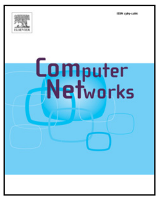
Computer Networks 91 (2015) 804–820

Contents lists available at ScienceDirect

Computer Networks

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



Optimal buffer management for 2-frame throughput maximization[☆]

Jun Kawahara[a,*], Koji M. Kobayashi[b]

[a] Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara, 6300192, Japan
[b] National Institute of Informatics, Japan

A R T I C L E I N F O

Article history:
Received 23 November 2014
Revised 7 May 2015
Accepted 27 August 2015
Available online 10 September 2015

Keywords:
Competitive analysis
Online buffer management
Packet fragmentation
Greedy algorithm

A B S T R A C T

We consider a variant of the online buffer management problem in network switches, called the *k-frame throughput maximization* problem (k-FTM). Large data, called *frames*, carried on the Internet are split into small k packets by a sender, and the receiver can reconstruct each frame only if he/she accepts all the k constituent packets of the frame. Packets pass through network switches on the Internet, and each switch is equipped with a FIFO buffer to temporarily store arriving packets. Since the size of the buffer is bounded, some packets must be discarded if it is full. It is impossible to reconstruct frames including discarded packets any more. Our goal is to maximize the number of reconstructed frames. Kesselman et al. proposed this problem, and showed that any online algorithm has an unbounded competitive ratio even when $k = 2$. Hence, they considered the "order-respecting" variant of k-FTM. They showed that the competitive ratio of their algorithm is at most $(\frac{2kB}{\lfloor B/k \rfloor} + k)$ for any $B \geq k$, where B is the size of the buffer. Also, they gave a lower bound of $\frac{B}{\lfloor 2B/k \rfloor}$ on the competitive ratio when $2B \geq k$ and k is a power of 2. Furthermore, they proved that the competitive ratio of a greedy algorithm is at most $(11 + \frac{8}{B-1})$ for any $B \geq 2$ and $k = 2$.

We analyze a greedy algorithm for $k = 2$, and show that its competitive ratio is at most 3 for any B, improving the previous upper bound of $\frac{4B}{\lfloor B/2 \rfloor} + 2(\geq 10)$. Moreover, we show that the competitive ratio of any deterministic algorithm is at least 3 for any B if $k = 2$, which matches our upper bound.

1. Introduction

Large and sequential data are currently used by real-time multimedia applications on the Internet. The data are called *frames*, which are too large to be transferred over the Internet. (For example, each frame in the case of video data corresponds to each picture of the video.) Thus, they are fragmented into small packets. When all the packets arrive at the receiver, each frame is reconstructed from the packets. Then, each packet has to pass through many switches (routers) on the Internet. Buffer management in the switches can become a bottleneck for transferring packets to the receiver. In particular, each switch is equipped with a buffer to store packets arriving at a burst. However, if the number of arriving packets surpasses the size of the buffer, the switch has to decide which packets can be accepted for insertion into its buffer.

Recently, this kind of problem was modeled as online problems, and a great amount of work has been done. Many models have been proposed, of which the most basic one is as follows [1]: A switch is equipped with a buffer (FIFO queue) of bounded size B. An input consists of a sequence of events. Each event is an arrival event or a send event. At an arrival

☆ A preliminary version of this paper was presented at the 20th International Colloquium on Structural Information and Communication Complexity, SIROCCO 2013.

* Corresponding author. Tel.: +81 -743 -72 -5368; fax: +81 743 72 5369.
E-mail address: jkawahara@is.naist.jp (J. Kawahara).

http://dx.doi.org/10.1016/j.comnet.2015.08.046
1389-1286/© 2015 Elsevier B.V. All rights reserved.

event, one packet arrives at an input port. Each packet is of unit size and has a value that represents its priority. A buffer can store packets provided that the total size of stored packets does not exceed B, namely, a switch can store up to B packets at the same time. Stored packets are delivered in the FIFO order. At an arrival event, if the buffer is full, the new packet is rejected. If there is room for the new packet, an online policy determines, without knowledge of the future, whether to accept the packet. At each send event, the packet at the head of the queue is transmitted. The goal of the problem is to maximize the sum of the values of the transmitted packets. The performance of an online algorithm is evaluated by competitive analysis [5,17]. If, for any input σ, a deterministic online algorithm *ALG* gains the benefit, which is at least $1/c$ of the optimal offline policy for σ, then we say that *ALG* is *c-competitive*.

Kesselman et al. [13] focused on the buffer management with frame reconstruction, and formulated the *k-frame throughput maximization* problem (k-FTM), where $k(\geq 2)$ is an integer. Each arriving packet belongs to some frame, and every frame consists of exactly k packets. We say that a frame f is *completed* if all the packets constituting f are transmitted. Otherwise, we say that f is *incomplete*. Our goal is to maximize the number of completed frames.

Previous Results. Kesselman et al. [13] showed that the competitive ratio of any algorithm for k-FTM is unbounded even when $k = 2$. The order of arrival of each packet in the instance used in the proof does not have the relation between packets in different frames. However, such an instance does not reflect the actual situation of networks since each packet generally arrives in the order of departure in a network such as a IP network. Hence, the authors introduced the *order-respecting* setting as follows: For any frame f, and the $i(\in [1, k])$th arriving packet p which is included in f, we call p the *i-packet* in f. The arrival order of the j-packets of frames f_i and $f_{i'}$ must obey the arrival order of the j'-packets of f_i and $f_{i'}$ $(j' < j)$ (a formal definition will be given later). We call the k-FTM problem in the order-respecting setting the *order-respecting k-frame throughput maximization* problem (k-OFTM).

For the k-OFTM problem, Kesselman et al. showed a lower bound of $\frac{B}{\lfloor 2B/k \rfloor}$ on the competitive ratio for any deterministic algorithm, where $B \geq k/2$ and k is a power of 2. Also, they presented a $(\frac{2kB}{\lfloor B/k \rfloor} + k)$-competitive deterministic algorithm when $B \geq k$. The authors proved that for $k \geq 3$, a greedy algorithm for k-OFTM is not competitive. They also showed that the competitive ratio of the preemptive greedy algorithm for 2-OFTM is at most $11 + 8/(B - 1)$ for any $B \geq 2$. (Roughly speaking, the greedy algorithm is defined as follows: The most preferable packets are 2-packets whose corresponding 1-packets have already been transmitted, and the second most preferable ones are both 1-packets and the corresponding 2-packets which have already arrived but not yet been transmitted.).

Our results. In this paper, we analyze a greedy algorithm (*GR*) for 2-OFTM, and improve the upper bound from $\frac{2kB}{\lfloor B/k \rfloor} + k = \frac{4B}{\lfloor B/2 \rfloor} + 2 \geq 10$ to 3 for any B. (When modifying some of the tie-breaking rules of the greedy algorithm by Kesselman et al. [13], we can have *GR*. The formal definition of *GR* including the tie-breaking rules is described in Section 2.2.) Furthermore, we prove a lower bound of 3 for any deterministic algorithm for any B in 2-OFTM, which matches our upper bound. In computational complexity theory, it is common to evaluate the performance of algorithms by its asymptotic behavior, e.g., when k approaches infinity. However, from a practical point of view, it is natural to assume that the number of packets constructing each frame, namely k, is bounded. Thus, it is significant to analyze the case where k is constant. In addition, *GR* is easy to implement and has a lower computational load than any other algorithm. Hence, it is meaningful to analyze the exact performance of *GR*. Our contribution in this paper is to show that *GR* is the best policy for 2-OFTM.

Let us briefly explain our idea of improvement. Our main idea is to "assign" packets in frames completed by an online algorithm *ALG* to the 1-packet in each frame completed by an optimal offline algorithm *OPT* at the end of the input. Suppose that any 1-packet (2-packet, respectively) in *ALG*'s completed frame is assigned at most x times (y times, respectively). Then, it can be shown that the competitive ratio of *ALG* is at most $x + y$. If the authors in [13] had showed the competitive ratio of some greedy algorithm is at most $11 + 8/(B - 1)$ using "assignments" of packets, $x = 1$ and $y = 2 + 8(1 + \frac{1}{B-1})$ would have been proven. (Note that they did NOT use "assignments".) We prove that the assignment such that $x = 1$ and $y = 2$ can be constructed in order to show that the competitive ratio of *GR* is at most 3. We construct the assignments with time. Specifically, we try to assign each *GR*'s packet at the time when it arrives at the buffer. However, when assigning *GR*'s 1-packet p_1 to *OPT*'s 1-packet, we do not know whether or not to complete the frame f including p_1 in the future. Specifically, the 2-packet in f can be discarded by *GR* after p_1 is transmitted. If p_1 is assigned to some 1-packet which arrives at *OPT*'s buffer, the competitive ratio of *GR* cannot correctly be evaluated. To that end, if we assign the 1-packet in an incomplete frame, (that is, the 2-packet corresponding to the 1-packet is discarded by *GR*), then we assign a packet in a completed frame to the discarded 2-packet. Hence, we can bound the competitive ratio of *GR* from above. To realize the above assignments and attain the tight competitive ratio, we need to classify all the packets into a large number of categories, and do exhaustive case analysis with respect to them. In addition, for some cases, it is necessary to consider not only the time when they occur but also some earlier moments. For example, in order to prove a certain case where some 2-packet p_2 is rejected by *GR*, we need to keep track of the state of assignments from the time when the 1-packet corresponding to p_2 arrives to the time when p_2 is rejected. The most part of the paper is devoted to the proofs to guarantee the assignments in each case. If we simplify our assignment routine or reduce the packet categories in our analysis to shorten the proofs, the upper bound on the competitive ratio must be larger.

After the preliminary version of this paper appeared on SIROCCO2013, Kawahara et al. [10] gave a lower bound of $\frac{2B}{\lfloor B/(k-1) \rfloor} + 1$ for deterministic online algorithms for any $k \geq 2$ and any $B \geq k - 1$, which is 3 when $k = 2$. That is, they generalized our lower bound shown in this paper. Also, they improved an upper bound of $O(k^2)$ by Kesselman et al. to $\frac{5B + \lfloor B/k \rfloor - 4}{\lfloor B/2k \rfloor} = O(k)$ for $B \geq 2k$. Note that this is tight up to a