数理解析研究所講究録
第 1799 巻 2012 年 84-91



2011 年度冬の LA シンポジウム [18]

# Greedy Algorithms for Multi-Queue Buffer Management with Class Segregation

TOSHIYA ITOH* SEIJI YOSHIMOTO†

**Abstract:** In this paper, we focus on a multi-queue buffer management in which packets of different values are segregated in different queues. Our model consists of $m$ packets values and $m$ queues. Recently, Al-Bawani and Souza [2] presented an online multi-queue buffer management algorithm Greedy and showed that it is 2-competitive for the general $m$-valued case, i.e., $m$ packet values are $0 < v_1 < v_2 < \cdots < v_m$, and it is $(1 + v_1/v_2)$-competitive for the two-valued case, i.e., two packet values are $0 < v_1 < v_2$. For the general $m$-valued case, let $c_i = (v_i + \sum_{j=1}^{i-1} 2^{j-1}v_{i-j})/(v_{i+1} + \sum_{j=1}^{i-1} 2^{j-1}v_{i-j})$ for $1 \leq i \leq m-1$, and let $c_m^* = \max_i c_i$. In this paper, we precisely analyze the competitive ratio of Greedy for the general $m$-valued case, and show that Greedy is $(1 + c_m^*)$-competitive.

**Key Words:** Online Algorithms, Competitive Ratio, Buffer Management, Class Segregation, Quality of Service (QoS), Class of Service (CoS).

## 1 Introduction

Due to the burst growth of the Internet use, network traffic has increased year by year. This overloads networking systems and degrades the quality of communications, e.g., loss of bandwidth, packet drops, delay of responses, etc. To overcome such degradation of the communication quality, the notion of Quality of Service (QoS) has received attention in practice, and is implemented by assigning nonnegative numerical values to packets to provide them with differentiated levels of service (priority). Such a packet value corresponds to the predefined Class of Service (CoS). In general, switches have several number of queues and each queue has a *buffer* to store arriving packets. Since network traffic changes frequently, switches need to control arriving packets to maximize the total priorities of transmitted packets, which is called *buffer management*. Basically, switches have no knowledge on the arrivals of packets in the future when it manages to control new packets arriving to the switches. So the decision made by buffer management algorithm can be regarded as an *online algorithm*, and in general, the performance of online algorithms is measured by *competitive ratio* [8]. Online buffer management algorithms can be classified into two types of queue management (one is *preemptive* and the other is *nonpreemptive*). Informally, we say that an online bufffer management algorithm is preemptive if it is allowed to discard packets buffered in the queues on the arrival of new packets; nonpreemptive otherwise (i.e., all packets buffered in the queues will be eventually transmitted).

### 1.1 Multi-Queue Model

In this paper, we focus on a multi-queue model in which packets of different values are segregated in different queues (see, e.g., [11], [18]). Our model consists of $m$ packet values and $m$ queues‡. Let $\mathcal{V} = \{v_1, v_2, \ldots, v_m\}$ be the set of $m$ nonnegative *packet values*, where $0 < v_1 < v_2 < \cdots < v_m$, and let $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_m\}$ be the set of $m$ queues. A packet of value $v_i \in \mathcal{V}$ is called a $v_i$*-packet*, and a queue storing $v_i$-packets is called a $v_i$-queue. Without loss of generality, we assume that $Q_i \in \mathcal{Q}$ is a $v_i$-queue for each $i \in [1, m]$§. Each $Q_i \in \mathcal{Q}$ has a capacity $B_i \geq 1$, i.e., each $Q_i \in \mathcal{Q}$ can store up to $B_i \geq 1$ packets. Since all packets buffered in queue $Q_i \in \mathcal{Q}$ have the same value $v_i \in \mathcal{V}$, the order of transmitting packets is irrelevant.

For convenience, we assume that time is dis-

*Imaging Science & Engineering Laboratory, Tokyo Institute of Technology. Email: titoh@ip.titech.ac.jp
†Department of Computer Science, Tokyo Institute of Technology. Email: yoshimoto.s.aa@m.titech.ac.jp

‡In general, we can consider a model of $m$ packet values and $n$ queues (with $m \neq n$), but in this paper, we deal with only a model of $m$ packet values and $m$ queues.

§For any pair of integers $a \leq b$, we use $[a, b]$ to denote $\{a, , a+1, \ldots, b\}$.

cretized into slot of unit length. Packets arrive over time and each arriving packet is assigned with a unique (nonintegral) arrival time, a value $v_i \in \mathcal{V}$, and its destination queue $Q_i \in \mathcal{Q}$ (as we have assumed, $Q_i \in \mathcal{Q}$ is a $v_i$-queue). We use $\sigma = \langle e_0, e_1, e_2, \ldots \rangle$ to denote a sequence of *arrive events* and *send events*, where an arrive event corresponds to the arrival of a new packet and a send event corresponds to the transmission of a packet buffered in queues at integral time (i.e., the end of time slot). An online (multi-queue) buffer management algorithm Alg consists of two phases: *admission* phase *schedulilng* phases. In the admission phase, Alg must decide on the arrival of a packet whether to accept or reject the packet without any knowledge on the future arrivals of packets (if Alg is preemptive, then it may discard packets buffered in queues in the admission phase). In the scheduling phase, Alg chooses one of the nonempty queues at send event and exactly one packet is transmitted out of the queue chosen. Since all packets buffered in the same queue have the same value, preemption does not make sense in our model. Thus a packet accepted must eventually be transmitted.

We say that an (online and offline) algorithm is *diligent* if (1) it must accept a packet arriving to its destination queue when the destination queue has vacancies, and (2) it must transmit a packet when it has nonempty queues. It is not difficult to see that any nondiligent algorithm can be transformed to a diligent algorithm without decreasing its benefit (sum of values of transmitted packets). Thus in this paper, we focus on only diligent algorithms.

### 1.2 Main Results

Al-Bawani and Souza [2] recently presented an online multi-queue buffer management algorithm Greedy and showed that it is 2-competitive for the general $m$-valued case, i.e., $m$ packet values are $0 < v_1 < v_2 < \cdots < v_m$, and $(1 + v_1/v_2)$-competitive for the two-valued case, i.e., $m = 2$.

For the general $m$-valued case, let $c_m^* = \max_i c_i$, where for each $1 \le i \le m-1$,

$$c_i = \frac{v_i + \sum_{j=1}^{i-1} 2^{j-1} v_{i-j}}{v_{i+1} + \sum_{j=1}^{i-1} 2^{j-1} v_{i-j}}.$$

In this paper, we precisely analyze the competitive ratio of Greedy for the general $m$-valued case, and show that Greedy is $(1 + c_m^*)$-competitive (see Theorem 4.1). Note that $c_m^* < 1$. Thus we have that $1 + c_m^* < 2$ and for the general $m$ valued case, our results improves the known result that Greedy is 2-competitive [2, Theorem 2.1].

For example, let us consider the case that $v_1 = 1, v_2 = 2$, and $v_{i+1} = v_i + \sum_{j=1}^{i-1} 2^{j-1} v_{i-j}$ for each $i \in [2, m-1]$. It is obvious that $0 < v_1 < v_2 < \cdots < v_m$ and $c_m^* = \max_i c_i = 1/2$. Thus for those packet values, our result guarantees that the algorithm Greedy is 3/2-competitive, while the known result only guarantees that the algorithm Greedy is 2-competitive [2, Theorem 2.1].

### 1.3 Related Works

The competitive analysis for the buffer management policies for switches were initiated by Aiello et al. [1], Mansour et al. [19], and Kesselman et al. [17], and the extensive studies have been made for several models (for comprehensive surveys, see, e.g., [4], [12], [16], [10], [13]). The model we deal with in this paper can be regarded as the generalization of unit-valued model, where the switches consist of $m$ queues of the same buffer size $B$ and all packets have unit value, i.e., $v_1 = v_2 = \cdots = v_m$. Tables 1 and 2 summarize the known results.

On the other hand, the model we deal with in this paper can be regarded as a special case of the general-valued multi-queue model where each of $m$ FIFO queues can buffer at most $B$ packets of different values. For the preemptive multi-queue buffer management, Azar and Richter [6] presented a $(4 + 2\ln\alpha)$-competitive algorithm for the general-valued case (packet values lie between 1 and $\alpha$) and a 2.6-competitive algorithm for the two-valued case (packet values are $v_1 < v_2$, where $v_1 = 1$ and $v_2 = \alpha$). For the general-valued case, Azar and Righter [7] proposed a more efficient algorithm TRANSMIT-LARGEST HEAD (TLH) that is 3-competitive, which is shown to be $(3 - 1/\alpha)$-competitive by Itoh and Takahashi [14].

## 2 Preliminaries

For a sequence $\sigma'$ of arriving packets, we use $\sigma = \langle e_0, e_1, e_2, \ldots \rangle$ to denote a sequence of arrive events and send events. Note that an arrive event corresponds to the arrival of a new packet (at nonintegral time) and a send event corresponds to the transmission of a packet buffered in queues at integral time. The online algorithm Greedy works as fol-

Table 1: Deterministic Competitive Ratio (Unit-Valued Multi-Queue Model)

| Upper Bound | | | Lower Bound | | |
|---|---|---|---|---|---|
| 2 | [6] | — | $2-1/m$ | [6] | $B=1$ |
| 1.889 | [3] | $m \gg B$ | $1.366-\Theta(1/m)$ | [6] | $B \geq 1$ |
| 1.857 | [3] | $B=2$ | $\frac{e}{e-1} \approx 1.582$ | [3] | — |
| $\frac{e}{e-1} \approx 1.582$ | [5] | large $B$ | | | |

Table 2: Randomized Competitive Ratio (Unit-Valued Multi-Queue Model)

| Upper Bound | | | Lower Bound | | |
|---|---|---|---|---|---|
| $\frac{e}{e-1} \approx 1.582$ | [6] | $B > \log m$ | $1.46-\Theta(1/m)$ | [6] | $B=1$ |
| 1.231 | [9] | $m=2$ | 1.466 | [3] | large $m$ |
| | | | 1.231 | [3] | $m=2$ |

lows: At send event, Greedy transmits a packet from the nonempty queue with highest packet value¶, i.e., Greedy transmits a $v_h$-packet if $v_h$-queue is nonempty and all $v_\ell$-queues are empty for $\ell \in [h+1, m]$. At arrive event, Greedy accepts packets in its destination queue until the corresponding queue becomes full.

For an *online* algorithm Alg and a sequence $\sigma$ of arrive and send events, we use $\mathsf{Alg}(\sigma)$ to denote the *benefit* of the algorithm Alg on the sequence $\sigma$, i.e., the sum of values of packets transmitted by the algorithms Alg on the sequence $\sigma$. For a sequence $\sigma$ of arrive and send events, we also use $\mathsf{Opt}(\sigma)$ to denote the *benefit* of the *optimal offline* algorithm Opt on the sequence $\sigma$, i.e., the sum of values of packets transmitted by the optimal offline algorithm Opt that knows the entire sequence $\sigma$ in advance. Our goal is to design an efficient online algorithm Alg that minimizes $\mathsf{Opt}(\sigma)/\mathsf{Alg}(\sigma)$ for any sequence $\sigma$.

At event $e_i$, let $A_h(e_i)$ and $A_h^*(e_i)$ be the total number of $v_h$-packets accepted by Greedy and Opt until the event $e_i$, respectively, $\delta_h(e_i)$ and $\delta_h^*(e_i)$ be the total number of $v_h$-packets transmitted by Greedy and Opt until the event $e_i$, respectively, and $q_h(e_i)$ and $q_h^*(e_i)$ be the total number of $v_h$-packets buffered in $v_h$-queue of Greedy and Opt just after the event $e_i$, respectively. It is immediate to see that for each $h \in [1, m]$ and each event $e_i$,

$$A_h(e_i) = \delta_h(e_i) + q_h(e_i); \tag{1}$$

$$A_h^*(e_i) = \delta_h^*(e_i) + q_h^*(e_i). \tag{2}$$

For a sequence $\sigma$, let $A_h(\sigma)$ and $A_h^*(\sigma)$ be the total number of $v_h$-packets accepted by Greedy and Opt until the end of the sequence $\sigma$, respectively, $\delta_h(\sigma)$ and $\delta_h^*(\sigma)$ be the total number of $v_h$-packets transmitted by Greedy and Opt until the end of the sequence $\sigma$, respectively, and $q_h(\sigma)$ and $q_h^*(\sigma)$ be the number of $v_h$-packets buffered in $v_h$-queue of Greedy and Opt at the end of the sequence $\sigma$, respectively. Note that $q_h(\sigma) = q_h^*(\sigma) = 0$ for each $h \in [1, m]$. From Eqs. (1) and (2), we have that $A_h(\sigma) = \delta_h(\sigma)$ and $A_h^*(\sigma) = \delta_h^*(\sigma)$ for each $h \in [1, m]$.

For the general $m$-valued case, Al-Bawani and Souza showed the following result on the number of packets accepted by Greedy and Opt, which is crucial in the subsequent discussions.

**Lemma 2.1** [2, Lemma 2.2]: *For each $h \in [1, m]$,*

$$\sum_{\ell=h}^{m} \{A_\ell^*(\sigma) - A_\ell(\sigma)\} \leq \sum_{\ell=h}^{m} A_\ell(\sigma).$$

Assume that in the sequence $\sigma = \langle e_0, e_1, e_2, \ldots \rangle$, there exist $k \geq 1$ send events, and for each $j \in [0, k]$, let $s_j$ be the $j$th send event, where $s_0 = e_0$ is an initial send event that transmits a null packet. For each $j \in [1, k]$, let $\Sigma_j$ be the set of arrive events between send events $s_{j-1}$ and $s_j$, i.e., $\Sigma_j$ consists of arrive events after send event $s_{j-1}$ and before send event $s_j$. Notice that $\Sigma_j$ could be an empty set.

¶Since $Q_i \in \mathcal{Q}$ is a $v_i$-queue, such a nonempty queue with highest packet value is *unique* if it exists.

# 3 Greedy vs. Opt

## 3.1 Number of Transmitted Packets

In this subsection, we investigate the relationships between the number of packets transmitted by Greedy and the number of packets transmitted by Opt. For each $h \in [1, m-1]$ and each event $e_i$, let $\xi_h(e_i) = \delta_h(e_i) + \cdots + \delta_m(e_i) - \delta_h^*(e_i)$.

**Claim 3.1:** *For each $h \in [1, m-1]$ and each $j \in [2, k]$, if $q_h(s_{j-1}) + \cdots + q_m(s_{j-1}) > 0$ (i.e., just after $s_{j-1}$, a nonempty $v_\ell$-queue of Greedy with $\ell \in [h, m]$ exists), then $\xi_h(s_j) \geq \xi_h(s_{j-1})$.*

**Proof:** Since every $e_i \in \Sigma_j$ is arrive event, we have that for each $g \in [h, m]$, the number of packets buffered in $v_g$-queue does not decrease at each arrive event $e_i \in \Sigma_j$. From the assumption that $q_h(s_{j-1}) + \cdots + q_m(s_{j-1}) > 0$, it follows that there exists an $\ell \in [h, m]$ such that $v_\ell$-queue of Greedy is nonempty just before send event $s_j$. Thus from the definition of Greedy, it is immediate that for some $r \in [\ell, m]$, Greedy transmits a $v_r$-packet at send event $s_j$, which implies that $\delta_h(s_j) + \cdots + \delta_m(s_j) = \delta_h(s_{j-1}) + \cdots + \delta_m(s_{j-1}) + 1$. So we have that

$$\begin{aligned}
\xi_h(s_j) &= \delta_h(s_j) + \cdots + \delta_m(s_j) - \delta_h^*(s_j) \\
&\geq \{\delta_h(s_{j-1}) + \cdots + \delta_m(s_{j-1}) + 1\} \\
&\qquad - \{\delta_h^*(s_{j-1}) + 1\} \\
&= \delta_h(s_{j-1}) + \cdots + \delta_m(s_{j-1}) - \delta_h^*(s_{j-1}) \\
&= \xi_h(s_{j-1}).
\end{aligned}$$

Notice that the inequality above follows from the fact that $\delta_h^*(s_j) \leq \delta_h^*(s_{j-1}) + 1$. ■

**Claim 3.2:** *For each $h \in [1, m-1]$ and each $j \in [1, k]$, if $q_h^*(s_{j-1}) = 0$ (i.e., just after $s_{j-1}$, $v_h$-queue of Opt is empty), then $\xi_h(s_j) \geq \xi_h(s_{j-1})$.*

**Proof:** Let us consider the following cases: (1) $v_h$-queue of Opt is empty just before send event $s_j$ and (2) $v_h$-queue of Opt is nonempty just before send event $s_j$. For the case (1), it is immediate to see that $\delta_h^*(s_j) = \delta_h^*(s_{j-1})$. So we have that

$$\begin{aligned}
\xi_h(s_j) &= \delta_h(s_j) + \cdots + \delta_m(s_j) - \delta_h^*(s_j) \\
&\geq \delta_h(s_{j-1}) + \cdots + \delta_m(s_{j-1}) - \delta_h^*(s_{j-1}) \\
&= \xi_h(s_{j-1}),
\end{aligned}$$

where the inequality follows from the fact that $\delta_h(s_j) + \cdots + \delta_m(s_j) \geq \delta_h(s_{j-1}) + \cdots + \delta_m(s_{j-1})$.

For the case (2), there exists arrive event $e_i \in \Sigma_j$ such that a $v_h$-packet arrives, because of the assumption that $v_h$-queue of Opt is empty just after the send event $s_{j-1}$. Then from the definition of Greedy, it is easy to see that $v_h$-queue of Greedy is nonemnty just before send event $s_j$ and that at send event $s_j$, Greedy transmits a $v_\ell$-packet with $\ell \in [h, m]$. This implies that $\delta_h(s_j) + \cdots + \delta_m(s_j) = \delta_h(s_{j-1}) + \cdots + \delta_m(s_{j-1}) + 1$. Thus it follows that

$$\begin{aligned}
\xi_h(s_j) &= \delta_h(s_j) + \cdots + \delta_m(s_j) - \delta_h^*(s_j) \\
&\geq \{\delta_h(s_{j-1}) + \cdots + \delta_m(s_{j-1}) + 1\} \\
&\qquad - \{\delta_h^*(s_{j-1}) + 1\} \\
&= \delta_h(s_{j-1}) + \cdots + \delta_m(s_{j-1}) - \delta_h^*(s_{j-1}) \\
&= \xi_h(s_{j-1}).
\end{aligned}$$

Notice that the inequality above follows from the fact that $\delta_h^*(s_j) \leq \delta_h^*(s_{j-1}) + 1$. ■

**Lemma 3.1:** *For each $h \in [1, m-1]$ and each event $e_i$, $\xi_h(e_i) \geq 0$.*

**Proof:** We show the lemma by induction on events $e_i$. It is obvious that $\xi_h(e_0) = 0$. For $t \geq 1$, we assume that $\xi_h(e_i) \geq 0$ for each $i \in [0, t-1]$. If $e_t$ is arrive event, then $\delta_\ell(e_t) = \delta_\ell(e_{t-1})$ for each $\ell \in [h, m]$ and $\delta_h^*(e_t) = \delta_h^*(e_{t-1})$. This implies that $\xi_h(e_t) = \xi_h(e_{t-1})$ and from the induction hypothesis, it follows that $\xi_h(e_t) = \xi_h(e_{t-1}) \geq 0$. In the rest of the proof, we focus on only send events and show the lemma by induction on send events $s_j$.

Base Step: We show that $\xi_h(s_1) \geq 0$. Consider the following cases: (1) there exists arrive event $e_t \in \Sigma_1$ at which a $v_\ell$-packet with $\ell \in [h, m]$ arrives and (2) there exists no arrive event $e_t \in \Sigma_1$ at which a $v_\ell$-packet with $\ell \in [h, m]$ arrives. For the case (1), we have that $v_\ell$-queue of Greedy is nonempty just before send event $s_1$. So from the definition of Greedy, it follows that $\delta_h(s_1) + \cdots + \delta_m(s_1) = 1$. Since $\delta_h^*(s_1) \leq 1$, this implies that $\xi_h(s_1) = \delta_h(s_1) + \cdots + \delta_m(s_1) - \delta_h^*(s_1) \geq 1 - 1 = 0$. For the case (2), it is immediate to see that $\delta_h(s_1) = \cdots = \delta_m(s_1) = 0$ and $\delta_h^*(s_1) = 0$. So we have that $\xi_h(s_1) = \delta_h(s_1) + \cdots + \delta_m(s_1) - \delta_h^*(s_1) = 0 - 0 = 0$.

Induction Step: For $t \in [2, k]$, we assume that $\xi_h(s_j) \geq 0$ for each $j \in [0, t-1]$. Since $\delta_h(s_t) + \cdots + \delta_m(s_t) \geq \delta_h(s_{t-1}) + \cdots + \delta_m(s_{t-1})$ and $\delta_h^*(s_t) \leq \delta_h^*(s_{t-1}) + 1$, we have that if $\xi_h(s_{t-1}) \geq 1$, then

$$\begin{aligned}
\xi_h(s_t) &= \delta_h(s_t) + \cdots + \delta_m(s_t) - \delta_h^*(s_t) \\
&\geq \delta_h(s_{t-1}) + \cdots + \delta_m(s_{t-1}) - \{\delta_h^*(s_{t-1}) + 1\}
\end{aligned}$$

$$
\begin{aligned}
&= \delta_h(s_{t-1}) + \cdots + \delta_m(s_{t-1}) - \delta_h^*(s_{t-1}) - 1 \\
&= \xi_h(s_{t-1}) - 1 \geq 0.
\end{aligned}
$$

Assume that $\xi_h(s_{t-1}) = \delta_h(s_{t-1}) + \cdots + \delta_m(s_{t-1}) - \delta_h^*(s_{t-1}) = 0$. If $\delta_h^*(s_{t-1}) = 0$, then $\delta_h(s_{t-1}) = \cdots = \delta_m(s_{t-1}) = 0$. From the definition of Greedy, it follows that for each $\ell \in [h, m]$, no $v_\ell$-packets arrive until send event $s_{t-1}$, which implies that $q_h^*(s_{t-1}) = 0$. So from Claim 3.2 and the induction hypothesis, it follows that $\xi_h(s_t) \geq \xi_h(s_{t-1}) \geq 0$.

Assume that $\delta_h^*(s_{t-1}) = n > 0$ and we consider the following cases: (3) Greedy does not reject any $v_h$-packet that arrives until send event $s_{t-1}$; (4) Greedy rejects $v_h$-packets that arrive until send event $s_{t-1}$. For the case (3), let $n_h$ be the number of $v_h$-packets that arrive until send event $s_{t-1}$. It is obvious that $n_h \geq \delta_h^*(s_{t-1}) = n > 0$. If $q_h(s_{t-1}) > 0$, then from Claim 3.1 and the induction hypothesis, it follows that $\xi_h(s_t) \geq \xi_h(s_{t-1}) \geq 0$. Assume that $q_h(s_{t-1}) = 0$. Since $n_h > 0$ $v_h$-packets arrive until send event $s_{t-1}$, $q_h(s_{t-1}) = 0$, and Greedy does not reject any $v_h$-packet that arrives until send event $s_{t-1}$, we have that $\delta_h(s_{t-1}) = n_h$. If $\delta_h^*(s_{t-1}) < n_h$, then $\delta_h^*(s_{t-1}) < n_h \leq \delta_h(s_{t-1}) + \cdots + \delta_m(s_{t-1})$, which contradiction the assumption that $\delta_h(s_{t-1}) + \cdots + \delta_m(s_{t-1}) - \delta_h^*(s_{t-1}) = 0$. So we assume that $\delta_h^*(s_{t-1}) = n_h$. From Eq. (2) and the fact that $n_h \geq A_h^*(s_{t-1})$, it is immediate that $q_h^*(s_{t-1}) = A_h^*(s_{t-1}) - \delta_h^*(s_{t-1}) \leq n_h - n_h = 0$, i.e., $q_h^*(s_{t-1}) = 0$. So from Claim 3.2 and the induction hypothesis, it follows that $\xi_h(s_t) \geq \xi_h(s_{t-1}) \geq 0$.

For the case (4), consider the following subcases: (4.1) $q_h(s_{t-1}) > 0$; (4.2) $q_h(s_{t-1}) = 0$. For the subcase (4.1), it is obvious that $q_h(s_{t-1}) + \cdots + q_m(s_{t-1}) > 0$. From Claim 3.1 and the induction hypothesis,we have that $\xi_h(s_t) \geq \xi_h(s_{t-1}) \geq 0$. For the subcase (4.2), let $e_\tau$ be the last arrive event at which a $v_h$-packet is rejected by Greedy. Assume that $e_\tau \in \Sigma_j$ for some $j \in [1, t-1]$, i.e., $e_\tau$ is arrive event between send events $s_{j-1}$ and $s_j$. Since $v_h$-queue of Greedy is *full* just before arrive event $e_\tau$, $v_h$-queue of Greedy is *full* just before send event $s_j$. Let $L_h \geq 0$ be the total number of $v_h$-packets that arrive between send events $s_j$ and $s_{t-1}$. Since $q_h(s_{t-1}) = 0$, Greedy must transmit $B_h + L_h$ $v_h$-packets from send event $s_j$ to send event $s_{t-1}$. Thus $\delta_h(s_{t-1}) + \cdots + \delta_m(s_{t-1}) \geq \delta_h(s_{j-1}) + \cdots + \delta_m(s_{j-1}) + B_h + L_h$. Assume that Opt transmits $K_h \geq 0$ $v_h$-packets at send events $s_j, \ldots, s_{t-1}$, i.e., $\delta_h^*(s_{t-1}) = \delta_h^*(s_{j-1}) + K_h$. From the induction hypothesis that $\xi_h(s_{j-1}) \geq 0$, it follows that

$$
\begin{aligned}
\xi_h(s_{t-1}) &= \delta_h(s_{t-1}) + \cdots + \delta_m(s_{t-1}) - \delta_h^*(s_{t-1}) \\
&\geq \{\delta_h(s_{j-1}) + \ldots + \delta_m(s_{j-1}) + B_h + L_h\} \\
&\qquad - \{\delta_h^*(s_{j-1}) + K_h\} \\
&= \delta_h(s_{j-1}) + \cdots + \delta_m(s_{j-1}) \\
&\qquad - \delta_h^*(s_{j-1}) + B_h + L_h - K_h \\
&= \xi_h(s_{j-1}) + B_h + L_h - K_h \\
&\geq B_h + L_h - K_h.
\end{aligned}
$$

Note that $K_h \leq B_h + L_h$. If $K_h < B_h + L_h$, then it is immediate that $\xi_h(s_{t-1}) > 0$, which contradicts the assumption that $\xi_h(s_{t-1}) = 0$. So we have $K_h = B_h + L_h$, which implies that $q_h^*(s_{t-1}) = 0$. Thus from Claim 3.2 and the induction hypothesis, it follows that $\xi_h(s_t) \geq \xi_h(s_{t-1}) = 0$. ■

## 3.2 Number of Accepted Packets

In this subsection, we investigate the relationships between the number of packets accepted by Greedy and the number of packets accepted by Opt. In the rest of this paper, we use $A_h$ and $A_h^*$ instead of $A_h(\sigma)$ and $A_h^*(\sigma)$ respectively, when $\sigma$ is clear from the context. For each $h \in [1, m]$, let $D_h = A_h^* - A_h$ and $S_h = A_h + A_{h+1} + \cdots + A_m$.

The following lemma shows the relationship between the number of $v_m$-packets accepted by Greedy and the number of $v_m$-packets accepted by Opt, which is a straightforward generalization of the result due to Al-Bawani and Souza [2, Lemma 2.5].

**Lemma 3.2:** $A_m = A_m^*$.

**Proof:** By definition of Greedy, $v_m$-packet has priority at send event. Thus at any event $e_i$, the number of $v_m$-packets transmitted by Greedy is maximum, i.e., $A_m(e_i) \geq A_m^*(e_i)$. Assume that at arrive event $e_t$, $A_m(e_t)$ becomes greater than $A_m^*(e_t)$ for the first time, which implies that at arrive event $e_t$, Opt rejects a $v_m$-packet but Greedy accepts a $v_m$-packet. Thus just before event $e_t$, $v_m$-queue of Opt is full but $v_m$-queue of Greedy has at least one vacancy. Since $A_m(e_{t-1}) = A_m^*(e_{t-1})$, there must exist send event $e_\tau$ (with $\tau \leq t-1$) at which Opt transmitted a $v_\ell$-packet with $\ell \in [1, m-1]$, while the $v_m$-queue of Opt was not empty. Change the behavior of Opt at send event $e_\tau$ by transmitting a $v_m$-packet instead of the $v_\ell$-packet. This yields an increase in the benefit of Opt and the $v_m$-packet rejected at arrive event $e_t$ can be accepted. ■

The following lemma is an extension of the result by Al-Bawani and Souza [2, Lemma 2.6] and plays a crucial role in the subsequent discussions.

**Lemma 3.3:** *For each* $h \in [1, m-1]$,

$$D_h = A_h^* - A_h \leq \sum_{\ell=h+1}^{m} A_\ell = S_{h+1}.$$

**Proof:** Let $\varphi_h(e_i) = A_h(e_i) + \cdots + A_m(e_i) - A_h^*(e_i)$. From Eqs. (1) and (2), we have that

$$\varphi_h(e_i) = \sum_{\ell=h}^{m} \{\delta_\ell(e_i) + q_\ell(e_i)\} - \{\delta_h^*(e_i) + q_h^*(e_i)\}.$$

By induction on events $e_i$ for $i \geq 0$, we show that $\varphi_h(e_i) \geq 0$.

Base Step: For the initial event $e_0$, it is immediate that $\delta_h(e_0) = \cdots = \delta_m(e_0) = 0$, $q_h(e_0) = \cdots = q_m(e_0) = 0$, $\delta_h^*(e_0) = 0$, and $q_h^*(e_0) = 0$. This implies that $\varphi_h(e_0) = 0$.

Induction Step: For $t \geq 1$, assume that $\varphi_h(e_i) \geq 0$ for each $i \in [0, t-1]$. Consider the case that $e_t$ is send event and the case that $e_t$ is arrive event.

($e_t$: send event) If Opt transmits a $v_h$-packet, then $\delta_h^*(e_t) + q_h^*(e_t) = \delta_h^*(e_{t-1}) + 1 + q_h^*(e_{t-1}) - 1 = \delta_h^*(e_{t-1}) + q_h^*(e_{t-1})$. It is obvious that $\delta_h^*(e_t) + q_h^*(e_t) = \delta_h^*(e_{t-1}) + q_h^*(e_{t-1})$ if Opt does not transmits a $v_h$-packet. For the case that Greedy transmits a $v_r$-packet with $r \in [h, m]$, it is immediate that $\delta_r(e_t) + q_r(e_t) = \delta_r(e_{t-1}) + 1 + q_r(e_{t-1}) - 1 = \delta_r(e_{t-1}) + q_r(e_{t-1})$ and that $\delta_\ell(e_t) + q_\ell(e_t) = \delta_\ell(e_{t-1}) + q_\ell(e_{t-1})$ for each $\ell \in [h, m] \setminus \{r\}$. For the case that Greedy transmits a $v_r$-packet with $r \in [1, h-1]$, it is easy to see that $\delta_\ell(e_t) + q_\ell(e_t) = \delta_\ell(e_{t-1}) + q_\ell(e_{t-1})$ for each $\ell \in [h, m]$. Then from the induction hypothesis, we have that

$$\begin{aligned}
\varphi_h(e_t) &= \sum_{\ell=h}^{m} \{\delta_\ell(e_t) + q_\ell(e_t)\} - \{\delta_h^*(e_t) + q_h(e_t)\} \\
&= \sum_{\ell=h}^{m} \{\delta_\ell(e_{t-1}) + q_\ell(e_{t-1})\} \\
&\qquad - \{\delta_h^*(e_{t-1}) + q_h(e_{t-1})\} \\
&= \varphi_h(e_{t-1}) \geq 0.
\end{aligned}$$

($e_t$: arrive event) It is easy to see that $\delta_h(e_t) = \delta_h(e_{t-1}), \ldots, \delta_m(e_t) = \delta_m(e_{t-1})$ and $\delta_h^*(e_t) = \delta_h^*(e_{t-1})$. Consider the following cases: (1) a $v_r$-packet with $r \in [1, h-1]$ arrives; (2) a $v_r$-packet with $r \in [h+1, m]$ arrives; (3) a $v_h$-packet arrives.

For the case (1), it is immediate that $q_h(e_t) = q_h(e_{t-1}), \ldots, q_m(e_t) = q_m(e_{t-1})$ and $q_h^*(e_t) = q_h^*(e_{t-1})$. From the induction hypothesis, it follows that $\varphi_h(e_t) = \varphi_h(e_{t-1}) \geq 0$. For the case (2), we have that $q_r(e_t) \geq q_r(e_{t-1})$, $q_\ell(e_t) = q_\ell(e_{t-1})$ for each $\ell \in [h, m] \setminus \{r\}$, and $q_h^*(e_t) = q_h^*(e_{t-1})$. Thus from the induction hypothesis, it follows that $\varphi_h(e_t) \geq \varphi_h(e_{t-1}) \geq 0$. For the case (3), let us consider the following subcases: (3.1) Greedy and Opt accept the $v_h$-packet, (3.2) Greedy and Opt reject the $v_h$-packet, (3.3) Greedy accepts the $v_h$-packet but Opt rejects the $v_h$-packet, (3.4) Greedy rejects the $v_h$-packet but Opt accepts the $v_h$-packet. For the subcase (3.1), it is immediate that $q_h(e_t) = q_h(e_{t-1})+1$, $q_\ell(e_t) = q_\ell(e_{t-1})$ for each $\ell \in [h+1, m]$, and $q_h^*(e_t) = q_h^*(e_{t-1}) + 1$. From the induction hypothesis, it follows that $\varphi_h(e_t) = \varphi_h(e_{t-1}) \geq 0$. For the subcase (3.2), we can show that $\varphi_h(e_t) = \varphi_h(e_{t-1}) \geq 0$ in a way similar to the subcase (3.1). For the subcase (3.3), we have that $q_h(e_t) = q_h(e_{t-1})+1$, $q_\ell(e_t) = q_\ell(e_{t-1})$ for each $\ell \in [h+1, m]$, and $q_h^*(e_t) = q_h^*(e_{t-1})$. From the induction hypothesis, it follows that $\varphi_h(e_t) = \varphi_h(e_{t-1}) + 1 \geq 0$. For the subcase (3.4), we have that the $v_h$-queue of Greedy is full, i.e., $q_h(e_t) = B_h$. From the fact that $B_h \geq q_h^*(e_t)$, it is obvious that $q_h^*(e_t) \leq q_h(e_t) \leq q_h(e_t) + \cdots + q_m(e_t)$. So from Lemma 3.1 and the definition of $\varphi_h$, we have that $\varphi_h(e_t) \geq 0$. ■

# 4 Analysis of Greedy

From Lemmas 2.1 and 3.2,. it follows that

$$\sum_{\ell=h}^{m-1} D_\ell = \sum_{\ell=h}^{m} D_\ell \leq \sum_{\ell=h}^{m} A_\ell = S_h, \tag{3}$$

for each $h \in [1, m-2]$. Then for each $h \in [1, m-1]$, we derive the $m-h$ upper bounds for $D_h + D_{h+1} + \cdots + D_{m-1}$ by applying Eq. (3) and Lemma 3.3. For each $j \in [h, m-3]$, apply Lemma 3.3 to $D_h, D_{h+1}, \ldots, D_j$ and apply Eq. (3) to $D_{j+1} + D_{j+2} + \cdots + D_{m-1}$, i.e.,

$$\begin{aligned}
&D_h + D_{h+1} + \cdots + D_{m-1} \leq S_{h+1} + S_{h+1}; \\
&D_h + D_{h+1} + \cdots + D_{m-1} \\
&\qquad \leq S_{h+1} + S_{h+2} + S_{h+2}; \\
&\qquad \vdots \\
&D_h + D_{h+1} + \cdots + D_{m-1} \\
&\qquad \leq S_{h+1} + S_{h+2} + \cdots + S_{j+1} + S_{j+1};
\end{aligned}$$

$$\vdots$$

$$D_h + D_{h+1} + \cdots + D_{m-1} \leq S_{h+1} + S_{h+2} + \cdots + S_{m-2} + S_{m-2}.$$

Applying Lemma 3.3 to $D_h, D_{h+1}, \ldots, D_{m-1}$, we have that $D_h+D_{h+1}+\cdots+D_{m-1} \leq S_{h+1}+S_{h+2}+\cdots+S_m$, and applying Eq. (3) to $D_h+D_{h+1}+\cdots+D_{m-1}$, we also have that $D_h+D_{h+1}+\cdots+D_{m-1} \leq S_h$. Let $U_h$ be the minimum among $m-h$ upper bounds for $D_h+D_{h+1}+\cdots+D_{m-1}$. From the definition of $U_h$, it is immediate that $U_{m-1} = A_m$. For $m$ nonnegative packet values $0 < v_1 < v_2 < \cdots < v_m$, let $c_m^* = \max\{c_1, c_2, \ldots, c_{m-1}\}$. where

$$c_i = \frac{v_i + \sum_{j=1}^{i-1} 2^{j-1} v_{i-j}}{v_{i+1} + \sum_{j=1}^{i-1} 2^{j-1} v_{i-j}},$$

for each $i \in [1, m-1]$. Note that $c_m^* < 1$. The following lemmas hold for $c_m^*$ and $U_h$ (for the proofs of these lemmas, see [15]).

**Lemma 4.1:** *For each $i \in [1, m-1]$,*

$$v_i + \sum_{j=1}^{i-1} 2^{j-1} v_{i-j} - c_m^* \left( v_{i+1} + \sum_{j=1}^{i-1} 2^{j-1} v_{i-j} \right) \leq 0.$$

**Lemma 4.2:** *For each $h \in [1, m-2]$, $U_h = \min\{A_h, U_{h+1}\} + S_{h+1}$, where $U_{m-1} = A_m$.*

For each $h \in [1, m-2]$, define $\Delta_h$ as follows:

$$\begin{aligned}
\Delta_h = & \left\{ v_h + \sum_{j=1}^{h-2} 2^{j-1} v_{h-1-j} \right. \\
& \left. - c_m^* \left( v_{h-1} + \sum_{j=1}^{h-2} 2^{j-1} v_{h-1-j} \right) \right\} U_h \\
& + \left( v_{h-1} + \sum_{j=1}^{h-2} 2^{j-1} v_{h-1-j} \right. \\
& \left. - c_m^* \sum_{j=1}^{h-2} 2^{j-1} v_{h-1-j} \right) S_h \\
& + (v_{h+1} - v_h) U_{h+1} + (v_{h+2} - v_{h+1}) U_{h+2} \\
& + \cdots + (v_{m-1} - v_{m-2}) U_{m-1}.
\end{aligned}$$

The following lemmas are crucial to analyze the competitive ratio of the algorithm Greedy (for the proofs of these lemmas, see [15]).

**Lemma 4.3:** *For each $h \in [1, m-3]$, it holds that $\Delta_h \leq c_m^* v_h A_h + \Delta_{h+1}$.*

**Lemma 4.4:** *For $\Delta_{m-2}$, it holds that $\Delta_{m-2} \leq c_m^* v_{m-2} A_{m-2} + c_m^* v_{m-1} A_{m-1} + c_m^* v_m A_m$.*

From Lemmas 4.2, 4.3, and 4.4, we can show the following theorem:

**Theorem 4.1:** *For the general $m$-valued case with class segregation,* Greedy *is $(1 + c_m^*)$-competitive.*

**Proof:** For any sequence $\sigma$, it is immediate that

$$\begin{aligned}
\frac{\mathsf{Opt}(\sigma)}{\mathsf{Greedy}(\sigma)} &= \frac{v_1 A_1^* + v_2 A_2^* + \cdots + v_m A_m^*}{v_1 A_1 + v_2 A_2 + \cdots + v_m A_m} \\
&= 1 + \frac{\sum_{i=1}^{m} v_i (A_i^* - A_i)}{v_1 A_1 + v_2 A_2 + \cdots + v_m A_m} \\
&= 1 + \frac{v_1 D_1 + v_2 D_2 + \cdots + v_{m-1} D_{m-1}}{v_1 A_1 + v_2 A_2 + \cdots + v_m A_m},
\end{aligned}$$

where the last equality follows from Lemma 3.2. We bound $v_1 D_1 + v_2 D_2 + \cdots + v_{m-1} D_{m-1}$.

$$\begin{aligned}
& v_1 D_1 + v_2 D_2 + \cdots + v_{m-1} D_{m-1} \\
&= v_1 (D_1 + D_2 + \cdots + D_{m-1}) \\
&\quad + (v_2 - v_1)(D_2 + D_3 + \cdots + D_{m-1}) \\
&\quad + (v_3 - v_2)(D_3 + D_4 + \cdots + D_{m-1}) \\
&\quad + \cdots + (v_{m-1} - v_{m-2}) D_{m-1} \\
&\leq v_1 U_1 + (v_2 - v_1) U_2 + (v_3 - v_2) U_3 \\
&\quad + \cdots + (v_{m-1} - v_{m-2}) U_{m-1} \\
&= \Delta_1,
\end{aligned}$$

where the inequqlity follows from $D_h + D_{h+1} + \cdots + D_{m-1} \leq U_h$ for each $h \in [1, m-1]$, and the last equality follows from the definition of $\Delta_h$. By the iterative use of Lemma 4.3, we have that

$$\begin{aligned}
\frac{\mathsf{Opt}(\sigma)}{\mathsf{Greedy}(\sigma)} &\leq 1 + \frac{\Delta_1}{v_1 A_1 + v_2 A_2 + \cdots + v_m A_m} \\
&\leq 1 + \frac{c_m^* v_1 A_1 + \Delta_2}{v_1 A_1 + v_2 A_2 + \cdots + v_m A_m} \\
&\leq 1 + \frac{c_m^* v_1 A_1 + c_m^* v_2 A_2 + \Delta_3}{v_1 A_1 + v_2 A_2 + \cdots + v_m A_m} \\
&\ \vdots \\
&\leq 1 + \frac{c_m^* v_1 A_1 + \cdots + c_m^* v_{m-3} A_{m-3} + \Delta_{m-2}}{v_1 A_1 + v_2 A_2 + \cdots + v_m A_m} \\
&\leq 1 + \frac{c_m^* v_1 A_1 + c_m^* v_2 A_2 + \cdots + c_m^* v_m A_m}{v_1 A_1 + v_2 A_2 + \cdots + v_m A_m} \\
&= 1 + c_m^* \cdot \frac{v_1 A_1 + v_2 A_2 + \cdots + v_m A_m}{v_1 A_1 + v_2 A_2 + \cdots + v_m A_m} \\
&= 1 + c_m^*,
\end{aligned}$$

where all inequalities but the first and last ones are due to Lemma 4.3 and the last inequality is due to Lemma 4.4. So Greedy is $(1 + c^*_m)$-competitive. ■

## References

[1] W.A. Aiello, Y. Mansour, S. Rajagopolan, and A. Rosen. Competitive Queue Policies for Differential Services. In *Proc. of IEEE INFOCOM*, pp.431–440 (2000).

[2] L. Al-Bawani and A. Souza. Competitive Buffer Management with Class Segregation. CoRR abs/1103.6049: (2011)

[3] S. Albers and M. Schmidt. On the Performance of Greedy Algorithms in Packet Buffering. In *Proc. of the 36th Annual ACM Symposium on Theory of Computing* pp.35–44 (2004).

[4] Y. Azar. Online packet Switching. In *Proc. of the 2nd Workshop on Approximation and Online Algorithms*, Lecture Notes in Computer Science 3351, pp.1-5 (2004).

[5] Y. Azar and A. Litichevskey. Maximizing Throughput in Multi-Queue Switches. In *Proc. of the 12th European Symposium on Algorithms*, Lecture Notes in Computer Science 3221, pp.53-64 (2004).

[6] Y. Azar and Y. Richter. Management of Multi-Queue Switches in QoS Networks. In *Proc. of the 35th Annual ACM Symposium on Theory of Computing*, pp.82–89 (2003).

[7] Y. Azar and Y. Richter. The Zero-One Principle for Switching Networks. In *Proc. of the 36th Annual ACM Symposium on Theory of Computing*, pp.64–71 (2004).

[8] A. Borodin and R. El-Yaniv. Online Computation and Competitive Analysis. Cambridge University Press (1998).

[9] M. Bienkowski and A. Mardy. Geometric Aspects of Online Packet Buffering: An Optimal Randomized Algorithm for Twp Buffers. In *Proc. of the 8th Latin American Symposium on Theoretical Informatics*, Lecture Notes in Computer Science ,4857 pp.252-263 (2008).

[10] M. Chrobak. Online Algorithms Column 13. *ACM SIGACT News*, Volume 39, Issue 3, pp.96-121 (2008).

[11] Cisco Systems, Inc. Quality of Service (QoS). Internetworking Technology Handbook, Chapter 49 (2006).

[12] L. Epstein and R. van Stee. Buffer Management Problems. *ACM SIGACT News*, Volume 35, Issue 3, pp.58-66 (2004).

[13] M.H. Goldwasser. A Survey of Buffer Management Policies for Packet Switches. *ACM SIGACT News*, Volume 41, Issue 1, pp.100-128 (2010).

[14] T. Itoh and N. Takahashi. Competitive Analysis of Multi-Queue Preemptive QoS Algorithms for General Priorities. *IEICE Trans. on Fundamentals of Electronics, Communications, and Computer Science*, Vo.E89-A, No.5, pp.1186-1197 (2006).

[15] T. Itoh and S. Yoshimoto. Greedy Algorithms for Multi-Queue Buffer Management with Class Segregation. CoRR abs/1109.6060: (2011)

[16] W. Jawor. Three Dozen Papers on Online Algorithmms. *ACM SIGACT News*, Volume 36, Issue 1, pp.71-85 (2004).

[17] A. Kesselman, Z. Lotker, Y. Mansour, B. Patt-Shamir, B. Schieber, and M. Sviridenko. Buffer Overflow Management in QoS Switches. In *Proc. of the 33rd Annual ACM Symposium on Theory of Computing*, pp.520–529 (2001).

[18] G. Li and H. Liu. Dynamic Resource Allocation with Finite Buffer Constraints in Broadband OFDMA Networks. In Proc. of *IEEE Wireless Communications and Networking Conference*, pp.1037-1042 (2003).

[19] Y. Mansour, B. Patt-Shamir, and O. Lapid. Optimal Smoothing Schedules for Real-Time Streams. In *Proc. of of the 19th ACM Symposium on Principles of Distributed Computing*, pp.21.29 (2000).