Optimal Scheduling Policies for a Class of Queues with Customer Deadlines to the Beginning of Service

SHIVENDRA S. PANWAR

Polytechnic University, Brooklyn, New York

DON TOWSLEY

University of Massachusetts, Amherst, Massachusetts

AND

JACK K. WOLF

University of California at San Diego, La Jolla, California

Abstract. Many problems can be modeled as single-server queues with impatient customers. An example is that of the transmission of voice packets over a packet-switched network. If the voice packets do not reach their destination within a certain time interval of their transmission, they are useless to the receiver and considered lost. It is therefore desirable to schedule the customers such that the fraction of customers served within their respective deadlines is maximized. For this measure of performance, it is shown that the shortest time to extinction (STE) policy is optimal for a class of continuous and discrete time nonpreemptive M/G/1 queues that do not allow unforced idle times. When unforced idle times are allowed, the best policies belong to the class of shortest time to extinction with inserted idle time (STEI) policies. An STEI policy requires that the customer closest to his or her deadline be scheduled whenever it schedules a customer. It also has the choice of inserting idle times while the queue is nonempty. It is also shown that the STE policy is optimal for the discrete time G/D/1 queue where all customers receive one unit of service. The paper concludes with a comparison of the expected customer loss using an STE policy with that of the first-come, first-served (FCFS) scheduling policy for one specific queue.

Categories and Subject Descriptors: C.2.1 **[Computer Communications Networks]**: Network Architecture and Design; D.4.8 **[Operating Systems]**: Performance—*queuing theory*; *stochastic analysis*; F.2.2 **[Analysis of Algorithms and Problem Complexity]**: Nonnumerical Algorithms and Problems—*sequencing and scheduling*; G.m **[Discrete Mathematics]**: Miscellaneous—*queuing theory*

General Terms: Design, Performance, Theory, Verification

Additional Key Words and Phrases: Control of Queues, Markov decision processes, packetized voice communications, queues with impatient customers, stochastic scheduling theory

1. *Introduction*

The problem of a queue with customers that have to begin service before their respective deadlines has several diverse applications. An example is that of impatient customers who leave the queue if they are not served within a certain time

This work was supported by the National Science Foundation under grant ECS 83-10771 and by the Center for Advanced Technology in Telecommunications, Polytechnic University, Brooklyn, N.Y.

Authors' present addresses: S. S. Panwar, Department of Electrical Engineering, Polytechnic University, Brooklyn, NY 11201; D. Towsley, Department of Computer and Information Science, University of Massachusetts, Amherst, MA 01003; J. K. Wolf, Center for Magnetic Recording Research, University of California at San Diego, La Jolla, CA 92093.

interval of their joining the queue. Another example is that of a blood bank in which the stored blood may be unusable if it is not used within a certain number of days after it is collected. Yet another application, which is the motivation for this paper, is the transmission of voice or video over a packet-switched network. These packets have to reach their destination within a certain time interval of their transmission or they are useless to the receiver and considered lost. Voice communication requires that the end-to-end propagation delay be no more than about 300 ms for a normal conversation [3]. However, the loss of a small percentage of packets is usually tolerable since the speech is still intelligible to the receiver [5].

A node of a packet-switched network or a local-area network is often modeled as a single-server queue [8]. Therefore, we consider as our model a single-server queue with customers with deadlines. The parameter that we wish to maximize is the fraction of customers that begin their service before their respective deadlines. We consider both continuous and discrete time queues in which no preemptions are allowed and in which the service times are not known at the beginning of service. We show that if an optimal policy exists, then it belongs to the class of shortest time to extinction with unforced idle time (STEI) policies. If an optimal policy does not exist, we show that the *best* policies belong to the class of STEI policies. Here, an STEI policy is one that, whenever the queue is not empty, may choose to schedule either no customer or the customer closest to its deadline.

When we restrict ourselves to the class of policies that do not allow unforced idle times, then the shortest time to extinction (STE) policy is optimal for the nonpreemptive M/G/1 queue. Here the STE policy schedules the customer closest to its deadline. Last, we show that the STE policy is optimal over all policies for the discrete time G/D/1 queue when the service time is exactly one time unit. This latter queue is of particular interest because it is a commonly used model for statistical multiplexers in data communication systems [13].

The STE policy is very similar to the earliest due date (EDD) scheduling policy proposed by Jackson [7]. Consider a set of n tasks $\{T_i,\ 1 \le i \le n\}$ with the corresponding n due dates $\{d_i,\ 1 \le i \le n\}$. 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 nondecreasing due dates. As we shall see in the next section, STE scheduling differs from EDD scheduling in that it never schedules tasks that are already past their due dates. Note that the tasks and their due dates are known a priori under Jackson's model. Using the same a priori information, Moore [9] devised an algorithm to minimize the number of late tasks. Pinedo [12] considered the problem of minimizing the number of late jobs (customers) when the processing times are exponentially distributed and the deadlines are randomly distributed. He assumed that no new jobs are allowed into the system once the processing begins. Su and Sevcik [14] consider the problem of scheduling customers with deadlines in a queue. They showed that EDD scheduling minimized performance parameters such as expected lateness and tardiness.

Pierskalla and Roach [11] showed that a policy similar to the STE policy is an optimal issuing policy under the conditions that prevail in blood banks. Here, the additions to the blood bank ("customer arrivals") are random as is the demand ("customer service times"), and the issuing policy should be such that the amount of blood that becomes unusable as a result of being stored too long is minimized. More recently, while considering scheduling problems that arise in the area of real-time systems, Dertouzos [2] has shown that for any arbitrary set of arrivals with arbitrary processing times and deadlines the EDD policy is optimal if preemptions

are allowed. Here a (real-time) scheduling policy is considered optimal if it produces a feasible schedule whenever a clairvoyant scheduling policy (which is aware of future job arrivals) can do so. In queuing theory literature, queues with impatient customers have been usually analyzed assuming a FCFS scheduling policy [1].

In Section 2 we introduce the notation used in this paper and define the STE policy and the class of STEI policies. Section 3 contains the results regarding the optimality of the STE policy and the class of STEI policies. In Section 4, we compute the throughput of an STE policy and compare its performance with the FCFS scheduling policy for the M/D/1 queue. We conclude this paper with Section 5, in which we summarize our results.

2. *Definitions and Notation*

We consider queues in which each customer has a deadline from the time of arrival to the beginning of service. Since we are interested in both continuous and discrete time queues, we introduce notation that applies to both. We assume that the reader is aware of when a random variable takes on continuous values (continuous time queues) and when it takes on discrete values (discrete time queues). In the case of the discrete time queue, we assume that the basic unit of time is of length 1.

Let T_i denote the arrival time of the ith customer. Let A_i denote the time between the arrivals of the $(i-1)$st and ith customers. We assume that A_i is a random variable with arbitrary distribution. Let E_i denote the extinction time of the ith 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 refer to D_i as the *real-time constraint* or the *relative deadline* for customer i. Last, let $\{B_i\}_{1 \le i}$ be an independent and identically distributed (i.i.d.) sequence of random variables denoting the service times of the customers.

We use the notation $\mathbf{A}_N = \{A_i\}_{1 \le i \le N}$, $\mathbf{D}_N = \{D_i\}_{1 \le i \le N}$, $\mathbf{B}_N = \{B_i\}_{1 \le i \le N}$, and $\mathbf{S}_N = (\mathbf{A}_N, \mathbf{D}_N, \mathbf{B}_N)$, $1 \le N$. In addition, whenever we focus on a specific sample realization of the above random variables, we shall use lowercase notation (i.e., a_i for A_i, etc.). Furthermore, we let $\mathbf{a} = \{a_i\}_{1 \le i}$, $\mathbf{b} = \{b_i\}_{1 \le i}$, $\mathbf{d} = \{d_i\}_{1 \le i}$, $\mathbf{a}_N = \{a_i\}_{1 \le i \le N}$, $\mathbf{b}_N = \{b_i\}_{1 \le i \le N}$, and $\mathbf{d}_N = \{5d_i\}_{1 \le i \le N}$. Last, let $\mathbf{s} = (\mathbf{a}, \mathbf{d}, \mathbf{b})$ and $\mathbf{s}_N = (\mathbf{a}_N, \mathbf{d}_N, \mathbf{b}_N)$, $N = 1, \ldots$. These last two quantities are referred to as an *input sample* and *finite input sample*, respectively.

We consider two rules for assigning service times to customers.

Rule 1 (assignment at arrival). According to this rule, B_i denotes the service time of the ith customer to arrive in the system.

Rule 2 (assignment at service). According to this rule, B_i denotes the service time of the ith customer to be served.

In either case we assume that the service times are independent of the arrival times and extinction times. Although not considered in this paper, other assignment rules are possible. Note that the analysis of single-server queues without deadlines does not depend on how service times are assigned to customers. Although this holds true for the queue with deadlines, we shall find that queues in which service times are assigned according to rule 2 are easier to analyze. Also note that, if the service times of all the customers are identical, the two assignments are also identical.

We use the notation A/B/C + D to denote a queue with customer deadlines where A, B, and C have the same meaning as in Kendall's notation and D gives the distribution of the relative deadlines. No preemptions are allowed in any of the queues that we consider in this paper.

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 bases its decision 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 total expected number of customers served for this system. Define the fraction of customers served for the system as $N \to \infty$ (under policy π) to be

$$V(\pi) = \lim_{N \to \infty} \inf \frac{V_N(\pi)}{N}.$$

Finally, let $V = \sup_\pi V(\pi)$. A policy π^* is *optimal* if $V(\pi^*) = V$.

We find it useful to calculate the fraction of customers that make their deadline for a specific sample path $\mathbf{s}$. Consequently, we define $V_N(\pi, \mathbf{s}_N)$ to be the number of customers served in a system with exactly N arrivals having sample path $\mathbf{s}_N$ and $V(\pi, \mathbf{s})$ to be the long-term fraction of customers that are served when the sample path is $\mathbf{s}$. Here $V(\pi, \mathbf{s})$ is defined as

$$V(\pi, \mathbf{s}) = \lim_{N \to \infty} \inf \frac{V_N(\pi, \mathbf{s}_N)}{N}.$$

In addition, $V_N(\pi)$ and $V(\pi)$ can be expressed as

$$V_N(\pi) = E[V_N(\pi, \mathbf{s}_N)],$$
$$V(\pi) = E[V(\pi, \mathbf{s})].$$

A customer is eligible under policy π at time t if it has neither exceeded its deadline nor begun service. Consequently, the set of customers of interest at any time t is denoted by $C_\pi(t) = \{c_{j1}, c_{j2}, \ldots, c_{jn}\}$ consisting of all the eligible customers at time t, $j_i \geq 1$, $1 \leq i \leq n$. The set of extinction times of these customers is denoted by $E_\pi(t)$.

Consider the actions that policy π can take at time t. If the server is busy, then π takes no action. If the server is idle at time t, then π can either schedule no customer or schedule a single customer from $C_\pi(t)$. Policy π is allowed to choose one of these actions according to some distribution that depends on π, $C_\pi(t)$, and the previous history H_t. We define $p_0(\pi, t, H_t)$ to be the probability that π chooses not to schedule a customer and $p_i(\pi, t, H_t)$ to be the probability that π schedules customer $j_i \in C_\pi(t)$, $1 \leq i \leq n$.

If π chooses not to schedule a customer at time t and $C_\pi(t) \neq \varnothing$, then it delays making a new scheduling decision by a random amount of time τ with some arbitrary distribution function $F_\tau(x \mid H_t)$ (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 when the queue is nonempty unless they are separated by the arrival of one or more customers.

The history of the system up to time t may be defined as the tuple $H_t = (\boldsymbol{\sigma}_t, \mathbf{d}_t, \mathbf{r}_t, \mathbf{e}_t, \mathbf{u}_t)$, where $\boldsymbol{\sigma}_t$ is an ordered set of arrival times of all customers that arrive prior to t; $\mathbf{d}_t$ is an ordered set of relative deadlines corresponding to the customers that arrive prior to t; and $\mathbf{r}_t$ is an ordered set of all times prior to time t at which customers began service. In addition, $\mathbf{e}_t$ is an ordered set of all customers

that were in service by time t, and $\mathbf{u}_t$ is an ordered set of the service times for completed customers prior to time t.

We are now in a position to show that $V_N(\pi)$, $N > 0$, and $V(\pi)$ do not depend on the rule used to assign service times to customers.

LEMMA 1. *The performance of a policy π does not depend on which assignment rule is used to assign service times to customers.*

PROOF. Let us consider the first N customers $c_1, \ldots, c_N$ that enter the system. Let $V_N^{(1)}(\pi)$ and $V_N^{(2)}(\pi)$ denote the expected number of customers served under policy π when service times are assigned according to rules 1 and 2, respectively. We shall show that $V_N^{(1)}(\pi) \geq V_N^{(2)}(\pi)$. A similar argument can then be used to establish the reverse inequality.

We define a new random variable $\mathscr{C}_N$ corresponding to the ordered set of completed customers out of the N customers. This set is ordered in increasing customer completion time and takes on value $\xi = \{c_{k_1}, c_{k_2}, \ldots, c_{k_n}\}$, $c_{k_i} \in \{c_1, c_2, \ldots, c_N\}$, $1 \leq i \leq n \leq N$. Given a specific finite input sample $\mathbf{s}_N$, the rules governing policy π allow us to compute the probability distribution for $\mathscr{C}$, $q_\xi(\mathbf{s}_N, \pi) = P[\mathscr{C}_N = \xi \mid \mathbf{S}_N = \mathbf{s}_N, \pi]$, $\xi \subseteq \{c_1, \ldots, c_N\}$. Let $\mathbf{b}'_N(\xi)$ be the set of service times associated with the customers in ξ also ordered in increasing customer completion time. Last, define $\overline{\mathbf{b}'_N(\xi)}$ to be the set of customers that did not make their deadlines, ordered in increasing customer arrival time; that is, $\overline{\mathbf{b}'_N(\xi)} = \{b_{m_1}, \ldots, b_{m_{N-n}}\} = \mathbf{b}_N - \mathbf{b}'_N(\xi)$, where $a_{m_i} \leq a_{m_j}$, $i < j$.

Construct a new finite input sample $\mathbf{s}_N^{(2)}$ that has the same arrival times and deadlines as $\mathbf{s}_N$ and the following service times: $\mathbf{b}_N^{(2)} = \mathbf{b}'_N(\xi) \circ \overline{\mathbf{b}'_N(\xi)}$.[1] Note that $\mathbf{s}_N^{(2)}$ differs from $\mathbf{s}_N$ only in $\mathbf{b}_N^{(2)}$ which is a permutation of $\mathbf{b}_N$. Since the service times are i.i.d., $\mathbf{s}_N^{(2)}$ and $\mathbf{s}_N$ have equal probability measure. We claim that policy π operating on this new finite input sample using rule 2 for assigning service times to customers will produce the completed set of customers $\mathscr{C}_N = \xi$ with probability $q_\xi(\mathbf{s}_N, \pi)$. Consequently, for every finite input sample $\mathbf{s}_N$ and set of completed customers $\mathscr{C}_N$, we can determine a new finite input sample having equal probability measure such that the same set of customers is completed with equal probability. Therefore, taking the expectation over all completed sets of customers and finite input samples yields $V_N^{(1)}(\pi) \geq V_N^{(2)}(\pi)$, $N = 1, \ldots$.

A similar argument can be used to show that the reverse inequality holds. Therefore we have $V_N^{(1)}(\pi) = V_N^{(2)}(\pi)$.

We have shown that the performance is independent of the assignment rule for any $N = 1, \ldots$. Consequently, $V(\pi)$ is also independent of the assignment rule used. Q.E.D.

As a consequence of the above lemma, we are free to choose either assignment rule in subsequent discussions. We now conclude this section with a definition of the STE policy and the class of STEI policies.

Let the kth customer to be served since time $t = 0$ be assigned to the server at time t'_k.

Definition 1. Policy π is a *shortest time to extinction* (*STE*) policy if at time t'_k, $1 \leq k$, it always schedules the eligible customer with the smallest extinction time. In addition, the server is always busy as long as there are eligible customers available

[1] Here, if $R = \{x_1, x_2, \ldots, x_n\}$ and $S = \{y_1, \ldots, y_m\}$ are ordered sets, then $R \circ S$ is the ordered set $\{x_1, \ldots, x_n, y_1, \ldots, y_m\}$.

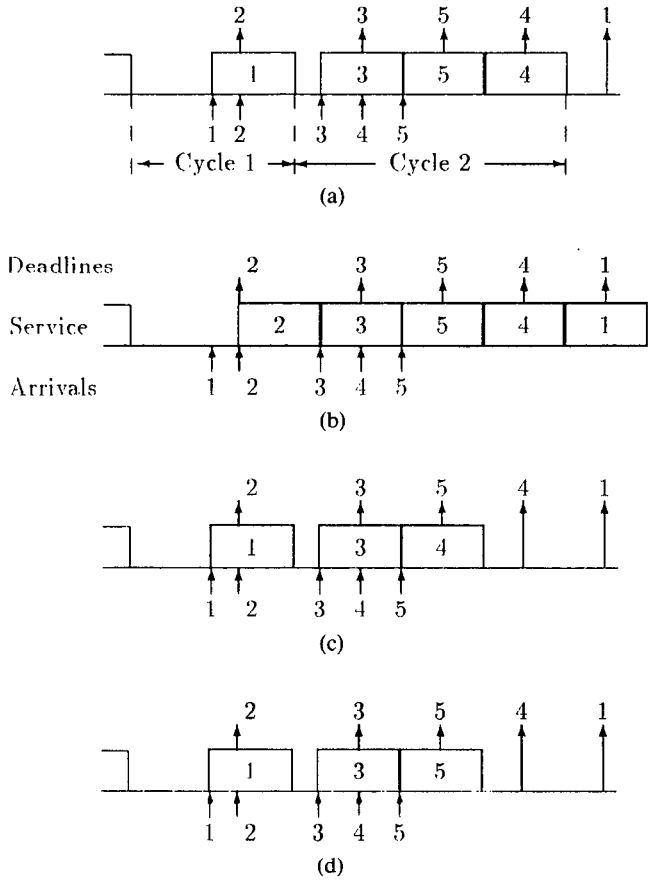

FIG. 1. (a) Behavior of STE. (b) Behavior of STEI. (c) Behavior of FCFS. (d) STEI emulating FCFS.

that have not yet been served; that is, $p_0(\pi, t) = 0$ whenever the server is available and $C_\pi(t) \neq \phi$.

An example of how the STE policy schedules a given set of arrivals is shown in Figure 1a.

Definition 2. *Unforced idle times* are time intervals when the server is idle while eligible customers are available.

Definition 3. Policy π is a *shortest time to extinction with unforced idle times* (*STEI*) policy if, whenever it schedules a customer, it schedules the eligible customer with the smallest extinction time. In other words, $p_0(\pi, t) \geq 0$, $p_I(\pi, t) \geq 0$, where $e_{j_I} = \min\{e_k : e_k \in C_\pi(t)\}$, and $p_I(\pi, t) = 0$, otherwise.

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

3. *The Optimality of the STE and STEI Scheduling Policies*

In this section we show that STE is optimal when unforced idle times are prohibited for the M/G/1 + G queue and that there is no class of policies better than the STEI class of policies for the G/G/1 + G queue when unforced idle times are allowed. In the course of proving these results, we shall compare sets of extinction times and show that one set *dominates* another set. Consequently, the first step is to define dominance and establish some properties that are satisfied by this relation.

Consider two ordered sets of nonnegative real numbers $R = \{x_1, x_2, \ldots, x_n\}$ and $S = \{y_1, y_2, \ldots, y_m\}$. We define a Large$(R, k)$ to be the ordered subset of R that contains the k largest values of R. More formally,

$$\text{Large}(R, k) = \begin{cases} \varnothing, & k \le 0, \\ \{\max R\} \cup \text{Large}(R - \{\max R\}, k - 1), & 0 < k < |R|, \\ R, & k \ge |R|, \end{cases}$$

where max R is the largest element in R.

Definition 4. We say that R *dominates* S $(R \succ S)$ if either

(1) $R = S$,
(2) $n = m$ and $x_{j_i} \ge y_{k_i}$, $1 < i < m$, for some permutations $j_1, \ldots, j_m$ and $k_1, \ldots, k_m$ of $1, \ldots, m$,
(3) $n > m$ and Large$(R, m) \succ S$.

We define the operation Shift$(R, x) = \{y - x \mid y \in R, y \ge x\}$. Observe that $|\text{Shift}(R, x)| \le |R|$ since there may be some $y \in R$ such that $x > y$. Here $|R|$ denotes the cardinality of the set R. The following lemma gives conditions under which dominance is preserved when set operations and the Shift operation are performed on R and S.

LEMMA 2. *If* $R \succ S$, *then*:

(1) $R + \{x\} \succ S + \{x\}$, *for* $x > 0$,
(2) $R - \{x\} \succ S$, *where* $x = \min_{1 \le i \le n}\{x_i\}$ *and* $n > m$,
(3) $R \succ S - \{y\}$, *where* $y \in S$,
(4) $R - \{x\} \succ S - \{y_k\}$, *where* $x = \min_{1 \le i \le n}\{x_i\}$ *and* $1 \le k \le m$,
(5) *Shift*$(R, x) \succ$ *Shift*(S, x).

PROOF

(1) The proof of this property follows directly from the definition of dominance for two sets of nonnegative numbers.
(2) Since $n > m$ and x is the smallest element of R, then Large$(R - \{x\}, m) =$ Large$(R, m) \succ S$.
(3) The proof of this property follows directly from the definition of dominance.
(4) This property is a consequence of properties (2) and (3).
(5) One consequence of the dominance relation is that Large$(R, i) \succ$ Large(S, i), $0 \le i \le m$, whenever $R \succ S$. Let $j = |\text{Shift}(S, x)|$. Since Large$(R, j) \succ$ Large(S, j) and all of the elements in both of these sets exceed x, then it follows that Shift$(R, x) \succ$ Shift(Large$(R, k), x) \succ$ Shift(Large$(S, k), x) =$ Shift(S, x). Q.E.D

We now state and prove a theorem regarding the optimality of the STE policy for the case of nonpreemptive M/G/1 + G queues where unforced idle times are prohibited.

THEOREM 1. *The STE policy is optimal for the nonpreemptive M/G/1 + G queue with deadlines to the beginning of service when no unforced idle times are allowed and when interarrival times and deadlines are i.i.d. random variables.*

PROOF. For ease in exposition, STE refers to the STE policy. We first introduce a new definition for $V(\pi)$ that is appropriate for the M/G/1 + G queue. Let $M_\pi(i)$ denote the number of customers served in the ith busy period under policy π. Let $X_\pi(i, j)$ denote the service time of the jth customer served during the ith busy period. Let $I_\pi(i)$ denote the length of the idle period following the ith busy period. Last, define $\mu(\pi)$ to be the system throughput:

$$\mu(\pi) = \lim_{n\to\infty} \inf \frac{\mathrm{E}[\sum_{i=1}^{n} M_\pi(i)]}{\mathrm{E}[\sum_{i=1}^{n} (\sum_{j=1}^{M_\pi(i)} X_\pi(i, j)) + I_\pi(i)]}. \tag{1}$$

As a consequence of the assumptions that interarrival times are i.i.d. exponential random variables with mean $1/\lambda$ and that service times are i.i.d. random variables, we can rewrite the above equation

$$\mu(\pi) = \lim_{n\to\infty} \inf \frac{\sum_{i=1}^{n} \mathrm{E}[M_\pi(i)]}{\sum_{i=1}^{n} \mathrm{E}[M_\pi(i)]\mathrm{E}[B_\pi] + 1/\lambda}. \tag{2}$$

Finally, $V(\pi)$ can be expressed in terms of $\mu(\pi)$ as follows,

$$V(\pi) = \frac{\mu(\pi)}{\lambda}.$$

In order to prove that STE is optimal for the M/G/1 + G queue, it suffices to show that $\mathrm{E}[M_{\mathrm{STE}}(i)] \geq \mathrm{E}[M_\pi(i)]$ for all $i \geq 1$. We focus on a single busy period under π. In order to simplify exposition, we assume that this busy period begins at time $t = 0$. We show that $C_{\mathrm{STE}}(t) \succ C_\pi(t)$ during this busy period for every input sample $\mathbf{s}$ and every policy π. Consequently, the number of customers served in this busy period is greater under STE than it is under π. We shall show that this dominance holds when service times are assigned in the order of service.

Consider a single policy π and a single input sample $\mathbf{s}$. We need only focus on the points of time at which either a customer arrives, a customer departs, or a customer misses a deadline in the systems operating under π and STE during the period of time covered by π's busy period. Let $t_0 = 0 \leq t_1 \leq \cdots \leq t_i \leq t_{i+1} \leq \cdots \leq t_n$ denote these times. Here t_n denotes the service completion under π that terminates the busy period.

It is useful to distinguish among the following events:

$\mathscr{E}_1$ – Arrival of a customer at both systems.
$\mathscr{E}_2$ – Service completion at one or both systems.
$\mathscr{E}_3$ – Loss of one or more customers at one or both systems due to missing of deadline.

A more complete description of the history of both systems is given by the sequence of event–time pairs $(t_0, A_0), (t_1, A_1), \ldots, (t_i, A_i), \ldots, (t_n, A_n)$ where $A_0 = \mathscr{E}_1$, $A_n = \mathscr{E}_2$, and t_i is the time at which an event of type $A_i \in \{\mathscr{E}_1, \mathscr{E}_2, \mathscr{E}_3\}$ $(1 \leq i < n)$ occurs. If two types of events occur simultaneously, we present them as separate events with the identical event times. The order in which they are listed in immaterial.

First observe that, whenever $C_{\mathrm{STE}}(t_i) \succ C_\pi(t_i)$ and $t_{i+1} > t_i$, then $C_{\mathrm{STE}}(t) \succ C_\pi(t)$, $t_i \leq t < t_{i+1}$. This is a consequence of property (5) in Lemma 2. The proof that $C_{\mathrm{STE}}(t) \succ C_\pi(t)$ is by induction on the event times $t_0, t_1, \ldots$.

Induction Hypothesis. As both systems are initially in the same state at $t = t_0 = 0$, the relation holds.

Inductive Step. Let us assume that the hypothesis is true for t_k, $k = 0, 1, \ldots, i$. We now show that it is also true for t_{i+1}. There are several cases according to the type of event that occurs at time t_{i+1}.

Case 1 ($A_{i+1} = \mathscr{E}_1$). First, note that $C_{\mathrm{STE}}(t_{i+1}^-) \succ C_\pi(t_{i+1}^-)$. Application of property (1) in Lemma 2 then yields $C_{\mathrm{STE}}(t_{i+1}) \succ C_\pi(t_{i+1})$.

Case 2 ($A_{i+1} = \mathscr{E}_2$). If a service completion occurs under π, then a service completion occurs under STE at the same time. Consequently, if $C_\pi(t_{i+1}^-) \neq \emptyset$, then, according to the inductive hypothesis, $C_{\mathrm{STE}}(t_{i+1}^-) \neq \emptyset$. A customer will be scheduled by each policy and $C_{\mathrm{STE}}(t_{i+1}) \succ C_\pi(t_{i+1})$ according to property (4) from Lemma 2.

Case 3 ($A_{i+1} = \mathscr{E}_3$). There are three subcases according to whether a customer is lost under STE, π, or both policies. Consider the case in which a customer is lost under STE but not π. For this to happen and the inductive hypothesis to hold, $C_{\mathrm{STE}}(t_{i+1}^-)$ must contain at least one more customer than $C_\pi(t_{i+1}^-)$. Consequently, property (2) of Lemma 2 can be applied to show that $C_{\mathrm{STE}}(t_{i+1}) \succ C_\pi(t_{i+1})$.

If $A_{i+1} = \mathscr{E}_3$ corresponds to the loss of a customer under π, then property (3) of Lemma 2 can be used to show $C_{\mathrm{STE}}(t_{i+1}) \succ C_\pi(t_{i+1})$. Similarly, property (4) of Lemma 2 can be used in the case of loss of a customer under both STE and π to show $C_{\mathrm{STE}}(t_{i+1}) \succ C_\pi(t_{i+1})$.

This completes the inductive step. Therefore, we have shown that $C_{\mathrm{STE}}(t) \succ C_\pi(t)$ during a busy period under π. Therefore the number of customers served under STE is at least as large as the number served under π during a single busy period when given any input sample s. As a consequence STE is the optimal policy over all policies that prohibit unforced idle times. Q.E.D

One can construct examples in which the performance of the system can be increased by allowing the policy the option of not scheduling a customer even when one exists in the queue. This can be useful in the situation in which the customers in the queue have deadlines that are substantially longer than the average deadline. In this case the policy may keep the server idle with the expectation that a customer may arrive with a deadline considerably shorter than those of the customers in the queue. We now examine this class of policies.

The following theorem states that for every policy that does not belong to the class of STEI policies, there exists an STEI policy with the same performance. Consequently, the STEI class of policies contains the best policies, that is, those with the highest performance. Thus the designer of a real-time system need only consider this class of policies.

THEOREM 2. *For any policy π, there exists an STEI policy π^* such that $V_N(\pi^*) = V_N(\pi)$ and $V(\pi^*) = V(\pi)$.*

PROOF. Consider any policy π not in the class of STEI policies. We shall construct an STEI policy π^* that exhibits the same performance as that of π.

Policy π^* is defined as follows:

(1) π^* maintains an ordered list of customers at time t, $\mathscr{A}(t)$, which would be eligible under π at that time when provided with the same input sample, that is, $\mathscr{A}(t) = C_\pi(t)$.

(2) π^* maintains a history H_t' identical to the history that π would produce when given the same input sample; that is, $H_t' = H_t$.

(3) π^* makes scheduling decisions according to the following rules:
   (a) At time t, it schedules the customer closest to its deadline with probability $1 - p_0(\pi, t, H_t')$.
   (b) At time t, it schedules no customer with probability $p_0(\pi, t, H_t')$.

(4) π^* modifies $\mathscr{A}(t)$ as follows:
   (a) Customer c is removed from $\mathscr{A}(t)$ (1) when its deadline expires, or (2) with probability $p_c(\pi, t, H_t')$ at a time when π^* schedules a customer.
   (b) Customer c is added to $\mathscr{A}(t)$ when it arrives to the system.

(5) π^* modifies H_t' as follows:
   (a) At the time of an arrival, the arrival time and relative deadline of the customer are added to $\mathbf{a}_t$ and $\mathbf{d}_t$.
   (b) At the time of a departure, the service time of the customer is added to $\mathbf{u}_t$.
   (c) At the time when a customer is assigned to service, that time and the identity of the customer that π^* removes from $\mathscr{A}(t)$ are added to $\mathbf{r}_t$ and $\mathbf{e}_t$, respectively.

We have defined a policy π^* that exhibits the same behavior as π (i.e., $V_N(\pi^*, \mathbf{s}) = V_N(\pi, \mathbf{s})$, $N = 1, 2, \ldots$ and $V(\pi^*, \mathbf{s}) = V(\pi, \mathbf{s})$) provided that $\mathscr{A}(t)$ and $C\pi(t)$ exhibit the same behavior. This latter statement is true if $\mathrm{E}_{\pi^*}(t) \succ \mathrm{E}_{\pi}(t)$ for input sample $\mathbf{s}$. This last dominance relation can be shown to hold for any input sample $\mathbf{s}$ by an induction argument on the times at which a customer enters or leaves the system for every input sample $\mathbf{s}$. This argument is similar to the one used in Theorem 1 and is omitted.

Finally, taking the expectation over all input samples yields $V_N(\pi^*) = V_N(\pi)$ and $V(\pi^*) = V(\pi)$. Q.E.D

COROLLARY 1. *If there exists an optimal policy π for the $G/G/1 + G$ queue, then there exists a STEI policy that is optimal.*

PROOF. This is a consequence of the last theorem.

We conclude this section with a proof that the STE policy is optimal for the discrete time G/D/1 + G queue when the service time is exactly one time unit. This is of practical interest because many data communication systems are modeled by such queues.

THEOREM 3. *The STE policy is optimal for the discrete time $G/D/1 + G$ queue where the service time is exactly one time unit.*

PROOF. Consider any STEI policy π. We construct a sequence of policies (possibly infinite in number) $\pi_0 = \pi, \pi_1, \ldots, \pi_i, \ldots$, such that (1) π_i is an STEI policy, $i = 0, 1, \ldots$, (2) the performance is nondecreasing function of i, and (3) if the sequence is infinite in number, then $\lim_{i\to\infty} \pi_i$ is the STE policy; otherwise, the last policy, say π_n, is the STE policy.

Before we provide the method for constructing the above sequence of policies, we introduce some terminology. First, define $\mathrm{Prefix}(t, \mathbf{s}) = (\mathbf{a}(t), \mathbf{d}(t))$ where $\mathbf{a}(t)$ and $\mathbf{d}(t)$ are the following ordered subsets of $\mathbf{a}$ and $\mathbf{d}$, $\mathbf{a}(t) = \{a_i \mid a_i \in \mathbf{a}, a_i \le t\}$ and $\mathbf{d}(t) = \{d_i \mid d_i \in \mathbf{d}, a_i \le t\}$. In addition, if $\mathscr{S}$ is a set of input samples, then define $\mathrm{Prefix}(t, \mathscr{S}) = \{\mathrm{Prefix}(t, \mathbf{s}) \mid \mathbf{s} \in \mathscr{S}\}$. Next define a function $g(\pi, \mathbf{s})$ whose value is the time at which policy π inserts the first idle time when the input sample is $\mathbf{s}$. Note that, if $g(\pi, \mathbf{s}) = t$, then $g(\pi, \mathrm{Prefix}(t, \mathbf{s})) = t$. Also note that $g(\pi, \mathbf{s}') = t$ for

each input sample $\mathbf{s}'$ such that Prefix(t, $\mathbf{s}'$) = Prefix(t, $\mathbf{s}$). Last, define $g(\pi) = \min_{\mathbf{s}}\{g(\pi, \mathbf{s})\}$. Here $g(\pi)$ is the earliest time that π inserts an idle time for any sample path.

We construct π_{i+1} from π_i in the following way. Choose the set of input samples $\mathcal{S} = \{\mathbf{s} \mid g(\pi_i, \mathbf{s}) = g(\pi_i)\}$. Define $\mathcal{P}(t) = \text{Prefix}(t, \mathcal{S})$. We require π_{i+1} to behave exactly like π_i for $0 \le t < g(\pi_i)$. At time $g(\pi_i)$, π_{i+1} always schedules the customer c closest to its deadline (unlike π_i which may insert an idle time). At time $g(\pi_i) < t$, π_{i+1} behaves exactly like π_i, except whenever π_i schedules the customer c that π_{i+1} previously scheduled at time $g(\pi_i)$. At this time, π_{i+1} does nothing. Policy π_{i+1} exhibits several properties.

(1) π_{i+1} is an STEI policy.

(2) $g(\pi_{i+1}) > g(\pi_i)$.

(3) π_{i+1} behaves exactly the same as π_i on all input samples $\mathbf{s} \notin \mathcal{S}$, $V_N(\pi_i, \mathbf{s}) = V_N(\pi_{i+1}, \mathbf{s})$.

(4) $V_N(\pi_{i+1}, \mathbf{s}) \ge V_N(\pi_i, \mathbf{s})$ for all $\mathbf{s} \in \mathcal{S}$.

As a result of these properties we have $V_N(\pi_{i+1}) \ge V_N(\pi_i)$, $N \ge 0$ and $V(\pi_{i+1}) \ge V(\pi_i)$.

The above procedure can be applied repeatedly resulting in a sequence of policies π_i, $i \ge 0$ ($\pi_0 = \pi$), such that $g(\pi_i)$ is a strictly increasing function of i, $V(\pi_i)$ is a nondecreasing function of i. Therefore, the limiting policy is the STE policy that has as good or better performance than π. Q.E.D

4. *Computation of Loss*

In this section we compare the performance of the STE scheduling policy with that of the FCFS scheduling policy for the M/D/1 queue, where the service time is taken to be one unit of time. We consider the case in which there are two classes of customers. Customers of class i arrive according to a Poisson process with parameter λ_i, $i = 1, 2$. Class 1 customers have a fixed deadline of L time units ($L > 0$), whereas class 2 customers have a fixed deadline M time units longer, that is, a deadline of $(L + M)$ time units ($M \ge 0$). In the remainder of this section we outline the procedure used to compute the losses under the FCFS and STE policies, respectively. We begin by considering the FCFS policy.

Our method for computing losses under the FCFS policy for an M/D/1 queue with customers with two possible deadlines is similar to that used in [4] for the M/G/1/K queue. It involves modeling the system as a Markov chain, where the state is defined as (M_1), where M_1 ($M_1 \le L + M$) denotes the number of customers in the queue. A more complete description is given in [10]. The loss, expressed as a percentage of the total arrival rate, can be computed for various values of λ_1, λ_2, L, and M (see Table I).

We used a somewhat different method to compute the loss for the same system when the STE policy was employed. The system operating under STE was modeled as a continuous time, continuous state Markov process with a two-dimensional state space. This Markov process was approximated as a discrete time, discrete state Markov chain by discretizing time. Specifically, we approximated each time unit by N discrete time units. We approximated the Poisson arrival process as a Bernoulli arrival process on these discrete time points with parameters $p_1 = \lambda_1/N$ and $p_2 = \lambda_2/N$ for class 1 and class 2 arrivals, respectively. This Bernoulli approximation has the effect of producing a *finite state* process. The accuracy of

TABLE I. CUSTOMER LOSS USING FCFS AND STE POLICIES FOR AN M/D/1 QUEUE

				Percentage loss		
		L	$L+M$			
λ_1	λ_2	Type 1	Type 2	FCFS	STE	(N)
0.4	0.4	1	1	19.96	19.88	(60)
0.1	0.1	1	2	1.05	0.73	(70)
0.2	0.2	1	2	4.26	3.33	(60)
0.3	0.3	1	2	9.35	8.02	(60)
0.4	0.4	1	2	15.73	14.33	(60)
0.2	0.2	1	3	4.06	2.35	(50)
0.1	0.3	1	3	2.36	0.98	(50)
0.3	0.1	1	3	5.44	4.22	(50)
0.2	0.2	1	4	4.04	2.05	(50)
0.2	0.2	2	2	1.38	1.37	(50)
0.4	0.4	2	2	10.33	10.17	(50)

this approximation improves with increasing values of N. A reward was associated with each state of the resulting discrete time Markov chain. The rewards were chosen so as to lead to a straightforward computation of the throughput. The resulting Markov chain with reward structure was then solved using the value iteration algorithm [6]. Further details are given in [10]. We can compute the value of the loss for various values of λ_1, λ_2, L, and M. For values of $M = 0$, the STE policy is the same as the FCFS policy. Therefore, we can check the accuracy of the method used to compute the loss under the STE policy by comparing it with that used to obtain the loss under the FCFS policy. For the values of N we utilized, the error never exceeded 0.2% of the throughput (see Table I). Losses were not computed for values of L and M larger than that shown in Table I because of the increased computation time and memory space required to obtain reasonably accurate results.

According to Corollary 1, the optimal scheduling policy for the M/D/1 queue belongs to the STEI class of policies. Though we considered some limited types of STEI policies, we were unable to improve on the losses obtained for an STE policy. The STE policy led to lower values of loss than the FCFS policy whenever $M > 0$. Indeed, the improvement of the STE policy over the FCFS policy tends to increase with M, the difference in deadlines between the two types of customers.

5. *Summary*

We have shown that, for a large class of queues, the best scheduling policies for minimizing the loss of impatient customers belong to the class of STEI policies. The STE policy is optimal for a class of queues if no unforced idle times are allowed. In addition, the STE policy is optimal for the discrete time G/D/1 + G queue, where a service time is one time unit, independent of whether or not unforced idle times are allowed. These results are illustrated by giving numerical values for the losses under various policies for the M/D/1 queue.

REFERENCES

1. BACCELLI, F., AND HEBUTERNE, G. On queues with impatient customers. In *Performance '81*, F. J. Klystra, Ed. North Holland, Amsterdam, 1981, pp. 159–179.
2. DERTOUZOS, M. Control robotics: The procedural control of physical processes. In *Proceedings of the IFIP Congress*, 1974, pp. 807–813.

3. GOLD, B. Digital speech networks. *Proc. IEEE 65* (Dec. 1977), 1636–1658.
4. GROSS, D., AND HARRIS, M. T. *Fundamentals of Queueing Theory.* Wiley, New York, 1974.
5. GRUBER, J. G., AND LE, N. H. Performance requirements for integrated voice/data networks. *IEEE J. Selected Areas Commun. SAC-1*, 6 (Dec. 1983), 981–1005.
6. HOWARD, R. *Dynamic Programming and Markov Processes.* M.I.T. Press, Cambridge, Mass., 1960.
7. JACKSON, J. R. Scheduling a production line to minimize maximum tardiness. Res. Rep. 43, Management Sci. Rep., Univ. of Calif., Los Angeles, 1955.
8. KLEINROCK, L. *Queueing Systems Volume II: Computer Applications.* Wiley, New York, 1976.
9. MOORE, J. M. An *n* job, one machine sequencing algorithm for minimizing the number of late jobs. *Manage. Sci. 15* (1968), 102–109.
10. PANWAR, S. S. Time constrained and multiaccess communications. Ph.D. dissertation. Dept. of Electrical and Computer Electrical Engineering, Univ. of Massachusetts, Amherst, Feb. 1986.
11. PIERSKALLA, W. P., AND ROACH, C. Optimal issuing policies for perishable inventory. *Manage. Sci. 18* (1972), 603–614.
12. PINEDO, M. Stochastic scheduling with release dates and due dates. *Oper. Res. 31* (1983), 559–572.
13. SCHWARTZ, M. *Telecommunication Networks: Protocols, Modeling and Analysis.* Addison-Wesley, Reading, Mass., 1987.
14. SU, Z.-S., AND SEVCIK, K. C. A combinatorial approach to scheduling problems. *Oper. Res. 26* (1978), 836–844.

RECEIVED FEBRUARY 1986; REVISED JANUARY 1987, MARCH 1988; ACCEPTED MARCH 1988