

# Lower and Upper Bounds on FIFO Buffer Management in QoS Switches\*

Matthias Englert

Matthias Westermann

Department of Computer Science  
RWTH Aachen  
52056 Aachen, Germany  
{englert,marsu}@cs.rwth-aachen.de

## Abstract

We consider the management of FIFO buffers for network switches providing differentiated services. In each time step, an arbitrary number of packets arrive and only one packet can be sent. The buffer can store a limited number of packets and, due to the FIFO property, the sequence of sent packets has to be a subsequence of the arriving packets. The differentiated service model is abstracted by attributing each packet with a value according to its service level. A buffer management strategy can drop packets, and the goal is to maximize the sum of the values of sent packets.

For only two different packet values, we introduce the account strategy and prove that this strategy achieves an optimal competitive ratio of  $\sqrt{2} - (\sqrt{5} + 4\sqrt{2} - 3)/2 \approx 1.282$  if the buffer size tends to infinity and an optimal competitive ratio of  $(\sqrt{13} - 1)/2 \approx 1.303$  for arbitrary buffer sizes. For general packet values, the simple preemptive greedy strategy (PG) is studied. We show that PG achieves a competitive ratio of  $\sqrt{3} \approx 1.732$  which is the best known upper bound on the competitive ratio of this problem. In addition, we give a lower bound of  $1 + 1/\sqrt{2} \approx 1.707$  on the competitive ratio of PG which improves the previously known lower bound. As a consequence, the competitive ratio of PG cannot be further improved significantly.

---

\*Supported by the DFG grant WE 2842/1. A preliminary version of this paper appeared in Proceedings of the 14th Annual European Symposium on Algorithms (ESA), 2006.

# 1 Introduction

Quality of Service (QoS) guarantees for network services allow providers to address the requirements of customers by offering different levels of service. In the network setting, where traffic volumes may exceed network capacity, effective management of buffers in switches is a key to achieve QoS guarantees. We consider FIFO buffers, i.e., the buffer can store a limited number of packets and, due to the FIFO property, the sequence of sent packets has to be a subsequence of the arriving packets. By differentiating service levels, packets of different types may be treated according to the level of service they require. This model is abstracted by attributing each packet with a value according to its service level. A buffer management strategy can drop packets, and the goal is to maximize the sum of the values of sent packets.

For only two different packet values, we introduce the account strategy and prove that this strategy achieves an optimal competitive ratio of  $\sqrt{2} - (\sqrt{5 + 4\sqrt{2}} - 3)/2 \approx 1.282$  if the buffer size tends to infinity and an optimal competitive ratio of  $(\sqrt{13} - 1)/2 \approx 1.303$  for arbitrary buffer sizes. For general packet values, the simple preemptive greedy strategy (PG) is studied. We show that PG achieves a competitive ratio of  $\sqrt{3} \approx 1.732$  which is the best known upper bound on the competitive ratio of this problem. In addition, we give a lower bound of  $1 + 1/\sqrt{2} \approx 1.707$  on the competitive ratio of PG which improves the previously known lower bound. As a consequence, the competitive ratio of PG cannot be further improved significantly.

## 1.1 The Model

Time is slotted in time steps. In each time step, an arbitrary number of packets arrive, and, at the end of each time step, only one packet can be sent. Packets that are not sent can be stored in a *FIFO buffer* with a limited storage capacity for  $b$  packets. Initially, the FIFO buffer is empty. Due to the FIFO property, the sequence of sent packets has to be a subsequence of the arriving packets, i.e., if a packet  $p$  is sent before a packet  $p'$ ,  $p$  has arrived before  $p'$ .

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 can *drop* arriving packets, i.e., these packets are never stored in the buffer, or can *drop* packets stored in the buffer, i.e., these packets are deleted from the buffer and not sent. The goal of the buffer management strategy is to maximize the sum of the values of sent packets.

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 react without knowledge about the future. Online strategies 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  $\sigma$  of arriving packets, let  $\text{OPT}(\sigma)$  denote the total value produced by an optimal offline strategy. An online strategy is denoted as *c-competitive* if it produces total value at least  $\text{OPT}(\sigma)/c$ , for each input sequence  $\sigma$  of arriving packets. The value  $c$  is also called the *competitive ratio* of the online strategy.

## 1.2 Previous Work

Aiello et al. [1] introduce the model of differentiated services for FIFO buffers without preemption. Mansour, Patt-Shamir, and Lapid [11] add preemption and general packet values to this

model. Kesselman and Mansour [8] study the value of the lost packets instead of the value of the sent packets.

Kesselman et al. [7] 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  $\approx 1.983$ . In addition, they give the previously best known lower bound of  $(1 + \sqrt{5})/2 \approx 1.618$  on the competitive ratio of the preemptive greedy strategy. Bansal et al. [5] study a modification of the preemptive greedy strategy and show that this strategy achieves a competitive ratio of  $7/4$  which is the previously best known upper bound on the competitive ratio of this problem. Note that their modification does not improve the overall performance of the strategy [6]. The best known lower bound on the competitive ratio of this problem is  $\approx 1.419$  [9].

The following results refer to the case where only two different packet values are considered. Lotker and Patt-Shamir [10] present a strategy that achieves a competitive ratio of  $\approx 1.30448$ . Kesselman et al. [7] show a lower bound of  $\approx 1.282$  on the competitive ratio. Andelman [2] presents a randomized strategy that achieves a competitive ratio of  $5/4$ . Further, he gives a lower bound of  $\approx 1.197$  on the competitive ratio of any randomized strategy.

Azar and Richter [4] extend the buffer management problem to multi-queues, i.e., several incoming queues have to be served by delivering packets that arrive at these queues through one output port, one packet per time step. They present a generic technique that transforms a strategy for a single queue to a strategy for several queues. They show that the competitive ratio of the constructed strategy is at most twice the competitive ratio of the single queue strategy.

### 1.3 Our Contributions

In Section 2, only two packet values are considered. We introduce the account strategy and prove that this strategy achieves an optimal competitive ratio of  $\sqrt{2} - (\sqrt{5 + 4\sqrt{2}} - 3)/2 \approx 1.282$  if the buffer size tends to infinity and an optimal competitive ratio of  $(\sqrt{13} - 1)/2 \approx 1.303$  for arbitrary buffer sizes. Note that this is the first non-trivial optimal result in this area.

In Section 3, general packet values are considered. We study the preemptive greedy strategy (PG) introduced in [9]. This is a simple strategy that can be implemented efficiently. We show that PG achieves a competitive ratio of  $\sqrt{3} \approx 1.732$  which is the best known upper bound on the competitive ratio of this problem. In addition, we give a lower bound of  $1 + 1/\sqrt{2} \approx 1.707$  on the competitive ratio of PG which improves the previously known lower bound of  $(1 + \sqrt{5})/2 \approx 1.618$ . Hence, the gap between upper and lower bound for PG narrows to approximately  $1/40$ . We conjecture that the lower bound is tight. As a consequence, new approaches are needed, since the competitive ratio of PG cannot be further improved significantly. Based on our lower bound for PG and our optimal account strategy for two packet values, we propose an approach to tackle the problems of PG.

## 2 Two Packet Values

In this section, only two packet values 1 and  $\alpha > 1$  are considered. A packet of value 1 is called 1-packet, and a packet of value  $\alpha$  is called  $\alpha$ -packet. Define

$$r := \frac{\sqrt{13} - 1}{2} \approx 1.303 \quad \text{and}$$

$$r_\infty := \sqrt{2} - \frac{\sqrt{5 + 4\sqrt{2}} - 3}{2} \approx 1.282 .$$

The following theorem states two lower bounds on the competitive ratio of any deterministic strategy. The proof for the first statement of this theorem can be found, e.g., in [3], and the proof for the second statement of this theorem can be found, e.g., in [7].

**Theorem 1.** *Consider only two packet values 1 and  $\alpha > 1$ .*

1. *The competitive ratio of any deterministic strategy is at least  $r$ , if the buffer size is 2.*
2. *The competitive ratio of any deterministic strategy is at least  $r_\infty$ , if the buffer size tends to infinity.*

The *account strategy (ACC)* tries to preempt 1-packets from the buffer in order to avoid losing too many  $\alpha$ -packets in case of a buffer overflow. The number of preempted 1-packets has to be chosen carefully. Obviously, the total number of preempted 1-packets should not exceed  $(x - 1)$  times the total value of sent packets if we want to achieve a competitive ratio of  $x$ . Hence, one basic idea of ACC is to preempt at most  $(x - 1) \cdot \alpha$  1-packets for each  $\alpha$ -packet entering the buffer and at most  $(x - 1)$  1-packets for each sent 1-packet. ACC tries to preempt as much 1-packets as possible without violating this constraint.

We define  $\text{ACC}(x)$  with one parameter  $x \geq 1$  which is the competitive ratio we aim for and which is therefore used to determine how aggressive the strategy is with respect to preemption.  $\text{ACC}(x)$  uses an *account*  $a$  which is initially set to 0. Basically, each packet sent by  $\text{ACC}(x)$  increases the account by  $(x - 1)$  times its own value, and each preempted 1-packet decreases the account by 1. More precisely, for each time step,  $\text{ACC}(x)$  does the following.

1. For each arriving packet  $p$ , do the following.
  - (a) If there is an unoccupied location in the buffer, store  $p$ . Otherwise, if a 1-packet is stored in the buffer, drop the 1-packet which is closest to the front of the buffer and store  $p$ .
  - (b) If  $p$  is an  $\alpha$ -packet that is stored in the buffer (observe that stored  $\alpha$ -packets are never dropped), increase the account  $a$  by  $(x - 1) \cdot \alpha$ .
  - (c) If the buffer is completely filled with  $\alpha$ -packets, reset the account  $a$  to 0.
2. After all packets have arrived, do the following.
  - (a) As long as the first packet is a 1-packet and  $a \geq 1$ , drop this packet, which is called *preempted*, and decrease the account  $a$  by 1.
  - (b) Send the first packet. If this packet is a 1-packet, increase the account by  $(x - 1)$ .
  - (c) If no packet is stored in the buffer, reset the account  $a$  to 0.

The following theorem shows that ACC achieves optimal competitive ratios.

**Theorem 2.** *Consider only two packet values 1 and  $\alpha > 1$ .*

1. *ACC( $r$ ) achieves a competitive ratio of  $r$  for arbitrary buffer sizes.*
2. *ACC( $r_\infty$ ) achieves a competitive ratio of  $r_\infty$  if the buffer size tends to infinity.*

*Proof.* We define a particular optimal offline strategy OPT (compare [10]). For each input sequence, the set of feasible work conserving schedules, i.e., the feasible schedules in which a packet is sent in each time step in which the buffer is not empty, is a matroid. Hence, a greedy strategy can compute an optimal solution. First, OPT considers all  $\alpha$ -packets in increasing order of their arrival, and thereafter, OPT considers all 1-packets in increasing order of their arrival.

We show that the analysis can be restricted to input sequences that satisfy the following two properties.

1. In each time step, except for the  $b - 1$  last ones, ACC sends a packet, where  $b$  denotes the buffer size.
2. In each  $\alpha$ -overflow time step, i.e., the buffer of ACC is completely filled with  $\alpha$ -packets, exactly  $b$   $\alpha$ -packets and no 1-packets arrive.

The following two observations show that we can assume w.l.o.g. that each input sequence satisfies the two properties.

**Observation 3.** *For each input sequence  $\sigma$ , it exists an input sequence for which ACC has at least the same competitive ratio and which satisfies the first property.*

*Proof.* After each time step in  $\sigma$  in which the buffer of ACC is empty, insert  $b - 1$  additional time steps in which no packets arrive. The set of packets sent by ACC does not change and the value of an optimal solution can only increase. Hence, the competitive ratio of ACC for the altered input sequence is at least as large as for the original sequence  $\sigma$ .

Now, we partition the input sequence into subsequences. A new subsequence starts after  $b - 1$  consecutive time steps in which no new packets arrive. Obviously, we can assume that there are never more than  $b - 1$  consecutive time steps in which no new packets arrive.

Fix a subsequence  $\sigma^{(i)}$ . The buffers of ACC and OPT are empty at the beginning of  $\sigma^{(i)}$ , since any packet stored in the buffers of size  $b$  is sent during one of the previous  $b$  time steps and no new packets arrive in between. Furthermore, the buffers of ACC and OPT are empty at the end of  $\sigma^{(i)}$ . However, the buffer of ACC is only empty for the last  $b - 1$  time steps of  $\sigma^{(i)}$ , due to the construction of the subsequences. In all other time steps, a packet is sent.

Finally note that the competitive ratio of ACC for one of the subsequences is at least as large as for the original sequence  $\sigma$ .  $\square$

**Observation 4.** *For each input sequence  $\sigma$ , it exists an input sequence for which ACC has at least the same competitive ratio and which satisfies both properties.*

*Proof.* In each  $\alpha$ -overflow time step of  $\sigma$ , add  $b$   $\alpha$ -packets to the arriving packets. None of these  $\alpha$ -packets can be stored by ACC. The set of packets sent by ACC does not change and the value of an optimal solution can only increase. Hence, the competitive ratio of ACC for the altered input sequence is at least as large as for the original sequence  $\sigma$ .

For each  $\alpha$ -overflow time step, we remove all arriving packets except for  $b$   $\alpha$ -packets. The sets of packets sent by ACC and OPT do not change, since in each time step only the  $b$  most valuable arriving packets are relevant.  $\square$

Now, fix an input sequence  $\sigma$  that satisfies both properties. We partition  $\sigma$  into time intervals. A time interval ends with an  $\alpha$ -overflow time step, and the next time interval begins with the time step following this  $\alpha$ -overflow. Let  $P_i$  denote the set of packets arriving in the  $i$ -th time interval, and let  $m$  denote the total number of different time intervals, i.e., each arriving packet in  $\sigma$  is in  $\bigcup_{i=1}^m P_i$ .

Let  $\text{ACC}^1(P_i)$  ( $\text{ACC}^\alpha(P_i)$ ) denote the subset of 1-packets ( $\alpha$ -packets) in  $P_i$  that are sent by ACC, and let  $\text{OPT}^1(P_i)$  ( $\text{OPT}^\alpha(P_i)$ ) denote the subset of 1-packets ( $\alpha$ -packets) in  $P_i$  that are sent by OPT. In order to show the theorem, we prove the claimed competitive ratio for each set of packets  $P_i$ , i.e., we prove, for each  $P_i$ ,

$$\frac{|\text{OPT}^1(P_i)| + \alpha \cdot |\text{OPT}^\alpha(P_i)|}{|\text{ACC}^1(P_i)| + \alpha \cdot |\text{ACC}^\alpha(P_i)|} \leq r \text{ (or } r_\infty, \text{ respectively)} . \quad (1)$$

The following two lemmata give upper bounds on the number of packets sent by OPT.

**Lemma 5.** *ACC sends the same number of packets as OPT from each set  $P_i$  with  $i < m$ .*

*Proof.* We prove the lemma by induction over  $i$ . Fix an  $i < m$  and assume that ACC sends the same number of packets as OPT from each set  $P_j$  with  $j < i$ . As a consequence, ACC and OPT start sending packets from  $P_i$  in the same time step.

Let  $t$  denote the last time step in which a packet from  $P_i$  arrives, i.e., the  $\alpha$ -overflow time step. In time step  $t + b - 1$ , ACC sends a packet from  $P_i$ , since in time step  $t$  the buffer of ACC is completely filled with  $\alpha$ -packets and the last  $\alpha$ -packet in the buffer is a packet from  $P_i$ .

- OPT does not send more packets from  $P_i$  than ACC.

Each packet is stored in the buffer for at most  $b - 1$  time steps. As a consequence, after time step  $t + b - 1$ , OPT can only send packets that arrive after time step  $t$ , and hence, these packets are not in  $P_i$ .

- OPT does not send less packets from  $P_i$  than ACC.

Assume for contradiction that OPT sends less packets from  $P_i$  than ACC. As a consequence, in time step  $t + b - 1$  a packet from  $P_j$  with  $j > i$  is sent by OPT. Hence, OPT does not send all  $\alpha$ -packets from  $P_i$ , since  $b$   $\alpha$ -packets arrive in time step  $t$ . When one of these  $\alpha$ -packets not sent by OPT was considered to be included in the schedule of OPT, it could have been added without making the schedule infeasible. This is a contradiction to our definition of OPT.

This concludes the proof of the lemma.  $\square$

Let  $D \subseteq P_m$  denote the set of preempted 1-packets from  $P_m$ , i.e.,  $D := \{p \in P_m \mid p \text{ is preempted by ACC}\}$ .

**Lemma 6.**  $\sum_{i=1}^m (|\text{OPT}^1(P_i)| + |\text{OPT}^\alpha(P_i)|) \leq \sum_{i=1}^m (|\text{ACC}^1(P_i)| + |\text{ACC}^\alpha(P_i)|) + |D|$ .

*Proof.* In the following, we add packets from  $D$  to the schedule of ACC, such that the resulting schedule is maximal, i.e., the schedule becomes infeasible if another packet is added. As a consequence, the schedule of OPT contains the same number of packets as our modified schedule, since the set of feasible work conserving schedules is a matroid.

Consider the last time step  $t$  in which the buffer of ACC is completely filled with packets. Let  $D'$  denote the set of packets that are either stored in the buffer of ACC at time step  $t$  or arrive after  $t$  and that are not contained in the schedule of ACC. Observe that each packet in  $D'$  is a preempted 1-packet from  $P_m$ , since  $t$  is the last time step in which the buffer of ACC is completely filled with packets. Hence,  $D' \subseteq D$ .

Adding as much packets as possible from  $D'$  to the schedule of ACC, such that the resulting schedule is feasible, produces a maximal schedule. Obviously, adding an additional packet that is not in  $D'$  to the schedule makes the schedule infeasible.  $\square$

Now, we are able to show Inequality (1) for  $P_m$ . Combining Lemma 5 and Lemma 6 yields  $|\text{OPT}^1(P_m)| + |\text{OPT}^\alpha(P_m)| \leq |\text{ACC}^1(P_m)| + |\text{ACC}^\alpha(P_m)| + |D|$ . Since ACC sends all  $\alpha$ -packets from  $P_m$ ,  $|\text{OPT}^\alpha(P_m)| = |\text{ACC}^\alpha(P_m)|$ . Hence,

$$|\text{OPT}^1(P_m)| + \alpha \cdot |\text{OPT}^\alpha(P_m)| \leq |\text{ACC}^1(P_m)| + \alpha \cdot |\text{ACC}^\alpha(P_m)| + |D| . \quad (2)$$

When the last packet of  $P_{m-1}$  arrives, the buffer of ACC is completely filled with  $\alpha$ -packets and the account  $a$  is reset to 0. Hence, the preemption of later arriving packets, i.e., packets in  $P_m$ , is caused by packets from  $P_m$  that are sent by ACC. As a consequence,

$$|D| \leq (r-1) \cdot (|\text{ACC}^1(P_m)| + \alpha \cdot |\text{ACC}^\alpha(P_m)|) .$$

In combination with Inequality (2), this gives

$$\begin{aligned} |\text{OPT}^1(P_m)| + \alpha \cdot |\text{OPT}^\alpha(P_m)| &\leq |\text{ACC}^1(P_m)| + \alpha \cdot |\text{ACC}^\alpha(P_m)| \\ &\quad + (r-1) \cdot (|\text{ACC}^1(P_m)| + \alpha \cdot |\text{ACC}^\alpha(P_m)|) \\ &= r \cdot (|\text{ACC}^1(P_m)| + \alpha \cdot |\text{ACC}^\alpha(P_m)|) . \end{aligned}$$

To show Inequality (1) for each  $P_i$  with  $i < m$ , we need to know by how much the number of  $\alpha$ -packets sent by OPT exceeds the number of  $\alpha$ -packets sent by ACC. For a  $P_i$  from which ACC sends only  $\alpha$ -packets, Inequality (1) holds obviously.

Consider a  $P_i$  with  $i < m$  from which ACC sends at least one 1-packet and  $b+y$   $\alpha$ -packets (ACC sends at least  $b$   $\alpha$ -packets from  $P_i$ ). The only  $\alpha$ -packets that cannot be sent by ACC are the ones arriving in the  $\alpha$ -overflow time step. For each  $\alpha$ -packet in the buffer of ACC at this time step that is already sent by OPT, OPT can store one additional  $\alpha$ -packet that cannot be sent by ACC.

The following lemma gives an upper bound on the number of  $\alpha$ -packets sent by OPT but not by ACC.

**Lemma 7.** *Consider a set  $P_i$  with  $i < m$  from which  $\text{ACC}(x)$  sends at least one 1-packet and  $b+y$   $\alpha$ -packets. At most*

$$\left\lfloor \frac{b-1+x+y \cdot (x-1)}{(x-1) \cdot \alpha + x} \right\rfloor$$

*$\alpha$ -packets in the buffer of  $\text{ACC}(x)$  are already sent by OPT right before the  $\alpha$ -overflow time step of  $P_i$ .*

*Proof.* Consider the latest time step  $t$  before the  $\alpha$ -overflow time step in which the number of  $\alpha$ -packets in the buffer of ACC that are already sent by OPT is increased from  $n - 1$  to  $n$ . Hence, ACC sends a 1-packet  $p$  and OPT sends an  $\alpha$ -packet that arrived after  $p$  and is stored in the buffer of ACC. Each  $\alpha$ -packet in the buffer of ACC that is already sent by OPT arrived later than  $p$ . Let  $q$  denote the first  $\alpha$ -packet in the buffer of ACC that is already sent by OPT, and let  $t'$  denote the time step in which  $q$  arrives.

Each  $\alpha$ -packet in the buffer of ACC has increased the account  $a$  by  $(x - 1) \cdot \alpha$ . In addition, the account  $a$  is increased by  $z \cdot (x - 1)$ , where  $z$  denotes the number of 1-packets sent by ACC from  $t'$  to  $t$ <sup>1</sup>. Observe that the account  $a$  is not reset to 0 from  $t'$  to  $t$ , since the 1-packet  $p$  is stored in the buffer from  $t'$  to  $t$ . However, the value of the account  $a$  is less than 1 right before  $p$  is sent by ACC, since otherwise  $p$  would have been preempted. Hence, at least  $n \cdot (x - 1) \cdot \alpha + z \cdot (x - 1) - 1$  1-packets are preempted from  $t'$  to  $t$ . All the preempted 1-packets arrive before  $p$ .

Since only one  $\alpha$ -packet can be sent by OPT in each time step, at least  $n - 1$  packets are sent from  $t'$  to  $t$ . In fact,  $z + y' \geq n - 1$  packets are sent from  $t'$  to  $t$ , where  $y'$  denotes the number of  $\alpha$ -packets sent by ACC from  $t'$  to  $t$ . Note that  $y \geq y'$ .

After the arrival of  $q$  in the time step  $t'$ , there are less or equal than  $b - 1$  other packets in the buffer of ACC and all of them arrived earlier than  $q$ . Until time step  $t$ , at least  $n \cdot (x - 1) \cdot \alpha + z \cdot (x - 1) - 1$  of them are preempted,  $z + y' \geq n - 1$  of them are sent, and  $p$  is still in the buffer. Hence,  $n \cdot (x - 1) \cdot \alpha + z \cdot (x - 1) - 1 + z + y' + 1 \leq b - 1$ .

Altogether,

$$\begin{aligned}
b - 1 + x + y \cdot (x - 1) &\geq b - 1 + x + y' \cdot (x - 1) \\
&\geq b - 1 + x + (n - 1 - z) \cdot (x - 1) \\
&\geq n \cdot (x - 1) \cdot \alpha + z \cdot (x - 1) - 1 + z + y' + 1 \\
&\quad + x + (n - 1 - z) \cdot (x - 1) \\
&= n \cdot (x - 1) \cdot \alpha + n \cdot (x - 1) + z + y' + 1 \\
&\geq n \cdot (x - 1) \cdot \alpha + n \cdot (x - 1) + n \\
&= n \cdot ((x - 1) \cdot \alpha + x) ,
\end{aligned}$$

which concludes the proof of the lemma. □

Due to Lemma 5,

$$\begin{aligned}
\frac{|\text{OPT}^1(P_i)| + \alpha \cdot |\text{OPT}^\alpha(P_i)|}{|\text{ACC}^1(P_i)| + \alpha \cdot |\text{ACC}^\alpha(P_i)|} &\leq \frac{\alpha \cdot |\text{OPT}^\alpha(P_i)|}{|\text{ACC}^1(P_i)| + \alpha \cdot |\text{ACC}^\alpha(P_i)| - |\text{OPT}^1(P_i)|} \\
&= \frac{\alpha \cdot |\text{OPT}^\alpha(P_i)|}{(\alpha - 1) \cdot |\text{ACC}^\alpha(P_i)| + |\text{OPT}^\alpha(P_i)|} .
\end{aligned}$$

Hence, it remains to show that

$$(\alpha - x) \cdot |\text{OPT}^\alpha(P_i)| \leq x \cdot (\alpha - 1) \cdot |\text{ACC}^\alpha(P_i)| ,$$

for  $x := r$  and  $x := r_\infty$  with  $b \rightarrow \infty$ . Due to Lemma 7, this inequality is equivalent to

$$(\alpha - x) \cdot \left( b + y + \left\lfloor \frac{b - 1 + x + (x - 1) \cdot y}{(x - 1) \cdot \alpha + x} \right\rfloor \right) \leq x \cdot (\alpha - 1) \cdot (b + y) ,$$

<sup>1</sup>From  $t'$  to  $t$  denotes the time interval from  $t'$  to  $t$  excluding time step  $t$ .



which is equivalent to

$$(\alpha - x) \cdot \left[ \frac{b - 1 + x + (x - 1) \cdot y}{(x - 1) \cdot \alpha + x} \right] \leq (x - 1) \cdot \alpha \cdot (b + y) . \quad (3)$$

- Suppose that  $x := r_\infty$  and  $b \rightarrow \infty$ .

Observe that

$$\begin{aligned} & \lim_{b \rightarrow \infty} (\alpha - r_\infty) \cdot \left( \frac{1 - 1/b + r_\infty/b}{(r_\infty - 1) \cdot \alpha + r_\infty} \right) - (r_\infty - 1) \cdot \alpha \\ &= (\alpha - r_\infty) \cdot \left( \frac{1}{(r_\infty - 1) \cdot \alpha + r_\infty} \right) - (r_\infty - 1) \cdot \alpha \leq 0 . \end{aligned}$$

Then, it follows that

$$\lim_{b \rightarrow \infty} (\alpha - r_\infty) \cdot \left( \frac{b - 1 + r_\infty}{(r_\infty - 1) \cdot \alpha + r_\infty} \right) - (r_\infty - 1) \cdot \alpha \cdot b \leq 0 .$$

Finally, Inequality (3) can be shown as follows

$$\begin{aligned} & \lim_{b \rightarrow \infty} (\alpha - r_\infty) \cdot \left[ \frac{b - 1 + r_\infty + (r_\infty - 1) \cdot y}{(r_\infty - 1) \cdot \alpha + r_\infty} \right] - (r_\infty - 1) \cdot \alpha \cdot (b + y) \\ & \leq \lim_{b \rightarrow \infty} (\alpha - r_\infty) \cdot \left( \frac{b - 1 + r_\infty + (r_\infty - 1) \cdot y}{(r_\infty - 1) \cdot \alpha + r_\infty} \right) - (r_\infty - 1) \cdot \alpha \cdot (b + y) \\ & \leq \lim_{b \rightarrow \infty} (\alpha - r_\infty) \cdot \left( \frac{b - 1 + r_\infty}{(r_\infty - 1) \cdot \alpha + r_\infty} \right) - (r_\infty - 1) \cdot \alpha \cdot b \leq 0 . \end{aligned}$$

- Suppose that  $x := r$ .

Defining  $k := b + (r - 1) \cdot y$ , we get

$$\begin{aligned} & (\alpha - r) \cdot \left[ \frac{b - 1 + r + (r - 1) \cdot y}{(r - 1) \cdot \alpha + r} \right] - (r - 1) \cdot \alpha \cdot (b + y) \\ &= (\alpha - r) \cdot \left[ \frac{k - 1 + r}{(r - 1) \cdot \alpha + r} \right] - (r - 1) \cdot \alpha \cdot (k + (2 - r) \cdot y) \\ & \leq (\alpha - r) \cdot \left[ \frac{k - 1 + r}{(r - 1) \cdot \alpha + r} \right] - (r - 1) \cdot \alpha \cdot k . \end{aligned}$$

We distinguish two cases.

- Suppose that  $k \geq 5/2$ .

Observe that

$$\begin{aligned} & (\alpha - r) \cdot \left[ \frac{k - 1 + r}{(r - 1) \cdot \alpha + r} \right] - (r - 1) \cdot \alpha \cdot k \\ & \leq (\alpha - r) \cdot \left( \frac{k - 1 + r}{(r - 1) \cdot \alpha + r} \right) - (r - 1) \cdot \alpha \cdot k \\ &= (\alpha - r) \cdot \left( \frac{-1 + r}{(r - 1) \cdot \alpha + r} \right) + \frac{-(r - 1)^2 \cdot \alpha^2 + (1 + r - r^2) \cdot \alpha - r}{(r - 1) \cdot \alpha + r} \cdot k . \end{aligned}$$

The last term decreases with increasing  $k$ . Hence, Inequality (3) can be shown as follows

$$\begin{aligned} & (\alpha - r) \cdot \left( \frac{k - 1 + r}{(r - 1) \cdot \alpha + r} \right) - (r - 1) \cdot \alpha \cdot k \\ & \leq (\alpha - r) \cdot \left( \frac{3/2 + r}{(r - 1) \cdot \alpha + r} \right) - \frac{5(r - 1) \cdot \alpha}{2} \leq 0 . \end{aligned}$$

– Suppose that  $2 \leq k < 5/2$ .

Observe that

$$0 \leq \frac{k - 1 + r}{(r - 1) \cdot \alpha + r} < \frac{3/2 + r}{(r - 1) \cdot \alpha + r} \leq \frac{3/2 + r}{2r - 1} < 2 .$$

As a consequence,  $\lfloor (k - 1 + r) / ((r - 1) \cdot \alpha + r) \rfloor$  equals either 0 or 1. If it equals 0, Inequality (3) follows obviously. Otherwise,  $k \geq (r - 1) \cdot \alpha + 1$ .

If  $k = 2$ , this gives  $\alpha \leq 1/(r - 1)$ . Hence, Inequality (3) can be shown as follows

$$(\alpha - r) \cdot \left\lfloor \frac{k - 1 + r}{(r - 1) \cdot \alpha + r} \right\rfloor - (r - 1) \cdot \alpha \cdot k = (\alpha - r) - 2(r - 1) \cdot \alpha \leq \left( \frac{1}{r - 1} - r \right) - 2 = 0 .$$

If  $k > 2$ ,  $y \geq 1$ , since otherwise  $k$  would be integral. Hence, Inequality (3) can be shown as follows

$$\begin{aligned} & (\alpha - r) \cdot \left\lfloor \frac{k - 1 + r}{(r - 1) \cdot \alpha + r} \right\rfloor - (r - 1) \cdot \alpha \cdot (k + (2 - r) \cdot y) \\ & \leq (\alpha - r) - (r - 1) \cdot \alpha \cdot (k + (2 - r)) \\ & \leq (\alpha - r) - (r - 1) \cdot \alpha \cdot ((r - 1) \cdot \alpha + 1 + (2 - r)) \leq 0 . \end{aligned}$$

This concludes the proof of the theorem.  $\square$

### 3 The Preemptive Greedy Strategy

Kesselman, Mansour, and van Stee [9] introduce the preemptive greedy strategy (PG) with the parameter  $\beta > 1$ . When a packet  $p$  arrives, PG does the following.

1. Find the first packet, i.e., the packet closest to the front of the buffer,  $p'$ , with  $v(p') \leq v(p)/\beta$ . If such a packet  $p'$  exists, drop it ( $p'$  is called *preempted* by  $p$ ).
2. If there is an unoccupied location in the buffer, store  $p$  in the buffer.
3. Otherwise, find a packet  $p'$  with the smallest value among the packets in the buffer. If  $v(p') < v(p)$ , drop  $p'$  ( $p'$  is called *ejected* by  $p$ ) and store  $p$  in the buffer. Otherwise, drop  $p$  ( $p$  is called *rejected*).

Bansal et al. [5] study a modified version of PG. The only difference is that step 1 of PG is substituted by the following.

1. Find the first packet  $p'$ , with  $v(p') \leq v(p)/\beta$  and  $v(p')$  is not larger than the value of the packet stored after  $p'$  in the buffer. If such a packet exists, drop it.

Note that this modification does not improve the overall performance of the strategy [6].

New approaches are needed, since, due to the following lower and upper bound, the competitive ratio of PG cannot be further improved significantly. A basic concept of PG is that, for each arriving packet  $p$ , the first packet whose value is at most  $v(p)/\beta$  is preempted. At first sight, it seems more reasonable that, instead, the packet with the smallest value is preempted. But in fact, the preemption of the first packet whose value is suitable small enough is a crucial property to achieve a competitive ratio smaller than 2. However, this can turn out to be a great disadvantage as the first input sequence in the following lower bound shows. This disadvantage diminishes with increasing  $\beta$ . On the other hand, too few packets are preempted for larger  $\beta$  as the second input sequence in the following lower bound shows. An approach to tackle this problem might be the following: If, for large  $\beta$ , the value of a single packet does not suffice to preempt another packet, the values of more than one packet are combined for preemption. Note that, in the case of only two packet values, we achieve with this idea an optimal strategy.

### 3.1 Lower Bound

The following theorem gives an lower bound on the competitive ratio of PG.

**Theorem 8.** *The competitive ratio of PG is at least  $1 + 1/\sqrt{2} \approx 1.707$ .*

*Proof.* Fix an even buffer size  $b$ . Depending on  $\beta$ , we distinguish the following two cases.

- Suppose that  $\beta \leq 2 + \sqrt{2}$ .

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

- Phase  $1 \leq i < n$  consists of  $b/2$  time steps. In time step 1 of the  $i$ -th phase, at first  $b$  packets of value  $\varepsilon$  and finally  $b/2$  packet of value  $\beta^i$  arrive. In the remaining  $b/2 - 1$  time steps, new packets do not arrive.
- Phase  $n$  consists of one time step. In this time step,  $b$  packets of value  $\beta^{n-1}$  arrive.

For this input sequence, PG produces value

$$\lim_{\varepsilon \rightarrow 0} \sum_{i=1}^{n-1} \left( \frac{b}{2} \cdot \varepsilon \right) + b \cdot \beta^{n-1} = b \cdot \beta^{n-1} ,$$

and the optimal value is

$$\sum_{i=1}^{n-1} \left( \frac{b}{2} \cdot \beta^i \right) + b \cdot \beta^{n-1} = b \cdot \frac{3\beta^n - 2\beta^{n-1} - \beta}{2(\beta - 1)} .$$

Hence, the competitive ratio is

$$\lim_{n \rightarrow \infty} \frac{3\beta^n - 2\beta^{n-1} - \beta}{2(\beta^n - \beta^{n-1})} = 1 + \frac{\beta}{2(\beta - 1)} \geq 1 + \frac{1}{\sqrt{2}} .$$

- Suppose that  $\beta > 2 + \sqrt{2}$ .

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

- Phase 1 consists of  $b - 1$  time steps. In time step 1, at first  $b - 1$  packets of value 1 and finally one packet of value  $\alpha < \beta$  arrive. In each of the remaining  $b - 2$  time steps, one packet of value  $\alpha$  arrives.
- Phase  $1 < i < n$  consists of  $b - 1$  time steps. In each of these time steps, one packet of value  $\alpha^i$  arrives.
- Phase  $n$  consists of one time step. In this time step,  $b$  packets of value  $\alpha^{n-1}$  arrive.

For this input sequence, PG produces value

$$\sum_{i=0}^{n-2} ((b-1) \cdot \alpha^i) + b \cdot \alpha^{n-1} = b \cdot \frac{\alpha^n - \frac{1}{b} \cdot \alpha^{n-1} - \frac{b-1}{b}}{\alpha - 1},$$

and the optimal value is

$$\sum_{i=1}^{n-1} ((b-1) \cdot \alpha^i) + b \cdot \alpha^{n-1} = b \cdot \frac{(2 - \frac{1}{b}) \cdot \alpha^n - \alpha^{n-1} - \frac{b-1}{b} \cdot \alpha}{\alpha - 1}.$$

Hence, the competitive ratio is

$$\begin{aligned} & \lim_{\alpha \rightarrow \beta} \lim_{n \rightarrow \infty} \lim_{b \rightarrow \infty} \frac{(2 - \frac{1}{b}) \cdot \alpha^n - \alpha^{n-1} - \frac{b-1}{b} \cdot \alpha}{\alpha^n - \frac{1}{b} \cdot \alpha^{n-1} - \frac{b-1}{b}} \\ &= \lim_{\alpha \rightarrow \beta} \lim_{n \rightarrow \infty} \frac{2\alpha^n - \alpha^{n-1} - \alpha}{\alpha^n - 1} = \lim_{\alpha \rightarrow \beta} \frac{2\alpha - 1}{\alpha} = 1 + \frac{\beta - 1}{\beta} \geq 1 + \frac{1}{\sqrt{2}}. \end{aligned}$$

This concludes the proof of the theorem. □

### 3.2 Upper Bound

The following theorem gives an upper bound on the competitive ratio of PG.

**Theorem 9.** *PG achieves a competitive ratio of  $\sqrt{3} \approx 1.732$  for  $\beta = 2 + \sqrt{3}$ .*

*Proof.* Let OPT denote an optimal offline strategy. We assume that OPT only stores packets in its buffer that are sent by OPT. Further, we assume that, at the arrival of each packet, the buffer of PG is completely filled with packets. If there are unoccupied locations in the buffer of PG, it is assumed that dummy packets of value 0 are stored at these locations which are always at the end of the buffer. Hence, each arriving packet either preempts another packet, ejects another packet, or is rejected.

Fix an input sequence of arriving packets. This input sequence can also be regarded as a sequence  $\sigma = \sigma_1 \sigma_2 \dots$  of arrival and send events, where each arrival of a new packet corresponds to an arrival event and each sending of a packet corresponds to a send event. The event sequence  $\sigma$  is partitioned into time steps, where the first time step starts with the first event and a new time step starts right after each send event.

Let  $S_t^{\text{pg}}$  ( $S_t^{\text{opt}}$ ) denote the set of packets sent by PG (OPT) by the end of event  $\sigma_t$ , i.e., all packets sent in the events  $\sigma_1, \dots, \sigma_t$  (including  $\sigma_t$ ). Let  $B_t^{\text{pg}}$  ( $B_t^{\text{opt}}$ ) denote the set of packets stored in the buffer of PG (OPT) at the end of  $\sigma_t$ . For a packet  $p \in B_t^{\text{pg}}$ , we call  $c_t(p)$  the *charge* of  $p$  at the end of  $\sigma_t$ . Further, we call  $D_t$  the set of packets with a *deposit* at the end

of  $\sigma_t$ . Note that charges and deposits are two independent concepts we use. Initially,  $D_0 := \emptyset$ . The goal is to choose  $c_t(p)$  and  $D_t$  in such a way that, for each event  $\sigma_t$ , the main inequality

$$\sum_{p \in S_t^{\text{pg}}} r \cdot v(p) + \sum_{p \in B_t^{\text{pg}}} c_t(p) \geq \sum_{p \in S_t^{\text{opt}} \cup D_t} v(p)$$

is true, with  $r := \sqrt{3}$ . As a consequence, this yields the theorem.

Let  $\Delta_t^{\text{pg}}$  ( $\Delta_t^{\text{opt}}$ ) denote the alterations of the left (right) side of the main inequality at the event  $\sigma_t$ , i.e.,

$$\begin{aligned} \Delta_t^{\text{pg}} &:= \sum_{p \in S_t^{\text{pg}} \setminus S_{t-1}^{\text{pg}}} r \cdot v(p) + \sum_{p \in B_t^{\text{pg}}} c_t(p) - \sum_{p \in B_{t-1}^{\text{pg}}} c_{t-1}(p) \quad \text{and} \\ \Delta_t^{\text{opt}} &:= \sum_{p \in (S_t^{\text{opt}} \cup D_t) \setminus (S_{t-1}^{\text{opt}} \cup D_{t-1})} v(p) . \end{aligned}$$

Obviously, the main inequality is true before the first event, since packets have not been sent so far and the buffers and the set of packets with a deposit are empty. Hence, it is sufficient to show, for each event  $\sigma_t$ ,  $\Delta_t^{\text{pg}} \geq \Delta_t^{\text{opt}}$ , since this yields the main inequality.

First, we give an intuition for the basic ideas of the proof. Then, we present the formal proof. The basic idea for the set  $D_t$  is simple. Packets stored exclusively in the buffer of OPT at the end of event  $\sigma_t$ , especially packets already sent by PG, could be a problem, if PG cannot send a packet, i.e., the buffer of PG is empty, when those packets are sent by OPT. The left side of the main inequality is not increased at these events, and it is crucial for the proof that the same is true for the right side of the main inequality. Hence, these packets have to be contained in  $D_t$ . Intuitively, PG has already gained enough value to pay these packets in advance, i.e., before they are sent by OPT.

The basic idea for  $c_t(p)$  is the following. In case of a send event  $\sigma_t$  in which OPT sends a much more valuable packet than PG that is not in  $D_{t-1}$ , the right side of the main inequality is increased by a large amount and we have to compensate this by increasing the charge of packets stored in the buffer of PG. It is fairly unproblematic to charge a packet up to  $(r - 1)$  times its own value because if such a packet is sent by PG and OPT in the same send event, the left side of the main inequality is still increased by the same amount as the right side of the main inequality. In any case, larger charges are only allowed for packets that are exclusively in the buffer of PG.

In case of a buffer overflow in the buffer of PG in which a charged packet is ejected, this charge has to be transferred to another packet in the buffer of PG. This is problematic for an ejected packet that is charged by more than  $(r - 1)$  times its own value, since, after this charge is transferred to another packet in the buffer of PG, there might be a packet charged by more than  $(r - 1)$  times its own value that is not exclusively in the buffer of PG. Therefore we introduce the concept of buddies. A packet stored exclusively in the buffer of PG might be charged by  $2(r - 1)$  times its own value only if there is another packet in the buffer of PG that is not charged at all. We call the packet with no charge buddy for the packet with the high charge.

Unfortunately, the precise definition of charges is slightly more complicated. Before we define the charges in detail, we need some preliminaries. For each two packets  $p$  and  $p'$ , we write  $p \prec p'$  if  $p$  arrives before  $p'$  in the input sequence. Further, for each packet  $p$  and the undefined symbol  $\perp$ ,  $p \prec \perp$ ,  $\perp \prec p$ , and  $\perp \prec \perp$ . Each  $p \in B_t^{\text{pg}}$  can have assigned another

$s_t(p)$	$c_t(p)$	comment
BC	$(r - 2) \cdot v(p)$	buddy with credit
B	0	buddy
U	$(r/\beta) \cdot v(p) + (2 - r) \cdot v_t^{\min}(p)$	unproblematic
E	$(r - 1) \cdot v(p)$	exclusively in $B_t^{\text{pg}}$ , i.e., not in $B_t^{\text{opt}}$
EB	$2(r - 1) \cdot v(p)$	exclusively in $B_t^{\text{pg}}$ with buddy

Figure 1: Definition of the charge  $c_t(p)$  of a packet  $p \in B_t^{\text{pg}}$  at the end of event  $\sigma_t$ . The charges are listed in increasing order, e.g., a packet in state E is at least as much charged as a packet of same value in state U. Note that the charge in case of state BC is negative. Further, note that  $v_t^{\min}(p) \leq v(p)$ . If  $v_t^{\min}(p) = v(p)$ , the charges in state U and E are the same for packet  $p$ .

$p' \in B_t^{\text{pg}}$  as *buddy* if  $p \prec p'$ . However, each  $p' \in B_t^{\text{pg}}$  is assigned as buddy for at most one other packet. If  $p \in B_t^{\text{pg}}$  has assigned another  $p' \in B_t^{\text{pg}}$  as buddy at the end of event  $\sigma_t$ , define  $b_t(p) := p'$ , otherwise, define  $b_t(p) := \perp$ . Further, for each  $p \notin B_t^{\text{pg}}$ ,  $b_t(p) := \perp$ . Finally, for each  $p \in B_t^{\text{pg}}$ , define  $v_t^{\min}(p) := \min\{v(p') \mid B_t^{\text{pg}} \ni p' \preceq p\}$ .

Each  $p \in B_t^{\text{pg}}$  is in one of the five states BC, B, U, E, and EB. Let  $s_t(p)$  denote the state of  $p$  at the end of event  $\sigma_t$ , and define  $s_t(\perp) := \perp$ . Let  $\text{BC}_t$ ,  $\text{B}_t$ ,  $\text{U}_t$ ,  $\text{E}_t$ , and  $\text{EB}_t$  denote the set of packets that are in state BC, B, U, E, and EB, respectively, at the end of event  $\sigma_t$ . The initial state of each packet is B, and dummy packets of value 0 are always in state B. The charge  $c_t(p)$  of a packet  $p$  at the end of event  $\sigma_t$  is defined in Figure 1. Note that the charge of a packet, except for packets in state U, does not change as long as this packet stays in the same state. The charge of a packet in state U can only increase, since  $v_t^{\min}(p) \leq v_{t+1}^{\min}(p)$ .

Let  $P_t$  denote the set of packets that are preempted by PG by the end of event  $\sigma_t$ . For each packet  $p$ , if  $p$  preempts another packet  $p'$ , define  $d(p) := p'$ , otherwise, define  $d(p) := \perp$ . A packet  $p$  transitively preempts another packet  $p'$ , if either  $d(p) = p'$ ,  $p$  preempts a packet that transitively preempts  $p'$ , or  $p$  ejects a packet that transitively preempts  $p'$ . For each  $p' \in P_t$ , if  $p'$  is transitively preempted by a packet  $p \in B_t^{\text{pg}}$ , define  $\hat{d}_t(p') := p$ , otherwise, define  $\hat{d}_t(p') := \perp$ . For each  $p' \notin P_t$ , define  $\hat{d}_t(p') := \perp$ . Figure 2 gives an overview of our notation.

In order to prove the theorem, we show the following five invariants by induction over the event sequence  $\sigma$ . To shorten notation, we define  $X_t := (P_t \cup S_t^{\text{pg}}) \cap (B_t^{\text{opt}} \setminus D_t)$ .

- I1:  $\Delta_t^{\text{pg}} \geq \Delta_t^{\text{opt}}$ .
- I2: If  $p \in \text{E}_t \cup \text{EB}_t$ , then  $p \notin B_t^{\text{opt}}$ .
- I3: If  $p \in \text{EB}_t$ , then  $b_t(p) \in \text{BC}_t \cup \text{B}_t$ .
- I4: If  $p \in X_t$ , then  $\hat{d}_t(p) \in \text{BC}_t \cup \text{B}_t$ .
- I5: If  $p \in B_t^{\text{pg}} \setminus \text{BC}_t$ , then  $b_t^{-1}(p) \prec d(p)$ .

Observe that the invariants have only to be verified in the following cases.

- I1: Always.
- I2: For each packet  $p \in (\text{E}_t \cup \text{EB}_t) \setminus (\text{E}_{t-1} \cup \text{EB}_{t-1})$ .

notation	comment
$S_t^{\text{pg}}, S_t^{\text{opt}}$	The set of packets sent by PG and OPT by the end of $\sigma_t$ .
$B_t^{\text{pg}}, B_t^{\text{opt}}$	The set of packets stored in the buffer of PG and OPT at the end of $\sigma_t$ .
$P_t$	The set of packets preempted by PG by the end of $\sigma_t$ .
$D_t$	The set of packets with a deposit at the end of $\sigma_t$ .
$X_t$	Short notation for $(P_t \cup S_t^{\text{pg}}) \cap (B_t^{\text{opt}} \setminus D_t)$ .
$c_t(p)$	The charge of the packet $p \in B_t^{\text{pg}}$ at the end of $\sigma_t$ , which is determined by its state.
$s_t(p)$	The state of the packet $p \in B_t^{\text{pg}}$ . Each packet $p \in B_t^{\text{pg}}$ is in one of the five states BC, B, U, E, or EB.
$v_t^{\min}(p)$	The value of the least valuable packet stored in the buffer of PG in front of $p \in B_t^{\text{pg}}$ .
$b_t(p)$	The buddy packet of the packet $p$ . Equals $\perp$ if $p$ has no buddy or $p \notin B_t^{\text{pg}}$ .
$b_t^{-1}(p)$	The packet for which $p$ is a buddy. Equals $\perp$ if $p$ is not a buddy for another packet.
$d(p)$	The packet that is preempted by $p$ . Equals $\perp$ if $p$ does not preempt another packet.
$\hat{d}_t(p)$	The packet $p' \in B_t^{\text{pg}}$ that transitively preempted $p$ . Equals $\perp$ if $p$ was not preempted, i.e., $p \notin P_t$ , or there is no packet in the buffer of PG which transitively preempted $p$ .
$p \prec p'$	The packet $p$ arrives before the packet $p'$ .

Figure 2: Informal overview of our notation.

- I3: For each packet  $p$  with  $(p \in \text{EB}_t \setminus \text{EB}_{t-1}) \vee (b_{t-1}(p) \in (\text{BC}_{t-1} \cup \text{B}_{t-1}) \setminus (\text{BC}_t \cup \text{B}_t)) \vee (b_{t-1}(p) \neq b_t(p))$ .
- I4: For each packet  $p$  with  $(p \in X_t \setminus X_{t-1}) \vee (\hat{d}_{t-1}(p) \in (\text{BC}_{t-1} \cup \text{B}_{t-1}) \setminus (\text{BC}_t \cup \text{B}_t)) \vee (\hat{d}_{t-1}(p) \neq \hat{d}_t(p))$ .
- I5: For each packet  $p$  with  $(p \in (B_t^{\text{pg}} \setminus \text{BC}_t) \setminus (B_{t-1}^{\text{pg}} \setminus \text{BC}_{t-1})) \vee (b_{t-1}^{-1}(p) \neq b_t^{-1}(p))$ .

The following lemma is used to dramatically reduce the number of cases we have to consider. Whenever we encounter a situation during the induction where  $B_t^{\text{opt}} \not\subseteq P_t \cup S_t^{\text{pg}} \cup B_t^{\text{pg}}$ , we manipulate the buffer contents of OPT in such a way that  $B_t^{\text{opt}} \subseteq P_t \cup S_t^{\text{pg}} \cup B_t^{\text{pg}}$ . The five invariants continue to hold after this manipulation. Thereafter, we can continue the induction.

**Lemma 10.** *Assume that  $\sigma_t$  is the first event with  $B_t^{\text{opt}} \not\subseteq P_t \cup S_t^{\text{pg}} \cup B_t^{\text{pg}}$ . Then, the buffer contents of OPT can be manipulated in such a way that  $B_t^{\text{opt}} \subseteq P_t \cup S_t^{\text{pg}} \cup B_t^{\text{pg}}$  and the five invariants continue to hold.*

*Proof.* Assume that  $\sigma_t$  is the first event with  $B_t^{\text{opt}} \not\subseteq P_t \cup S_t^{\text{pg}} \cup B_t^{\text{pg}}$ , i.e., the buffer of OPT contains a packet that was ejected or rejected by PG. Since  $\sigma_t$  is the first event with  $B_t^{\text{opt}} \not\subseteq$

$P_t \cup S_t^{\text{pg}} \cup B_t^{\text{pg}}$ , a packet  $p$  must have been ejected or rejected by PG in  $\sigma_t$ . This also implies that  $\sigma_t$  is an arrival event. In the following, we assume that  $p$  is rejected by PG but stored in the buffer of OPT. The arguments for the case that  $p$  is ejected are analogous.

Since OPT stores  $p$  in its buffer and the buffer of PG is completely filled with packets, there exists a packet  $q \in B_t^{\text{pg}} \setminus B_t^{\text{opt}}$ . The value  $v(q)$  of  $q$  has to be at least as large as  $v(p)$ . Otherwise,  $q$  would have been ejected by PG and  $p$  would have been stored in the buffer of PG. Define  $v := v(p)$ .

After  $p$  arrived, we manipulate the buffer contents of OPT in the following way: The arrival time of  $p$  is set to the arrival time of  $q$ , i.e., the packets stored in the buffer of OPT are reordered such that  $p$  is placed at the position of  $q$  if  $q$  would be contained in the buffer of OPT. This reordering does not change the set of packets sent by OPT and hence, does not change the total value gained by OPT.

In addition, we manipulate the value of  $p$ . We increase the value of  $p$  to the value of  $q$ . After both manipulations, the attributes of the packet  $p \in B_t^{\text{opt}} \setminus B_t^{\text{pg}}$  are identical to the packet  $q \in B_t^{\text{pg}} \setminus B_t^{\text{opt}}$ . As a consequence,  $p$  can be identified with  $q$ , i.e., we can assume that  $p$  is actually the packet  $q$  and therefore stored in the buffer of PG.

The Invariants I3, I4, and I5 are not effected by our manipulation, since changes are not made in the buffer of PG and  $q \notin P_t \cup S_t^{\text{pg}}$ . If  $s_t(q) \notin \{E, EB\}$ , Invariant I2 is not effected either. Otherwise, set  $s_t(q) := U$  and, if  $q$  was in state EB, set  $s_t(b_{t-1}(q)) := U$  (due to I3  $b_{t-1}(q)$  exists and is in state BC or B in this case). Thus, Invariants I2–I5 continue to hold.

It remains to study the effect of our manipulation on the main inequality.

- If  $s_t(q) \notin \{E, EB\}$  the main inequality does not change.
- If  $q$  was in state E and its state changed to U, the left side of the main inequality is decreased by at most  $(r-1) \cdot v(q) - ((r/\beta) \cdot v(q) + (2-r) \cdot v_t^{\min}(q)) = (2-r) \cdot (v(q) - v_t^{\min}(q)) \leq v(q) - v$ , since  $r/\beta = 2r - 3$  and  $p$  is rejected at  $\sigma_t$ .
- If  $q$  was in state EB and its state changed to U, the left side of the main inequality is decreased by at most  $2(r-1) \cdot v(q) - ((r/\beta) \cdot v(q) + (2-r) \cdot v_t^{\min}(q)) = v(q) + (r-2) \cdot v_t^{\min}(q) \leq v(q) + (r-2) \cdot v$ . In this case, the state of  $b_{t-1}(q)$  changed from BC or B to U. This increases the left side of the main inequality by at least  $(r/\beta) \cdot v(b_{t-1}(q)) + (2-r) \cdot v_t^{\min}(b_{t-1}(q)) \geq (r/\beta) \cdot v + (2-r) \cdot v \geq (r-1) \cdot v$ . Hence, in total the left side of the main inequality is decreased by at most  $v(q) + (r-2) \cdot v - (r-1) \cdot v = v(q) - v$ .

Hence, the left side of the main inequality is decreased by at most  $v(q) - v$ . As a consequence, we can only guarantee that

$$v(q) - v + \sum_{p' \in S_\ell^{\text{pg}}} r \cdot v(p') \geq \sum_{p' \in S_\ell^{\text{opt}}} v(p')$$

after the last event  $\sigma_\ell$  in the sequence of events  $\sigma$ . This is not sufficient to show the theorem. Fortunately, by virtually increasing the value of  $p$  we have also increased  $\sum_{p' \in S_\ell^{\text{opt}}} v(p')$  by  $v(q) - v$ , i.e., the real total value of OPT is smaller by  $v(q) - v$ . Finally,

$$v(q) - v + r \cdot \text{PG}(\sigma) = v(q) - v + \sum_{p' \in S_\ell^{\text{pg}}} r \cdot v(p') \geq \sum_{p' \in S_\ell^{\text{opt}}} v(p') = \text{OPT}(\sigma) + v(q) - v .$$

which concludes the proof of the lemma.  $\square$



case		packets concerned	verification
a1	I2	–	
	I3	$q, b_{t-1}^{-1}(q)$	$s_t(q) = \perp, b_t(b_{t-1}^{-1}(q)) = p \in B_t$
	I4	$q, \{p'   \hat{d}_{t-1}(p') = q\}$	$\hat{d}_t(q) = p \in B_t, \hat{d}_{t-1}(p') = q \Rightarrow \hat{d}_t(p') = p \in B_t$
	I5	$p, q$	$b_t^{-1}(p) = b_{t-1}^{-1}(q) \prec q = d(p), q \notin B_t^{\text{pg}}$
a2	I2	–	
	I3	$q, b_{t-1}^{-1}(q)$	$s_t(q) = \perp, s_t(b_{t-1}^{-1}(q)) = s_{t-1}(b_{t-1}^{-1}(q)) \stackrel{\text{I3}}{\neq} \text{EB}$
	I4	$q$ $\{p'   \hat{d}_{t-1}(p') = q\}$	$q \notin B_{t-1}^{\text{opt}} \Rightarrow q \notin B_t^{\text{opt}} \supseteq X_t$ $\hat{d}_{t-1}(p') = q \stackrel{\text{I4}}{\Rightarrow} p' \notin X_{t-1} \cup \{q\} \supseteq X_t$
	I5	$p, q$	$b_t^{-1}(p) = \perp \prec d(p), q \notin B_t^{\text{pg}}$
a3	I2	–	
	I3	$q, b_{t-1}^{-1}(q)$	$s_t(q) = \perp, s_t(b_{t-1}^{-1}(q)) = s_{t-1}(b_{t-1}^{-1}(q)) \stackrel{\text{I3}}{\neq} \text{EB}$
	I4	$q, \{p'   \hat{d}_{t-1}(p') = q\}$	$q \in D_t \Rightarrow q \notin X_t, \hat{d}_{t-1}(p') = q \stackrel{\text{I4}}{\Rightarrow} p' \notin X_{t-1} \cup \{q\} \supseteq X_t$
	I5	$p, q$	$b_t^{-1}(p) = \perp \prec d(p), q \notin B_t^{\text{pg}}$
a4	I2	–	
	I3	$q, b_{t-1}^{-1}(q)$	$s_t(q) = \perp, b_t(b_{t-1}^{-1}(q)) = p \in B_t$
	I4	$q, \{p'   \hat{d}_{t-1}(p') = q\}$	$q \notin P_t \cup S_t^{\text{pg}} \supseteq X_t, \hat{d}_{t-1}(p') = q \Rightarrow \hat{d}_t(p') = p \in B_t$
	I5	$p, q$	$b_t^{-1}(p) = b_{t-1}^{-1}(q) \prec \perp = d(p), q \notin B_t^{\text{pg}}$
a5	I2	–	
	I3	$q, b_{t-1}^{-1}(q)$	$s_t(q) = \perp, s_t(b_{t-1}^{-1}(q)) = s_{t-1}(b_{t-1}^{-1}(q)) \stackrel{\text{I3}}{\neq} \text{EB}$
	I4	$q, \{p'   \hat{d}_{t-1}(p') = q\}$	$q \notin P_t \cup S_t^{\text{pg}} \supseteq X_t, \hat{d}_{t-1}(p') = q \stackrel{\text{I4}}{\Rightarrow} p' \notin X_{t-1} \supseteq X_t$
	I5	$p, q$	$b_t^{-1}(p) = \perp \prec \perp = d(p), q \notin B_t^{\text{pg}}$
a6	I2	–	
	I3	$q, b_{t-1}^{-1}(q), b_{t-1}(q)$	$s_t(q) = \perp, s_t(b_{t-1}^{-1}(q)) = s_{t-1}(b_{t-1}^{-1}(q)) \stackrel{\text{I3}}{\neq} \text{EB}, b_{t-1}(q) \in U_t$
	I4	$q, \{p'   \hat{d}_{t-1}(p') = q\}$ $b_{t-1}(q)$	$q \notin P_t \cup S_t^{\text{pg}} \supseteq X_t, \hat{d}_{t-1}(p') = q \stackrel{\text{I4}}{\Rightarrow} p' \notin X_{t-1} \supseteq X_t$ $b_{t-1}(q) \notin P_t \cup S_t^{\text{pg}} \supseteq X_t$
	I5	$p, q$ $b_{t-1}(q)$	$b_t^{-1}(p) = \perp \prec \perp = d(p), q \notin B_t^{\text{pg}}$ $b_t^{-1}(b_{t-1}(q)) = \perp \prec d(b_{t-1}(q))$

Figure 3: Verification of the Invariants I2–I5 for the Cases a1–a6.

Fix an arrival event  $\sigma_t$  in which a packet  $p$  arrives. We distinguish the following cases. If not mentioned otherwise, everything remains unchanged at event  $\sigma_t$ . We only consider the Invariant I1. For the verification of the Invariants I2–I5, see Figure 3.

- $p$  preempts another packet  $q$

a1:  $q \in B_{t-1} \cup BC_{t-1}$

Changes:  $b_t(b_{t-1}^{-1}(q)) := p$  and  $s_t(p) := B$

$$\text{I1: } \Delta_t^{\text{pg}} \geq 0 = \Delta_t^{\text{opt}}$$

a2:  $q \in E_{t-1} \cup EB_{t-1}$

Changes:  $s_t(p) := U$

$$\begin{aligned} \text{I1: } \Delta_t^{\text{pg}} &\geq (r/\beta) \cdot v(p) - 2(r-1) \cdot v(q) \\ &\geq (r/\beta) \cdot v(p) - 2(r-1) \cdot v(p)/\beta \\ &= ((2-r)/\beta) \cdot v(p) > 0 = \Delta_t^{\text{opt}} \end{aligned}$$

a3:  $q \in U_{t-1}$

Changes:  $s_t(p) := U$  and  $D_t := D_{t-1} \cup \{q\}$

$$\begin{aligned} \text{I1: } \Delta_t^{\text{pg}} &\geq (r/\beta) \cdot v(p) - ((r/\beta) \cdot v(q) + (2-r) \cdot v(q)) \\ &\geq (r/\beta) \cdot (\beta \cdot v(q)) - (r-1) \cdot v(q) = v(q) = \Delta_t^{\text{opt}} \end{aligned}$$

•  $p$  ejects another packet  $q$

a4:  $q \in B_{t-1} \cup BC_{t-1}$

Changes:  $s_t(p) := B$  and  $b_t(b_{t-1}^{-1}(q)) := p$

$$\text{I1: } \Delta_t^{\text{pg}} \geq 0 = \Delta_t^{\text{opt}}$$

a5:  $q \in E_{t-1} \cup U_{t-1}$

Changes:  $s_t(p) := U$

$$\begin{aligned} \text{I1: } \Delta_t^{\text{pg}} &\geq (r/\beta) \cdot v(p) + (2-r) \cdot v_{t-1}^{\min}(p) - (r-1) \cdot v(q) \\ &\geq (r/\beta) \cdot v(q) + (2-r) \cdot v(q) - (r-1) \cdot v(q) \\ &= ((2r-3) + (2-r) - (r-1)) \cdot v(q) = 0 = \Delta_t^{\text{opt}} \end{aligned}$$

a6:  $q \in EB_{t-1}$

Changes:  $s_t(p) := U$  and  $s_t(b_{t-1}(q)) := U$

(Due to I3,  $b_{t-1}(q) \in B_{t-1} \cup BC_{t-1}$ .)

$$\begin{aligned} \text{I1: } \Delta_t^{\text{pg}} &\geq (r/\beta) \cdot v(p) + (2-r) \cdot v_{t-1}^{\min}(p) \\ &\quad + (r/\beta) \cdot v(b_{t-1}(q)) + (2-r) \cdot v_{t-1}^{\min}(b_{t-1}(q)) - 2(r-1) \cdot v(q) \\ &\geq (r/\beta) \cdot v(q) + (2-r) \cdot v(q) \\ &\quad + (r/\beta) \cdot v(q) + (2-r) \cdot v(q) - 2(r-1) \cdot v(q) \\ &= (2((2r-3) + (2-r)) - 2(r-1)) \cdot v(q) = 0 = \Delta_t^{\text{opt}} \end{aligned}$$

•  $p$  is rejected

Changes: –

(Due to Lemma 10,  $p$  is also not stored in the buffer of OPT.)

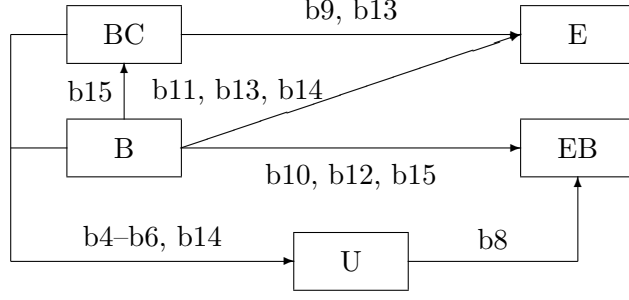


Figure 4: Possible state transitions at a send event. The labels at the edges specify the cases in which the respective state transition could occur.

Fix a send event  $\sigma_t$  in which PG sends packet  $p$  and OPT sends packet  $q$ . Note that due to Lemma 10,  $q \in P_{t-1} \cup S_{t-1}^{\text{pg}} \cup B_{t-1}^{\text{pg}}$ . Since a new dummy packet of value 0 is stored in the buffer of PG after a packet is sent, a packet  $u_B \in B_t^{\text{pg}} \setminus B_{t-1}^{\text{pg}}$  exists with  $s_t(u_B) = B$ . We can assign  $u_B$  as buddy to another packet at this event, since  $u_B \notin B_{t-1}^{\text{pg}}$ . We distinguish the following cases. If not mentioned otherwise, everything remains unchanged at event  $\sigma_t$ . We only consider the Invariant I1. For the verification of the Invariants I2–I5, see Figure 5, Figure 6, and Figure 7. In Figure 4, we depict the possible state transitions at  $\sigma_t$ .

- $q \in P_{t-1} \cup S_{t-1}^{\text{pg}}$

b1:  $q \in D_{t-1}$  and  $p \in B_{t-1} \cup BC_{t-1}$

Changes:  $D_t := D_{t-1} \cup \{p\} \cup \{p' \mid \hat{d}_{t-1}(p') = p\}$

$$\text{I1: } \Delta_t^{\text{pg}} \geq r \cdot v(p) \geq v(p) + \sum_{i=1}^{\infty} v(p)/\beta^i \geq \Delta_t^{\text{opt}}$$

b2:  $q \in D_{t-1}$  and  $p \in B_t^{\text{opt}} \setminus (B_{t-1} \cup BC_{t-1})$

Changes:  $D_t := D_{t-1} \cup \{p\}$

(Due to I2,  $p \in U_{t-1}$ .)

$$\text{I1: } \Delta_t^{\text{pg}} \geq r \cdot v(p) - ((r/\beta) \cdot v(p) + (2-r) \cdot v_{t-1}^{\min}(p)) \geq v(p) = \Delta_t^{\text{opt}}$$

b3:  $q \in D_{t-1}$  and  $p \notin B_t^{\text{opt}} \cup (B_{t-1} \cup BC_{t-1})$

Changes: –

$$\text{I1: } \Delta_t^{\text{pg}} \geq r \cdot v(p) - 2(r-1) \cdot v(p) \geq 0 = \Delta_t^{\text{opt}}$$

b4:  $q \notin D_{t-1}$  and  $p \in B_{t-1} \cup BC_{t-1}$

Changes:  $s_t(\hat{d}_{t-1}(q)) := U$ ,  $D_t := D_{t-1} \cup \{p\} \cup \{p' \mid \hat{d}_{t-1}(p') = p\} \cup \{q' \neq q \mid \hat{d}_{t-1}(q') = \hat{d}_{t-1}(q)\}$ ,  $b_t(b_{t-1}^{-1}(\hat{d}_{t-1}(q))) := u_B$

(Due to I4,  $\hat{d}_{t-1}(q) \in BC_{t-1} \cup B_{t-1}$ .)

$$\begin{aligned} \text{I1: } \Delta_t^{\text{pg}} &\geq r \cdot v(p) + (r/\beta) \cdot v(\hat{d}_{t-1}(q)) \\ &\geq v(p) + \sum_{i=1}^{\infty} v(p)/\beta^i + \sum_{i=1}^{\infty} v(\hat{d}_{t-1}(q))/\beta^i \\ &\geq v(p) + \sum_{i=1}^{\infty} v(p)/\beta^i + \sum_{q', \hat{d}_{t-1}(q') = \hat{d}_{t-1}(q)} v(q') \geq \Delta_t^{\text{opt}} \end{aligned}$$

case		packets concerned	verification
b1	I2	–	
	I3	$p, b_{t-1}^{-1}(p)$	$s_t(p) = \perp, b_{t-1}^{-1}(p) = \perp$
	I4	$p, \{p'   \hat{d}_{t-1}(p') = p\}$	$p \in D_t \Rightarrow p \notin X_t, \hat{d}_{t-1}(p') = p \Rightarrow p' \in D_t \Rightarrow p' \notin X_t$
	I5	$p, b_{t-1}(p), u_B$	$p \notin B_t^{\text{pg}}, b_t^{-1}(b_{t-1}(p)) = \perp \prec d(b_{t-1}(p)), d(u_B) = \perp$
b2 and b3	I2	–	
	I3	$p, b_{t-1}^{-1}(p)$	$s_t(p) = \perp, b_{t-1}^{-1}(p) = \perp$
b3	I4	$p, \{p'   \hat{d}_{t-1}(p') = p\}$	$p \notin B_t^{\text{opt}} \setminus D_t \supseteq X_t, \hat{d}_{t-1}(p') = p \stackrel{\text{I4}}{\Rightarrow} p' \notin X_{t-1} \supseteq X_t$
	I5	$p, b_{t-1}(p), u_B$	$p \notin B_t^{\text{pg}}, b_t^{-1}(b_{t-1}(p)) = \perp \prec d(b_{t-1}(p)), d(u_B) = \perp$
b4	I2	–	
	I3	$p, b_{t-1}^{-1}(p), b_{t-1}^{-1}(\hat{d}_{t-1}(q))$	$s_t(p) = \perp, b_{t-1}^{-1}(p) = \perp, b_t(b_{t-1}^{-1}(\hat{d}_{t-1}(q))) = u_B \in B_t$
	I4	$p, \{p'   \hat{d}_{t-1}(p') = p\}$	$p \in D_t \Rightarrow p \notin X_t, \hat{d}_{t-1}(p') = p \Rightarrow p' \in D_t \Rightarrow p' \notin X_t$
		$q$ $\{p'   \hat{d}_{t-1}(p') = \hat{d}_{t-1}(q)\}$	$q \notin B_t^{\text{opt}} \Rightarrow q \notin X_t$ $\hat{d}_{t-1}(p') = \hat{d}_{t-1}(q) \Rightarrow p' \in D_t \cup \{q\} \Rightarrow p' \notin X_t$
	I5	$p, b_{t-1}(p)$ $\hat{d}_{t-1}(q), u_B$	$p \notin B_t^{\text{pg}}, b_t^{-1}(b_{t-1}(p)) = \perp \prec d(b_{t-1}(p))$ $b_t^{-1}(\hat{d}_{t-1}(q)) = \perp \prec d(\hat{d}_{t-1}(q)), d(u_B) = \perp$
b5 and b6	I2	–	
	I3	$p, b_{t-1}^{-1}(p), b_{t-1}^{-1}(\hat{d}_{t-1}(q))$	$s_t(p) = \perp, b_{t-1}^{-1}(p) = \perp, b_t(b_{t-1}^{-1}(\hat{d}_{t-1}(q))) = u_B \in B_t$
	I4	$p, \{p'   \hat{d}_{t-1}(p') = p\}$	$p \notin B_t^{\text{opt}} \setminus D_t \supseteq X_t, \hat{d}_{t-1}(p') = p \Rightarrow p' \notin X_{t-1} \supseteq X_t$
		$q$ $\{p'   \hat{d}_{t-1}(p') = \hat{d}_{t-1}(q)\}$	$q \notin B_t^{\text{opt}} \Rightarrow q \notin X_t$ $\hat{d}_{t-1}(p') = \hat{d}_{t-1}(q) \Rightarrow p' \in D_t \cup \{q\} \Rightarrow p' \notin X_t$
I5	$p, b_{t-1}(p)$ $\hat{d}_{t-1}(q), u_B$	$p \notin B_t^{\text{pg}}, b_t^{-1}(b_{t-1}(p)) = \perp \prec d(b_{t-1}(p))$ $b_t^{-1}(\hat{d}_{t-1}(q)) = \perp \prec d(\hat{d}_{t-1}(q)), d(u_B) = \perp$	

Figure 5: Verification of the Invariants I2–I5 for the Cases b1–b6.

b5:  $q \notin D_{t-1}$  and  $p \in B_t^{\text{opt}} \setminus (B_{t-1} \cup BC_{t-1})$

Changes:  $s_t(\hat{d}_{t-1}(q)) := U, D_t := D_{t-1} \cup \{p\} \cup \{q' \neq q | \hat{d}_{t-1}(q') = \hat{d}_{t-1}(q)\},$   
 $b_t(b_{t-1}^{-1}(\hat{d}_{t-1}(q))) := u_B$

(Due to I2,  $p \in U_{t-1}$ . Due to I4,  $\hat{d}_{t-1}(q) \in BC_{t-1} \cup B_{t-1}$ .)

$$\begin{aligned}
\text{II: } \Delta_t^{\text{pg}} &\geq r \cdot v(p) - ((r/\beta) \cdot v(p) + (2-r) \cdot v_{t-1}^{\min}(p)) + (r/\beta) \cdot v(\hat{d}_{t-1}(q)) \\
&\geq v(p) + (r/\beta) \cdot v(\hat{d}_{t-1}(q)) \\
&\geq v(p) + \sum_{i=1}^{\infty} v(\hat{d}_{t-1}(q))/\beta^i \geq \Delta_t^{\text{opt}}
\end{aligned}$$

b6:  $q \notin D_{t-1}$  and  $p \notin B_t^{\text{opt}} \cup (B_{t-1} \cup BC_{t-1})$

Changes:  $s_t(\hat{d}_{t-1}(q)) := U, D_t := D_{t-1} \cup \{q' \neq q | \hat{d}_{t-1}(q') = \hat{d}_{t-1}(q)\},$   
 $b_t(b_{t-1}^{-1}(\hat{d}_{t-1}(q))) := u_B$

(Due to I4,  $\hat{d}_{t-1}(q) \in \text{BC}_{t-1} \cup \text{B}_{t-1}$ .)

$$\begin{aligned} \text{I1: } \Delta_t^{\text{pg}} &\geq r \cdot v(p) - 2(r-1) \cdot v(p) + (r/\beta) \cdot v(\hat{d}_{t-1}(q)) \\ &\geq (r/\beta) \cdot v(\hat{d}_{t-1}(q)) \\ &\geq \sum_{i=1}^{\infty} v(\hat{d}_{t-1}(q))/\beta^i \geq \Delta_t^{\text{opt}} \end{aligned}$$

- b7:  $q = p$

Changes: –

(Due to I2,  $p \in \text{U}_{t-1} \cup \text{BC}_{t-1} \cup \text{B}_{t-1}$ .)

$$\text{I1: } \Delta_t^{\text{pg}} \geq r \cdot v(p) - ((r/\beta) \cdot v(p) + (2-r) \cdot v_{t-1}^{\min}(p)) \geq v(p) = \Delta_t^{\text{opt}}$$

- $q \in B_{t-1}^{\text{pg}} \setminus \{p\}$

- b8:  $q \in \text{U}_{t-1}$

Changes:  $b_t(q) := u_B$ ,  $s_t(q) := \text{EB}$

$$\begin{aligned} \text{I1: } \Delta_t^{\text{pg}} &\geq r \cdot v(p) - c_{t-1}(p) \\ &\quad + 2(r-1) \cdot v(q) - ((r/\beta) \cdot v(q) + (2-r) \cdot v_{t-1}^{\min}(q)) \\ &\geq r \cdot v(p) - 2(r-1) \cdot v(p) \\ &\quad + 2(r-1) \cdot v(q) - ((r/\beta) \cdot v(q) + (2-r) \cdot v(p)) \\ &= v(q) = \Delta_t^{\text{opt}} \end{aligned}$$

- b9:  $q \in \text{BC}_{t-1}$

Changes:  $b_t(b_{t-1}^{-1}(q)) := u_B$ ,  $s_t(q) := \text{E}$

$$\text{I1: } \Delta_t^{\text{pg}} \geq r \cdot v(p) - c_{t-1}(p) + (r-1) \cdot v(q) - (r-2) \cdot v(q) \geq v(q) = \Delta_t^{\text{opt}}$$

- b10:  $q \in \text{B}_{t-1}$  and  $v(p) < v(q)/\beta$

Changes:  $b_t(q) := u_B$ ,  $s_t(q) := \text{EB}$

(Due to I5,  $b_{t-1}^{-1}(q) \prec d(q)$ , i.e.,  $b_{t-1}^{-1}(q) \notin B_{t-1}^{\text{pg}}$ , since  $v(p) < v(q)/\beta$ .)

$$\text{I1: } \Delta_t^{\text{pg}} \geq r \cdot v(p) - c_{t-1}(p) + 2(r-1) \cdot v(q) > v(q) = \Delta_t^{\text{opt}}$$

- b11:  $q \in \text{B}_{t-1}$  and  $v(p) \geq v(q)/\beta$  and  $p \notin \text{EB}_{t-1}$

Changes:  $b_t(b_{t-1}^{-1}(q)) := u_B$ ,  $s_t(q) := \text{E}$

$$\begin{aligned} \text{I1: } \Delta_t^{\text{pg}} &= r \cdot v(p) - c_{t-1}(p) + (r-1) \cdot v(q) \\ &\geq v(p) + (r-1) \cdot v(q) \\ &\geq (1/\beta + (r-1)) \cdot v(q) = v(q) = \Delta_t^{\text{opt}} \end{aligned}$$

- b12:  $q \in \text{B}_{t-1}$  and  $v(p) \geq v(q)/\beta$  and  $p \in \text{EB}_{t-1}$  and  $b_{t-1}^{-1}(q) = \perp$

Changes:  $b_t(q) := u_B$ ,  $s_t(q) := \text{EB}$

$$\text{I1: } \Delta_t^{\text{pg}} \geq r \cdot v(p) - c_{t-1}(p) + 2(r-1) \cdot v(q) \geq v(q) = \Delta_t^{\text{opt}}$$

case		packets concerned	verification
b7	I2	–	
	I3	$p, b_{t-1}^{-1}(p)$	$s_t(p) = \perp, b_{t-1}^{-1}(p) = \perp$
	I4	$p$ $\{p'   \hat{d}_{t-1}(p') = p\}$	$p \notin B_t^{\text{opt}} \Rightarrow p \notin X_t,$ $\hat{d}_{t-1}(p') = p \Rightarrow p' \prec p \Rightarrow p' \notin B_t^{\text{opt}} \supseteq X_t$
	I5	$p, b_{t-1}(p), u_B$	$p \notin B_t^{\text{pg}}, b_{t-1}^{-1}(b_{t-1}(p)) = \perp \prec d(b_{t-1}(p)), d(u_B) = \perp$
b8	I2	$q$	$q \notin B_t^{\text{opt}}$
	I3	$p, b_{t-1}^{-1}(p), q$	$s_t(p) = \perp, b_{t-1}^{-1}(p) = \perp, b_t(q) = u_B \in B_t$
	I4	$p$ $\{p'   \hat{d}_{t-1}(p') = p\}$	$p \notin B_t^{\text{opt}} \Rightarrow p \notin X_t,$ $\hat{d}_{t-1}(p') = p \Rightarrow p' \prec p \prec q \Rightarrow p' \notin B_t^{\text{opt}} \supseteq X_t$
	I5	$p, b_{t-1}(p), u_B$	$p \notin B_t^{\text{pg}}, b_{t-1}^{-1}(b_{t-1}(p)) = \perp \prec d(b_{t-1}(p)), d(u_B) = \perp$
b9	I2	$q$	$q \notin B_t^{\text{opt}}$
	I3	$p, b_{t-1}^{-1}(p), b_{t-1}^{-1}(q)$	$s_t(p) = \perp, b_{t-1}^{-1}(p) = \perp, b_t(b_{t-1}^{-1}(q)) = u_B \in B_t$
	I4	$p$ $\{p'   \hat{d}_{t-1}(p') = p\}$ $\{p'   \hat{d}_{t-1}(p') = q\}$	$p \notin B_t^{\text{opt}} \Rightarrow p \notin X_t,$ $\hat{d}_{t-1}(p') = p \Rightarrow p' \prec p \prec q \Rightarrow p' \notin B_t^{\text{opt}} \supseteq X_t$ $\hat{d}_{t-1}(p') = q \Rightarrow p' \prec q \Rightarrow p' \notin B_t^{\text{opt}} \supseteq X_t$
	I5	$p, b_{t-1}(p)$ $q, u_B$	$p \notin B_t^{\text{pg}}, b_{t-1}^{-1}(b_{t-1}(p)) = \perp \prec d(b_{t-1}(p))$ $b_{t-1}^{-1}(q) = \perp, d(u_B) = \perp$
b10	I2	$q$	$q \notin B_t^{\text{opt}}$
	I3	$p, b_{t-1}^{-1}(p), q, b_{t-1}^{-1}(q)$	$s_t(p) = \perp, b_{t-1}^{-1}(p) = \perp, b_t(q) = u_B \in B_t, b_{t-1}^{-1}(q) \stackrel{I5}{=} \perp$
	I4	$p$ $\{p'   \hat{d}_{t-1}(p') = p\}$ $\{p'   \hat{d}_{t-1}(p') = q\}$	$p \prec q \Rightarrow p \notin B_t^{\text{opt}} \supseteq X_t$ $\hat{d}_{t-1}(p') = p \Rightarrow p' \prec p \prec q \Rightarrow p' \notin B_t^{\text{opt}} \supseteq X_t$ $\hat{d}_{t-1}(p') = q \Rightarrow p' \prec q \Rightarrow p' \notin B_t^{\text{opt}} \supseteq X_t$
	I5	$p, b_{t-1}(p), u_B$	$p \notin B_t^{\text{pg}}, b_{t-1}^{-1}(b_{t-1}(p)) = \perp \prec d(b_{t-1}(p)), d(u_B) = \perp$
b11	I2	$q$	$q \notin B_t^{\text{opt}}$
	I3	$p, b_{t-1}^{-1}(p), b_{t-1}^{-1}(q)$	$s_t(p) = \perp, b_{t-1}^{-1}(p) = \perp, b_t(b_{t-1}^{-1}(q)) = u_B \in B_t$
	I4	$p$ $\{p'   \hat{d}_{t-1}(p') = p\}$ $\{p'   \hat{d}_{t-1}(p') = q\}$	$p \prec q \Rightarrow p \notin B_t^{\text{opt}} \supseteq X_t$ $\hat{d}_{t-1}(p') = p \Rightarrow p' \prec p \prec q \Rightarrow p' \notin B_t^{\text{opt}} \supseteq X_t$ $\hat{d}_{t-1}(p') = q \Rightarrow p' \prec q \Rightarrow p' \notin B_t^{\text{opt}} \supseteq X_t$
	I5	$p, b_{t-1}(p), u_B$	$p \notin B_t^{\text{pg}}, b_{t-1}^{-1}(b_{t-1}(p)) = \perp \prec d(b_{t-1}(p)), d(u_B) = \perp$

Figure 6: Verification of the Invariants I2–I5 for the Cases b7–b11.

b13:  $q \in B_{t-1}$  and  $v(p) \geq v(q)/\beta$  and  $p \in EB_{t-1}$  and  $b_{t-1}(p) \preceq b_{t-1}^{-1}(q)$

Changes:  $s_t(b_{t-1}(p)) := E, b_t(b_{t-1}^{-1}(q)) := u_B, s_t(q) := E$

(Due to I5,  $b_{t-1}^{-1}(q) \prec d(q)$ , i.e.,  $v(b_{t-1}(p)) \geq v(q)/\beta$ . Due to I3,  $b_{t-1}(p) \in B_{t-1} \cup$

case		packets concerned	verification
b12	I2	$q$	$q \notin B_t^{\text{opt}}$
	I3	$p, b_{t-1}^{-1}(p), q, b_{t-1}^{-1}(q)$	$s_t(p) = \perp, b_{t-1}^{-1}(p) = \perp, b_t(q) = u_B \in B_t, b_{t-1}^{-1}(q) = \perp$
	I4	$p$ $\{p'   \hat{d}_{t-1}(p') = p\}$ $\{p'   \hat{d}_{t-1}(p') = q\}$	$p \prec q \Rightarrow p \notin B_t^{\text{opt}} \supseteq X_t$ $\hat{d}_{t-1}(p') = p \Rightarrow p' \prec p \prec q \Rightarrow p' \notin B_t^{\text{opt}} \supseteq X_t$ $\hat{d}_{t-1}(p') = q \Rightarrow p' \prec q \Rightarrow p' \notin B_t^{\text{opt}} \supseteq X_t$
	I5	$p, b_{t-1}(p), u_B$	$p \notin B_t^{\text{pg}}, b_t^{-1}(b_{t-1}(p)) = \perp \prec d(b_{t-1}(p)), d(u_B) = \perp$
b13	I2	$q, b_{t-1}(p)$	$q \notin B_t^{\text{opt}}, b_{t-1}(p) \preceq b_{t-1}^{-1}(q) \prec q \Rightarrow b_{t-1}(p) \notin B_t^{\text{opt}}$
	I3	$p, b_{t-1}^{-1}(p), b_{t-1}^{-1}(q)$	$s_t(p) = \perp, b_{t-1}^{-1}(p) = \perp, b_t(b_{t-1}^{-1}(q)) = u_B \in B_t$
	I4	$p$ $\{p'   \hat{d}_{t-1}(p') = p\}$ $\{p'   \hat{d}_{t-1}(p') = q\}$	$p \prec q \Rightarrow p \notin B_t^{\text{opt}} \supseteq X_t$ $\hat{d}_{t-1}(p') = p \Rightarrow p' \prec p \prec q \Rightarrow p' \notin B_t^{\text{opt}} \supseteq X_t$ $\hat{d}_{t-1}(p') = q \Rightarrow p' \prec q \Rightarrow p' \notin B_t^{\text{opt}} \supseteq X_t$
	I5	$p, b_{t-1}(p), u_B$	$p \notin B_t^{\text{pg}}, b_t^{-1}(b_{t-1}(p)) = \perp \prec d(b_{t-1}(p)), d(u_B) = \perp$
b14	I2	$q$	$q \notin B_t^{\text{opt}}$
	I3	$p, b_{t-1}^{-1}(p), b_{t-1}^{-1}(q)$	$s_t(p) = \perp, b_{t-1}^{-1}(p) = \perp, b_t(b_{t-1}^{-1}(q)) = u_B \in B_t$
	I4	$p$ $\{p'   \hat{d}_{t-1}(p') = p\}$ $\{p'   \hat{d}_{t-1}(p') = q\}$ $\{p'   \hat{d}_{t-1}(p') = b_{t-1}(p)\}$	$p \prec q \Rightarrow p \notin B_t^{\text{opt}} \supseteq X_t$ $\hat{d}_{t-1}(p') = p \Rightarrow p' \prec p \prec q \Rightarrow p' \notin B_t^{\text{opt}} \supseteq X_t$ $\hat{d}_{t-1}(p') = q \Rightarrow p' \prec q \Rightarrow p' \notin B_t^{\text{opt}} \supseteq X_t$ $\hat{d}_{t-1}(p') = b_{t-1}(p) \Rightarrow p' \in D_t \Rightarrow p' \notin X_t$
	I5	$p, b_{t-1}(p), u_B$	$p \notin B_t^{\text{pg}}, b_t^{-1}(b_{t-1}(p)) = \perp \prec d(b_{t-1}(p)), d(u_B) = \perp$
b15	I2	$q$	$q \notin B_t^{\text{opt}}$
	I3	$p, b_{t-1}^{-1}(p)$ $q, b_{t-1}^{-1}(q)$	$s_t(p) = \perp, b_{t-1}^{-1}(p) = \perp$ $b_t(q) = u_B \in B_t, b_t(b_{t-1}^{-1}(q)) = b_{t-1}(p) \in BC_t$
	I4	$p$ $\{p'   \hat{d}_{t-1}(p') = p\}$ $\{p'   \hat{d}_{t-1}(p') = q\}$	$p \prec q \Rightarrow p \notin B_t^{\text{opt}} \supseteq X_t$ $\hat{d}_{t-1}(p') = p \Rightarrow p' \prec p \prec q \Rightarrow p' \notin B_t^{\text{opt}} \supseteq X_t$ $\hat{d}_{t-1}(p') = q \Rightarrow p' \prec q \Rightarrow p' \notin B_t^{\text{opt}} \supseteq X_t$
	I5	$p, b_{t-1}(p), u_B$	$p \notin B_t^{\text{pg}}, b_{t-1}(p) \in BC_t, d(u_B) = \perp$

Figure 7: Verification of the Invariants I2–I5 for the Cases b12–b15.

$BC_{t-1}$ .)

$$\begin{aligned}
\text{I1: } \Delta_t^{\text{pg}} &\geq r \cdot v(p) - 2(r-1) \cdot v(p) + (r-1) \cdot v(q) + (r-1) \cdot v(b_{t-1}(p)) \\
&\geq (2-r) \cdot v(q)/\beta + (r-1) \cdot v(q) + (r-1) \cdot v(q)/\beta \\
&= (1/\beta + (r-1)) \cdot v(q) = v(q) = \Delta_t^{\text{opt}}
\end{aligned}$$

b14:  $q \in B_{t-1}$  and  $v(p) \geq v(q)/\beta$  and  $p \in EB_{t-1}$  and  $b_{t-1}^{-1}(q) \prec b_{t-1}(p)$  and  $v(b_{t-1}(p)) > 2v(q)$

Changes:  $s_t(b_{t-1}(p)) := U$ ,  $D_t := D_{t-1} \cup \{p' | \hat{d}_{t-1}(p') = b_{t-1}(p)\}$ ,  
 $b_t(b_{t-1}^{-1}(q)) := u_B$ ,  $s_t(q) = E$

(Due to I3,  $b_{t-1}(p) \in \mathbf{B}_{t-1} \cup \mathbf{BC}_{t-1}$ .)

$$\begin{aligned}
\text{I1: } \Delta_t^{\text{pg}} &\geq r \cdot v(p) - c_{t-1}(p) + (r/\beta) \cdot v(b_{t-1}(p)) - c_{t-1}(b_{t-1}(p)) + (r-1) \cdot v(q) \\
&\geq (2-r) \cdot v(p) + (r/\beta) \cdot v(b_{t-1}(p)) + (r-1) \cdot v(q) \\
&= (2-r) \cdot v(p) + (3r-5)/2 \cdot v(b_{t-1}(p)) \\
&\quad + (r-1) \cdot v(q) + v(b_{t-1}(p))/(\beta-1) \\
&\geq (2-r) \cdot v(q)/\beta + (3r-5) \cdot v(q) + (r-1) \cdot v(q) \\
&\quad + v(b_{t-1}(p))/(\beta-1) \\
&= v(q) + \sum_{i=1}^{\infty} v(b_{t-1}(p))/\beta^i \geq \Delta_t^{\text{opt}}
\end{aligned}$$

b15:  $q \in \mathbf{B}_{t-1}$  and  $v(p) \geq v(q)/\beta$  and  $p \in \mathbf{EB}_{t-1}$  and  $b_{t-1}^{-1}(q) \prec b_{t-1}(p)$  and  $v(b_{t-1}(p)) \leq 2v(q)$

Changes:  $s_t(b_{t-1}(p)) := \mathbf{BC}$ ,  $b_t(b_{t-1}^{-1}(q)) := b_{t-1}(p)$ ,  $b_t(q) := u_{\mathbf{B}}$ ,  $s_t(q) := \mathbf{EB}$   
(Due to I3,  $b_{t-1}(p) \in \mathbf{B}_{t-1} \cup \mathbf{BC}_{t-1}$ .)

$$\begin{aligned}
\text{I1: } \Delta_t^{\text{pg}} &= r \cdot v(p) - c_{t-1}(p) \\
&\quad + (r-2) \cdot v(b_{t-1}(p)) - c_{t-1}(b_{t-1}(p)) + 2(r-1) \cdot v(q) \\
&\geq (2-r) \cdot v(p) + (r-2) \cdot v(b_{t-1}(p)) + 2(r-1) \cdot v(q) \\
&\geq (2-r) \cdot v(q)/\beta + (r-2) \cdot 2v(q) + 2(r-1) \cdot v(q) = v(q) = \Delta_t^{\text{opt}}
\end{aligned}$$

This concludes the proof of the theorem. □

## References

- [1] W. Aiello, Y. Mansour, S. Rajagopalan, and A. Rosen. Competitive queue policies for differentiated services. *Journal of Algorithms*, 55(2):113–141, 2005.
- [2] N. Andelman. Randomized queue management for DiffServ. In *Proceedings of the 17th ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 1–10, 2005.
- [3] 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.
- [4] Y. Azar and Y. Richter. Management of multi-queue switches in QoS networks. *Algorithmica*, 43(1–2):81–96, 2005.
- [5] N. Bansal, L. Fleischer, T. Kimbrel, M. Mahdian, B. Schieber, and M. Sviridenko. Further improvements in competitive guarantees for QoS buffering. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP)*, pages 196–207, 2004.
- [6] W. Jawor. Three dozen papers on online algorithms. *SIGACT News*, 36(1):71–85, 2005.
- [7] 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.



- [8] A. Kesselman and Y. Mansour. Loss-bounded analysis for differentiated services. *Journal of Algorithms*, 46(1):79–95, 2003.
- [9] A. Kesselman, Y. Mansour, and R. van Stee. Improved competitive guarantees for QoS buffering. *Algorithmica*, 43(1–2):63–80, 2005.
- [10] Z. Lotker and B. Patt-Shamir. Nearly optimal FIFO buffer management for two packet classes. *Computer Networks*, 42(4):481–492, 2003.
- [11] Y. Mansour, B. Patt-Shamir, and O. Lapid. Optimal smoothing schedules for real-time streams. *Distributed Computing*, 17(1):77–89, 2004.