Considering Suppressed Packets Improves Buffer Management in QoS Switches*

Matthias Englert† Matthias Westermann†

Abstract

The following buffer management problem arises in network switches providing differentiated services: At the beginning of each time step, one packet can be sent, and afterwards an arbitrary number of new packets arrive. Packets that are not sent can be stored in a buffer. Each packet is attributed by a deadline, and a packet is automatically deleted from the buffer if it is still stored in the buffer by the end of its deadline. The differentiated service model is abstracted by attributing each packet with a value according to its service level. A buffer management strategy determines the packet to be sent in each time step. The goal of a buffer management strategy is to maximize the sum of the values of sent packets.

We introduce the concept of suppressed packets and present a deterministic strategy that is based on this concept. We show that this strategy achieves a competitive ratio of $2\sqrt{2} - 1 \approx 1.828$, which is the best known competitive ratio in the deterministic case. Further, we present a memoryless version of this strategy that achieves a competitive ratio of ≈ 1.893. This is the first memoryless strategy that achieves a competitive ratio less than 2, and the competitive ratio of this strategy is even better than the ratios of all previously known deterministic strategies. This demonstrates the potential of the concept of suppressed packets. In addition, we present a simple strategy that achieves the optimal competitive ratio of $\min\{(1+\alpha)/\alpha, 2\alpha/(\alpha+1)\} \leq \sqrt{2}$, if only two packet values 1 and $\alpha > 1$ are possible.

1 Introduction

Quality of Service (QoS) guarantees for network services allow service providers to address the service requirements of customers by providing different levels of service. In the network setting, where traffic volumes may exceed network capacity, effective management of packets at buffers in switches is a key to achieving QoS guarantees. By differentiating service levels, packets of different types may be treated according to the level of service they require.

1.1 The Model

Time is slotted in *time steps*. At the beginning of each time step, one packet can be sent, and afterwards an arbitrary number of new packets arrive. Hence, in the first time step, i.e., time step 0, a packet cannot be sent and only new packets arrive. The sequence of time steps can also be regarded as a sequence of *send and arrival events* $\sigma_1\sigma_2\cdots$, where each sending of a packet corresponds to a send event and each arrival of a new packet corresponds to an arrival event. Obviously, the event sequence is partitioned into time steps, where the first time step starts with the first event and a new time step starts right before each send event.

Packets that are not sent can be stored in a *buffer*. Each packet p is attributed by a *deadline* $d(p)$, which is greater than the time step in which p arrives. If a packet p is still stored in the buffer by the end of time step $d(p)$, p is automatically deleted from the buffer at the end of this time step. Note that an explicit bound on the size of the buffer does not exist, instead the possible delay of each packet is bounded. Hence, this model is known as the *bounded-delay model*.

The differentiated service model is abstracted by attributing each packet p with a *value* $v(p)$ according to its service level. A *buffer management strategy* determines the packet to be sent in each time step. The goal of a buffer management strategy is to maximize the sum of the values of sent packets.

For a given sequence of events $\sigma_1\sigma_2\cdots$ and a buffer management strategy A, let S_t^{A} denote the set of packets sent by A by the end of event σ_t, and let B_t^{A} denote the set of packets stored in the buffer of A at the end of event σ_t. Initially, define $S_0^{\mathrm{A}} := \emptyset$ and $B_0^{\mathrm{A}} := \emptyset$.

The notion of an online strategy is intended to formalize the realistic scenario where the strategy does not have knowledge about the whole input sequence of arriving packets in advance. The *online strategy* gets to know this sequence packet by packet, and has to act without knowledge about the future. Online strategies

*Supported by DFG grant WE 2842/1.

†Department of Computer Science, RWTH Aachen, Germany. {englert, marsu}@cs.rwth-aachen.de.

are typically evaluated in a competitive analysis. In this kind of analysis the total value produced by the online strategy is compared with the total value produced by an optimal offline strategy.

For a given input sequence I of arriving packets, let $\mathrm{OPT}(I)$ denote the total value produced by an optimal offline strategy. An online strategy is called *c-competitive* if it produces total value at least $\mathrm{OPT}(I)/c - \kappa$, for each input sequence I of arriving packets, where κ is a term that does not depend on I. The value c is also called the *competitive ratio* of the online strategy.

1.2 Related Work

Kesselman et al. [8] show that the greedy strategy which always sends the available packet with maximum value achieves a competitive ratio of 2 and, if only the two packet values 1 and $\alpha > 1$ are possible, a better competitive ratio of $1 + 1/\alpha$. Chrobak et al. [5] present a strategy that achieves a competitive ratio of 64/33 which is the only previously known deterministic upper bound on the competitive ratio of this problem less than 2. Note that this strategy is not memoryless. Andelman, Mansour, and Zhu [1], Chin and Fung [4], and Hajek [7] show a lower bound of $(\sqrt{5}+1)/2 \approx 1.618$ on the competitive ratio of any deterministic strategy. Chin et al. [3] present a randomized strategy that achieves a competitive ratio of $e/(e-1) \approx 1.582$. Chin and Fung [4] present a lower bound of 5/4 on the competitive ratio of any randomized strategy.

Concurrently and independently of our work, Li, Sethuraman, and Stein [11] developed the DP (for dummy packets) strategy that achieves a competitive ratio of $6/(\sqrt{5}+1) \approx 1.854$. Similar to our approach they use an optimal provisional schedule and identify two packets similar to our first- and max-packet. However, instead of considering suppressed packets they manipulate the buffer contents to store informations about the past. In some situations, the value of a packet is artificially reduced by a certain factor, and a dummy packet is added to the buffer and linked to a real packet stored in the buffer. Dummy packets are not sent but they influence the behavior of the strategy. Their proof is, in contrast to our proof, not explicitly based on a potential function. Instead, the buffer of the optimal offline strategy is modified after each step.

Several restricted variants of this problem have been considered. Define the span of a packet to be the difference between its deadline and the time step in which it arrives. An instance is s-bounded, if the span of each packet is at most s, and an instance is s-uniform, if the span of each packet is exactly s. Further, an instance has agreeable deadlines, if for each packets p and each packet p' that arrives after p, $d(p) \leq d(p')$. Note that s-uniform instances are a special case of instances with agreeable deadlines.

The lower bound of $(\sqrt{5}+1)/2$ on the competitive ratio of any deterministic strategy in [1, 4] and the lower bound of 5/4 on the competitive ratio of any randomized strategy in [4] use only instances that are 2-bounded and therefore also have agreeable deadlines. For s-bounded instances, Chin et al. [3] present a strategy that achieves a competitive ratio of $2 - 2/s + o(1/s)$. This strategy achieves an optimal competitive ratio of $(\sqrt{5}+1)/2$, for $s = 2, 3$, and a competitive ratio of $\sqrt{3}$, for $s = 4$. Further, for 2-bounded instances, they give a randomized strategy that achieves an optimal competitive ratio of 5/4. For 2-uniform instances, Chrobak et al. [5] present a strategy that achieves a competitive ratio of ≈ 1.377 and a matching lower bound. For instances with agreeable deadlines, Li, Sethuraman, and Stein [10] give a strategy that achieves an optimal competitive ratio of $(\sqrt{5}+1)/2$.

The following results refer to the similar FIFO model, in which an explicit bound on the delay of each packet does not exists, instead the size of the buffer is bounded and reordering of packets is not allowed, i.e., the sequence of the sent packets is a subsequence of the arriving packets. Kesselman et al. [8] show that the greedy strategy achieves a competitive ratio of 2. Kesselman, Mansour, and van Stee [9] introduce the preemptive greedy strategy and prove that this strategy achieves a competitive ratio of ≈ 1.983. Englert and Westermann [6] show that the preemptive greedy strategy achieves a competitive ratio of $\sqrt{3} \approx 1.732$, which is the best known upper bound on the competitive ratio of this problem. Kesselman, Mansour, and van Stee [9] present a lower bound of ≈ 1.419 on the competitive ratio of this problem.

1.3 Our Contributions.

In Section 2, we introduce the basic concept of provisional schedules. After each event σ_t, our strategies compute the so-called optimal provisional schedule for the set of pending packets B_t^{ONL} stored in the buffer at the end of σ_t. Note that this optimal provisional schedule is computed under the assumption that new packets do not arrive in the future.

In Section 3, we consider a first approach which is simple and natural. For each send event, define the first-packet as the packet that is basically earliest in the respective optimal provisional schedule and the max-packet as the packet that has basically maximum value in the respective optimal provisional schedule. We study the natural approach to send either the first-packet or the max-packet, depending on the value of these two packets. We state that this approach is very promising if only two packet values 1 and $\alpha > 1$ are possible, i.e., for this case, we present a strategy based on this approach that achieves an optimal competitive ratio of $\min\{1+\alpha)/\alpha, 2\alpha/(\alpha+1)\} \leq \sqrt{2}$. However, we also show that this approach is disappointing for general packet values, i.e., we prove that this approach cannot achieve a competitive ratio better than 2. Note that there are two natural greedy strategies: Either always send the first-packet or always send the max-packet. These two greedy strategies already achieve a competitive ratio of 2 [2, 8].

In Section 4, we enhance the first approach by introducing the concept of suppressed packets. Consider the optimal provisional schedule S for a set of pending packets P. Suppose that a packet $q \in P$ does not appear in S, but it can be added to S if another packet $p \in P$ that appears in S is removed from P and as a consequence also from S. Then, q is called suppressed by p. Obviously, if p is sent and p is not the first-packet, q can appear in the optimal provisional schedule. Hence, the sending of packets that are not first-packets can lead to the appearance of suppressed packets in the optimal provisional schedule. We present a deterministic strategy that is based on the concept of suppressed packets and show that this strategy achieves a competitive ratio of $2\sqrt{2}-1 \approx 1.828$. Note that this the best known competitive ratio in the deterministic case. In addition, we present a memoryless version of this strategy that achieves a competitive ratio of ≈ 1.893. This is the first memoryless strategy that achieves a competitive ratio less than 2, and the competitive ratio of this strategy is even better than the ratios of all previously known deterministic strategies. This demonstrates the potential of the concept of suppressed packets. The proofs of this results are given in Section 5.

2 Provisional Schedules

We introduce the basic concept of provisional schedules. For two packets p and q, $p \prec q$ according to the *canonical order* $\prec$, if either $d(p) < d(q)$, or $d(p) = d(q)$ and $v(p) > v(q)$, or $d(p) = d(q)$, $v(p) = v(q)$, and the arrival event of p is before the arrival event of q (the last condition only ensures that ties are broken in some arbitrary but consistent way).

A *provisional schedule* S for a set of pending packets P specifies which packet should be sent in which time step. To simplify notation, a provisional schedule S is sometimes regarded as a set of packets, e.g., we write $p \in S$ to indicate that the packet p is scheduled in S. Let $S(p)$ denote the time step at which a packet $p \in S$ is scheduled in a provisional schedule S. Obviously, only one packet can be scheduled at each single time step and, for each $p \in S$, $S(p) \leq d(p)$. A provisional schedule S is called a schedule for a time step τ, if all packets in S are scheduled after time step τ, i.e., for each $p \in S$, $S(p) \geq \tau + 1$.

After each event σ_t, our strategies compute the *optimal provisional schedule* S_t for the set of pending packets B_t^{ONL} stored in the buffer at the end of σ_t as follows: Start with an empty set S. Consider the packets in B_t^{ONL} for inclusion into S in descending order of their value (ties are broken in favor of smaller packets according to the canonical order). A packet p is added to the set S if

$$|\{p' \in S \cup \{p\} \mid d(p') \leq \tau'\}| \leq \tau' - \tau \ ,$$

for each $\tau' \geq \tau$ with τ denoting the time step the event σ_t belongs to.

The final set S can be interpreted as the optimal provisional schedule S_t: Let $p_i \in S$ denote the i-th smallest packet in S according to the canonical order. Then, p_i can be scheduled for the time step $\tau + i$ since $d(p_i) \geq \tau + i$ due to

$$|\{p' \in S \mid d(p') \leq \tau'\}| \leq \tau' - \tau \ ,$$

for each $\tau' \geq \tau$.

The optimal provisional schedule S_t is computed under the assumption that new packets do not arrive in the future. Further, note that S_t is a schedule for the time step τ if the event σ_t belongs to the time step τ, i.e., packets cannot be scheduled for the past. Finally, observe that the schedule S_t is in canonical order, i.e., for each pair of packets $p, q \in S_t$, $S_t(p) < S_t(q)$ if and only if $p \prec q$.

3 First Approach

In this section, we consider a first approach which is simple and natural. For each send event σ_t, define the *first-packet* $p_f \in S_{t-1}$ as the first packet in S_{t-1} according to the canonical order and the *max-packet* $p_m := \mathrm{argmax}_{p' \in S_{t-1}} v(p')$ (ties are broken in favor of the smallest packet according to the canonical order). We study the natural approach to send either the first-packet or the max-packet, depending on the value of these two packets. We state that this approach is very promising if only two different packet values are possible. However, we also show that this approach is disappointing for general packet values.

Consider only two different packet values 1 and $\alpha > 1$. Depending on α, the following simple strategy always sends either the first-packet or the max-packet.

- If $\alpha < \sqrt{2} + 1$, send the first-packet.
- Otherwise, send the max-packet.

The following theorem states an upper bound on the competitive ratio of this strategy. If the max-packet is always sent, the proof of the competitive ratio $1 + 1/\alpha$ can be found in [8], and if the first-packet is always sent, the competitive ratio $2\alpha/(\alpha + 1)$ follows directly from the definition of the strategy.

THEOREM 3.1. *If only two packet values* 1 *and* $\alpha > 1$ *are possible, the above strategy achieves a competitive ratio of* $\min\{1 + 1/\alpha, 2\alpha/(\alpha + 1)\} \leq \sqrt{2}$.

The following theorem states a matching lower bound on the competitive ratio of any deterministic strategy. The input sequence for this lower bound can be found in the proof of a lower bound on the competitive ratio for 2-bounded instances [8].

THEOREM 3.2. *If only two packet values* 1 *and* $\alpha > 1$ *are possible, the competitive ratio of any deterministic strategy is at least* $\min\{1 + 1/\alpha, 2\alpha/(\alpha + 1)\}$.

Consider general packet values. There are two natural greedy strategies: Either always send the first-packet or always send the max-packet. These two greedy strategies achieve a competitive ratio of 2 [2, 8]. The following natural strategy uses a parameter $\beta > 1$ and either sends the first-packet p_f or the max-packet p_m, depending on the value of these two packets.

- If $v(p_f) \geq v(p_m)/\beta$, send the first-packet p_f.
- Otherwise, send the max-packet p_m.

The following theorem shows that this approach does not achieve a competitive ratio better than the competitive ratio of the greedy strategies.

THEOREM 3.3. *The competitive ratio of the above strategy is at least 2.*

Proof. Depending on β, we distinguish the following two cases.

- Suppose that $\beta > 2$.

  The input sequence consists of $n + 1$ consecutive phases defined as follows.

  – Phase $1 \leq i \leq n$ consists of 2^{n-i} time steps. In the first time step of each phase, 2^{n-i} packets with value 2^{i-1} and deadline $2^n - 2^{n-i+1} + 1, \ldots, 2^n - 2^{n-i}$, respectively, and 2^{n-i} packets with value 2^i and deadline $2^n - 2^{n-i} + 1, \ldots, 2^n$, respectively, arrive. In the remaining $2^{n-i} - 1$ time steps, new packets do not arrive.

  – Phase $n + 1$ consists of one time step. In this time step, one packet with value 2^n and deadline 2^n arrives.

  For this input sequence, the above strategy produces value $\sum_{i=1}^{n}(2^{n-i} \cdot 2^{i-1}) + 2^n$, and the optimal value is $\sum_{i=1}^{n}(2^{n-i} \cdot 2^i) + 2^n$. Hence, the competitive ratio is

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n} 2^n + 2^n}{\sum_{i=1}^{n} 2^{n-1} + 2^n} = 2 \ .$$

- Suppose that $1 < \beta \leq 2$.

  The input sequence is an extension of the previous one. It consists of $n + 1$ consecutive phases defined as follows.

  – Phase $1 \leq i \leq n$ consists of 2^{n-i} time steps. In the first time step of each phase, 2^{n-i} packets with value 2^{i-1} and deadline $2^n - 2^{n-i+1} + 1, \ldots, 2^n - 2^{n-i}$, respectively, 2^{n-i} packets with value 2^i and deadline $2^n - 2^{n-i} + 1, \ldots, 2^n$, respectively, and 2^{n-i} packets with value $(2+\varepsilon) \cdot 2^{i-1} > 2^i$ and deadline 2^{n+1} arrive. In the remaining $2^{n-i} - 1$ time steps, new packets do not arrive.

  – Phase $n + 1$ consists of one time step. In this time step, one packet with value 2^n and deadline 2^n arrives.

  For this input sequence, the above strategy produces value $\sum_{i=1}^{n}(2^{n-i} \cdot (2 + \varepsilon) \cdot 2^{i-1}) + 2^n$, and the optimal value is $\sum_{i=1}^{n}(2^{n-i} \cdot (4 + \varepsilon) \cdot 2^{i-1}) + 2^n$. Hence, the competitive ratio is

$$\begin{aligned} &\lim_{n\to\infty} \lim_{\varepsilon\to 0} \frac{\sum_{i=1}^{n}(2^{n-i} \cdot (4 + \varepsilon) \cdot 2^{i-1}) + 2^n}{\sum_{i=1}^{n}(2^{n-i} \cdot (2 + \varepsilon) \cdot 2^{i-1}) + 2^n} \\ = &\lim_{n\to\infty} \frac{\sum_{i=1}^{n} 2^{n+1} + 2^n}{\sum_{i=1}^{n} 2^n + 2^n} = 2 \ . \end{aligned}$$

This concludes the proof of the theorem. □

4 Our Strategies

We enhance the natural approach to send either the first-packet or the max-packet by introducing the concept of suppressed packets. Consider the optimal provisional schedule S for a set of pending packets P. Suppose that a packet $q \in P$ does not appear in S, but it can be added to S if another packet $p \in P$ that appears in S is removed from P and as a consequence also from S. Then, q is called *suppressed* by p.

More precisely, consider the optimal provisional schedule S_t at the end of event σ_t for the set of pending

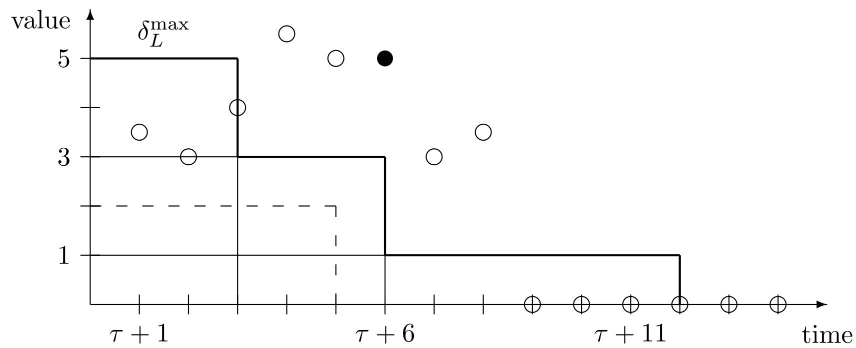

Figure 1: The optimal provisional schedule for a time step τ with 8 packets and additional dummy packets of value 0. The marked packet is scheduled on its deadline $\tau+6$ and this is a tight time step. In addition, the set of levels $L = \{(\tau+3,5),(\tau+5,2),(\tau+6,3),(\tau+12,1)\}$ and $\delta_L^{\max}$ are depicted. In the second step of our strategy, the level $(\tau+6,3)$ would be added to the set of levels for the tight time step $\tau+6$. Since the level $(\tau+5,2)$ is dominated by the level $(\tau+6,3)$, it does not need to be retained.

packet B_t^{ONL}. For each $p \in S_t$, let S_t^p denote the optimal provisional schedule at the end of event σ_t for the set of pending packets without p, $B_t^{\text{ONL}} \setminus \{p\}$. If $S_t^p \setminus S_t \neq \emptyset$, let $s_t(p) := S_t^p \setminus S_t$ denote the packet that is suppressed by $p \in S_t$. Note that $s_t(p)$ is well defined, since $|S_t^p \setminus S_t| \leq 1$. For simplicity, if $S_t^p \setminus S_t = \emptyset$, let $s_t(p)$ be a dummy packet with value 0 and an infinite deadline.

In addition, we need further preliminaries. Consider the optimal provisional schedule S for a time step τ. A time step $\tau' > \tau$ is called a *tight time step* in S if

$$|\{p' \in S \mid d(p') \leq \tau'\}| = \tau' - \tau \ .$$

Roughly speaking, a tight time step is a time step that prevents further packets with an earlier deadline from being added to the schedule. Another characterization is the following. A tight time step is a time step in which a packet is scheduled on its deadline, i.e., τ' is a tight time step in S if and only if a packet $p \in S$ exists with $S(p) = d(p) = \tau'$. Note that, for a suppressed packet p' and each packet $p \in S$ with $S(p) \leq \tau'$ where τ' denotes the earliest tight time step that is greater or equal than $d(p')$, $v(p') \leq v(p)$.

For a time step τ and a packet value δ, (τ, δ) is called a *level*. An intuition for a level is the following: For a certain level (τ, δ), our strategy provides an increase of the total value of sent packets by at least δ in each time step less or equal than τ. If in one of these time steps the actual value of the sent packet is less than δ, our strategy can nevertheless guarantee the claimed increase by amortization.

For a set of levels L and a time step τ, let $\delta_L^{\max}(\tau)$ denote the value of the level in L with maximum value that contains time step τ, i.e.,

$$\delta_L^{\max}(\tau) := \max\{\delta' \mid (\tau', \delta') \in L, \tau \leq \tau'\} \ .$$

If $\{\delta' \mid (\tau', \delta') \in L, \tau \leq \tau'\} = \emptyset$, define $\delta_L^{\max}(\tau) := 0$. Roughly speaking, the function $\delta_L^{\max}$ describes the upper envelope of all levels in L. Figure 1 depicts the optimal provisional schedule for a time step τ including a tight time step. In addition, a set of levels L and $\delta_L^{\max}$ are depicted.

Our strategy uses a parameter $\beta > 1$. For each event σ_t, a set of levels L_t is defined. Initially, define $L_0 := \emptyset$. For each event σ_t, our strategy does the following.

1. If σ_t is the send event of a time step τ:

   Define $p_f \in S_{t-1}$ as the first packet in S_{t-1} according to the canonical order and

   $$p_m := \operatorname*{argmax}_{p' \in S_{t-1}} (v(p') + (\beta - 1) \cdot v(s_{t-1}(p')))$$

   (ties are broken in favor of the smallest packet according to the canonical order).

   If

   $$\max\{v(p_f), \delta_{L_{t-1}}^{\max}(\tau)\} \geq \frac{v(p_m) + (\beta-1) \cdot v(s_{t-1}(p_m)}{\beta} \ ,$$

   send p_f. Otherwise, send p_m.

2. After event σ_t, i.e., after a packet has been sent or has arrived:

   Compute S_t and set

   $$L_t := L_{t-1} \cup \{(\tau', \min\{v(p) \mid p \in S_t, d(p) \leq \tau'\}) \mid \tau' \text{ is a tight time step in } S_t\} \ .$$

Note that our strategy does not have to compute the optimal provisional schedules completely new at each event. Instead, it suffices to remove and to insert the respective packets. Further, note that our strategy does not have to accumulate all levels. Instead, it suffices to retain only the values of $\delta_{L_t}^{\max}$ for future time steps.

The following theorem shows that our strategy achieves the best known competitive ratio in the deterministic case. The proof of this theorem follows in the next section.

THEOREM 4.1. *The above strategy achieves the competitive ratio $r := 2\sqrt{2} - 1 \approx 1.8284$ for $\beta := 1 + 1/\sqrt{2}$.*

The above strategy can easily be transformed to the following memoryless strategy that does not have to store $\delta_{L_t}^{\max}$. For each event σ_t, our memoryless strategy does the following.

1. If σ_t is the send event of a time step τ:

   Define $p_f \in S_{t-1}$ as the first packet in S_{t-1} according to the canonical order and

   $$p_m := \operatorname*{argmax}_{p' \in S_{t-1}} (v(p') + v(s_{t-1}(p'))/2)$$

   (ties are broken in favor of the smallest packet according to the canonical order).

   If

   $$v(p_f) \geq \frac{v(p_m) + v(s_{t-1}(p_m))/2}{\beta} \quad ,$$

   send p_f. Otherwise, send p_m.

2. After event σ_t, i.e., after a packet has been sent or has arrived:

   Compute S_t.

The following theorem shows that our memoryless strategy achieves a better competitive ratio than all previously known deterministic memoryless and even non-memoryless strategies. The proof of this theorem follows in the next section.

THEOREM 4.2. *The memoryless strategy achieves the competitive ratio $r := (2\beta^2 + \beta - 5)/2 \approx 1.893$ for $\beta := 4\cos((\pi - \arccos(3\sqrt{3}/16))/3)/\sqrt{3}$ (β is the largest real root of $X^3 - 4X + 1$).*

5 Analysis of the Strategies

In this section, we give the proofs of Theorem 4.1 and Theorem 4.2.

5.1 Proof of Theorem 4.1

Proof. Let OPT denote an optimal offline strategy, and let ONL denote our online strategy. W.l.o.g. assume that the sequence of packets $p_1 p_2 \cdots$ sent by OPT is in canonical order, i.e., for each $1 \leq i < j$, either $p_i \prec p_j$ or p_i is sent before p_j arrives. Note that each sequence of sent packets can easily be converted into canonical order by rearranging its packets.

For simplicity, we assume that, at each send event, ONL and OPT always send a packet, i.e., if the buffer of one of these strategies is empty at a send event, we suppose that a dummy packet of value 0 is sent. Further, we assume, for each time step, that a packet is scheduled in the optimal provisional schedule, i.e., we suppose that the schedule is filled up with dummy packets with an infinite deadline and value 0 (see Figure 1). The definitions of p_f and p_m depend on a send event. Nevertheless, we refer to p_f and p_m without explicitly referencing a send event. It is always obvious from the context which send event is meant.

Our proof is based on a potential function argument. In the following, we give some basic ideas. If we could show, for each time step, that the value of the packet sent by OPT in this step is at most r times more valuable than the packet sent by ONL in this step, the theorem would follow immediately. Of course this is not true: Time steps can exist such that the value of the packet sent by OPT in this step is much larger than the value of the packet sent by ONL in this step. For this case, there are two basic scenarios.

In the first scenario, OPT sends a packet p that was not sent by ONL yet, i.e., p is still stored in the buffer of ONL. In this case, value is lent on p, i.e., a $(r-1)/2$ fraction of the value of p is allocated to this time step. This lent value has to be amortized by the time step when p leaves the buffer of ONL (either because it is sent or because its deadline expires). The lent value cannot be amortized for more than one packet in one time step. Hence, we maintain, for each time step, the invariant that all packets on which value is lent can be scheduled in a feasible schedule (see Lemma 5.1). This guarantees, for each time step, that the deadline of at most one packet on which value is lent expires.

In the second scenario, OPT sends a packet p that was sent by ONL in a previous time step. In this case, we allocate value that is amortized in previous time steps as follows. For a certain level (τ, δ), ONL provides an increase of the total value of sent packets by at least δ in each time step less or equal than τ. If in one of these time steps the actual value of the sent packet is less than δ, ONL can nevertheless guarantee the claimed increase by amortization (see the $V(L_t, S_t)$ term in the potential function which is defined later).

Hence, ONL can guarantee the value $\delta_{L_t}^{\max}(d(p))$ at the send event σ_t when OPT sends p. It remains to amortize the value $v(p) - \delta_{L_t}^{\max}(d(p))$ in previous time steps (see the $A(L_t, B_t^{\mathrm{OPT}} \setminus B_t^{\mathrm{ONL}})$ term in the potential function which is defined later).

In the following, these basic ideas are formalized. For a set of levels L and a packet p, define

$$m_L(p) := \min\{v(p), \delta_L^{\max}(d(p))\} \ .$$

Then, for a set of levels L and a set of packets P, define

$$A(L, P) := \sum_{p \in P} (v(p) - m_L(p)) \ .$$

The following observation states an upper bound on $A(L_t, B_t^{\mathrm{OPT}} \setminus B_t^{\mathrm{ONL}}) - A(L_{t-1}, B_{t-1}^{\mathrm{OPT}} \setminus B_{t-1}^{\mathrm{ONL}})$.

Observation 1. *Fix an event σ_t and define*

$$\Delta A := A(L_t, B_t^{OPT} \setminus B_t^{ONL}) - A(L_{t-1}, B_{t-1}^{OPT} \setminus B_{t-1}^{ONL}) \ .$$

- *Suppose that σ_t is an arrival event. Then*

$$\Delta A \leq 0 \ .$$

- *Suppose that σ_t is a send event in which ONL sends the packet p and OPT sends the packet q.*
  - *If $q \notin B_{t-1}^{ONL}$ and $p \in B_t^{OPT}$,*

$$\Delta A \leq v(p) - m_{L_{t-1}}(p) - (v(q) - m_{L_{t-1}}(q)) \ .$$

  - *If $q \in B_{t-1}^{ONL}$ and $p \in B_t^{OPT}$,*

$$\Delta A \leq v(p) - m_{L_{t-1}}(p) \ .$$

  - *If $q \notin B_{t-1}^{ONL}$ and $p \notin B_t^{OPT}$,*

$$\Delta A \leq -(v(q) - m_{L_{t-1}}(q)) \ .$$

  - *If $q \in B_{t-1}^{ONL}$ and $p \notin B_t^{OPT}$,*

$$\Delta A \leq 0 \ .$$

Proof. For a set of levels L, a set of packets P, and a level (τ', δ'), obviously $A(L \cup (\tau', \delta'), P) \leq A(L, P)$. The successive application of this argument yields

$$A(L_t, B_{t-1}^{\mathrm{OPT}} \setminus B_{t-1}^{\mathrm{ONL}}) \leq A(L_{t-1}, B_{t-1}^{\mathrm{OPT}} \setminus B_{t-1}^{\mathrm{ONL}}) \ . \tag{5.1}$$

Suppose that σ_t is an arrival event. Then, $B_t^{\mathrm{OPT}} \setminus B_t^{\mathrm{ONL}} = B_{t-1}^{\mathrm{OPT}} \setminus B_{t-1}^{\mathrm{ONL}}$. Hence, $A(L_{t-1}, B_t^{\mathrm{OPT}} \setminus B_t^{\mathrm{ONL}}) = A(L_{t-1}, B_{t-1}^{\mathrm{OPT}} \setminus B_{t-1}^{\mathrm{ONL}})$. Together with Inequality (5.1), this yields the first statement of the observation.

Suppose that σ_t is a send event in which ONL sends the packet p and OPT sends the packet q. Then, $p \in B_{t-1}^{\mathrm{ONL}} \setminus B_t^{\mathrm{ONL}}$ and $q \in B_{t-1}^{\mathrm{OPT}} \setminus B_t^{\mathrm{OPT}}$.

- If $q \notin B_{t-1}^{\mathrm{ONL}}$ and $p \in B_t^{\mathrm{OPT}}$,

$$B_t^{\mathrm{OPT}} \setminus B_t^{\mathrm{ONL}} = \{p\} \cup (B_{t-1}^{\mathrm{OPT}} \setminus B_{t-1}^{\mathrm{ONL}}) \setminus \{q\} \ .$$

- If $q \in B_{t-1}^{\mathrm{ONL}}$ and $p \in B_t^{\mathrm{OPT}}$,

$$B_t^{\mathrm{OPT}} \setminus B_t^{\mathrm{ONL}} = \{p\} \cup (B_{t-1}^{\mathrm{OPT}} \setminus B_{t-1}^{\mathrm{ONL}}) \ .$$

- If $q \notin B_{t-1}^{\mathrm{ONL}}$ and $p \notin B_t^{\mathrm{OPT}}$,

$$B_t^{\mathrm{OPT}} \setminus B_t^{\mathrm{ONL}} = (B_{t-1}^{\mathrm{OPT}} \setminus B_{t-1}^{\mathrm{ONL}}) \setminus \{q\} \ .$$

- If $q \in B_{t-1}^{\mathrm{ONL}}$ and $p \notin B_t^{\mathrm{OPT}}$,

$$B_t^{\mathrm{OPT}} \setminus B_t^{\mathrm{ONL}} = (B_{t-1}^{\mathrm{OPT}} \setminus B_{t-1}^{\mathrm{ONL}}) \ .$$

Together with Inequality (5.1), this yields the second statement of the observation. □

For a set of levels L and the optimal provisional schedule S for a time step τ, define

$$V(L, S) := \sum_{p \in S} (\delta_L^{\max}(S(p)) - m_L(p)) \ .$$

Note that $\sum_{p \in S} \delta_L^{\max}(S(p)) = \sum_{\tau' \geq \tau+1} \delta_L^{\max}(\tau')$, since for each time step a packet is scheduled in S. The following observation states an upper bound on $V(L_t, S_t) - V(L_{t-1}, S_{t-1})$. The proof of this observation is similar to the proof of Observation 1.

Observation 2. *Fix an event σ_t in a time step τ and define*

$$\Delta V := V(L_t, S_t) - V(L_{t-1}, S_{t-1}) \ .$$

- *If σ_t is an arrival event,*

$$\Delta V \leq 0 \ .$$

- *If σ_t is a send event in which ONL sends p_f,*

$$\Delta V \leq m_{L_{t-1}}(p_f) - \delta_{L_{t-1}}^{\max}(\tau) \ .$$

- *If σ_t is a send event in which ONL sends p_m,*

$$\begin{aligned} \Delta V \quad \leq \quad & m_{L_{t-1}}(p_m) - v(s_{t-1}(p_m)) \\ & + m_{L_{t-1}}(p_f) - \delta_{L_{t-1}}^{\max}(\tau) \ . \end{aligned}$$

Proof. In order to show

$$V(L_t, S_t) \leq V(L_{t-1}, S_t) \ , \tag{5.2}$$

we prove, for a set of levels L, the optimal provisional schedule S, and a level (τ', δ'), where τ' is a tight time step in S and, for each packet $p \in S$ with $d(p) \leq \tau'$, $v(p) \geq \delta'$ that

$$V(L \cup (\tau', \delta'), S) \leq V(L, S) \ .$$

The successive application of this argument yields Inequality (5.2).

The last inequality follows immediately if, for each $p \in S$,

$$(5.3) \quad \delta^{\max}_{L\cup(\tau',\delta')}(S(p)) - m_{L\cup(\tau',\delta')}(p) \leq \delta^{\max}_{L}(S(p)) - m_L(p) \ .$$

Obviously, for each $p \in S$, $m_{L\cup(\tau',\delta')}(p) \geq m_L(p)$. Hence, Inequality (5.3) is true if, for each $p \in S$, $\delta^{\max}_{L\cup(\tau',\delta')}(S(p)) \leq \delta^{\max}_{L}(S(p))$.

Suppose that a $p \in S$ exists with $\delta^{\max}_{L\cup(\tau',\delta')}(S(p)) > \delta^{\max}_{L}(S(p))$. Then, $\delta^{\max}_{L\cup(\tau',\delta')}(S(p)) = \delta'$ and $S(p) \leq \tau'$. Hence, $d(p) \leq \tau'$ since τ' is a tight time step in S. This implies that $\delta^{\max}_{L\cup(\tau',\delta')}(d(p)) \geq \delta'$. Then, $\min\{v(p), \delta^{\max}_{L\cup(\tau',\delta')}(d(p))\} \geq \delta'$ since $v(p) \geq \delta'$ due to the definition of δ'. As a consequence,

$$\begin{aligned} &\delta^{\max}_{L\cup(\tau',\delta')}(S(p)) - m_{L\cup(\tau',\delta')}(p) \\ &= \delta' - \min\{v(p), \delta^{\max}_{L\cup(\tau',\delta')}(d(p))\} \\ &\leq \delta' - \delta' = 0 \ . \end{aligned}$$

Further, $\delta^{\max}_{L}(S(p)) - m_L(p) \geq 0$ since $\delta^{\max}_{L}(S(p)) \geq \delta^{\max}_{L}(d(p))$. Altogether, this yields Inequality (5.3).

In the following, we show the three statements of the observation. Fix an event σ_t in a time step τ.

- Suppose that σ_t is an arrival event in which a packet p arrives.

  Due to Inequality (5.2), it remains to show that $V(L_{t-1}, S_t) \leq V(L_{t-1}, S_{t-1})$. Obviously,

$$\begin{aligned} \sum_{p' \in S_t} \delta^{\max}_{L_{t-1}}(S(p')) &= \sum_{\tau' \geq \tau+1} \delta^{\max}_{L_{t-1}}(\tau') \\ &= \sum_{p' \in S_{t-1}} \delta^{\max}_{L_{t-1}}(S(p')) \ . \end{aligned}$$

  Hence, it remains to show that

$$\sum_{p' \in S_t \setminus S_{t-1}} m_{L_{t-1}}(p) \geq \sum_{p' \in S_{t-1} \setminus S_t} m_{L_{t-1}}(p) \ .$$

  Only the following three possibilities exist for S_t: $S_t = S_{t-1}$, $S_t = \{p\} \cup S_{t-1}$, or $S_t = \{p\} \cup S_{t-1} \setminus \{s_t(p)\}$. The above inequality follows immediately in the first and second case. In the third case, we have to show that

$$\begin{aligned} &\min\{v(p), \delta^{\max}_{L_{t-1}}(d(p))\} \\ &\geq \min\{v(s_t(p)), \delta^{\max}_{L_{t-1}}(d(s_t(p)))\} \ . \end{aligned}$$

  Obviously, $v(p) \geq v(s_t(p))$. Further, a tight time step $\tau' \geq d(p)$ exists in S_{t-1} such that each packet in S_{t-1} with a deadline smaller or equal than τ' has a value of at least $v(s_t(p))$. Hence, a level (τ', δ') with $\delta' \geq v(s_t(p))$ exists in L_{t-1}. As a consequence, $\delta^{\max}_{L_{t-1}}(d(p)) \geq v(s_t(p))$.

- Suppose that σ_t is a send event in which ONL sends p_f.

  Due to Inequality (5.2), it remains to show that

$$\begin{aligned} V(L_{t-1}, S_t) \leq\ & V(L_{t-1}, S_{t-1}) \\ &+ m_{L_{t-1}}(p_f) - \delta^{\max}_{L_{t-1}}(\tau) \ . \end{aligned}$$

  Obviously, $S_t = S_{t-1} \setminus \{p_f\}$ and, for each $p \in S_t$, $S_t(p) = S_{t-1}(p)$. Hence,

$$\begin{aligned} &\sum_{p \in S_t} (\delta^{\max}_{L_{t-1}}(S_t(p)) - m_{L_{t-1}}(p)) \\ &= \sum_{p \in S_{t-1}} (\delta^{\max}_{L_{t-1}}(S_{t-1}(p)) - m_{L_{t-1}}(p)) \\ &\quad - (\delta^{\max}_{L_{t-1}}(\tau) - m_{L_{t-1}}(p_f)) \ . \end{aligned}$$

- Suppose that σ_t is a send event in which ONL sends p_m.

  Due to Inequality (5.2), it remains to show that

$$\begin{aligned} V(L_{t-1}, S_t) \leq\ & V(L_{t-1}, S_{t-1}) \\ &+ m_{L_{t-1}}(p_m) - v(s_{t-1}(p_m)) \\ &+ m_{L_{t-1}}(p_f) - \delta^{\max}_{L_{t-1}}(\tau) \ . \end{aligned}$$

  Obviously,

$$\begin{aligned} &\sum_{p \in S_{t-1}} \delta^{\max}_{L_{t-1}}(S_{t-1}(p)) \\ &= \delta^{\max}_{L_{t-1}}(\tau) + \sum_{p \in S_t} \delta^{\max}_{L_{t-1}}(S_t(p)) \ . \end{aligned}$$

  Hence, it remains to show that

$$(5.4) \quad m_{L_{t-1}}(p_f) + m_{L_{t-1}}(p_m) + \sum_{p \in S_t} m_{L_{t-1}}(p) \geq v(s_{t-1}(p_m)) + \sum_{p \in S_{t-1}} m_{L_{t-1}}(p) \ .$$

  Only the following two possibilities exist for S_t: $S_t = S_{t-1} \setminus \{p_m\}$ or $S_t = \{s_{t-1}(p_m)\} \cup S_{t-1} \setminus \{p_m, f\}$, where f is the packet in S_{t-1} with minimum value and a deadline smaller or equal to the first tight time step in S_{t-1} (ties are broken in favor of the largest packet according to the canonical order).

– Suppose that $S_t = S_{t-1} \setminus \{p_m\}$.

A tight time step τ' exists in S_{t-1} that prevents $s_{t-1}(p_m)$ from being scheduled in S_{t-1}. The value of each packet in S_{t-1} with a deadline smaller or equal than τ' is at least $v(s_{t-1}(p_m))$. Hence, a level (τ', δ') with $\delta' \geq v(s_{t-1}(p_m))$ exists in L_{t-1}. This implies that $\delta^{\max}_{L_{t-1}}(d(p_f)) \geq v(s_{t-1}(p_m))$. Then,

$$\min\{v(p_f), \delta^{\max}_{L_{t-1}}(d(p_f))\} \geq v(s_{t-1}(p_m)) \ ,$$

since $v(p_f) \geq v(s_{t-1}(p_m))$. This yields Inequality (5.4).

– Suppose that $S_t = \{s_{t-1}(p_m)\} \cup S_{t-1} \setminus \{p_m, f\}$.

Due to the definition of f, $v(p_f) \geq v(f)$ and a level (τ', δ') with $\tau' \geq d(f)$ and $\delta' \geq v(f)$ exists in L_{t-1}. Hence,

$$\begin{aligned} &\min\{v(p_f), \delta^{\max}_{L_{t-1}}(d(p_f))\} \\ &\quad \geq \ \min\{v(f), \delta^{\max}_{L_{t-1}}(d(f))\} \ . \end{aligned}$$

Further, a tight time step $\tau'' \geq d(s_{t-1}(p_m))$ exists in S_{t-1} that prevents $s_{t-1}(p_m)$ from being scheduled in S_{t-1}. Hence, a level (τ'', δ'') with $\delta'' \geq v(s_{t-1}(p_m))$ exists in L_{t-1}. This implies that

$$\begin{aligned} &\min\{v(s_{t-1}(p_m)), \delta^{\max}_{L_{t-1}}(d(s_{t-1}(p_m)))\} \\ &\quad = \ v(s_{t-1}(p_m)) \ . \end{aligned}$$

Altogether, this yields Inequality (5.4).

This concludes the proof of the observation. □

For each event σ_t, define the potential function

$$\begin{aligned} \Phi_t \ := \ & r \sum_{p \in S^{\mathrm{ONL}}_t} v(p) - \sum_{p \in S^{\mathrm{OPT}}_t} v(p) \\ & - A(L_t, B^{\mathrm{OPT}}_t \setminus B^{\mathrm{ONL}}_t) - V(L_t, S_t) \\ & + \frac{r-1}{2} \sum_{p \in C_t} v(p) \ , \end{aligned}$$

where $C_t \subseteq B^{\mathrm{ONL}}_t \setminus B^{\mathrm{OPT}}_t$ is specified later (see Lemma 5.1). Initially, define $C_0 := \emptyset$. In order to prove the theorem, we show that the potential function Φ_t is monotonously increasing in t, for appropriately chosen sets of packets $C_t \subseteq B^{\mathrm{ONL}}_t \setminus B^{\mathrm{OPT}}_t$.

Obviously, $\Phi_0 = 0$, since $S^{\mathrm{ONL}}_0 = S^{\mathrm{OPT}}_0 = B^{\mathrm{OPT}}_0 = B^{\mathrm{ONL}}_0 = L_0 = C_0 = \emptyset$ by definition. Then, if the potential function Φ_t is monotonously increasing in t, $\Phi_T \geq 0$, where σ_T is the last event. As a consequence, $r \sum_{p \in S^{\mathrm{ONL}}_T} v(p) \geq \sum_{p \in S^{\mathrm{OPT}}_T} v(p)$, since $A(L_t, B^{\mathrm{OPT}}_t \setminus B^{\mathrm{ONL}}_t) \geq 0$ and $V(L_t, S_t) \geq 0$, for each event σ_t, and $C_T \subseteq B^{\mathrm{ONL}}_T \setminus B^{\mathrm{OPT}}_T = \emptyset$. This yields the theorem.

It remains to show that the potential function Φ_t is monotonously increasing in t, for appropriately chosen sets of packets $C_t \subseteq B^{\mathrm{ONL}}_t \setminus B^{\mathrm{OPT}}_t$. The following lemma states how to choose, for each event σ_t, $C_t \subseteq B^{\mathrm{ONL}}_t \setminus B^{\mathrm{OPT}}_t$ such that certain lower bounds on $\sum_{p' \in C_t} v(p') - \sum_{p' \in C_{t-1}} v(p')$ are true.

Lemma 5.1. *Define*

$$\Delta C_t := \sum_{p' \in C_t} v(p') - \sum_{p' \in C_{t-1}} v(p') \ .$$

For each event σ_t, the set of packets $C_t \subseteq B^{ONL}_t \setminus B^{OPT}_t$ can be chosen such that the following is true.

(a) If σ_t is a send event in which ONL and OPT send the same packet p,

$$\Delta C_t \geq -v(p_f) \ .$$

(b) If σ_t is a send event in which ONL sends a packet $p \in C_{t-1}$ and OPT sends a packet $q \notin S_{t-1}$,

$$\Delta C_t \geq -v(p) - v(p_f) \ .$$

(c) If σ_t is a send event in which ONL sends a packet $p \notin C_{t-1}$ and OPT sends a packet $q \notin S_{t-1}$,

$$\Delta C_t \geq -v(p_f) \ .$$

(d) If σ_t is a send event in which ONL sends p_f and OPT sends a packet $q \in S_{t-1} \setminus \{p_f\}$,

$$\Delta C_t \geq -2 \cdot v(p_f) + v(q) \ .$$

(e) If σ_t is a send event in which ONL sends $p_m \notin C_{t-1}$ and OPT sends a packet $q \in S_{t-1} \setminus \{p_m\}$,

$$\Delta C_t \geq -v(p_f) - v(s_{t-1}(q)) + v(q) \ .$$

(f) If σ_t is a send event in which ONL sends $p_m \in C_{t-1}$ and OPT sends a packet $q \in S_{t-1} \setminus \{p_m\}$,

$$\Delta C_t \geq -v(p_m) - v(p_f) - v(s_{t-1}(q)) + v(q) \ .$$

(g) If σ_t is an arrival event,

$$\Delta C_t = 0 \ .$$

Proof. In order to show the lemma, we maintain, for each event σ_t in a time step τ, the invariant

$$\forall \tau' \geq \tau : |\{p' \in C_t \mid d(p') \leq \tau'\}| \leq \tau' - \tau \ ,$$

i.e., all packets in C_t can be scheduled in a feasible schedule. Due to the invariant, at most one packet with a deadline less or equal than $\tau+1$ exists in C_t. Recall that $C_t \subseteq B_t^{\mathrm{ONL}} \setminus B_t^{\mathrm{OPT}}$. Hence, the deadline of at most one packet in C_t expires in the next time step $\tau+1$.

Obviously, the invariant is true for $C_0 = \emptyset$. By induction over t, we show how to choose appropriately the set of packets $C_t \subseteq B_t^{\mathrm{ONL}} \setminus B_t^{\mathrm{OPT}}$ such that the invariant and the statements in the lemma are true. Due to space limitations this case analysis is omitted. □

For each σ_t, the upper bounds on $A(L_t, B_t^{\mathrm{OPT}} \setminus B_t^{\mathrm{ONL}}) - A(L_{t-1}, B_{t-1}^{\mathrm{OPT}} \setminus B_{t-1}^{\mathrm{ONL}})$ from Observation 1, the upper bounds on $V(L_t, S_t) - V(L_{t-1}, S_{t-1})$ from Observation 2, and the lower bounds on $\sum_{p' \in C_t} v(p') - \sum_{p' \in C_{t-1}} v(p')$ from Lemma 5.1 are used in a straightforward case analysis to show $\Phi_t - \Phi_{t-1} \geq 0$. Due to space limitations this case analysis is omitted. □

5.2 Proof of Theorem 4.2

Proof. In our non-memoryless strategy either p_f or p_m is sent depending on the condition

$$\max\{v(p_f), \delta^{\max}_{L_{t-1}}(\tau)\} \geq \frac{v(p_m) + (\beta-1) \cdot v(s_{t-1}(p_m)}{\beta} .$$

The only difference of our memoryless strategy to our non-memoryless strategy is that this condition is replaced by

$$v(p_f) \geq \frac{v(p_m) + v(s_{t-1}(p_m))/2}{\beta} .$$

The fact that either p_f or p_m is sent based on the aforementioned condition is only exploited in the case analysis of the proof of Theorem 4.1. Other parts of the proof are not affected by a change of this condition. However, note that $A(L_t, B_t^{\mathrm{OPT}} \setminus B_t^{\mathrm{ONL}})$ and $V(L_t, S_t)$ depend on L_t and that our memoryless strategy does not compute L_t. However, it suffices that L_t is defined in the proof.

Using the above observations, we can adopt the proof of Theorem 4.1. For each event σ_t, the potential function has to be redefined

$$\begin{aligned} \Phi_t \;:=\; & r \sum_{p \in S_t^{\mathrm{ONL}}} v(p) - \sum_{p \in S_t^{\mathrm{OPT}}} v(p) \\ & - A(L_t, B_t^{\mathrm{OPT}} \setminus B_t^{\mathrm{ONL}}) - V(L_t, S_t) \\ & + \alpha \sum_{p \in C_t} v(p) \;, \end{aligned}$$

with $\alpha := (\beta^2 - 3)/2$.

For each event σ_t, the upper bounds from Observation 1, the upper bounds from Observation 2, and the lower bounds from Lemma 5.1 are used in a straightforward case analysis to show $\Phi_t - \Phi_{t-1} \geq 0$. Due to space limitations this case analysis is omitted. □

References

[1] N. Andelman, Y. Mansour, and A. Zhu. Competitive queueing policies for QoS switches. In *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 761–770, 2003.

[2] E.-C. Chang and C. Yap. Competitive on-line scheduling with level of service. *Journal of Scheduling*, 6(3):251–267, 2003.

[3] F. Y. L. Chin, M. Chrobak, S. P. Y. Fung, W. Jawor, J. Sgall, and T. Tichý. Online competitive algorithms for maximizing weighted throughput of unit jobs. *Journal of Discrete Algorithms*, 4(2):255–276, 2006.

[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. In *Proceedings of the 12th European Symposium on Algorithms (ESA)*, pages 204–215, 2004.

[6] M. Englert and M. Westermann. Lower and upper bounds on FIFO buffer management in QoS switches. In *Proceedings of the 14th European Symposium on Algorithms (ESA)*, pages 352–363, 2006.

[7] B. Hajek. On the competitiveness of on-line scheduling of unit-length packets with hard deadlines in slotted time. In *Proceedings of the 2001 Conference on Information Sciences and Systems (CISS)*, pages 434–438, 2001.

[8] A. Kesselman, Z. Lotker, Y. Mansour, B. Patt-Shamir, B. Schieber, and M. Sviridenko. Buffer overflow management in QoS switches. *SIAM Journal on Computing*, 33(3):563–583, 2004.

[9] A. Kesselman, Y. Mansour, and R. van Stee. Improved competitive guarantees for QoS buffering. *Algorithmica*, 43(1–2):63–80, 2005.

[10] F. Li, J. Sethuraman, and C. Stein. An optimal online algorithm for packet scheduling with agreeable deadlines. In *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 801–802, 2005.

[11] F. Li, J. Sethuraman, and C. Stein. Better online buffer management. In *Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007. To appear.