COMPARISON OF SERVICE AND BUFFER OVERFLOW POLICIES FOR MULTIPLE SERVER QUEUES THAT SERVE CUSTOMERS WITH DEADLINES

D. Towsley and S. Panwar

COINS Technical Report 89-72

July 1989

this fraction and that the longest time to extinction policy (LTE) minimizes this fraction for the continuous and discrete time non-preemptive, G/M/c/K queues when customers must either begin service by their deadline or complete service by

[†] This work was supported by the National Science Foundation under Grant NSF ECS 83-10771, by the Office of Naval Research under Contract N0014-87-K-0796, and by the Center for Advanced Technology in Telecommunications, Polytechnic University, Brooklyn, NY.

their deadline. For the preemptive G/M/c/K queue with deadlines until the end of service, STE maximizes the fraction of customers completing by their deadlines out of the class of service independent policies and LTE minimizes the fraction completing by their deadlines out of the class of non-idling, service independent policies. In all of these systems, STE is the policy that schedules customers closest to their deadlines and, in the case of buffer overflow removes the customer closest to its deadline. LTE, on the other hand, schedules and removes customers farthest from their deadlines. We further show in the case of the nonpreemptive G/M/c/K queue where deadlines are to the end of service, that the best service time independent policies (i.e., those that provide low loss probabilities) belong to the class of shortest time to extinction with inserted idle time (STEI) policies. This result is shown to be true for the G/GI/c/K queue in the case that deadlines are to the beginning of service. Here an STEI policy requires that the customer closest to his deadline be scheduled whenever a customer is scheduled. An STEI policy also has the choice of inserting idle times while the queue is nonempty. Last, some results are given for systems where servers take vacations.

1 Introduction

Increasing interest has been shown recently in the design and analysis of real-time multiprocessor systems. The workloads served by these systems consist of customers that have real-time constraints, i.e., customers must complete or enter service by specified deadlines. For some systems it is unacceptable for any task to miss its deadline. In these systems task service demands are usually well understood and a substantial literature has focussed on the development and evaluation of scheduling policies for these workloads, [14,15]. Other workloads consist of tasks for which it is not critical that all tasks meet their constraints. Usually, the service requirements and the arrival patterns are not as well understood and the objective is to design policies that will minimize the fraction of tasks that miss their deadlines. The purpose of this paper is to study optimal policies for this second class of workloads.

In this paper we consider as our model for a multiprocessor, a multiple server queue with either finite or infinite capacity, that serves customers with deadlines. We study the effect of different service and buffer overflow policies on the fraction of customers which successfully complete service, i.e., do not miss their deadlines. We show that, out of the class of service-time independent policies, the *shortest time to extinction* policy (STE) maximizes and that out of the same class of policies that are also *non-idling* the *longest time to extinction* policy (LTE) minimizes the fraction of customers that successfully complete service by their deadlines for the preemptive continuous/discrete time G/M/c/K queue when customer deadlines are to the end of service. Here STE is the non-idling policy that schedules the customer closest to his deadline and, in the case of buffer overflow, removes the customer closest to its deadline. LTE schedules or removes customers farthest from their deadlines.

When we restrict ourselves to systems that do not allow preemptions, we have two kinds of results. First, out of the class of non-idling service time independent policies, STE and LTE respectively maximize and minimize the fraction of customers making their deadlines for the continuous/discrete time G/M/c/K queue. This result holds for systems where deadlines are to the beginning of service or to the end of service. Second, out of the class of service time independent policies, the best policies lie in the class of *shortest time to extinction with inserted idle times* (STEI) policies. This holds true for the G/M/c/K queue when deadlines are until the end of service and for the G/GI/c/K queue when deadlines are to the beginning of service. preemptions. Whenever the queue is not empty, an STEI policy either schedules no customer or schedules the customer closest to its deadline. Buffer overflow is handled by these policies by removing customers closest to their deadlines.

Last, some additional results are presented for preemptive systems. These include 1) an extension of the above results to systems where servers take vacations, 2) STE is the optimum STEI policy for the G/G/c queue, i.e., STE maximizes the fraction of completions before deadlines, and 3) the policy that maximizes the fraction of customers that complete by their deadlines in the G/G/c/K queue belongs to the class of non-idling policies.

The shortest time to extinction (STE) policy, which will be described in Section 2, is very similar to the earliest due date (EDD) scheduling policy proposed by Jackson [12]. Consider a set of n tasks $\{T_i, 1 \leq i \leq n\}$ with the corresponding n due dates $\{d_i, 1 \leq i \leq n\}$. Let the finishing times under schedule S be $f_i(S)$. Then the lateness of T is defined as $f_i(S) - d_i$ and the tardiness is defined as $max\{0, f_i(S) - d_i\}$. Jackson showed that the maximum lateness and maximum tardiness are minimized by sequencing the tasks in the order of non-decreasing due dates. As we shall see in Section 3, STE scheduling differs from EDD scheduling in that it never schedules tasks which are already past their due dates. Note that the tasks and their due dates are known a priori under Jackson's model. Similar problems, for models other than queueing systems, have also been studied in [8,16,19,20,23]. In the packet-switching context, variations of the EDD policy for queueing models have been studied in [13,4]. Doshi and Lipper consider optimal service disciplines for queues with delay dependent customer behavior [9]. In queueing theory literature, queues with impatient customers have been usually analyzed assuming a FCFS scheduling policy [1,6,10]. An analysis of STE for the preemptive M/M/1 queue (deadlines to the end of service) and the nonpreemptive M/M/c queue (deadlines to the beginning of service) can be found in [11].

In [18], we have considered the problem of a single server queue with impatient customers under the assumption that deadlines are until customers enter service. We showed that STE is optimal for a large class of infinite capacity single server queues. The shortest time to extinction with unforced idle times (STEI) class of policies are shown to be optimal for a larger class of queues. Similar results for the continuous time single server infinite capacity queue when the deadlines are to the end of service can be found in [17,2]. Our results generalize these previous results in several ways. First, our results are for multiple server queues, and in some cases, servers are allowed to take vacations. Second, in the case, of finite capacity queues, they include the effect of buffer overflow policies. Last, the results in [17,2] are based on interchange arguments which obscure their physical interpretation. The proofs in this paper are based on defining a *state* of the system based on the set of extinction times of the customers in the system and using a forward induction argument to establish dominance of one state over another when operating under different policies. This method provides a clearer understanding of the differences between STE, LTE, and non-STE policies. In addition, we are able to establish orderings among buffer occupancies between STE, LTE, and non-STE policies (Section 6). This approach has also been used to develop bounds on the performance of these policies, (see [11]).

This paper is organized as follows. Section 2 contains a model of the system under study along with definitions of the different scheduling policies of interest to us. The main results of the paper are contained in sections 3, 4, and 5. Section 3 contains the results for systems with deadlines to the end of service that allow preemptions, section 4 contains results for systems for systems with deadlines to the beginning of service, and section 5 contains results for systems with deadlines to the end of service without preemptions. Section 6 provides some extensions of these ordering results to buffer occupancies. We summarize our results in Section 7.

2 Definitions and Notation

We consider three different multiple server queues,

- Preemptive queues with deadlines to the end of service.
- Nonpreemptive queues with deadlines to the end of service where a customer that misses its deadline while in service is aborted,
- Nonpreemptive queues with deadlines to the beginning of service,

We assume that the queue has a capacity for K customers. In all of these systems let T_i denote the arrival time of the i-th customer and A_i denote the time between the arrivals of the $(i-1)$-th and i-th customers. We assume that A_i is a random variable with arbitrary distribution. Let E_i denote the extinction time of the i-th customer (i.e., the time by which it must be served). Here $E_i = T_i + D_i$ where D_i is a random variable with a general distribution. We shall refer to D_i as the real time constraint or the relative deadline for customer i. Last, let $\{B_i\}_{1\leq i}$ be an independent and identically distributed (i.i.d.) sequence of random variables with a general distribution which will be used to assign service times to customers.

We shall use the notation $\boldsymbol{A}_N = \{A_i\}_{1\le i\le N}$, $\boldsymbol{D}_N = \{D_i\}_{1\le i\le N}$, $\boldsymbol{B}_N = \{B_i\}_{1\le i\le N}$, and $\boldsymbol{S}_N = (\boldsymbol{A}_N, \boldsymbol{D}_N, \boldsymbol{B}_N), 1 \le N$. In addition, whenever we focus on a specific sample realization of the above r.v.'s, we shall use lowercase notation (i.e., a_i for A_i, etc ...). Furthermore, we shall let $\boldsymbol{a} = \{a_i\}_{1\le i}$, $\boldsymbol{b} = \{b_i\}_{1\le i}$, $\boldsymbol{d} = \{d_i\}_{1\le i}$, $\boldsymbol{a}_N = \{a_i\}_{1\le i\le N}$, $\boldsymbol{b}_N = \{b_i\}_{1\le i\le N}$, and $\boldsymbol{d}_N = \{d_i\}_{1\le i\le N}$. Last, let $\boldsymbol{s} = (\boldsymbol{a}, \boldsymbol{d}, \boldsymbol{b})$ and $\boldsymbol{s}_N = (\boldsymbol{a}_N, \boldsymbol{d}_N, \boldsymbol{b}_N)$, $N = 1, \ldots$. These last two quantities will be referred to as an input sample and finite input sample respectively.

At this point in the paper we will not specify how service times from the sequence $\{B_i\}$ are assigned to customers. The assignment rule will depend on which system we are interested in and on what property we wish to prove with regard to that system. We use the notation A/C/D/E+ F to denote a queue with customer deadlines where A, C, D, and E have the same meaning as in Kendall's notation while F describes the distribution of the relative deadlines. Last, we make the assumption that $\{B_i | 1 \le i\}$ is independent of $\{A_i\}$ and $\{D_i\}$.

Let π be a policy that determines what customer in the queue is to be executed (if any) whenever the server is free. This policy makes its decision based on the customers that are eligible for service as well as on the past history of the system. We wish to choose π so that we maximize the fraction of customers beginning service before their respective extinction times. Consider a system in which exactly N customers arrive for service. We define $V_N(\pi)$ to be the number of customers served for this system. We are interested in the fraction, $\overline{V}_N(\pi) = E[V_N(\pi)]/N$, of customers served in this system. We define the fraction of customers served in the system as $N \longrightarrow \infty$ (under policy π) to be

$$\overline{V}(\pi) = \liminf_{N\to\infty} \overline{V}_N(\pi).$$

Finally, let $\overline{V} = \sup_\pi \overline{V}(\pi)$.

Let $C_\pi(t) = \{c_{j_1}, c_{j_2}, \cdots, c_{j_n}\}$ denote the set of customers in the queue at time t and $\mathcal{R}_\pi(t) = \{c_{j_1}, c_{j_2}, \cdots, c_{j_n}, c_{j_{n+1}}, \cdots, c_{j_m}\}$ denote the set of all customers in the system at time $t, j_i \ge 1$, $1 \le i \le m$. Here c_i denotes the i-th customer to arrive to the system. We denote the sets of extinction times associated with these two sets of customers are denoted by $\boldsymbol{E}_\pi(t)$ and $\boldsymbol{R}_\pi(t)$.

Consider the actions that policy π can take at time t. If all the servers are busy, then π takes no action if preemptions are not allowed. If any server is idle at time t or if preemptions are allowed, then π can either schedule no customer or schedule customers from $C_\pi(t)$. Policy π is allowed to choose one of these actions according to some distribution that depends on $\pi, C_\pi(t)$ and the previous history $\boldsymbol{H}_t$ (to be defined later in this section). We define $p_j(\pi, t, \boldsymbol{H}_t)$ to be the probability that π schedules customer $c_j \in C_\pi(t), j = 1, 2, \cdots$ on an idle server and $p_0(\pi, t, H_t)$ to be the probability that π chooses to schedule no customer.

If π chooses not to schedule a customer at time t and $C_\pi(t) \neq 0$, then it delays making a new scheduling decision by a random amount of time τ with some arbitrary distribution function $F_\tau(x|H_t)$ (τ takes on discrete values in the case of a discrete time queue). The policy does not perform a scheduling decision until either τ time units elapse or an arrival occurs. Without loss of generality, we may impose one last constraint on π, namely, π is prohibited from scheduling two successive idle times on the same server when the queue is nonempty unless they are separated by the arrival of one or more customers.[1]

In the case that π is allowed to preempt customers, we introduce some additional parameters. If π decides to schedule a customer at time t, then $q(\pi, t, \boldsymbol{H}_t)$ is the probability that the customer will not be preempted in the absence of customer arrivals and service completions. The customer is scheduled for preemption with probability $1 - q(\pi, t, \boldsymbol{H}_t)$ and is provided with τ units of service where τ has cumulative distribution function $H_\tau(x|\boldsymbol{H}_t)$. The customer is preempted after τ units of time provided it has not completed by that time and there have been no arrivals or service completions of other customers. If an arrival or a service completion occurs, then π is allowed to reschedule the customer if it so desires.

Last, when a customer arrives to find the queue full, a policy π is required to determine which customer to remove. In this paper we assume that any customer is a candidate for rmoval, whether in service or waiting for service. Let customer c_i arrive at time t. We let $r_j(\pi, t, \boldsymbol{H}_t)$ denote the probability that customer $c_j \in \boldsymbol{R}_\pi(t) \cup \{c_i\}$ is removed.

The history of the system up to time t may be defined by $\boldsymbol{H}_t = (\boldsymbol{a}_t, \boldsymbol{d}_t, \boldsymbol{r}_t, \boldsymbol{f}_t, \boldsymbol{e}_t, \boldsymbol{u}_t, o_t)$ where $\boldsymbol{a}_t$ is an ordered set of arrival times of all customers that arrive prior to $t, \boldsymbol{d}_t$ is an ordered set of relative deadlines corresponding to the customers that arrive prior

[1] Any policy that schedules two successive idle times can be transformed into a policy that does not schedule two successive idle times.

to t, r_t, f_t, e_t, are ordered sets containing the times of all scheduling decisions prior to time t, the identities of the customers and the servers to which they were scheduled respectively. The set o_t the identities of customers that were removed from the system due to buffer overflow along with the times at which the removals ocurred. In addition, u_t is an ordered set of the service times for customers completed prior to time t.

2.1 Scheduling Policies

We now introduce the scheduling policies of interest to us in this paper. Let t'_k denote the time of the k^{th} scheduling decision since time $t = 0$.

Definition 1 *Policy π is the shortest time to extinction policy (STE) if at time $t'_k, (1 \leq k)$, it always schedules the eligible customer with the smallest deadline on any one of the available servers. In addition, the server is always busy as long as eligible customers are available which have not yet been served, i.e $p_0(\pi, t) = 0$ whenever the server is available and $C_\pi(t) \neq \phi$.*

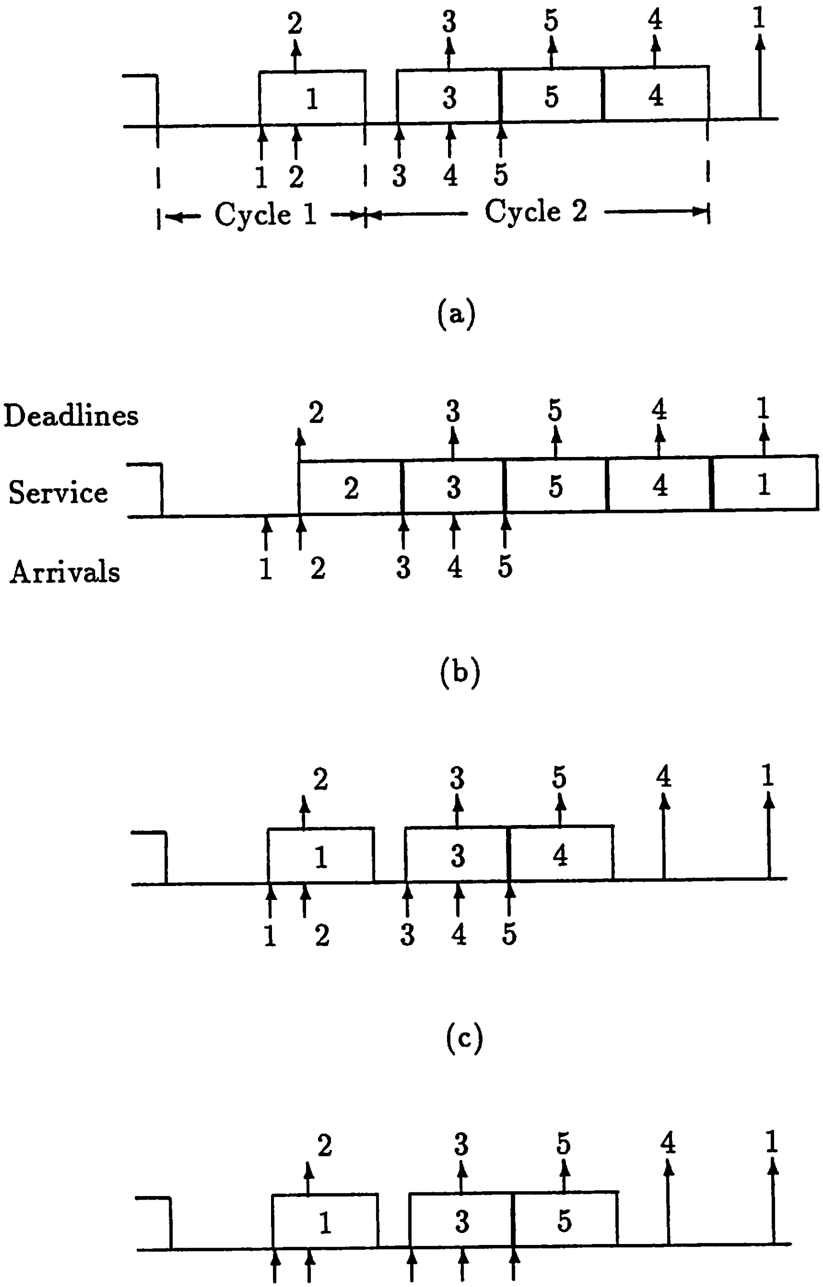
An example of how STE schedules a given set of arrivals is shown in Fig. 1(a) for a single server system when deadlines are to the beginning of service.

Definition 2 *Policy π is a shortest time to extinction with inserted idle times (STEI) policy if, at time t'_k, it schedules the eligible customer with the smallest deadline on any one of the available servers. In other words, $p_0(\pi, t'_k) \geq 0, p_j(\pi, t'_k) \geq 0$ if $j = \arg\min_{j \text{ s.t. } c_j \in C_\pi(t'_k)} E_\pi(t'_k)$ and $p_j(\pi, t'_k) = 0$ otherwise.*

STE is an example of a STEI policy. Fig.1(b) shows how an STEI policy might schedule the same set of arrivals as shown in Fig. 1(a). Note that the STEI policy schedules all the arrivals while the STE policy leads to the loss of one arrival in this particular case. Fig. 1(c) illustrates how a FCFS (first-come, first-served) policy schedules the arrivals.

Last, we define the longest time to extinction policy (LTE).

Definition 3 *Policy π is the longest time to extinction policy (LTE) if at time $t'_k, (1 \leq k)$, it always schedules the eligible customer with the largest deadline on any one of the*



(d)

Figure 1: (a) Behavior of STE, (b) Behavior of STEI, (c) Behavior of FCFS, (d) STEI emulating FCFS

available servers. In addition, the server is always busy as long as eligible customers are available which have not yet been served, i.e $p_0(\pi, t) = 0$ whenever the server is available and $C_\pi(t) \neq \phi$.

2.2 Buffer Overflow Policies

We are concerned with the class of buffer overflow policies that can remove any customer, whether in service or not. Let a customer arrive at time t to find B customers in the system.

Definition 4 *Policy π is the shortest time to extinction (STE) buffer overflow policy if at time t it always removes the customer with the smallest deadline from the system.*

Definition 5 *Policy π is the longest time to extinction (LTE) buffer overflow policy if at time t it always removes the customer with the largest deadline from the system.*

When talking about policies for finite capacity systems, it is necessary to specify both a scheduling policy, π_1, and a buffer overflow policy, π_2. However, in the remainder of the paper, π will refer to the combination of two such policies, STE will refer to the combination of the STE scheduling policy and the STE buffer overflow policy, LTE will refer to the combination of the LTE scheduling policy and the LTE buffer overflow policy, and STEI will refer to the STEI scheduling policies coupled with the STE buffer overflow policy.

3 Preemptive Systems with Deadlines to the End of Service

In this section we show that STE is the best service time independent policy for the preemptive continuous time and discrete time G/M/c+G queue when deadlines are to the end of service. We also show that LTE is the worst non-idling service time independent policy for the G/M/c+G queue. Both of these results are shown to apply to queues where servers take vacations. We conclude the section with a proof that the best policies are non-idling policies for any preemptive system and that STE is the best STEI policy for any preemptive multiple server queue. The basis of our proofs of most of these results and the results for nonpreemptive policies is the comparison of sets of extinction times. We will show that the set of extinction times for eligible customers under STE *dominates* the set of extinction times under any other policy and that these, in the case of non-idling policies, dominate the set of extinction times under LTE. Consequently, we turn our attention to the definition of dominance and the derivation of properties that it satisfies.

Consider two sets of nonnegative real numbers $R = \{x_1, x_2, \cdots, x_n\}$ and $S = \{y_1, y_2, \cdots, y_m\}$ each ordered so that $x_i \geq x_{i+1}$, $i = 1, \cdots n$ and $y_i \geq y_{i+1}$, $i = 1, \cdots m$.

Definition 6 *We say that R dominates S ($R \succ S$) if $n \geq m$ and $x_i \geq y_i$, $i = 1, 2, \cdots m$.*

We define the following three operations

- $Large(R, k) = \{x_1, x_2, \cdots, x_k\}, \;\; 0 \leq k \leq n.$
- $Small(R, k) = \{x_{n-k+1}, \cdots, x_n\}, \;\; 0 \leq k \leq n.$
- $Shift(R, x) = \{x_i - x \mid x_i \geq x\}.$

The following lemma gives conditions under which dominance is preserved when set operations, the *Large* operation, and the *Shift* operation are performed on R and S.

Lemma 1 *If $R \succ S$, then:*

1. *$R + \{x\} \succ S + \{x\}$, for $x > 0$,*
2. *$R - \{x_n\} \succ S$, when $n > m$,*
3. *$R \succ S - \{y\}$, where $y \in S$,*
4. *$R - \{x\} \succ S - \{y\}$, where $x \in R$, $y \in S$, and $x \leq y$,*
5. *Assume that $R = \{x_1, \cdots, x_n\}$ where $x_i \geq x_{i+1}$, $1 \geq i < n$ and $S = \{y_1, \cdots, y_m\}$ where $y_i \geq y_{i+1}$, $1 \leq i < m$. Then $R - \{x_k\} \succ S - \{y_j\}$ for $k \geq j$,*
6. *$Shift(R, x) \succ Shift(S, x)$.*

7. *Large*$(R, |S|) \succ S$.

8. *Assume that R and S can be expressed as* $R = R_1 + R_2$ *and* $S = S_1 + S_2$ *such that* $R_1 \succ S_1$, $|R_2| = n$, $|S_2| = n'$ *with* $n \geq n'$. *Let* $R_2 = (x_1, x_2, \cdots, x_n)$ *and* $S_2 = (y_1, y_2, \cdots, y_{n'})$ *where* $x_i \geq x_{i+1}$ *and* $y_i \geq y_{i+1}$, $i = 1, \cdots n' - 1$, *then* $R - \{x_i\} \succ S - \{y_i\}$ *for* $i = 1, \cdots, n'$.

Proof: The proof of 1, 2, 3, and 6 may be found in [18]. Properties 4, 5 and 7 follow from the operations performed on R and S and the definition of "$\succ$". The proof of 8 is more intricate and is given below.
The proof of 8 is by induction on n, the cardinality of R_2. We first observe that the case $n > n'$ can be reduced to the case $n = n'$ by simply inserting $n - n'$ zero elements into R, S, R_1, and S_2. Thus we assume that $n = n'$.
Basis Step. When $n = 1$ and $S_2 = \emptyset$, the Lemma is trivially true. When $S_2 = \{y_1\}$, then $R - \{x_1\} = R_1 \succ S_1 = S - \{y_1\}$.
Inductive step. Assume that the Lemma is true for $|R_2| \leq n$. We now establish it for $|R_2| = n + 1$.There are three subcases according to the number of elements x_i in R_2 such that $x_i > y_i \in S_2$. If the number is zero, then according to property 4, $R - \{x_i\} \succ S - \{y_i\}$ for $1 \leq i \leq n'$. Consider the case that the number is two or more. Let x_i and y_i be elements such that $x_i > y_i$. Define $R_1' = R_1 + \{x_i\}$, $S_1' = S_1 + \{y_i\}$, $R_2' = R_2 - \{x_i\}$, and $S_2' = S_2 - \{y_i\}$. We have $R = R_1' + R_2'$, $S = S_1' + S_2'$. Since $x_i > y_i$, we also have $R_1' \succ S_1'$. Thus we can apply the inductive hypothesis to show that $R - \{x_j\} \succ S - \{y_j\}$ for $j \neq i$. Since there are two elements for which $x_i > y_i$, we can extend it to $j = i$.

We now consider the case where there is only one element in R_2 such that $x_i > y_i$. Let $r_1, r_2, \cdots, r_m$ denote the elements in R in non-increasing order and $s_1, s_2, \cdots, s_{m'}$ denote the elements in S also in non-increasing order. Here $m = |R|$ and $m' = |S|$. Let $\{k_1, k_2, \cdots, k_n\}$ and $\{j_1, j_2, \cdots, j_n\}$ be sets of integers such that $r_{k_l} = x_l$ and $s_{j_l} = y_l$, $l = 1, \cdots n$. Note that $x_i > y_i$ implies that $k_{i-1} + 1 < j_i \leq m' - (n - i)$. If $j_i \geq k_i$, then property 5 can be applied to yield $R - \{x_i\} \succ S - \{y_i\}$. Let us now consider the case $k_{i-1} + 1 < j_i < k_i$. The sets $R - \{x_i\}$ and $S - \{y_i\}$ can be expressed as $\{r_1', r_2', \cdots, r_{m-1}'\}$ and $\{s_1', s_2', \cdots s_{m'-1}'\}$ where

$$r_l' = \begin{cases} r_l, & 1 \leq l < k_i, \\ r_{l+1}, & k_i \leq l < m, \end{cases}$$

$$s'_l = \begin{cases} s_l, & 1 \leq l < j_i, \\ s_{l+1}, & j_i \leq l < m'. \end{cases}$$

Since $R \succ S$, it follows that $r'_l \geq s'_l$ when $1 \leq l \leq j_i - 1$ and when $k_i \leq l \leq m' - 1$ In addition $r'_l \geq s'_l$ when $k_i + 1 \leq l \leq j_i$ because $R_1 \succ S_1$ and $j_i > k_{i-1} + 1$. Therefore, we have shown that $R - \{x_i\} \succ S - \{y_i\}$ for this case. The relation $R - \{x_j\} \succ S - \{y_j\}$, $j \neq i$ follows from $x_j < y_j$ and property 5. □

In order to proceed with our treatment of preemptive systems, we introduce the notation $\boldsymbol{X}_\pi(t) = (n_\pi(t), \boldsymbol{E}_\pi(t))$ where $n_\pi(t)$ is the number of customers that have made their deadlines by time t. We refer to this as the state of the system at time t under policy π. We introduce the following notion of dominance between states.

Definition 7 *We say that $\boldsymbol{X}_{\pi_1}(t)$ dominates $\boldsymbol{X}_{\pi_2}(t)$ ($\boldsymbol{X}_{\pi_1}(t) \succ \boldsymbol{X}_{\pi_2}(t)$) iff*

1. $n_{\pi_1}(t) \geq n_{\pi_2}(t)$,

2. $\boldsymbol{E}_{\pi_1}(t) \succ Small(\boldsymbol{E}_{\pi_2}(t), |\boldsymbol{E}_{\pi_2}(t)| + n_{\pi_2}(t) - n_{\pi_1}(t))$.

Before we prove the main result of this section we describe some guidelines used to assign customers to servers and service times to customers. Without loss of generality we restrict ourselves to policies that satisfy the following rules.

- If the number of customers being served at some point in time is $i < c$, then the first i servers are busy.
- If servers i and j are occupied where $i < j$, then the deadlines of the customers assigned to these servers must be in non-decreasing order.

If policy π does not satisfy the above rules, we can always construct a policy π^* that satisfies these rules so that $\overline{V}_N(\pi) = \overline{V}_N(\pi^*)$ for all N and $\overline{V}(\pi) = \overline{V}(\pi^*)$. There also exists an STE policy that satisfies the above rules.

We now discuss the method by which we assign service times to jobs. Divide $\boldsymbol{B}$ into $c+1$ sequences, $\boldsymbol{B}^{(j)} = \{B_{i,j}\}_{i=1,\cdots}$, $j = 1, 2, \cdots, c+1$. Consider the i-th customer. Let m'_i denote the number of times it is scheduled. Let $s_{i,1}, s_{i,2}, \cdots, s_{i,m'_i}$ be the times at which it is scheduled, $q_{i,1}, q_{i,2}, \cdots, q_{i,m'_i-1}$ be the times at which it is preempted, k_i the identity of the server at which it completes, and $m_i = \min\{j \,|\, \sum_{l=1}^{j} > s_{i,m'_i}\}$. If the i-th customer misses its deadline, then $k_i = 0$. The service time, X_i of the i-th customer is

$$X_i = \begin{cases} \sum_{l=1}^{m'_i-1}(q_{i,l} - s_{i,l}) + \sum_{l=1}^{m_i} B_{l,k} - s_{i,m'_i}, & k_i \neq 0, \\ \sum_{l=1}^{m'_i-1}(q_{i,l} - s_{i,l}) + B_{i,c+1}, & k_i = 0. \end{cases} \tag{1}$$

We claim that the service times received by customers according to this assignment rule are i.i.d. exponential r.v.'s with parameter μ.

Theorem 1 *STE maximizes the fraction of customers that complete service before theri deadlines out of the class of service time independent policies for the G/M/c/K+G queue when the deadlines are to the end of service, i.e., $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$, $N > 0$, $\overline{V}(STE) \geq \overline{V}(\pi)$ where π is any service time independent policy.*

Proof: The proof by forward induction on the times that the following events can occur,

- $\mathcal{E}_0$ - arrival to both systems,
- $\mathcal{E}_1$ - completion of a job in either or both systems,
- $\mathcal{E}_2$ - job missing deadline under one or both policies,

Let $(t_0, \sigma_0), (t_1, \sigma_1), \cdots$ be the sequence of times and events that occur at those times, i.e., event σ_i occurs at time t_i where $\sigma_i \in \{\mathcal{E}_0, \mathcal{E}_1, \mathcal{E}_2\}$.

We will demonstrate that $\boldsymbol{X}_{STE}(t) \succ \boldsymbol{X}_\pi(t)$ for every sample $\boldsymbol{S} = s$ and $t \geq 0$ provided that $\boldsymbol{X}_{STE}(t_0) \succ \boldsymbol{X}_\pi(t_0)$.

According to property 6 of Lemma 1, if $\boldsymbol{X}_{STE}(t_i) \succ \boldsymbol{X}_\pi(t_i)$, and $t_i < t_{i+1}$, then $\boldsymbol{X}_{STE}(t) \succ \boldsymbol{X}_\pi(t)$ for $t_i \leq t < t_{i+1}$.

We proceed with our inductive argument.

Basis Step: The hypothesis is trivially true for $t = t_0$.

Inductive step: Assume that $\boldsymbol{X}_{STE}(t_l) \succ \boldsymbol{X}_\pi(t_l)$ for $l \leq i$. We now show that it also holds for $i + 1$. There are three cases according to the type of event.

Case 1 ($\sigma_{i+1} = \mathcal{E}_0$): There are four subcases according to whether there is overflow in neither, either, or both systems. In all four subcases, neither n_π nor n_{STE} are affected. In the case of no overflow in either system, property 1 of Lemma 1 guarantees that $\boldsymbol{X}_{STE}(t_{i+1}) \succ \boldsymbol{X}_\pi(t_{i+1})$. In the case of an overflow under π, property 3 of Lemma 1 guarantees that $\boldsymbol{X}_{STE}(t_{i+1}) \succ \boldsymbol{X}_\pi(t_{i+1})$. If overflow occurs under STE, then it must be the case that $|\boldsymbol{E}_{STE}(t_{i+1}^-)| > |Small(\boldsymbol{E}_\pi(t_{i+1}^-), |\boldsymbol{E}_\pi(t_{i+1}^-)| + n_\pi(t_{i+1}) - n_{STE}(t_{i+1}))|$. Hence property 2 of Lemma 1 is applicable and the result follows. The last case corresponding to overflow under π and STE is handled by property 5 from the Lemma.

Case 2 ($\sigma_{i+1} = \mathcal{E}_1$): There are three subcases according to whether the completion is under π, STE, or both policies. If the completion is under π only, then it occurs on server j where $j > |\boldsymbol{E}_{STE}(t_{i+1}^-)|$. This implies that $|\boldsymbol{E}_\pi(t_{i+1}^-)| > |\boldsymbol{E}_{STE}(t_{i+1}^-)|$ which further implies that $n_{STE}(t_{i+1}^-) > n_\pi(t_{i+1}^-)$. Consequently $n_{STE}(t_{i+1}) = n_{STE}(t_{i+1}^-) \geq n_\pi(t_{i+1}^-) = n_\pi(t_{i+1}^-) + 1$ and $\boldsymbol{E}_{STE}(t_{i+1}) = \boldsymbol{E}_{STE}(t_{i+1}^-) \succ Small(\boldsymbol{E}_\pi(t_{i+1}^-), |\boldsymbol{E}_\pi(t_{i+1}^-) + n_\pi(t_{i+1}^-) - n_{STE}(t_{i+1}^-)) = Small(\boldsymbol{E}_\pi(t_{i+1}), |\boldsymbol{E}_\pi(t_{i+1}) + n_\pi(t_{i+1}) - n_{STE}(t_{i+1}))$. Hence $\boldsymbol{X}_{STE}(t_{i+1}) \succ \boldsymbol{X}_\pi(t_{i+1})$.
If the completion is under STE only, then a similar calculation yields $\boldsymbol{X}_{STE}(t_{i+1}) \succ \boldsymbol{X}_\pi(t_{i+1})$.
If the completion is under both policies, then property 5 of Lemma 1 ensures that $\boldsymbol{X}_{STE}(t_{i+1}) \succ \boldsymbol{X}_\pi(t_{i+1})$.

Case 3 ($\sigma_{i+1} = \mathcal{E}_2$): Again there are three subcases according to whether the customer misses his deadline under π, STE, or both policies. If under π, property 3 of Lemma 1 is applicable. If under STE, property 2 of Lemma 1 is applicable. Last, property 4 of Lemma 1 is applicable when the losses occur under both policies.

It follows that $E[V_N(STE)|\boldsymbol{S} = s] \geq E[V_N(\pi)|\boldsymbol{S} = s]$ and $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$ for $N = 1, 2, \cdots$ and $\overline{V}(STE) \geq \overline{V}(\pi)$. □

Similar arguments can be used to show the following result at the other extreme.

Theorem 2 *LTE minimizes the fraction of customers that complete by their deadlines out of the class of non-idling service time independent policies for the G/M/c/K+G queue when the deadlines are to the end of service, i.e.,* $\overline{V}_N(LTE) \leq \overline{V}_N(\pi)$, $N > 0$, $\overline{V}(LTE) \leq \overline{V}(\pi)$ *where* π *is any non-idling service time independent policy.*

Remark. Similar results can also be proven for the discrete time bulk arrival G/M/c/K+G queue. Here the service time consists of an integer number of time units that is given by a geometric r.v. This model is of particular use in data communications in the case that the service time is always a single time unit. It forms the basis of many models of statistical multiplexers. In the case that customers require a single time unit of service, there is no distinction between preemptive and non-preemptive systems. Furthermore, there is no distinction between systems in which customers must meet their deadlines either by the time service begins or by the time service completes.

Theorem 1 can be generalized to include systems in which servers take vacations. This is of interest for at least two reasons. First, processors in any multiprocessor system are prone to failures. Second, systems in which servers take vacations can be used to model real-time systems with two classes of customers. For example, one class of tasks may be unable to tolerate missed deadlines. The second class of jobs may be able to tolerate some missed deadlines. If the tasks in the first class are well understood (i.e., known service times, arrival times), they can be given higher priority than the second class of tasks and scheduled independently of the second class. The second class of tasks are like the customers that we have considered in our model for which the object is to develop policies that will minimize the fraction of tasks that miss their deadlines. Thus tasks in the second class see a system where servers take vacations.

Let $\{U_{i,j}, W_{i,j}\}_{i=1,\dots}$, $j = 1, 2, \cdots, c$ be families of r.v.'s such that $U_{i,j}$ is the length of the i-th time interval during which the j-th server is available for service and $W_{i,j}$ is the length of the i-th time interval during which the j-th server is on vacation (unavailable for service). We allow these sequences of r.v.'s to have arbitrary statistics so long as they are independent of $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{D}$. In this case we state the following result.

Theorem 3 *STE maximizes the fraction of customers that make their deadlines in the preemptive continuous and discrete time $G/M/c/K+G$ queue with vacations when the deadlines are to the end of service, i.e.,* $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$, $N > 0$, $\overline{V}(STE) \geq \overline{V}(\pi)$ *for any service time independent policy* π.

Proof. The proof is similar to that given for Theorem 1 and is omitted here. □

Remark. Theorem 2 can be generalized in a similar way.

We conclude this section with the statement of properties of the best policies for the preemptive system. These are

Theorem 4 *STE is the STEI policy that maximizes the fraction of customers that complete by their deadlines for the preeemptive G/G/c/K+G queue. In other words,* $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$ $(1 \leq N)$ *and* $\overline{V}(STE) \geq \overline{V}(\pi)$ *for any* π *in STEI.*

Proof. This theorem is more easily proven using an interchange argument rather than forward induction. We sketch the proof. Consider an input sample $\boldsymbol{S}$. Consider some STEI policy $\pi \neq$ STE. Consider the first time t_0 that π deviates from STE. Let c denote the customer with the shortest time to extinction at time t_0. It is possible to construct a new policy π^* that schedules c at time t_0 and behaves like π elsewhere except that whenever π schedules c and c has completed under π^*, π^* lets the server remain idle. It is easy to convince oneself that π^* will have the same or better performance as π on the input sample $\boldsymbol{S}$ and that successive interchanges will produce STE with the best performance. As this is true for any sample path, it follows that $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$ $(1 \leq N)$ and $\overline{V}(STE) \geq \overline{V}(\pi)$. □

A similar argument can be used to prove the following result.

Theorem 5 *Consider the preemptive G/G/c/K+G queue with deadlines until the end of service. For any policy* π *that idles servers when there are waiting customers, there exists a non-idling policy* π^* *such that* $\overline{V}_N(\pi^*) \geq \overline{V}_N(\pi)$, $N > 0$, *and* $\overline{V}(\pi^*) \geq \overline{V}(\pi)$.

4 Non-Preemptive Systems with Deadline to End of Service

In this section we show that STE is the best policy and that LTE is the worst policy from the class of non-idling policies for the non-preemptive G/M/c/K+G queue when deadlines are to the end of service. Furthermore, we show that there exists a STEI policy that provides performance better than or equal to that of any service time independent non- STEI policy for the G/M/c/K+G queue.

Consider a policy π that is allowed to preempt a customer solely to move him to another server. We refer to this as a *limited preemption* policy and claim that the performance of this policy does not differ from a policy that uses the same scheduling rules except that it does not allow preemptions. We will find it easier to work with these limited preemption policies. Specifically, we consider limited preemption policies that enforce the following rules:

- If the number of customers in service, n is less than the number of servers, then they are placed on the first n servers.
- Customers are placed on servers such that the deadline associated with the customer on the i-th server is greater than or equal to that associated with the customer on the $(i+1)$-th server.

Customers are assigned service times according to the same rule used in analyzing the system that allows preemptions (see section 3).

Theorem 6 *STE provides the best performance of all non-idling service time independent policies for the non-preemptive $G/M/c/K+G$ queue when the deadlines are to the end of service, i.e., $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$, $N > 0$, $\overline{V}(STE) \geq \overline{V}(\pi)$ for any non-idling service time independent policy π.*

Proof: Though similar to the proof of Theorem 1, the proof of this theorem is more intricate because of the fact that the set of deadlines of the customers in service under STE may not dominate the set of deadlines of the customers under an arbitrary policy π. We show instead that both $\boldsymbol{E}_{STE}(t) \succ \boldsymbol{E}_\pi(t)$ and $\boldsymbol{R}_{STE}(t) \succ \boldsymbol{R}_\pi(t)$ for every sample path $\boldsymbol{S} = s$ using a forward induction argument on the times of events. These events are the same as defined in Theorem 1. Let $(t_0, \sigma_0), (t_1, \sigma_1), \cdots$ be the sequence of times and events that occur at those times, i.e., event σ_i occurs at time t_i where $\sigma_i \in \{\mathcal{E}_0, \mathcal{E}_1, \mathcal{E}_2\}$.

We note as in Theorem 1 that if $\boldsymbol{E}_{STE}(t_i) \succ \boldsymbol{E}_\pi(t_i)$ and $\boldsymbol{R}_{STE}(t_i) \succ \boldsymbol{R}_\pi(t_i)$ and $t_i < t_{i+1}$, then $\boldsymbol{E}_{STE}(t) \succ \boldsymbol{E}_\pi(t)$ and $\boldsymbol{R}_{STE}(t) \succ \boldsymbol{R}_\pi(t)$ for $t_i \leq t < t_{i+1}$.

We proceed with our inductive argument.

Basis Step: The hypothesis is trivially true for $t = t_0$.

Inductive step: Assume that $\boldsymbol{E}_{STE}(t_l) \succ \boldsymbol{E}_\pi(t_l)$ and $\boldsymbol{R}_{STE}(t_l) \succ \boldsymbol{R}_\pi(t_l)$ for $l \leq i$. We now show that it also holds for $i+1$. There are three cases according to the type of event.

Case 1 ($\sigma_{i+1} = \mathcal{E}_0$): This case is similar to case 1 in Theorem 1 and the details are omitted.

Case 2 ($\sigma_{i+1} = \mathcal{E}_1$): There are two subcases according to whether the completion is under STE or both policies. (Note: according to the inductive hypothesis and the server assignment rule, a completion under π implies a completion under STE.) If the completion is under STE only, then $\boldsymbol{E}_\pi(t_{i+1}) = \emptyset$ which implies that $\boldsymbol{E}_{STE}(t_{i+1}) \succ \boldsymbol{E}_\pi(t_{i+1})$. Because of the way that customers are assigned to servers, the deadline of the completed customer cannot reside in $Large(\boldsymbol{R}_{STE}(t_i^-), |\boldsymbol{R}_\pi(t_i^-)|)$. Consequently $\boldsymbol{R}_{STE}(t_{i+1}) \succ \boldsymbol{R}_\pi(t_{i+1})$. If the completion is under both policies, then property 8 of Lemma 1 and the inductive hypothesis ensure that $\boldsymbol{R}_{STE}(t_{i+1}) \succ \boldsymbol{R}_\pi(t_{i+1})$. The inductive hypothesis and the fact that STE will schedule the customer with the smallest deadline from $C_{STE}(t_i^-)$ ensures that property 5 of Lemma 1 can be applied to show that $\boldsymbol{E}_{STE}(t_{i+1}) \succ \boldsymbol{E}_\pi(t_{i+1})$.

Case 3 ($\sigma_{i+1} = \mathcal{E}_2$): Again there are three subcases according to whether the customer misses his deadline under π, STE, or both policies. If under π, property 3 of Lemma 1 is applicable. If under STE, property 2 of Lemma 1 is applicable. If under both STE and π, then we have further subcases according to whether the customers were in service or in the queue. In all of these cases, the result is obtained by using property 4 from Lemma 1.

It follows that $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$ for $N = 1, 2, \cdots$ and $\overline{V}(STE) \geq \overline{V}(\pi)$. □

Similar arguments can be used to prove the following theorem.

Theorem 7 *LTE provides the worst performance of all non-idling service time independent policies for the non-preemptive G/M/c/K+G queue when the deadlines are to the end of service, i.e.,* $\overline{V}_N(LTE) \leq \overline{V}_N(\pi)$, $N > 0$, $\overline{V}(LTE) \leq \overline{V}(\pi)$ *for any non-idling service time independent policy* π.

Let us consider now policies that may permit idle processors. We state and prove the following result.

Theorem 8 *For any arbitrary policy* π, *there exists a STEI policy* π^* *such that* $\overline{V}_N(\pi^*) \geq \overline{V}_N(\pi)$, $N > 0$, $\overline{V}(\pi^*) \geq \overline{V}(\pi)$ *for the G/M/c/K+G queue with no preemptions when the deadline is to end of service.*

Proof: Consider any policy π not in the class of STEI policies. We construct an STEI policy π^* that exhibits the same behavior as π. Policy π^* is given below.

1. π^* maintains ordered lists of customers, $\mathcal{A}(t)$, $\mathcal{R}(t)$, corresponding to customers that would be in the queue and to all of the customers that would be in the

system respectively at time t under π when provided the same input sample, i.e., $\mathcal{A}(t) = \mathcal{C}_\pi(t)$.

2. π^* maintains a history H_t' identical to the history produced by π when given the same input sample, i.e., $H_t' = H_t$.

3. π^* makes scheduling decisions according to the following rules

   (a) If at time t $|\mathcal{R}(t)| - |\mathcal{A}(t)| = |\boldsymbol{R}_{\pi^*}(t)| - |\mathcal{C}_{\pi^*}(t)|$ then π^* schedules the customer closest to its deadline with probability $1 - p_0(\pi, t, H_t')$. Otherwise, π^* does not schedule a customer.

   (b) At time t, π^* schedules no customer with probability $p_0(\pi, t, H_t')$.

4. π^* modifies $\mathcal{A}(t)$ as follows,

   (a) customer c_i is removed from $\mathcal{A}(t)$ either 1) when its deadline expires, or 2) with probability $p_i(\pi, t, H_t')$ at a time t when π^* schedules a customer,

   (b) customer c_i is added to $\mathcal{A}(t)$ when it arrives to the system.

5. π^* modifies $\mathcal{R}(t)$ as follows,

   (a) customer c is removed from $\mathcal{R}(t)$ either when its deadline expires or it corresponds to a customer in $\boldsymbol{R}_{\pi^*}(t)$ that completes service.

   (b) customer c is added to $\mathcal{R}(t)$ when it arrives to the system.

6. π^* modifies H_t' as follows,

   (a) at the time of an arrival the arrival time and relative deadline the customer are added to $\boldsymbol{a}_t$ and $\boldsymbol{d}_t$.

   (b) at the time of a departure, the service time of the customer is added to $\boldsymbol{d}_t$.

   (c) at the time that π^* assigns a customer to service, the identity of the customer removed from $\mathcal{A}(t)$ (see 4.(a) above) and the time of the assignment are added to $\boldsymbol{r}_t$ and $\boldsymbol{e}_t$, respectively.

7. π^* removes the customer closest to extinction whenever the system is full at the time of a customer arrival.

These rules allow us to couple sample paths under each policy in such a way that we can show that $\boldsymbol{E}_{\pi^*}(t) \succ \boldsymbol{E}_{\pi}(t)$ and $\boldsymbol{R}_{\pi^*}(t) \succ \boldsymbol{R}_{\pi}(t)$ for $0 \leq t$ using a forward induction argument as before. The details of this argument are omitted as they differ very little from the arguments given in the last two theorems. It follows that $\overline{V}_N(\pi^*) \geq \overline{V}_N(\pi)$ for $N > 0$ and $\overline{V}(\pi^*) \geq \overline{V}(\pi)$.

□

Remark. Similar results can be proven for discrete time queues.

5 The Nonpreemptive Queue with Deadlines to Beginning of Service

In this section we show that there is no class of policies better than the STEI policies for the non-preemptive G/G/c+G queue when the deadline is to the beginning of service. We also show that STE is the best policy and LTE is the worst policy out of the class of *non-idling policies* for these queues when service times are restricted to be independent and identically distributed exponential random variables.

We first show that any non-STEI policy π^* can be emulated by some STEI policy π in the sense that $\overline{V}_N(\pi) = \overline{V}_N(\pi^*)$ for all N and $\overline{V}_t(\pi) = \overline{V}_t(\pi^*)$ for all t. As the proof of this theorem is similar to that of Theorem 1 and of Theorem 1 in [18] we will merely provide a sketch of the proof.

Theorem 9 *For any policy π, there exists an STEI policy π^* such that $\overline{V}_N(\pi^*) = \overline{V}_N(\pi)$, $0 < N$, $\overline{V}(\pi^*) = \overline{V}(\pi)$ for the non-preemptive G/G/c/K + G queue with deadlines to the beginning of service.*

Proof: Consider any policy π not in the class of STEI policies. Using the methods described in the proof of Theorem 1 in [18], it is possible to construct an STEI policy π^* such that the sample paths are coupled under both π^* and π. We define $n_{\pi}(t)$ to denote the number of customers that have satisfied their deadline under policy π and $\boldsymbol{T}_{\pi}(t)$ to be the set of remaining service times for all of the customers in service under π. The proof that π^* has the same performance as π is based a forward induction argument to show that $\boldsymbol{E}_{\pi^*}(t) \succ \boldsymbol{E}_{\pi}(t)$, $n_{\pi}(t) = n_{\pi^*}(t)$ and $\boldsymbol{T}_{\pi}(t) = \boldsymbol{T}_{\pi^*}(t)$ at the time of all events on a single sample path, i.e., arrivals, service completions, assignment of customers to processors, and times of extinction). These relations then hold for the times between events as a consequence of property 7 of Lemma 1. Since this is true for any sample path, it follows that $\overline{V}_N(\pi) = \overline{V}_N(\pi^*)$ for $N = 1, 2, \cdots$ and that $\overline{V}(\pi) = \overline{V}(\pi^*)$. $\Box$

We complete the section on multiple server queues with deadlines until the beginning of service by proving that STE is the best non-idling, service time independent policy for the non-preemptive G/M/c/K+G system. Before proceeding with the proof, we discuss the method for assigning service times to customers. Let $\boldsymbol{B}^{(j)} = \{B_{i,j}\}_{i=1,\cdots}$ $(1 \leq j \leq c)$ be c mutually independent sequences of i.i.d. exponential random variables with parameter μ. Let these sequences be mutually independent. If a customer is assigned to the kth server at time t, then it receives an amount of service equal to $\sum_{l=1}^{m} B_{l,k} - t$ where $m = \min\{i | \sum_{l=1}^{i} B_{l,k} > t\}$. We emphasize that, due to the assumptions on $\boldsymbol{B}^{(j)}(1 \leq j \leq c)$, the service time received by this customer is exponentially distributed and independent of other events in the system. We redefine $\boldsymbol{S}$ and $\boldsymbol{s}$ to be $\boldsymbol{S} = (\boldsymbol{A}, \boldsymbol{D}, \boldsymbol{B}^{(1)}, \cdots, \boldsymbol{B}^{(c)})$ and $\boldsymbol{s} = (\boldsymbol{a}, \boldsymbol{d}, \boldsymbol{b}^{(1)}, \cdots, \boldsymbol{b}^{(c)})$

Theorem 10 *STE is the best non-idling, service time independent policy for the non-preemptive G/G/c/K+G system with deadlines until the beginning of service, i.e.,* $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$, $N > 0$, $\overline{V}(STE) \geq \overline{V}(\pi)$ *for any non-idling, service time independent policy* π.

Proof: Define $\boldsymbol{T}_\pi(t) = (t_\pi^{(1)}(t), \cdots, t_\pi^{(c)}(t))$ where $t_\pi^{(j)}(t) = 1$ if server j is busy under π at time t and 0 otherwise.

As before, the proof is by forward induction. Using the properties of the dominance relation "$\succ$" We show that $\boldsymbol{E}_{STE}(t) \succ \boldsymbol{E}_\pi(t)$ and $\boldsymbol{T}_{STE}(t) \geq \boldsymbol{T}_\pi(t)$ at each possible event (i.e., arrival, departure, and deadline miss) for every sample path $\boldsymbol{S}$. The arguments are similar to those found in Theorems 1 and 6. It follows that $\overline{V}_N(STE) \geq \overline{V}_N(\pi)$ for $N = 1, 2, \cdots$ and $\overline{V}(STE) \geq \overline{V}(\pi)$. $\Box$

Similar arguments yield the following result.

Theorem 11 *If* π *is any non-idling policy, then* $\overline{V}_N(LTE) \geq \overline{V}_N(\pi)$ *for* $N = 1, \cdots$, $\overline{V}(LTE) \geq \overline{V}(\pi)$ *for the non-preemptive, G/M/c/K+G system with deadlines until the beginning of service.*

Remark. Analogous results hold also for bulk arrival discrete time G/M/c/K+G queues. The STEI result extends to the case where service times are i.i.d. positive integer valued r.v.'s with an arbitrary distribution..

6 Extensions

If one is interested in the performance of a multiple server system during a finite period of time [2], then there exist results analogous to those proven in the preceding sections, provided that the arrival process is nonexplosive [3, p. 19]. Let $V_t(\pi)$ denote the number of customers that make their deadline within the interval $(0, t]$, then we have the following result for preemptive G/M/c/K+G queues with deadlines to the end of service and non-preemptive G/M/c/K+G queues with deadlines to the beginning or end of service,

$$V_t(STE) \geq_{st} V_t(\pi) \geq_{st} V_t(LTE) \tag{2}$$

for all non-idling policies π. In the case of preemptive systems

$$V_t(STE) \geq_{st} V_t(\pi).$$

Here the "$\geq_{st}$" relation between two r.v.'s is defined as in Ross [21, p. 251]. Random variable X is said to ***stochastically dominate*** random variable Y ($X \geq_{st} Y$) iff $\Pr[X \leq x] \leq \Pr[Y \leq x]$, $-\infty < x < \infty$.

Let $M_t(\pi)$ denote the number of customers in the system at time t under policy π, i.e., $M_t(\pi) = |\boldsymbol{R}_\pi(t)|$, $0 < t$. The following relations can be established provided the arrival process is nonexplosive for the three G/M/c/K+G queues considered in this paper,

$$M_t(STE) \geq_{st} M_t(\pi) \geq_{st} M_t(LTE)$$

for all non-idling policies π. It should be apparent from earlier proofs that these results are a consequence of our state definitions and forward induction arguments.

7 Summary

We have shown that STEI policies are among the best scheduling policies for maximizing the fraction of customers making a deadline in many finite or infinite capacity multiple server queues. Furthermore, out of the class of non-idling service time independent policies, STE maximizes and LTE minimizes the fraction of customers that complete by their deadlines for the nonpreemptive G/M/c/K+G queue. Last, if deadlines are to the end of service, then the best policy that does not use service time information for the preemptive G/M/c/K+G queue is STE. The worst non-idling policy that does not use service time infoprmation for this system is LTE. These results hold for systems in which servers take vacations.

References

[1] F. Baccelli, P. Boyer, and G. Hebuterne, "Single-Server Queues with Impatient Customers" *Adv. Appl. Prob.*, Vol. 16, pp. 887-905, 1984.

[2] P. Bhattacharya and A. Ephremides, "Scheduling of Time-Critical Messages," *Proc. 1988 Conf. Inf. Sciences Systems* pp. 623-628, 1988.

[3] P. Bremaud, *Point Processes and Queues: Martingale Dynamics*, Springer-Verlag, New York, 1981.

[4] T.M. Chen, J. Walrand, and D.G. Messerschmitt, "Dynamic Priority Protocols for Packet Voice", *IEEE Journal on Sel. Areas in Comm.*, 7, 5, pp. 632-643, June 1989.

[5] E.G. Coffman (Ed.), *Computer and Job-Shop Scheduling Theory*, John Wiley, New York, 1976.

[6] J.W. Cohen, "Single Server Queues with Restricted Accessibility," *J. Eng. Math.*, Vol. 3, 4, pp. 265-284, Oct. 1969.

[7] R.W. Conway, W.L. Maxwell and L.W. Miller, *Theory of Scheduling*, Addison-Wesley, Reading, Mass., 1967.

[8] M. Dertouzos, "Control Robotics: The Procedural Control of Physical Processes," *Proceedings of the IFIP Congress*, pp. 807-813, 1974.

[9] B.T. Doshi and E.H. Lipper, "Comparisons of Service in a Queueing System with Delay Dependent Customer Behavior", *Applied Probability - Computer Science: The Interface, Vol. II*, R.L. Disney. T.J. Ott, Eds., Cambridge, MA, Birkhauser, pp. 269-301, 1982.

[10] B. Gavish and P.J. Schweitzer, "The Markovian Queue with Bounded Waiting Time," *Management Sci.*, Vo. 21, 12, pp. 1349-1357, Aug. 1972.

[11] J. Hong, X. Tan, D. Towsley, "A Performance Analysis of Minimum Laxity and Earliest Deadline Scheduling in a Real-Time System," to appear in *IEEE Trans. Computers*, Dec. 1989.

[12] J.R. Jackson, "Scheduling a Production Line to Minimize Maximum Tardiness," Research Report 43, Management Science Research Report, UCLA, 1955.

[13] Y. Lim and J. Kobza, "Analysis of a Delay-Dependent Priority Discipline in a Multi-Class Traffic Packet Switching Node", *Proc. IEEE INFOCOM-88*, New Orleans, LA, 1988.

[14] C. Liu and J. Layland, "Scheduling Algorithms for Multi-Programming in a Hard-Real-Time Environment," *J. ACM*, Vol. 20, pp. 46-61, 1973.

[15] A. Mok and M. Dertouzos, "Multiprocessor Scheduling in a Hard Real-Time Environment," *Proc. 7th Texas Conf. on Comp. Syst.*, Nov. 1978.

[16] J.M. Moore, "An n Job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs," *Management Science*, vol. 15, pp. 102-109, 1968.

[17] S.S. Panwar, *Time Constrained and Multi-access Communications*, Ph.D. Thesis, Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, Feb. 1986.

[18] S.S. Panwar, D. Towsley, and J.K. Wolf, "Optimal Scheduling Policies for a Class of Queues with Customer Deadlines to the Beginning of Service," *J. ACM*, **35**, 4, pp. 832-844, Oct. 1988.

[19] W.P. Pierskalla and C. Roach, "Optimal Issuing Policies for Perishable Inventory," *Management Science*, vol. 18, pp. 603-614, 1972.

[20] M. Pinedo, "Stochastic Scheduling with Release Dates and Due Dates," *Operations Research*, vol. 31, pp. 559-572, 1983.

[21] S.M. Ross, *Stochastic Processes*, John Wiley & Sons, New York, 1983.

[22] M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis*, Addison Wesley, 1987.

[23] Zaw-Sing Su and Kenneth C. Sevcik, "A Combinatorial Approach to Scheduling Problems," *Operations Research*, vol. 26, pp. 836-844, 1978.