Analysis of Reduced Rate Scheduling for Switches with Reconfiguration Overhead

Xin Li and Mounir Hamdi
Department of Computer Science
Hong Kong University of Science & Technology
Clear Water Bay, Kowloon, Hong Kong
Email: {lixin,hamdi}@cs.ust.hk

***Abstract*— Hybrid switch architecture with electronic buffering/processing and optical switching fabric is receiving a lot of attention as potential candidate for the design of high-performance and scalable switches/routers. However, the reconfiguration overhead of optical fabrics brings new challenges to system and scheduling algorithm design. For example, speedup is compulsory to make the switch stable; the scheduling rate has to be reduced compare to the traditional slot-by-slot scheduling in electronic switch. This paper provides instructions on how to choose speedup, scheduling algorithm and holding time. Main results include: 1. A speedup of 2 ensures switch stability and is cost-effective. 2. Effect of reconfiguration delay is exaggerated at high traffic load and it is worth using complex scheduling algorithm. 3. AVERAGE holding method performs better under most traffic scenarios.**

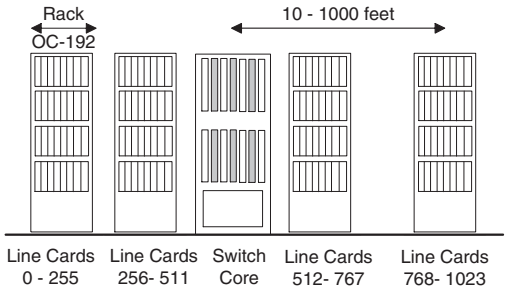

Fig. 1. A multi-rack switching system (1024x1024, OC192, 4 racks)

I. Introduction

The ever increasing Internet is creating an insatiable demand for access and transmission technologies, which has led to a greater need for high-performance switches/routers than ever before. For example, switches of size 256 to 1024, running at OC-192 (10Gb/s) which have a capacity of over 1Tb/s are becoming (will soon become) a necessity for most IP core networks.

Although transmission line rates increases rapidly over the past years (OC-3 (155Mb/s) to OC-192 (10Gb/s) and OC-768 (40Gb/s)), the aggregate switch throughput grows by increasing the number of ports rather than port speed [1]. The main reasons are: 1. New technology used in transmission links, such as as Dense Wavelength Division Multiplexing (DWDM), vastly increases the number of channels available on a single fiber rather than the speed of a single channel. This translates to more ports. 2. On the electrical side, the line rate times the SRAM clock cycle cannot exceed the minimum packet size. Given the popular minimum packet size in range 32B to 64B, and feasible cycle of 2ns, the maximum line rate is 128Gb/s to 256Gb/s, which is only one generation away from the current line rate. 3. Processing in the network processor (NP) at these rates will be extremely difficult and costly.

Contrary to the expansion of port number, port density on line card is not increasing due to considerations in physical implementation, such as port failure avoidance. Port number boost and the constant port density directly lead to an increase in physical line card space. However, Network Equipment Building System (NEBS) physical-packaging requirements impose strict limitations on the number of cards in a single rack. This necessitates the distributed design of whole switch/router over several racks. One rack would contain the switch fabric and the scheduler, and the other racks would contain the line cards as shown in Fig. 1.

In multi-rack systems, cables replace backplane, making it possible to transport electrical signals between different racks, using coax, twisted-pair, etc. But there maybe thousands of signals travelling between different racks, and this would lead to a cabling nightmare. One technology increasingly used to solve the problem is optical fiber interconnections [2]. Assuming that traditional electronic switching is deployed, packets that arrive from the external interfaces are first processed and buffered in the electronic domain. Data are then transmitted over an optical fiber link to the switch fabric rack, where they are processed again electronically. Once they are switched, a different fiber connection is used to transmit the packets to the egress line cards, where they are again processed and buffered in electronics, before finally being transmitted in optics. Inside the switch, a series of opto-electronic conversions are needed, suggesting a significant cost and power requirements. In other words, if the whole switch/router design cannot fit within a single rack, then it becomes extremely difficult to come up with a cost-effective solution using an electronic switching fabric.

For exactly these reasons, a hybrid switch architecture with electronic buffers and optical switching fabric, as shown in

This work was supported in part by a grant from Hong Kong Research Grant Council (Grant Number: RGC HKUST6160/03E). Corresponding author: Mounir Hamdi (hamdi@cs.ust.hk).

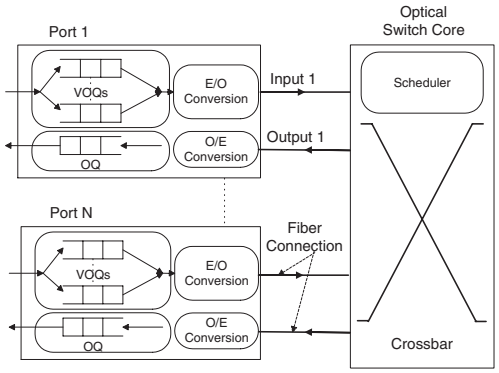

Fig. 2. System architecture of multi-rack hybrid packet switch

Fig. 2, seems to be promising in designing high-capacity scalable switches. Moreover, optical fabrics have many advantages over their electrical counterparts in terms of higher capacity, lower power consumption and lower cost.

However, large-scale hybrid packet switch/routers present unique challenges: the reconfiguration time of an optical switch fabric is *much longer* than that of an electronic fabric. The reconfiguration overheads mainly include laser tuning(in the order of 20-30ns), system clock synchronization (in the range of 10-20ns or higher) and extra margins to avoid data loss [2]. The total overhead can be in the range of 50-100ns. Yet the transmission time of a typical 64B packet at 40Gb/s is less than 10ns. It is obvious that traditional slot-by-slot scheduling algorithms may severely cripple the switch performance by having too much reconfiguration overhead. Instead, schedules should be held for some time to avoid frequent fabric reconfigurations. In other words, the scheduling rate of hybrid packet switch needs to be *reduced*.

The configuration delay brings new challenges and our main goal in this work is to address some important design issues for hybrid multi-rack switch, including choices related to speedup, scheduling algorithm and holding time. Through theoretical deduction and extensive simulations, we provide guidance on all problems mentioned above. The paper is organized as follows. Section II briefly introduces the execution of reduced rate scheduling. It also gives instructions on what is an appropriate speedup and how to choose the scheduling algorithm. Section III discusses two holding schemes: fixed rate and variable rate. Their performance under different traffic scenarios are presented and analyzed. Section IV is the conclusion.

II. Reduced rate scheduling

As mentioned earlier, the use of optical fabric in hybrid packet switch/router brings significant reconfiguration delay. Assuming that the line rate is R, packet size is s and reconfiguration delay is λ. If slot-by-slot scheduling is used, the percentage of bandwidth for actual traffic sending (effective bandwidth) is $\frac{s/R}{s/R+\lambda}$. For a switch which transfers 64B packet at 40Gb/s line rate and suffers 100ns reconfiguration delay, effective bandwidth is just around 10%. This suggests the fabric can sustain high throughput only with a very large speedup.

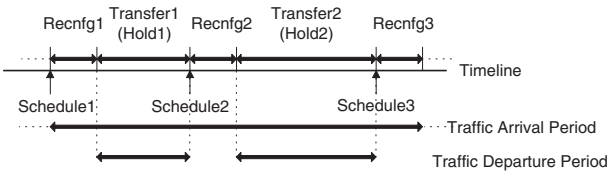

Fig. 3. Illustration of reduced rate scheduling

It is clear the scheduling rate must be *reduced* to compensate for reconfiguration delay. Each schedule holds for some time rather than changing at every time slot. Similar ideas can be found in burst-mode switching [3], packet-mode scheduling [4] and envelope scheduling [5]. The scheduling algorithm generates one schedule and a corresponding holding length at each run. The system works in schedule-reconfiguration-transfer(hold) cycle, as shown in Fig. 3. Maximum Size Matching (MSM) or Maximum Weight Matching (MWM) algorithms and their iterative approximations can be used as scheduling algorithms. The scheduling decision is made based on the queuing state at that moment. The holding length of a particular schedule can be either fixed or vary with the switch state.

The discussion in the previous paragraphs presents the research issues on scheduling hybrid optical switches. These issues include:

- Is speedup needed to ensure the stability of the switch? If so, what is the required speedup value?
- How to choose the scheduling algorithm? Is it worth of choosing complex algorithms rather than simple ones? Furthermore, is it practical to implement those complex algorithms?
- What is a good holding length, or the executing rate of the scheduling algorithm?

In this paper, we assume a $N \times N$ Virtual Output Queuing / Output Queuing (VOQ/OQ) non-blocking switch with a reconfiguration delay of λ, as shown in Fig. 2. The switch operates on fixed-size data units (say, cell) in slotted manner and has integer speedup S. There is at most 1 arrival for each input at each time slot. Denote λ_{ij} to be the arrival rate of traffic from input i to output j. Only *admissible* traffic is considered in this paper. That is, $\forall j \in 1 \cdots N, \sum_{i=1}^{N} \lambda_{ij} < 1$ and $\forall i \in 1 \cdots N, \sum_{j=1}^{N} \lambda_{ij} < 1$. All the simulations in this section run on a 8×8 switch with reconfiguration delay $\lambda = 5T$ (T is the slot time). Input traffic pattern is bernoulli i.i.d. uniform.

A. Speedup

Fig. 3 shows the traffic arrival and departure periods. Traffic which arrives during the reconfiguration periods is blocked before being switched. When the traffic load is high, the fabric has to run at higher switching speed than the line rate during the transfer periods. In other words, speedup is always needed to compensate for the blockage during the reconfiguration period.

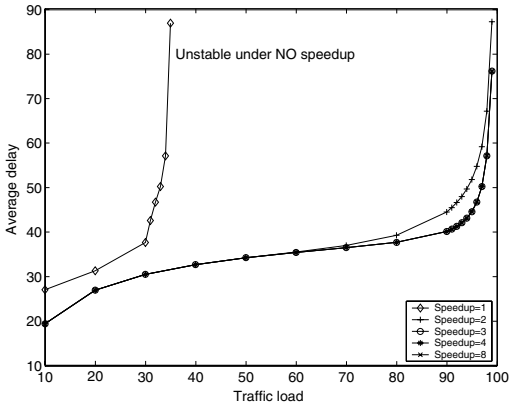

Fig. 4. Effect of speedup on average cell delay

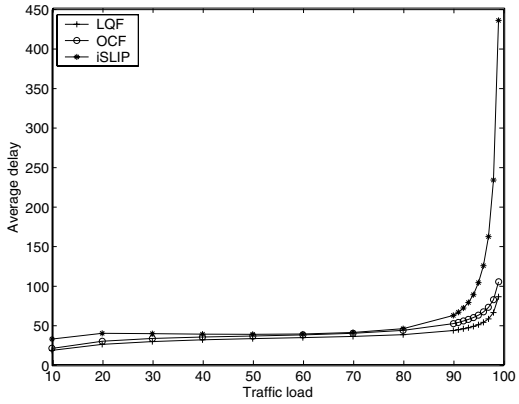

Fig. 5. Effect of scheduling algorithm on average cell delay (uniform traffic)

Observation 1: Speedup is mandatory for switches with reconfiguration delay.

Next, we try to figure out the effect of speedup on switch performance (e.g. average cell delay). The scheduling algorithm is Longest Queue First(LQF)[6] and each schedule is held at the same length as reconfiguration delay. Fig. 4 shows the average cell delay under different speedup values. With no speedup, the switch turns unstable at around 40% traffic load, which supports Observation 1. Other observations can be made from Fig. 4 are,

Observation 2: The switch is stable at speedup 2 under uniform traffic.

Observation 3: Larger speedup does not help a lot in reducing the average cell delay, especially under low or medium traffic load.

The following theorem shows the required speedup value which ensures stability under any admissible traffic pattern. (For detailed explanation and proof of Theorem 1, please refer to [7]).

Theorem 1: Assume a switch has reconfiguration delay λ, uses LQF scheduling algorithm and holds every schedule for a predefined constant time h. The switch is stable under any admissible i.i.d. input traffic pattern as long as it has a speedup $S \geq \lceil \frac{\lambda+h}{h} \rceil$.

As slotted-time switch has integer speedup, from all above, a switch running at speedup of 2 guarantees high performance and is cost-effective.

B. Choice of Scheduling Algorithm

After settling down the required speedup, the focus now is how to choose an appropriate scheduling algorithm. Fig. 5 shows the average cell delay achieved by three scheduling algorithms: Longest Queue First(LQF)[6], Oldest Cell First (OCF)[6] and iSLIP[8]. The switch has a speedup of 2. Each schedule is held at the same length as reconfiguration delay.

In Fig. 5, LQF and OCF achieve shorter average delay than iSLIP. When traffic load is low, most of the arrivals will be switched out immediately, or as soon as the current reconfiguration period finishes. The delay is mainly determined by the performance of scheduling algorithm. It is not surprising that delay of MWM (e.g. LQF and OCF) algorithms is shorter than that of iSLIP, but with no big difference. However, when traffic are crowded, a less optimized schedule may cause packets waiting several reconfiguration-transmission cycles before being switched out. Inefficiency of the algorithm is exaggerated by the reconfiguration delay. For example, the average cell delay of iSLIP algorithm is much larger than LQF at high load. Since most switches work under their full capacity (high load), it is necessary to choose complex algorithm to guarantee good performance.

However, the better performance achieved by optimized weight/size scheduling algorithm comes at the cost of increased computational complexity. Assuming that the switch has N ports, MWM algorithms have complexity $O(N^3 \log N)$, and MSM algorithms have complexity $O(N^{2.5})$. Luckily, reduced rate scheduling allows a longer time to make a scheduling decision, thus relaxing the constraint on scheduling algorithm complexity. This gives us the freedom to choose complex scheduling algorithm (e.g. LQF) to achieve better performance.

III. Holding Time of Reduced Rate Scheduling

As we mentioned in the previous section, issues like speedup, scheduling algorithm and scheduling rate (equals to holding time of each schedule) are concerned on scheduling optical switch. This section first investigates the relationship between holding time and reconfiguration delay. Then, issues of fixed and variable holding methods are discussed.

A. Tradeoff between Outdated Schedule (Empty Time Slots) and Reconfiguration Delay

As the scheduling rate is reduced, the schedule of the switch is updated once several time slots. The schedule is based on queueing states at the moment it is made. However, traffic arrives and departs during the holding time while queuing states change. Even if the schedule is the optimum when the decision is made, from a 'slot-to-slot' point of view, it is becoming outdated during its holding time. A drawback of this 'outdated schedule' is that it may cause bandwidth loss as shown in Fig. 6. Part of the fabric connections may have no

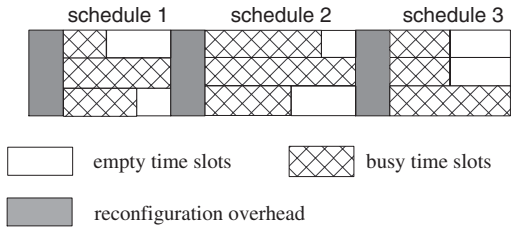

Fig. 6. Bandwidth wastage due to outdated schedule, e.g. 3×3 switch

more packets to send during the holding time, but still can not be freed for reuse. Increasing the scheduling frequency helps to relieve the problem, but this induces bandwidth wastage on resettling the switch. It is obvious there exists a tradeoff between outdated schedule and reconfiguration delay. Good holding time should achieve a balance between these two inverse factors.

The holding time can be a predefined constant or a variable. These two schemes are named as *fixed rate* and *variable rate* scheduling in this paper. Choice of the holding time may take into account reconfiguration delay, queueing state and traffic arrival statistics, etc.

B. Fixed Rate (Holding Time) Scheduling

The holding time h of fixed rate scheduling is a predefined constant. The switch is rescheduled every h time slots. An appropriate value of h should be chosen based on both system performance (e.g. stability, average cell delay) and cost (e.g. required speedup). From Theorem 1, the switch is stable under any admissible i.i.d. traffic as long as it has a speedup $S \leq \lceil \frac{\lambda+h}{h} \rceil$. If holding time is set to be less than reconfiguration delay (e.g. $h = \lambda/2$), the system needs to be equipped with higher speedup capacity (e.g. equal or larger than 3). The advantage is that schedules are more update, but this may increase the cost on hardware. If the holding time is set to be larger than the reconfiguration delay, only a speedup of 2 is needed. To determine the value of holding time may require knowledge of traffic arrival statistics and the scheduling algorithm been used.

C. Variable Rate (Holding Time) Scheduling

Traffic statistics is hardly to be obtained and varies from time to time. A predefined holding length may cause unbalanced performance under a different traffic load, which is a direct result of the tradeoff between outdated schedule and reconfiguration delay. Under low traffic load, most VOQs are empty or short. A long holding time generates large amounts of empty time slots. However if a short holding time is chosen, under high traffic load, the switch may needlessly waste bandwidth on frequent reconfiguration. The discussion naturally leads to the idea of variable rate scheduling. The holding time is no longer fixed but changes with the queuing states.

In this section, we investigates and compares the performance of four holding methods. Assume in schedule i, k connections are set up and the corresponding connected VOQs are $Q_i, i = 1, \ldots, k$. Holding length h_i is determined by the following methods:

- *FIXED*: h_i equals to the reconfiguration delay λ.
- *AVERAGE*: h_i equals to the average queue length, that is, $h_i = \sum_{i=1}^{k} Q_i/k$.
- *MAX*: h_i equals to the longest queue length, that is, $h_i = max\{Q_i\}$.
- *MIN*: h_i equals to the shortest queue length, that is, $h_i = min\{Q_i\}$.

MAX and MIN holding methods above are two extreme approaches: MIN eliminates empty time slots during the transmission period; MAX executes a kind of exhaustive service, holding the connections as long as any connected VOQs have cells to send .

The performance of those four holding methods is evaluated by the average cell delay on a 8×8 hybrid switch with reconfiguration delay $\lambda = 5T$ (T is the length of a time slot). Assume the traffic load is p, traffic from input i to output j arrives at rate λ_{ij} and the switch has N ports. Four representative traffic scenarios are used to generate arrivals. They are,

- *bernoulli_iid_uniform*: Bernoulli arrivals, i.i.d., destination uniformly distributed over all outputs, unicast. $\lambda_{ij} = p/N$, $\forall i, j \in 1, \ldots, N$.
- *bursty*: Bursts of cells in busy-idle periods, destinations uniformly distributed burst-by-burst over all outputs. Mean burst length is 10.
- *hot-spot*: One of the output is a hot-spot and loaded at a higher rate of traffic. The traffic matrix of hot-spot traffic is shown below (for a 4×4 switch) and $x = p/2N$.

$$\begin{bmatrix} 2x & x & x & x \\ 2x & x & x & x \\ 2x & x & x & x \\ 2x & x & x & x \end{bmatrix}$$

- *two-diagonal*: The traffic is concentrated on two diagonals. One is heavier than the other and $x = p/3$.

$$\begin{bmatrix} x & 2x & 0 & 0 \\ 0 & x & 2x & 0 \\ 0 & 0 & x & 2x \\ 2x & 0 & 0 & x \end{bmatrix}$$

Figure 7 to 10 shows the simulation results of the four holding methods under different traffic scenarios. As a whole, AVERAGE and FIXED perform better than the extreme methods of MAX and MIN. This demonstrates the importance of the tradeoff between empty time slots and reconfiguration delay mentioned in section III-A. Paying attention to only reducing empty time slots (MIN) or reconfiguration delay (MAX) pulls down the performance. Furthermore, it may cause the switch to become unstable under certain traffic pattern (e.g. using MIN under hot-spot traffic and using MAX under bursty traffic).

Although variable rate holding methods AVERAGE performs better (hot-spot, two-diagonal scenario) or almost identical (bursty scenario) to FIXED holding method under all traffic load, their performance under uniform scenario is an exception. As anticipated, cell delay of AVERAGE is shorter

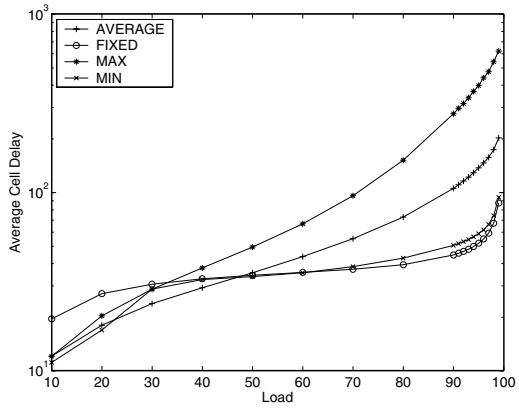

Fig. 7. Average cell delay comparison under uniform traffic

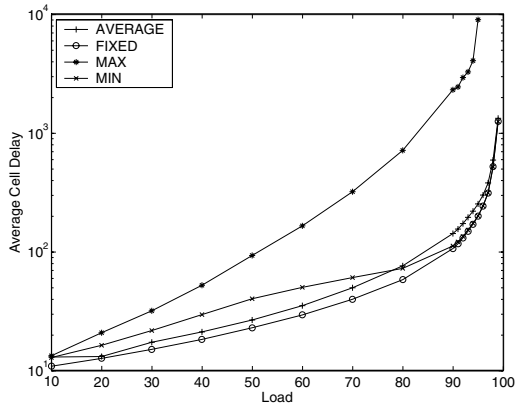

Fig. 8. Average cell delay comparison under bursty traffic

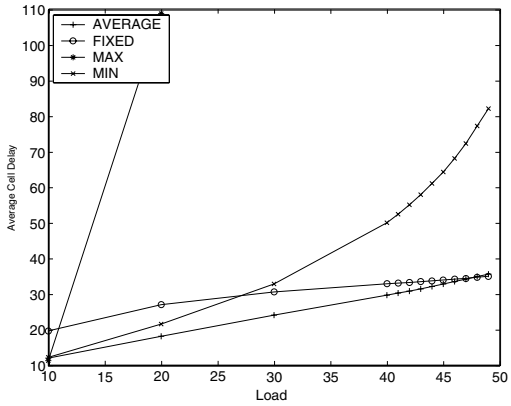

Fig. 9. Average cell delay comparison under hot-spot traffic

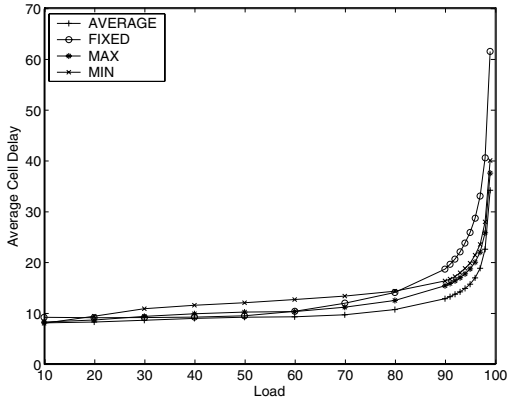

Fig. 10. Average cell delay comparison under two-diagonal traffic

than that of FIXED at lower load. But FIXED has better performance under high traffic load. When traffic load is high, the behavior of LQF is approximate to TDM under uniform traffic. VOQs are been served alternatively and of similar length. Because the holding time equals to reconfiguration delay and the speedup is 2, blocked cells during the reconfiguration period plus newly arrived cells of the transmission period leave almost no empty time slots. In addition, the scheduling rate is the minimum possible under speedup of 2. The two influencing factors: empty time slots and reconfiguration delay are both taken good care of; and hence explains the above situation.

IV. Conclusion

The rapid growth of the Internet is putting demand on high-performance switches/routers. An increase on port number and strict NEBS requirements necessitate the use of multi-rack hybrid switching system. However, the non-negligible reconfiguration overhead has brought new challenges. This paper provides illumination on the choice of speedup, scheduling algorithm and holding time through extensive simulations, result comparisons and theoretical analysis.

References

[1] C. Minkenberg, R. P. Luijten, F. Abel, W. Denzel, and M. Gusat, "Current issues in packet switch design," in *Proc. First Workshop on Hot Topics in Networks HotNets-I*, Princeton, NJ, Oct. 2002.

[2] M. Zirngibl, *Optical Fiber Communications*. Kaminow/Li Academic Press, 2002.

[3] G. Nong and M. Hamdi, "Burst-based scheduling algorithms for non-blocking atm switches with multiple input queues," *IEEE Commun. Lett.*, vol. 4, pp. 202–204, June 2000.

[4] M. A. Marsan, A. Bianco, P. Giaccone, E. Leonardi, and F. Neri, "Pakcet-mode scheduling in input-queued cell-based switches," *IEEE/ACM Trans. Networking*, vol. 10, pp. 666–678, Oct. 2002.

[5] K. Kar, T. V. Lakshman, D. Stiliadis, and L. Tassiulas, "Reduced complexity input buffered switches," in *Proc. the 10th Symposium on High Performance Interconnects (Hoti-10)*, Stanford, Aug. 2002.

[6] N. Mckeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100throughput in an input-queued switch," *IEEE Trans. Commun.*, vol. 47, pp. 1260 –1267, Aug. 1999.

[7] X. Li and M. Hamdi, "Design and analysis of scheduling algorithms for switches with reconfiguration overhead," in *2003 workshop on High Performance Switching and Routing (HPSR'03)*, Torino, Italy, June 2003.

[8] N. Mckeown, "The islip scheduling algorithm for input-queued switches," *IEEE/ACM Trans. Networking*, vol. 7, pp. 188–201, Apr. 1999.