Non-Preemptive Buffer Management for Latency Sensitive Packets

Moran Feldman* Joseph (Seffi) Naor†

July 5, 2017

Abstract

The delivery of latency sensitive packets is a crucial issue in real time applications of communication networks. Such packets often have a firm deadline and a packet becomes useless if it arrives after its deadline. The deadline, however, applies only to the packet's journey through the entire network; individual routers along the packet's route face a more flexible deadline.

We consider policies for admitting latency sensitive packets at a router. Each packet is tagged with a value. A packet waiting at a router loses value over time as its probability of arriving at its destination on time decreases. The router is modeled as a non-preemptive queue, and its objective is to maximize the total value of the forwarded packets. When a router receives a packet, it must either accept it (and delay future packets), or reject it immediately. The best policy depends on the set of values that a packet can take. We consider three natural sets: an unrestricted model, a real-valued model, where any value over 1 is allowed, and an integral-valued model.

We obtain the following results. For the unrestricted model, we prove that there is no constant competitive ratio algorithm. For the real valued model, we provide a randomized 4-competitive algorithm and a matching lower bound (up to low order terms). We also give a deterministic lower bound of $\phi^3 - \varepsilon \approx 4.236$, almost matching the previously known 4.24-competitive algorithm. For the integral-valued model, we show a deterministic 4-competitive algorithm, and prove that this is tight even for randomized algorithms (up to low order terms).

1 Introduction

A router in a communication network receives, buffers, and transmits packets. Given that the router has only bounded output capacity (and in some architectures also bounded capacity for transferring packets between its components), the router has to decide which packets to transmit now and which ones to keep buffered, hoping to transmit them later. Commonly studied router policies usually make the assumption that either packets are indifferent to delays, or each packet has a firm deadline such that the packet must be forwarded before the deadline (or else it is deemed worthless). The first option corresponds to data packets, which are not very sensitive to reasonable delays. In the presence of data packets, throughput is the main parameter by which a router policy is measured. The second option corresponds to real-time applications, *e.g.*, movie streaming, where a packet arriving too late often becomes useless, making delay a significant issue for such applications.

In both settings the problem faced by a router can be modeled as an online problem in which each packet is associated with a value, and the router wants to maximize the total value of the

*The Open University of Israel, Raanana, Israel. E-mail: moranfe@openu.ac.il.

†Computer Science Department, Technion, Haifa, Israel. E-mail: naor@cs.technion.ac.il. The research of Seffi Naor is supported by ISF grant 1366/07.

transmitted packets, subject to restrictions imposed by the architecture of the router and the requirements of the application. Different router architectures and network applications give rise to various issues, many of which have been extensively studied in the literature (see Section 1.3 for a few examples).

The assumption of a firm deadline for packets belonging to real-time applications is justified from the network's perspective. If a packet arrives at the destination on time, the network gets credit for it, otherwise, the packet is worthless. However, when shifting our perspective from the network to an *individual router*, the situation changes dramatically since a packet goes through many routers on its way to the destination. We can assume that a router is "aware" that delaying a packet increases the odds that it will not arrive on time to its destination, yet a router cannot be expected to "know" for how long can a packet be delayed. Therefore, the objective of a router is to maximize throughput, but without inducing significant delay on any of the forwarded packets. Fiat et al. [10] describe an online model with this objective in mind. In their model, the value of each packet is a decreasing function of the time it waits in the router. This requires the router to consider both throughput and delays. Formally, the model of Fiat et al. [10] assumes a router with a FIFO buffer (queue). Each time the router receives a packet it has to make an irrevocable decision whether to buffer it or reject it. For each buffered packet, the router gets revenue equal to the value of the packet minus the delay that the packet incurs while waiting in the queue. An online algorithm for this problem faces a trade-off between buffering too many packets, which imposes a large delay and negligible revenue from each packet, and buffering too few packets.

In queuing theory, one usually assumes that packet arrival is governed by some known stochastic process. However, in many real applications, defining analytically the stochastic process is difficult, if not impossible. For example, it is common to model network traffic as a Poisson process, yet many studies indicate that real network traffic does not behave at all this way [14, 16, 15]. In order to bypass this difficulty, we use *competitive analysis* as a performance measure for our online algorithms. The advantage of competitive analysis is that it assumes nothing on how the input is generated. Instead, the performance of an online algorithm is compared against an optimal off-line algorithm (that knows the input ahead of time). Formally, let ALG be an online deterministic algorithm, and let OPT be the optimal *off-line* algorithm. Given an input sequence σ, we denote by $ALG(\sigma)$ and $OPT(\sigma)$ the value of the solutions that ALG and OPT output given σ. The competitive ratio of ALG is defined as $\sup_\sigma \frac{OPT(\sigma)}{ALG(\sigma)}$. If ALG is a randomized algorithm then the competitive ratio is defined by

$$\sup_\sigma \frac{OPT(\sigma)}{E[ALG(\sigma)]} \ ,$$

where the expectation is over the randomness of ALG. This definition corresponds to an *oblivious adversary*[1]. Notice that under the above definition, the competitive ratio of an algorithm for a maximization problem is at least 1, as no algorithm can be better than OPT. Sometimes the competitive ratio is defined as the inverse of the above expression, which results in competitive ratios smaller than 1 for maximization problems. However, we prefer the above definition, since we believe that ratios larger than 1 give a better intuitive understanding of the competitiveness they represent.

1.1 The Model

We consider a router model with a single non-preemptive FIFO buffer (queue) and continuous time. At time 0, the queue is empty. Packets arrive at the router at non-integral times. Each

[1] An oblivious adversary is familiar with ALG, but does not know the results of the actual coin tosses of ALG.

packet is either accepted to the queue or rejected. At any integral time, the packet at the head of the queue is dequeued and transmitted, unless the queue is empty at that time.[2] As the queue is non-preemptive, packets may not leave the queue in any other way. For packet d, define $a(d)$ to be its arrival time and $w(d)$ to be its value.

In the HETEROGENOUS DELAY SENSITIVE PACKETS problem (HDSP), each packet d has a revenue depending on its transmission time. If d is never transmitted (*i.e.*, it is rejected), then its revenue is 0. Otherwise, its revenue is equal to $w(d)$ minus the delay it suffers (*i.e.*, the number of integral times d spends in the queue without being transmitted). The objective is to decide which packets to accept into the queue in order to maximize the sum of the revenues of all packets.

In the online setting, each time a packet arrives, the online algorithm must immediately make an irrevocable decision whether to accept the packet to the queue or reject it. The decision must be made before any knowledge is revealed regarding future packet arrivals.

Remark: Notice that the above definition of revenue allows for negative revenues, however, this is not an issue since the revenue of a packet can be determined upon arrival, and no reasonable algorithm would ever let into the queue a packet having a negative revenue.

It turns out that the competitive ratio achievable for HDSP highly depends on the set of values that the packets may take. We consider three natural sets of values, defining three variants of HDSP.

Unrestricted model:	A packet can have any positive real value.
Real-valued model:	A packet can have any real value of at least 1.
Integral-valued model:	A packet can have any integral positive value.

Notice that the lower bound of 1 on the value of packets in the real-valued model is not an arbitrary choice. This lower bound comes up naturally as the revenue of a packet is reduced by 1 for each unit of time that it is delayed.

1.2 Results

We first consider the unrestricted model. We show that, unfortunately, even a randomized algorithm cannot have a constant competitive ratio for this model. This improves upon the lower bound of 3 proved by [10] for the integral-valued model, which currently is the best lower bound for the unrestricted model.

We then consider the real-valued model, for which we give a 4-competitive randomized algorithm as well as a matching lower bound (up to low order terms). The best previous algorithm for this model was a deterministic 4.9-competitive algorithm given by [10], whose competitive ratio proof was based on heavy numerical calculations. Using additional numerical calculations, [9] was able to fine tune the algorithm and improve this bound to 4.24. We also give an analytical proof of a deterministic lower bound of $\phi^3 - \varepsilon \approx 4.236$ (where $\phi = (\sqrt{5}+1)/2 \approx 1.618$ is the golden ratio and ε is an arbitrarily small positive constant) for the same model. This bound improves upon a deterministic lower bound of 3, and a lower bound of 4.1 for deterministic memory-less algorithms whose, again, depended on heavy numerical calculations [10]. The last bound was later improved to 4.23 by [9] using additional numerical calculations. Our result strengthens [10]'s conjecture that the "right" deterministic competitive ratio, for this model, is ϕ^3.

[2] We forbid packets from arriving at integral times to simply notation and avoid the need to decide whether a packet arriving at an integral time t can be transmitted in t (assuming the queue is empty when the packet arrives). Regardless of which decision is made, general arrival times can be easily reduced to non-integral arrival times.

	Unrestricted Model		Real Valued Model		Integral-Valued Model	
	Known Result	New Result	Known Result	New Result	Known Result	New Result
Deterministic Lower Bound	3	∞	3	$\phi^3 - \varepsilon \approx 4.236$	3	$4 - \varepsilon$
Deterministic Upper Bound	-	-	4.24	-	4.24	4
Randomized Lower Bound	$\phi \approx 1.618$	∞	$\phi \approx 1.618$	$4 - \varepsilon$	$\phi \approx 1.618$	$4 - \varepsilon$
Randomized Upper Bound	-	-	4.24	4	4.24	4

Table 1: Summary of known and new results. ε is an arbitrarily small positive constant. All known results in the table are inferred from [10] and [9].

Notice that randomization improves the achievable competitive ratio for the real-valued model from ϕ^3 to 4. On the other hand, randomization does not help the integral-valued model. For the integral-valued model, we give a deterministic 4-competitive algorithm and prove that this is best possible (up to low order terms), even for randomized algorithms. The previously known results for this model were a lower bound of 3 [10] and an upper bound of 4.24 [9].

Our positive results are achieved using the following technique. First, the set of possible inputs is reduced by showing a reduction from the general case to a more specialized set of inputs. Then, an algorithm is given for the specialized set of inputs. The algorithms we present are combinatorial, yet are analyzed via a linear program using the dual-fitting technique, *i.e.*, the value of OPT is bounded using an assignment to the dual of a linear relaxation of the problem.

Table 1 gives a short summary of the known results and our improvements. All known results in the table either come directly from [10] and [9] or can be immediately deduced from them. A preliminary version of this paper [8] stated the lower bounds of Table 1 without the ε term. We added the ε's in this version to emphasis that a lower bound of $c-\varepsilon$ does not rule out an algorithm whose competitive ratio is $c-f(n)$ for some function $f(n)$ of the number of packets that diminishes as n increases.

Beside the three models we consider, there is another natural model where all packets have equal value of R. For this model [10] gave a deterministic ϕ-competitive algorithm, and showed that this is best possible for large values of R, even for randomized algorithms.

1.3 Related Work

There is a very rich set of online buffering problems, modeling many different kinds of router architectures. We survey here only a few representative results and the ones that are most closely related to the specific problem we consider in this paper. The problem in which the algorithm is, perhaps, least restricted is the "bounded delay" problem. In this problem the router buffers all packets that it receives, and the algorithm simply has to choose which packet to transmit at each time slot. Each packet has a deadline, and it must be deleted from the router's buffer if not transmitted before its deadline. The objective is to maximize the total value of transmitted packets.

The best known competitive ratios for the "bounded delay" problem are a deterministic 1.828 and a randomized $e/(e-1) \approx 1.582$ (see [7, 3]). On the negative side, the known lower bounds for this problem are the golden ratio ($\phi \approx 1.618$) for deterministic algorithms and 5/4 for randomized algorithms [1, 4]. Assumptions on the input can often improve the attainable bounds. One such assumption is called "agreeable deadlines". This assumption simply says that if a packet d_1 arrives before packet d_2, then the deadline of d_2 cannot be before that of d_1. Under this assumption, there is a deterministic online algorithm with a competitive ratio of ϕ, matching the lower bound of [1] (which is still valid even with this assumption), and a randomized online algorithm with a competitive ratio of 4/3 [13]. Other common assumptions limit the possible lengths of the period between the arrival of a packet and its deadline (s-bounded inputs), or make this period equal for all packets (s-uniform inputs). See [12, 5] and the references in [7] for some results under these assumptions.

A more restrictive problem is the "preemptive FIFO" problem, in which the buffer is of limited size B and is managed as a FIFO buffer, *i.e.*, packets can only be transmitted in the order in which they arrive. Somewhat surprisingly, results for variants of this problem tend to have better competitive ratios than their corresponding "bounded delay" variants. The most general variant of this problem has deterministic lower and upper bounds of 1.419 and $\sqrt{3} \approx 1.732$ [6]. Restrictions on the input can help in this problem as well. For example, if the input is restricted to packets of two values 1 and α, then the deterministic lower and upper bounds can be improved to 1.281 and 1.303 [6].

The most restrictive problem that is often considered is the "non-preemptive FIFO" model. In this model the setting is exactly the same as in the previous problem, but the algorithm is not allowed to remove packets from the queue. Instead, the algorithm has to decide whenever it gets a new packet whether to accept or reject it. This new restriction has no implications on OPT, as there is no reason for OPT to accept a packet it does not intend to transmit. Therefore, the competitive ratios for variants of this problem are much worse in comparison to their counterparts for the previous problem. For example, if the input packets are restricted to the values 1 and α, then there is only a $(2\alpha - 1)/\alpha$ competitive algorithm, and this is tight for both deterministic and randomized algorithms [1]. If the packets are allowed to have any value in the range $[1, \alpha]$, then the situation becomes even worse; the best known deterministic lower and upper bounds for this variant are $1 + \ln \alpha$ and $e \cdot \lceil \ln \alpha \rceil$ by [1]. Notice that HDSP is also a variant of "non-preemptive FIFO" with two changes: the size of the buffer is unlimited, and the revenue of a packet changes as a function of the time it waits in the buffer.

Azar and Richter [2] proposed a model in which the router has multiple inputs and one output. Each input has its own FIFO buffer of size B which can be either preemptive or non-preemptive. Each time the router has an opportunity to transmit a packet, the algorithm has a choice from which input buffer to extract the packet. It was showed by [2] that any algorithm for a single buffer can be extended to this model, with a factor 2 loss in the competitive ratio.

The rest of this paper is organized as follows. Section 2 gives an impossibility result for the unrestricted model. Section 3 gives a reduction which is used to prove our positive results. Sections 4 and 5 give the positive results themselves: a 4-competitive deterministic algorithm for the integral-valued model and a 4-competitive randomized algorithm for the real-valued model. It is recommended to read Section 4 before Section 5 as the two sections use many common ideas, but the second one is more involved. Sections 6 and 7 give the remaining lower bounds: a lower bound of $\phi^3 - \varepsilon$ for deterministic algorithms in the real-valued model and a lower bound of $4 - \varepsilon$ for randomized algorithms in both the real-valued and the integral-valued models.

2 The Unrestricted Model

The unrestricted model is the most general model we consider. The only results known for this model are a lower bound of 3 for deterministic algorithms and a lower bound of $\phi \approx 1.618$ for randomized algorithms [10]. These lower bounds were proved for the integral-valued model and for a model where all packets have equal values, respectively, but they apply to the unrestricted model as well. We show that, unfortunately, no constant competitive ratio randomized algorithm for the unrestricted model exists even against an oblivious adversary. The rest of this section is devoted to proving this hardness result.

Let ALG be any randomized algorithm for HDSP, and let $c \geq 1$ be any integer constant. Algorithm 1 defines an oblivious adversary against which ALG is not c-competitive. The adversary gives ALG a series of increasing value packets. ALG must accept each one of these packets with some positive probability in order to be competitive. However, no packet in the sequence increases the expected value of ALG by much since ALG already accepted the previous packets of the sequence with a positive probability. The optimal solution OPT, on the other hand, has the privilege of accepting only the last packet in the series and obtaining its entire value.

Algorithm 1: Adversary for the Unrestricted Model

```
Let t_1, t_2, ..., t_{2c} be 2c arbitrary times such that 0 < t_1 < t_2 < ... < t_{2c} < 1.
for i = 1 to 2c do
    Give ALG a packet d_i of value (2c)^{i-2c} at time t_i.
    Let p_i be the probability ALG's queue is empty at this point.
    if p_i ≥ 1 - i c^{-1}/2 then Terminate.
```

Observe that the choice of arrival times in Algorithm 1 implies that all packets accepted by ALG are placed in the queue before the first one of them can be transmitted.

Theorem 2.1. *ALG is not c-competitive against the adversary given by Algorithm 1.*

Proof. For consistency of the notation, we define $p_0 = 1$. Assume the adversary generates $1 \leq \ell \leq 2c$ packets. If $1 < \ell < 2c$, then we must have:

$$p_\ell \leq 1 - \ell c^{-1}/2 \qquad \text{and} \qquad p_{\ell-1} \geq 1 - (\ell - 1)c^{-1}/2 \ . \tag{1}$$

The same inequalities also hold for $\ell = 1$ because $p_{\ell-1} = p_0 = 1 = 1 - (\ell - 1)c^{-1}/2$ and for $\ell = 2c$ because $p_\ell \geq 0 = 1 - \ell c^{-1}/2$.

The adversary generates only packets of value at most 1, thus, only the first packet accepted has a positive revenue. By the linearity of the expectation d_i is accepted into an empty queue (and therefore, has a positive revenue) with probability $p_i - p_{i-1}$. Thus, the revenue of ALG is:

$$\begin{aligned}
\sum_{i=1}^{\ell}(p_i - p_{i-1}) \cdot (2c)^{i-2c} &\leq p_{\ell-1} \cdot (2c)^{\ell-2c-1} + (p_\ell - p_{\ell-1}) \cdot (2c)^{\ell-2c} \\
&= [p_\ell - p_{\ell-1} \cdot (1 - c^{-1}/2)] \cdot (2c)^{\ell-2c} \\
&\leq [(1 - \ell c^{-1}/2) - (1 - (\ell-1)c^{-1}/2) \cdot (1 - c^{-1}/2)] \cdot (2c)^{\ell-2c} \\
&= [c^{-1} - \ell c^{-1}/2 - (\ell-1)c^{-2}/4] \cdot (2c)^{\ell-2c} < c^{-1} \cdot (2c)^{\ell-2c} \ ,
\end{aligned}$$

where the second inequality holds due to (1). Hence, ALG is not c-competitive because OPT can get a revenue of $(2c)^{\ell-2c}$ by accepting only d_ℓ. □

Remark: Theorem 2.1 holds only for constant values of c (*i.e.*, c cannot depend on the number of packets in the instance). The reason for that is that the number of packets in the instance produced by the adversary might depend on c. For deterministic algorithms, one can observe that an adversary sending at most two packets is sufficient to get an unbounded competitive ratio, and therefore, deterministic algorithms for the unbounded model have in fact an unbounded competitive ratio.

3 Reduction

Sections 4 and 5 present positive results for the integer-valued and real-valued models. To simply the exposition and analysis of these results, we give in this section a reduction that allows us to consider a slightly simpler problem. Intuitively, we show that deterministic algorithms that are r-competitive for a particular set of inputs called *one-slot* inputs can be transformed into algorithms that are r-competitive for arbitrary inputs. We also show that the same observation holds for randomized algorithms that obey a property we call *tight-concentration.*

We use the term *input sequence* to denote the packets arriving as well as their arrival times. An input sequence is called *one-slot* if all its packets arrive before the first integral time, *i.e.*, before the algorithm has an opportunity to transmit any of them. Consider an algorithm ALG for one-slot input sequences, and let Q be a random variable denoting the number of packets in the queue of ALG. We say that ALG is *tightly-concentrated* if Q is always equal to $\mathbb{E}[Q]$ up to rounding. More formally, ALG always obeys either $Q = \lfloor \mathbb{E}[Q] \rfloor$ or $Q = \lceil \mathbb{E}[Q] \rceil$. Observe that every deterministic algorithm is tightly-concentrated.

Reduction 1. *If A is a tightly-concentrated algorithm for one-slot input sequences with a competitive ratio of r against an oblivious adversary, then there exists an algorithm B achieving the same competitive ratio against an oblivious adversary for all input sequences. Moreover,*

- *If A is deterministic, then so is B.*
- *If A is r competitive against one-slot input sequences of the integer-valued/real-valued model, then B is r competitive against general input sequences of the same model.*

Proof. We first describe B, and then analyze its competitive ratio. Algorithm B uses an integral counter c, initially set to 0, and a simulation of A. When B receives a packet d, it feeds d to the simulation of A and accepts d if and only if the simulation of A accepts d. However, B makes some changes to the packet d before feeding it to the simulation of A. First, $a(d)$ is changed to be less than 1; this way A "observes" a one-slot input sequence. Second, $w(d)$ is increased by c. Each time an integral time arrives, B checks whether there is a positive probability that its queue is empty just before the integral time. If so (regardless of whether the queue is in fact empty), B sets c to 0 and resets the simulation of A. Otherwise, B increases c by 1.

Consider any input sequence σ. Let k be the number of times A's simulation is reset by B, and let $\sigma_{A,1}, \sigma_{A,2}, \ldots, \sigma_{A,k+1}$ denote the input sequences A's simulation receives from B between consecutive resets ($\sigma_{A,i}$ is equal to a sub-sequence of σ as far as the order in which the packets arrive is concerned, but is different from this sub-sequence in terms of the times in which the packets arrive and their values). Notice that the input received by the simulation of A does not depend on the random choices A makes (it depends only on the probabilities of these choices), and thus, $\sigma_{A,1}, \sigma_{A,2}, \ldots, \sigma_{A,k+1}$ are determined completely by σ. Thus, the simulations of A indeed face an oblivious adversary. Hence, by the definition of A, $r \cdot A(\sigma_{A,i}) \geq OPT(\sigma_{A,i})$.

Consider an arbitrary integral time t in which B resets the simulation of A. Since B reset the simulation, we learn that it had a positive probability to have an empty queue just before t. Since A is strongly-concentrated, B is guaranteed to have at most one packet in its queue just before t, and thus, its queue is guaranteed to be empty immediately after t. In other words, B's queue is always empty immediately after a reset.

Assume OPT decides to accept from $\sigma_{A,i}$ the same packets it would have accepted from the corresponding sub-sequence of σ, had σ been the input sequence. Let c_d be the value of the counter c when packet d is received. Each packet $d \in \sigma_{A,i}$ appears in $\sigma_{A,i}$ before the first integral time, whereas in σ there are c_d integral times between the first packet of $\sigma_{A,i}$ and the appearance of d. Therefore, it is easy to see that d suffers in $OPT(\sigma_{A,i})$ at most an additional delay of c_d (in comparison to its delay when OPT receives σ). However, d's value is larger in $\sigma_{A,i}$ by c_d, hence, the revenue of d in $OPT(\sigma_{A,i})$ is at least as large as its revenue in $OPT(\sigma)$. This is true for every packet $d \in \sigma_{A,i}$, hence, $\sum_{i=1}^{k+1} OPT(\sigma_{A,i}) \geq OPT(\sigma)$.

Combining the last two results we have $r \cdot \sum_{i=1}^{k+1} A(\sigma_{A,i}) \geq OPT(\sigma)$. To complete the proof we only need to show that B gets from every packet d at least the same revenue as the simulation of A. Let $\sigma_{A,i}$ be the input sub-sequence containing d and c_d be the value of the counter c when d is received. Since B transmits a packet at each integral time in which A's simulation is not reset, B's queue must be shorter by at least c_d packets than A_d's queue when d is received. Consequently, d suffers a delay shorter by at least c_d under B than under A's simulation. Note that $w(d)$ is larger in $\sigma_{A,d}$ only by c_d than in σ, thus, B's revenue from accepting d is at least as large as the simulation's revenue from accepting this packet. Since B accepts d if and only if the simulation does so, B's revenue from d is always at least as large as the simulation's. □

4 A 4-Competitive Algorithm for the Integral-Valued Model

The integral-valued model is simpler than the real-valued one, but the best algorithm known for both models is the 4.24-competitive algorithm of [10]. In this section we present a 4-competitive deterministic algorithm for the integral-valued model. This might not seem like much of an improvement, yet Section 7 shows that this is best possible, even for randomized algorithms. To simplify the exposition and analysis of the algorithm, we assume one-slot input sequences throughout this section. By Reduction 1, the same result holds also for general input sequences.

Consider Algorithm 2.[3] Notice that Algorithm 2 is strongly-concentrated since it is deterministic.

Algorithm 2: Nearly Doubling Threshold (NDT)

Let $Q \leftarrow 0$. `/*` Q `denotes the number of packets in the the queue. */`
foreach *packet d arriving* **do**
    if $w(d) \geq 2Q + 1$ **then**
        Accept d.
        Let $Q \leftarrow Q + 1$.

We start the analysis of Algorithm 2 by showing that we can further reduce the set of inputs we need to consider (beyond our restriction to one-slot input sequences). Let Q_d be the number of packets in the queue of NDT when packet d arrives. A one-slot input sequence is NDT-based if

[3] Algorithm 2 is inspired by the 5.25-competitive DT algorithm presented by [10].

for every packet d, $w(d) \leq 2Q_d + 1$. The following reduction shows that it is enough to prove that NDT is 4-competitive for NDT-based input sequences.

Reduction 2. *If* NDT *is r-competitive for* NDT*-based input sequences, then it is r-competitive for one-slot input sequences.*

Proof. Given a one-slot input sequence σ, let σ' be an identical input sequence in which the value of every packet d is changed to $w'(d) = \min\{w(d), 2Q_d + 1\}$. By the definition of NDT it accepts the same set of packets D given either σ or σ'. Thus, $\mathsf{NDT}(\sigma) - \mathsf{NDT}(\sigma') = \sum_{d \in D}[w(d) - \min\{w(d), 2Q_d + 1\}]$.

Let us compare $OPT(\sigma)$ and $OPT(\sigma')$. One possible option for $OPT(\sigma')$ is to accept the same packets $OPT(\sigma)$ accept. Hence, $OPT(\sigma')$ is at least as large as $OPT(\sigma)$ minus the total decrease in packet values during the transition from σ to σ'. Let us calculate this loss. The value of a packet $d \in D$ is reduced by $w(d) - \min\{w(d), 2Q_d + 1\}$. On the other hand, a packet $d \notin D$ obeys, by definition of NDT, $w(d) < 2Q_d + 1$. Thus, such packets do not suffer any value loss. Adding these observations together gives:

$$OPT(\sigma) - OPT(\sigma') \leq \sum_{d \in D}[w(d) - \min\{w(d), 2Q_d + 1\}] \ .$$

Since NDT is r competitive for NDT-based input sequences, we get:

$$\begin{aligned}\mathsf{NDT}(\sigma) &= \mathsf{NDT}(\sigma') + \sum_{d \in D}[w(d) - \min\{w(d), 2Q_d + 1\}] \\ &\geq \frac{OPT(\sigma')}{r} + \sum_{d \in D}[w(d) - \min\{w(d), 2Q_d + 1\}] \\ &\geq \frac{OPT(\sigma') + \sum_{d \in D}[w(d) - \min\{w(d), 2Q_d + 1\}]}{r} \geq \frac{OPT(\sigma)}{r} \ . \end{aligned}$$

□

The analysis of Algorithm 2 for NDT-based input sequences uses a dual-fitting argument. The following LP formulation represents the off-line version of HDSP. Let $A(d)$ be the set of times in which packet d can be sent, *i.e.*, $A(d)$ consists of all integer times between $a(d)$ and $w(d) + a(d)$. Variable $y(d,t)$ is an indicator for the event that packet d is transmitted in integral time t.

$$\begin{array}{llllll} (\texttt{LP1}) & \max & \sum_d \sum_{t \in A(d)} (w(d) - \lfloor t - a(d) \rfloor) \cdot y(d,t) & & & \\ & & \sum_{t \in A(d)} y(d,t) & \leq 1 & \forall\, d & \text{(packet constraint)} \\ & & \sum_{d | t \in A(d)} y(d,t) & \leq 1 & \forall\, t & \text{(time constaint)} \\ & & y(d,t) & \geq 0 & \forall\, d, t \in A(d) & \end{array}$$

The coefficient of each variable $y(d,t)$ in the objective function is the value of the corresponding packet ($w(d)$) minus the number of integral times it will spend in the queue if transmitted at time t. The packet constraints make sure that a packet is transmitted at most once. The time constraints allow at most one packet to be transmitted at each integral time. Notice that no constraint enforces the FIFO transmission order, however, this is not a problem because we only need the optimal value of $\texttt{LP1}$ to upper bound OPT (in fact, the FIFO requirement does not affect OPT, and therefore, adding a corresponding constraint to the LP will not make it any more powerful).

The dual of $\texttt{LP1}$ is:

$$\begin{array}{lllll} (\texttt{LP2}) & \min & \sum_t x_t + \sum_d z_d & & \\ & & x_t + z_d & \geq w(d) - \lfloor t - a(d) \rfloor & \forall\, d, t \in A(d) \\ & & x_t, z_d & \geq 0 & \forall\, d, t \end{array}$$

By weak duality, every feasible solution for LP2 is an upper bound on the optimal solution for LP1, and therefore, also on OPT. Thus, if ALG is an on-line algorithm for HDSP and for any NDT-based input sequence σ there exists a feasible solution for LP2 of cost $r \cdot A(\sigma)$, then ALG is r-competitive on NDT-based input sequences. The rest of this section is devoted for constructing a solution for LP2 of cost at most $4 \cdot \mathsf{NDT}(\sigma)$ for an arbitrary NDT-based input sequence σ, proving that NDT is 4-competitive for any NDT-based input sequence. Given an input sequence σ, we denote by $\mathtt{LP2}(\sigma)$ the instance of LP2 resulting from this input sequence, and by $\hat{Q}_\sigma$ the number of packets accepted by $\mathsf{NDT}(\sigma)$ (in other words, $\hat{Q}_\sigma$ is the final value of Q in NDT when the input is σ).

Lemma 4.1. $\mathsf{NDT}(\sigma) = \hat{Q}_\sigma(\hat{Q}_\sigma + 1)/2$.

Proof. Let d be the i^{th} packets accepted by NDT when it receives σ. Since σ is a one-slot input sequence, when d arrives NDT has $i-1$ packets in its queue. In addition, since σ is NDT-based, the value d is $2(i-1)+1$. Therefore, the revenue NDT gets from d is $[2(i-1)+1]-(i-1) = i$. Summing over all packets accepted by NDT, we get: $\mathsf{NDT}(\sigma) = \sum_{i=1}^{\hat{Q}_\sigma} i = \hat{Q}_\sigma(\hat{Q}_\sigma+1)/2$. □

Consider the a dual solution for $\mathtt{LP2}(\sigma)$ defined as following. Variables of type z_d are assigned a value of 0, and variables of type x_t are assigned a value of $\max\{2\hat{Q}_\sigma + 1 - t, 0\}$.

Lemma 4.2. *The above dual solution of* LP2 *is feasible and its cost is* $\hat{Q}_\sigma(2\hat{Q}_\sigma + 1)$.

Proof. Consider an arbitrary constraint $x_t + z_d \geq w(d) - \lfloor t - a(d) \rfloor$ of LP2. Since σ is a one-slot input sequence $a(d) \in (0,1)$, which implies $\lfloor t - a(d) \rfloor = t - 1$. Since σ is also NDT-based:

$$w(d) \leq \begin{cases} 2Q_d + 1 \leq 2\hat{Q}_\sigma - 1 & \text{if } d \text{ is accepted by } \mathsf{NDT} \ , \\ 2Q_d \leq 2\hat{Q}_\sigma & \text{otherwise} \ . \end{cases}$$

Plugging the above bounds into the constraint, we get: $x_t + z_d \geq 2\hat{Q}_\sigma + 1 - t$, which trivially holds for the above dual solution.

The only variables of the dual solution which get a non-zero assignment are $x_1, x_2, \ldots, x_{2\hat{Q}_\sigma}$. The coefficients of all these variables in the objective function is 1, hence, the cost of the dual solution is:

$$\sum_{t=1}^{2\hat{Q}_\sigma} [2\hat{Q}_\sigma + 1 - t] = \frac{2\hat{Q}_\sigma(2\hat{Q}_\sigma + 1)}{2} = \hat{Q}_\sigma(2\hat{Q}_\sigma + 1) \ .$$ □

Corollary 4.3. NDT *is a* 4*-competitive algorithm for* NDT*-based input sequences, and thus, there exists a* 4*-competitive deterministic algorithm for general input sequences of* HDSP *in the integral-valued model.*

Proof. The ratio between the cost of the above dual solution and $\mathsf{NDT}(\sigma)$ is:

$$\frac{\hat{Q}_\sigma(2\hat{Q}_\sigma + 1)}{\hat{Q}_\sigma(\hat{Q}_\sigma + 1)/2} = \frac{4\hat{Q}_\sigma + 2}{\hat{Q}_\sigma + 1} \leq 4 \ .$$ □

5 A 4-Competitive Randomized Algorithm for the Real-Valued Model

Fiat et al. [10] prove that randomization does not help when all packets have equal values, but leaves open the question of the usefulness of randomization in models that allow multiple packet values. Sections 4 and 7 show that for the integral-valued model, a deterministic algorithm can achieve the best possible competitive ratio. Surprisingly, the real-valued model is different. This section presents a 4-competitive randomized algorithm for the real valued model; bypassing the lower bound for deterministic algorithms described in Section 6. However, Section 7 shows that this is as far as randomization can take us – no randomized algorithm can do better.

We would like to limit ourselves to one-slot input sequences using Reduction 1. In a sense, Reduction 1 requires an algorithm for one-slot input sequences with somewhat limited randomness – the size of the algorithm's queue must always be what it is expected to be (up to rounding). On the other hand, some randomness is required because of the lower bound for deterministic algorithms proved in Section 6. The compromise between these requirements is Algorithm 3 (RNDT). We show that RNDT is 4-competitive tightly-concentrated algorithm for one-slot input sequences, implying a 4-competitive randomized algorithm for general input sequences, together with Reduction 1.

Algorithm 3: Randomized Nearly Doubling Threshold (RNDT)

`// Initialization`
Let $Q \leftarrow 0$.
Let s be a uniformly random number from the range $[0, 1)$.
`// Main Loop`
foreach *packet d arriving* **do**
  if *d is the first packet* **then** Let $p_d = 1/2$.
  else Let $p_d = \max\{0, \min\{1, w(d)/2 - Q - 0.25\}\}$.
  Let $Q_d \leftarrow Q$.
  Update $Q \leftarrow Q + p_d$.
  if *$p_d > 0$ and the range $(Q_d, Q]$ contains a point q such that $q - \lfloor q \rfloor = s$* **then**
    Accept d.

Before analyzing RNDT, let us give a short intuitive explanation of the underlying ideas. Recall that NDT is 4-competitive for the integral-valued model. In the analysis of NDT, we used the integrality of packet values only in the proof of Lemma 4.2, where we needed the observation that if a packet d is rejected (*i.e.*, it obeys $w(d) < 2Q_d + 1$, where Q_d is the queue size before d arrives), then it must obey $w(d) \leq 2\hat{Q}_\sigma$, where $\hat{Q}_\sigma$ is the final size of the queue. To get a similar property to hold in a non-integral setting, RNDT accepts a packet with probability $p_d = \max\{0, \min\{1, w(d)/2 - Q_d - 0.25\}\}$, where Q_d is the value of Q before d arrives. Observe that this guarantees $w(d) \leq 2\mathbb{E}[Q_\sigma] + 0.5$ for every packet d which is not accepted with probability 1, where Q_σ is the final value of Q given the input σ.

A naive implementation of the above idea would result in an algorithm which is not tightly-concentrated. To overcome this issue, RNDT uses dependent rounding. The algorithm creates an interval of size p_d for every packet d, and packs these intervals in the positive real numbers axis. A packet is then accepted into the queue if and only if its interval contains a point q whose fractional part is equal to s, where s is a uniformly chosen value from the range $[0, 1)$. Observation 5.1 shows that this rounding method results in a tightly-concentrated algorithm.

Observation 5.1. RNDT *is tightly-concentrated.*

Proof. Consider the state of RNDT at an arbitrary point, and let $s' = Q - \lfloor Q \rfloor$. Exactly one of the following must happen:

- With probability s', $s < s'$. In this case, there are $\lceil Q \rceil$ packets in the queue of the algorithm.

- With probability $1 - s'$, $s \geq s'$. In this case, there are $\lfloor Q \rfloor$ packets in the queue of the algorithm.

Hence, if Q is an integral, the number of packets in the queue of RNDT is exactly Q. Otherwise, the expected number of packets in the queue of RNDT is:

$$s' \cdot \lceil Q \rceil + (1 - s') \cdot \lfloor Q \rfloor = s' \cdot (Q - s' + 1) + (1 - s') \cdot (Q - s') = Q \ . \qquad \square$$

Let σ be an arbitrary one-slot input sequence, and let $\hat{Q}_\sigma$ denote the final value of Q in RNDT given σ. We say that a one-slot input sequence σ is RNDT-based input sequence if the weight of the first packet of σ is 1 and no packet of σ has a weight larger than $2\hat{Q}_\sigma + 0.5$. Reduction 3 is the counterpart of Reduction 2 from Section 4.

Reduction 3. *If* RNDT *is r-competitive for* RNDT*-based input sequences for some $r \geq 2$, then it is r-competitive for one-slot input sequences.*

Proof. Given a one-slot input sequence σ, let d_1 be its first packet. Let σ' be an input sequence with the same packets as σ in which the value of every packet d is changed as following. If $d = d_1$, then $w'(d) = 1$. Otherwise, $w'(d) = \min\{w(d), 2\bar{Q}_d + 2.5\}$. RNDT accepts d_1 with probability $1/2$ regardless of its value. Every other packet d whose value is changed is accept with probability 1. Thus, RNDT behaves the same (*i.e.*, given the same s it will accept the same packets) given either σ or σ'. Hence, $\mathbb{E}[\mathsf{RNDT}(\sigma) - \mathsf{RNDT}(\sigma')] = \sum_{d \in D \setminus \{d_1\}} [w(d) - w'(d)] + [w(d_1) - w'(d_1)]/2$, where D is the set of packets whose value is changed during the construction of σ'.

Let us compare $OPT(\sigma)$ and $OPT(\sigma')$. One possible option for $OPT(\sigma')$ is to accept the same packets $OPT(\sigma)$ accept. Hence, $OPT(\sigma) - OPT(\sigma') \leq \sum_{d \in D \cup \{d_1\}} [w(d) - w'(d)]$. If σ' is RNDT-based input sequence, then since RNDT is r competitive for RNDT-based input sequences:

$$\begin{aligned}
\mathsf{NDT}(\sigma) &= \mathsf{NDT}(\sigma') + \sum_{d \in D \setminus \{d_1\}} [w(d) - w'(d)] + [w(d_1) - w'(d_1)]/2 \\
&\geq \frac{OPT(\sigma')}{r} + \sum_{d \in D \setminus \{d_1\}} [w(d) - w'(d)] + [w(d_1) - w'(d_1)]/2 \\
&\geq \frac{OPT(\sigma') + \sum_{d \in D \cup \{d_1\}} [w(d) - w'(d)]}{r} \geq \frac{OPT(\sigma)}{r} \ .
\end{aligned}$$

To complete the proof, we just need to prove that σ' is an RNDT-based input sequence. Consider some packet d. If $d = d_1$, then it is accepted with probability half, and thus, $\hat{Q}_{\sigma'} \geq 1/2$. Consequently, $w'(d) = 1 < 2 \cdot 1/2 + 0.5 \leq 2\hat{Q}_{\sigma'} + 0.5$. Otherwise, either $w'(d) = w(d) \leq 2Q_d + 0.5 \leq 2\hat{Q}_{\sigma'} + 0.5$, or $w(d) = 2\bar{Q}_d + 0.5 + c$ for some $0 < c \leq 2$. In the second case, d d is accepted by RNDT with probability $p_d = w(d)/2 - Q_d - 0.25 = c/2$. Hence:

$$w'(d) = 2Q_d + 0.5 + c \leq 2(\hat{Q}_{\sigma'} - p_d) + 0.5 + c = 2\hat{Q}_{\sigma'} + 0.5 \ . \qquad \square$$

The rest of the analysis of RNDT goes along the same lines as the analysis of NDT in Section 4, *i.e.*, we consider a RNDT-based input sequence σ and show that $\mathtt{LP2}(\sigma)$ has a solution of cost at most $4 \cdot E[\mathsf{RNDT}(\sigma)]$.

Lemma 5.2. *The expected revenue for* RNDT *from a packet d is at least $p_d(w(d) - Q_d)$.*

Proof. The claim is clearly true for every packet with $p_d = 0$. Hence, we may assume $p_d > 0$. Consider the range $R = (Q_d, Q_d + p_d]$. If this range contains no integral points or $Q_d + p_d$ is the sole integral point in this range, then d is accepted with probability p_d, and when it is accepted is suffers a delay of $\lfloor Q_d \rfloor$. Hence, the revenue of d is:

$$p_d(w(d) - \lfloor Q_d \rfloor) \geq p_d(w(d) - Q_d) \ .$$

If R contains an integral point other than $Q_d + p_d$, then there are two cases in which d is accepted. Let $q = Q_d - \lfloor Q_d \rfloor$. If s happens to fall within the range $(q, 1]$ (which happens with probability $1 - q$), then d is accepted and it suffers a delay of $\lfloor Q_d \rfloor$. On the other hand, if s happens to fall within the range $(0, p_d + q - 1]$ (which happens with probability $p_d + q - 1$), then d is accepted and it suffers a delay of $\lceil Q_d \rceil$. Hence, the expected revenue of RNDT from d is:

$$\begin{aligned}
&(1-q)(w(d) - \lfloor Q_d \rfloor) + (p_d + q - 1)(w(d) - \lceil Q_d \rceil) \\
=&(1-q)(w(d) - Q_d + q) + (p_d + q - 1)(w(d) - Q_d + q - 1) \\
=&p_d(w(d) - Q_d) + (1 - p_d)(1 - q) \geq p_d(w(d) - Q_d) \ . \qquad \square
\end{aligned}$$

Lemma 5.3. *The expected revenue of* RNDT *is at least* $\frac{(2\hat{Q}_\sigma - 1)^2}{8}$*, unless* σ *contains no packets.*

Proof. Let d_i denote the i^{th} packet accepted by RNDT with a positive probability, and let m be the number of packets accepted with a positive probability. We abuse notation and let p_i and Q_i denote p_{d_i} and Q_{d_i}, respectively. Since $w(d_1) = 1$ (recall that σ is a RNDT-based input seequence) and $p_1 = 1/2$:

$$\begin{aligned}
\sum_{i=1}^{m} p_i(w(d_i) - Q_i) &= \frac{1}{2} + \sum_{i=2}^{m} p_i(w(d_i) - Q_i) \geq \frac{1}{2} + \sum_{i=2}^{m} \left(Q_i + 2p_i + \frac{1}{2}\right) \\
&\geq \frac{1}{2} + \sum_{i=2}^{m} \left(p_i \cdot Q_i + \frac{p_i^2}{2} + \frac{p_i}{2}\right) = \frac{1}{2} + \sum_{i=2}^{m} \left[\int_{Q_i}^{Q_i + p_i} \left(x + \frac{1}{2}\right)\right] \\
&= \frac{1}{2} + \int_{1/2}^{\hat{Q}_\sigma} \left(x + \frac{1}{2}\right) dx = \frac{1}{2} + \left.\frac{x^2 + x}{2}\right|_{1/2}^{\hat{Q}_\sigma} = \frac{(2\hat{Q}_\sigma - 1)^2}{8} \ .
\end{aligned}$$

Let us explain why the first inequality holds. By definition, d_i was accepted with positive probability, *i.e.*, $p_i > 0$. If $p_i = 1$ then $w(d_i) \geq 2Q_i + 2.5 = 2Q_i + 2p_i + 0.5$. If $0 < p_i < 1$ then:

$$p_i = w(d_i)/2 - 0.25 - Q_i \Rightarrow w(d_i) = 2Q_i + 2p_i + 0.5 \ .$$

This completes the proof of the first inequality. The lemma now follows immediately since by Lemma 5.2 the revenue of each packet d_i is at least $p_i(w(d_i) - Q_i)$. $\square$

Consider the following dual solution for LP2(σ). Variables of type z_d are assigned a value of 0 and variables of type x_t are assigned a value of $\max\{2\hat{Q}_\sigma + 1.5 - t, 0\}$.

Lemma 5.4. *The above dual solution of* LP2(σ) *is feasible and its cost is* $(2E_f + 1)^2/2$.

Proof. Consider an arbitrary constraint $x_t + z_d \geq w(d) - \lfloor t - a(d) \rfloor$ of LP2. Since σ is a one-slot input sequence $a(d) \in (0, 1)$, which implies $\lfloor t - a(d) \rfloor = t - 1$. Since σ is also RNDT-based, $w(d) \leq 2\hat{Q}_\sigma + 0.5$. Plugging the above bounds into the constraint, we get: $x_t + z_d \geq 2\hat{Q}_\sigma + 1.5 - t$, which trivially holds for the above dual solution.

The only variables of the dual solution which get a non-zero assignment are $x_1, x_2, \ldots, x_{2\hat{Q}_\sigma + 1}$. The coefficients of all these variables in the objective function is 1, hence, the cost of the dual solution is:

$$\sum_{t=1}^{2\hat{Q}_\sigma + 1} [2\hat{Q}_\sigma + 1.5 - t] = \frac{(2\hat{Q}_\sigma + 1)^2}{2} \ . \qquad \square$$

Corollary 5.5. RNDT *is a* 4*-competitive algorithm for* RNDT*-based input sequences, and thus, there exists a* 4*-competitive randomized algorithm for general input sequences of* HDSP *in the real-valued model.*

Proof. The ratio between the cost of the above dual solution and $\mathsf{RNDT}(\sigma)$ is:

$$\frac{(2\hat{Q}_\sigma + 1)^2/2}{(2\hat{Q}_\sigma + 1)^2/8} = 4 \ . \qquad \square$$

6 Lower Bound for Deterministic Algorithms in the Real Valued Model

The previously best known lower bound for deterministic algorithms in the real valued model is 3 [10]. In this section we show an improved lower bound of $\phi^3 - \varepsilon \approx 4.236$ (where $\phi = (\sqrt{5}+1)/2$ is the golden ratio and ε is an arbitrary small positive constant). The proof uses the same technique used by [10] for proving a lower bound of 4.1 for memory-less deterministic algorithms. Given a constant $\beta > 1$, consider the following recursive series:

$$b_{\beta,k} = \begin{cases} 1 & \text{if } k = 0 \ , \\ \min\left\{x \in [1,\infty) \mid \sum_{j=0}^{\lfloor x \rfloor}(x-j) \geq \beta \cdot \sum_{j=0}^{k-1}(b_{\beta,j} - j)\right\} & \text{otherwise} \ . \end{cases}$$

Notice that the definition of $b_{\beta,k}$ guarantees that if an algorithm accepts one packet of each value $b_{\beta,0}, b_{\beta,1}, \ldots, b_{\beta,k-1}$ (in that order, without any of them being transmitted in the mean time), then it will get a revenue lower by at least a factor of β compared to a scenario where it accepts $\lfloor b_{\beta,k} \rfloor + 1$ packets of value $b_{\beta,k}$. The following observation gives a useful property of $b_{\beta,k}$.

Observation 6.1. $b_{\beta,k}$ *is a non-decreasing function of both* β*.*

Proof. We prove the lemma by induction on k. For $k = 0$, the lemma is trivial since $b_{\beta,0} = 1$ for any β. Assume the lemma holds for every $k' < k$, and let us prove it for k. Let $\beta' \in (1, \beta)$. By the induction hypothesis:

$$\sum_{j=0}^{k-1}(b'_{\beta,j} - j) \leq \sum_{j=0}^{k-1}(b_{\beta,j} - j) \ .$$

On other other hand $\sum_{j=0}^{\lfloor x \rfloor}(x-j)$ is a non-negative increasing function of x. Hence,

$$\begin{aligned} b_{\beta,k} = &\min\left\{x \in [1,\infty) \mid \sum_{j=0}^{\lfloor x \rfloor}(x-j) \geq \beta \cdot \sum_{j=0}^{k-1}(b_{\beta,j} - j)\right\} \\ \geq &\min\left\{x \in [1,\infty) \mid \sum_{j=0}^{\lfloor x \rfloor}(x-j) \geq \beta' \cdot \sum_{j=0}^{k-1}(b_{\beta',j} - j)\right\} = b_{\beta',k} \ . \end{aligned} \qquad \square$$

It is proved by [10] that if there exists a k obeying $b_{\beta,k} < k$, then no memoryless deterministic algorithm can be better than β-competitive in the real valued model.[4] We prove that this result extends to non-memoryless algorithms. Moreover, we show that for every $\beta < \phi^3$, there exists a k obeying $b_{\beta,k} < k$, hence, no deterministic algorithm is $\phi^3 - \varepsilon$ competitive in the real valued model for any constant $\varepsilon > 0$.

Algorithm 4: Adversary for Deterministic Algorithms in the Real Valued Model

// All packets generated by this adversary are given to ALG during the time range $(0, 1)$ in the order in which they are generated.

for $k = 0$ **to** ∞ **do**
  Let $i_k \leftarrow 0$.
  repeat
    Generate a packet of value $b_{\beta,k}$.
    Update $i_k \leftarrow i_k + 1$.
  until *ALG accepted a packet of value* $b_{\beta,k}$ *or* $i_k = \lfloor b_{\beta,k} \rfloor + 1$
  if *ALG accepted no packet of value* $b_{\beta,k}$ **then** Terminate.

Let ALG be any deterministic online algorithm for HDSP in the real valued model, and consider the adversary described by Algorithm 4.

The intuition behind the adversary given by Algorithm 4 is as following. If ALG is better than β-competitive, then it must accept a packet in every iteration (otherwise, by the definition of $b_{\beta,k}$, OPT can accept only the $\lfloor b_{\beta,k} \rfloor + 1$ packets of value $b_{\beta,k}$ and be at least β times better than ALG). However, if $k < b_{\beta,k}$ then the revenue ALG gets from the packet it accepts on the k^{th} iteration is negative $(b_{\beta,k} - k)$, so it cannot help ALG anyway.

Notice that Algorithm 4 gives to ALG all the packets it generates before A has an opportunity to transmit any of them. This is consistent with Reduction 1, which shows that one can give all packets before the first integral time without affecting the hardness of the problem. We begin the proof by showing that if $k < b_{\beta,k}$ for some k, then no deterministic algorithm is better than β-competitive in the real valued model.

Lemma 6.2. *If ALG is β'-competitive, for some constant $\beta' < \beta$, then the adversary given by Algorithm 4 never terminates.*

Proof. Assume, for the sake of contradiction, that adversary stops after its k^{th} iteration, *i.e.*, immediately after giving $\lfloor b_{\beta,k} \rfloor + 1$ packets of value $b_{\beta,k}$ to ALG. By the definition of adversary, ALG must have accepted exactly one packet in each previous iteration. Hence, the content of A's queue at the beginning of the k^{th} iteration is $b_{\beta,0}, b_{\beta,1}, \ldots, b_{\beta,k-1}$.

Since ALG does not accept any packet given during the k^{th} iteration (otherwise, the adversary would not have stopped after that iteration), its revenue is $\sum_{i=0}^{k-1}(b_{\beta,i} - j)$. On the other hand OPT can accept the $\lfloor b_{\beta,k} \rfloor + 1$ packets of value $b_{\beta,k}$ generated during the k^{th} iteration, and get a revenue of $\sum_{i=0}^{\lfloor b_{\beta,k} \rfloor}(b_{\beta,k} - j)$. The ratio between our bounds on the revenue of OPT and the revenue of ALG is at least β because:

- For $k > 0$ this follows from the definition of $b_{\beta,k}$.
- For $k = 0$ this holds since our bound on the revenue of OPT is $\sum_{i=0}^{\lfloor b_{\beta,k} \rfloor}(b_{\beta,k} - j) = b_{\beta,k} = 1$ and our bound on the revenue of ALG is $\sum_{i=0}^{k-1}(b_{\beta,i} - j) = 0$. □

Theorem 6.3. *If there exists $k \geq 0$ obeying $b_{\beta,k} < k$, then no deterministic algorithm is β'-competitive in the real valued model for any constant $\beta' < \beta$.*

Proof. Assume, for the sake of contradiction, that ALG is β'-competitive algorithm. Clearly, we can assume ALG never accepts packets of negative revenue (otherwise, we consider an algorithm

[4] The definition of $b_{\beta,k}$ in [10] is slightly different, but both definitions use the same idea.

ALG' which simulated ALG, but rejects packets of negative revenue that ALG accepts – clearly ALG' is at least as good as ALG given any input sequence, and therefore, is also β'-competitive).

Consider the behavior of ALG when faced with the adversary described by Algorithm 4. Lemma 6.2 implies that the adversary will never stop. By the definition of the adversary, this implies that ALG accepts exactly one packet of value $b_{\beta,k'}$ for every $k' \geq 0$. Thus, ALG accepts exactly one packet d of value $b_{\beta,k}$, and when it does so, there are k previous packets in its queue. Hence, the revenue of ALG from d is $b_{\beta,k} - k < 0$, which contradicts our assumption that ALG accepts no packets of negative revenue. □

The rest of this section is devoted for proving Theorem 6.4. Notice that the hardness we want to prove, namely Corollary 6.5, follows immediately from Theorems 6.3 and 6.4.

Theorem 6.4. *For any $\beta < \phi^3$, there exists a k obeying $b_{\beta,k} < k$.*

Corollary 6.5. *No deterministic algorithm is $\phi^3 - \varepsilon$ competitive for any constant $\varepsilon > 0$ in the real-valued model.*

We prove Theorem 6.4 by showing that there exists a linear function (in k) of slope in the range $(0, 1)$ upper bounding the $b_{\beta,k}$'s. This is done in three steps:

- We first prove the $b_{k,\beta}$'s are upper bounded by a linear function.
- We then show that if the $b_{k,\beta}$'s are upper bounded by a linear function of some slop, then they are also upper bounded by another linear function with a significantly smaller slope.

Let us start the first step of the proof by introducing a new recursive series. Let $a = 0.795$ and $b = 0.81$.

$$c_k = \begin{cases} 1 & \text{for } k = 0 \ , \\ \phi + a & \text{for } k = 1 \ , \\ 2\phi + b & \text{for } k = 2 \ , \\ k\phi + a & \text{for } k > 2 \ . \end{cases}$$

Notice that the c_k's are clearly upper bounded by a linear function of slope ϕ. The next Lemma 6.7 shows that the c_k's upper bound the $b_{\beta,k}$'s (and therefore, the $b_{\beta,k}$'s are also upper bounded by a linear function of slope ϕ).

Observation 6.6. *For every integer $k > 5$,*

$$\frac{(k\phi + a)(k\phi + a + 1)}{2} \geq \phi^3 \cdot \sum_{i=0}^{k-1} (c_i - i) \ . \tag{2}$$

Proof. The right hand side of (2) is:

$$\phi^3 \cdot \sum_{i=0}^{k-1} (c_i - i) = \phi^3 \cdot \left(1 + b - 2a + \sum_{i=0}^{k-1} (i\phi + a - i)\right) = \phi^3(1 + b - 2a + ka) + \phi^3 \cdot \sum_{i=0}^{k-1} \frac{i}{\phi}$$
$$= \phi^3(1 + b - 2a + ka) + \frac{\phi^2 k(k-1)}{2} \ .$$

On the other hand, the left hand side of (2) is:

$$\frac{(k\phi + a)(k\phi + a + 1)}{2} = \frac{k^2\phi^2 + 2k\phi a + a^2 + k\phi + a}{2} \ .$$

Thus, the observation holds if:

$$\begin{aligned}
&\frac{k^2\phi^2 + 2k\phi a + a^2 + k\phi + a}{2} \geq \phi^3(1 + b - 2a + ka) + \frac{\phi^2 k(k-1)}{2} \\
\Leftrightarrow\ & 2k\phi a + k\phi + k\phi^2 - 2k\phi^3 a \geq 2\phi^3(1 + b - 2a) - a^2 - a \\
\Leftrightarrow\ & k \geq \frac{2\phi^3(1 + b - 2a) - a^2 - a}{2\phi a + \phi + \phi^2 - 2\phi^3 a} \ ,
\end{aligned}$$

which is true for integer $k > 5$ since

$$\frac{2\phi^3(1 + b - 2a) - a^2 - a}{2\phi a + \phi + \phi^2 - 2\phi^3 a} \approx 5.962 \ . \qquad \square$$

Lemma 6.7. *For every $k \geq 0$ and $\beta \leq \phi^3$, $c_k \geq b_{\phi^3,k} \geq b_{\beta,k}$.*

Proof. The inequality $b_{\phi^3,k} \geq b_{\beta,k}$ follows from Observation 6.1. The proof of the inequality $c_k \geq b_{\phi^3,k}$ is done by induction on k. For $k = 0$, $b_{\phi^3,0} = 1 = c_0$ so there is nothing to prove. Assume the lemma holds for $k - 1 \geq 0$, and let us prove it for k. Since $b_{\phi,k}$ is the lowest number (not smaller than 1) obeying $\sum_{i=0}^{\lfloor c_k \rfloor}(c_k - i) \geq \phi^3 \cdot \sum_{i=0}^{k-1}(b_{\phi^3,i} - i)$, it is sufficient to prove:

$$\sum_{i=0}^{\lfloor c_k \rfloor}(c_k - i) \geq \phi^3 \cdot \sum_{i=0}^{k-1}(c_i - i) \geq \phi^3 \cdot \sum_{i=0}^{k-1}(b_{\phi^3,i} - i) \ . \tag{3}$$

The second inequality in (3) follows from the induction hypothesis. The first one can be checked manually for $1 \leq k \leq 5$, so it is enough to prove it for $k > 5$. Let $d_k = c_k - \lfloor c_k \rfloor \in (0, 1)$. Then, the left hand side of the first inequality in (3) can be lower bounded by:

$$\begin{aligned}
\sum_{i=0}^{\lfloor c_k \rfloor}(c_k - i) &= \frac{(\lfloor c_k \rfloor + 1)(2c_k - \lfloor c_k \rfloor)}{2} = \frac{(c_k - d_k + 1)(c_k + d_k)}{2} = \frac{c_k(c_k + 1) + d_k - d_k^2}{2} \\
&\geq \frac{c_k(c_k + 1)}{2} = \frac{(k\phi + a)(k\phi + a + 1)}{2} \geq \phi^3 \cdot \sum_{i=0}^{k-1}(c_i - i) \ ,
\end{aligned}$$

where the last inequality follows from Observation 6.6. $\square$

Recall that the next step we need to prove is that if $b_{\beta,k}$ is upper bounded by a linear function of some slope, then it is also upper bounded by a linear function of significantly smaller slope.

Lemma 6.8. *Fix some $\beta \leq \phi^3$. If, for every $k \geq 0$, $b_{\beta,k} \leq mk + n$ for some $m \in [1, \phi]$ and $n \geq 0$, then, for every $k \geq 0$, $b_{\beta,k} \leq \sqrt{\beta/\phi^3} \cdot mk + n'$ for some $n' \geq 0$.*

Proof. For $k > 5$, we have:

$$\begin{aligned}
\phi^3 \cdot \sum_{i=0}^{k-1}(b_{\beta,i} - i) &\leq \phi^3 \cdot \sum_{i=0}^{k-1}[(m-1)i + n] \\
&\leq \frac{m-1}{\phi - 1} \cdot \phi^3 \cdot \left(1 + b - 2a + \sum_{i=0}^{k-1}[(\phi - 1) \cdot i + a]\right) + \phi^3 nk \\
&= \phi(m-1) \cdot \phi^3 \cdot \sum_{i=0}^{k-1}(c_i - i) + \phi^3 nk \leq \phi(m-1) \cdot \frac{(k\phi + a)(k\phi + a + 1)}{2} + \phi^3 nk
\end{aligned}$$

$$\le \frac{(k\phi \cdot \sqrt{\phi(m-1)} + a)(k\phi \cdot \sqrt{\phi(m-1)} + a + 1)}{2} + \phi^3 nk$$
$$\le \frac{(mk+a)(mk+a+1)}{2} + \phi^3 nk \le \frac{(mk+\phi^3 n + a)(mk + \phi^3 n + a + 1)}{2} \ ,$$

where the second inequality holds since $a, 1 + b - 2a \ge 0$, the third from Observation 6.6, the forth since $(m-1)\phi \le 1$ and the sixth since $\phi \cdot \sqrt{\phi(m-1)} \le m$. Multiplying the last inequality by $\beta/\phi^3 \le 1$ gives:

$$\beta \cdot \sum_{i=0}^{k-1} (b_{\beta,i} - i) \le \frac{(\sqrt{\beta/\phi^3} \cdot mk + \phi^3 n + a)(\sqrt{\beta/\phi^3} \cdot mk + \phi^3 n + a + 1)}{2}$$
$$\le \sum_{i=0}^{\lfloor(\sqrt{\beta/\phi^3} \cdot mk + \phi^3 n + a + 1\rfloor} \left[(\sqrt{\beta/\phi^3} \cdot mk + \phi^3 n + a + 1) - i\right] \ .$$

By the definition of $b_{\beta,k}$, the last inequality implies:

$$b_{\beta,k} \le \max\{1, \sqrt{\beta/\phi^3} \cdot mk + \phi^3 n + a + 1\} \ .$$

This completes the proof of the lemma for $k > 5$ with $n' = \max\{1, \phi^3 n + a + 1\}$. We can always make the lemma hold for $k \le 5$ by choosing a large enough n'. $\square$

Corollary 6.9. *For any $\beta < \phi^3$ there are constants $m \in (0, 1)$ and $n \ge 0$ such that $b_{\beta,k} \le mk + n$ for every $k \ge 0$.*

Proof. By Lemma 6.7, $b_{\beta,k} \le c_k \le \phi \cdot k + 1$. Assume, for the sake of contradiction, that the corollary is not correct. Hence, there must be $m \in [1, \phi]$ and $n \ge 0$ such that $b_{\beta,k} \le mk + n$ for every $k \ge 0$, but the same is not true for any $m' \le m + \sqrt{\beta/\phi^3} - 1$.

On the other hand, by Lemma 6.8, there exists $n' \ge 0$ such that for every $k \ge 0$, $b_{\beta,k} \le \sqrt{\beta/\phi^3} \cdot mk + n'$, which is a contradiction since $\sqrt{\beta/\phi^3} \cdot m = m + m(\sqrt{\beta/\phi^3} - 1) \le m + \sqrt{\beta/\phi^3} - 1$. $\square$

Theorem 6.4, now, follows immediately from Corollary 6.9.

7 Lower Bound for Randomized Algorithms in the Integral Valued Model

For a model in which all packets have equal values, [10] gave a lower bound of ϕ on the competitive ratio of randomized algorithms (where ϕ is the golden ratio). This is the strongest lower bound known for randomized algorithms even in models allowing multiple packet values. In this section, we show that no randomized algorithm is $(4 - \varepsilon)$-competitive for any constant $\varepsilon > 0$ against an oblivious adversary in the integral valued model. Notice that this result extends to the real valued model as well, since the real valued model generalizes the integral valued model. Throughout the section, we consider some $\beta \in [1, 4)$ and assume, for the sake of contradiction, the existence of a β-competitive randomized algorithm ALG. Consider the adversary given by Algorithm 5.

The idea of Algorithm 5 is as following. If at some iteration $R_k < k(k+1)/(2\beta)$, then ALG is not β-competitive because OPT can get a revenue of $k(k+1)/2$ by simply accepting only the k packets of value k. Thus, ALG must accept at each iteration enough packets to guarantee $R_k \ge k(k+1)/(2\beta)$. However, as more and more iterations go by, ALG has to accept more and

Algorithm 5: Adversary for Randomized Algorithms in the Integer Valued Model

```
// All packets generated by this adversary are given to ALG during the time
   range (0,1) in the order in which they are generated.
for k = 1 to ∞ do
    Generate k packets of value k.
    Let R_k be the expected revenue of the algorithm at this point.
    if R_k < k(k+1)/(2β) then Terminate.
```

more packets to keep up with this goal. This makes ALG's queue increase, which diminishes the revenue received from new packets, and eventually fail ALG.

The following lemma gives a lower bound on R_k.

Lemma 7.1. *Since ALG is β-competitive by our assumption, the adversary never terminates. Hence, $R_k \geq k(k+1)/(2\beta)$ for every $k \geq 1$.*

Proof. If the adversary terminates after the k^{th} iteration, then the expected revenue of ALG is $R_k < k(k+1)/(2\beta)$. On the other hand, OPT can accept only the k packets generated by the adversary on the k^{th} iteration, which results in a revenue of $k(k+1)/2$. Thus, the competitive ratio of ALG is worse than:

$$\frac{k(k+1)/2}{k(k+1)/(2\beta)} = \beta \ .$$

□

The last lemma gives a lower bound on R_k; the next lemma establishes an upper bound. Let $d_{i,j}$ denote the j^{th} packet the adversary sends in the i^{th} iteration, $p_{i,j}$ denote the probability that ALG accepts $d_{i,j}$ and $p_i = \sum_{j=1}^{i} p_{i,j}$. Notice that p_i is also the expected number of packets of value i accepted by ALG.

Lemma 7.2. *For every $k \geq 1$, $R_k \leq \sum_{i=1}^{k} p_i(i+1/2) - \frac{\left(\sum_{i=1}^{k} p_i\right)^2}{2}$.*

Proof. Let Q_k and W_k be random variables denoting the number of packets in ALG's queue after the k^{th} iteration of the adversary (*i.e.*, immediately after the k packets of value k are generated) and the total value of the packets ALG accepted up to this this point, respectively. The revenue of ALG after the k^{th} iteration can be written as:

$$W_k - \sum_{i=0}^{Q_k - 1} i = W_k - \frac{Q_k(Q_k-1)}{2} \ .$$

Notice that R_k is the expectation of the above expression. Thus,

$$\begin{aligned} R_k = \mathbb{E}\left[W_k - \frac{Q_k(Q_k-1)}{2}\right] &= \mathbb{E}[W_k] - \frac{\mathbb{E}[Q_k^2] - \mathbb{E}[Q_k]}{2} \\ = \mathbb{E}[W_k] - \frac{\mathtt{Var}[Q_k] + (\mathbb{E}[Q_k])^2 - \mathbb{E}[Q_k]}{2} &\leq \mathbb{E}[W_k] - \frac{(\mathbb{E}[Q_k])^2 - \mathbb{E}[Q_k]}{2} \ , \end{aligned} \tag{4}$$

where $\mathtt{Var}[Q_k]$ is the variance of Q_k. The expected values of Q_k and W_k can be calculated as following:

$$E[W_k] = \sum_{i=1}^{k}\sum_{j=1}^{i}[p_{i,j} \cdot w(d_{i,j})] = \sum_{i=1}^{k}\left(i \cdot \sum_{j=1}^{i} p_{i,j}\right) = \sum_{i=1}^{k} ip_i \ ,$$

and

$$E[Q_k] = \sum_{i=1}^{k} \sum_{j=1}^{i} p_{i,j} = \sum_{i=1}^{k} p_i \ \ .$$

Plugging these expectations into Inequality (4) completes the proof of the lemma. □

The lower and upper bounds given by Lemmata 7.1 and 7.2, together, imply the following inequality:

$$\frac{k(k+1)}{2\beta} \leq \sum_{i=1}^{k}(i+1/2) \cdot p_i - \frac{\left(\sum_{i=1}^{k} p_i\right)^2}{2} \Leftrightarrow \frac{k(k+1)}{\beta} \leq \sum_{i=1}^{k}(2i+1) \cdot p_i - \left(\sum_{i=1}^{k} p_i\right)^2 \ \ .$$

Let $\bar{Q}_k = \sum_{i=1}^{k} p_i$ denote the expected number of packets in ALG's queue after the k^{th} iteration of the adversary. For consistency, we also define $\bar{Q}_0 = 0$. The above inequality can be written in terms of $\bar{Q}_k$ as following:

$$\frac{k(k+1)}{\beta} \leq \sum_{i=1}^{k}(2i+1) \cdot p_i - \left(\sum_{i=1}^{k} p_i\right)^2 = \bar{Q}_k(2k+1) - \bar{Q}_k^2 - 2 \cdot \sum_{i=1}^{k} \bar{Q}_{i-1} \ \ . \tag{5}$$

The rest of this section is devoted for proving that the above formula leads to a contradiction.

Observation 7.3. *For every $m \in [0, 1-\beta^{-1}]$, $1 - \sqrt{1-m-\beta^{-1}} \geq m + \beta^{-1} - 1/4$.*

Proof. Notice that:

$$\begin{aligned} & 1 - \sqrt{1-m-\beta^{-1}} \geq m + \beta^{-1} - 1/4 \Leftrightarrow 5/4 - m - \beta^{-1} \geq \sqrt{1-m-\beta^{-1}} \\ \Leftrightarrow\ & 25/16 + m^2 + \beta^{-2} - 5m/2 - 5\beta^{-1}/2 + 2m\beta^{-1} \geq 1 - m - \beta^{-1} \\ \Leftrightarrow\ & m^2 + m(2\beta^{-1} - 3/2) + 9/16 + \beta^{-2} - 3\beta^{-1}/2 \geq 0 \ \ . \end{aligned}$$

For the last inequality to hold for every m, we need:

$$\begin{aligned} & (2\beta^{-1} - 3/2)^2 - 4(9/16 + \beta^{-2} - 3\beta^{-1}/2) \leq 0 \\ \Leftrightarrow\ & (4\beta^{-2} + 9/4 - 6\beta^2) - (9/4 + 4\beta^{-2} - 6\beta^{-1}) \leq 0 \ \ . \end{aligned}$$

□

Lemma 7.4. *If there exist constants $m \geq 0$ and $n \leq 0$ such that $\bar{Q}_k \geq mk + n$ for every $k \geq 0$, then: $m \leq 1 - \beta^{-1}$ and there exists $n' \leq 0$ such that $\bar{Q}_k \geq (m + \beta^{-1}/2 - 1/8)k + n'$ for every $k \geq 0$.*

Proof. By Equation (5):

$$\begin{aligned} \bar{Q}_k - \frac{\bar{Q}_k^2}{2k+1} &\geq \frac{2 \cdot \sum_{i=1}^{k} \bar{Q}_{i-1} + \frac{k(k+1)}{\beta}}{2k+1} \geq \frac{2 \cdot \sum_{i=1}^{k}[m(i-1)+n] + \frac{k(k+1)}{\beta}}{2k+1} \\ &= \frac{mk(k-1) + 2nk + \frac{k(k+1)}{\beta}}{2k+1} \geq \frac{(m+\beta^{-1})}{2} \cdot k + n - m \ \ . \end{aligned}$$

If $m > 1 - \beta^{-1}$, then for large enough k, $\frac{(m+\beta^{-1})}{2} \cdot k + n - m \geq (k+1)/2$. Hence, $\bar{Q}_k$ must obey:

$$\frac{\bar{Q}_k^2}{2k+1} - \bar{Q}_k + \frac{k+1}{2} \leq 0 \ \ ,$$

which is a contradiction since this inequality has no solution because:

$$1 - 4 \cdot \frac{1}{2k+1} \cdot \frac{k+1}{2} = 1 - \frac{2k+2}{2k+1} < 0 \ .$$

Thus, it is safe to assume $m \leq 1 - \beta^{-1}$. Let $A = (m + \beta^{-1})/2 \leq 1/2$ and $B = n - m \leq 0$. Then,

$$\begin{aligned}
\bar{Q}_k &\geq \frac{1 - \sqrt{1 - \frac{4Ak+4B}{2k+1}}}{\frac{2}{2k+1}} \geq k \cdot \left(1 - \sqrt{\frac{(2-4A)k + (1-4B)}{2k+1}}\right) \\
&\geq k \cdot \left(1 - \sqrt{1 - 2A + \frac{1-4B}{2k}}\right) \geq k \cdot \left(1 - \sqrt{1-2A}\right) - \sqrt{k} \cdot \sqrt{\frac{1-4B}{2}} \\
&= k \cdot \left(1 - \sqrt{1 - m - \beta^{-1}}\right) - \sqrt{k} \cdot \sqrt{\frac{1-4B}{2}} \geq k(m + \beta^{-1} - 1/4) - \sqrt{k} \cdot \sqrt{\frac{1-4B}{2}} \ ,
\end{aligned}$$

where the forth inequality holds since $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for $a, b \geq 0$ and the last one follows from Observation 7.3. Since $\beta < 4$, for large enough k: $k(\beta^{-1}/2 - 1/8) \geq \sqrt{k} \cdot \sqrt{\frac{1-4B}{2}}$. For such k's the lemma holds with $n' = 0$. By choosing small enough n', the lemma can be made to hold also for the finitely many smaller values of k. ☐

Theorem 7.5. *No algorithm is β-competitive for any $\beta \in [1, 4)$.*

Proof. Recall that we assumed ALG is β competitive, and let us show this assumption leads to a contradiction. Let $m \geq 0$ and $n \leq 0$ be such values obeying:

- $\bar{Q}_k \geq mk + n$ for every $k \geq 0$.
- For every $n' \leq 0$, there exists $k \geq 0$ for which $\bar{Q}_k < (m + \beta^{-1}/2 - 1/8)k + n'$.

Such a pair of values must exist since $\bar{Q}_k \geq 0 \cdot k + 0$ for every $k \geq 0$, but by Lemma 7.4 for every $m' > 1 - \beta^{-1}$ and $n' \leq 0$ there exists $k \geq 0$ for which $\bar{Q}_k < (m + \beta^{-1}/2 - 1/8)k + n'$. On the other hand, the existence of such a pair of values is an immediate contradiction to the second part of Lemma 7.4. ☐

8 Conclusion

We considered three variants of HDSP corresponding to three sets of allowed packet values. For the unrestricted model we showed that there is no hope of a constant competitive ratio algorithm. For the integral valued model we gave a 4-competitive deterministic algorithm, and showed that no better constant competitive ratio is possible, even for randomized algorithms. For the real valued model we gave a 4-competitive randomized algorithm, and a matching lower bound (up to low order terms). However, there is still a small gap in this model in terms of deterministic algorithms. We gave a lower bound of $\phi^3 - \varepsilon \approx 4.236$ while the best upper bound known for this model is 4.24. Closing the gap, probably by finding a ϕ^3-competitive deterministic algorithm, is an obvious open problem.

One can consider a general model in which all values larger than c are allowed, for some $c \geq 0$. Notice that this general model includes the real valued and the unrestricted models as special cases (with $c = 1$ and $c = 0$, respectively). As the real valued model has a 4-competitive randomized algorithm, and the unrestricted model has no constant competitive ratio, it is clear that the competitive ratio depends on c. It may be interesting to explore the exact connection between c and the competitive ratio.

In a sense, the online setting captures the worst case scenario in which there is no future knowledge. For many real life situations, this assumption is too pessimistic. One model that makes a weaker assumption is the stochastic model where the adversary is only allowed to choose a distribution for the input requests. The requests are then drawn from this distribution using independent sampling. Garg et al. [11] showed that in this model there is an $O(1)$-competitive algorithm for Steiner Tree, Steiner Forest, facility location and vertex cover. The positive results we present in this paper have matching lower bounds showing that they are tight (up to low order terms). To improve over these results one must consider a somewhat different setting such as a stochastic model.

References

[1] N. Andelman, Y. Mansour, and A. Zhu. Competitive queueing policies for QoS switches. In *SODA*, pages 761–770, 2003.

[2] Y. Azar and Y. Richter. Management of multi-queue switches in QoS networks. *Algorithmica*, 43(1-2):81–96, 2005.

[3] Y. Bartal, F. Y. L. Chin, M. Chrobak, S. P. Y. Fung, W. Jawor, R. Lavi, J. Sgall, and T. Tichy. Online competitive algorithms for maximizing weighted throughput of unit jobs. In *STACS*, pages 187–198, 2004.

[4] F. Y. L. Chin and S. P. Y. Fung. Online scheduling with partial job values: Does timesharing or randomization help? *Algorithmica*, 37(3):149–164, 2003.

[5] M. Chrobak, W. Jawor, J. Sgall, and T. Tichý. Improved online algorithms for buffer management in QoS switches. *ACM Trans. Algorithms*, 3(4):50, 2007.

[6] M. Englert and M. Westermann. Lower and upper bounds on fifo buffer management in QoS switches. *Algorithmica*, 53(4):523–548, 2009.

[7] M. Englert and M. Westermann. Considering suppressed packets improves buffer management in quality of service switches. *SIAM J. Comput.*, 41(5):1166–1192, 2012.

[8] M. Feldman and J. S. Naor. Non-preemptive buffer management for latency sensitive packets. In *INFOCOM*, pages 186–190, 2010.

[9] A. Fiat, Y. Mansour, and U. Nadav. Competitive queue management for latency sensitive packets. PEGG, 2007.

[10] A. Fiat, Y. Mansour, and U. Nadav. Competitive queue management for latency sensitive packets. In *SODA*, pages 228–237, 2008.

[11] N. Garg, A. Gupta, S. Leonardi, and P. Sankowski. Stochastic analyses for online combinatorial optimization problems. In *SODA*, pages 942–951, 2008.

[12] L. Jeż. A universal randomized packet scheduling algorithm. *Algorithmica*, 67(4):498–515, 2013.

[13] L. Jeż, F. Li, J. Sethuraman, and C. Stein. Online scheduling of packets with agreeable deadlines. *ACM Trans. Algorithms*, 9(1):5:1–5:11, Dec 2012.

[14] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Netw.*, 2(1):1–15, 1994.

[15] V. Paxson and S. Floyd. Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3:226–244, 1995.

[16] A. Veres and M. Boda. The chaotic nature of tcp congestion control. In *INFOCOMM*, pages 1715–1723, 2000.