The Case of a Precognition Optical Packet Switch

Stephen Suryaputra[1], Joseph D. Touch[1], Joseph Bannister[2]

***Abstract*—This paper describes simulation analysis of an optical packet switch that has a limited capability to "see" packets that have not yet arrived at the input port. The goal of the effort is to design a variable-length optical packet switch without random access buffering. Relying on this future information, the switch tries to maximize the number of bytes switched. While we found that this optimization does not improve the throughput significantly, we are currently studying improvements of this architecture that hold greater promise.**

Keywords-optical; switch; packet; throughput; variable-length; future; look-ahead

I. INTRODUCTION

On account of the difficulty of implementing random-access storage of optical signals, optical switches have limited buffering capability. Buffering allows electronic switches to avoid dropping. Using virtual output queuing (VOQ) [3], [4], electronic switches can even avoid head-of-the-line blocking (HOL) and achieve 100% throughput. The challenge for optical switches is how to achieve comparable throughput. Specifically, we want to know by how much the effect of dropping packets can be minimized if the switch looks beyond the packet at the head of the line. We evaluate the throughput of an optical switch that exhaustively searches for the optimal packet to admit by looking a finite time into the future. The evaluation is done using simulation of a diversity of traffic patterns.

II. PRECOGNITION OPTICAL SWITCH

We assume a network in which only end stations do E/O (electrooptic) and O/E (optoelectronic) conversion. They also determine the output port of each incoming packet (*i.e.*, source routing) and notify the switch when the packet arrives at the input port of the yuuswitch. This allows us to focus solely on the switching function at intermediate switches. There is also a fiber delay line (FDL) at the ingress of each port of the switch. This delay increases the time difference between when the previous node puts the packet on the link and when the packet arrives at the switch. We refer to this time difference as *the look-ahead time*. The switch collects information about all the packets in the look-ahead time, and picks a packet among multiple packets contending for the same output port. It then connects input and output ports when the chosen packet arrives at the switch. Anthropomorphizing the switch's ability to sense in advance information about all packets in a time window, we call this a *precognition switch*. It works on variable-length packets.

Without buffering, a switch needs to operate asynchronously, *i.e.* it makes decisions based on the first packet that arrives at each port. But, if the switch sees only the first packet on the line, it does not realize that there is potential contention and will not consider other packets. If traffic has exponentially distributed packet lengths and interarrival times, then the output port reduces to an M/M/1/1 queue because packets that arrive while the first packet is in service are dropped.

In order not to degenerate into a collection of M/M/1/1 queues, the switch needs to be aware of future packets. While it is collecting information about future packets, it seems beneficial to look beyond the first packets on different input ports. We want to know if this improves throughput, because look-ahead in a linear FDL may be achievable in optics (and because random-access buffering, *e.g.,* for VOQ, is not). In order to characterize the performance of this switch, we simulate a precognition switch that does an exhaustive search for the right packet to admit at each port based on the maximum number of bytes accepted during the look-ahead time. We also develop some baselines to compare to the achieved throughput.

III. PRIOR WORK

Electronic switches have been extensively studied. HOL blocking (Figure 1) limits the throughput of input buffered switches. Karol *et al*. showed that a packet switch with unlimited FIFO input buffers has a saturation throughput of 59% [1]. They also showed that at 100% input load the throughput increases to 63% when cells that are not selected by the scheduling algorithm are dropped rather than buffered. Fuhrman later showed that

[1]The authors are with USC/ISI, Marina del Rey, CA 90292, with emails surya@isi.edu and touch@isi.edu.
[2]The author is with the Aerospace Corp., El Segundo, CA 90245 with email joseph.a.bannister@aero.org
This material is based upon work supported by the U.S. Air Force, MILSATCOM Systems Wing SMC/MCX under the National Science Foundation Grant No. CNS-0626788. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the Air Force.

the throughput is reduced even more when the input traffic has variable-length packets [2].

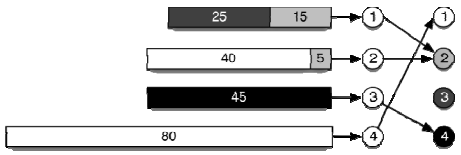

Figure 1: Head-of-the-line (HOL) blocking. Input port 1 and 2 are contending for output port 2. If the switch picks input port 2, then the second packet at input port 1 is HOL blocked.

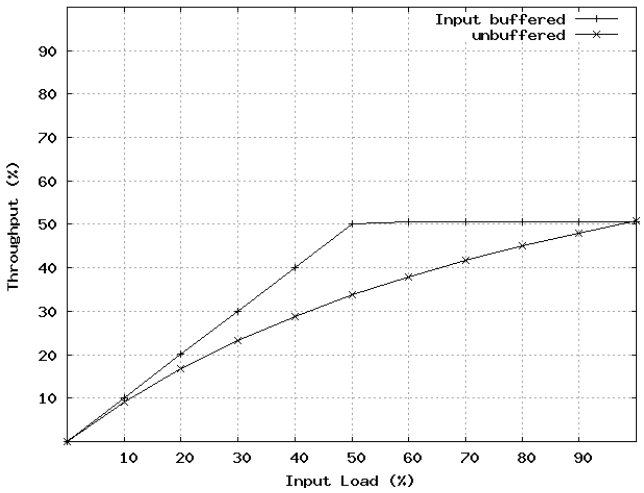

Figure 2: Simulated throughput of 32x32 switches for Poisson arrival, uniform switching matrix and exponentially distributed packet length.

We cannot use Karol's result as our baseline because the precognition switch operates on variable length packets. We also cannot compare with Furhman's result because he did not analyze the unbuffered case. Thus we still need to develop some baselines beyond these key results. These baselines are obtained through simulation. Figure 2 shows the throughput of both the unbuffered and input-buffered variable-length switches. The input-buffered switch has infinite buffers. The throughput reduction in the unbuffered case is due to random dropping.

Optical packet switches have also been studied. There are several approaches for contention resolution: buffering using fiber delay lines (FDL), deflection routing, and wavelength conversion [7]. Both FDL buffering and deflection routing are used by the staggering switch [9]. With multiple FDLs of different lengths and fixed size cell inputs, this switch achieves good performance. Our switch is different in three ways: 1) the inputs are variable-length packets, 2) simpler architecture with a single length FDLs, and 3) it operates asynchronously. Furthermore, it looks deep into future arrivals.

The staggering switch is also evaluated for bursty traffic where cells are transmitted back-to-back. The author reported a significant drop in its performance. These bursts can be viewed as variable length packets and the performance drop motivates the idea of our switch.

Optical burst switching (OBS) is another relevant approach [8]. Our approach to have a time offset between the arrival of the packets and the arrival of the control signals is similar to OBS. OBS uses the offset to avoid buffering (by deflecting the bursts) and reserve switch connections along the path. We, however, use the time offset to figure out which packets to accept. Another difference is: OBS focuses on the performance of an optical network with multiple wavelengths, while we are focusing on a single-wavelength switch performance.

To the best of our knowledge, there is not any existing result of an exhaustive search for the maximum number of bytes transferred on a bufferless optical switch with variable length packets.

IV. THE NEED FOR EXHAUSTIVE SEARCH

Using variable-length packets means that picking one packet can have the consequence of dropping a longer packet, thus lowering the throughput. Picking a longer packet does not necessarily make the throughput higher because it can potentially drop a lot of small packets and the total length of those smaller ones can be bigger than the long packet. We illustrate this problem in Figure 3.

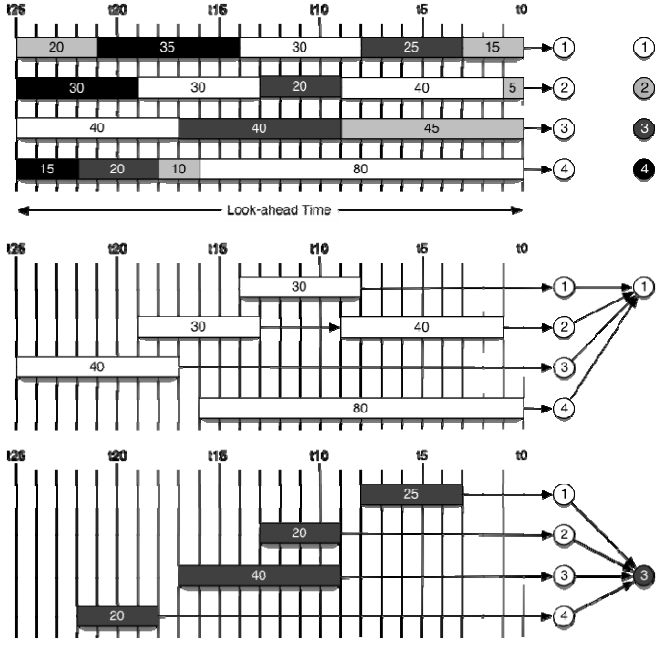

Figure 3: A precognition switch with look-ahead time of t25 and 100% input load. The top picture shows the packets for all output ports, the middle shows the ones for output port 1 and the bottom shows the ones for port 2.

The contention relationship for multiple packets forms a contention graph. For example, packets destined to out-

put port 1 in Figure 3 form the graph in Figure 4. Picking the packet at t1 drops packets at t0 and t8. But if t8 is dropped then t0 may be accepted. The decision of which packet to accept is nontrivial. To achieve higher throughput, the precognition switch needs to pick a packet so that the overall number of bytes accepted by the switch is maximized. This requires an exhaustive search among all packets in the graph.

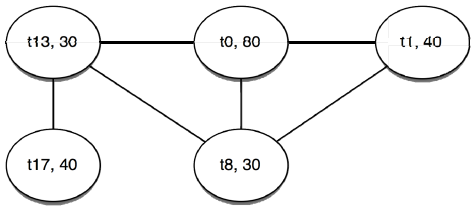

Figure 4: Contention graph for packets destined to output port 1 in Figure 3. Each vertex represents a packet with (arrival time, length) and an edge represents a contention.

V. NUMBER OF PACKETS IN A CONTENTION GRAPH

We need to know how many packets could be in a contention graph in order to figure out the complexity of the search. The packet length is modeled using the following distributions: exponential (labeled as *expo(x)* in the graph), fixed (*fixed(x)*), uniform (*uniform(x_1,x_2)*) and bimodal (*bimodal(x_1-p_1,x_2-p_2)*). The parameter x is the mean packet length. On the uniform distribution, x_1 and x_2 are the smallest and the biggest length. On bimodal, x_1 is generated with probability p_1 and x_2 is generated with probability p_2. Every input port uses the same traffic and each uses uniform output port selection for each packet. The uniform selection applies for all arrival processes.

We primarily simulate quasipoisson arrivals. Quasipoisson is a Poisson arrival with a modification to prevent an arrival while the previous packet is still transmitting. We choose Poisson for ease of modeling, but there is recent evidence that traffic tends to be Poisson as the load increases [6]. As an aside, we also simulate Pareto on/off arrivals with parameter α=1.4 and k=4000 bytes. This model has been used for simulating web traffic [10]. The value k determines the burst and silence period. We set α=1.4 so that the period has infinite variance. For this traffic pattern, the packet length is bimodal in which 80% are 40 bytes and the rest are 1500 bytes (referred to as *bimodal(40-0.8,1500-0.2)* in the plots). This reflects recently measured packet length distributions [5]. The destination of packets in a single burst is uniformly distributed. We indicate the arrival process in the plots by prefixing its name to the notation for the packet length distribution.

Figure 5 shows the simulator output for the number of packets in the contention graph for 100% input load for traffic with mean packet length of 165 bytes. We pick this packet length to be close to the average packet length in recently observed traffic statistics [5]. The switch size is 32x32 and look-ahead time is 8ms. The higher the variance of the packet length distribution, the more packets are in the contention graph, making exhaustive search more expensive. For most of the traffic in these figures, we need to consider a look-ahead of just 16 packets. The bimodal traffic *quasipoisson-bimodal(40-0.9,1290-0.1),* in which most of the packets are small and very few are big, requires a look-ahead of more than 16 packets.

We also look at the effect of switch size and input load on the number of packets in the contention graph. We show in Figure 5 the output of only the bimodal distribution because it has the flattest distribution. The simulation shows that as the switch gets bigger, the number of packets in the contention graph also gets higher (Figure 6). However, as the switch size approaches 32x32, the CDF plots converge. This result is consistent with the throughput plot in Karol's paper [1]. Thus, we focus our throughput simulation on a 32x32 switch. The input load also affects the contention graph size distribution. We presented 3 cases of input load: 10%, 50% and 100% in Figure 7. Higher input load has greater contention probability. The contention graph gets denser as the input load increases. This also makes the exhaustive search more expensive.

The number of packets in a contention graph presented here also clarifies that the throughput achieved by our simulation is close to the maximum throughput. To ensure that our simulation gives the maximum throughput, we use a very long look-ahead (8ms or 6000 packets per input port on average) in the simulation.

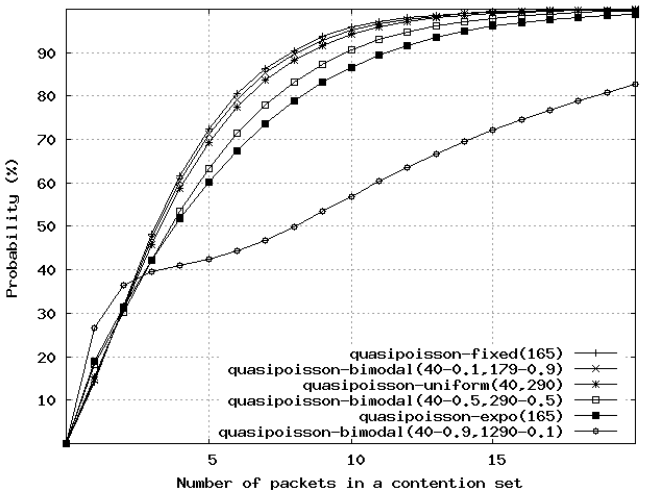

Figure 5: CDF of the number of packets in a contention graph for Poisson arrival with multiple distributions of packet length (mean = 165). The distribution with lower variance is listed first.

VI. EXHAUSTIVE SEARCH ALGORITHM

The search space is the entire set of possible combination of accept and drop alternatives for each packet in the contention graph; the switch calculates the number of bytes accepted for each combination. It then picks the combination with the maximum throughput. As illustrated in Table 1, the number of possible combinations is 2^n, where n is the number of packets in the graph. The search space grows very big even for a small n.

To limit the search space, we reduce the number of packets to be considered in the graph to m, where $m < n$, so that we run n/m phases of the search algorithm. The packets that are part of the original graph but are not considered in this set form another graph. The combination picked for the second graph might contend with the combination picked for the first. When this happens the decision based on the first graph wins because the switch already connects the input port to the output port. When the packet chosen on the second graph arrives, it will be discarded.

For example, Table 1 shows that when contention is detected for combination #3, the throughput is zero. In this case, combination #17 yields the maximum throughput. Table 2 shows the case when $m < n$. The top table shows the first graph, where the scheduler picks combination #4. The one on the bottom shows the second graph, where the scheduler picks combination #1. In this case, the output of the second graph does not have any conflict with the output of the first graph and the combined output matches to the output of Table 1 (m=n).

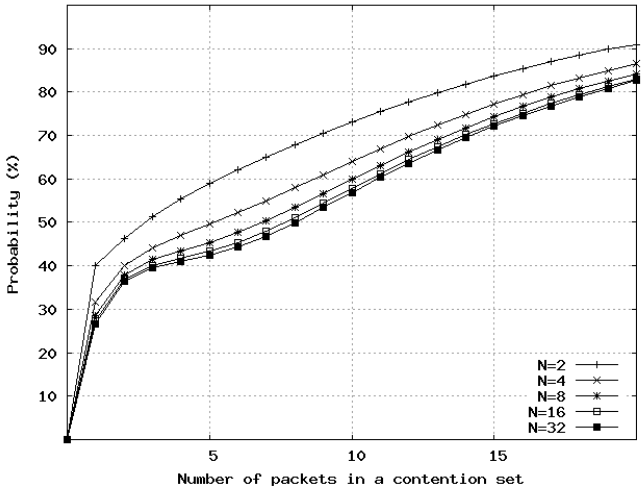

Figure 6: CDF of the number of packets in a contention graph for *quasipoisson-bimodal(40-0.9,1290-0.1)* at 100% input load for multiple switch sizes (*N*x*N*). Look-ahead time is 8ms.

The algorithm is explained more formally as follows:

The switch receives notification from the upstream node.

1. Record information about the packet in the contention graph G for the destination output port of the packet. Sort the graph based on the time of arrival to the switch.

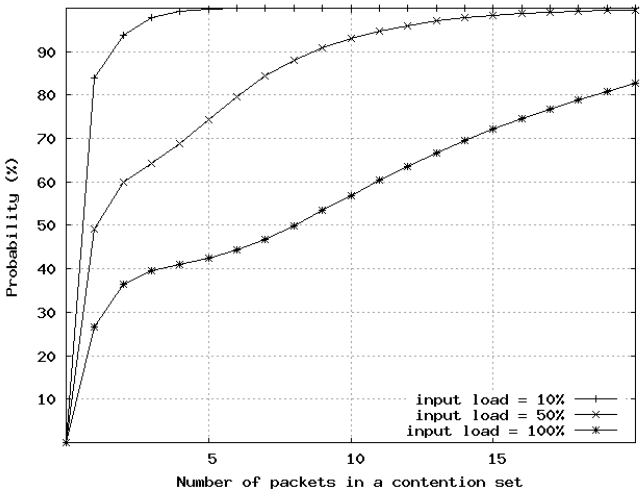

Figure 7: CDF of the number of packets in a contention graph for *quasipoisson-bimodal(40-0.9,1290-0.1)* at 10%. 50% and 100% input load for 32x32 switch. Look-ahead time is 8ms.

	Packets					Number of bytes accepted
	t0	t1	t8	t13	t17	
1	drop	drop	drop	drop	accept	40
2	drop	drop	drop	accept	drop	30
3	drop	drop	drop	accept	accept	0 (contention)
…						
17	accept	drop	drop	drop	accept	120 (max)
…						

Table 1: Exhaustive search (m=n=5) for the whole set of packets destined to output port 1 in Figure 3.

	Packets			Number of bytes accepted
	t0	t1	t8	
1	drop	drop	accept	30
2	drop	accept	drop	40
3	drop	accept	accept	0 (contention)
4	accept	drop	drop	80 (max)
5	accept	drop	accept	0 (contention)
6	accept	accept	drop	0 (contention)
7	accept	accept	accept	0 (contention)

	Packets		Number of bytes accepted
	t13	t17	
1	drop	accept	40 (max)
2	accept	drop	30
3	accept	accept	0 (contention)

Table 2: Exhaustive search for output port 1 in Figure 3 but only considering the first m=3 packets.

A packet arrives at the switch.

1. Check if the packet has been marked. If yes, accept/drop based on the marker. Done.
2. Else:
    a. Execute *ExhaustiveSearch*(m, G) on the contention graph for the output port.

b. Repeat step 1.

ExhaustiveSearch(*m*, *G*) searches for the combination of accepted/dropped packets that produces a contention-free schedule and yields maximum number of bytes accepted. The search considers only the first *m* packets in graph *G*.

1. For the first *m* packets in *G* produce 2^m-1 possible combinations of accepts/drops.
2. For each combination do:
   a. Check if the accepted packets in the combination are free of contention.
   b. Set the $b \leftarrow 0$ if there is a contention. Else, set $b \leftarrow$ sum of accepted packet lengths.
   c. Check if *b* is greater than the recorded maximum. If yes, record *b* and the combination that produced it.
3. Mark the first *m* packets according to the recorded maximum throughput combination.
4. Remove the first *m* packets from *G*.

VII. Simulation Results

A. Quasi-Poisson Arrivals

We simulate unbuffered and precognition switches with exhaustive search, varying the number of packets being considered (*m*) from 4 to 12: *m*=4, *m*=8, and *m*=12. Due to the high computational cost, we are not able to run the simulation for higher *m*. To overcome this limitation, we run the simulation with increasing *m* to look at the trend. All the plots in this section are generated from simulation of a 32x32 switch and look-ahead time of 8ms. We vary the switch size in Figure 10 but keep the look-ahead time unchanged.

The exhaustive search shows some improvement over the baseline. Figure 8 shows that result for the exponential packet length case. The exhaustive search with *m*=12 graph indicates the maximum achievable throughput of the precognition switch, because the throughput does not get better as we increase *m* from 8 to 12. As indicated on the plots in Figure 9 and Figure 10, the throughput increase is minimal for distributions with lower variance. The throughput increase is the difference between the throughput achieved by precognition exhaustive search and the throughput achieved by the baseline unbuffered switch.

The increase for all the considered packet length distributions is minimal (see Figure 9). This indicates that looking into the future to find the combination of packets to accept does not improve throughput. Most of the packets are still dropped.

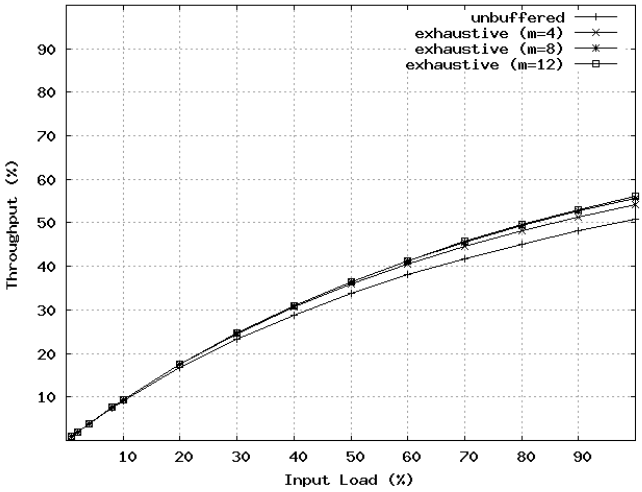

Figure 8: Throughput of a 32x32 switch for *quasipoisson-expo(165)* for unbuffered switch and precognition optical switch running exhaustive search algorithm.

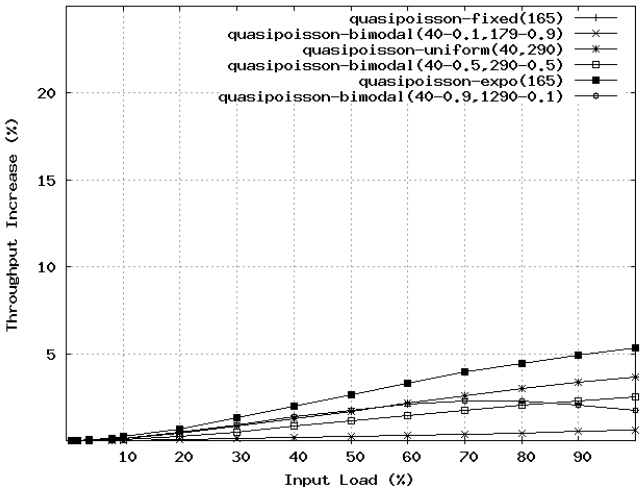

Figure 9: Throughput increase of a 32x32 switch for quasi-poisson arrivals for precognition exhaustive search (*m*=12). The plot is for various distributions of packet length with the same mean of 165 bytes.

For a bimodal distribution with mostly small packet lengths and very infrequent large packet lengths, the improvement for *m*=12 maximizes at 60% input load (Figure 9). The plots in Figure 6 show that the number of packets in the contention graph is almost uniformly distributed. Considering only 12 packets in the graph is not enough for this distribution. It is likely that increasing *m* will eventually improve performance, but the computational load to compute a schedule is excessive.

Figure 10 shows that the switch size has some effect to the throughput improvement. The bigger the switch the higher the throughput increase. Smaller switches do not result in much increase. As expected, an exhaustive search on fixed-size packets does not improve performance.

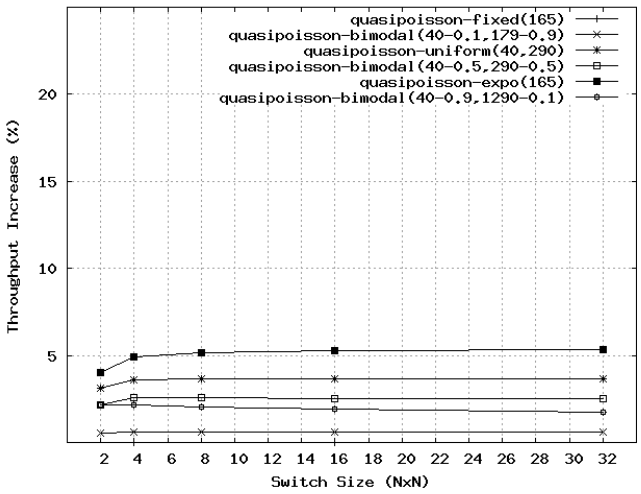

Figure 10: Throughput increase of a *N*x*N* switch at 100% input load for quasipoisson arrivals for precognition exhaustive search (*m*=12). The plot is for various distributions of packet length with the same mean=165 bytes.

In summary, for quasipoisson arrivals the precognition switch improves the throughput by only 5%. This improvement is disappointingly marginal, especially considering the complexity of the mechanism.

B. Pareto On/Off Arrivals

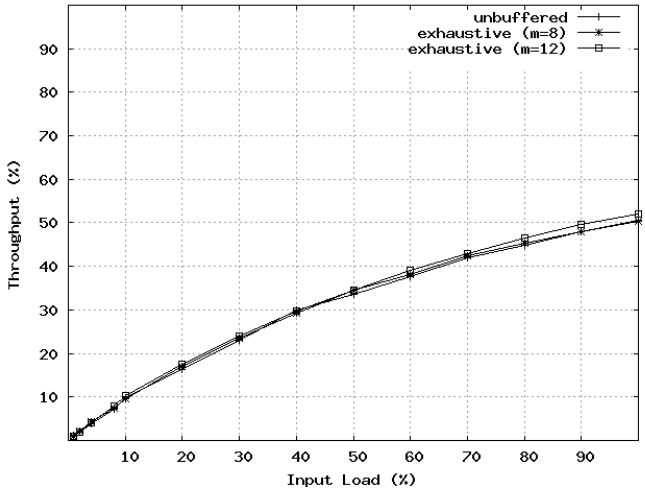

Figure 11: Throughput of a 32x32 switch for *pareto-bimodal(α=1.4, k=4000, 40-0.8, 1500-0.2)* for the unbuffered switch and precognition optical switch running exhaustive search algorithm.

We also simulate Pareto on/off arrivals as a comparison. The packet length distribution in this case is *bimodal(40-0.8,1500-0.2)*. Using a precognition switch increases the throughput only 1% (Figure 11). The increase is only 1% for all switch sizes tested. We even see that the 1% throughput improvement happens when *m*=12, and is even lower when *m*=8. Although we do not simulate higher *m*, we expect that further improvement is not provided.

VIII. CONCLUSIONS

From the result of our simulations, we conclude that looking into the future and exhaustively searching for packets that yield the maximum number of bytes accepted does not significantly improve the throughput. Although the approach does offer some improvement, it cannot adequately prevent the substantial loss of packets from which the unbuffered switches suffer. To improve throughput an optical switch must avoid dropping. The challenge remains on how to do that without buffering.

As the next step, we are looking into a different design where the buffering is done using the propagation delay property in the fiber material and using time shifting for collision avoidance. We expect to combine look-ahead with this shifting to provide some of the benefits of buffering without requiring random-access buffers.

REFERENCES

[1] M. J. Karol *et al*. Input Versus Output Queueing on a Space-Division Packet Switch. *IEEE Transactions on Communications*, Vol. COM-35, No. 12, Dec 1987.

[2] S. W. Fuhrman. Performance of a Packet Switch with Crossbar Architecture. *IEEE Transactions on Communications*, Vol. 41, No. 3, Mar 1993.

[3] T. E. Anderson *et al*. High Speed Switch Scheduling for Local Area Networks. *ACM Transactions on Computer Systems*, Vol. 11, No. 4, Nov 1993.

[4] N. McKeown. Scheduling Algorithms for Input-Queued Cell Switches. PhD Thesis, Stanford University, 1995.

[5] Rishi Sinha *et al*. Internet Packet Size Distributions: Some Observations. USC/ISI Technical Report ISI-TR-2007-643, May 2007.

[6] Jin Cao *et al*. Internet Traffic Tends Toward Poisson and Independent as the Load Increases *Nonlinear Estimation and Classification*, edited by C. Holmes, D. Denison, M. Hansen, B. Yu, and B. Mallick, Springer, New York, 2002.

[7] S. Yao et al. Advances in Photonic Packet Switching: An Overview. *IEEE Communications Magazine*, Feb. 2000.

[8] C. Qiao and M. Yoo. Optical Burst Switching (OBS) – A New Paradigm for an Optical Internet. *Journal of High Speed Networks*, 1999.

[9] Z. Haas. The "Staggering Switch": An Electronically Controlled Optical Packet Switch. *Journal of Lightwave Technology*, Vol. 11, No. 5/6, 1993.

[10] M.E. Crovella et al. Heavy-Tailed Probability Distributions in the World Wide Web. *A Practical Guide To Heavy Tails*, edited by R. J. Adler, R. E. Feldman, M. S. Taqqu, Chapman and Hall, New York, 1998.