Utility Accrual Dynamic Routing in Real-Time Parallel Systems

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

Abstract— One of the main properties of today's distributed and parallel systems, such as mobile ad-hoc networks and grids, is their heterogeneity in the available resources. Further, many applications of such systems are subject to Time/Utility Function (TUF) time constraints for jobs, unavoidable variability in job characteristics and arrivals, and statistical assurance requirements on timeliness behaviors. In this paper, we propose an exact analytical solution for performance evaluation of dynamic policies used for routing of TUF-constrained Firm Real-Time (FRT) jobs among parallel single-processor queues with arbitrary processing rates and capacities. The analytical method can be used for the evaluation of the compliance of some important statistical assurance requirements. Furthermore, we present a utility-aware dynamic routing policy to improve the expected accrued utility of the parallel system. The policy called Maximum Expected Utility (MEU) behaves based on the information gathered from the analytical solution. MEU is compared with some well-known Dynamic Routing (DR) policies for different TUF shapes and both cases of homogeneous and heterogeneous processors of a two-queue system. The comparisons show the efficiency of MEU for the former case and its good behavior in most situations for the latter case.

Index Terms— Analytical modeling, Firm real-time systems, Performance modeling, Time/utility function, Utility accrual dynamic routing.

1 Introduction

Parallel and distributed real-time systems usually include heterogeneous nodes with different capabilities and available resources (e.g., processing rate, buffer capacity, energy storage, and network bandwidth). Examples include time constrained distributed systems over wireless local-area networks, mobile ad-hoc networks, and grid environments. However, such systems are often used in critical and harsh environments with uncertain properties. These uncertainties are encountered in critical applications such as space and defense domains or in less critical applications such as multimedia and streaming systems. 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.

Applications of such systems are also becoming more complicated in the description of the respective Quality of Service (QoS) requirements. The classical notion of deadlines may not be able to exactly express them, especially, when the applications have relatively complex soft and possibly firm timing requirements, i.e., the time of successful completion of a job affects the importance of the job completion. Precise description of the timing requirements and accurate cognition of their effects can help us to improve the overall performance of such systems.

- *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.*
- *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 (July 22, 2009).

1.1 Time/Utility Functions for Real-Time Systems

As indicated in [27], 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 time that the job completes its execution. The latter criterion is usually characterized by the job's Time/Utility Function (TUF), as proposed for the first time by Jensen et al. [12]. A TUF, which precisely specifies the semantics of Soft Real-Time (SRT) constraints, determines the utility resulting from the completion of a job as a function of its completion time (possibly with respect to its arrival time). On the other hand, the TUF-constrained systems in which jobs passing their deadline become of no value (their TUF reaches zero on their deadline) and are thrown away immediately are called Firm Real-Time (FRT) [1]. Therefore, the difference between SRT and FRT jobs is that the latter consume no more resources (e.g., buffer or processing time) after missing their deadline.

One major goal in such systems is to complete the TUF-constrained jobs as close as possible to their optimal completion times. Further, the optimality criteria in the systems are known as Utility Accrual (UA) which is based on the accrued utility by the system or the individual jobs. Example criteria are maximizing the job expected attained utility or the assurance level of satisfying lower bounds on the attained utilities.

TUFs can be classified into unimodal and multimodal functions. Unimodal TUFs are those for which any decrease in the utility function cannot be followed by an increase. Multimodal TUFs do not follow this constraint. Irrespective of the generality of multimodal TUFs, most of the studies on the UA scheduling algorithms consider simpler TUF shapes, e.g., step TUFs or special cases of non-step unimodals. TUF for the classical deadline is a binary-valued downward step function. A number of sample TUF shapes for different applications are presented in [32] and the references therein. 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 [30]. Some notes on TUF shapes in mobile ad-hoc network applications can also be found in [9].

1.2 Motivation and Paper Outline

This paper studies the problem of dynamic routing of TUF-constrained FRT jobs (with no limitations on the TUF shapes) among parallel subsystems.

As mentioned earlier, many of the today's parallel and distributed systems suffer from uncertainties and stochastic nature of events. In such systems, *dynamic load balancing* techniques, which we mostly refer to as Dynamic Routing (DR) policies throughout this paper, play important roles in the optimization of performance measures and resource utilizations.

Traditionally, dynamic load balancing techniques are used to prevent overusing of a processor in a parallel or distributed system while another processor is idle, as well as adapting the load on different processors in proportion to the processors speeds in the system. However, especially for TUF-constrained real-time systems with the UA criteria, dynamic load balancing cannot follow the traditional policies. Rather, they may need quite different criteria for route selections based on different TUF shapes.

In this paper, the subsystems, among which dynamic routing is applied, are considered as single-processor FRT queueing systems with arbitrary processing rates and capacities. It is assumed that no migration is allowed among the subsystems and no preemption is permitted within a subsystem. The problem is studied for specific stochastic properties of the parallel system through an exact analytical modeling. Based on the modeling, several important UA performance measures are calculated for general DR policies in the class of policies that are aware of only the specifications and populations of the subsystems. The accurate calculation of these measures helps with a better understanding of the effects of various TUF time constraints on the behavior of the system. According to the analytical solution, a *utility-aware* DR policy, called Maximum Expected Utility (MEU), is also proposed, which uses the calculated information about the system behavior with respect to job TUFs to select the target routes. MEU is compared to some well-known DR policies in the aforementioned class for different TUF shapes. The numerical results show the efficiency of this policy in most instances.

Thus, this paper's contribution includes exact analytical formulation of dynamic routing of TUF-constrained FRT jobs among parallel heterogeneous subsystems, presentation of a utility-aware DR policy which uses the analytical results, and evaluation of the effects of different TUF shapes on the efficacy of the DR policy. To the best of the authors' knowledge, there exists no previous exact analytical method for a similar system with TUF-constrained jobs. Moreover, we know of no previous utility-aware DR policy for such a system with arbitrary TUF shapes (except [3] for a different system model and a specific TUF shape).

The rest of this paper is organized as follows. The model of the parallel real-time system and the respective performance measures are presented in Section 2. In Section 3, the queueing model of the system, some conditional parameters, and the analytical solution are described. Then, in Section 4, three TUF-independent DR policies besides the proposed utility-aware DR policy are presented. Moreover, in the same section, the basic calculations are carried out for two case studies. Afterwards, some numerical examples are presented in Section 5 for performance evaluation of the DR policies in different system configurations. In Section 6, we discuss the related works. Finally, the conclusions of the paper as well as some topics for future studies are presented in Section 7.

2 System Model and Performance Measures

This section first identifies the parallel system and application models, and then, presents the formulation of our favorite system 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 Parallel System, Job and Utility Models

We consider a system consisting of s parallel single-processor queues (subsystems) $Q_i : (K_i, \mu_i)$ numbered as i=1,…,s, where K_i is the capacity of the i-th queue and μ_i is the processing rate of the respective processor. 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 $\mathbf{n}=(n_1,\dots,n_s)$, where $n_i \le K_i$ is the number of jobs in the i-th queue, i=1,…,s. These parameters are defined more precisely in the following paragraphs.

The job arrival times (a) follow a state-dependent Poisson process with rate $\lambda(|\mathbf{n}|)$, where $|\mathbf{n}| = \sum_{i=1}^{s} n_i$ is the total number of jobs in the system at state $\mathbf{n}$. A dispatcher assigns an incoming job to the proper queue according to a stationary DR policy. Upon arrival in a non-full state $\mathbf{n}$, the arriving job joins the j-th queue with a probability $r_{\mathbf{n}}(j)$ (where $\sum_{j=1}^{s} r_{\mathbf{n}}(j) = 1$). An arriving job who finds the system full is blocked and must leave the system immediately.

Jobs entering each queue are served in the order of their arrival, i.e., the service discipline in the subsystems is FCFS. Migration of jobs among the queues is not permitted. 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, $P(\theta=\infty)=1-G(\infty-0)\geq 0$. 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 execution times and relative deadlines form sequences of i.i.d. random variables which are mutually independent. Given the number of jobs in the system at any time, the future arrival process is assumed to be conditionally independent of the past history of the system.

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:

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

where $0\leq\tau\leq\theta$ is the sojourn time (response time) of that job. (We use these two terms interchangeably throughout this paper.) $U(.)$ is the same function for all jobs of the system. (This assumption can simply be relaxed by considering multiclass jobs, where the jobs of each class c with a class-specific TUF arrive according to a state-dependent Poisson process with rate λ^c.) The TUF of each job can take non-zero values only between the arrival time and deadline of that job, namely in the interval $[a, a+\theta]$.

2.2 The System Performance Measures

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

$V\equiv$ the time an arriving job with infinite (no) deadline must wait before it completes its service in the long run. (2)

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

$$F_V(\tau)=P(V\leq\tau), \tag{3}$$

or, equivalently, the probability density function (PDF)

$$f_V(\tau)=\frac{dF_V(\tau)}{d\tau}. \tag{4}$$

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

$$\alpha_d=P(\theta<V<\infty)=\int_0^\infty G(\tau)dF_V(\tau). \tag{5}$$

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,

$$\alpha_b=P(V=\infty)=1-F_V(\infty-0). \tag{6}$$

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 defined as

$$\alpha=\alpha_d+\alpha_b=P(V>\theta)=\int_0^\infty G(\tau)dF_V(\tau)+P(V=\infty). \tag{7}$$

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. Let

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

Ω is called the job offered accrued utility. We will be interested in finding the CDF of Ω

$$F_\Omega(\omega)=P(\Omega\leq\omega), \tag{9}$$

or, equivalently, the PDF

$$f_\Omega(\omega)=\frac{dF_\Omega(\omega)}{d\omega}. \tag{10}$$

The job expected accrued utility $\overline{\Omega}$, as the main performance measure in this paper, can then be obtained as

$$\overline{\Omega}=\int_0^\infty \omega\, dF_\Omega(\omega)=\int_0^\infty\left(\int_\tau^\infty g(x)\,U(\tau,x)dx\right)dF_V(\tau), \tag{11}$$

where $g(.)$ 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 (11) is the expected utility of a successful job with a response time τ (who 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., $1-\alpha$, can be calculated through either (7) or (11).

As another important measure of performance, the assurance level of satisfying a lower bound η on the attained utilities can be calculated as

$$A(\eta)=P(\Omega\geq\eta)=1-F_\Omega(\eta). \tag{12}$$

3 Queueing Model

In this section, first a few additional notations which we use throughout this paper are introduced. Then, in Subsection 3.1, the conditional performance variables required for solving the queueing model of the FRT system under discussion are presented. Afterwards, in Subsection 3.2, a Markovian model for the analysis of the system is presented which its long-run solution is obtained using standard Markovian solution techniques.

Let $\mathcal{N}$ be the set of natural numbers (including 0) and $\mathcal{R}^+$ the set of positive real numbers. Suppose $\mathbf{n}=(n_1,\ldots,n_i,\ldots,n_s)$ be a s-tuple of natural numbers and $\mathbf{e}_i$ one such s-tuple with value 0 at each coordinate except for coordinate i at which it has a value of 1, i.e., $\mathbf{e}_i=(0,\ldots,1,\ldots,0)$. We use the following notations:

- $\mathbf{n}+\mathbf{e}_i\equiv(n_1,\ldots,n_i+1,\ldots,n_s)$, where $i=1,\ldots,s$,
- $\mathbf{n}-k\mathbf{e}_i\equiv(n_1,\ldots,n_i-k,\ldots,n_s)$, where $n_i>0$, $i=1,\ldots,s$, and $1\leq k\leq n_i$,
- $q(\mathbf{n})=\{i: n_i>0,\ i=1,\ldots,s\}$,
- $l(\mathbf{n})=\{i: n_i<K_i,\ i=1,\ldots,s\}$,
- $\mathbf{0}=(0,\ldots,0)$.

We also use $E^i(n)$ to denote an Erlang random variable with parameters n and μ_i ($E^i(0)=0$) and the CDF of

$$F_{E^i(n)}(\tau)=P(E^i(n)\leq\tau)=1-e^{-\mu_i\tau}\sum_{k=0}^{n-1}\frac{(\mu_i\tau)^k}{k!},\quad \text{for } n>0. \tag{13}$$

The following definition is also considered:

$V^i\equiv$ the time a job with infinite (no) deadline which arrives at the i-th processor's queue, in the long run, must wait before it completes its service. (14)

3.1 Conditional Performance Variables

In this subsection, we introduce some conditional parameters and performance variables on the system state $\mathbf{n}$ and derive the respective formulations in the long run. First, we identify an important modeling parameter for the queueing problem described in Section 2. Denote $\psi^i(t,\mathbf{n},\varepsilon)$ to be the probability that a job in the i-th queue misses its deadline during $[t,t+\varepsilon)$, given there are n_j jobs in the j-th queue, j=1,…,s, at time t. Let

$$\Gamma^i(t,\mathbf{n}) = \lim_{\varepsilon\to 0}\frac{\psi^i(t,\mathbf{n},\varepsilon)}{\varepsilon}. \tag{15}$$

Assuming statistical equilibrium, we have

$$\Gamma^i(\mathbf{n}) = \lim_{t\to\infty}\Gamma^i(t,\mathbf{n}), \tag{16}$$

where $\Gamma^i(.)$ is said to be the long-run conditional loss rate function for the i-th queue, given state $\mathbf{n}$.

Note that the concept of (long-run) conditional loss rate as defined in (16) is in fact independent of the model of job behavior considered in this paper. It may similarly be defined for other models of job behavior, including the case where the deadlines of jobs are effective only until the beginning of their service, as studied in [22] for parallel queues and the references therein for single queues.

However, we solve the problem for the case where the deadlines of jobs are effective until the end of their service. We also derive a closed-form solution for the conditional PDF of the offered sojourn time of individual queues, given the state $\mathbf{n}$ of the system. More formally, let

$V^i_{\mathbf{n}} \equiv$ the time a job with infinite (no) deadline which arrives at the i-th processor's queue, in the long run, must wait before it completes its service, given it finds n_j jobs in the j-th processor's queue, j=1,…,s. (17)

$V^i_{\mathbf{n}}$ is called the conditional offered sojourn time of the i-th processor's queue, given the system state $\mathbf{n}$. We can show:

Lemma 1. *Let $\theta^i_{n_i}$ and $E^i_{n_i}(1)$ represent the relative deadline and execution time of the n_i-th job waiting in the i-th queue (i=1,…,s), respectively, in the long run. Thus, $\theta^i_{n_i}$ is a random variable with CDF $G(.)$, and $E^i_{n_i}(1)$ is a random variable with an exponential CDF with rate μ_i, which is independent of $\theta^i_{n_i}$ and $V^i_{\mathbf{n}-\mathbf{e}_i}$. Then, we have*

$$P(V^i_{\mathbf{n}}\le\tau)=\begin{cases}1-e^{-\mu_i\tau}, & n_i=0,\\ P(V^i_{\mathbf{n}-\mathbf{e}_i}+E^i_{n_i+1}(1)\le\tau\mid V^i_{\mathbf{n}-\mathbf{e}_i}\le\theta^i_{n_i}), & n_i\ge 1.\end{cases} \tag{18}$$

Proof: Using similar techniques as in [24], we can present the proof. According to the CDF of exponential distribution, the proof for n_i=0 is immediate. Define

$V^i(t) \equiv$ the time a virtual job with infinite (no) deadline which arrives at the i-th processor's queue at time t must wait before it completes its service, (19)

$V^i_{\mathbf{n}}(t) \equiv$ the time a virtual job with infinite (no) deadline which arrives at the i-th processor's queue at time t must wait before it completes its service, given there are n_j jobs in the j-th processor's queue, j=1,…,s. (20)

Let T^i_k be the time of the k-th possible arrival of a job to the i-th processor's queue and S^i_k the time of the k-th possible arrival of a job who will successfully be served to the i-th processor's queue, k=1,2,... . For any time t, we also use t+ and $t-$ to denote a time immediately after and before t, respectively. Clearly, we have

$$V^i = \lim_{k\to\infty}V^i(T^i_k-),$$

$$V^i_{\mathbf{n}} = \lim_{k\to\infty}V^i_{\mathbf{n}}(T^i_k-),$$

where V^i and $V^i_{\mathbf{n}}$ are defined as in (14) and (17), respectively. Define

$$\tilde{V}^i_{\mathbf{n}} = \lim_{t\to\infty}V^i_{\mathbf{n}}(t), \tag{21}$$

$$\hat{V}^i_{\mathbf{n}} = \lim_{k\to\infty}V^i_{\mathbf{n}}(S^i_k+). \tag{22}$$

$\tilde{V}^i_{\mathbf{n}}$ represents the steady-state time average of the conditional offered sojourn time in the i-th processor's queue, given there are n_j jobs in the j-th processor's queue, j=1,…,s. $\hat{V}^i_{\mathbf{n}}$ represents the steady-state conditional sojourn time in the i-th processor's queue immediately after the arrival of a new successful job to the i-th processor's queue, given the new arrival finds $\mathbf{n}-\mathbf{e}_i$ jobs in system. Now, consider the system in the long run with $\mathbf{n}-\mathbf{e}_i$ jobs. Suppose, a new job arrives to the i-th processor's queue in this system which will successfully be served. Clearly, the offered sojourn time before and after this new arrival, may be represented as $V^i_{\mathbf{n}-\mathbf{e}_i}$, conditioned by the event $\{V^i_{\mathbf{n}-\mathbf{e}_i}\le\theta^i_{n_i}\}$, and $\hat{V}^i_{\mathbf{n}}$, respectively. Moreover, the offered sojourn time will increase immediately after the new arrival by exactly the same value as the execution time of the new job. Thus, we can write

$$P(\hat{V}^i_{\mathbf{n}}\le\tau)=P(V^i_{\mathbf{n}-\mathbf{e}_i}+E^i_{n_i+1}(1)\le\tau\mid V^i_{\mathbf{n}-\mathbf{e}_i}\le\theta^i_{n_i}), \tag{23}$$

where $E^i_{n_i+1}(1)$ is an exponentially distributed random variable with rate μ_i, representing the execution time of the new job, which is independent of $V^i_{\mathbf{n}-\mathbf{e}_i}$ and $\theta^i_{n_i}$. From conditional ASTA [20], we can write

$$P(\tilde{V}^i_{\mathbf{n}}\le\tau)=P(V^i_{\mathbf{n}}\le\tau). \tag{24}$$

Using the memoryless property of the execution times, we can also have

$$P(\hat{V}^i_{\mathbf{n}}\le\tau)=P(\tilde{V}^i_{\mathbf{n}}\le\tau). \tag{25}$$

From (23), (24), and (25) above, we finally get

$$P(V^i_{\mathbf{n}}\le\tau)=P(V^i_{\mathbf{n}-\mathbf{e}_i}+E^i_{n_i+1}(1)\le\tau\mid V^i_{\mathbf{n}-\mathbf{e}_i}\le\theta^i_{n_i}), \tag{26}$$

which completes the proof.☐

We now get along to derive the PDF of $V^i_{\mathbf{n}}$. Let

$$F_{V^i_{\mathbf{n}}}(\tau)=P(V^i_{\mathbf{n}}\le\tau),$$

$$f_{V^i_{\mathbf{n}}}(\tau)=\frac{dF_{V^i_{\mathbf{n}}}(\tau)}{d\tau}. \tag{27}$$

From Lemma 1, we have

$$F_{V^i_{\mathbf{n}}}(\tau)=\begin{cases}1-e^{-\mu_i\tau}, & n_i=0\\ \dfrac{\int_0^\tau(1-e^{-\mu_i(\tau-x)})(1-G(x))dF_{V^i_{\mathbf{n}-\mathbf{e}_i}}(x)}{P(V^i_{\mathbf{n}-\mathbf{e}_i}\le\theta^i_{n_i})}, & n_i\ge 1\end{cases} \tag{28}$$

or, equivalently,

$$f_{V^i_{\mathbf{n}}}(\tau)=\begin{cases}\mu_i e^{-\mu_i\tau}, & n_i=0\\ \dfrac{\mu_i e^{-\mu_i\tau}}{P(V^i_{\mathbf{n}-\mathbf{e}_i}\le\theta^i_{n_i})}\int_0^\tau f_{V^i_{\mathbf{n}-\mathbf{e}_i}}(x)e^{\mu_i x}(1-G(x))dx, & n_i\ge 1.\end{cases} \tag{29}$$

A solution for (29) may be given as

$$f_{V_{\mathbf{n}}^i}(\tau)=\begin{cases}\mu_i e^{-\mu_i\tau}, & n_i=0\\ \dfrac{\mu_i^{\,n_i+1}}{n_i!\prod_{k=1}^{n_i}P(V_{\mathbf{n}-k\mathbf{e}_i}^i\le\theta_{n_i-k+1}^i)} \left[\int_0^\tau(1-G(x))dx\right]^{n_i}e^{-\mu_i\tau}, & n_i\ge 1.\end{cases} \tag{30}$$

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

$$\Phi_n(s)=\int_0^\infty[\int_0^\tau(1-G(x))dx]^n e^{-s\tau}d\tau. \tag{31}$$

We can have the following lemmas:

Lemma 2. *Let* $f_{V_{\mathbf{n}}^i}(\tau)$ *be defined as in (27). Then*

$$f_{V_{\mathbf{n}}^i}(\tau)=\frac{1}{\Phi_{n_i}(\mu_i)}\left[\int_0^\tau(1-G(x))dx\right]^{n_i}e^{-\mu_i\tau}. \tag{32}$$

Proof: The proof is simple by noting that $f_{V_{\mathbf{n}}^i}(\tau)$ is a PDF and can also be derived as in (30).□

Lemma 3. *Let* $V_{\mathbf{n}}^i$ *be the conditional offered sojourn time of the i-th processor's queue, given the system is in state* $\mathbf{n}$, *as in (17) and* $\theta_{n_i}^i$ *be the relative deadline of the* n_i*-th job in the same processor's queue. Then*

$$P(V_{\mathbf{n}}^i\le\theta_{n_i+1}^i)=\frac{\mu_i\Phi_{n_i+1}(\mu_i)}{(n_i+1)\Phi_{n_i}(\mu_i)}. \tag{33}$$

Proof: Comparing (30) and (32), the proof is immediate.□

We are now in a position to give a closed-form solution for the conditional loss rates as follows:

Lemma 4. *Let* $\Gamma_{\mathbf{n}}^i$ *be defined as in (16). Then, we have*

$$\Gamma_{\mathbf{n}}^i=\begin{cases}0, & n_i=0,\\ n_i\dfrac{\Phi_{n_i-1}(\mu_i)}{\Phi_{n_i}(\mu_i)}-\mu_i, & n_i>0.\end{cases} \tag{34}$$

Proof: The proof is very similar to a lemma presented in [24] for the single queue case.□

3.2 Model Solution

This subsection presents a Markovian model for the analysis of the system described in Section 2. Let

$\pi(\mathbf{n})\equiv$ the probability that there are n_i jobs in the i-th queue, i=1,…,s, in the long-run. (35)

We can write

$$\begin{aligned}0=&\sum_{i\in q(\mathbf{n})}\left(\lambda(|\mathbf{n}|-1)r_{\mathbf{n}-\mathbf{e}_i}(i)\pi(\mathbf{n}-\mathbf{e}_i)\right)\\&-\left(\lambda(|\mathbf{n}|)+\sum_{i\in q(\mathbf{n})}(\mu_i+\Gamma_{\mathbf{n}}^i)\right)\pi(\mathbf{n})\\&+\sum_{i\in l(\mathbf{n})}\left((\mu_i+\Gamma_{\mathbf{n}+\mathbf{e}_i}^i)\pi(\mathbf{n}+\mathbf{e}_i)\right)\end{aligned} \tag{36}$$

It turns out that each equation in the above system of equations is dependent and may be derived from the other equations in the system. To have a set of independent equations, an equation in (36) may be replaced by the normalizing condition

$$\sum_{\mathbf{n}=\mathbf{0}}^{\mathbf{K}}\pi(\mathbf{n})=1, \tag{37}$$

where $\mathbf{K}=(K_1,...,K_s)$ is the vector of capacities for the parallel queues, showing the state in which the system is full. The resulting set of independent equations can then be solved using standard Markovian solution techniques to find $\pi(\mathbf{n})$. Let $\sigma(\mathbf{n})$ represent the steady-state probability that an incoming job finds the system in state $\mathbf{n}$. Thus, we have

$$\sigma(\mathbf{n})=\frac{\lambda(|\mathbf{n}|)\pi(\mathbf{n})}{\sum_{\mathbf{m}=\mathbf{0}}^{\mathbf{K}}\lambda(|\mathbf{m}|)\pi(\mathbf{m})}. \tag{38}$$

Subsequently, the PDF of V^i, namely the job offered sojourn time of the i-th processor's queue defined in (14), can be obtained as

$$f_{V^i}(\tau)=\sum_{\mathbf{n}=\mathbf{0}}^{\mathbf{K}}\sigma(\mathbf{n})r_{\mathbf{n}}(i)f_{V_{\mathbf{n}}^i}(\tau), \tag{39}$$

where $f_{V_{\mathbf{n}}^i}(\tau)$ and $\sigma(\mathbf{n})$ are defined in (32) and (38), respectively. Using (39), the PDF of job offered sojourn time in the parallel system is determined as

$$f_V(\tau)=\sum_{i=1}^s f_{V^i}(\tau). \tag{40}$$

Having $f_V(\tau)$, one can use (5) to compute the probability of missing deadline in the system, namely the probability that an incoming job misses its deadline in the system in the long run. The probability of blocking as in (6) can also be obtained as $\alpha_b=\sigma(\mathbf{K})$.

With a similar interpretation to (11), the job expected accrued utility at the i-th queue can also be derived as

$$\overline{\Omega}^i=\int_0^\infty f_{V^i}(\tau)\left(\int_\tau^\infty g(x)U(\tau,x)dx\right)d\tau. \tag{41}$$

The job expected accrued utility $\overline{\Omega}$ defined in (11), as the main performance measure of the FRT system under discussion, is calculated through

$$\overline{\Omega}=\sum_{i=1}^s\overline{\Omega}^i. \tag{42}$$

In order to find the value of $f_\Omega(.)$ at some specific utility ω, we first define a set $S(\omega,\theta)=\{\tau\,|\,U(\tau,\theta)=\omega\}$. 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 is ω. For an equation of 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

$$S(\omega,\theta)=S^{\mathrm{D}}(\omega,\theta)\cup S^{\mathrm{C}}(\omega,\theta). \tag{43}$$

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

$$f_\Omega^{\mathrm{D}}(\omega)=\int_0^\infty\sum_{\tau\in S^{\mathrm{D}}(\omega,x)}f_V(\tau)g(x)dx, \tag{44}$$

and for the continuous portion as

$$f_\Omega^{\mathrm{C}}(\omega)=\int_0^\infty\int_{S^{\mathrm{C}}(\omega,x)}f_V(\tau)g(x)d\tau\,dx. \tag{45}$$

Summing these two portions together, the PDF of job offered accrued utility Ω is calculated as

$$f_\Omega(\omega)=f_\Omega^{\mathrm{D}}(\omega)+f_\Omega^{\mathrm{C}}(\omega). \tag{46}$$

Next, using (46) and (12), the assurance level of satisfying a lower bound η on the accrued utility, i.e., $A(\eta)$ can be calculated.

4 Routing Policy and Case Studies

In this section, besides some well known DR policies, a utility-aware DR policy is presented. The policies' decisions are made according to the individual queue's population (given the system state **n**) and specifications (including the processing rate and capacity). Afterwards, in the following subsections, some elementary calculations are presented for two distributions of relative deadlines.

First, we introduce some extra notations to be able to express the behavior of the DR policies. Suppose $h(.)$ to be an arbitrary function of a processor's queue population and specifications and $h^k(.)$ be the respective function for the k-th processor's queue. $h^k(.)$ is considered as the decision parameter and determines how the DR policy behaves. Define

$$\Lambda_{\mathbf{n}}^{h(.)} = \{i \in l(\mathbf{n}) \mid \forall j \in l(\mathbf{n}), h^i(.) \geq h^j(.)\}, \tag{47}$$

where $l(\mathbf{n})$ is the set of non-full queues in state $\mathbf{n}$ as defined in Section 3. The set $\Lambda_{\mathbf{n}}^{h(.)}$ comprises all the parallel non-full queues which maximize $h(.)$, given the state $\mathbf{n}$. If $\left|\Lambda_{\mathbf{n}}^{h(.)}\right| = 1$, namely the set consists only of a unique queue, an arriving job joins the queue with probability 1. If the set contains more members, ties are broken by random selection of one of the queues in the set. More precisely, the joining probability of the respective DR policy is determined as follows:

$$r_{\mathbf{n}}(j) = \begin{cases} \dfrac{1}{|\Lambda_{\mathbf{n}}^{h(.)}|}, & j \in \Lambda_{\mathbf{n}}^{h(.)}, \\ 0, & \text{otherwise.} \end{cases} \tag{48}$$

As an example, by the consideration of $h(.) = -n$ (or equivalently $h^k(.) = -n_k$ for the k-th processor's queue), $r_{\mathbf{n}}(j)$ for the Joining Shortest Queue (JSQ) [10][16] DR policy can be obtained from (48) above. According to this policy, an arriving job joins the shortest non-full queue. Ties are broken by random selection of one of the non-full queues with the minimum number of jobs. JSQ for exponential [21] and deterministic [23] relative deadlines is known as an optimal DR policy for exponential homogeneous processors (i.e., processors with the same processing rates). However, JSQ is not necessarily a good policy for heterogeneous processors.

Minimum Expected Delay (MED) [19], as another well-known DR policy, is a generalization of JSQ for heterogeneous processors. $h(.) = -(n+1)/\mu$ (or equivalently $h^k(.) = -(n_k+1)/\mu_k$ for the k-th processor's queue) is supposed for this policy. Accordingly, an arriving job joins a non-full queue in which the minimum response time is expected for the job, assuming that the existing jobs in the queue remain there until they finish their service. Even though MED does not take the timing constraints of jobs into account and isn't designed for real-time systems; it can provide an excellent performance in non real-time systems with heterogeneous processors.

We also consider another DR policy called Minimum Expected Sojourn Time (MEST) with $h^k(.) = -\overline{V}_{\mathbf{n}}^k$, where

$$\overline{V}_{\mathbf{n}}^k = \int_0^\infty \tau . f_{V_{\mathbf{n}}^k}(\tau) d\tau. \tag{49}$$

$\overline{V}_{\mathbf{n}}^k$ is the expected sojourn time of a job joining to the k-th processor's queue that finds $\mathbf{n}$ as the system state before joining. As its name indicates, MEST assigns an incoming job to a non-full queue with the minimum expected response time, regarding that the aforehand jobs may leave the system due to deadline miss.

Utility-Aware Dynamic Routing: None of the above mentioned DR policies take utility performance measures into account for their decision makings. Meanwhile, most such policies try to minimize the jobs' response times, while this may be a good criteria for only some limited TUF shapes. In the following, we present a utility-aware DR policy called Maximum Expected Utility (MEU) to improve the utility-related QoS measures of the parallel systems for more TUFs.

Let $g(.)$ be the PDF of relative deadlines and $U(.)$ the jobs' TUF as defined in (1). Likewise, $f_{V_{\mathbf{n}}^i}(.)$ as in (32), is the PDF of the conditional offered sojourn time of the i-th processor's queue, given the system state $\mathbf{n}$. Then, the conditional job expected utility for the i-th processor's queue, given the system state $\mathbf{n}$, is defined as

$$\overline{\Omega}_{\mathbf{n}}^i = \int_0^\infty f_{V_{\mathbf{n}}^i}(\tau) (\int_\tau^\infty g(x) U(\tau, x) dx) d\tau. \tag{50}$$

The interpretation of (50) is near to that of (11). $\overline{\Omega}_{\mathbf{n}}^i$ specifies the expected accrued utility of an arriving job in state $\mathbf{n}$ that joins the i-th queue, which is a function of only the parameters of the i-th processor's queue. Consequently, the state-dependent joining probability of the MEU policy, namely $r_{\mathbf{n}}(j)$, is determined by the consideration of $h^k(.) = \overline{\Omega}_{\mathbf{n}}^k$. According to this policy, an arriving job is randomly joined to one of the parallel non-full queues with the largest conditional expected utility.

The behavior of MEU policy for two distributions of relative deadline and different TUF shapes is compared to JSQ, MED, and MEST in Section 5. The remainder of this section is designated to the required calculations for these two distributions.

4.1 Deterministic Relative Deadline

In this subsection, the deterministic distribution for job relative deadline is considered. For this distribution, we obtain the formulations of $\Gamma_{\mathbf{n}}^i$ and $f_{V_{\mathbf{n}}^i}(\tau)$ as the basic parameters for the computation of the DR policies performance measures and $\overline{\Omega}_{\mathbf{n}}^i$ for the setup of the MEU policy.

The PDF of relative deadline θ with deterministic distribution is given by

$$g_D(\tau) = \begin{cases} \delta(\tau), & \text{for } \tau = \overline{\theta}, \\ 0, & \text{for } \tau \neq \overline{\theta}, \end{cases} \tag{51}$$

where $\overline{\theta}$ is a constant denoting the mean job relative deadline and $\delta(\tau)$ is a Dirac delta (impulse) function. Using (31), we get

$$\Phi_{n_i}(\mu_i) = \frac{n_i!}{\mu_i^{n_i+1}} F_{E^i(n_i)}(\overline{\theta}), \tag{52}$$

where $F_{E^i(n)}(\overline{\theta})$ is defined as in (13). Using (34), we find

$$\Gamma_{\mathbf{n}}^i = \begin{cases} 0, & n_i = 0, \\ \mu_i \left(\dfrac{F_{E^i(n_i-1)}(\overline{\theta})}{F_{E^i(n_i)}(\overline{\theta})} - 1 \right), & n_i > 0. \end{cases} \tag{53}$$

Further, the PDF of the conditional offered sojourn time of the i-th processor's queue, $f_{V_{\mathbf{n}}^i}(\tau)$, is given by

$$f_{V_{\mathbf{n}}^i}(\tau) = \begin{cases} \dfrac{\mu_i^{n_i+1}}{n_i! F_{E^i(n_i)}(\overline{\theta})} \tau^{n_i} e^{-\mu_i \tau}, & \tau < \overline{\theta}, \\ \dfrac{\mu_i^{n_i+1}}{n_i! F_{E^i(n_i)}(\overline{\theta})} \overline{\theta}^{n_i} e^{-\mu_i \tau}, & \tau \geq \overline{\theta}, \end{cases} \tag{54}$$

using (32) and (51). The conditional expected utility of the i-th processor's queue, as defined in (50), which is used in the determination of the state-dependent joining probabilities of the MEU policy, can also be obtained as

$$\overline{\Omega}^i_{\mathbf{n}} = \int_0^{\bar{\theta}} \frac{\mu_i^{n_i+1}}{n_i!\,F_{E^i(n_i)}(\bar{\theta})} \tau^{n_i} e^{-\mu_i \tau} U(\tau,\bar{\theta})\,d\tau \tag{55}$$

for this distribution. Subsequently, using (48) to calculate $r_{\mathbf{n}}(j)$ for an arbitrary DR policy such as JSQ, MED, MEST, and MEU, and then, following the solution method presented in Subsection 3.2, the other performance measures and parameters for the DR policy can be calculated.

4.2 Exponential Relative Deadline

In this subsection, we consider the exponential distribution for job relative deadline and follow a scenario similar to Subsection 4.1. The PDF of relative deadline θ with exponential distribution is given by

$$g_E(\tau) = \frac{1}{\bar{\theta}} e^{-\frac{\tau}{\bar{\theta}}}, \tag{56}$$

where $\bar{\theta}$ is the mean job relative deadline. Then, using (31) and (34) respectively, we obtain

$$\Phi_{n_i}(\mu_i) = \frac{n_i!}{\prod_{k=0}^{n_i}(\mu_i + \frac{k}{\bar{\theta}})}, \tag{57}$$

$$\Gamma^i_{\mathbf{n}} = \frac{n_i}{\bar{\theta}}. \tag{58}$$

Likewise, the PDF of the conditional offered sojourn time of the i-th processor's queue, $f_{V^i_{\mathbf{n}}}(\tau)$, is derived as

$$f_{V^i_{\mathbf{n}}}(\tau) = \frac{\bar{\theta}^{n_i} \prod_{k=0}^{n_i}(\mu_i + \frac{k}{\bar{\theta}})}{n_i!}(1-e^{-\frac{\tau}{\bar{\theta}}})^{n_i} e^{-\mu_i \tau}. \tag{59}$$

The conditional expected utility of the i-th processor's queue, as defined in (50), which is used in the determination of the state-dependent joining probabilities of the MEU policy, can then be obtained as

$$\overline{\Omega}^i_{\mathbf{n}} = \int_0^{\infty} \frac{\bar{\theta}^{n_i-1} \prod_{k=0}^{n_i}(\mu_i + \frac{k}{\bar{\theta}})}{n_i!}(1-e^{-\frac{\tau}{\bar{\theta}}})^{n_i} e^{-\mu_i \tau} \left(\int_\tau^\infty e^{-\frac{x}{\bar{\theta}}} U(\tau,x)\,dx\right) d\tau \tag{60}$$

for the exponential relative deadline.

The other performance measures and parameters for an arbitrary DR policy can then be calculated through the solution method presented in Subsection 3.2. More tangible results for the deterministic and exponential relative deadlines are presented in Section 5.

5 Numerical Evaluation

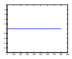
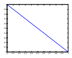
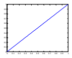
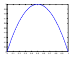
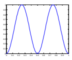
In this section, through numerical examples, a comparative study for four DR policies in a parallel system is carried out under different parameter settings. The study is done for the job expected accrued utility, namely $\overline{\Omega}$, as one important performance measure in TUF-constrained FRT systems. In the following subsections, first the parameter settings are specified, and then, the numerical results are evaluated.

5.1 Sample Configurations

The studies have been done for two parallel queues with different capacities, namely Q_1:(K_1=5,μ_1) and Q_2:(K_2=4,μ_2). Further, both cases of homogeneous and heterogeneous processors are investigated. More precisely, for the case of homogeneous processors, the processing rates are considered as μ_1=μ_2=1. Also, for the case of heterogeneous processors, they are assumed as μ_1=2 and μ_2=1.

TABLE 1

$U(.)$ FOR DIFFERENT TUF TYPES

Type	$U(.)$	TUF shape
I (Binary-valued downward step)	1	
II (Non-increasing)	$1-\frac{t}{D}$	
III (Non-decreasing)	$\frac{t}{D}$	
IV (Bell-shaped)	$4\frac{t}{D}(1-\frac{t}{D})$	
V (Two-bell)	$Sin^2(\frac{2\pi.t}{D})$	

Job arrivals to the system follow a Poisson process of constant rate $\lambda(|\mathbf{n}|) = \lambda$. Two distributions of relative deadline are studied, i.e., deterministic and exponential, for which the elementary calculations are described in Subsections 4.1 and 4.2, respectively. We assume $\bar{\theta} = 4$ as the mean relative deadline for both distributions. The studies have been done for five different types of TUFs, namely binary-valued downward step, non-increasing, non-decreasing, bell-shaped, and two-bell functions, referred to as Types I, II, III, IV, and V, respectively. TUF Type I is the classic al deadline. AWACS tracker [30] 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 [30] is also an example for functions with one peak, similar to TUF Type IV. TUF Type V is also a sample for the most general shape of such functions, namely multimodal TUFs. More details about the functions considered in the experiments of this section are summarized in Table 1. All the functions are supposed to take only values in the range of [0, 1].

Thus, we study the behavior of the four stationary DR policies defined in Section 4, namely JSQ, MED, MEST, and MEU. The comparative study presented in this section, which also validated by extensive simulations, show the relative behavior of these four policies with respect to variations of TUF and traffic intensity for both homogeneous and heterogeneous configurations.

5.2 Numerical Results

As defined in Section 4, $h^k(.)$ is the elementary function used to determine the joining probabilities $r_{\mathbf{n}}(.)$ for a specific DR policy. The values of this function for the deterministic relative deadlines and both homogeneous and heterogeneous configurations are shown in Tables 2 and

3, respectively. Similar values for the exponential relative deadlines are shown in Tables 4 and 5. Since the values for MEU differ for various TUF types, the respective values for each TUF type are represented separately, specified as MEU-I to MEU-V.

First, we consider the homogeneous configuration. According to the information presented in Tables 2 and 4, it is obvious that JSQ, MED, and MEST policies have exactly the same behavior for the case of homogeneous processors. (Thus, we consider JSQ as the indicator.) It is also obvious that MEU for TUF Types I and II behaves exactly as these policies. According to this fact, the respective curves are not presented for these TUF types. However, TUF Types III, IV, and V are discussed: As can be seen in Figs. 1(a), 1(b), and 1(c), for the deterministic distribution and the latter TUF types, MEU accrues considerable more utility than JSQ for low to moderate traffic intensities and almost the same utility as JSQ for moderate to high traffic intensities. Similar behaviors with smaller differences between MEU and JSQ can be observed for the exponential distribution, as shown in Figs. 2(a), 2(b), and 2(c).

The justifications for these behaviors can be drawn from Tables 2 and 4. One can simply infer that a successful DR policy is the one that with higher probabilities $\pi(\mathbf{n})$ puts the system into states $\mathbf{n}$ in which the routes with the high expected utilities could be selected. First, we discuss on low to moderate traffic intensities wherein the expected queue populations are oftentimes less than three. In the tables, noting to the differences between the expected utilities of queue populations n_k at which the utilities are higher and those of their adjacent populations, we find that for TUF Type IV, the difference is trenchant for deterministic distribution and milder for exponential distribution. Following the behavior of MEU for this TUF type based on the values, we find that, in spite of the former distribution, the behavior of MEU for the latter distribution is very close to JSQ. Also, wherever it deviates trivially from JSQ, the expected benefit is small. Accordingly, JSQ behaves analogous to MEU for the exponential distribution. Almost similar justifications can also be provided for TUF Type V. On the other hand, for heavy traffic intensities, the chances of finding the queues in the states with the high expected utilities as well as the choices for route selections are more limited, and therefore, both DR policies converge together.

Second, we discuss the heterogeneous configuration. As can be observed in Figs. 1(d) and 1(e) for TUF Types I and II, MEU, MEST, and MED behave almost similar to each other for the deterministic relative deadline. Also, they behave better than JSQ for light traffics. However, according to Fig. 1(f), MEU considerably outperforms the other policies for TUF Type III, especially for low to moderate arrival rates. For large arrival rates, due to the limited choices of route selections, all the policies converge together. Figs. 2(d), 2(e), and 2(f) show the relative behavior of these policies for the exponential distribution and TUF Types I, II, and III, respectively. For TUF Type I, the behaviors of MEU and MED are exactly the same (see also Table 5). For Types II and III of TUFs, MEU performs better than the other policies, whereas, the difference is trenchant for Type III of TUFs. It should be noted that trying to reduce the response time, as is done by JSQ, MED, and MEST, is consistent with accruing more utility for TUF Types I and II, not for Type III. Conversely, for the latter type, it is desired to complete the service of a successful job as late as possible. Hence, in spite of these three policies, MEU exposes an excellent behavior for TUF Type III.

The relative performance of the policies is different for Types IV and V of TUFs. As can be seen in Fig. 1(g) for TUF Type IV and the deterministic relative deadline, for light to moderate traffics, MEU outperforms the other policies, whereas for heavy traffics, it converges to MED and MEST. The same behavior for light to moderate traffics is iterated for the exponential relative deadline. However, for moderate to heavy traffics, MEST and then MED absolutely outperform MEU (see Fig. 2(g)). Considering TUF Type V, for the deterministic relative deadline, MED, MEST, and then JSQ utterly get ahead of MEU for moderate to heavy traffics (Fig. 1(h)). On the other hand, for the exponential relative deadline, MEST and then MED almost converge to MEU for heavy traffics (Fig. 2(h)).

These relative behaviors of the DR policies can better be explained by the information presented in Tables 3 and 5. However, we pay attention to the relative behavior of MED and MEU. (Similar discussions can be presented for MEST and MEU.) As indicated above, for the deterministic distribution and TUF Type V as well as the exponential distribution and TUF Type IV, for some specific ranges of traffic intensity, MED absolutely outperforms MEU. Substantially, wherever MEU behaves weak, it infrequently leads the system to the states $\mathbf{n}$ in which higher conditional utilities are expected. By the consideration of the respective values of $h^k(.)$ in Tables 3 and 5, and due to the fact that MEU is a greedy DR policy, the arriving jobs to the empty system start to join the 2nd queue (Q_2) and almost fill out all the capacity of this queue before starting to join the 1st queue (Q_1). However, these selected routes lead to only trivial amounts of extra utility. Whereas, partial drift from this routing could result in considerable higher accrued utilities if some jobs were joined to Q_1 and subsequent jobs could be expected to see at least 1 job in that queue. However, it is obvious that the validity of such an expectation is completely affected by the arrival rate of the real-time jobs λ. Therefore, it is not anticipated to propose an arrival-rate independent optimal DR policy for general TUFs.

In summary, for homogeneous processors and all the TUF types as well as heterogeneous processors and TUF Types I, II, and III, a routing policy such as MEU, may show quite good results irrespective of the arrival rate. However, due to the greedy nature and arrival-rate independence of MEU, it may deviate from an optimal solution for some TUFs. Further, we can conclude from the results that the efficiency of MEU as well as the other policies can be affected by various distributions of relative deadlines.

6 Related Work

In this section, we first present an overview on some

TABLE 2

THE VALUES OF $h^k(.)$ FOR A HOMOGENEOUS SYSTEM WITH μ_1=1, μ_2=1 AND DETERMINISTIC RELATIVE DEADLINE

n_k	$h^k(.)$															
	JSQ		MED		MEST		MEU-I		MEU-II		MEU-III		MEU-IV		MEU-V	
	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2
0	0	0	-1	-1	-0.908	-0.908	0.982	0.982	0.755	0.755	0.227	0.227	0.527	0.527	0.446	0.446
1	-1	-1	-2	-2	-1.552	-1.552	0.925	0.925	0.537	0.537	0.388	0.388	0.687	0.687	0.504	0.504
2	-2	-2	-3	-3	-1.871	-1.871	0.839	0.839	0.371	0.371	0.468	0.468	0.645	0.645	0.436	0.436
3	-3	-3	-4	-4	-1.949	-1.949	0.744	0.744	0.256	0.256	0.487	0.487	0.539	0.539	0.373	0.373
4	-4	-4	-5	-5	-1.896	-1.896	0.655	0.655	0.181	0.181	0.474	0.474	0.431	0.431	0.326	0.326
5	-5	----	-6	----	-1.789	----	0.579	----	0.132	----	0.447	----	0.343	----	0.284	----

TABLE 3

THE VALUES OF $h^k(.)$ FOR A HETEROGENEOUS SYSTEM WITH μ_1=2, μ_2=1 AND DETERMINISTIC RELATIVE DEADLINE

n_k	$h^k(.)$															
	JSQ		MED		MEST		MEU-I		MEU-II		MEU-III		MEU-IV		MEU-V	
	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2
0	0	0	-0.5	-1	-0.498	-0.908	0.999	0.982	0.875	0.755	0.125	0.227	0.375	0.527	0.356	0.446
1	-1	-1	-1	-2	-0.987	-1.552	0.997	0.925	0.751	0.537	0.247	0.388	0.627	0.687	0.560	0.504
2	-2	-2	-1.5	-3	-1.441	-1.871	0.989	0.839	0.629	0.371	0.360	0.468	0.763	0.645	0.573	0.436
3	-3	-3	-2	-4	-1.826	-1.949	0.971	0.744	0.515	0.256	0.456	0.487	0.801	0.539	0.516	0.373
4	-4	-4	-2.5	-5	-2.111	-1.896	0.940	0.655	0.412	0.181	0.528	0.474	0.767	0.431	0.469	0.326
5	-5	----	-3	----	-2.288	----	0.898	----	0.326	----	0.572	----	0.693	----	0.444	----

TABLE 4

THE VALUES OF $h^k(.)$ FOR A HOMOGENEOUS SYSTEM WITH μ_1=1, μ_2=1 AND EXPONENTIAL RELATIVE DEADLINE

n_k	$h^k(.)$															
	JSQ		MED		MEST		MEU-I		MEU-II		MEU-III		MEU-IV		MEU-V	
	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2
0	0	0	-1	-1	-1.000	-1.000	0.800	0.800	0.598	0.598	0.202	0.202	0.414	0.414	0.342	0.342
1	-1	-1	-2	-2	-1.800	-1.800	0.667	0.667	0.422	0.422	0.245	0.245	0.444	0.444	0.353	0.353
2	-2	-2	-3	-3	-2.467	-2.467	0.571	0.571	0.322	0.322	0.250	0.250	0.414	0.414	0.314	0.314
3	-3	-3	-4	-4	-3.038	-3.038	0.500	0.500	0.258	0.258	0.242	0.242	0.374	0.374	0.274	0.274
4	-4	-4	-5	-5	-3.538	-3.538	0.444	0.444	0.214	0.214	0.231	0.231	0.337	0.337	0.240	0.240
5	-5	----	-6	----	-3.983	----	0.400	----	0.182	----	0.218	----	0.305	----	0.212	----

TABLE 5

THE VALUES OF $h^k(.)$ FOR A HETEROGENEOUS SYSTEM WITH μ_1=2, μ_2=1 AND EXPONENTIAL RELATIVE DEADLINE

n_k	$h^k(.)$															
	JSQ		MED		MEST		MEU-I		MEU-II		MEU-III		MEU-IV		MEU-V	
	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2	k=1	k=2
0	0	0	-0.5	-1	-0.500	-1.000	0.889	0.800	0.725	0.597	0.164	0.202	0.373	0.414	0.316	0.342
1	-1	-1	-1	-2	-0.944	-1.800	0.800	0.667	0.575	0.422	0.225	0.245	0.464	0.444	0.391	0.353
2	-2	-2	-1.5	-3	-1.344	-2.467	0.727	0.571	0.476	0.322	0.252	0.250	0.481	0.414	0.394	0.314
3	-3	-3	-2	-4	-1.708	-3.038	0.667	0.500	0.405	0.258	0.262	0.242	0.471	0.374	0.373	0.274
4	-4	-4	-2.5	-5	-2.041	-3.538	0.615	0.444	0.351	0.214	0.265	0.231	0.452	0.337	0.347	0.240
5	-5	----	-3	----	-2.349	----	0.571	----	0.309	----	0.262	----	0.429	----	0.320	----

studies on QoS-aware distributed non real-time and deadline-based real-time systems. Then, some of the most related works to the application model of the current study are reviewed. These studies include scheduling algorithms for TUF-constrained single-processor and Symmetric MultiProcessor (SMP) real-time systems. Afterwards, some existing studies on distributed TUF-constrained real-time systems are discussed. Finally, we have made a review on some efforts on modeling and analysis of DR policies in non real-time and deadline-based real-time systems.

Providing better routing policies to improve some

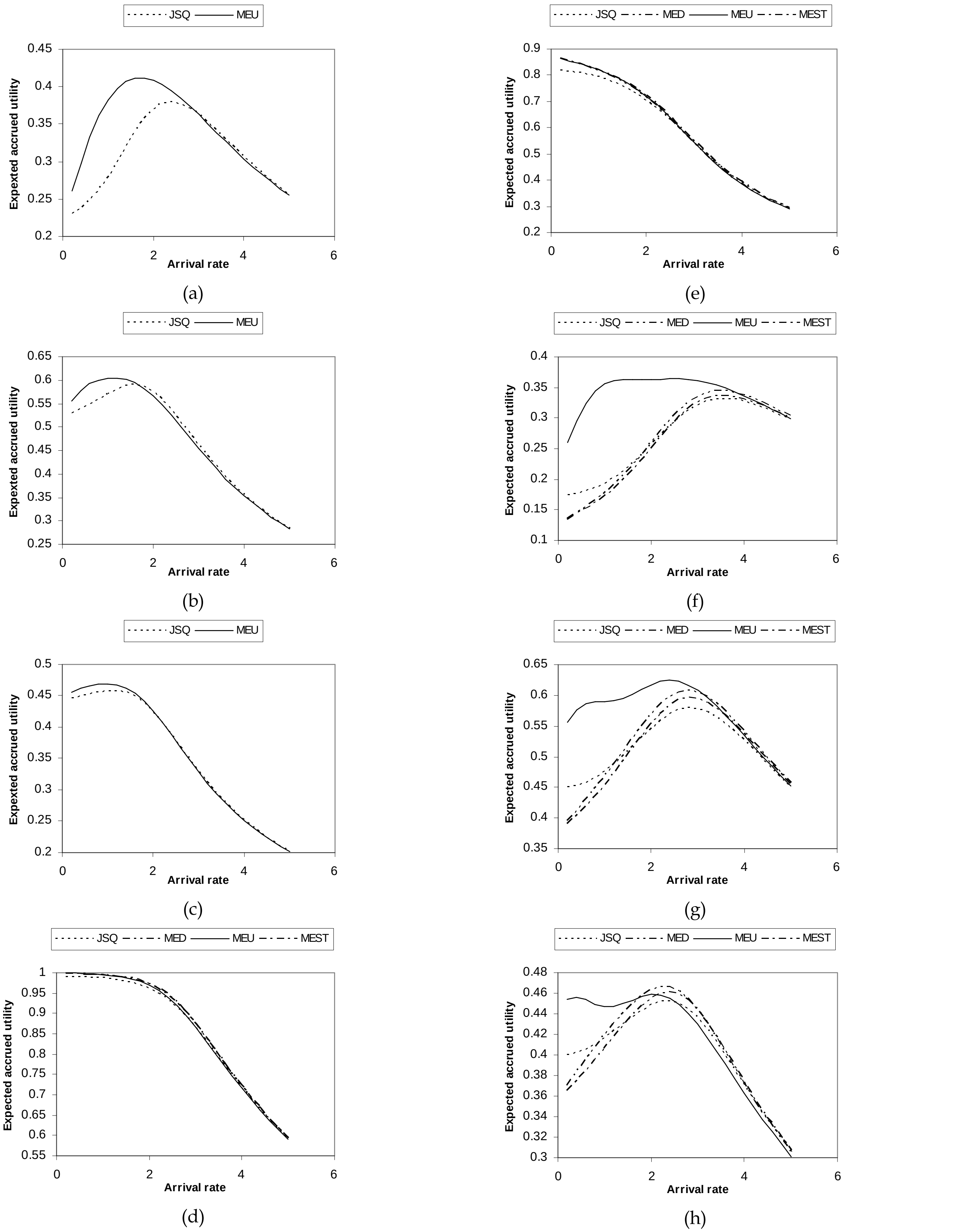


Fig. 1. Job expected accrued utility for the deterministic relative deadline, (a) homogeneous and TUF Type III, (b) homogeneous and TUF Type IV, (c) homogeneous and TUF Type V, (d) heterogeneous and TUF Type I, (e) heterogeneous and TUF Type II, (f) heterogeneous and TUF Type III, (g) heterogeneous and TUF Type IV, (h) heterogeneous and TUF Type V.

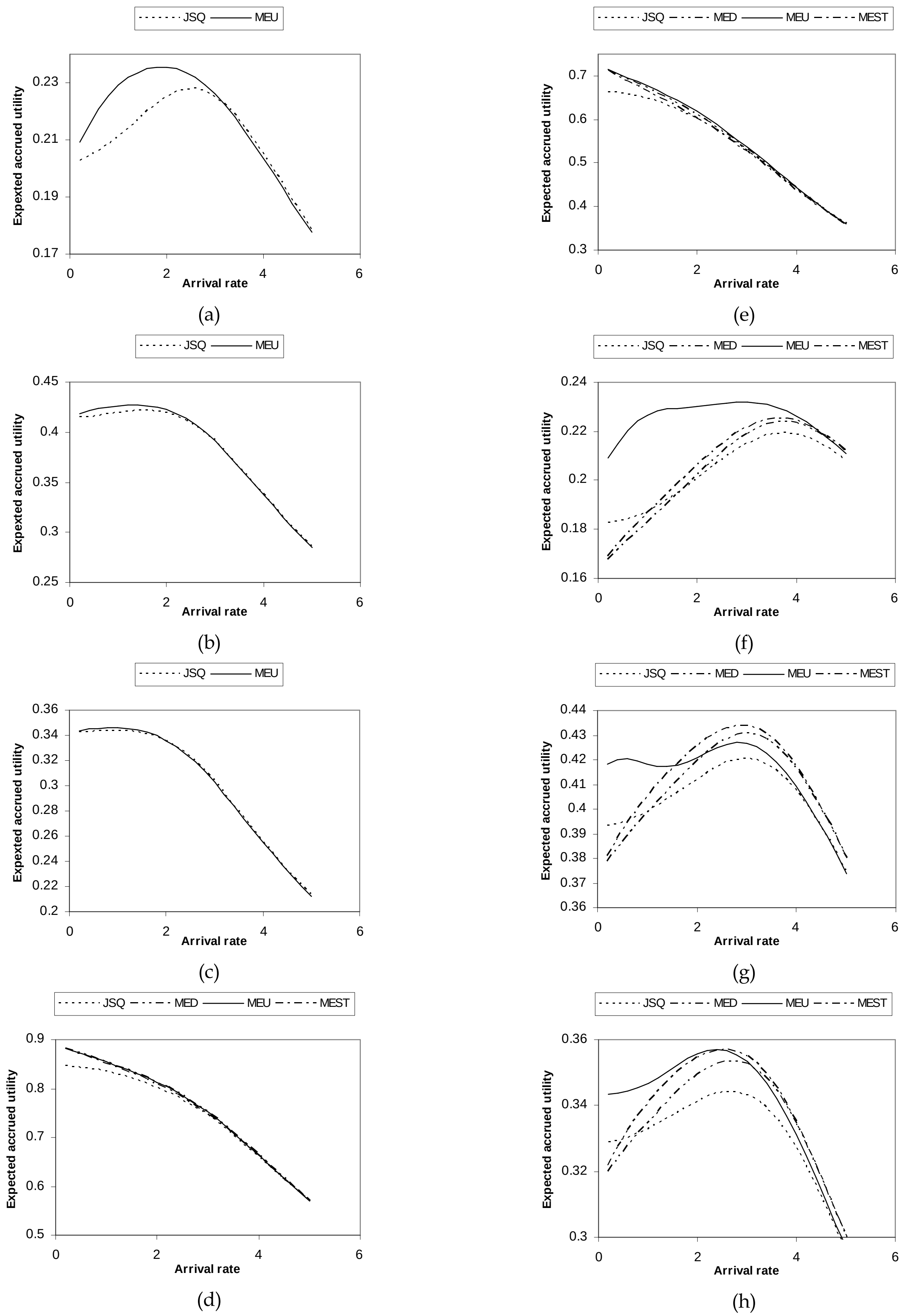


Fig. 2. Job expected accrued utility for the exponential relative deadline, (a) homogeneous and TUF Type III, (b) homogeneous and TUF Type IV, (c) homogeneous and TUF Type V, (d) heterogeneous and TUF Type I, (e) heterogeneous and TUF Type II, (f) heterogeneous and TUF Type III, (g) heterogeneous and TUF Type IV, (h) heterogeneous and TUF Type V.

specific QoS criteria (e.g., criteria related to load balancing, security, energy, etc.) in distributed systems has been thoroughly addressed using heuristic approaches. For example, in [29], MELISA and LBA have been proposed as adaptive load balancing algorithms for migratory non real-time heterogeneous grids. In [8], a distributed dynamic load balancing technique with migration among heterogeneous nodes with different computing and communication delays is proposed. The method is based on analytical characterization of the average overall completion time in a distributed system. Although this work is based on a rigid foundation, however, no explicit deadlines have been considered for the workload. In [39], AdaptLoad as an adaptive policy for load balancing of non real-time traffics among homogeneous clustered Web servers is presented.

Most studies on real-time systems focus on deadline-based algorithms, i.e., algorithms whose main target is meeting jobs' deadlines. Examples of such algorithms for single-processor systems are EDF and RM [17]. Similarly, the literature on distributed real-time systems is often concentrated on this kind of timing constraints. For example, [11] presents a polynomial-time distributed multicast routing algorithm which works according to the available bandwidth to satisfy the constraints of end-to-end delay and inter-destination delay variation. In [28], a dynamic repartitioning algorithm is suggested that dynamically balances the periodic real-time task loads of multiple homogeneous cores of a processor (via migration) to optimize dynamic power consumption during execution. In [38], two CPU allocation schemes are presented for a cluster of processors with a centralized queue for the admitted jobs. The algorithms take the jobs' timing and security requirements into account for processor assignments. The first algorithm, TAPADS, is applied to communicating jobs with precedence constraints which run on homogeneous clusters. Whereas, the other algorithm, SHARP, works for parallel jobs with no precedence constraints and no communications which run on heterogeneous clusters.

The studies on TUF-constrained real-time jobs mainly concentrate on single-processor systems and their respective scheduling. For independent jobs with step TUFs in an overloaded system, D^{over} is shown to have the optimal competitive factor [13], even though its average performance is quite poor for random jobs [14]. DASA without Dependency [7] 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 preemptable independent jobs is LBESA [18]. Assume a metric called Potential Utility Density (PUD) for a job as the ratio between the job expected utility and the remaining job execution time [18][37]. LBESA examines jobs in an EDF order and performs a feasibility check where it rejects jobs with lower PUDs until the schedule is feasible. Non-step TUFs are also studied by GUS [14] and RUA [34] algorithms which both use the concept of PUD. GUS allows resource sharing among jobs with arbitrary TUFs. RUA considers preemptable jobs in a FRT system subject to arbitrarily shaped TUFs 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 [15] provides probabilistic bounds on the task-level accrued utilities. Several more UA scheduling algorithms such as CMA [4], UPA [30] (which is shown to have higher accrued utility than EDF and CMA), and CUA [33] have also been developed. 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. Among these studies, only CMA and UPA consider non-preemptable jobs. Further, few of them present analytical solutions to their proposed algorithms (see [15], [35], and [36] for non-increasing TUFs). Meanwhile, they all have concentrated on single-processor systems.

In the context of multiprocessor scheduling, Cho et al. [5] present a non-quantum-based, migratory optimal scheduling algorithm called LLREF for step TUFs, periodic arrivals, and under-loads. LLREF works for SMP systems, i.e., systems with a single queue and homogeneous processors. The first UA multiprocessor real-time scheduling algorithm for non-step TUFs, gMUA [6], is also presented for SMP systems. The algorithm considers an application model with FRT jobs which are subject to non-increasing TUFs, variable execution time demands, and resource overloads, where the total job utilization demand exceeds the total capacity of all processors. gMUA, which again permits job migration, probabilistically satisfies lower bounds on individual job's accrued utility, as well as the system-wide total accrued utility.

In the realm of TUF-constrained distributed systems, [3] considers distributable threads in multi-hop networks which are subject to non-increasing TUFs. Thus, in the system, minimizing the path-distance (hop-count) is consistent with maximizing a thread utility. The paper proposes some heuristics to check the channels for minimum hop-count for distributable threads which is shown through simulation that significantly outperform OSPF. As another study, RTQoS [2] is proposed as a stateless scalable policy-based QoS architecture in the core routers of IP internetworks for SRT traffics with non-increasing TUFs. [37] at first categorizes UA scheduling algorithms based on their key decision-making metrics such as PUD (e.g., DASA and GUS) and deadline (e.g., LBESA). Then, for each class, presents class-appropriate TUF decomposition techniques that map the TUFs of distributable threads into sub-TUFs of the threads' segments and evaluates their effectiveness on the system QoS. In fact, [37] does not provide a routing algorithm for TUF-constrained distributable threads. The aforementioned algorithms for TUF-constrained distributed systems neither consider general TUFs nor present analytical solutions. However, they can be used to improve the QoS of specific distributed real-time systems.

Analytical modeling of multi-queue systems is also the concentration of some other studies. However, the literature on the analysis of dynamic routing of real-time jobs among a number of parallel queues is quite limited. To the best of our knowledge, most of the reported analytical studies on dynamic assignment of jobs to parallel queues consider non real-time jobs and especially the JSQ DR policy [25][26]. The importance of the JSQ policy is due to its optimality for non real-time systems with homogeneous processors [10][31]. Moreover, JSQ is a natural load balancing mechanism to minimize the average job response time. In [16], Lin and Raghavendra presented an accurate analytical model to estimate the performance of JSQ in terms of average response time of a system with homogeneous processors. On the other hand, a generalization of JSQ, namely MED DR policy has been analyzed in [19] in order to estimate the mean response time of jobs. The MED policy can provide an excellent performance in non real-time queueing systems with heterogeneous processors. However, when jobs are real-time, the problem becomes more complex. Among the few studies on the dynamic assignment of real-time jobs to parallel queues, Zhu in [40] presents an idea for approximating the performance of JSQ policy in a SRT system with the homogeneous configuration. In [21], it has been shown that even in a FRT system, JSQ is optimal in assigning jobs with exponentially distributed relative deadlines to a system with homogeneous exponential servers and finite capacity. In [23], it has also been shown that for jobs with deterministic relative deadlines in the same conditions, JSQ policy is again optimal. Moreover, in [22], an analytical method for the analysis of general DR policies for assignment of state-dependent Poisson arrival non TUF-constrained FRT jobs to a number of parallel queues is presented. The method works for the case of deadlines until the beginning of service, generally distributed relative deadlines, and exponential servers. In the paper, JSQ is also evaluated as a case study.

The main differences between this work and the above mentioned studies can be summarized as follows. None of the above methods presents an exact analytical solution for dynamic routing of TUF-constrained FRT jobs among parallel heterogeneous queues. Further, none of them presents a utility-aware dynamic routing policy for a similar system with arbitrary TUF shapes (except [3] for a different system model and only non-increasing TUFs). The few previous studies on assignment of TUF-constrained jobs to parallel subsystems/processors have some limitations from one or more of the following aspects: 1) they are evaluated by simulation, not based on analytical methods, 2) they have been presented for limited TUF shapes, 3) they work for SMP systems, and 4) they require job migration among processors.

7 Conclusions and Future Work

This paper proposes an exact analytical method for performance evaluation of DR policies in parallel TUF-based FRT systems with general distribution of relative deadlines. The DR policies make their decisions based on the queues' specifications and populations. Using the analytical method, some classical performance measures as well as some utility-related ones are calculated for the real-time systems. Further, a utility-aware DR policy, namely MEU, is proposed which is based on the conditional expected accrued utility of jobs on assignments to the individual queues of the parallel system.

The comparative study of MEU with respect to JSQ, MED, and MEST for a two-queue system and different TUF shapes show the efficiency of MEU in the homogeneous configuration as well as its good performance in most situations for the heterogeneous configuration. According to our results, policies other than MEU behave well for only specific TUFs. Meanwhile, it is shown that the distribution of relative deadlines can affect the relative efficacy of the DR policies. Further, analysis of the results shows that we cannot have an optimal utility-aware DR policy for arbitrary TUFs without the consideration of traffic intensity of the overall system.

Investigation of adaptive utility-aware DR policies with respect to the dynamics of workload characteristics is one idea that we are working on as a future study. Another idea is to study the effects of utility-aware scheduling algorithms on the behavior of utility-aware DR policies.

Acknowledgment

The authors wish to thank anonymous referees for their valuable comments and careful review of the paper.

References

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

[2] K. Channakeshava, K. Phanse, L.A. DaSilva, B. Ravindran, S. F. Midkiff, and E.D. Jensen, "IP Quality of Service Support for Soft Real-Time Applications," *4th Int. Workshop on Real-Time Networks (RTN), IEEE Euromicro Conf. Real-Time Systems*, pp. 1-4, 2005.

[3] K. Channakeshava, B. Ravindran, and E.D. Jensen, "Utility Accrual Channel Establishment in Multihop Networks," *IEEE Trans. Computers*, vol. 55, no. 4, pp. 428-442, 2006.

[4] 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, 1996.

[5] H. Cho, B. Ravindran, and E.D. Jensen, "An Optimal Real-Time Scheduling Algorithm for Multiprocessors," *Proc. IEEE Real-Time Systems Symp.*, pp. 101-110, 2006.

[6] H. Cho, H. Wu, B. Ravindran, and E.D. Jensen, "On Multiprocessor Utility Accrual Real-Time Scheduling with Statistical Timing Assurances," *Proc. Int. Conf. Embedded and Ubiquitous Computing, LNCS 4096*, pp. 274-286, 2006.

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

[8] S. Dhakal, M.M. Hayat, J.E. Pezoa, C. Yang, and D.A. Bader, "Dynamic Load Balancing in Distributed Systems in the Presence of Delays: A Regeneration-Theory Approach," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 4, pp. 485-497, 2007.

[9] 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, 2007.

[10] A. Hordijk and G. Kook, "On the Optimality of the Generalized Shortest Queue Policy," *J. Probability in the Engineering and Informational Sciences*, vol. 4, pp. 477-487, 1990.

[11] J. Huang, X. Du, Z. Yang, and W. Cheng, "Available Bandwidth-Based Real-Time Multicast Routing Distributed Algorithm," *Proc. Int. Conf. Computer Networks and Mobile Computing*, pp. 54-60, 2003.
[12] 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, 1985.
[13] 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, 1992.
[14] P. Li, "Utility Accrual Real-Time Scheduling: Models and Algorithms," Ph.D. dissertation, Virginia Polytechnic Institute and State Univ., Blacksburg, Virginia, 2004.
[15] 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.
[16] H.C. Lin and C.S. Raghavendra, "An Approximation Analysis of the Join the Shortest Queue (JSQ) Policy," *IEEE Trans. Parallel and Distributed Systems*, vol. 7, no. 3, pp. 301-307, 1996.
[17] 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.
[18] C.D. Locke, "Best-Effort Decision Making for Real-Time Scheduling," Ph.D. dissertation, Dept. Computer Eng., Carnegie Mellon Univ., Pittsburgh, 1986.
[19] J.C.S. Lui, R.R. Muntz, and D. Towsley, "Bounding the Mean Response Time of a Minimum Expected Delay Routing System: An Algorithmic Approach," *IEEE Trans. Computers*, vol. 44, no. 12, pp. 1371-1382, 1995.
[20] B. Melamed and D.D. Yao, *The ASTA Property*, Invited chapter in *Frontiers in Queuing: Models, Methods and Problems,* J. Dshalalow, Ed., CRC Press, 1995.
[21] A. Movaghar, "Optimal Assignment of Impatient Customers to Parallel Queues with Blocking," *Scientia Iranica*, vol. 3, pp. 137-146, 1997.
[22] A. Movaghar, "On Dynamic Assignment of Impatient Customers to Parallel Queues," *Proc. IEEE Int. Conf. Dependable Systems and Networks*, pp. 751-759, 2003.
[23] A. Movaghar, "Optimal Control of Parallel Queues with Impatient Customers," *Performance Evaluation*, vol. 60, no. 1-4, pp. 327-343, 2005.
[24] A. Movaghar, "On Queueing with Customer Impatience until the End of Service," *J. Stochastic Models*, vol. 22, pp. 149–173, 2006.
[25] R.D. Nelson and T.K. Philips, "An Approximation to the Response Time for Shortest Queue Routing," *Performance Evaluation Review*, vol. 17, no. 1, pp. 181-189, 1989.
[26] R.D. Nelson and T.K. Philips, "An Approximation for the Mean Response Time for Shortest Queue Routing with General Interarrival and Service Times," *Performance Evaluation*, vol. 17, pp. 123-139, 1993.
[27] 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.
[28] E. Seo, J. Jeong, S. Park, and J. Lee, "Energy Efficient Scheduling of Real-Time Tasks on Multicore Processors," *IEEE Trans. Parallel and Distributed Systems*, vol. 19, no. 11, pp. 1540-1552, 2008.
[29] R. Shah, B. Veeravalli, and M. Misra, "On the Design of Adaptive and Decentralized Load Balancing Algorithms with Load Estimation for Computational Grid Environments," *IEEE Trans. Parallel and Distributed Systems*, vol. 18, no. 12, pp. 1675-1686, 2007.
[30] 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, 2004.
[31] W. Winston, "Optimality of the Shortest Line Discipline," *J. Applied Probability*, vol. 14, pp. 181-189, 1977.
[32] H. Wu, "Energy-Efficient, Utility Accrual Real-Time Scheduling," Ph.D. dissertation, Virginia Polytechnic Institute and State Univ., Blacksburg, Virginia, 2005.
[33] 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, 2004.
[34] 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, 2004.
[35] 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, 2007.
[36] 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, 2004.
[37] 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, 2005.
[38] T. Xie and X. Qin, "Security-Aware Resource Allocation for Real-Time Parallel Jobs on Homogeneous and Heterogeneous Clusters," *IEEE Trans. Parallel and Distributed Systems*, vol. 19, no. 5, pp. 682-697, 2008.
[39] Q. Zhang, A. Riska, W. Sun, E. Smirni, and G. Ciardo, "Workload-Aware Load Balancing for Clustered Web Servers," *IEEE Trans. Parallel and Distributed Systems*, vol. 16, no. 3, pp. 219-233, 2005.
[40] W. Zhu, "An Approximate Analysis of the Shortest Queue Policy on Soft Real-Time Scheduling," *ACM SIGMETRICS Performance Evaluation Review*, vol. 28, no. 3, pp. 3-10, 2000.

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.