RC23272 (W0407-034) July 14, 2004
Computer Science

IBM Research Report

A Simple Proof of the 2-Competitiveness of the Greedy FIFO Buffering Algorithm

Tracy Kimbrel
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

LIMITED DISTRIBUTION NOTICE: This report has been submitted for publication outside of IBM and will probably be copyrighted if accepted for publication. Ithas been issued as a Research Report for early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distributionoutside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies of the article (e.g, payment of royalties). Copies may be requested from IBM T. J. Watson Research Center , P. O. Box 218, Yorktown Heights, NY 10598 USA (email: reports@us.ibm.com). Some reports are available on the internet at http://domino.watson.ibm.com/library/CyberDig.nsf/home

A simple proof of the 2-competitiveness of the greedy FIFO buffering algorithm

Tracy Kimbrel
IBM T.J. Watson Research Center, Yorktown Heights, NY 10598
kimbrel@us.ibm.com

July 13, 2004

Abstract

We consider the behavior of a simple algorithm for buffering packets weighted by different levels of Quality of Service (QoS) guarantees in a single FIFO queue. Buffer space is limited, and packet loss occurs when the buffer overflows. A simple greedy algorithm is known to have a competitive ratio of 2. Here we give a very simple proof of this.

1 Introduction

Several recent papers address the problem of managing a single queue in a limited capacity buffer [3, 4, 1]. At each time step, many packets may enter the buffer, some packets may have to be be dropped, and the packet at the head of the queue is delivered. Packets are delivered on a FIFO (first in first out) basis: once packets enter the buffer, they may not be reordered. Differentiated service is achieved by assigning different nonnegative values to different packets. The goal is to deliver a subset of packets of greatest possible total value.

A natural and simple greedy policy, which drops minimum value packets only when the buffer is full, has been shown to be 2-competitive, and this is tight [3]. Refinements of this algorithm that drop (low-value) packets more aggressively, i.e., sometimes even when the buffer is not full, have been shown to achieve ratios of 1.98 [4] and 1.75 [1]. Here we give a much simpler analysis of the greedy algorithm.

Our model is as follows. In each unit time, some number of packets arrive. The arriving stream of packets and the set of packets in the buffer from previous steps must be trimmed to size B. Then the packet at the head of the buffer is delivered. We assume the buffer is empty at time 0, and that all buffered packets are allowed to be delivered after the last packet arrival.

We denote by OPT an optimal offline algorithm. An online algorithm ON is called (strictly) c-competitive, if for every instance, the total value of packets delivered by OPT is at most c times the total delivered by ON. See Borodin and El-Yaniv [2] for background on competitive analysis of online algorithms.

2 Analysis of the greedy algorithm

We consider the set of packets arriving at a single time step to arrive one by one. When a packet p arrives, the greedy algorithm (hereafter denoted ON) simply accepts p if there is free space in the buffer. Otherwise it drops a packet p' with the smallest value among p and those in the buffer.

Theorem 1 *The greedy algorithm is 2-competitive.*

Proof: We may assume that OPT never drops a packet that has been admitted to the buffer, since it could instead simply reject the packet on arrival. We denote the queue of packets for an algorithm A by Q_A. Because ON drops packets only when its buffer is full, we know that $|Q_{ON}| \geq |Q_{OPT}|$ at all times.

We use the well-known potential function method; see, for example, [2]. We define the potential

$$\Phi = \sum_{i=1}^{|Q_{OPT}|} v_i - \sum_{i=1}^{|Q_{ON}|} w_i - \sum_{i=1}^{|Q_{OPT}|} w_i$$

where v_i (resp. w_i) is the value of the i^{th} packet in OPT's (resp. ON's) buffer, in FIFO order.

We show for each time step t that $2 \cdot ON(t) \geq OPT(t) + \Delta\Phi$ where $OPT(t)$ and $ON(t)$ denote the values of the packets delivered at time t by OPT and ON, respectively, and $\Delta\Phi$ denotes the change in the value of the potential Φ. We show that this inequality holds separately for arrivals and deliveries; summing the inequalities yields the claim for a single time step as a whole, and summing over all time steps yields the theorem. Strict competitiveness follows since $\Phi = 0$ before any packets have arrived and after the last has been delivered.

For deliveries, the inequality is easily seen to be satisified (and tight if OPT's queue is not empty) if we consider the algorithms to simultaneously deliver the first packets (if any) in their queues.

For arrival of a packet p of value v, we must show $\Delta\Phi \leq 0$. We consider several cases. By Q_A, w_i, etc. we mean the state *before* the arrival of p.

1. OPT rejects p and $|Q_{ON}| < B$. ON accepts p and $\Delta\Phi = -v$.

2. OPT rejects p and $|Q_{ON}| = B$. The first sum in Φ is unchanged. ON must drop p or some other packet. Depending on the position of the dropped packet, each of the second and third sums is unchanged or increases, since the packet dropped by ON is one of smallest value and is replaced in the sum, if present, by a packet of at least the same value.

3. OPT accepts p and $|Q_{ON}| < B$. ON accepts p and the changes in the first two terms in Φ add up to 0. The change in the third term is 0 or less.

4. OPT accepts p and $|Q_{ON}| = B$. Let $x = |Q_{OPT}|$. We consider several subcases.

   (a) p is dropped. $\Delta\Phi = v - w_{x+1} \leq 0$ since v is no greater than any value in ON's buffer.

   (b) p' at position $i > x + 2$ in ON's queue is dropped. $\Delta\Phi = v - v + w_i - w_{x+1} \leq 0$ since w_i is the smallest value in ON's buffer.

   (c) p' at position $i \leq x$ is dropped. $\Delta\Phi = v - v + 2w_i - w_{x+1} - w_{x+2} \leq 0$ since w_i is the smallest value in ON's buffer.

   (d) p' at position $i = x + 1$ or $i = x + 2$ is dropped. These are similar to the previous case.

■

3 Conclusion

Hitherto the analysis of the greedy algorithm [3] and its derivatives [4, 1] has been rather complicated. As mentioned, the greedy algorithm cannot achieve a ratio less than 2. However, it is our hope and conjecture that the simple proof given here can be adapted to obtain improved results for policies that drop packets more aggressively.

References

[1] N.Bansal, L. Fleischer, T. Kimbrel, M. Mahdian, B. Schieber, and M. Sviridenko. Further Improvements in Competitive Guarantees for QoS Buffering. In *Proc. International Colloquium on Automata, Languages and Programming (ICALP)*, 2004.

[2] A. Borodin and R. El-Yaniv.

[3] A. Kesselman, Z. Lotker, Y. Mansour, B. Patt-Shamir, B. Schieber, and M. Sviridenko. Buffer overflow management in QoS switches. In *Proc. ACM Symposium on Theory of Computing (STOC)*, pages 520–529, 2001.

[4] A. Kesselman, Y. Mansour, and R. van Stee. Improved competitive guarantees for QoS buffering. In *Proc. 11th Annual European Symposium on Algorithms (ESA)*, pages 361–372, 2003.