Performance Optimization Based on Analytical Modeling in a Real-Time System with Constrained Time/Utility Functions

Mehdi Kargahi, *Member, IEEE*, Ali Movaghar, *Senior Member, IEEE*

Abstract— We consider a single-processor firm real-time (FRT) system with exponential inter-arrival and execution times for jobs with relative deadlines following a general distribution. The scheduling policy of the system is first-come first-served (FCFS) and the capacity of the system is arbitrary. This system is subject to an arbitrary shaped time/utility function (TUF) which determines the accrued utility of each job according to its completion time. It is considered that the system power consumption at different working states is predetermined for each processor speed. We have proposed an exact analytical method for the calculation of specific performance and power-related measures of the system. The resulting analytical formulations for the performance measures which are functions of the processor speed and system capacity are then optimized through appropriate selection of the former parameter using derivatives and the latter parameter employing numerical search methods. Some experimental results are presented for different unimodal TUFs in systems with deterministic and exponential relative deadlines. For the latter distribution, the results are compared against similar results obtained through simulation for the non-preemptive earliest-deadline-first (NP-EDF) scheduling policy. The comparisons show that FCFS is superior to NP-EDF for some measures and TUFs.

Index Terms— Analytical modeling, Firm real-time system, Optimization, Performance modeling, Time/utility function.

1 Introduction

Many emerging real-time embedded and mobile systems such as those related to applications in the space and defense domains are used in critical and harsh environments with uncertain properties. These uncertainties are also applied to less critical systems such as multimedia devices and applications. Such non-determinisms which can be observed in the arrival pattern of jobs as well as job characteristics (e.g., execution times and deadlines) are usually described by stochastic models.

One important issue in many such real-time systems is the determination of their time constraints. As indicated in [30], there exist two criteria for the execution of a real-time job; one is known as urgency and is specified by its deadline and the other is known as importance and is determined by the exact instant of time that the job completes its execution. Classical studies on real-time scheduling algorithms have concentrated on the former criterion through deadline-based decision makings. For example, in earliest-deadline-first (EDF) [21] the job with the earliest deadline is the one with the highest priority and in minimum-laxity-first (MLF) [26] the job with the minimum laxity time (the difference between the time left to the job's deadline and its remaining execution time) has the highest priority. However, a number of studies focus on the latter criterion, namely the importance of job execution [6], [7], [19], [20], [22], [33]. This criterion is usually characterized by the job's time/utility function (TUF), as proposed for the first time by Jensen et al. [14].

According to the traditional classification of real-time systems [5], systems in which jobs meeting their deadlines attain a utility of 100% and those missing their deadlines can continue their execution with a decreasing completion-time dependent utility are categorized as soft real-time (SRT); while systems in which jobs missing their deadlines are of no value and are usually thrown away immediately after their deadlines are called firm real-time (FRT). This traditional classification can be generalized using TUFs. Indeed, a TUF, which specifies the semantics of SRT systems in a more precise manner, determines the utility resulting from the execution of a job as a function of its completion time (possibly before and/or after its deadline), where the TUF is not necessarily a decreasing function. Similar relations can also be defined between TUFs and FRT systems which are discussed in the following subsection.

In general, TUFs can be classified into unimodal and multimodal functions. Unimodal TUFs are those in which any decrease in utility cannot be followed by an increase. Multimodal TUFs do not follow this constraint. TUF for the classical deadline is a binary-valued, downward step

- *M. Kargahi is with the School of Electrical and Computer Engineering, College of Engineering, University of Tehran, P.O. Box 14399-57131, Tehran, Iran. E-mail: kargahi@ut.ac.ir. His second affiliation is School of Computer Science, Institute for research in fundamental sciences (IPM), P.O.Box 19395-5746, Tehran, Iran.*
- *A. Movaghar is with the Department of Computer Engineering, Sharif University of Technology, P.O. Box 11365-9517, Tehran, Iran. E-mail: movaghar@sharif.edu.*

Manuscript received (Jan. 5, 2009).

function. A number of sample TUF shapes are presented in [34] and the references therein for different applications. For example, brief discussions on the Airborne WArning and Control System (AWACS) surveillance mode tracker system and a coastal air defense system can be found in [33] and [40]. Some notes on TUF shapes in mobile ad-hoc network applications can also be found in [11].

In TUF-constrained systems, completing the time-constrained jobs as close as possible to their optimal completion times is desired. In such systems, an important optimality criterion, known as utility accrual (UA), is based on the accrued utility, e.g., maximizing the sum of attained utilities by the real-time jobs or assuring satisfaction of lower bounds on the individual jobs' attained utilities. In the environments with uncertain characteristics, such performance measures are usually expressed probabilistically. Even though the most general TUFs are multimodals, most of the studies on the UA scheduling algorithms consider simpler TUFs, e.g., step TUFs or special cases of non-step unimodals.

The real-time embedded and mobile systems mentioned above have another major restriction, namely the limitation of their energy resources. Recently, a vast number of studies have focused on the reduction of the energy consumption of such systems through techniques such as dynamic voltage scaling (DVS) or adaptive body biasing (ABB) [42], regarding their performance. With DVS, an appropriate task or job dependent dynamic clock frequency and voltage selection can be used for quadratic energy saving at the expense of, roughly, linearly increased job response time. Most of these studies such as [3], [17], [23], [29], and [43] have concentrated on deadline-based scheduling algorithms with their respective optimality criteria, whereas some studies such as [2], [34], [37], [38], and [41] present scheduling algorithms that consider the optimality of UA criteria.

1.1 Motivation and Paper Outline

In this study, we consider FRT systems with non-preemptive TUF-constrained independent jobs. TUF-constrained FRT systems can be defined as systems in which jobs missing their deadlines should be thrown away immediately, however, the time of the successful completion of a job affects the importance of the job completion. This importance is based on the job's TUF which reaches zero on the respective deadline. Few studies such as [6] and [33] have considered non-preemptive jobs in a TUF-constrained system, where TUF shapes are too restrictive. In the current study, however, TUFs are allowed to take arbitrary shapes. Furthermore, we consider the class of non-idling service-time independent scheduling algorithms that, when the system capacity is not full, do not decide on accepting or rejecting a job on its arrival. In particular, we concentrate on the first-come first-served (FCFS) scheduling algorithm. We are interested in the calculation of some performance and power-related measures of the system. For the calculation of the favorite power-related measure, the energy characterization of a real CPU, namely PXA 270 processor [13] is taken into account. We use the specification of the processor with the DVS capability only for the calculation of the power usage of the system, i.e., no dynamic selection of speeds is considered in the current study. (Throughout this paper we use the terms speed and service rate interchangeably.)

Thus, the contributions of this paper are as follows. Even though, we have not proposed a new scheduling algorithm, a precise analytical method for the calculation of some performance and power-related measures are presented for the FRT system with arbitrary shaped TUFs. The utility perforamce measures are the expected values of the accrued utility for all jobs as well as jobs meeting their deadlines, job accrued utility per unit of energy, and the assurance level of satisfying a predetermined lower-bound on the percentage of job accrued utility. Moreover, since the optimal configuration of the system can usually be affected by different TUFs, the resulting analytical formulations may be used for finding the optimal processor speeds as well as optimal system capacities in order to maximize the above mentioned measures. The optimal processor speed for a specified system capacity is computed using derivatives and the optimal capacity for the speed-optimized system is obtained through a numerical search method. Finally, it is shown that the system with an optimal configuration outperforms non-preemptive EDF (NP-EDF) for some TUFs. To the best of the authors' knowledge, there exists no precise analytical method for the evaluation and parameter setup of a similar system with TUF-constrained jobs.

The rest of this paper is organized as follows. The system and application models as well as the respective performance measures are presented in Section 2. The queueing model, its solution, and the details of calculation and optimization of our favorite performance measures are described in Section 3. The experimental setup and evaluation, and the comparison of FCFS with respect to NP-EDF are provided in Section 4. Section 5 discusses the related works. Finally, concluding remarks are presented beside some notes on future work in Section 6.

2 System Model and Performance Measures

This section first identifies the system and application models, and then, presents the formulations of our favorite performance measures. Throughout this paper, we assume statistical equilibrium and use τ and ω to denote variables with values in the set of non-negative real numbers.

2.1 The System, Job and Utility Models

We consider a single-processor queue with an arbitrary capacity K ($K \le \infty$) and a service rate of μ. Jobs of this system are defined as $J=(a,e,\theta,U)$, where a, e, θ, and U are the job's arrival time, execution time, relative deadline, and TUF, respectively. A state of the system is shown with n, where $n \le K$ is the number of jobs in the system. These parameters are defined more precisely in the following paragraphs.

The job arrival times (a) follow a Poisson process with rate λ. An arriving job which finds the system full (i.e., finds n=K) is blocked and must leave the system immediately. Jobs entering the system are served in the order of their arrival, i.e., the service discipline in the system is FCFS.

Each job has an exponential execution time (e) with an expected value $\bar{e}$. Throughout this paper, all times are normalized with respect to $\bar{e}$.

Further, each job has a deadline. The difference between the deadline of a job and its arrival time, referred to as a relative deadline, is a random variable θ with a cumulative distribution function (CDF) $G(\tau)$. We assume that $G(\tau)$ is a general CDF which may have a mass at the infinity, i.e., in general, $P(\theta=\infty)=1-\lim_{t\to\infty}G(t)\geq 0$. A job must leave the system as soon as it misses its deadline irrespective of whether or not it is being served, i.e., each job has a deadline until the end of its service. Job service times and relative deadlines form sequences of independently and identically distributed (i.i.d.) random variables which are mutually independent.

As indicated in Section 1, we consider TUF-constrained FRT jobs. More precisely, the importance of meeting the deadline of a job strongly depends on the instant of time that the job completes its service. This importance is specified by the job's TUF, namely $U(\tau,\theta)$ as defined below

$$U(\tau,\theta)\equiv \text{the utility of the successful service completion of a job at time } a+\tau, \quad (1)$$

where $0\leq\tau\leq\theta$ is the sojourn time (response time) of that job, namely the time between the arrival and service completion of the job. (We use the terms sojourn time and response time interchangeably throughout this paper.) $U(.)$ is the same function for all jobs of the system. The TUF of each job can take non-zero values only between the arrival time and deadline of that job, namely in the interval $[a, a+\theta]$.

As also indicated in some previous studies such as [2], [34], [37], [38], and [41], changing the processor speed (and voltage) level can affect the accrued utility of the jobs in the system. However, each processor has a predefined power function $P^{processor}(\mu)$, which can be considered as the energy consumption of the processor at speed μ, normalized with respect to the time unit. This power function may differ according to the working state (e.g., idle or busy) of the processor. A similar power function, depicted as $P^{memory}(\mu)$, can be considered for the memory unit which is affected by the capacity as well as working state of the memory unit. The values of these functions may be influenced by the number of available jobs in the system, as discussed in a more precise manner in Subsection 3.3. Therefore, the total power consumption of the system is computed as

$$P(\mu)=P^{processor}(\mu)+P^{memory}(\mu). \quad (2)$$

We will investigate some tradeoffs between the accrued utility and power consumption of the system in the following sections. In our study, without loss of generality, we will ignore the power consumption of the memory unit, but consider the possible differences between processor idle and busy states.

2.2 The System Performance Measures

This subsection introduces our favorite system performance measures. We begin with the definition of our principal performance variable. Let

$$V\equiv \text{the time an arriving job with infinite (no) deadline, in the long run, must wait before it completes its service.} \quad (3)$$

V is called the job offered sojourn (response) time. We assume $V=\infty$ if the arriving job is blocked due to full system. We will be interested in finding the CDF of V

$$F_V(\tau)=P(V\leq\tau), \quad (4)$$

or, equivalently, the probability density function (PDF)

$$f_V(\tau)=\frac{dF_V(\tau)}{d\tau}. \quad (5)$$

More specific measures of performance may also be defined using the PDF of V. In particular, we will be interested in the probability of missing deadline, defined as

$$\alpha_d=P(\theta<V<\infty)=\int_0^\infty f_V(\tau)G(\tau)d\tau. \quad (6)$$

α_d represents the steady-state probability that a job misses its deadline. Another important measure of performance is the probability of blocking α_b, defined as

$$\alpha_b=P(V=\infty)=1-\lim_{t\to\infty}F_V(t). \quad (7)$$

α_b is interpreted as the steady-state probability that an arriving job is rejected due to full system. Further, the probability of loss may be derived as

$$\alpha=\alpha_d+\alpha_b=P(V>\theta)=\int_0^\infty f_V(\tau)G(\tau)d\tau+P(V=\infty). \quad (8)$$

α is viewed as the steady-state probability that a job is lost due to either missing its deadline or being rejected because of a full system.

The above performance measures assume no significance for the completion instant of a job that meets its deadline. Rather, as indicated before, the instant of time is quite effective on the performance of many applications. For a large number of applications, the accrued utility at the completion instant of time is determined by the respective TUF. In the following, we consider this property as the basis for the further performance measures. Let

$$\Omega\equiv \text{the long-run accrued utility of an arriving job.} \quad (8)$$

Ω is called the job offered accrued utility. The job expected accrued utility $\overline{\Omega}$, as the first utility performance measure in this paper, can be obtained as

$$\overline{\Omega}=\int_0^\infty f_V(\tau)\left(\int_\tau^\infty g(x)U(\tau,x)dx\right)d\tau, \quad (9)$$

where $U(.)$ is defined as in (1) and $g(.)$ is the PDF of the random variable θ, i.e., $G(\tau)=\int_0^\tau g(x)dx$. $\overline{\Omega}$ is interpreted as the steady-state utility which is expected to be accrued by a job. The inner integral in the right hand side of (9) is the expected utility of a successful job with a response time τ (which meets its deadline) and the outer integral calculates the overall expected value of the same measure. Assuming $U(\tau,\theta)=1$ for all values of relative deadline θ and response time $\tau\leq\theta$ (binary-valued, downward step TUF), the probability of meeting deadline, i.e., $1-\alpha$ is calculated through (9).

The second utility performance measure that we consider is $\overline{\Omega}_{succ}$ which defines the expected accrued utility for the successful jobs, namely the jobs meeting their deadlines. This measure expresses the quality of the successful jobs. It may be preferable for an application to have less jobs meeting their deadlines, but each with a utility closer to its optimal value. This may be the case for many cost-effective (such as military and space related) applications. $\overline{\Omega}_{succ}$ is calculated as

$$\overline{\Omega}_{succ}=\frac{\overline{\Omega}}{1-\alpha}=\frac{\int_0^\infty f_V(\tau)(\int_\tau^\infty g(x)U(\tau,x)dx)d\tau}{\int_0^\infty f_V(\tau)(1-G(\tau))d\tau}. \qquad (10)$$

For the third utility performance measure, we first define

$$\delta(\tau,\theta)=\frac{U(\tau,\theta)}{u_{\max}(\theta)}\equiv \text{the utility-ratio of the successful service completion of a job at time } a+\tau, \qquad (11)$$

where $0\le\tau\le\theta$ and

$$u_{\max}(\theta)=\max_{\tau\in[0,\theta]}U(\tau,\theta) \qquad (12)$$

is the maximum possible utility of a job with a relative deadline θ. Consider

$$\Delta\equiv \text{the long-run utility-ratio of an arriving job}. \qquad (13)$$

We will be interested in finding the CDF of Δ

$$F_\Delta(\omega)=P(\Delta\le\omega), \qquad (14)$$

or, equivalently, the PDF

$$f_\Delta(\omega)=\frac{dF_\Delta(\omega)}{d\omega}. \qquad (15)$$

As the third utility performance measure, we consider $A(v)$ as the assurance level of satisfying a lower bound $v\in[0\%,100\%]$ on the job offered utility-ratio Δ. More precisely, for a given PDF $g(.)$ for the job relative deadline, $A(v)$ defines the probability that the attained utility by a job with a relative deadline θ is at least $vu_{\max}(\theta)$. In this regards, the performance measure $A(.)$ can be calculated as

$$A(v)=P(\Delta\ge v)=1-F_\Delta(v). \qquad (16)$$

The forth utility performance measure of interest is the expected job accrued utility-power ratio (UPR), defined as

$$\overline{\Theta}=\int_0^\infty\frac{f_V(\tau)}{P(\mu)}(\int_\tau^\infty g(x)U(\tau,x)dx)d\tau. \qquad (17)$$

This measure defines the expected job accrued utility per unit of energy at an specified service rate μ, which determines how well the energy resources are used for utility accrual.

The precise calculations for the derivation of the above performance and power-related measures are presented in the following section.

3 Queueing Model

In the following, in Subsection 3.1, the formulations of some important conditional performance variables of the FRT system are presented. Then, in Subsection 3.2, a solution for the queueing model of the system is provided beside the way of calculating some elementary performance measures. Afterwards, in Subsection 3.3., some complementary calculations are carried out for the utility performance measures discussed in Subsection 2.2 and the optimization scenario is also described.

3.1 Conditional Performance Variables

We begin with some notations which are used throughout this paper. Let $\mathcal{N}$ be the set of natural numbers and $\mathcal{R}^+$ the set of positive real numbers. For $t,\varepsilon\in\mathcal{R}^+$ and $n\in\mathcal{N}$, let

$$\Psi_n(t,\varepsilon)\equiv \text{the probability that one of the jobs in the system misses its deadline during } [t,t+\varepsilon), \text{ given there are } n \text{ jobs in the system at time } t. \qquad (18)$$

Define

$$\gamma_n(t)=\lim_{\varepsilon\to 0}\frac{\Psi_n(t,\varepsilon)}{\varepsilon}, \qquad (19)$$

$$\gamma_n=\lim_{t\to\infty}\gamma_n(t). \qquad (20)$$

γ_n is the (steady-state) rate of missing deadlines, given there are n jobs in the system (including the one being served).

Barrer [4] first derived an expression for the parameter γ_n, in terms of μ and θ, for a model with a deterministic job relative deadline, infinite capacity, Poisson arrival process, and FCFS scheduling algorithm. Barrer's result is extended to a larger class of models, namely those with a general job relative deadline, an arbitrary capacity, and a state-dependent Poisson arrival process in [28]. The concept of this parameter has also been used for approximating the performance of systems with EDF scheduling algorithm in both preemptive [16] and non-preemptive [15] cases.

In [28], Movaghar has derived a closed-form solution for the PDF of the job offered sojourn time, given the number of jobs in the system. Let,

$$V_n\equiv \text{the time an arriving job with infinite (no) deadline, in the long run, must wait before it completes its service, given it finds } n \text{ jobs in the system}. \qquad (21)$$

V_n is called the job offered sojourn-time, given there are n jobs in the system. Let

$$F_{V_n}(\tau)=P(V_n\le\tau), \qquad (22)$$

$$f_{V_n}(\tau)=\frac{dF_{V_n}(\tau)}{d\tau}. \qquad (23)$$

Define $\Phi_n(s)$ to be the Laplace transform of $\left[\int_0^\tau(1-G(x))dx\right]^n$, i.e.,

$$\Phi_n(s)=\int_0^\infty[\int_0^\tau(1-G(x))dx]^n e^{-s\tau}d\tau. \qquad (24)$$

As shown in [28], $f_{V_n}(\tau)$ defined above can be obtained as

$$f_{V_n}(\tau) = \frac{1}{\Phi_n(\mu)} \left[\int_0^\tau (1 - G(x)) dx \right]^n e^{-\mu\tau}. \tag{25}$$

Similarly, it has been proven that a closed-form solution to the loss rate γ_n, as defined in (20), is as follows

$$\gamma_0 = 0, \qquad \gamma_n = n \frac{\Phi_{n-1}(\mu)}{\Phi_n(\mu)} - \mu, \quad \text{if } n > 0. \tag{26}$$

3.2 Model Solution and Elementary Performance Measures

In this subsection, we present a Markovian view to the long-run behavior of the queueing system considered in Section 2. The resulting model will be solved analytically using some standard Markovian solution techniques. Let

$$\pi_n(t) \equiv \text{The probability that there are } n \text{ jobs in the systems at time } t. \tag{27}$$

Assuming K as the capacity of the system, as defined in Subsection 2.1, we can write

$$\begin{aligned} \frac{d\pi_0(t)}{dt} &= -\lambda\pi_0(t) + (\mu + \gamma_1(t))\pi_1(t), \\ \frac{d\pi_n(t)}{dt} &= \lambda\pi_{n-1}(t) - (\lambda + \mu + \gamma_n(t))\pi_n(t) \\ &\quad + (\mu + \gamma_{n+1}(t))\pi_{n+1}(t), \qquad \text{if } 0 < n < K. \end{aligned} \tag{28}$$

Let

$$\pi_n = \lim_{t\to\infty} \pi_n(t). \tag{29}$$

Then, in equilibrium, (28) can be rewritten as follows

$$\begin{aligned} 0 &= -\lambda\pi_0 + (\mu + \gamma_1)\pi_1, \\ 0 &= \lambda\pi_{n-1} - (\lambda + \mu + \gamma_n)\pi_n + (\mu + \gamma_{n+1})\pi_{n+1}, \quad \text{if } 0 < n < K. \end{aligned} \tag{30}$$

Solving the equilibrium, we obtain

$$\pi_n = \frac{\lambda^n}{\prod_{i=1}^{n} (\gamma_i + \mu)} \pi_0, \qquad \text{if } 0 < n \le K. \tag{31}$$

Using (26), we get

$$\pi_n = \frac{\mu\lambda^n}{n!} \Phi_n(\mu) \pi_0, \qquad \text{if } 0 < n \le K, \tag{32}$$

where $\Phi_n(\mu)$ is defined as in (24). π_n is the steady-state probability that an incoming job finds n jobs in the system. The normalizing condition is

$$\sum_{n=0}^{K} \pi_n = 1. \tag{33}$$

From (32) and (33) above, we obtain

$$\pi_0 = \left(1 + \sum_{n=1}^{K} \frac{\mu\lambda^n}{n!} \Phi_n(\mu) \right)^{-1}. \tag{34}$$

Using some algebra on (34) for the case of infinite capacity, we get

$$\pi_0 = \frac{1}{\mu \int_0^\infty e^{\lambda \int_0^\tau (1-G(x))dx - \mu\tau} d\tau}. \tag{35}$$

Given $f_{V_n}(\tau)$ as in (25), the PDF of job offered sojourn time can then be obtained as

$$f_V(\tau) = \sum_{n=0}^{K-1} \pi_n f_{V_n}(\tau). \tag{36}$$

In the case of infinite capacity, we can write

$$\begin{aligned} f_V(\tau) &= \mu\pi_0 e^{\lambda \int_0^\tau (1-G(x))dx - \mu\tau} \\ &= \frac{e^{\lambda \int_0^\tau (1-G(x))dx - \mu\tau}}{\int_0^\infty e^{\lambda \int_0^\tau (1-G(x))dx - \mu\tau} d\tau}. \end{aligned} \tag{37}$$

Having $f_V(\tau)$ and using (6), the probability of missing deadline can simply be calculated. Meanwhile, given γ_n as in (26), the probability of missing deadline in the system can also be derived as

$$\alpha_d = \frac{\sum_{n=1}^{K} \pi_n \gamma_n}{\sum_{n=0}^{K} \pi_n \lambda} = \frac{\sum_{n=1}^{K} \pi_n \gamma_n}{\lambda}, \tag{38}$$

which is the average deadline miss rate divided by the average job arrival rate. The probability that an incoming job finds that the system is full, i.e., the blocking probability is simply

$$\alpha_b = \pi_K. \tag{39}$$

(Note that $\alpha_b = 0$ for a system with infinite capacity.) Using some algebra on (38) with the consideration of (26) and (32), we can rewrite (38) as

$$\alpha_d = 1 - \frac{\mu}{\lambda}(1 - \pi_0) - \alpha_b. \tag{40}$$

3.3 Utility Performance Measures and Model Optimization

This subsection presents the details of the calculations for the utility performance measures previously defined in Subsection 2.2 and the optimization scenario for these measures. The main goal of the optimization is to find the best service rates (processor speeds) to maximize such measures. Next, the optimal service rates can be used to find the optimal system capacity among some conceivable system capacities for maximizing each measure.

Utility Performance Measures. Given a PDF $g(.)$ for the relative deadline of arriving jobs, we can calculate the PDF of the job offered sojourn time $f_V(\tau)$ through (36) (or (37) for infinite capacity) by following the solution method presented in Subsection 3.2. Subsequently, given an specified TUF $U(.)$ for the jobs, the job expected accrued utility $\overline{\Omega}$, as defined in (9), as well as the expected accrued utility for the successful jobs, namely $\overline{\Omega}_{succ}$, as defined in (10), can be calculated. Meanwhile, since $\alpha = \alpha_d + \alpha_b$ can simply be obtained through (40), $\overline{\Omega}_{succ}$ as in (10) can be rewritten as

$$\overline{\Omega}_{succ} = \frac{\overline{\Omega}}{\frac{\mu}{\lambda}(1 - \pi_0)}, \tag{41}$$

where π_0 can be calculated through (34) (or (35) for infinite capacity).

For the calculation of $A(\nu)$, $\nu \in [0\%, 100\%]$, as the assurance level of satisfying a lower bound ν on the job offered utility-ratio, we need to have the PDF of the long-run utility-ratio of arriving jobs, namely $f_{\Delta}(.)$.

In order to find the value of $f_{\Delta}(.)$ at some specific utility-ratio ω, we first define a set $S(\omega,\theta)=\{\tau \mid \delta(\tau,\theta)=\omega\}$, where the definition of $\delta(\tau,\theta)$ is given in (11). This set consists of all the relative times with respect to the arrival time of a job with relative deadline θ, at which the accruable utility-ratio is ω. For a given TUF $U(.)$, this set may be constructed from two discrete and continuous subsets of relative times, shown with $S^{\mathrm{D}}(\omega,\theta)$ and $S^{\mathrm{C}}(\omega,\theta)$, respectively. Thus, we have

$$S(\omega,\theta)=S^{\mathrm{D}}(\omega,\theta)\cup S^{\mathrm{C}}(\omega,\theta). \tag{42}$$

In this regards, the PDF of Δ for the discrete portion can be calculated as

$$f_{\Delta}^{\mathrm{D}}(\omega)=\int_0^{\infty}\sum_{\tau\in S^{\mathrm{D}}(\omega,x)} f_V(\tau)g(x)dx. \tag{43}$$

For the continuous portion, it can be obtained as

$$f_{\Delta}^{\mathrm{C}}(\omega)=\int_0^{\infty}\int_{S^{\mathrm{C}}(\omega,x)} f_V(\tau)g(x)d\tau\, dx. \tag{44}$$

Summing these two portions together, the PDF of job offered utility-ratio Δ is calculated as

$$f_{\Delta}(\omega)=f_{\Delta}^{\mathrm{D}}(\omega)+f_{\Delta}^{\mathrm{C}}(\omega). \tag{45}$$

Next, using (45) and (16), the assurance level of satisfying at least a fraction $\nu\in[0\%,100\%]$ of the maximum possible utility of arriving jobs in the long run, i.e., $A(\nu)$ can be calculated. Some clarifying examples on this matter are presented in Section 4.

As mentioned earlier, many modern processors can operate in different speed levels. Even though these speed levels are discrete in nature, we can approximate any service rate in a continuous range by switching between its two adjacent speeds using DVS [17]. However, in order to avoid losing the main goals of this paper, and due to the fact that we have mentioned in Section 1 that dynamic speed selection is not studied in the current paper, we simply assume continuous speed levels. Nevertheless, according to the method presented in [17], the usage of discrete speed levels is also straightforward. We assume an instantaneous power function $P(\mu;t)$ for our system, which determines the power usage of the system at time t with a processor working at a service rate μ.. Depending on the instantaneous population of the system, this function can be calculated as

$$P(\mu;t)=\sum_{n=0}^{K}\pi_n(t)\left(P_n^{processor}(\mu)+P_n^{memory}(\mu)\right), \tag{46}$$

where $P_n^{processor}(\mu)$ and $P_n^{memory}(\mu)$, respectively, are the power consumption of the processor and memory at service rate μ when n jobs are in the system. $P_n^{processor}(\mu)$ may differ when the processor is idle (n=0) or busy ($n\geq1$). $P_n^{memory}(\mu)$ may also differ based on the number of jobs which their code and data are stored in the memory. Let

$$P(\mu)=\lim_{t\to\infty}P(\mu;t). \tag{47}$$

$P(\mu)$, as defined in (47), is the long-run rate of energy depletion of the system. Then, in steady state, (46) can be rewritten as follows

$$P(\mu)=\sum_{n=0}^{K}\pi_n\left(P_n^{processor}(\mu)+P_n^{memory}(\mu)\right). \tag{48}$$

In this regards, for a given PDF $g(.)$ for the job relative deadline and an specified TUF $U(.)$, the expected job accrued utility-power ratio (UPR), namely $\overline{\Theta}$, can be obtained using (17) , (36) (or (37) for infinite capacity), and (48).

Model Optimization. All of the above mentioned performance measures are functions of the service rate μ and the system capacity K. First, we assume that the capacity K is given. In order to maximize the expected values of the accrued utility for all jobs $\overline{\Omega}$ as well as successful jobs $\overline{\Omega}_{succ}$, the service rates satisfying the equalities $\partial\overline{\Omega}/\partial\mu=0$ and $\partial\overline{\Omega}_{succ}/\partial\mu=0$ should be computed as the optimal service rates (μ_{opt}). For the assurance level of satisfying a lower bound ν on the job offered utility-ratio $A(\nu)$, the optimal value of the service rate, i.e., μ_{opt} should be computed through solving the equation $\partial A(.)/\partial\mu=0$. However, sometimes it is needed to guarantee a statistical performance requirement for the system as $A(\nu)\geq A_0$ by selecting an appropriate processor speed, where A_0 is a predetermined threshold [38]. If the minimum statistical performance requirements determined by the threshold cannot be satisfied through appropriate selection of the service rate, the objective is usually to maximize the expected job accrued utility $\overline{\Omega}$ as described above. Finally, the same scenario, namely finding the optimal service rate via solving $\partial\overline{\Theta}/\partial\mu=0$, will result in the maximization of the expected job accrued UPR $\overline{\Theta}$. Since these equations may be hard to solve analytically (depending on the complexity of the TUF and the distribution of relative deadlines), we have used some numerical methods to find the optimal service rate μ_{opt} in the studied cases, as indicated in Section 4.

Now, assume that K can also be changed. After finding the optimal speeds for each utility performance measure and some conceivable system capacities (K), we can use a simple numerical search method to find the optimal value of K for that measure. The experimental evaluation presented in Section 4 will further illustrate this matter.

4 Experimental Evaluation

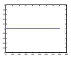
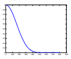
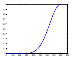
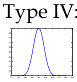
In this section, we present some numerical examples for the comparative evaluation of the discussed utility performance measures for two popular scheduling policies. In Subsection 4.1, the parameter settings, some important calculations for the settings, and some optimization methods which are used in the evaluations are specified. Afterwards, in Subsections 4.2 and 4.3, the numerical results for two distributions of relative deadlines are discussed.

4.1 Experimental Framework

The experiments have been done for both FCFS and NP-EDF scheduling policies. NP-EDF is an optimal policy within the class of non-idling service-time independent non-preemptive scheduling policies [9], [31]. The scheduling policies are compared together with respect to the utility performance measures for different capacities of the queueing system as well as different TUFs.

TABLE 1

$U(.)$ FOR TUF TYPES I, II, III, AND IV AND THE BASIC FORMUKATIONS FOR THE RESPECTIVE $f_\Delta(\omega)$ AS WELL AS $A(\nu)$ FOR $\nu = 80\%$

TUF Type & Shape	$U(.)$	$S^{D}(\omega,\theta)$	$S^{C}(\omega,\theta)$	$f_\Delta^{D}(\omega)$	$f_\Delta^{C}(\omega)$	$A(80\%) = P(\Delta \geq 80\%)$
Type I: 	1	$\{\}$	$[0,\theta]$	0	$\int_0^\infty \int_0^x f_V(\tau) g(x) d\tau dx$	$\int_0^\infty \int_0^x f_V(\tau) g(x) d\tau dx$
Type II: 	$\cos^{10}(\pi t / 2\theta)$	$\{\frac{2\theta}{\pi} \arccos \sqrt[10]{\omega}\}$	$\{\}$	$\int_0^\infty f_V(\frac{2x}{\pi} \arccos \sqrt[10]{\omega}) g(x) dx$	0	$\int_0^\infty \int_{\frac{2x}{\pi} \arccos \sqrt[10]{1} = 0}^{\frac{2x}{\pi} \arccos \sqrt[10]{0.8} = 0.1340x} f_V(\tau) g(x) d\tau dx$
Type III: 	$\sin^{10}(\pi t / 2\theta)$	$\{\frac{2\theta}{\pi} \arcsin \sqrt[10]{\omega}\}$	$\{\}$	$\int_0^\infty f_V(\frac{2x}{\pi} \arcsin \sqrt[10]{\omega}) g(x) dx$	0	$\int_0^\infty \int_{\frac{2x}{\pi} \arcsin \sqrt[10]{0.8} = 0.8660x}^{\frac{2x}{\pi} \arcsin \sqrt[10]{1} = x} f_V(\tau) g(x) d\tau dx$
Type IV: 	$\sin^{10}(\pi t / \theta)$	$\{\frac{\theta}{\pi} \arcsin \sqrt[10]{\omega},$ $\theta - \frac{\theta}{\pi} \arcsin \sqrt[10]{\omega}\}$	$\{\}$	$\begin{cases} \int_0^\infty f_V(\frac{x}{2}) g(x) dx, & \omega = 1 \\ \int_0^\infty \Big[f_V(\frac{x}{\pi} \arcsin \sqrt[10]{\omega}) + & \omega \neq 1 \end{cases}$ $f_V(x - \frac{x}{\pi} \arcsin \sqrt[10]{\omega}) \Big] g(x) dx$	0	$\int_0^\infty \int_{0.4330x}^{0.567x} f_V(\tau) g(x) d\tau dx$

The parameter values are as follows: the offered load to the system $\lambda = 1$, the mean relative deadline $\overline{\theta} = 8$, and the minimum desired utility-ratio $\nu = 80\%$. Furthermore, the experiments have been done for two distributions of relative deadlines, namely deterministic (for which EDF mimics FCFS) and exponential.

We consider four unimodal TUFs, namely, binary-valued downward step, non-increasing, non-decreasing, and bell-shaped functions, referred to as Types I, II, III, and IV, respectively. TUF Type I is the classical deadline. AWACS tracker [33] is an example for functions similar to TUF Type II. As examples for TUF Type III, we can refer to many forecasting systems (e.g., weather, earthquake, stock price, etc.) that the time at which the results are needed is the deadline, after which the forecasting is of no utility. Further, as the time goes ahead, the gathered information for the forecasting are more accurate and updated, and therefore, the results are more valuable. The coastal air defense system [33] is also an example for functions with one peak, similar to TUF Type IV. $U(.)$ for these TUFs are summarized in Table 1. All these functions are bounded to the utility range of [0, 1] and tried to have similar formulations. However, any other utility function similar to the ones presented in [35] and [40] (which are based on a formulation proposed in [22]) could also be used.

Meanwhile, the basic formulations for the PDF of the job offered utility-ratio Δ which are used in the calculation of $A(\nu)$ are presented in Table 1. Since $A(\nu)$, as defined in (16), is desired for $\nu = 80\%$, we are interested in the values of the PDF $f_\Delta(\omega)$ for only $\omega \in [80\%, 100\%]$. As can be seen in the table, for TUF Type I, we have a continuous interval for the relative times with the value of $\omega = 100\%$, namely the interval $[0,\theta]$ for a job with a relative deadline θ. Then, $f_\Delta^{C}(\omega)$ can be obtained using (44). However, for TUF Types II and III, only one discrete relative time exists for each value of ω in the above mentioned range, for which the value of $f_\Delta^{D}(\omega)$ can simply be obtained through (43). For TUF Type IV, two discrete relative times exist for each ω in the mentioned range, except for $\omega = 100\%$, for which one discrete relative time exists. (See the depicted TUF shapes for better realization.) According to the formulation $A(80\%) = P(\Delta \geq 80\%)$, for each TUF type, the summation of the PDFs for the relative times τ with $\omega = \delta(\tau, x) \geq 80\%$ for each relative deadline x should be calculated, where x is a relative deadline inside the integrals in the last column of Table 1. In this regards, the reasoning for the bounds of the inner integrals in that column is specified.

For the power-related measure, we ignore the power consumption of the memory unit, i.e., we assume $P_n^{memory}(\mu) = 0$. However, the possible difference between the processor idle (n=0) and busy (n≥1) power usage is taken into account. The processor power model used in our experiments has been extracted from the data sheets of the Intel XScale PXA270 processor [13]. The speed levels (s) in MHz, normalized service rates with respect to the maximum processor speed (μ_{Norm}), as well as the actual power consumption of the processor in active (pow_A) and idle (pow_I) states, both in Watts, are summarized in Table 2. As indicated in [13], although these values may vary across different platforms, they can be used as a guideline for power consumption in a sample platform. In order to use this information in our optimization problem and due to the fact that we assume a continuous range of speeds normalized within the range of [0, 1], two cubic fitting functions for pow_A, as $-0.17\mu^3 + 0.55\mu^2 + 0.52\mu + 0.029$, and for pow_I, as $0.24\mu^3 - 0.47\mu^2 + 0.48\mu + 0.003$, are introduced.

In order to find the optimal processor speeds for maximizing the objective functions of the system with FCFS scheduling policy, we have used the method proposed in Subsection 3.3. Primarily, the interval analysis method [12] using Hessians is applied for this optimization problem through employing the INTLAB V5.5 [25] toolbox. In some cases where the computations are hard due to the

TABLE 2

SPECIFICATIONS OF THE INTEL XSCALE PXA270 PROCESSOR [13]

s	s_1	s_2	s_3	s_4	s_5	s_6	s_7
(MHz)	13	104	208	312	416	520	624
μ_{Norm}	0.0208	0.1666	0.3333	0.5000	0.6666	0.8333	1
pow_A (Watts)	0.044	0.116	0.279	0.390	0.570	0.747	0.925
pow_I (Watts)	0.015	0.064	0.129	0.154	0.186	0.222	0.260

nature of the TUFs besides the PDF of relative deadlines, we have used quadrature method with an absolute error tolerance of 10^{-10} and global line search (GLS) method using a respective MATLAB toolset [24]. (For relatively simpler utility functions, as the ones used in a number of previous studies, we can find the optimal speeds for the resulting objective functions solely through the interval analysis method and Hessians.) The results for deterministic and exponential relative deadlines are examined by checking different initial values for the numerical optimization methods. The selected configurations have only one local (or equivalently global) optimal value. Therefore, the search methods converge to the desired global optimal value. The same performance measures for the NP-EDF scheduling policy are obtained through simulations. The simulations have been done using a discrete event simulator written in C++ via the Visual C++ compiler. The experiments have been done for 10 times for every system configuration, each with 100000 arrivals. The confidence level for our simulations is 99% within the confidence interval of 0.001. The optimal speeds for the objective functions of the queueing system with the NP-EDF scheduling policy are obtained through the GLS method.

4.2 Results for Deterministic Relative Deadline

As the first example, we consider deterministic distribution of relative deadlines. Assume

$$g_D(\bar{\theta},\tau) = \delta(\tau - \bar{\theta}), \tag{49}$$

as the PDF for deterministic relative deadline, where $\bar{\theta}$ is the mean value of the random variable θ and $\delta(\tau)$ is a Dirac delta (impulse) function. Let $E(n)$ denote an Erlang random variable with parameters n and μ ($E(0)=0$) and the CDF of

$$F_{E(n)}(\tau) = P(E(n) \le \tau) = 1 - e^{-\mu\tau}\sum_{k=0}^{n-1}\frac{(\mu\tau)^k}{k!}, \quad \text{for } n > 0. \tag{50}$$

Using (24), we get

$$\Phi_n(\mu) = \frac{n!}{\mu^{n+1}}F_{E(n)}(\bar{\theta}), \tag{51}$$

where $F_{E(n)}(\bar{\theta})$ is defined in (50). Using (26), we find

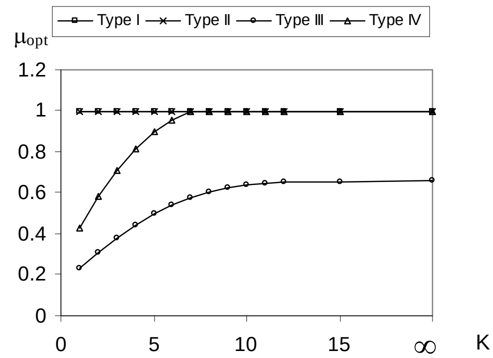
$$\gamma_n = \begin{cases} 0, & n = 0, \\ \mu\left(\dfrac{F_{E(n-1)}(\bar{\theta})}{F_{E(n)}(\bar{\theta})} - 1\right), & n > 0. \end{cases} \tag{52}$$

Further, the PDF of the conditional job offered sojourn time, $f_{V_n}(\tau)$, is given by

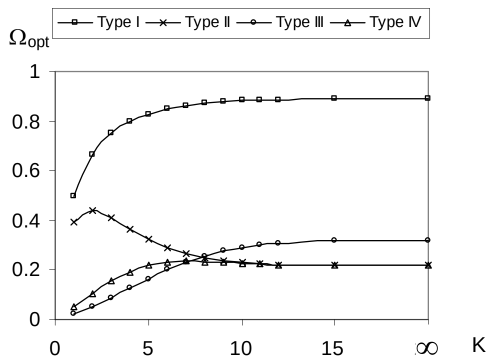
$$f_{V_n}(\tau) = \begin{cases} \dfrac{\mu^{n+1}}{n!\,F_{E(n)}(\bar{\theta})}\tau^n e^{-\mu\tau}, & \tau < \bar{\theta}, \\ \dfrac{\mu^{n+1}}{n!\,F_{E(n)}(\bar{\theta})}\bar{\theta}^n e^{-\mu\tau}, & \tau \ge \bar{\theta}, \end{cases} \tag{53}$$

through (25) and (49). Using the above calculations beside the solution method presented in Subsection 3.2 and the discussions of Subsection 3.3, $\bar{\Omega}$ and $\bar{\Omega}_{succ}$ can be calculated. Further, with the addition of the information presented in Table 1, $A(v)$ for $v = 80\%$ can be obtained. Meanwhile, using the functions of pow_A and pow_I, which are presented in Subsection 4.1, beside (48), $\bar{\Theta}$ can also be derived. Afterwards, the optimization method described in Subsection 3.3 can be applied.

Fig. 1 shows the optimal speeds (μ_{opt}) as well as their respective expected job accrued utility (Ω_{opt}) for different values of the system capacity K. As can be observed, for a Type II system, K=2 and for a Type IV system, K=7 are the optimal system capacities to maximize Ω_{opt}. Fig. 2 shows the optimal speeds as well as their respective expected job accrued UPR (Θ_{opt}) for the same capacities. It can be seen that K=1 and K=3 are the best capacities for Type II and Type IV systems, respectively. On the other hand, both Ω_{opt} and Θ_{opt} increase with K for Type I and Type III systems. Fig. 3(a) shows the assurance level of satisfying a utility-ratio of at least 80%, namely $A(80\%)$ for the service rates depicted in Fig. 1(a). The maximum probabilities are obtained for capacities 1 and 7 for Type II and Type IV systems, respectively. Meanwhile, such probabilities



(a)



(b)

Fig. 1. Expected job accrued utility for deterministic relative deadlines: (a) optimal speed (b) respective optimal job accrued utility.

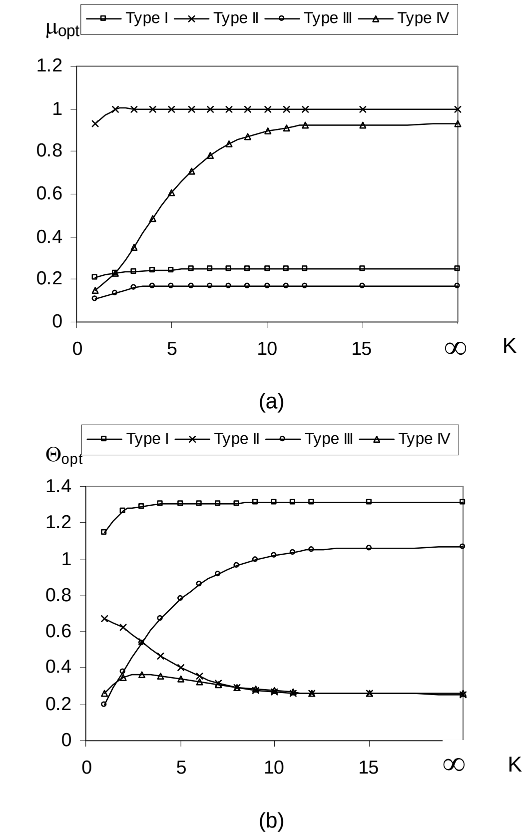


Fig. 2. Expected job accrued UPR for deterministic relative deadlines: (a) optimal speed (b) respective optimal job accrued UPR.

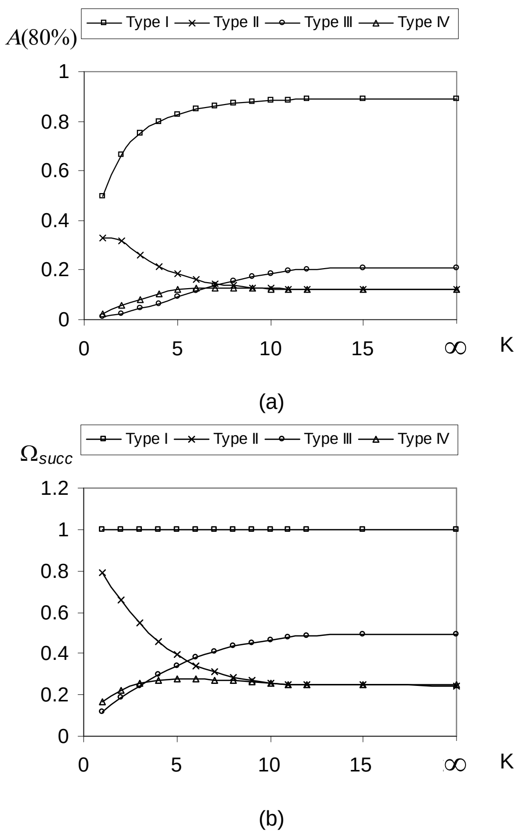


Fig. 3. Deterministic relative deadlines: (a) assurance level $A(80\%)$, (b) expected successful job accrued utility, $\overline{\Omega}_{succ}$.

TABLE 3

THE BEST SYSTEM CAPACITIES FOR DIFFERENT UTILITY PERFORMANCE MEASURES AND TUF TYPES

Deterministic	*K*			
Relative Deadline	Type I	Type II	Type III	Type IV
Ω_{opt}	↑[a]	2	↑	7
Θ_{opt}	↑	1	↑	3
$A(80\%)$	↑	1	↑	7
$\Omega_{succ}(.)$	fixed	↓[b]	↑	5

[a]*Increases with K*, [b]*Decreases with K*

increase with K for Type I and Type III systems. Fig. 3(b) shows $\overline{\Omega}_{succ}$ for the service rates of Fig. 1(a). Since every job that meets its deadline accrues utility 1 in a Type I system, the value of $\overline{\Omega}_{succ}$ is constantly 1. Such values for the Type II system decrease with K and for the Type III system increase with K. The best value for the Type IV system is obtained at K=5. The above descriptions are summarized in Table 3.

4.3 Results for Exponential Relative Deadline

As the second example, we study the exponential distribution of relative deadlines with the following PDF for the random variable θ:

$$g_E(\overline{\theta},\tau)=\frac{1}{\overline{\theta}}e^{-\frac{1}{\overline{\theta}}\tau}, \tag{54}$$

where $\overline{\theta}$ is the mean job relative deadline. Using (24) and (26) respectively, we obtain

$$\Phi_n(\mu)=\frac{n!}{\prod_{k=0}^{n}(\mu+\frac{k}{\overline{\theta}})}, \tag{55}$$

$$\gamma_n=\frac{n}{\overline{\theta}}. \tag{56}$$

Likewise, the PDF of the conditional job offered sojourn time $f_{V_n}(\tau)$ is derived as

$$f_{V_n}(\tau)=\frac{\overline{\theta}^n\prod_{k=0}^{n}(\mu+\frac{k}{\overline{\theta}})}{n!}(1-e^{-\frac{\tau}{\overline{\theta}}})^n e^{-\mu\tau}, \tag{57}$$

using (25) and (54). Afterwards, similar to the case of deterministic relative deadline, using the method presented in Subsections 3.2 and 3.3 beside the information provided in Table 1, the calculation and optimization of the desired utility performance measures can be carried out. Fig. 4 shows the optimal speeds as well as their respective expected job accrued utility for different values of the system capacity K and both FCFS and NP-EDF scheduling policies. In all cases, for K=1 and 2, NP-EDF mimics FCFS. As can be observed, for the Type II system, FCFS accrues more utility than NP-EDF. It is due to the reason that FCFS tries to execute a job as close as possible to its arrival time, which seems a good decision for this type of TUFs. Fig. 5 presents the results for the expected job accrued UPR. It can be seen that for the Type II system, FCFS behaves better than NP-EDF. For the Type IV system, except in the case of K=3, FCFS is again better than NP-EDF. In a similar manner, we can expect more improvements of FCFS over NP-EDF for specific TUFs. In

TABLE 4

COMPARISON OF THE BEST SYSTEM CAPACITIES BETWEEN FCFS AND NP-EDF FOR DIFFERENT UTILITY PERFORMANCE MEASURES AND TUF TYPES

Exponential	*K*							
Relative	Type I		Type II		Type III		Type IV	
Deadline	FCFS	NP-EDF	FCFS	NP-EDF	FCFS	NP-EDF	FCFS	NP-EDF
Ω_{opt}	↑[a]	↑	3	2	↑	↑	↑	9
Θ_{opt}	↑	↑	1	1	↑	↑	8	3
$A(80\%)$	↑	↑	2	2	↑	↑	↑	8
$\Omega_{succ}(.)$	fixed	fixed	1	1	15	15	↑	6

[a]*Increases with K*

other words, regarding Θ_{opt}, FCFS may be a good policy for some TUFs with special properties. However, finding such TUFs is not the main concern of this study; nevertheless, it can be considered as a further investigation. Fig. 6(a) shows $A(80\%)$ for the service rates depicted in Fig. 4(a). According to the results, FCFS outperforms NP-EDF for the Type II system. Fig. 6(b) shows $\overline{\Omega}_{succ}$ for the service rates of Fig. 4(a). Regarding $\overline{\Omega}_{succ}$, for the Type II system as well as the Type IV system with K>10, FCFS outperforms NP-EDF. Table 4 summarizes the best values of K for each of the above cases with both FCFS and NP-EDF policies.

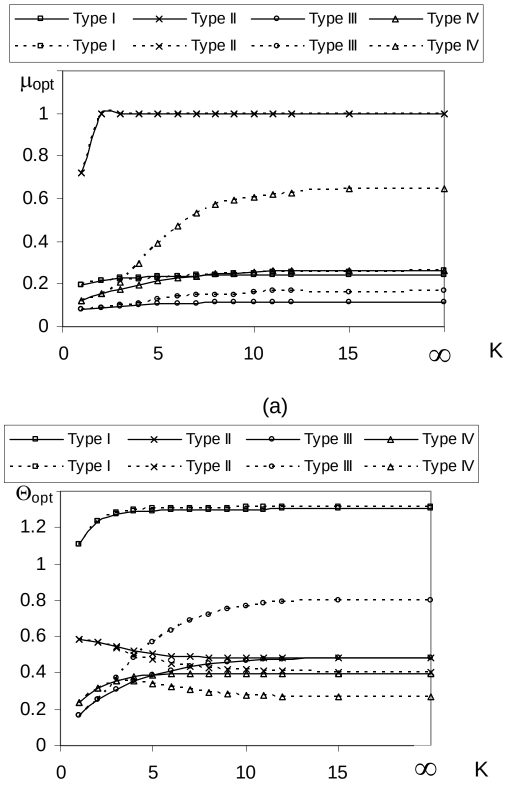


Fig. 5. Expected job accrued UPR for exponential relative deadlines: (a) optimal speed (b) respective optimal job accrued UPR. (Solid lines: FCFS, dotted lines: NP-EDF)

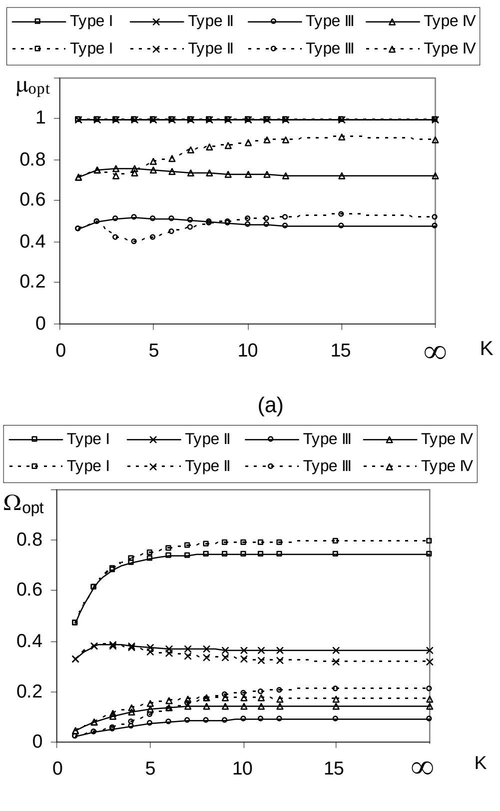


Fig. 4. Expected job accrued utility for exponential relative deadlines: (a) optimal speed (b) respective optimal accrued utility. (Solid lines: FCFS, dotted lines: NP-EDF)

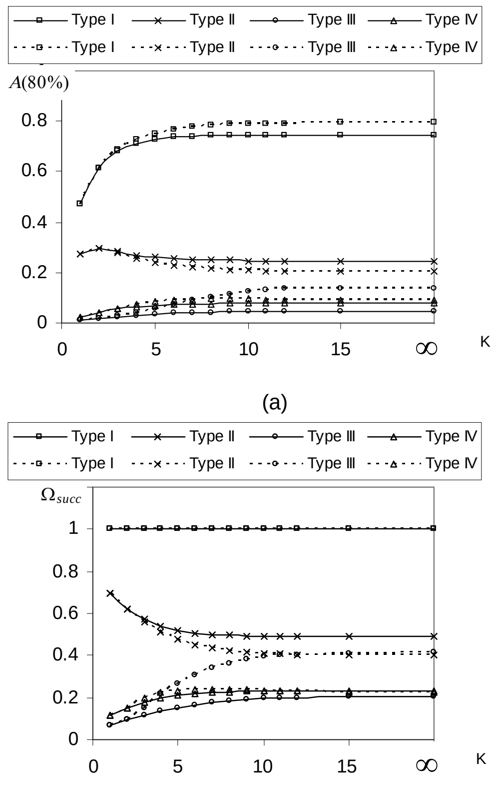


Fig. 6. (a) The assurance level $A(80\%)$, (b) expected successful job accrued utility, $\overline{\Omega}_{succ}$.(Solid lines: FCFS, dotted lines: NP-EDF)

5 Related Work

In this section, we first present an overview on some deadline-based scheduling algorithms. Afterwards, a survey on UA scheduling algorithms with energy concerns as well as without energy concerns is presented.

Most of the previous studies on real-time scheduling algorithms have concentrated on deadline-based algorithms, i.e., algorithms which their main targets are meeting job deadlines. Some examples are EDF [21] and MLF [26]. It is well known that in an underloaded system, EDF and MLF are optimal algorithms among deadline-based scheduling policies. Thus, they can propose feasible schedules satisfying all the job deadlines. The optimality of preemptive EDF within the class of non-idling service-time independent preemptive scheduling policies has been shown in [9]. Further, the optimality of non-preemptive EDF within the class of non-idling service-time independent non-preemptive policies is shown in [10]. It is also proved in [31] that among the non-UA scheduling algorithms in an overloaded FRT system, EDF is optimal and maximizes the fraction of independent jobs meeting their deadlines. In other words, assuming binary-valued downward step TUF, EDF in a FRT system maximizes the accrued utility.

For independent jobs with step TUFs in an overloaded system, D^{over} is shown to have the optimal competitive factor [18], even though its average performance is quite poor for random jobs [19]. DASA without Dependency (or DASA-ND) [8] also considers step TUFs and overloads. DASA allows jobs to mutually exclusively share non-CPU resources under the single-unit resource request model.

The first publicized UA scheduling algorithm that considers almost arbitrary TUF shapes for preemptive independent jobs is LBESA [22]. Assume a metric called potential utility density (PUD) for a job (or task) as the ratio of the expected job utility to the remaining job execution time [40]. LBESA examines jobs in an EDF order and performs a feasibility check where it rejects jobs with lower PUDs until the schedule is feasible. Some extensions to the algorithm, with the same basic idea as LBESA, are presented in [1] and [26]. Non-step TUFs are also considered by GUS [19] and RUA [36] algorithms which both use the concept of PUD. GUS allows resource sharing among jobs with arbitrary TUFs. RUA considers preemptive jobs in a FRT system subject to arbitrarily shaped TUFs (where all job TUFs are assumed to reach zero value at the respective deadlines) and concurrent sharing of non-CPU resources. Despite GUS that assumes single-unit resource request model, RUA considers the multi-unit resource request model. As another study, the S-UA algorithm proposed in [20] provides probabilistic bounds on task-level accrued utilities.

Both LBESA and DASA yield optimal total utility under downward step TUFs during underloads [7]. GUS, DASA, and LBESA have the best performance among existing UA algorithms [19]. Moreover, DASA and LBESA mimic EDF to reap its optimality during underloads. Several more UA scheduling algorithms have also been developed. Examples include CMA [6], UPA [33] (which is shown to have higher accrued utility than EDF and CMA), as well as CUA [35]. The CMA and UPA algorithms which require the knowledge of job execution times consider non-increasing TUFs in the context of non-preemptive scheduling of independent jobs. On the other hand, GUS, CUA, and RUA consider arbitrary TUFs, preemptive scheduling, and resource dependencies among jobs.

None of the above efforts on UA scheduling algorithms considers energy consumption limitations. They try to maximize the accrued utility using scheduling decisions, while the capabilities available in today CPUs can be used to maximize the accrued utility, energy saving, or even both. Most of the past efforts on energy-efficient real-time scheduling focus on maximizing the energy savings, while guaranteeing some deadline-based timeliness criteria, such as meeting all or some fraction of deadlines. Examples are [3], [23] and [17] (and the references therein) for hard, soft, and firm real-time systems, respectively.

One of the first studies on energy-efficient UA scheduling is PA-BTA [32], which heuristically computes schedules to maximize a proposed performance metric called Energy and Real-time performance Grade (ERG) for jobs with non-increasing TUFs. ERG is a linear combination of accrued utility and saved energy. As another example of studies in this issue, [39] presents an algorithm called EUA for jobs with a minimum inter-arrival time, which are subject to step TUFs. EUA provides statistical assurances on some performance measures while taking into account the system energy efficiency. In [37], an energy-efficient multicriteria real-time scheduling algorithm called EUA* is presented. Its objective is to probabilistically satisfy lower bounds on the accrued utility while maximizing the system-level energy efficiency, for a relatively general model of arrivals. EUA* achieves optimal timeliness during underloads, and identifies the conditions under which timeliness assurances hold. ReUA [41], as another algorithm, considers an application model where jobs are subject to non-increasing TUFs. The algorithm targets mobile embedded systems where system-level energy consumption is also a major concern. It satisfies the statistical performance requirements on individual job timeliness behavior as well as maximizing the system-level energy efficiency, while respecting resource constraints. Since the problem is *NP-hard*, ReUA allocates CPU cycles using statistical properties of application cycle demands, and heuristically computes schedules with a polynomial time cost.

In [38], a DVS-based CPU scheduling algorithm called EBUA is presented. It considers preemptive jobs that are subject to non-increasing TUFs, mutual exclusion resource dependencies, statistical task-level timeliness assurance requirements, and an energy budget which cannot be exceeded at run-time. This algorithm tries to maximize a metric called Utility and Energy Ratio (or UER), namely the amount of utility that can be accrued per unit of energy, by executing the jobs (and their dependents due to resource dependencies) in a proper manner. In spite of other UA scheduling algorithms that maximize the collective utility attained by all jobs, EBUA provides assurance on individual job's timeliness behavior, i.e., it probabilistically satisfies a lower bound on individual job's accrued utility. Whenever the tasks are independent and no DVS is used, EBUA behaves like the Dynamic Timeliness-Density (DTD) heuristic proposed in [1].

Most of the above mentioned algorithms (especially the ones focusing on energy-efficient UA scheduling) consider only non-increasing TUFs. Further, among these studies, only CMA and UPA consider non-preemptive jobs. Meanwhile, few of them present analytical solutions to their proposed algorithms (see [39] for step TUFs, [20], [37], [38], and [41] for non-increasing TUFs, and [19] for unimodal TUFs which presents bounds on the accrued utility of a non work-conserving algorithm). However, none of these studies is based on an analytical queueing modeling method. To the best of the authors' knowledge, there exists no exact analytical method for the evaluation of the proposed UA scheduling algorithms with arbitrary TUFs. The proposed method in this paper can be used to precisely analyze and optimally configure the FRT system (indicated in Section 2) with non-preemptive jobs and a work-conserving scheduling algorithm while at the same time it considers more general TUF shapes.

6 Conclusions and Future Work

The importance of completing the execution of jobs in many emerging real-time systems is more precisely described by TUFs. Meanwhile, these systems may vastly be subject to limited sources of energy. Therefore, maximizing different measures related to the accrued utility as well as accrued utility per unit of energy are important goals in the design of such systems.

In this paper, we consider a FRT system with an arbitrary capacity and Poisson arrival jobs. The jobs have exponential execution times, generally distributed relative deadlines, and arbitrary shaped TUFs. The scheduling policy in the system is FCFS. An analytical method is proposed for the calculation of some performance and power-related measures of the system. Using the analytical method, one can find the optimal processor speeds to maximize the measures through equating the derivatives of the respective formulations to zero and finding the roots. However, due to the complexity of finding closed-form solutions for the resulting equations, we have obtained the optimal speeds through interval analysis and GLS methods. For each measure, the optimal service rate is then used to find the optimal system capacity among some conceivable capacities for maximizing the same measure. Some experiments are carried out for different TUFs as well as the deterministic and exponential distributions of relative deadlines. The results for the latter distribution are also compared against the NP-EDF scheduling policy, which is known as an optimal policy for binary-valued, downward step TUFs. It is shown through the results which for some measures and TUFs FCFS outperforms NP-EDF.

Several aspects of this work can be taken as insights into directions for further research. Examples include consideration of multipriority jobs or jobs with different TUFs as well as presentation of policies based on similar analytical methods for dynamic speed selection.

Acknowledgment

This work was supported in part by IPM under Grant CS1387-4-02.

References

[1] S.A. Aldarmi and A. Burns, "Dynamic Value-Density for Scheduling Real-Time Systems," *Proc. Euromicro Conf. Real-Time Systems*, pp. 270–277, June 1999.

[2] H. Aydin, R. Melhem, D. Mosse, and P. Mejia-Alvarez, "Dynamic and Aggressive Scheduling Techniques for Power-Aware Real-Time Systems," *Proc. IEEE Real-Time Systems Symp.*, pp. 95–105, Dec. 2001.

[3] H. Aydin, R. Melhem, D. Mosse, and P.M. Alvarez, "Powe-Aware Scheduling for Periodic Real-Time Tasks," *IEEE Trans. Computers*, vol. 53, no. 5, pp. 584–600, 2004.

[4] D.Y. Barrer, "Queueing with Impatient Customers and Ordered Service," *J. Operational Research*, vol. 5, pp. 650-656, 1957.

[5] G. Bernat, A. Burns, A. Llamosi, "Weakly Hard Real-Time Systems," *IEEE Trans. Computers*, vol. 50, no. 4, pp. 308-321, 2001.

[6] K. Chen and P. Muhlethaler, "A Scheduling Algorithm for Tasks Described by Time Value Function," *J. Real-Time Systems*, vol. 10, no. 3, pp. 293-312, May 1996.

[7] H. Cho, "Utility Accrual Real-Time Scheduling and Synchronization on Single and Multiprocessors: Models, Algorithms, and Tradeoffs," Ph.D. dissertation, Virginia Polytechnic Institute and State Univ., Blacksburg, Virginia, 2006.

[8] R.K. Clark, "Scheduling Dependent Real-Time Activities," Ph.D. dissertation, Dept. of Computer Eng., Carnegie Mellon Univ., Pittsburgh, 1990.

[9] L. George, P. Muhlethaler, and N. Rivierre, "Optimality and Non-Preemptive Real-Time Scheduling Revisited," Technical Report *RR-2516*, INRIA, Le Chesnay Cedex, France, 1995.

[10] L. George, N. Rivierre, and M. Spuri, "Preemptive and Non-Preemptive Real-Time Uni-Processor Scheduling," Technical Report RR-2966, INRIA, Le Chesnay Cedex, France, 1996.

[11] K. Han, B. Ravindran, and E.D. Jensen, "Probabilistic, Real-Time Scheduling of Distributable Threads under Dependencies in Mobile, Ad-hoc Networks," *Proc. IEEE Wireless Communications and Networking Conf.*, pp. 4307-4312, Mar. 2007.

[12] E.R. Hansen and G.W. Walster, *Global Optimization using Interval Analysis*, CRC Press, 2004.

[13] *Intel PXA270 Processor Electrical, Mechanical and Thermal Specification Data Sheet*, http://www.intel.com/design/embeddedpca/products/pxa270/techdocs.htm.

[14] E.D. Jensen, C.D. Locke, and H. Tokuda, "A Time-Driven Scheduling Model for Real-Time Systems," *Proc. IEEE Real-Time Systems Symp.*, pp. 112-122, Dec. 1985.

[15] M. Kargahi and A. Movaghar, "Non-Preemptive Earliest-Deadline-First Scheduling Policy: A Performance Study," *Proc. IEEE Int. Symp. Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, Georgia, Atlanta, pp. 201–210, 2005.

[16] M. Kargahi and A. Movaghar, "A Method for Performance Analysis of Earliest-Deadline-First Scheduling Policy," *J. Supercomputing*, vol. 37, no. 2, pp. 197–222, 2006.

[17] M. Kargahi and A. Movaghar, "Stochastic DVS-Based Dynamic Power Management for Soft Real-Time Systems," *J. Microprocessors and Microsystems*, vol. 32, no. 3, pp. 121-144, 2008.

[18] G. Koren and D. Shasha, "D-over: An Optimal On-Line Scheduling Algorithm for Overloaded Real-Time Systems," *Proc. IEEE Real-Time Systems Symp.*, pp. 290-299, Dec. 1992.

[19] P. Li, "Utility Accrual Real-Time Scheduling: Models and Algorithms," Ph.D. dissertation, Virginia Polytechnic Institute and State Univ., Blacksburg, Virginia, 2004.

[20] P. Li, H. Cho, B. Ravindran, and E.D. Jensen, "Stochastic, Utility Accrual Real-Time Scheduling with Task-Level and System-Level Timeliness Assurances," *Proc. 8th IEEE Int. Symp. Object-Oriented Real-Time Distributed Computing*, pp. 216-223, 2005.

[21] C.L. Liu and J.W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment," *J. ACM*, vol. 20, no. 1, pp. 46-61, 1973.

[22] C.D. Locke, "Best-Effort Decision Making for Real-Time Scheduling," Ph.D. dissertation, Dept. Computer Eng., Carnegie Mellon Univ., Pittsburgh, 1986.

[23] J.R. Lorch and A.J. Smith, "PACE: A New Approach to Dynamic Voltage Scaling," *IEEE Trans. Computers*, vol. 53, no. 7, pp. 856–869, 2004.

[24] *MATLAB Line Search Routines, Global Line Search (GLS)*, http://www.mat.univie.ac.at/~neum/software/ls/.

[25] *MATLAB Toolbox for Reliable Computing*, Hamburg University of Technology, http://www.ti3.tu-harburg.de/rump/intlab/.

[26] A.K.-L. Mok, "Fundamental Design Problems of Distributed Systems for Hard Real-Time Environment," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, 1983.

[27] D. Mosse, M.E. Pollack, and Y. Ronen, "Value-Density Algorithm to Handle Transient Overloads in Scheduling," *Proc. 11th Euromicro Conf. Real-Time Systems*, pp. 278–286, June 1999.

[28] A. Movaghar, "On Queueing with Customer Impatience until the End of Service," *J. Stochastic Models*, vol. 22, pp. 49–173, 2006.

[29] P. Pillai and K.G. Shin, "Real-Time Dynamic Voltage Scaling for Low-Power Embedded Operating Systems," *Proc. ACM Symp. Operating Systems Principles*, pp. 89–102, 2001.

[30] B. Ravindran, E.D. Jensen, and P. Li, "On Recent Advances in Time/Utility Function Real-Time Scheduling and Resource Management," *Proc. 8th IEEE Int. Symp. Object-Oriented Real-Time Distributed Computing*, pp. 55-60, 2005.

[31] D. Towsley and S.S. Panwar, "Optimality of the Stochastic Earliest Deadline Policy for the G/M/c Queue Serving Customers with Deadlines," *Proc. 2nd ORSA Telecommunications Conf.*, 1992.

[32] J. Wang, B. Ravindran, and T. Martic, "A Power-Aware, Best-Effort Real-Time Task Scheduling Algorithm," *Proc. IEEE Workshop on Software Technologies for Future Embedded Systems*, *IEEE Int. Symp. Object-Oriented Real-Time Distributed Computing*, pp. 21-28, May 2003.

[33] J. Wang and B. Ravindran, "Time-Utility Function-Driven Switched Ethernet: Packet Scheduling Algorithm, Implementation, and Feasibility Analysis," *IEEE Trans. Parallel and Distributed Systems*, vol. 15, no. 2, pp. 119–133, Feb. 2004.

[34] H. Wu, "Energy-Efficient, Utility Accrual Real-Time Scheduling," Ph.D. dissertation, Virginia Polytechnic Institute and State Univ., Blacksburg, Virginia, 2005.

[35] H. Wu, B. Ravindran, and E.D. Jensen, "Utility Accrual Scheduling under Joint Utility and Resource Constraints," *Proc. IEEE Int. Symp. Object-Oriented Real-time Distributed Computing*, pp. 307–314, May 2004.

[36] H. Wu, B. Ravindran, E.D. Jensen, and U. Balli, "Utility Accrual Scheduling under Arbitrary Time/Utility Functions and Multi-unit Resource Constraints," *Proc. IEEE Int. Conf. Real-Time Computing Systems and Applications*, pp. 80–98, Aug. 2004.

[37] H. Wu, B. Ravindran, and E.D. Jensen, "Energy-Efficient, Utility Accrual Real-Time Scheduling under the Unimodal Arbitrary Arrival Model," *Proc. Design, Automation and Test in Europe*, vol. 1, pp. 474-479, 2005.

[38] H. Wu, B. Ravindran, and E.D. Jensen, "Utility Accrual Real-Time Scheduling under the Unimodal Arbitrary Arrival Model with Energy Bounds," *IEEE Trans. Computers*, vol. 56, no. 10, pp. 1358-1371, Oct. 2007.

[39] H. Wu, B. Ravindran, E.D. Jensen, and P. Li, "CPU Scheduling for Statistically-Assured Real-Time Performance and Improved Energy Efficiency," *Proc. 2nd IEEE/ACM/IFIP Int. Conf. Hardware/Software Codesign and System Synthesis*, pp. 110-115, Sep. 2004.

[40] H. Wu, B. Ravindran, E.D. Jensen, and P. Li, "Time/Utility Function Decomposition Techniques for Utility Accrual Scheduling Algorithms in Real-Time Distributed Systems," *IEEE Trans. Computers*, vol. 54, no. 9, pp. 1138-1153, Sep. 2005.

[41] H. Wu, B. Ravindran, E.D. Jensen, and P. Li, "Energy-Efficient, Utility Accrual Scheduling under Resource Constraints for Mobile Embedded Systems," *ACM Trans. Embedded Computing Systems*, vol. 5, no. 3, pp. 513-542, Aug. 2006.

[42] L. Yan, J. Luo, and N.K. Jha, "Joint Dynamic Voltage Scaling and Adaptive Body Biasing for Heterogeneous Distributed Real-Time Embedded Systems," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 7, pp. 1030-1041, July 2005.

[43] W. Yuan and K. Nahrstedt, "Energy-Efficient Soft Real-Time CPU Scheduling for Mobile Multimedia Systems," *Proc. ACM Symp. Operating Systems Principles*, pp. 149–163, 2003.

Mehdi Kargahi (M'09) is currently an assistant professor in the Department of Electrical and Computer Engineering at University of Tehran, Iran. He received his B.S. degree in Computer Engineering from Amir-Kabir University of Technology (Tehran Poly-Techniques) in Tehran (1998) and both M.S. (2001) and Ph.D. (2006) in Computer Engineering from Sharif University of Technology in Tehran. He visited University of Victoria in Canada as a researcher in 2006 and has also been a researcher in the School of Computer Science at Institute for studies in theoretical Physics and Mathematics (IPM) from 2003. His research interests include performance modeling and power management of dependable and distributed real-time systems with stochastic properties.

Ali Movaghar (M'92, SM'07) is currently a professor in the Department of Computer Engineering at Sharif University of Technology in Tehran, Iran where he joined first as an assistant professor in 1993. He received his B.S. in electrical engineering from the University of Tehran in 1977 and both M.S. and Ph.D. in computer, information and control engineering from the University of Michigan at Ann Arbor in 1979 and 1985, respectively. He visited INRIA in France in 1984, worked at AT&T Laboratories during 1985-1986, and taught at the University of Michigan during 1987-1989. His main areas of interest are performance and dependability modeling, verification and validation, computer networks, and distributed real-time systems.