Competitive Management of Non-Preemptive Queues with Multiple Values

Nir Andelman
andelman@cs.tau.ac.il

Yishay Mansour
mansour@cs.tau.ac.il

School of Computer Science
Tel-Aviv University

April 7, 2003

Abstract

We consider the online problem of active queue management. In our model, the input is a sequence of packets with values $v \in [1, \alpha]$ that arrive to a queue that can hold up to B packets. Specifically, we consider a FIFO non-preemptive queue, where any packet that is accepted into the queue must be sent, and packets are sent by the order of arrival. The benefit of a scheduling policy, on a given input, is the sum of values of the scheduled packets. Our aim is to find an online policy that maximizes its benefit compared to the optimal offline solution.

Previous work proved that no constant competitive ratio exists for this problem, showing a lower bound of $\ln(\alpha)+1$ for any online policy. An upper bound of $e\lceil\ln(\alpha)\rceil$ was proved for a few online policies. In this paper we suggest and analyze a RED-like online policy with a competitive ratio that matches the lower bound up to an additive constant proving an upper bound of $\ln(\alpha) + 2 + O(\ln^2(\alpha)/B)$. For large values of α, we prove that no policy whose decisions are based only on the number of packets in the queue and the value of the arriving packet, has a competitive ratio lower than $\ln(\alpha) + 2 - \epsilon$, for any constant $\epsilon > 0$.

Submitted to the regular track.
Nir Andelman is a full time student at Tel-Aviv University.

1 Introduction

One of the main bottlenecks in the traffic flow inside communication networks is the management of queues in the connection points, such as switches and routers. Traffic may arrive simultaneously from various sources to the connection point. If incoming traffic from several sources is directed towards the same destination, it may be impossible to immediately direct all the traffic towards the outgoing link, since the bandwidth of the outgoing link is limited, and therefore packet loss is unavoidable.

The traffic in communication networks tends to arrive in bursts, which is the motivation of buffering the packets in queues, located either at the incoming links, or the outgoing links, or both. The packets arriving in a burst are stored in queues, and are later sent in a speed determined by the bandwidth of both outgoing links and the backbone of the connection device. If the traffic is not too heavy, the queues will drain before more bursts arrive, avoiding packet loss. This best effort approach relies on statistical characteristics of communication traffic, assuming that the capacity of the links is not always fully used.

The best effort approach is not sufficient for service providers who wish to guarantee Quality of Service (QoS) to their users. QoS can be considered as a contract between the communication service provider and the network user. As long as the user's traffic does not exceed certain quotas agreed in the contract, the service provider guarantees specific characteristics of the network service, such as minimal bandwidth, maximal delay time, jitter, packet loss, etc. The demand for QoS results in a couple of problems that the service provider must solve. One problem is to determine what the service provider can guarantee to a user. The other is what to do with the user's traffic when it does not fulfill the terms of the contract.

One solution to the problem of QoS is *Premium Service* [14], in which the traffic is shaped upon the entry to the network. The service provider guarantees that customers who payed for QoS will have a certain portion of the bandwidth allocated for their use. Packets that are not part of the Premium Service are treated by best effort mechanism, and might be dropped due to congestion. From the point of view of the user, the shared network is almost indistinguishable from a private link. From the point of view of the service provider, this is a solution with high utilization that allows dedicating private links, since the dedicated bandwidth may be used for general traffic when the traffic from paying customers is idle.

Assured Service [6] is a different approach that relies on *statistical multiplexing*, which takes into account the fact that usually worst case scenarios, where all users use the same network resources at once, do not occur. This assumption allows the service provider to apply an 'overbooking' policy, which relaxes the constraints on the users' usage of the communication network, risking a potential buffer overflow, when congestion occurs. Assured Service supplies a relative guarantee to the user, which is a vague promise that certain traffic will be treated with higher priority at times of network congestion, achieving better throughput relative to the rest of the traffic. The concept that some packets are worth more than others is the basis of *Differentiated Services*. Packets that have QoS guarantee are distinguished from normal packets. Furthermore, the service provider may decide to treat differently packets of different paying customers (in a 'pay more - get more' fashion).

We use the following abstraction to analyze the performance of a single queue: The system is a queue that can hold up to B packets. The input packets that arrive to the queue have values of $v \in [1, \alpha]$. A packet's value represents the priority given to the packet by the service provider. The queue management is an online policy that decides for each packet, when it arrives, whether to place it in the queue or to reject it. Preemption is not allowed, meaning that any accepted packet is eventually transmitted and cannot be rejected later. At arbitrary times a packet is sent from the queue (if not empty), at FIFO order, i.e. at the order of arrival. We assume that packet values are accumulative, so the benefit on the policy over a given input is the sum of values of accepted packets. We use competitive analysis [5] to analyze online policies, meaning that we derive bounds on the ratio between the benefit of the optimal offline scheduling, to the benefit of the online policy, over the worst case input.

Online packet scheduling with a non-preemptive queue was first analyzed in [1] for two packet values only, 1 and $\alpha > 1$. A general lower bound of $2 - 1/\alpha$ was derived, and several online policies were analyzed, deriving upper and lower bounds for each policy. An optimal online policy for the two value case with a competitive ratio of $2 - 1/\alpha$ was analyzed in [2]. The continuous case, were packet values may vary along the range $[1, \alpha]$ was also considered in [2]. A general lower bound of $\ln(\alpha) + 1$ was derived for any online policy, and an upper bound of $e\lceil \ln(\alpha) \rceil$ was proved for two policies.

Our results. In this paper we analyze a RED-like [7] online policy for continuous values, with a provable upper bound of $\ln(\alpha) + 2 + O(\ln^2(\alpha)/B)$, which nearly matches the lower bound. Similarly to Random Early Detection, we suggest a policy that detects the possibility of congestion before the queue is full, and drops packets in advance. While RED uses probabilities to decide whether to drop a packet, we use the packet's value for this decision. Our analysis measures the exact influence of the queue size on the competitive ratio. For large values of α, we prove that $\ln(\alpha) + 2$ is a lower bound for a large family of policies, which in addition to the value of the arriving packet, consider only the number of packets in the queue. For a low value of α, we analyze an alternative policy, which is an extension of the optimal online policy for the case of two packet values, and prove that its competitive ratio in this case is strictly lower than $\ln(\alpha) + 2$, for any low value of α.

Related work. A preemptive FIFO queue allows preempting packets, which were already accepted, from the queue. Online algorithms in the preemptive FIFO model were analyzed in [2, 9, 10, 13, 15]. Since the optimal offline is non-preemptive, preemptive policies have better competitive ratios than non-preemptive ones. For inputs with two values, the currently known best lower and upper bounds are approximately 1.28 [15] and 1.30 [13], respectively. For unlimited values, the lower and upper bounds are $\sqrt{2}$ and 2, respectively [2, 9]. Recently, Kesselman et al. [12] have designed a policy that beats the 2-competitive ratio of the greedy policy, and is $2 - \epsilon$, for some constant $\epsilon > 0$.

Generalized versions of managing FIFO queues include managing a bulk of FIFO queues [3], where the policy has also to decide from which queue to send a packet, and shared memory switches [8, 11], where the queues are not fixed in size, but their total size is fixed. Another extension is the delay bounded queue [2, 9], where packets may be sent in arbitrary order, but each packet arrives with a expiration time. Packets that are not sent by this deadline expire and are lost.

Paper organization. Section 2 defines the non-preemptive queue model. Section 3 defines the Selective Barrier Policy, which is analyzed in Section 4. We improve the policy for low values of α in Section 5. In Section 6 we prove a lower bound of $\ln(\alpha) + 2$ for a family of policies. Our concluding remarks appear in Section 7.

2 Model and Notations.

2.1 Packets

A packet is a basic unit that describes the input of the system. Each packet is identified by its value v, which is the benefit that the system gains by sending the packet. Packet values may vary from 1 to α, where $\alpha \geq 1$ is an arbitrary value, known in advance. For simplicity, v refers both to the packet itself and to its value.

2.2 Input Streams

The input to the system is a finite stream of events, occurring at discrete times $t = 1, 2, ..., n$. There are two types of events, `arrive(`v`)`, which is the arrival of a packet with value v to the system and `send()`, which allows the system to send one of the packets that arrived earlier.

2.3 FIFO Queues

The system uses a queue for buffering the arriving packets. B denotes the maximal number of packets that can be held in the queue. In our analysis we assume that $B \geq \ln(\alpha) + 2$. This is a reasonable assumption since usually buffers are fairly large. In addition, if $B = o(\ln(\alpha))$, no online policy has a logarithmic order competitive ratio. Each `arrive(`v`)` event invokes a response from the system, which is either to accept the packet and place it inside the queue, or the reject it. The decision whether to accept or to reject a packet depends on the specific policy used to manage the queue. No more than B packets can be stored in the queue at the same time, and preemption is not allowed. During a `send()` event the system extracts the most recent packet in the queue and sends it, if the queue is not empty.

2.4 Online Policies

A policy is an algorithm that given an input stream decides which packets to accept and which packets to reject. Given an input stream and a policy A, let $h_A(t)$ denote the number of packets in the queue at time t. Let $V_A(t)$ denote the total value of packets accepted until time t. Notice that since we restrict ourselves to non-preemptive queues, each accepted packet is eventually sent, so $V_A(t)$ can be regarded as the total benefit assured by policy A until time t over the input stream.

A policy is considered an online policy if the decision to accept or reject a packet determines only on the previous and current events. We use competitive ratio [5] to analyze the performance of online policies. An online policy is *c-competitive* if for any input sequence $c \cdot V_{ON} \geq V_{OPT}$, where V_{ON} is the benefit of the online policy and V_{OPT} is the benefit of an optimal offline policy.

3 Smooth Selective Barrier Policy

We define a policy that is analogous to the mechanism of Random Early Detection (RED) [7] gateways for congestion avoidance in packet switched networks. The RED gateway measures the number of packets in the FIFO queue, and marks or drops packets with a certain probability, if this number exceeds a certain threshold. The probability of marking/droping a packet depends on the number of packets in the queue. RED gateways are designed to accompany a congestion control protocol such as TCP. By marking or dropping packets, the gateway hails the connections to reduce their windows, and thus keeps the number of packets in the queue low.

The `Selective Barrier Policy` is an online policy intuitively designed as a derandomized variation of RED. Like RED, the policy becomes more restrictive as the queue size increases. Unlike RED (which is ignorant to packet values), the decision to drop a packet is deterministically based on the packet's value, and the number of packets in the queue.

Formally, we define $\mathcal{F}(v) : [1, \alpha] \to [1, B]$ to be a monotone function that bounds the maximal number of packets that can be in the queue if a packet of value v was just accepted. If an `arrive(`v`)` event occurs at time t then the packet is accepted if $\mathcal{F}(v) \geq h_{ON}(t) + 1$ and is rejected otherwise. Intuitively, $\mathcal{F}^{-1}(h)$ is the lowest value of a packet that will be accepted by the policy when the queue already holds $h - 1$ packets.

The competitive ratio of the Selective Barrier Policy depends on the exact mapping of $\mathcal{F}(\cdot)$. In [2] an upper bound of $e \cdot \lceil \ln(\alpha) \rceil$ was derived, where $\mathcal{F}(v) = \frac{i}{\lceil \ln(\alpha) \rceil} B$ for each $v \in [e^{i-1}, e^i)$. This mapping divides the queue equally into $\lceil \ln(\alpha) \rceil$ sub-queues, and increments the threshold for accepting packets in exponential steps. Due to the division to sub-queues, the policy might accept only a $1/\lceil \ln(\alpha) \rceil$ fraction of the packets in a certain range. The e factor in the competitive ratio is mainly due to the fact that packet values in $[e^{i-1}, e^i)$ may differ by a factor of e in their value, but the policy treats them identically.

The main contribution of this work is defining a finer $\mathcal{F}(\cdot)$, and even more importantly, being able to prove a tight bound on its competitive ratio. Specifically, we consider the mapping $\mathcal{F}(v) = \mathcal{F}(1) + \frac{\ln(v)}{\ln(\alpha)} S$, where $\mathcal{F}(1) = \lceil \frac{B}{\ln(\alpha)+2} \rceil$ and $S = B - \mathcal{F}(1)$. We name this variant of the Selective Barrier Policy as the `Smooth Selective Barrier Policy`. Intuitively, the first $\mathcal{F}(1)$ slots in the queue are unrestricted, meaning that the policy accepts any packet into these slots, regardless of the packet value. In the remaining slots we accept packets if their value is larger than a threshold that increases in an exponential rate, yet it is smooth, as can be observed by the following lemma:

Lemma 3.1 *For any $h \in [\mathcal{F}(1), B - 1]$, we have $\mathcal{F}^{-1}(h + 1) = \mathcal{F}^{-1}(h)\alpha^{1/S}$*

Proof: Let v be the value such that $\mathcal{F}(v) = h$. Since the mapping of $\mathcal{F}(\cdot)$ is monotone increasing, then v is unique and $\mathcal{F}^{-1}(\mathcal{F}(v)) = v$. By the definition of $\mathcal{F}(\cdot)$, we have the following:

$$\begin{aligned}
\mathcal{F}(v) + 1 &= \mathcal{F}(1) + \frac{\ln(v)}{\ln(\alpha)} S + 1 = \mathcal{F}(1) + \frac{\ln(v) + \frac{\ln(\alpha)}{S}}{\ln(\alpha)} S \\
&= \mathcal{F}(1) + \frac{\ln(v) + \ln(\alpha^{1/S})}{\ln(\alpha)} S = \mathcal{F}(1) + \frac{\ln(v\alpha^{1/S})}{\ln(\alpha)} S = \mathcal{F}(v\alpha^{1/S})
\end{aligned}$$

Therefore, we have

$$\mathcal{F}^{-1}(h+1) = \mathcal{F}^{-1}(\mathcal{F}(v)+1) = \mathcal{F}^{-1}(\mathcal{F}(v\alpha^{1/S})) = v\alpha^{1/S} = \mathcal{F}^{-1}(h)\alpha^{1/S}$$

which completes the proof. □

4 Analysis

We prove an upper bound for the Smooth Selective Barrier Policy with the function $\mathcal{F}(\cdot)$ defined as in Section 3 by comparing it to an optimal offline policy over an arbitrary input. Our proof is constructed from two stages: First, we simplify the input by modifying the values of the packets in a conservative way such that the competitive ratio cannot decrease due to the modification. In the second stage we define a potential function that bounds the possible additional benefit that the offline can gain, without any gain to the benefit of the online. We use this potential function to prove inductively that given a modified input, the ratio between the benefit of the online and the benefit of the offline plus its potential is always bounded.

In our proof we use ON to denote the Smooth Selective Barrier Policy while OPT denotes the optimal offline. We perform a simple relaxation on the input stream, defined as follows: If the online accepts a packet v at time t, then we may reduce the value of v to $v' = \mathcal{F}^{-1}(h_{ON}(t))$, i.e. the value of v' is the lowest value that the online policy still accepts. We derive the following claims for relaxed inputs:

Lemma 4.1 *The Competitive ratio of a relaxed input is at least the same as the competitive ratio of the original input.*

Proof: We analyze the competitive ratio after each change of a packet value v to v', and notice that it does not reduce the competitive ratio. After changing the value of a single packet, the decisions of the Smooth Selective Barrier Policy remain unchanged, since the policy does not consider the value of previously accepted packets, only their amount, and therefore loses a benefit of exactly $v - v'$. The optimal offline is unaffected if it rejected v in the original schedule, and loses at most benefit $v - v'$ if it did accept the packet. Since the competitive ratio is at least 1, it cannot decrease over the relaxed input. □

In the following lemma we prove an upper bound on the value of the packets that the offline accepts, which depends on the state of the online queue.

Lemma 4.2 *In a relaxed input, if the offline accepts v at time t, then $\mathcal{F}^{-1}(h_{ON}(t)) \geq v$*

Proof: If the online policy also accepts the packet v, then by the relaxation we have $\mathcal{F}^{-1}(h_{ON}(t)) = v$. Otherwise, the packet was rejected, which means that $\mathcal{F}^{-1}(h_{ON}(t)) > v$. □

We define a potential function $\phi(t)$. Intuitively, $\phi(t)$ measures the extra benefit that the offline can gain without changing the online. The potential $\phi(t)$ is defined as follows:

$$\phi(t) = \begin{cases} \max\left(0, \frac{h_{ON}(t)}{\mathcal{F}(1)}B - h_{OPT}(t)\right) & : \quad h_{ON}(t) < \mathcal{F}(1) \\ (B - h_{OPT}(t))\mathcal{F}^{-1}(h_{ON}(t)) + \sum_{i=\mathcal{F}(1)}^{h_{ON}(t)-1} \mathcal{F}^{-1}(i) & : \quad h_{ON}(t) \geq \mathcal{F}(1) \end{cases}$$

Notice that both functions used in the definition of $\phi(t)$ are equal at $h_{ON}(t) = \mathcal{F}(1)$. When $h_{ON}(t) \geq \mathcal{F}(1)$, $\phi(t)$ measures the maximal possible gain in the benefit of the offline, without increasing the benefit of the online. The offline can fill its queue with $B - h_{OPT}(t)$ packets, which will be rejected by the online if their values are less than $\mathcal{F}^{-1}(h_{ON}(t))$. Then, the offline can send packets in order to accept more packets, but the online sends packets too, so the new packets must have decreasing values, otherwise the online will accept them. When $h_{ON}(t) < \mathcal{F}(1)$, $\phi(t)$ increments in linear steps, measuring how near the online queue is to $\mathcal{F}(1)$, which is the threshold for accepting packets with value 1. Using the potential $\phi(t)$ we prove the following invariant:

Theorem 4.3 *For any relaxed input sequence and for any time t,*

$$C \cdot V_{ON}(t) \geq V_{OPT}(t) + \phi(t),$$

where $C = \max\left\{\ln(\alpha) + 2, B\alpha^{1/S} - B + 1\right\}$.

Proof: The proof is by induction. For $t = 0$ we have $V_{ON}(0) = 0 = V_{OPT}(0)$. Since $h_{ON}(0) = 0$ we also have $\phi(0) = 0$ and the claim holds. Assuming the claim holds for $t = t_0$, we will prove it holds for time $t + 1$.

If at time t_0 both queues are empty and the input has ended, then the inequality holds trivially for any $t \geq t_0$. Otherwise, the event at time $t + 1$ is either `arrive(`v`)` or `send()`. We analyze each case separately.

If the next event is `arrive(`v`)`, we analyze its effect in two stages: First, the online policy decides whether to accept v or to reject it, then the offline decides. This separation obviously has no effect on the behavior of both policies, so it is sufficient to prove that the claim holds after each stage. Using this technique, at each state either $V_{ON}()$ or $V_{OPT}()$ remains unchanged. If at any stage the active policy rejects the packet, then the inequality remains unchanged and therefore the claim obviously holds. If the packet is accepted we analyze the inequality separately for each policy and for each status of $h_{ON}(t)$.

If the online policy accepts v and $h_{ON}(t) < \mathcal{F}(1)$: Since the online accepts any packet when $h_{ON}(t) < \mathcal{F}(1)$, its value in the relaxed input must be 1. Since $V_{ON}(t+1) - V_{ON}(t) = v = 1$, the left side of the inequality increases by C. As for the right size of the inequality, it remains unchanged if $\frac{h_{ON}(t+1)}{F^{-1}(1)} B \leq h_{OPT}(t)$, since the potential remains 0. Otherwise the increase of the potential is bounded by:

$$\begin{aligned}
& \frac{h_{ON}(t+1)}{\mathcal{F}(1)} B - h_{OPT}(t) - \frac{h_{ON}(t)}{\mathcal{F}(1)} B + h_{OPT}(t) = \frac{h_{ON}(t+1) - h_{ON}(t)}{\mathcal{F}(1)} B \\
= \quad & \frac{B}{\mathcal{F}(1)} = \frac{B}{\lceil \frac{B}{\ln(\alpha)+2} \rceil} \leq \frac{B}{\frac{B}{\ln(\alpha)+2}} = \ln(\alpha) + 2
\end{aligned}$$

The claim holds for $C \geq \ln(\alpha) + 2$.

If the offline policy accepts v and $h_{ON}(t+1) < \mathcal{F}(1)$: By Lemma 4.2 we must have $v = 1$. The left side of the inequality remains unchanged while the right side changes as follows:

$$\begin{aligned}
& V_{OPT}(t+1) - V_{OPT}(t) + \frac{h_{ON}(t+1)}{\mathcal{F}(1)} B - h_{OPT}(t+1) - \frac{h_{ON}(t+1)}{\mathcal{F}(1)} B + h_{OPT}(t) \\
= \quad & v - [h_{OPT}(t) + 1] + h_{OPT}(t) = 1 - h_{OPT}(t) - 1 + h_{OPT}(t) = 0,
\end{aligned}$$

and the claim holds.

If the online policy accepts v and $h_{ON}(t) \geq \mathcal{F}(1)$: Due to the relaxation of the input, we have $v = \mathcal{F}^{-1}(h_{ON}(t))$. The benefit of the online increases by v, therefore the left side of the equation increases by vC. The change in the potential is:

$$\begin{aligned}
& (B - h_{OPT}(t)) \mathcal{F}^{-1}(h_{ON}(t+1)) + \sum_{i=\mathcal{F}(1)}^{h_{ON}(t+1)-1} \mathcal{F}^{-1}(i) \\
& \quad - (B - h_{OPT}(t)) \mathcal{F}^{-1}(h_{ON}(t)) - \sum_{i=\mathcal{F}(1)}^{h_{ON}(t)-1} \mathcal{F}^{-1}(i) \\
= \quad & (B - h_{OPT}(t)) \left[\mathcal{F}^{-1}(h_{ON}(t) + 1) - \mathcal{F}^{-1}(h_{ON}(t))\right] + \mathcal{F}^{-1}(h_{ON}(t) + 1 - 1) \\
= \quad & (B - h_{OPT}(t)) \left(v\alpha^{1/S} - v\right) + v = v\left[(B - h_{OPT}(t)) \left(\alpha^{1/S} - 1\right) + 1\right] \\
\leq \quad & v\left[B\left(\alpha^{1/S} - 1\right) + 1\right]
\end{aligned}$$

Where the second identity is by Lemma 3.1. The inductive claim holds for $C \geq B\alpha^{1/S} - B + 1$.

If the offline policy accepts v and $h_{ON}(t) \geq \mathcal{F}(1)$: By Lemma 4.2, we have $v \leq \mathcal{F}^{-1}(h_{ON}(t+1))$. The left side of the inequality remains unchanged, which the change to the right side is as follows:

$$\begin{aligned}
& V_{OPT}(t+1) - V_{OPT}(t) + (B - h_{OPT}(t+1)) \mathcal{F}^{-1}(h_{ON}(t+1)) \\
& \quad - (B - h_{OPT}(t)) \mathcal{F}^{-1}(h_{ON}(t+1)) + \sum_{i=\mathcal{F}(1)}^{h_{ON}(t+1)-1} \mathcal{F}^{-1}(i) - \sum_{i=\mathcal{F}(1)}^{h_{ON}(t+1)-1} \mathcal{F}^{-1}(i) \\
= \quad & v + (h_{OPT}(t) - h_{OPT}(t+1)) \mathcal{F}^{-1}(h_{ON}(t+1)) + 0 \\
= \quad & v - \mathcal{F}^{-1}(h_{ON}(t+1)) \leq 0
\end{aligned}$$

This implies that the right side of the inequality cannot increase, and therefore the inductive claim remains true.

In the case of a `send()` event, the benefit of both policies remains unchanged. It is sufficient to prove that the potential does not increase due to a `send()` event. If the queue of the Smooth Selective Barrier Policy is already empty when the `send()` event occurs, then the potential is 0, and remains unchanged. If the queue of the optimal policy is empty when the `send()` event occurs, then the queue of the online must be empty as well, otherwise the offline has rejected at least one packet falsely, and can be improved. Therefore, it is sufficient to analyze the case where both queues drain one packet.

If $h_{ON}(t) < \mathcal{F}(1)$ then the potential is either 0 or $\frac{h_{ON}(t)}{\mathcal{F}(1)}B - h_{OPT}(t)$. If after a packet is sent the potential is 0 then trivially the potential did not increase. Otherwise, the change in the potential is at most:

$$\begin{aligned}
&\phi(t+1) - \phi(t) \\
&= \frac{h_{ON}(t+1)}{\mathcal{F}(1)}B - h_{OPT}(t+1) - \frac{h_{ON}(t)}{\mathcal{F}(1)}B + h_{OPT}(t) \\
&= \frac{h_{ON}(t) - 1 - h_{ON}(t)}{\mathcal{F}(1)}B - h_{OPT}(t) + 1 + h_{OPT}(t) = 1 - \frac{B}{\mathcal{F}(1)}
\end{aligned}$$

Since $\mathcal{F}(1) \leq B$ the potential can only decrease, and therefore the inductive claim holds.

If $h_{ON}(t) \geq \mathcal{F}(1)$ the change in the potential is:

$$\begin{aligned}
&\phi(t+1) - \phi(t) \\
&= (B - h_{OPT}(t+1))\mathcal{F}^{-1}(h_{ON}(t+1)) + \sum_{i=\mathcal{F}(1)}^{h_{ON}(t+1)-1} \mathcal{F}^{-1}(i) \\
&\quad - (B - h_{OPT}(t))\mathcal{F}^{-1}(h_{ON}(t)) - \sum_{i=\mathcal{F}(1)}^{h_{ON}(t)-1} \mathcal{F}^{-1}(i) \\
&= (B - h_{OPT}(t) + 1)\mathcal{F}^{-1}(h_{ON}(t) - 1) + \sum_{i=\mathcal{F}(1)}^{h_{ON}(t)-2} \mathcal{F}^{-1}(i) \\
&\quad -(B - h_{OPT}(t))\mathcal{F}^{-1}(h_{ON}(t)) - \sum_{i=\mathcal{F}(1)}^{h_{ON}(t)-1} \mathcal{F}^{-1}(i) \\
&= (B - h_{OPT}(t)) \left[\mathcal{F}^{-1}(h_{ON}(t) - 1) - \mathcal{F}^{-1}(h_{ON}(t))\right] \leq 0
\end{aligned}$$

As the potential does not increase, the claim holds.

Since the claim is not violated after any step of the induction, it holds for any appropriate value of C. The constraints on C are dictated by the values of B and α, and are set to:

$$C \geq \max\left(\ln(\alpha) + 2, B\alpha^{1/S} - B + 1\right)$$

□

Theorem 4.3 proves the existence of a competitive ratio C, for the Smooth Selective Barrier Policy with the suggested function $\mathcal{F}(\cdot)$. Assuming $B \geq \ln(\alpha) + 2$, we derive the following bound on C.

Theorem 4.4 *For $B \geq \ln(\alpha) + 2$ The competitive ratio of the Smooth Selective Barrier Policy is at most* $C \leq \max\left\{8.155, \ln(\alpha) + 2 + \frac{9\ln^2(\alpha)}{B} + \frac{3\ln(\alpha)}{B}\right\}$

Proof: We first prove the bound for $\alpha \geq e$. We prove the claim separately for each part of the maximum expression derived in Theorem 4.3. For $C \geq \ln(\alpha) + 2$ The claim trivially holds. We notice that for $\alpha \geq e$ we have $\ln(\alpha) \geq 1$ and analyze $B\alpha^{1/S} - B + 1$.

$$B\alpha^{1/S} - B + 1 = B\alpha^{\frac{1}{B-\mathcal{F}(1)}} - B + 1 = B\alpha^{\frac{1}{B-\lceil B/(\ln(\alpha)+2)\rceil}} - B + 1 = \quad Be^{\frac{\ln(\alpha)}{B-B/(\ln(\alpha)+2)-1}} - B + 1$$

Given the inequality $e^x \leq 1 + x + x^2$, which holds for any $0 \leq x \leq 1$, we substitute x with $\ln(\alpha)/[B - B/(\ln(\alpha)+2) - 1]$. Note that $B - \lceil B/(\ln(\alpha)+2) \rceil \geq \ln(\alpha)$ since we assume $B \geq \ln(\alpha) + 2$.

$$
\begin{aligned}
& B\left(1 + \frac{\ln(\alpha)}{B - B/(\ln(\alpha)+2) - 1} + \left(\frac{\ln(\alpha)}{B - B/(\ln(\alpha)+2) - 1}\right)^2\right) - B + 1 \\
= \quad & \frac{B\ln(\alpha)}{B - B/(\ln(\alpha)+2) - 1} + \frac{B\ln^2(\alpha)}{(B - B/(\ln(\alpha)+2) - 1)^2} + 1 \\
= \quad & 1 + \frac{\ln(\alpha)}{1 - 1/(\ln(\alpha)+2) - 1/B} + \frac{\ln^2(\alpha)}{B[1 - 1/(\ln(\alpha)+2) - 1/B]^2}
\end{aligned}
$$

Since $\alpha \geq e$ and $B \geq \ln(\alpha) + 2$, we have $\ln(\alpha) \geq 1$ and $B \geq 3$. Therefore, the last part of the expression is bounded by:

$$
\frac{\ln^2(\alpha)}{B[1 - 1/(\ln(\alpha)+2) - 1/B]} \leq \frac{\ln^2(\alpha)}{B[1 - 1/3 - 1/3]^2} = \frac{9\ln^2(\alpha)}{B}
$$

The rest of the expression is bounded as follows:

$$
\begin{aligned}
& 1 + \frac{\ln(\alpha)}{1 - 1/(\ln(\alpha)+2) - 1/B} = \ln(\alpha) + 1 + \frac{\frac{\ln(\alpha)}{\ln(\alpha)+2} + \frac{\ln(\alpha)}{B}}{1 - 1/(\ln(\alpha)+2) - 1/B} \\
= \quad & \ln(\alpha) + 1 + \frac{1 - \frac{2}{\ln(\alpha)+2} + \frac{\ln(\alpha)}{B}}{1 - 1/(\ln(\alpha)+2) - 1/B} = \ln(\alpha) + 2 + \frac{\frac{\ln(\alpha)}{B} - \frac{1}{\ln(\alpha)+2} + \frac{1}{B}}{1 - 1/(\ln(\alpha)+2) - 1/B} \\
\leq \quad & \ln(\alpha) + 2 + \frac{3\ln(\alpha)}{B}
\end{aligned}
$$

The last inequality holds since $B \geq \ln(\alpha) + 2$ and $\alpha \geq e$. Adding the parts of the analysis together, we have

$$
B\alpha^{1/S} - B + 1 \leq \ln(\alpha) + 2 + \frac{9\ln^2(\alpha)}{B} + \frac{3\ln(\alpha)}{B} = \ln(\alpha) + 2 + O\left(\frac{\ln^2(\alpha)}{B}\right)
$$

This completes the proof for $\alpha \geq e$ and $B \geq \ln(\alpha) + 2$. For $1 \leq \alpha \leq e$ we use a different analysis. We rely on the following lemma from [1], relating to a greedy policy, which accepts packets as long as the queue is not full, but might use a smaller queue:

Lemma 4.5 *[1] A greedy policy with a queue of size xB, for some $0 < x \leq 1$ accepts at least a x fraction of the number of packets that any other policy with a queue of size B accepts.*

The Smooth Selective Barrier Policy accepts packets greedily until the number of packets in the queue reaches the threshold of $B/\mathcal{F}(1)$. Inductively, its queue always holds more packets than a greedy policy with a queue of size $B/\mathcal{F}(1)$, and therefore, by Lemma 4.5 the competitive ratio of accepted packets (ignoring packet values) is at least:

$$
\frac{B}{\mathcal{F}(1)} = \frac{B}{\lceil \frac{B}{\ln(\alpha)+2} \rceil} \leq \ln(\alpha) + 2
$$

At a worst case scenario, the Smooth Selective Barrier Policy accepts only packets of value 1, while the offline accepts only packet of value α. Therefore, the competitive ratio (considering packet values) is at most the ratio of the number of accepted packets times the ratio between the largest and smallest packet values, meaning that for $\alpha \leq e$,

$$
\alpha(\ln(\alpha) + 2) \leq e(\ln(e) + 2) = 3e \approx 8.155,
$$

which completes the proof. □

5 Improved Bounds for Low α

For a sufficiently large B such that $B = \omega(\ln^2 \alpha)$ factor of $\ln^2(\alpha)/B$ is negligible. The main bottleneck in Theorem 4.4 is when $1 \leq \alpha \leq e$, and causes the competitive ratio to be 8.155 rather than approximately 3. If α is sufficiently small, setting $\mathcal{F}(v) = B$ for any $v \in [1, \alpha]$ may be better. This policy is is equivalent to a greedy policy which accepts any packet, and its competitive ratio, as shown is [1], is as follows.

Theorem 5.1 *[1] The competitive ratio of the greedy policy is α*

Since $\alpha \leq \ln(\alpha) + 2$ for $1 \leq \alpha \leq e$ we have the following corollary:

Corollary 5.2 *A combined policy that runs the Smooth Selective Barrier Policy fur $\alpha \geq e$ and the Greedy Policy for $1 \leq \alpha < e$ has a competitive ratio of at most $\ln(\alpha) + 2 + O(\frac{\ln^2(\alpha)}{B})$*

Solving numerically, a competitive ratio of α is better than $\ln(\alpha)+2$ for any $1 \leq \alpha \leq 3.146$. However, better bounds exist for low values of α. We define in Appendix A the `Rounded Ratio Partition Policy`, which has the following competitive ratio:

Theorem 5.3 *The competitive ratio of the Rounded Ratio Partition Policy is at most $2\sqrt{\alpha} - 1 + O(\sqrt{\alpha}/B)$*

The definition of the policy and the proof of Theorem 5.3 appear in Appendix A. If B is large enough such that $O\left(\frac{\sqrt{\alpha}}{B}\right)$ is negligible, and assuming that α is small, the Rounded Ratio Policy is preferable over the greedy policy and the Smooth Selective Barrier Policy. Combining policies, we derive the following corollary:

Corollary 5.4 *A combined policy that based on the value of α, runs either the Smooth Selective Barrier Policy or the Rounded Ratio Partition Policy, has a competitive ratio of*

$$\min\left\{2\sqrt{\alpha} - 1, \ln(\alpha) + 2\right\} + O\left(\frac{\ln^2(\alpha)}{B}\right)$$

Solving numerically, the competitive ratio is improved for $1 \leq \alpha \leq 5.558$, since in that range $2\sqrt{\alpha} - 1 \leq \ln(\alpha) + 2$.

6 Lower Bound

By changing the function $\mathcal{F}(\cdot)$ used by the Smooth Selective Barrier Policy we can define any online policy that takes into account only the value of the arriving packet and the number of packets in the queue in order to decide whether to accept or reject a packet, ignoring the values of the packets in the queue, or the history of the input sequence. We refer to such policies as `Static Threshold Policies`. In this section we prove a lower bound of asymptotically $\ln(\alpha) + 2$ for the competitive ratio of Static Threshold policies. Specifically, we prove the following theorem:

Theorem 6.1 *For any value $\epsilon > 0$, there exists a value β such that for any $\alpha \geq \beta$, no Static Threshold policy has a competitive ratio of less than $\ln(\alpha) + 2 - \epsilon$.*

To prove Theorem 6.1, we observe the performance of an arbitrary Static Threshold policy on two sets of $n + 1$ different inputs each, where n will be determined later. We denote the first set as $Up(i)$, and the second set as $UpDown(i)$, where $i \in [0, n]$. First, we prove several properties that are common to all online policies, using the Up series. We then define the $UpDown$ series as extensions to Up series, and observe how these extensions can increase the competitive ratio for Static Threshold policies. Before defining the input series, we introduce the notations that will be used in our analysis:

Definition 6.2 *Let V_i and $\widehat{V_i}$ denote the benefits of an arbitrary Static Threshold policy on inputs $Up(i)$ and $UpDown(i)$, respectively. Let U_i and $\widehat{U_i}$ denote the benefits of the optimal offline on inputs $Up(i)$ and $UpDown(i)$, respectively. Let $\rho_i = U_i/V_i$ and $\widehat{\rho_i} = \widehat{U_i}/\widehat{V_i}$.*

$Up(0)$ is defined as a burst of B packets with value 1 that arrive at $t = 0$. $Up(i)$ is defined as the same packets from $Up(i-1)$ followed by B packets with value $\alpha^{i/n}$. The optimal offline response to $Up(i)$ is to accept only the last B packets, gaining a benefit of $U_i = B\alpha^{i/n}$. The online will accept $x_0 = \lfloor \mathcal{F}(1) \rfloor$ packets with value 1, $x_1 = \max\left\{0, \lfloor \mathcal{F}(\alpha^{1/n}) \rfloor - x_0\right\}$ packets with value $\alpha^{1/n}$, etc. gaining a total benefit of $V_i = \sum_{0 \le j \le i} x_j \alpha^{j/n}$. The competitive ratio of the online over the i-th input is $\rho_i = U_i / V_i$.

The Up series was used in [2] to prove a general lower bound of $\ln(\alpha) + 1$, for the competitive ratio of any online policy. Without loss of generality, we assume that $\sum_{0 \le i \le n} x_i = B$, otherwise we can increase x_0 appropriately and lower all the ρ_i. The following convenient convexity property holds for the sequence:

Lemma 6.3 *Assume $\sum_{0 \le i \le n} x_i = B$. If we increase x_j, then ρ_j decreases, but for some other $k \ne j$, ρ_k increases.*

The proof of Lemma 6.3 appears in Appendix B.

The main conclusion from Lemma 6.3 is that given a series of constraints $\rho_i \le a_i$, we can search for a feasible solution $x_0, x_1, ..., x_n$ in a greedy manner: Select x_0 such that $\rho_0 = a_0$, then select x_1 such that $\rho_1 = a_1$, etc. If at the end of the process we have $\sum_i x_i \le B$ then we have a feasible solution. Otherwise, if $\sum_i x_i > B$ then by Lemma 6.3 no feasible solution exists, since any change in the x_i that will decrease one of the ρ_i, will necessarily increase another.

We now define the *UpDown* series. $UpDown(0)$ is identical to $Up(0)$. The beginning of $UpDown(i)$ is similar to $Up(i)$, however the sequence continues after $t = 0$, with i bursts of packets. At time $t = x_i$, x_i packets of value $\alpha^{(i-1)/n}$ arrive. At time $t = x_i + x_{i-1}$, x_{i-1} packet of value $\alpha^{(i-2)/n}$ arrive. The sequence continues with packets of decreasing value, until at time $x_1 + x_2 + ... + x_i$, the last burst containing x_1 packets with value 1 arrives. Since the number of packets in each burst equals the number of time units that passed since the last burst, there has to be enough space in the queue to accept all the packets in the burst, regardless of the policy. However, for a Static Threshold policy, when a burst arrives, the number of packets in the queue is equal to the threshold for which the policy rejects packets with this value. Therefore, we have the following observation:

Observation 6.4 *The benefit of a Static Threshold policy for input UpDown(i) is V_i, the same as the benefit for input Up(i). The benefit of the optimal policy for input UpDown(i) is $\widehat{U_i} = U_i + \sum_{1 \le j \le i} x_j \alpha^{j/n}$.*

Unfortunately, the convexity property from Lemma 6.3 does not hold for *UpDown* series. For example, for large n and small α, increasing x_0 and decreasing x_1 appropriately, may lower all the ρ_i. Therefore, we cannot search for feasible solutions in a greedy manner. Instead, we measure how bad is the *Down* part in the *UpDown* series by comparing $\widehat{\rho_i}$ to ρ_i.

The following lemma lower bounds the gain to the competitive ratio derived by *UpDown* inputs, compared to the competitive ratio over *Up* inputs:

Lemma 6.5 $\widehat{\rho_i} \ge \max\left\{\rho_i, \rho_i + \alpha^{-1/n} - \rho_i \alpha^{-(i+1)/n}\right\}$

The proof of Lemma 6.5 appears in Appendix B.

Proof of Theorem 6.1: Obviously, if $\rho_i \ge \ln(\alpha) + 2$, then also $\widehat{\rho_i} \ge \ln(\alpha) + 2$. Otherwise, we have from Lemma 6.5 that $\widehat{\rho_i} \ge \rho_i + \alpha^{-1/n} - \frac{\ln(\alpha)+2}{\alpha^{(i+1)/n}}$. Due to the general lower bound of $\ln(\alpha) + 1$ (from [2]), for sufficiently large n there exists an index i such that $\rho_i \ge \ln(\alpha) + 1 - \frac{1}{2}\epsilon$, where $\epsilon > 0$ is arbitrary small. If $\rho_i < \ln(\alpha) + 2$ we have to prove that $\widehat{\rho_i} \ge \rho_i + 1 - \frac{1}{2}\epsilon \ge \ln(\alpha) + 2 - \epsilon$. For $n \ge \frac{4}{\epsilon}\ln(\alpha) \ge -\ln(\alpha)/\ln(1 - \frac{1}{4}\epsilon)$, we have $\alpha^{-1/n} \ge 1 - \frac{1}{4}\epsilon$. To complete our proof we have to show that $(\ln(\alpha) + 2)\alpha^{-(i+1)/n} \le \frac{1}{4}\epsilon$ holds for sufficiently large α. However, this holds only if $i \ge cn - 1$, for some constant $c > 0$. For $i < cn - 1$ we can only conclude that $\widehat{\rho_i} \ge \rho_i$.

The following lemma proves that for any integer m, even if we allow ρ_0, ρ_1, ...,$\rho_{n/m}$ to be slightly larger than $\ln(\alpha) + 1$ (but less than $\ln(\alpha) + 2$), the competitive ratio for the remaining inputs cannot be much lower than $\ln(\alpha) + 1$.

Lemma 6.6 *Assume $\alpha > e^2$. Let $m \ge 4$, let $\delta = \frac{2}{m} + \frac{1}{\ln(\alpha)} < 1$ and let $n > m\ln(\alpha)(\ln(\alpha)+1)$, such that $n = km$ for some integer k. There is no feasible solution for $x_0, x_1, ..., x_n$ such that $\rho_i < \ln(\alpha) + 2 - \delta$ for $0 \le i \le n/m$ and $\rho_i < \ln(\alpha) + 1 - \delta$ for $n/m + 1 \le i \le n$.*

The proof of Lemma 6.6 appears in Appendix B.

For any $0 < \epsilon < 1$ we choose $m = \frac{5}{\epsilon}$ and $\beta = m^{3m}$. This ensures that for any $\alpha \geq \beta$ we have $\frac{\epsilon}{2} > \delta = \frac{2}{m} + \frac{1}{\ln(\alpha)}$ and $\frac{\epsilon}{4} > (\ln(\alpha) + 2)\alpha^{-1/m}$. For any fixed α we set $n > m \ln(\alpha)(\ln(\alpha) + 1)$, which ensures that $\alpha^{-1/n} > 1 - \frac{\epsilon}{4}$.

If there exists an i such that $\rho_i > \ln(\alpha) + 2 - \epsilon$, we are done. Otherwise, by Lemma 6.6, there exists an $i > \frac{n}{m}$ such that $\rho_i > \ln(\alpha) + 1 - \frac{\epsilon}{2}$, and by Lemma 6.5, we have $\widehat{\rho_i} > \rho_i + 1 - \frac{\epsilon}{4} - \frac{\epsilon}{4} \geq \ln(\alpha) + 2 - \epsilon$, which completes our proof. □

7 Conclusion

We have presented a nearly optimal online policy for non-preemptive queue management with continuous values. The policy is suboptimal due to the static threshold for the value of the next accepted packets. We note that the same threshold used when the queue was filled is still used as the queue drains. Potentially, the online policy may lose packets with accumulating value of almost the queue's contents as the queue drains, which explains the difference between the lower bound of $\ln(\alpha)+1$ to the asymptotic upper bound $\ln(\alpha) + 2$. It might be the case that a policy with dynamic thresholds, that considers the entire history of the sequence, i.e. arrival times and values of both the packets in the queue and the packets that were already sent or rejected, will have a competitive ratio that beats the asymptotic $\ln(\alpha) + 2$ bound.

References

[1] W. Aiello, Y. Mansour, S. Rajagopolan and A. Rosen. Competitive Queue Policies for Differentiated Services. In *Proc. IEEE INFOCOM*, 2000.

[2] N. Andelman, Y. Mansour and A. Zhu. Competitive Queueing Policies for QoS Switches. In *Proc. 14th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2003.

[3] Y. Azar and Y. Richter. Management of Multi-Queue Switches In QoS Networks In *Proc. 35th ACM Symposium on Theory of Computing (STOC)*, 2003. To appear.

[4] D. Black, S. Blake, M. Carlson, E. Davis, Z. Wang and W. Weiss. An Architechture for Differentiated Services. *Internet RFC 2475*, 1998.

[5] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[6] D. Clark and J. Wroklawski. An Approach to Service Allocation in the Internet. Internet draft, 1997. Available from `diffserv.lcs.mit.edu`.

[7] S. Floyd and V. Jacobson. Random Early Detection Gateways for Congestion Avoidance. In *IEEE/ACM Trans. on Networking*, Vol. 1(4), 1993.

[8] E. L. Hahne, A. Kesselman and Y. Mansour. Competitive Buffer Management for Shared Memory Switches. In *Proc. 13th ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, 2001.

[9] A. Kesselman, Z. Lotker, Y. Mansour, B. Patt-Shamir, B. Schieber and M. Sviridenko. Buffer Overflow Management in QoS Switches. In *Proc. 33rd ACM Symposium on Theory of Computing (STOC)*, 2001.

[10] A. Kesselman and Y. Mansour. Loss-Bounded Analysis for Differentiated Services. In *Proc. 12th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2001.

[11] A. Kesselman and Y. Mansour. Harmonic Buffer Management Policy for Shared Memory Switches. In *Proc. IEEE INFOCOM*, 2002.

[12] A. Kesselman, Y. Mansour and R. van Stee. Improved Competitive Guarantees for Qos Buffering. Work in progress, 2003.

[13] Z. Lotker and B. Patt-Shamir. Nearly Optimal FIFO Buffer Management for DiffServ. In *Proc. 21st ACM Symposium on Principles of Distributed Computing (PODC)*, 2002.

[14] K. Nichols, V. Jacobson, and L. Zhang. A Two-bit Differentiated Services Architecture for the Internet, *Internet draft*, 1997.

[15] M. Sviridenko. A Lower Bound for On-Line Algorithms in the FIFO Model. Unpulished Manuscript, 2001.

A The Rounded Ratio Partition Policy

The `Ratio Partition` (RR) policy was suggested in [2] for the two value case, i.e. where packet values can be only 1 (low value) or α (high value). The competitive ratio of this policy is $(2\alpha - 1)/\alpha$, which is optimal for an online policy. The RR policy accepts any high value packet (as long as the queue is not full), and marks $\alpha/(\alpha - 1)$ low value packets that are in the queue (marking is done from bottom to top). A low value packet is accepted if by filling the remaining space in the queue with high value packets marks the new low value packet. In the analysis, rejected high value packets that were accepted by the offline match the marked low value packets, that were rejected by the offline. Rejected low value packets accepted by the offline match the remaining low value packets accepted by the online.

The analysis, however, assumed the queue size B is large enough in order to ignore rounding errors, due to the fact that when the online first rejects a low value packet, the analysis may assume that a fraction of the packet was indeed accepted. The cost of this difference is proportional to $1/B$, so a more accurate upper bound to the policy is $(2\alpha - 1)/\alpha + O(1/B)$

We convert the Ratio Partition Policy to a multiple value policy by translating the inputs to two values. The translation function $\mathcal{T}(\cdot)$ is defined as follows:

$$\mathcal{T}(t) = \begin{cases} 1 & : \quad 1 \leq v < \sqrt{\alpha} \\ \sqrt{\alpha} & : \quad \sqrt{\alpha} \leq v \leq \alpha \end{cases}$$

We define the converted policy for multiple values as follows:

Definition A.1 *The* `Rounded Ratio Partition Policy` *executes a simulation of the Ratio Partition Policy with values* 1 *and* $\sqrt{\alpha}$*. Each event* `arrive(`v`)` *is translated to the event* `arrive(`$\mathcal{T}(v)$`)`*.* v *is accepted if the Ratio Partition accepts* $\mathcal{T}(v)$*, and rejected otherwise.*

Theorem A.2 *(5.3) The competitive ratio of the Rounded Ratio Partition Policy is at most* $2\sqrt{\alpha} - 1 + O(\sqrt{\alpha}/B)$

Proof: The competitive ratio of the Ratio Partition is $\frac{2\alpha-1}{\alpha} + O(\frac{1}{B})$. Since the simulation uses values of 1 and $\sqrt{\alpha}$, we substitute α with $\sqrt{\alpha}$ and get a competitive ratio of $\frac{2\sqrt{\alpha}-1}{\sqrt{\alpha}} + O(\frac{1}{B})$ for the simulation. The Rounded Ratio Partition loses a factor of at most $\sqrt{\alpha}$ due to the translation of packet values. Therefore, the competitive ratio is upper bounded by $2\sqrt{\alpha} - 1 + O(\sqrt{\alpha}/B)$ □

B Proofs from Section 6

B.1 Proof of Lemma 6.3

Proof: Assume we increase x_j by δ, then we need to decrease some other variable x_k appropriately. If more than one variable changes then we split the change in x_j into several steps. The U_i are invariant to the x_i variables and therefore remain unchanged. Therefore, it is sufficient to observe the changes in V_i, $0 \leq i \leq n$, which are in opposite direction to the changes in ρ_i.

If $k > j$ then V_j increases by $\delta\alpha^{j/n}$, as well as V_i for any $j < i < k$. V_k decreases by $\delta(\alpha^{k/n} - \alpha^{j/n})$ as well as V_i for any $i \geq k$. For $i < j$, V_i remains unchanged.

If $k < j$ then V_j increases by $\delta(\alpha^{j/n} - \alpha^{k/n})$ as well as V_i for any $i \geq j$. For any $k \leq i < j$, V_i decreases by $\delta\alpha^{k/n}$. For $i < k$, V_i remains unchanged.

Summing over the changes in all of the steps, V_j increases, and therefore ρ_j decreases. If the minimal index k such that x_k decreased fulfills $k < j$ then V_k decreases and therefore ρ_k increases. Otherwise, if only indexes larger than j changed, then V_n decreases, meaning that ρ_n increases. □

B.2 Proof of Lemma 6.5

Proof: Since $\widehat{V_i} = V_i$ and $\widehat{U_i} \geq U_i$, $\widehat{\rho_i} \geq \rho_i$ trivially holds. We now prove the second part of the maximum expression:

$$\begin{aligned}
\widehat{\rho_i} &= \frac{\widehat{U_i}}{\widehat{V_i}} = \frac{U_i + \sum_{1\leq j\leq i} x_j \alpha^{(j-1)/n}}{V_i} = \rho_i + \frac{\left(\sum_{0\leq j\leq i} x_j \alpha^{j/n} - x_0\right)\alpha^{-1/n}}{V_i} \\
&= \rho_i + \frac{(V_i - x_0)\,\alpha^{-1/n}}{V_i} = \rho_i + \alpha^{-1/n} - \frac{x_0\alpha^{-1/n}}{V_i} = \rho_i + \alpha^{-1/n} - \frac{x_0\alpha^{-1/n}\rho_i}{U_i} \\
&= \rho_i + \alpha^{-1/n} - \frac{x_0\alpha^{-1/n}\rho_i}{B\alpha^{i/n}} \geq \rho_i + \alpha^{-1/n} - \frac{\rho_i}{\alpha^{(i+1)/n}}
\end{aligned}$$

□

B.3 Proof of Lemma 6.6

Proof: We shall attempt to find a solution for $x_0, x_1, ..., x_n$ which satisfies the constraints with equalities, i.e. $\rho_i = \ln(\alpha) + 2 - \delta$ for $i \leq \frac{n}{m}$ and $\rho_i = \ln(\alpha) + 1 - \delta$ otherwise. According to Lemma 6.3, if such a solution exists, it can be found in a greedy manner.

To satisfy $\rho_0 = U_0/V_0 = \ln(\alpha) + 2 - \delta$, since $U_0 = B$ and $V_0 = x_0$, we have $x_0 = B(\ln(\alpha) + 2 - \delta)^{-1}$. For $i > 0$, we have:

$$\rho_i^{-1} = \frac{V_i}{U_i} = \frac{V_{i-1} + x_i\alpha^{i/n}}{B\alpha^{i/n}} = \frac{V_{i-1}}{B\alpha^{i/n}} + \frac{x_i\alpha^{i/n}}{B\alpha^{i/n}} = \frac{V_{i-1}}{U_{i-1}\alpha^{1/n}} + \frac{x_i}{B} = \rho_{i-1}^{-1}\alpha^{-1/n} + \frac{x_i}{B}$$

With the exception of $\rho_{n/m+1}$, we have $\rho_i = \rho_{i-1}$ for any $i > 0$. Therefore, we have that $x_i = B\rho_{i-1}^{-1}(1 - \alpha^{-1/n})$. For $1 \leq i \leq n/m$, where $\rho_{i-1} = \ln(\alpha) + 2 - \delta$, we have

$$x_i = B\frac{1 - \alpha^{-1/n}}{\ln(\alpha) + 2 - \delta}.$$

For $n/m + 2 \leq i \leq n$, where $\rho_{i-1} = \ln(\alpha) + 1 - \delta$, we have

$$x_i = B\frac{1 - \alpha^{-1/n}}{\ln(\alpha) + 1 - \delta}.$$

For $i = n/m + 1$ we have

$$x_{\frac{n}{m}+1} = B\left(\frac{1}{\ln(\alpha) + 1 - \delta} - \frac{\alpha^{-1/n}}{\ln(\alpha) + 1 - \delta}\right) > B\frac{1 - \alpha^{-1/n}}{\ln(\alpha) + 1 - \delta}.$$

We continue by testing the constraint $\sum_i x_i \leq B$:

$$\frac{\sum_{i=0}^n x_i}{B} \quad \geq \quad \frac{1}{\ln(\alpha) + 2 - \delta} + \frac{n}{m}\frac{1 - \alpha^{-1/n}}{\ln(\alpha) + 2 - \delta} + \left(n - \frac{n}{m}\right)\frac{1 - \alpha^{-1/n}}{\ln(\alpha) + 1 - \delta}$$

We note that $\alpha^{-1/n} = e^{-\ln(\alpha)/n}$ and that for any $x \in [0,1]$, $e^{-x} \leq 1 - x + x^2$. After applying these substitutions, we have:

$$\begin{aligned}
\frac{\sum_{i=0}^n x_i}{B} &\geq \frac{1}{\ln(\alpha) + 2 - \delta} + \frac{n}{m}\left(\frac{\frac{\ln(\alpha)}{n} - \frac{\ln^2(\alpha)}{n^2}}{\ln(\alpha) + 2 - \delta}\right) + \left(n - \frac{n}{m}\right)\left(\frac{\frac{\ln(\alpha)}{n} - \frac{\ln^2(\alpha)}{n^2}}{\ln(\alpha) + 1 - \delta}\right) \\
&= \frac{1 + \frac{\ln(\alpha)}{m}}{\ln(\alpha) + 2 - \delta} + \frac{\frac{m-1}{m}\ln(\alpha)}{\ln(\alpha) + 1 - \delta} - \frac{\frac{\ln^2(\alpha)}{m}}{n(\ln(\alpha) + 2 - \delta)} - \frac{\frac{m-1}{m}\ln^2(\alpha)}{n(\ln(\alpha) + 1 - \delta)}
\end{aligned}$$

Since $\delta < 1$, we can bound the second half of the expression above:

$$\frac{\frac{\ln^2(\alpha)}{m}}{n(\ln(\alpha)+2-\delta)} + \frac{\frac{m-1}{m}\ln^2(\alpha)}{n(\ln(\alpha)+1-\delta)} \le \frac{\frac{\ln^2(\alpha)}{m}}{n\ln(\alpha)} + \frac{\frac{m-1}{m}\ln^2(\alpha)}{n\ln(\alpha)} = \frac{\ln(\alpha)}{n}$$

As for the first part of the expression, we have:

$$\frac{1+\frac{\ln(\alpha)}{m}}{\ln(\alpha)+2-\delta} + \frac{\frac{m-1}{m}\ln(\alpha)}{\ln(\alpha)+1-\delta} = 1 + \frac{\delta}{\ln(\alpha)+1-\delta} - \frac{1+\frac{\ln(\alpha)}{m}}{(\ln(\alpha)+1-\delta)(\ln(\alpha)+2-\delta)}$$

Since $1 > \delta = 2/m + 1/\ln(\alpha) > 0$,

$$\frac{1+\frac{\ln(\alpha)}{m}}{\ln(\alpha)+2-\delta} + \frac{\frac{m-1}{m}\ln(\alpha)}{\ln(\alpha)+1-\delta} > 1 + \frac{\frac{2}{m}+\frac{1}{\ln(\alpha)}}{\ln(\alpha)+1} - \frac{1+\frac{\ln(\alpha)}{m}}{\ln(\alpha)(\ln(\alpha)+1)} = 1 + \frac{\ln(\alpha)}{m\ln(\alpha)(\ln(\alpha)+1)}$$

Combining everything together we have

$$\frac{\sum_{i=0}^{n} x_i}{B} > 1 + \frac{\ln(\alpha)}{m\ln(\alpha)(\ln(\alpha)+1)} - \frac{\ln(\alpha)}{n},$$

which is more than 1 if $n > m\ln(\alpha)(\ln(\alpha)+1)$, which proves that no feasible solution exists. $\square$