ARTICLE IN PRESS

Theoretical Computer Science ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

A near-optimal memoryless online algorithm for FIFO buffering two packet classes☆

Fei Li

Department of Computer Science, George Mason University, Fairfax, VA 22030, United States

ARTICLE INFO

Keywords:
Online algorithm
Competitive analysis
Buffer management
Packet scheduling

ABSTRACT

We consider scheduling packets with values in a capacity-bounded buffer in an online setting. In this model, there is a buffer with limited capacity B. At any time, the buffer cannot accommodate more than B packets. Packets arrive over time. Each packet has a non-negative value. Packets leave the buffer only because they are either sent or dropped. Those packets that have left the buffer will not be reconsidered for delivery any more. In each time step, at most one packet in the buffer can be sent. The order in which the packets are sent should comply with the order of their arrival time. The objective is to maximize the total value of the packets sent in an online manner. In this paper, we study a variant of this FIFO buffering model in which a packet's value is either 1 or $\alpha > 1$. We present a deterministic memoryless 1.304-competitive algorithm. This algorithm has the same competitive ratio as the one presented in Lotker and Patt-Shamir [Z. Lotker, B. Patt-Shamir, Nearly optimal FIFO buffer management for DiffServ, in: Proceedings of the 21st Annual ACM Symposium on Principles of Distributed Computing, PODC, 2002, pp. 134–142; Z. Lotker, B. Patt-Shamir, Nearly optimal FIFO buffer management for DiffServ, Computer Networks 17 (1) (2003) 77–89]. However, our algorithm is simpler and does not employ any marking bits. The idea used in our algorithm is novel and different from all previous approaches that have been applied for the general model and its variants. We do not proactively preempt one packet when a new packet arrives. Instead, we may preempt more than one 1-value packet at the time when the buffer contains sufficiently many α-value packets.

1. Introduction

We consider online algorithms to schedule packets with values in a capacity-bounded buffer. There is a buffer with a limited size $B \in \mathbb{Z}^+$. At any time, the buffer can accommodate at most B packets. Packets arrive over time. The buffer is preemptive: packets already in the buffer are allowed to be dropped at any time before they are delivered. We use $r_p \in \mathbb{R}^+$ and $v_p \in \mathbb{R}^+$ to denote the *release time* (*arriving time*) and the *value* of a packet p respectively. Packets leave the buffer only because they are either sent or dropped. Those sent and dropped packets will not be reconsidered for delivery any more. Time is discrete. In each time step, at most one packet in the buffer can be sent. The order of the packets being sent should comply with the order in which they are released. The objective is to maximize the total value of the packets sent in an online manner. We call this model a *FIFO buffering model*; this model has attracted a lot of attention in the past ten years and has been studied extensively [1–5]. In this paper, we study a variant of this model in which packets have value either 1 or $\alpha > 1$: 1-*value packets* and α-*value packets*. This variant is called a *two-valued model* and has been investigated in [6,3,5].

☆ This material is based upon work supported by the National Science Foundation under Grant No. CCF-0915681. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

E-mail address: lifei@cs.gmu.edu.

0304-3975/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2011.11.039

Please cite this article in press as: F. Li, A near-optimal memoryless online algorithm for FIFO buffering two packet classes, Theoretical Computer Science (2011), doi:10.1016/j.tcs.2011.11.039

Without knowing the future input, an online algorithm has to make its decision over time based on the input information that it has seen so far. If an online algorithm decides which packet to send only based on the contents of its current buffer and independent of the packets that have already been released and processed, we call this algorithm *memoryless*. Consider a maximization problem as an example. A deterministic online algorithm is called *k-competitive* if its objective value on *any* instance is at least $1/k$ times of the objective of an optimal offline algorithm applied on the same instance [7]. The *upper bounds* of competitive ratio are achieved by some known online algorithms. A competitive ratio less than the *lower bound* cannot be reached by any online algorithm [7]. For the two-valued model, the previously known results include a 1.544-competitive memoryless algorithm [6], a 1.304-competitive algorithm [3] which uses marking bits for all pending packets in the buffer [3], and a non-memoryless *optimal* 1.282-competitive algorithm [5] which applies to the case when the buffer size B goes to infinity. In this paper, we present a 1.304-competitive memoryless algorithm. Our algorithm is simpler than the one in [3] and it does not use marking bits.

It is instructive to compare and contrast the algorithm in [3] with ours since both have the same competitive ratio 1.304. Based on the definition of *memoryless algorithms* [8,9] (a memoryless online algorithm should make the decisions independent of any packets that it has processed), we know that the marking bits used by [3] reflect the packets that the online algorithm has processed and they affect the marking and flush procedure for later arriving α-value packets. Hence the algorithm in [3] is not memoryless.

In Section 2, we describe a deterministic memoryless online algorithm called ON. In Section 3, we give the algorithm's analysis, showing that it is 1.304-competitive. Related work and conclusion remarks are presented in Section 4.

2. Algorithm

Without loss of generality, we assume that all packets have distinct release time. Consider m packets released in the same time step t. We let these m packets have distinct release time of $t, t+\delta, t+2\delta, \ldots, t+(m-1)\delta$ respectively, where $\delta > 0$ and $m \cdot \delta < 1$, in the order of being released.

2.1. The idea

The greedy approach might be the first intuitive method to design competitive online algorithms for the FIFO buffering model. It works as follows: if packets overflow, the minimum-value packet is dropped, with ties broken arbitrarily. In each time step, the earliest released packet in the buffer is sent. The greedy algorithm is asymptotically no better than 2-competitive for the FIFO buffering model, even for the two-valued model. Based on the observation from the tight instance for the greedy approach, Kesselman et al. [4] came up with another idea, which is to proactively preempt the 1-value packets in the buffer released before the α-value packets. Consider a 1-value packet p and an α-value packet q with $r_p < r_q$. On one hand, if q but not p is the packet sent by the optimal offline algorithm, we would like to preempt p proactively at q's arrival and expect that q can be sent before packet overflow happens. On the other hand, if both p and q are sent by the optimal offline algorithm, we would like to preempt p only if p's value is bounded by a fraction of q's value. This idea leads to a memoryless algorithm called PG, which is 1.732-competitive for the general case [5] and 1.544-competitive for the two-valued setting [6].

In algorithm PG, a packet p's is preempted at the time when the algorithm buffers another packet q. A new arrival preempts at most one packet that is released earlier than it. Motivated by the greedy algorithm and the algorithm PG, we propose the following strategy.

Idea 1. *We preempt a set of* 1*-value packets due to the existence of a set of* α*-value packets in the buffer to make room for the potential* α*-value packets that will arrive at the buffer later.*

Different from PG, we take into account the values of multiple α-value packets to preempt one or more 1-value packets. Based on this idea, a 1-value packet is preempted only when the buffer has buffered sufficiently many of later-released α-value packets. In addition, multiple 1-value packets may be preempted at the arrival of one α-value packet.

2.2. A memoryless online algorithm for the two-valued model

We name our algorithm ON. ON represents a family of deterministic memoryless online algorithms parameterized by a real number $\beta > 0$. Denote Q_t^{ALG} as the buffer of an algorithm ALG at time t. If no confusion can be made, we omit the subscript t in our notation Q_t^{ALG}.

Definition 1 (*Ejectable Packet*)**.** Consider a 1-value packet p and an α-value packet q in the buffer with $r_p < r_q$, $v_p = 1$ and $v_q = \alpha$. Such a packet p may prevent us from sending q before a future possible packet overflow. We call the packet p an *ejectable packet*.

Please cite this article in press as: F. Li, A near-optimal memoryless online algorithm for FIFO buffering two packet classes, Theoretical Computer Science (2011), doi:10.1016/j.tcs.2011.11.039