An improved algorithm for CIOQ switches

Yossi Azar[1,*] and Yossi Richter[1,**]

School of Computer Science, Tel Aviv University, Tel Aviv, 69978, Israel.
{azar,yo}@tau.ac.il

Abstract. The problem of maximizing the weighted throughput in various switching settings has been intensively studied recently through competitive analysis. To date, the most general model that has been investigated is the standard CIOQ (Combined Input and Output Queued) switch architecture with internal fabric speedup $S \geq 1$. CIOQ switches, that comprise the backbone of packet routing networks, are $N \times N$ switches controlled by a switching policy that incorporates two components: admission control and scheduling. An admission control strategy is essential to determine the packets stored in the FIFO queues in input and output ports, while the scheduling policy conducts the transfer of packets through the internal fabric, from input ports to output ports. The online problem of maximizing the total weighted throughput of CIOQ switches was recently investigated by Kesselman and Rosén in [12]. They presented two different online algorithms for the general problem that achieve non-constant competitive ratios (linear in either the speedup or the number of distinct values or logarithmic in the value range). We introduce the first constant-competitive algorithm for the general case of the problem, with arbitrary speedup and packet values. Specifically, our algorithm is 8-competitive, and is also simple and easy to implement.

1 Introduction

Recently, packet routing networks have become the dominant platform for data transfer. The backbone of such networks is composed of $N \times N$ switches, that accept packets through multiple incoming connections and route them through multiple outgoing connections. As network traffic continuously increases and traffic patterns constantly change, switches routinely have to efficiently cope with overloaded traffic, and are forced to discard packets due to insufficient buffer space, while attempting to forward the more valuable packets to their destinations.

Traditionally, the performance of queuing systems has been studied within the stability analysis framework, either by a probabilistic model for packet injection (queuing theory, see e.g. [7, 14]) or an adversarial model (adversarial queuing theory, see e.g. [4, 8]). In stability analysis packets are assumed to be identical,

---

and the goal is to determine queue sizes such that no packet is ever dropped. However, real-world networks do not usually conform with the above assumptions, and it seems inevitable to drop packets in order to maintain efficiency. As a result, the competitive analysis framework, which avoids any assumptions on the input sequence and compares the performance of online algorithms to the optimal solution, has been adopted recently for studying throughput maximization problems. Initially, online algorithms for single-queue switches were studied in various settings [1, 3, 10, 11, 13]. Later on, switches with multiple input queues were investigated [2, 5, 6], as well as CIOQ switches with multiple input and output queues [12].

To date, the most general switching model that has been studied using competitive analysis is CIOQ (Combined Input and Output Queued) switching architecture. A CIOQ switch with speedup $S \geq 1$ is an $N \times N$ switch, with N input ports and N output ports. The internal fabric that connects the input and output FIFO queues is S times faster than the queues. A switching policy for a CIOQ switch consists of two components. First, an admission control policy to determine the packets stored in the bounded-capacity queues. Second, a scheduling strategy to decide which packets are transferred from input ports to output ports through the intermediate fabric at each time step. The goal is to maximize the total value of packets transmitted from the switch. The online problem of maximizing the total throughput of a CIOQ switch was studied by Kesselman and Rosén in [12]. They presented two different online algorithms for the general problem that achieve non-constant competitive ratios (linear in either the speedup or the number of distinct values or logarithmic in the value range). In the following we introduce the first constant-competitive algorithm for the general case of the problem, with arbitrary speedup and packet values.

2 The algorithm

We begin with a definition of a parameterized preemptive admission control policy GR_β for a single queue (figure 1). This policy is a generalization of the ordinary greedy policy from [10], that is obtained by setting $\beta = 1$. The latter will be denoted by GR. We then present our parameterized algorithm $\mathsf{SG}(\beta)$ (abbreviation for **Semi-Greedy(β)**) for the problem (figure 2).

Algorithm GR_β [Single-Queue]

Enqueue a new packet p if:

- The queue is not full.
- Or $v(p) > \beta \cdot \min(Q(t))$. In this case the smallest packet is discarded and p is enqueued.

Fig. 1. Algorithm GR_β.

Algorithm SG(β) [CIOQ switch]

1. **Admission control**:
   (a) Input queues: use algorithm GR.
   (b) Output queues: use algorithm GR$_\beta$.
2. **Scheduling:** at each scheduling round do:
   (a) Define the set of eligible packets as the subset of the packets at the head of the input queues, that will be accepted at their corresponding output queues if transferred.
   (b) Compute a maximum weighted matching over the bipartite graph that corresponds to these eligible packets. Transmit the packets that correspond to the computed matching.

Fig. 2. Algorithm SG(β).

Theorem 1. *Algorithm* SG(β) *achieves constant competitive ratio. Specifically, for an optimal choice of the parameter β, algorithm* SG(β) *is 8-competitive.*

References

1. W. A. Aiello, Y. Mansour, S. Rajagopolan, and A. Rosén. Competitive queue policies for differentiated services. In *Proceedings of the IEEE INFOCOM 2000*, pages 431–440.
2. S. Albers and M. Schmidt. On the performance of greedy algorithms in packet buffering. In *Proc. 36th ACM Symp. on Theory of Computing*, 2004. To appear.
3. N. Andelman, Y. Mansour, and A. Zhu. Competitive queueing policies for QoS switches. In *Proc. 14th ACM-SIAM Symp. on Discrete Algorithms*, pages 761–770, 2003.
4. M. Andrews, B. Awerbuch, A. Fernández, J. Kleinberg, T. Leighton, and Z. Liu. Universal stability results for greedy contention-resolution protocols. In *Proc. 37th IEEE Symp. on Found. of Comp. Science*, pages 380–389, 1996.
5. Y. Azar and Y. Richter. Management of multi-queue switches in QoS networks. In *Proc. 35th ACM Symp. on Theory of Computing*, pages 82–89, 2003.
6. Y. Azar and Y. Richter. The zero-one principle for switching networks. In *Proc. 36th ACM Symp. on Theory of Computing*, 2004. To appear.
7. A. Birman, H. R. Gail, S. L. Hantler, Z. Rosberg, and M. Sidi. An optimal service policy for buffer systems. *Journal of the Association Computing Machinery (JACM)*, 42(3):641–657, 1995.
8. A. Borodin, J.Kleinberg, P. Raghavan, M. Sudan, and D. Williamson. Adversarial queuing theory. In *Proc. 28th ACM Symp. on Theory of Computing*, pages 376–385, 1996.
9. E. L. Hahne, A. Kesselman, and Y. Mansour. Competitive buffer management for shared-memory switches. In *Proceedings of the 13th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 53–58, 2001.
10. A. Kesselman, Z. Lotker, Y. Mansour, B. Patt-Shamir, B. Schieber, and M. Sviridenko. Buffer overflow management in QoS switches. In *Proc. 33rd ACM Symp. on Theory of Computing*, pages 520–529, 2001.
11. A. Kesselman and Y. Mansour. Loss-bounded analysis for differentiated services. In *Proc. 12th ACM-SIAM Symp. on Discrete Algorithms*, pages 591–600, 2001.

12. A. Kesselman and A. Rosén. Scheduling policies for CIOQ switches. In *Proceedings of the 15th Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 353–362, 2003.

13. Z. Lotker and B. Patt-Shamir. Nearly optimal fifo buffer management for diffserv. In *Proc. 21st ACM Symp. on Principles of Distrib. Computing*, pages 134–143, 2002.

14. M. May, J. C. Bolot, A. Jean-Marie, and C. Diot. Simple performance models of differentiated services for the internet. In *Proceedings of the IEEE INFOCOM 1999*, pages 1385–1394.