes is proposed for switching variable sized packets over a crossbar switch to minimize reconfiguration frequency of the switch fabric. This method provides delay bounds.

In this paper we present two classes of matching algorithms, fixed size synchronous frame matching and variable-size asynchronous frame matching, where connections can be updated less frequently to reduce the bandwidth lost. All these matching schemes, are suited to bursty traffic because of the frame mechanisms ability to absorb bursts.

In a fixed size synchronous ~~frame-based~~ matching scheme, $K$ time slots ~~are~~ grouped into a frame. One matching set is computed for each frame, and the switch fabric is updated only on frame boundaries. Since the time to compute the new matching set is not limited to one time slot, the complexity of a frame-based matching scheme can be higher than that of a cell-based matching scheme (where $K$ is 1). Three frame-based matching algorithms are introduced in this paper. *Frame-based MWM* is proved to be stable under any admissible Bernoulli i.i.d. traffic. However, the complexity for each arbitration is O($N^3$). *Frame-based maximal weight matching* can be used to approximate a frame-based MWM. The throughput is 100% under uniform traffic and close to 100% under most nonuniform traffic patterns. The delay performance under uniform traffic is presented for different switch and frame sizes. The complexity of the frame-based maximal weight match scheme is O(NlgN). *Frame-based multiple iteration weighted matching* can be implemented in a distributed manner and the complexity can be further reduced to O($\lg^2$N) for lgN iterations. Simulation results show that with lgN iterations, frame-based multiple iterative weighted matching can achieve performance similar to that of frame-based maximal weight matching.

In order to further reduce the overhead, an asynchronous variable-size frame matching scheme that only runs arbitration and updates connections when necessary, i.e., asynchronously, is presented. Exhaustive Service Dual Round-Robin Matching (EDRRM) [17,18] can be used in this manner with a small modification. Simulated and analytical performance of EDRRM that includes the impact of the reconfiguration time is presented.

In section II we discuss fixed size synchronous frame matching schemes and their performances. In section III, variable-size asynchronous frame matching is introduced.

## II. Fixed Size Synchronous Frame-Based Matching and Connection Updating

In a fixed size synchronous frame-based matching and connection updating switch, every $K$ time slots are grouped into a frame, and switch fabric is only updated on frame boundaries. Since the time to compute the new matching set is not limited to one time slot, the complexity of a frame-based matching scheme can be higher than that of a cell-based matching scheme (when $K$ is 1).

If it takes $m$ time slots to compute a new match, a frame-based matching scheme starts to perform arbitration for the next frame $m$ time slots ahead by using the VOQ status at that time, and updating the matching set for cell transmission at the beginning of the next frame. An example of how a frame-based matching scheme works is shown in Fig. 1. The connection reconfiguration time $L$ is included in each frame.

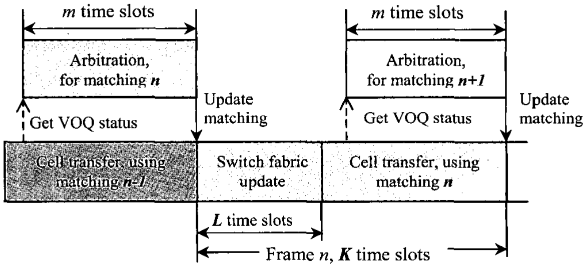


Fig. 1. An example of a fixed size synchronous frame-based matching scheme.

In this section we present three kinds of fixed size synchronous frame-based matching schemes and simulation results for performance. For a fixed size synchronous frame-based matching scheme, the overhead introduced by reconfiguration time $L$ is $L/K$. The bandwidth lost can be simply compensated by a speed up $K/(K-L)$, where $K$ is typically much larger than $L$.

### A. Frame-based Maximum Weight Matching

Among all cell-based matching schemes, Maximum Weight Matching (MWM) has been proved to be stable for any admissible traffic and has the best delay performance. A frame-based MWM does not find a maximum weight match for every time slot. Instead, it computes one matching for all time slots in a frame, which is a maximum weight matching based on the weight $m$ time slots ahead of the frame boundary. As a corollary to a theorem from [21], we will now show that a frame-based MWM is stable under any admissible Bernoulli i.i.d. traffic.

**Theorem** (Theorem 1, [21]) Let $W^*(t)$ denote the weight of maximum weight matching scheduling at time $t$, with respect to switch state $Q(t)$. Let $W^B(t)$ denotes the weight of a scheduling algorithm $B$ at time $t$. Further, $B$ has property that,

$$W^B(t) \geq W^*(t) - f(W^*(t)), \forall t,$$

where $f(.)$ is a sub-linear function. Then, the scheduling algorithm B is stable under any admissible Bernoulli i.i.d. input traffic.

**Corollary** A frame-based MWM is stable under any admissible Bernoulli i.i.d. traffic.

***Proof.*** Suppose $M^*(t)$ is the maximum weight match at time $t$ with weight $W^*(t)$, and $M(t)$ is the match used in a frame-based MWM at time $t$ with weight $W(t)$. The weight of a matching can increase or decrease at most by $N$ between two consecutive time slots. Assume $t_0$ is the beginning time of a frame, and $t \in [t_0, t_0+K]$. Then $M(t)=M^*(t_0-m)$ and $W(t) \geq W^*(t_0-m)-(K+m)N$. Since $W^*(t-1) \geq W^*(t)-N$, $W^*(t_0-m) \geq W^*(t_0)-(K+m)N$. Therefore we have $W(t) \geq W^*(t)-2(K+m)N$. Applying the above theorem, this proves that frame-based MWM is stable under any admissible Bernoulli i.i.d. input traffic. ∎

The complexity of MWM is O($N^3$) [22], which is not practical to implement, even with the relaxed timing available under frame-based matching. For a practical optical switch simpler matching algorithms are needed. Two frame-based matching algorithms that approximate frame-based MWM are discussed next.

### *B. Frame-based Maximal Weight Matching*

MWM always find the matching set with the largest weight among *N!* possible input-output matching sets. One possible way to approximate MWM is to use a maximal weight matching algorithm. For an optical switch, a Frame-based Maximal Weight Matching algorithm, which finds a new matching set at each frame boundary, can be used to approximate Frame-based MWM with reduced complexity. The most straightforward maximal weight matching algorithm is to sort all $N^2$ VOQs by weight (e.g. VOQ length) and always select the VOQs with the largest weights for service. The complexity of this sorting operation is $O(N^2 lgN)$. Sorting VOQs in a distributed manner, at each input line card, can further reduce the complexity to O(NlgN). The details of the algorithm are as follows. Suppose the algorithm takes *m* time slots to compute a new matching set, and the frame size is *K* time slots, as shown in Fig. 1.

***Step 1:*** At *m* time slots before a new frame starts, collect weights for all VOQs. At each input, sort all *N* VOQs by their weights in decreasing order. Let *h=N*.

***Step 2:*** Consider the *h* VOQs at the top of the *h* sorting lists. Select the one with the largest weight and match the corresponding input and output. Delete the sorting list of the corresponding input. Delete all VOQs destined to the corresponding output from the other sorting lists.

***Step 3:*** *h* = *h* -1. If *h* >0, go to step 2; otherwise, stop.

***Step 4:*** Update the matching set at the boundary of a new frame.

The complexity of step 1 is O(lgN). Step 2 also takes O(lgN) steps, and at most *N* executions are needed, which leads to a complexity O(NlgN).

### *C. Frame-based Multiple Iteration Weighted Matching*

In a cell-based electronic switch, multiple iterative matching schemes, such as PIM, iSLIP, DRRM, use multiple iterations to converge on a maximal size matching. Theoretically, up to N iterations leads to a maximal size matching. Similarly, multiple iterative weighted matching scheme, such as Longest Queue First (iLQF) and Oldest Queue First (iOQF) [5], converge on a maximal weight matching within N iterations.

In an optical switch, a frame-based multiple iteration weighted matching can be used to converge on a frame-based maximal weight matching. Simulation results show that by using *lgN* iterations the scheme can achieve almost the same performance as that of a frame-based maximal weight matching, which reduces the complexity to $O(lg^2N)$, and can be implemented in a distributed manner.

A frame-based multiple iterative weighted matching scheme works as follows in every iteration.

***Step 1:*** Each unmatched input sends requests for all nonempty VOQs along with VOQ weights.

***Step 2:*** If an unmatched output receives any request, it selects the request with the largest weight and sends a grant to the corresponding input. Ties are broken randomly.

***Step 3:*** If an unmatched input receives multiple grants, it accepts the grant corresponding to the largest weight. Ties are broken randomly.

The complexity of one iteration is O(lgN). According to simulations, when the scheme is run for *lgN* iterations, the throughput and delay performances converge and are almost as the frame-based maximal weight matching. Therefore, the complexity of this scheme is as low as $O(lg^2N)$.

### *D. Simulated Performance*

#### *1) Throughput*

When L is set to 0, our simulation results show that the throughput of a frame-based Maximal Weight Matching algorithm under uniform traffic is 100%, and is close to 100% under all nonuniform traffic patterns considered in [12] except the *diagonal traffic* pattern, no matter what the frame size is. According to previous studies, the diagonal traffic pattern, in which for an input *i* a ratio *p* of arrivals is destined to output *i* and *1-p* to output *(i+1)modN,* is an extremely unbalanced nonuniform pattern. Switches usually have worse performance under diagonal traffic than under other more balanced traffic patterns. According to simulation results, the throughput of frame-based maximal weight matching under diagonal traffic is always higher than 88%. Simulation results show that under the hot-spot traffic pattern [23], the hot-spot throughput is maintained at 100%.

When L is considered and is larger than 0, the simulated throughput for all schemes under uniform traffic is *(1-L/K)* as expected.

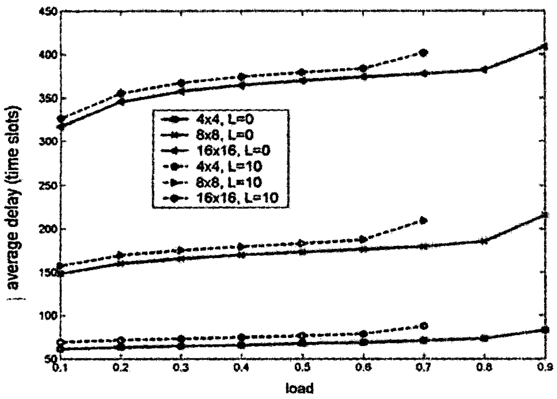
The simulated throughput performance of the frame-base multiple iteration weighted matching was found to be almost the same as that of a frame-based maximal weight matching.

#### *2) Average delay*

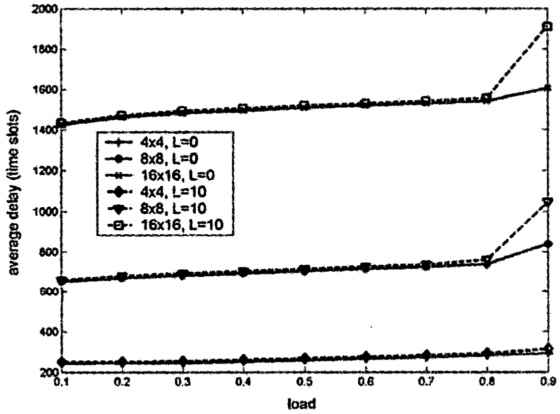
In the simulations presented here we assume that *m=1*. To allow comparison with cell-based switches we assume a fixed packet size. The average delay is measured in time slots, where one time slot is the time to transfer one fixed-size packet. Since the performance of frame-based multiple iteration weighted matching is quite close to the performance of the frame-based maximal weight matching, from this point we will only show simulation results of frame-based maximal weight matching to save space.

Fig. 2 shows the average delay of frame-based maximal weight matching for different switch sizes when the reconfiguration time *L* is 0 or 10, and the frame length *K* is 50 or 200 time slots. It shows that under light load, the average delay with nonzero *L* is close to the sum of L and the average delay when *L* is zero. Note that the average delay is close to *NK/2.*This corresponds to the intuition that under a moderately loaded regime the average arriving packet has to wait for N/2 frames before its own VOQ is served.

The delay, measured at over a thousand time slots for *L=10* and *K=200*, appears to be discouraging at first sight. However, if we assume a future reduction in switch reconfiguration time to 1μs, and with a packet length of 100ns, a delay of 1000 time slots is only 100μs. Given a switch size *N*, reconfiguration time *L*, and delay requirement, the corresponding frame size and internal speedup can then be determined.



(a) Frame size K=50



(b) Frame size *K*=200

Fig. 2. The average cell delay of frame-based maximal weight matching switch under uniform traffic with different switch sizes and reconfiguration times.

## III. Asynchronous Variable-Size Frame Matching and Connection Updating

In fixed size synchronous frame-based matching schemes, the switch fabric is updated for each frame instead of each time slot to reduce the bandwidth lost. If connections are updated asynchronously only when necessary, the overhead can be reduced, especially under heavy load. For example, this is feasible in MEMS based optical switches where it is possible to reconfigure a subset of inputs and outputs while the other input-output pairs continue to switch packets. While this allows a new class of matching algorithms, we will only consider here the Exhaustive Service Dual Round-Robin Matching (EDRRM) scheme [12,13].

For an optical switch, EDRRM can be modified so that arbitration will only be done when necessary, not in every time slot. In the original EDRRM, arbitration is done by input arbiters and output arbiters based on the round-robin service discipline. When an output is matched to an input, this output is *locked* by the input. When the VOQ under service is emptied, the corresponding input sends a message to the locked output to *release* it. The released input and output increment their arbiter pointers by one location, so that the VOQ just being served will have the lowest priority in the next arbitration. A locked output cannot grant any request from other inputs. An input sends requests to outputs if, and only if, it is not matched. When a match is found, only this new connection will be updated which leads to a reconfiguration time. All other connections continue uninterrupted. In order to reduce the frequency of switch fabric reconfigurations, EDRRM is modified as follows. When a VOQ is completely served and the corresponding input or output has not successfully found a new match, the connection for this VOQ will not be disconnected. In this way, new arrivals to this VOQ can still be transferred before the switch fabric is updated.

A detailed description of the two step EDRRM algorithm in an optical switch follows:

***Step 1*: *Request.*** Each unmatched input moves its pointer to the first nonempty VOQ in a fixed round-robin order, starting from the current position of the pointer, and sends a request to the output corresponding to the VOQ. The pointer of the input arbiter is incremented by one location beyond the selected output if the request is not granted in Step 2, or if the request is granted and after one cell is served this VOQ becomes empty. Otherwise, the pointer remains at that (nonempty) VOQ.

***Step2: Grant.*** If an unmatched (not locked) output receives one or more requests, it chooses the one that appears next in a fixed round-robin schedule starting from the current position of the pointer. The pointer is moved to this position. The output notifies each requesting input whether or not its request was granted. The pointer of the output arbiter remains at the granted input. The output is locked by the selected input. If there are no requests, the pointer remains where it is.

To further reduce the bandwidth overhead, one possible variation of EDRRM is to start searching for the next match when the number of cells waiting in the VOQ under service drops below a threshold. Additionally, since the arbitration time is not necessarily limited to one time slot, a matching scheme of higher complexity can be used to improve the performance. We will investigate these in future work.

### A. *Simulated Performance*

#### 1) *Performance When L=0*

When the reconfiguration time *L* is not introduced, the EDRRM in an optical switch has similar performance as that of the original EDRRM [12].

### 2) *Performance When L>0*

When the reconfiguration time $L$ is larger than 0 in an optical switch, throughput is expected to reduce. Simulated throughput with different values of non-zero $L$ and different switch size is shown in Table II. It shows that throughput is relatively insensitive to $L$.

The average cell delay of EDRRM in an optical switch is shown in Fig. 4 with different switch sizes when $L$=10, and in Fig. 5 with different values of $L$ for a 16x16 switch. Note that variable frame schemes have better delay performance than fixed frame schemes under low and moderate loads. This is due to the fact that under low loads, fixed frames are often not filled, leading to unnecessary additional delay. In variable-sized frame matching schemes, the frame size adapts to VOQ loading.

TABLE I THROUGHPUT OF EDRRM OPTICAL SWITCH

| L(time slots) | 0 | 10 | 100 | 1000 |
|---|---|---|---|---|
| 4x4 switch | 0.9926 | 0.9913 | 0.9760 | 0.8946 |
| 8x8 switch | 0.9408 | 0.9399 | 0.9383 | 0.9232 |
| 16x16 switch | 0.9680 | 0.9673 | 0.9587 | 0.9279 |
| 32x32 switch | 0.9794 | 0.9768 | 0.9636 | 0.9153 |

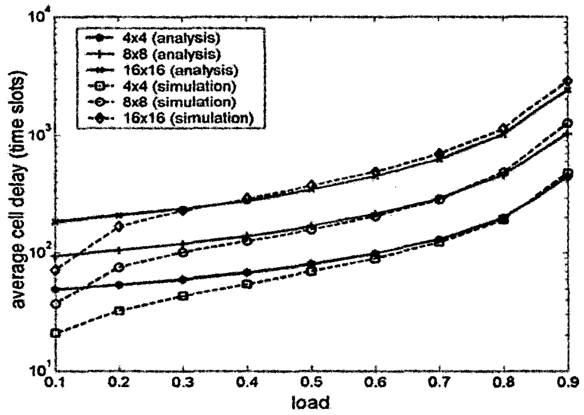


Fig. 4. The average cell delay of EDRRM switch under uniform traffic for different switch sizes when $L$=10.

## B. *Performance Analysis*

The delay performance of an EDRRM switch under uniform traffic can be analyzed by using an exhaustive random polling system model. This is used to predict the performance of switches too large to be simulated within a reasonable run time. The performance analysis with L is similar to the analysis without introducing L, details for which can be found in [13]. In this paper we only show the final expressions.

### 1) *Switch Over Time and Average Delay*

The *switch over time* is the sum of $L$ and $B$, which is the time taken for the input to switch from one VOQ after service completion to another VOQ for a new service period.

The final expressions for the average and second moment of switch over time $S$ are shown as follows.

$\mathrm{E(S) = E(B) + L}$, and $\mathrm{E(S^2) = E(B^2) + 2LE(B) + L^2}$,

where

$$\mathrm{E(B)} = \frac{1-q}{Q} + \frac{\rho^{2(N-1)}}{1-\rho^{N-1}},$$

$$\mathrm{E(B^2)} = \frac{1-q}{Q}\left[\frac{2(1-Q)}{Q}+1\right] + \frac{\rho^{2(N-1)}}{1-\rho^{N-1}}\left[\frac{2\rho^{N-1}}{1-\rho^{N-1}} + \frac{2(1-q)}{Q}+1\right],$$

where $\rho$ is the arrival rate,

$$q = \frac{1}{w(1-\rho^{N-1})(N-1)}\left[1-(1-w)(1-w(1-\rho))^{N-1} + \frac{(1-w(1-\rho))^N - 1}{N(1-\rho)}\right],$$

$$Q = \frac{1}{Nw}\left[1-(1-w)(1-w(1-\rho))^{N-1}\right], \text{ and}$$

$$w = \frac{1}{N}\left[1-\left(1-\frac{\rho}{N}\right)^N\right].$$

In [24] the delay of a random polling system is analyzed. For a fully symmetric system, using the notation in [24], the average delay for a cell is described as

$$\mathrm{E(T)} = \frac{1}{2}\left[\frac{\delta^2}{r} + \frac{\sigma^2}{(1-N\mu)\mu} + \frac{Nr(1-\mu)}{1-N\mu} + \frac{(N-1)r}{1-N\mu}\right],$$

where $\mu$ is the arrival rate for one VOQ, $\sigma^2$ is the variance of the arrival process for one VOQ, and $r=E(S)$, $\delta^2=Var(S)=E(S^2)-E^2(S)$. For each VOQ, under i.i.d. Bernoulli traffic, $\mu = \frac{\rho}{N}$ and $\sigma^2 = \frac{\rho}{N}$.

Fig. 4 and Fig. 5 show the analysis results of the average delay E(T) with different switch sizes and reconfiguration times, respectively, comparing to simulation results. Under heavy load, analysis results are quite close to simulation delays. The overestimation of delay under light loads can be ascribed to the fact that the polling model allows wasteful switching to empty VOQs, which is not present in the real system. As reconfiguration time L increases, analysis results approximate simulation results better.

### 2) *When N is large*

When $N$ goes to infinity, the average switch over time and its second moment converge to a limit for $\rho$<1. The average delay is a function of $N$ and $\rho$. It always has a finite value and is linear in $N$ when $N$ is large for all $\rho$<1, as shown below.

$$\lim_{N\to\infty} \mathrm{E(S)} = \frac{1-e^{-\rho}}{1-e^{-(1-\rho)(1-e^{-\rho})}} - 1 + \mathrm{L},$$

$$\mathrm{E(T)} \xrightarrow{N\to\infty} \mathrm{E(S)}\frac{N-\rho}{1-\rho} + \frac{2-\rho}{2(1-\rho)} \sim \mathrm{E(S)}\frac{N}{1-\rho}.$$

Fig. 6 shows the calculated average delay *E(T)* of four switches of large size when the reconfiguration time *L* is 0 and 10.

## IV. Conclusion

In an optical packet switch, switch fabric reconfiguration is much slower than in an electronic switch. New matching schemes are needed to reduce the bandwidth overhead thus introduced. In this paper synchronous fixed size and asynchronous variable size frame-based matching algorithms are presented. The latter is an entirely new concept wiich we believe merits further investigation. Throughput and average delay performance of these algorithms are studied by simulation and analysis.

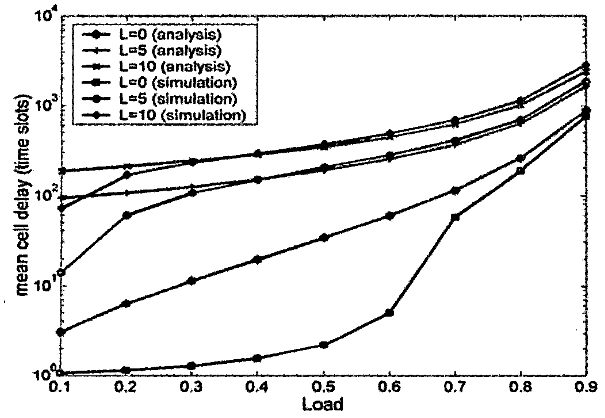


Fig. 5. The average cell delay of a 16x16 EDRRM switch under uniform traffic with varying reconfiguration times *L*.

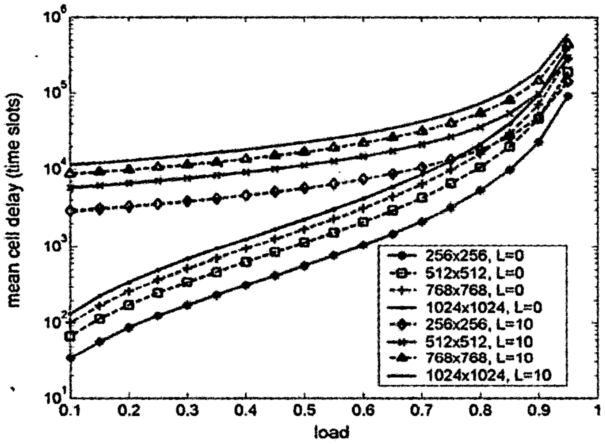


Fig. 6. The average cell delay of EDRRM switches with large switch sizes under uniform traffic.

## REFERENCES

[1] M. J. Karol, M. Hluchyj, and S. Morgan, "Input versus output queuing on a space-division packet switch," IEEE Trans. on Communications, vol.35, pp. 1347-1356, 1987.

[2] L. Tassiulas, A. Ephremides, "Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks," IEEE Trans. Automatic Control, Vol. 37, No. 2, pp. 1936-1949.

[3] N. McKeown, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," IEEE INFOCOM'96, pp. 296-302.

[4] N. McKeown, "Scheduling algorithms for input-queued cell switches," Ph.D. Thesis, UC Berkeley, May 1995.

[5] A. Mekkittikul and N. McKeown, "A practical scheduling algorithm to achieve 100% throughput in input-queued switches," IEEE INFOCOM 98, Vol 2, pp. 792-799, April 1998.

[6] J. Dai, B. Prabhakar, "The throughput of data switches with and without speedup," IEEE INFOCOM 2000, Tel Aviv, Israel, March 2000.

[7] M. Ajmone, E. Leonardi, M. Mellia, F. Neri, "On the stability of Input-buffer cell switches with speed-up," IEEE INFOCOM 2000, Tel Aviv, Israel, March 2000.

[8] T. E. Anderson, S. S. Owicki, J. B. Saxe and C. P. Thacker, "High speed switch scheduling for local area networks," ACM Trans. on Computer Systems, vol. 11, No. 4, pp. 319-352, Nov. 1993.

[9] N. McKeown, "The iSLIP scheduling algorithm for Input-Queued switches," IEEE/ACM Trans. Networking, vol. 7, pp. 188-201, April 1999.

[10] H. J. Chao, "Saturn: a terabit packet switch using Dual Round-Robin", IEEE Communication Magazine, vol. 38 12, pp. 78-84, Dec. 2000.

[11] Y. Li, S. Panwar, H. J. Chao, "On the performance of a Dual Round-Robin switch," IEEE INFOCOM 2001, vol. 3, pp. 1688-1697, April 2001.

[12] Y. Li, S. Panwar, H. J. Chao, "The Dual Round-Robin Matching switch with exhaustive service," 2002 Workshop on High Performance Switching and Routing (HPSR 2002)}, pp. 58-63, May 2002.

[13] Y. Li, S. Panwar, H. J. Chao, "Performance Analysis of a Dual Round Robin Matching Switch with Exhaustive Service," IEEE GLOBECOM 2002.

[14] C-S. Chang, D. Lee and Y. Jou,, "Load balanced Birkhoff-von Neumann switches, part I: one-stage buffering ," Special issue of Computer Communications on "Current Issues in Terabit Switching," 2001.

[15] I. Keslassy, N. McKeown, "Maintaining packet order in two-stage switches", IEEE INFOCOM 2002, vol.2, New York, 2002, pp. 1032-1041.

[16] L. Tassiulas, "Linear complexity algorithms for maximum throughput in radio networks and input queued switches," IEEE INFOCOM'98, vol.2, New York, 1998, pp.533-539.

[17] C. Pu, S. Lee, S. Park, P. Chu, I. Brener, "MEMS for optical communication: present and future," Proceedings of SPIE, vol. 4870, pp. 84-92, 2002.

[18] A. Bianco, M. Franceschinis, S. Ghisoolfi, A. M. Hill, E. Leonardi, F. Neri, R. Webb, "Frame-based Matching algorithms for Input-Queued switches," HPSR 2002, pp. 69-76.

[19] B. Towles, W. Dally, "Guaranteed scheduling for switches with configuration overhead," IEEE INFOCOM 2002, vol. 1, pp. 342-351.

[20] D. Shah, M. Kopikare, "Delay bounds for approximate maximum weight matching algorithms for input queued switches", IEEE INFOCOM 2002, New York, 2002, pp. 1024-1031.

[21] K. Kar, T.V. Lakshman, D. Stiliadis, L. Tassiulas, "Reduced complexity input buffered switches," Proceedings of Hot Interconnects VIII, 2000.

[22] R.E. Tarjan, *Data structures and network algorithms*, Society for Industrial and Applied Mathematics, Pennsylvania, Nov. 1983.

[23] G. F. Pfister, " 'Hot Spot' contention and combining in multistage interconnection networks," IEEE Trans. on Computers, vol. C-34, No. 10, pp. 943-948, Oct. 1985.

[24] L. Kleinrock, H. Levy, "The analysis of random polling systems," Operations Research, Vol.36, No.5 (September-October), pp. 716-732, 1988.