Bandwidth Allocation for Fixed-Priority-Scheduled Compositional Real-Time Systems

FARHANA DEWAN and NATHAN FISHER, Wayne State University

Recent research in compositional real-time systems has focused on determination of a component’s real-time interface parameters. An important objective in interface-parameter determination is minimizing the bandwidth allocated to each component of the system while simultaneously guaranteeing component schedulability. With this goal in mind, in this paper we explore fixed-priority schedulability in compositional setting. First we derive an efficient exact test based on iterative convergence for sporadic task systems scheduled by fixed-priority (e.g., deadline monotonic, rate monotonic) upon an Explicit-Deadline Periodic (EDP) resource. Then we address the time complexity of the exact test by developing a fully-polynomial-time approximation scheme (FPTAS) for allocating bandwidth to components. Our parametric algorithm takes the task system and an accuracy parameter \( \epsilon > 0 \) as input, and returns a bandwidth which is guaranteed to be at most a factor \( (1 + \epsilon) \) times the optimal minimum bandwidth required to successfully schedule the task system. We perform thorough simulation over synthetically generated task systems to compare the performance of our proposed efficient-exact and the approximate algorithm and observe a significant decrease in runtime and a very small relative error when comparing the approximate algorithm with the exact algorithm and the sufficient algorithm.

Categories and Subject Descriptors: D.4.7 [Operating Systems]: Organization and Design – Real-Time and Embedded Systems; C.3 [Special-Purpose and Application-Based Systems]: Real-Time and Embedded Systems

General Terms: Theory, Design, Algorithms

Additional Key Words and Phrases: Compositional Real-Time Systems, Bandwidth Allocation, FPTAS, Fixed-Priority Scheduling

ACM Reference Format:


1. INTRODUCTION

Recent research in real-time systems has focused on designing frameworks for enabling component-based design in real-time systems. State-of-the-art frameworks for compositional real-time systems include [Deng and Liu 1997; Feng and Mok 2002; Shin and Lee 2008; Albers et al. 2008]. Component-based design is highly desirable due to its well-known benefits of reducing overall system complexity and enhancing system designers’ understanding of the system. One of the major benefits of these systems is achieved by the goal of component abstraction, which hides the internal complexity and details of one component from developers of other components and only exposes information necessary to use the component via an interface. In most compositional real-time frameworks, a component uses a real-time interface to communicate with the other components of the system. The major difference of such interface with traditional functional interfaces for softwares or modules is the fact that the real-time interfaces represent the overall timing constraint of the component along with computational resource requirements. A component specifies its resource re-
requirements to meet its real-time constraints by the attribute *interface bandwidth*. Thus, an important design issue of these frameworks is addressing the problem *minimization of interface bandwidth* (MIB-RT).

One simple, yet flexible, real-time compositional framework is the *explicit-deadline periodic resource (EDP)* model [Easwaran et al. 2007; Shin and Lee 2008]. An EDP resource $Ω$ is characterized by a three-tuple $(Π, Θ, Δ)$ where $Π$ is referred to as the *period of repetition*, $Θ$ is the *capacity*, and $Δ$ is the *relative deadline*. The interpretation of such a resource is that a component $C$ executed upon $Ω$ is guaranteed $Θ$ units of processing resource supply for successive $Π$-length intervals (given some initial starting time). Furthermore, the $Θ$ units of resource supply must be provided within $Δ (≤ Π)$ time units after the start of the $Π$-length interval. In compositional systems, since more than one component share the processing resource (single processor in our case), when one component receives the processing supply, all the remaining components have “no-supply period” at that instant of time. The no-supply period of $Ω$ is the duration of time at which the component $C$ does not receive any processing resource; more formally, $Π − Θ$ units of time in any $Π$-interval of time in this case. The interface bandwidth of $C$ for this framework is the ratio of capacity and period ($\frac{Θ}{Π}$).

A system-level scheduling algorithm allocates the processor time among the different periodic resources that share the same processor, such that each resource receives (for every period) aggregate processor time equivalent to its capacity. A component’s tasks are then hierarchically scheduled by a component-level scheduling algorithm upon the processing time supplied to resource $Ω$.

In this paper, we obtain solutions to MIB-RT for an EDP resource when components use fixed-priority as the component-level scheduling algorithm. (The system-level scheduling algorithm is not considered for this paper). Specifically, we consider the problem of determining the optimal choice of capacity parameter (i.e., $Θ$) for an EDP resource $Ω$ with a fixed period $Π$ and deadline $Δ$ for component $C$. Algorithms exist for determining $Π$ and $Δ$ by searching over possible values and using a capacity-determination algorithm as a subroutine (e.g., see [Easwaran; Fisher 2009]); thus, since the search space may be quite large, efficient capacity-determination algorithms are necessary.

The MIB-RT problem for fixed-priority periodic resource model has been previously studied. An exact solution, based on exact schedulability techniques for uniprocessor real-time systems ([Baruah et al. 1993; Lehoczky et al. 1989]) has been proposed by Easwaran et al. [Easwaran et al. 2007]; however, the proposed solution has pseudo-polynomial-time complexity. There is also a $O(n)$-time sufficient solution to MIB-RT for periodic resource ($Π$ equals $Δ$) by Shin and Lee [Shin and Lee 2008]. The exact resource allocation is computationally expensive (pseudo-polynomial in this case) and thus might be impractical for algorithms that search for optimal values of $Π$ and $Δ$. On the other hand, though the sufficient resource allocation has lower (linear) computational complexity, these algorithms might provide over-estimated resource allocations and induce lower system utilization. This might be impractical for developing real-time systems in which resources are very scarce. However, in many real-time systems where tasks may be added or removed dynamically, it is important to provision resources efficiently at run-time and an efficient allocation algorithm is desirable. Our goal is to design an algorithm which is computationally efficient on real-time guarantee verification as well as to provide the system designer control over accuracy of resource allocation.

In the above setting, existing schedulability results for dedicated uniprocessor system [Lehoczky et al. 1989; Fisher and Baruah 2005] can be applied to obtain schedulability test for fixed-priority systems scheduled upon periodic-resources. A component consisting of sporadic tasks [Mok 1983] and scheduled upon periodic resource can be converted to an equivalent task system scheduled upon a dedicated resource by modeling the “no-supply period” of the periodic resource model as a special highest priority task [Okwudire et al. 2010]. However, the exact solution obtained in this approach is highly inefficient computationally, which makes this test impractical for real-time open system environment where application interface requires to change dynamically.

In our prior work [Fisher and Dewan 2012], we devised approximate bandwidth allocation algorithm for EDP resource with component level scheduling algorithm EDF. In this paper we extend those results for fixed-priority scheduled components. However, the compositional results for EDF
does not directly apply for fixed-priority, as we have to do maximum response time analysis for each task; this fundamentally differs from the demand-based approach of [Fisher and Dewan 2012].

§Our Contribution. For EDP resource model with sporadic tasks [Mok 1983] as components, first we derive a response time based exact schedulability condition similar to dedicated uniprocessor scheduling by considering the “no-supply period” of the EDP resource as special higher priority tasks. We derive heuristics (i.e., lower bound and upper bound for response time) to efficiently perform the schedulability test similar to [Davis et al. 2008]. A major contribution of this article is the development of a parametric approximation algorithm that addresses the current gap between computationally-expensive, exact solutions and computationally-inexpensive, sufficient solutions for MIB-RT problem:

Given $\Pi, \Delta, \text{task system } \tau$, and accuracy parameter $\epsilon > 0$, let $\Theta^*(\Pi, \Delta, \tau)$ be the optimal minimum capacity for $\tau$ to be fixed-priority-schedulable upon EDP resource $\Omega^* = (\Pi, \Theta^*(\Pi, \Delta, \tau), \Delta)$. Our algorithm returns $\hat{\Theta}$ for the given parameters where $\Theta^*(\Pi, \Delta, \tau) \leq \hat{\Theta} \leq (1 + \epsilon) \cdot \Theta^*(\Pi, \Delta, \tau)$. Furthermore, the time complexity of our algorithm is polynomial in the number of tasks in $\tau$ and $1/\epsilon$.

In other words, our algorithm is a fully-polynomial-time approximation scheme (FPTAS) for the MIB-RT problem with approximation ratio $(1 + \epsilon)$. This implies that the system designer can pre-specify an arbitrary level of accuracy in obtaining solution to MIB-RT with the tunable algorithm. The second major contribution of this article is a comprehensive comparison of our approximate algorithm with the previously-existing exact and sufficient algorithms via simulations.

In [Easwaran et al. 2009] Easwaran et al. applied compositional framework in avionics domain to schedule the ARINC 653 specified software components of the system. Our proposed algorithm can be applied in this setting which will significantly speed up the analysis while exploring the design parameters in design phase. Further, another potential area of application may be in designing thermally constrained real-time systems, where the power-aware components dynamically tune their interfaces to meet the temporal and thermal constrains [Hettiarchechi et al. 2012; Ahmed et al. 2011] with the change in environmental temperature. For systems with large number of thermal operating modes, exact schedulability analysis will take long time, whereas, an approximate bandwidth allocation technique can help the system achieve this goal efficiently.

§Organization. In Section 2, we briefly review the current literature on compositional frameworks and MIB-RT problem for fixed-priority scheduling. In Section 3, we give necessary notations required for the rest of the paper. For our capacity determination problem, we first derive an efficient exact schedulability test based on response time in Section 4. In Section 5, we present an approximate algorithm based on testing set points (defined in Section 3), and prove its correctness. In Section 6 we prove the approximation ration for our proposed algorithm. Simulation results comparing our algorithm with both previously-known exact and sufficient algorithms are given in Section 7.

2. RELATED WORK

The concept of compositional systems was first introduced by Deng and Liu [Deng and Liu 1997] in their work real-time open environments and Rajkumar et al. [Rajkumar et al. 2001] in their work resource kernels. Since then, researchers have proposed many different real-time compositional models and studied the MIB-RT problem of the proposed models. Feng and Mok [Feng and Mok 2002] proposed the concept of temporal partitions to support hierarchical sharing of a processing resource. Shin et al. [Shin and Lee 2003] proposed the related periodic resource model to characterize the supply guaranteed to any component in compositional system, which was generalized by Easwaran et al. [Easwaran et al. 2007] to explicit deadline periodic (EDP) resource model.

The resource allocation for fixed-priority scheduled periodic resource model has been previously studied for constrained deadline case (i.e., task deadline is less or equal task period). An exact solution based on exact schedulability techniques for uniprocessor real-time systems has been proposed by Easwaran et al. [Easwaran et al. 2007] for EDP resource. [Shin and Lee 2008] proposed an
$O(n)$-time sufficient solution for periodic resource (II equals $\Delta$). For the temporal partition models where components are scheduled by fixed-priority, [Lipari and Bini 2003] proposed exact, pseudo-polynomial time algorithm for resource allocation, and [Almeida and Pedreiras 2004] proposed sufficient, polynomial-time resource allocation techniques.

The schedulability analysis of fixed-priority-scheduled components upon periodic resources can be compared to existing fixed-priority servers [Lehoczky et al. 1987; Sprunt et al.; Sprunt et al. 1989; Strosnider et al. 1995] used to serve the aperiodic jobs in the system. When system consists of both periodic and aperiodic jobs, the periodic jobs require fraction of resource proportional to their utilization. A fixed-priority server reserves a fraction of resource for the upcoming aperiodic jobs, and serves them whenever such job arrives in the system depending on the server “budget”. Several fixed-priority servers have been proposed in the literature to schedule aperiodic and periodic jobs in system, for example, periodic server, polling server, deferrable server [Lehoczky et al. 1987; Strosnider et al. 1995], sporadic server [Sprunt et al. 1989] etc. In [Bernat and Burns 1999], a comprehensive comparison among deferrable and sporadic servers are shown.

Traditional uniprocessor schedulability analysis for sporadic task system with dedicated resource can be applied in compositional setting to obtain a solution to MIB-RT, by modeling the “no-supply period” of the periodic resource model as special tasks. The idea is basically similar to the fixed priority servers described in the previous paragraph, except that the special task has higher priority than all the other (periodic) tasks in the system. Using this approach, for two-level hierarchical system, [Okwudire et al. 2010] gave a linear-time sufficient schedulability test for fixed-priority scheduled components with harmonic task periods. We use similar approach to derive an exact schedulability test for compositional systems modeled by sporadic tasks (Section 4). Furthermore, response time bounds for such system can be derived similar to [Sjödin and Hansson 1998; Davis et al. 2008] to obtain an efficient exact test.

Thus, for a variety of models, relatively efficient, sufficient algorithms for MIB-RT have been proposed; however, the existence of any work on obtaining polynomial-time algorithms (prior to our work) with constant-factor approximation ratios where components are scheduled by fixed-priority is unknown. Although research by [Albers et al. 2008] has developed parametric algorithms, without known approximation ratios, for the hierarchical event stream model. The goal of our work is to fill this needed gap by obtaining an FPTAS for MIB-RT in the periodic resource model with uniprocessor platforms. In our preliminary work [Fisher and Dewan 2012], we obtained such ratios for the periodic resource model where components are scheduled by dynamic-priority (EDF). In this paper, our aim is to extend those results by developing an FPTAS for EDP framework where components are scheduled by fixed-priority (deadline-monotonic or rate-monotonic). The algorithm of [Fisher 2009] may be used in conjunction with the results of this paper to find an optimal period.

3. MODELS AND NOTATION

In this section, we present background and notation for the task model, workload functions, and periodic resource model that we use throughout the paper.

§Sporadic Task Model. A sporadic task $\tau_i = (e_i, d_i, p_i)$ is characterized by a worst-case execution requirement $e_i$, a (relative) deadline $d_i$, and a minimum inter-arrival separation $p_i$. Such a sporadic task generates a potentially infinite sequence of jobs, with successive job-arrivals separated by at least $p_i$ time units. A sporadic task system $\tau = \{\tau_1, \ldots, \tau_n\}$ is a collection of $n$ such sporadic tasks. We assume that the task system is constrained-deadline, that is, each task $\tau_i \in \tau$ has $d_i \leq p_i$. The task utilization for $\tau_i$ is defined as $u_i = e_i/p_i$ and the system utilization is denoted by $U(\tau) = \sum_{\tau_i \in \tau} u_i$, and $U(\tau) \leq 1$; otherwise, task system $\tau$ cannot be scheduled (by any algorithm) to meet all deadlines upon a dedicated preemptive uniprocessor.

We assume that task priorities are preassigned and each task has a fixed-priority. Tasks are indexed in non-increasing priority order. That is, $\tau_i$ has higher (or equal) priority than $\tau_j$, if and only if, $i \leq j$. As tasks generate jobs, each job inherits the priority of its generating task. For this paper, we assume each component uses fixed-priority scheduling as the component-level scheduling
algorithm. Whenever component $C$ is allocated the processor, $C$ executes the highest-priority job with remaining execution; ties are broken in favor of the job generated from the lower task index. The optimal fixed-priority scheduling algorithm for constrained-deadline sporadic tasks is **deadline monotonic** (DM) [Leung and Whitehead 1982], which assigns each task a priority equal to the inverse of its relative deadline (i.e., tasks with shorter relative deadlines have priority greater than tasks with longer relative deadlines).

§Workload Functions. For sporadic task systems, it is known that the worst-case phasing over any given interval is the **synchronous arrival sequence** of jobs. This occurs when all tasks of a sporadic task system release jobs at the same time instant and subsequent jobs as soon as permissible. Researchers [Lehoczky et al. 1989] have derived the **request-bound function**, as defined below.

**Definition 1 (Request-Bound Function).** For any $t > 0$ and sporadic task $\tau_i$, the **request-bound function** ($RBF$) quantifies the maximum cumulative execution requests that could be generated by jobs of $\tau_i$ arriving within a contiguous time-interval of length $t$. It has been shown that for sporadic tasks, $RBF$ can be calculated as follows [Lehoczky et al. 1989].

$$RBF(\tau_i, t) \triangleq \left\lfloor \frac{t}{p_i} \right\rfloor \cdot e_i. \quad (1)$$

Figure 1 shows the request-bound function for a sporadic task $\tau_i$, which is a right continuous function with discontinuities at time points of the form $t \equiv a \cdot p_i$ where $a \in \mathbb{N}$. The **cumulative request-bound function** for task $\tau_i$ is defined as follows:

$$R_i(t) \triangleq e_i + \sum_{j=1}^{i-1} RBF(\tau_j, t). \quad (2)$$

Audsley *et al.* [Audsley et al. 1991] have given a necessary and sufficient condition for sporadic task system $\tau$ to be fixed-priority-schedulable upon a preemptive uniprocessor platform of unit speed: $\exists t \in (0, d_i]$ such that $R_i(t) \leq t, \forall i$. Furthermore, it has also been shown [Audsley et al. 1993] that this condition needs to be verified at only time points in the following ordered set:

$$S_i(\tau) \triangleq \left\{ t = b \cdot p_a : a = 1, \ldots, i; b = 1, \ldots, \left\lfloor \frac{d_i}{p_a} \right\rfloor \right\} \cup \{d_i\}. \quad (3)$$

The above set is known as the **testing set** for sporadic task $\tau_i$. The size of this set may be as large as $\sum_{j=1}^{i} \left\lfloor \frac{d_j}{p_j} \right\rfloor$ which is dependent on the task periods, and thus requires pseudo-polynomial time feasibility test. Fisher and Baruah [Fisher and Baruah 2005] proposed the following approximation to $RBF$ to reduce the number of points in the testing set.

$$\delta(\tau_i, t, k) \triangleq \begin{cases} RBF(\tau_i, t), & \text{if } t \leq (k-1)p_i \\ e_i + \frac{t}{p_i}, & \text{otherwise.} \end{cases} \quad (4)$$
This function tracks RBF for exactly \( k - 1 \) steps and after the \( k - 1 \)-th step, it uses linear interpolation of subsequent discontinuous points of RBF (with slope equal to \( u_i \)). The steps in Figure 1 correspond to RBF(\( \tau_i, t, k \)), and the thick steps and the sloped-dashed line correspond to \( \delta(\tau_i, t, k) \).

The approximate cumulative request bound function is defined as follows:

\[
\hat{R}_i(t) \triangleq e_i + \sum_{j=1}^{i-1} \delta(\tau_j, t, k).
\]

(5)

For any fixed \( k \in \mathbb{N}^+ \), [Fisher and Baruah 2005] showed that if for all \( \tau_i \in \tau \) there exists a \( t \in (0, d_i] \) such that \( \hat{R}_i(t) \leq t \) then the sporadic task system \( \tau \) is static priority schedulable upon a preemptive uniprocessor platform of unit speed. The testing set for this condition is as follows:

\[
\hat{S}_i(\tau, k) \triangleq \{ t = b \cdot p_a \mid a = 1, \ldots, i - 1; b = 1, \ldots, k - 1; t \in (0, d_i] \} \cup \{ d_i \} \cup \{ 0 \}
\]

Let \( t_a, t_{a+1} \) denote any pair of consecutive values in the above ordered set.

Next, we give the relation between RBF and the approximate request bound function \( \delta \).

**Lemma 1 (from Fisher and Baruah 2005).** Given a fixed integer \( k \in \mathbb{N}^+ \), \( \text{RBF}(\tau_i, t) \leq \delta(\tau_i, t, k) \leq \left( \frac{k+1}{k} \right) \text{RBF}(\tau_i, t) \) for all \( \tau_i \in \tau \) and \( t \in \mathbb{R}_{\geq 0} \).

We will use this lemma in our approximation algorithm (Section 6).

Next, we define notation to represent the discontinuous line segments of the cumulative request bound function (\( \hat{R}_i \)). Let \( \mathcal{L}_i \equiv \{(t_a, D_{t_a}), (t_{a+1}, D_{t_{a+1}}), \alpha) \} \) be a line segment in the Euclidean space, \( \mathbb{R}^2 \), originating at open left end point \((t_a, D_{t_a})\) and ending at closed right end point \((t_{a+1}, D_{t_{a+1}})\). Let \( \alpha \) be a line in \( \mathbb{R}^2 \) with slope \( \alpha \geq 0 \) (Figure 1); more formally,

\[
\mathcal{L}_i \equiv \{(x, y) \in \mathbb{R}^2 \mid (x \in (t_a, t_{a+1})) \land (y = \alpha(x - t_a) + D_{t_a})\}.
\]

(7)

Please note the term \( \alpha \) is included in the notation for convenience only; it is possible to determine the slope from points \((t_a, D_{t_a})\) and \((t_{a+1}, D_{t_{a+1}})\) alone.

\[
\alpha \equiv \sum_{\tau_i \in \tau: \{t_a \geq (k-1)p_h, (b < i) \}} u_h.
\]

(8)

The connection between \( \mathcal{L}_i \) and \( \hat{R}_i \) is as follows. Consider a time \( t_a \in \hat{S}_i(\tau, k) \). Define \( D_{t_a} \) to be request bound function at time \( t_a \), that is, \( \hat{R}_i(t_a) \) (Figure 1). At time \( t_a \), some set of tasks with priority greater \( \tau_i \) have job arrivals in the synchronous arrival sequence. Let \( r_i(t) \) be the sum of the executions of these tasks. That is, \( r_i(t) \equiv \sum_{\tau_j \in \tau: \{ j < i \} \land (p_j \text{ divides } t) e_j \} \). At time \( t_a \) there is a discontinuity in the function \( \hat{R}_i \) in which \( \hat{R}_i \) increases by \( r_i(t_a) \) and then is linear until the next discontinuity in \( \hat{R}_i \) (i.e., at time \( t_{a+1} \in \hat{S}_i(\tau, k) \)). Thus, \( \delta_i \) is a line segment from \( t_a \) to \( t_{a+1} \) with slope equal to the total utilization of all task \( \tau_j \) such that \( j < i \) and \( t_a \geq (k-1)p_j \). We denote \( D_{t_a} \) by the sum of request bound function at time point \( t_a \) and job release at time \( t_a \), that is, \( D_{t_a} = D_{t_a} + r_i(t_a) \). From the above definitions of \( t_a, t_{a+1}, D_{t_a}, D_{t_{a+1}}, \) and \( \alpha \), it is straightforward to verify that \( \mathcal{L}_i \) is equivalent to \((t, \hat{R}_i(t))\) for all \( t \in (t_a, t_{a+1}] \). The exception is at \( t_a \) where \( D_{t_a} \) is not equal to \( \hat{R}_i(t_a) \); this difference exists for the notational and algebraic convenience throughout Section 5. From these definitions, the following lemma is apparent.

**Lemma 2.** For any consecutive pairs of values \((t_a, t_{a+1}) \in \hat{S}_i(\tau, k), \hat{R}_i(t) \leq D_t \) for all \((t, D_t) \in \mathcal{L}_i \).
Explicit-Deadline Periodic (EDP) Resource Model. An EDP resource, denoted by \( \Omega = (\Pi, \Theta, \Delta) \), guarantees that a component \( C \) executed upon resource \( \Omega \) receives at least \( \Theta \) units of execution between successive time points in \( \{ t \equiv t_0 + \ell \Pi \mid \ell \in \mathbb{N} \} \) where \( t_0 \) is some initial service start-time for the periodic resource. The \( \Theta \) units of service must occur \( \Delta \) units after each successive time point in the aforementioned set. Obviously, \( \Theta \leq \Delta \); for this paper, we make the simplifying assumption that \( \Delta \leq \Pi \), as well. The “no-supply period” of \( \Omega \) is the duration of time at which the component \( C \) does not receive any processing resource; i.e., \( \Pi - \Theta \) units of time in any \( \Pi \)-interval of time in this case. Furthermore, we assume in this paper that each component \( C \) is a sporadic task system \( \tau \) scheduled by fixed-priority upon \( \Omega \).

**Definition 2 (Supply-Bound Function).** For any \( t > 0 \), the supply-bound function (sbf) quantifies the minimum execution supply that a component executed upon periodic resource \( \Omega \) may receive over any interval of length \( t \). It is defined as follows [Easwaran et al. 2007]:

\[
\text{sbf}(\Omega, t) \overset{\text{def}}{=} \begin{cases} 
  y_\Omega t + \max(0, t - x_\Omega - y_\Omega t), & \text{if } t \geq \Delta - \Theta \\
  0, & \text{otherwise.}
\end{cases}
\]  

(9)

where \( y_\Omega \overset{\text{def}}{=} \left\lfloor \frac{\ell - (\Delta - \Theta)}{\Pi} \right\rfloor \) and \( x_\Omega \overset{\text{def}}{=} (\Pi + \Delta - 2\Theta) \).

We denote the optimal minimum capacity for \( \Omega \) given task system \( \tau \) by \( \Theta^*(\Pi, \Delta, \tau) \). We use the concept of \( \ell\)-feasibility region of \( \Omega \) similar to [Fisher and Dewan 2012] to define the region under the \( \ell\)-th step of sbf. For our convenience, we redefine \( \ell\)-feasibility region as follows:

**Definition 3 (\( \ell\)-Feasibility Region of \( \Omega \)).**

\[
\mathcal{F}_\ell(\Pi, \Delta, \Theta) \overset{\text{def}}{=} \left\{ (t, D_\ell) \in \mathbb{R}^2_{\geq 0} \mid \left( \Theta \geq \frac{D_\ell - t + \ell \Pi + \Delta}{\ell + 1} \right) \text{ and } \left( \Theta \geq \frac{D_\ell}{\ell} \right) \right\}.
\]  

(10)

Figure 2 shows graphical depiction of the supply bound function sbf for EDP resource \( \Omega \). The shaded region in the figure corresponds to the \( \ell\)-feasibility region for some step \( \ell \in \mathbb{N} \) of the sbf.

4. Determining Minimum Capacity Using Response Time

In this section, we derive an efficient exact schedulability test for fixed-priority scheduled component upon periodic resources similar to dedicated uniprocessor schedulability test. A fixed-priority scheduled component upon EDP resource \( \Omega \) can be considered as an equivalent dedicated uniprocessor system by adding special higher priority tasks to the original task system which corresponds to the “no-supply period” of \( \Omega \). The traditional response time based fixed-priority schedulability test [Lehoczky et al. 1989] can be applied to the modified task system.

Given a task system \( \tau \) and EDP resource \( \Omega = (\Pi, \Theta, \Delta) \), we derive fixed-priority schedulability by considering two special periodic tasks \( \tau_{-1} \) and \( \tau_{0} \) with higher priority then all the tasks in task system \( \tau \). Priority of \( \tau_{-1} \) is higher than priority of \( \tau_{0} \). Let \( \tau' = \{ \tau \cup \tau_{-1} \cup \tau_{0} \} \) be the new task system with \( \tau_{-1}(e_{-1}, d_{-1}, p_{-1}) \equiv (\Delta - \Theta, \Delta - \Theta, \infty) \) and \( \tau_{0}(e_0, p_0, d_0) \equiv (\Pi - \Theta, \Pi, \Pi) \). Let \( \tau_0 \) has a release jitter \( j_0 = e_{-1} \), and for all the other task \( \tau_i \in \tau' \), release jitter \( j_i = 0 \). This
assumption enables us to correctly model the initial starvation period of $\Pi + \Delta - 2\Theta$ units of the periodic resource $\Omega$. Clearly, these two tasks together represent the “no-supply period” of $\Omega$, where $\tau_{-1}$ accounts for initial non-recurring starvation period $\Delta - \Theta$, and $\tau_0$ accounts for the resource unavailability in every $t - (\Delta - \Theta)$ interval. The exact fixed-priority schedulability test of $\tau'$ is the reduction of the compositional schedulability test of $\tau$ with periodic resource $\Omega$.

The request bound function for the two special tasks $\tau_{-1}$ and $\tau_0$ is as follows.

$$RBF_0(t, \Omega) \equiv (\Delta - \Theta) + \max \left\{ 0, \left[ \frac{t - (\Delta - \Theta)}{\Pi} \right] \cdot (\Pi - \Theta) \right\}. \quad (11)$$

We obtain response time\(^1\) for all tasks $\tau_i \in \tau'$ as given by the following iterative equation.

$$R_i = e_i + \sum_{\forall j \in hp(i)} \left[ \frac{R_j}{p_j} e_j + RBF_0(R_i, \Omega) \right]$$

Thus, the exact schedulability test checks for each $\tau_i \in \tau'$, whether the response time $R_i$ is less or equal its relative deadline $d_i$. In the next two subsections, we derive response time lower bound $R_i^{lb}$ and upper bound $R_i^{ub}$ for each task $\tau_i \in \tau'$, and use these heuristics to derive an efficient iterative schedulability test following the suggestions of [Davis et al. 2008]. Now, given fixed resource period $\Pi$ and resource deadline $\Delta$ of $\Omega$, we use a binary search of $\Theta$ over the range $[0, \Pi]$ to solve our problem of obtaining minimum capacity $\Theta$ along with the above exact test.

### 4.1. Deriving Response Time Lower Bound

At any time $t > 0$, the response time lower bound $R_i^{lb}$ for $\tau_{-1}$ must be equal to $e_{-1}$, which follows the response time lower bound for next highest priority task $\tau_0$ to be at least $R_i^{lb} = e_{-1} + e_0$ after it is released at time $t = \Delta - \Theta$. All the tasks in $\tau$ have lower priority than these two tasks, and we have two cases while determining their response time lower bound. When $t < \Delta - \Theta$, $R_i^{lb}$ must be greater than $R_i^{lb}_0$ and when $t \geq \Delta - \Theta$, $R_i^{lb}$ must be greater $R_i^{lb}$, $\forall \tau_i \in \tau$.

From Equation 11, we obtain the following inequality (by dropping the ceilings) when $t \geq \Delta - \Theta$:

$$R_i \geq e_i + \sum_{\forall j \in hp(i)} \frac{R_j}{p_j} e_j + (\Delta - \Theta) + \frac{R_i - (\Delta - \Theta)}{\Pi} (\Pi - \Theta) \quad (13)$$

Let $U_{i-1} = \sum_{\forall j \in hp(i)} \frac{e_j}{p_j}$ (excluding $\tau_{-1}$ and $\tau_0$). Solving the above equation for $R_i$, we obtain response time lower bound $R_i^{lb}$.

$$R_i \geq e_i + \frac{\Theta}{\Pi} (\Delta - \Theta) - U_{i-1}. \quad (14)$$

### 4.2. Deriving Response Time Upper Bound

We derive the upper bound of response time for all $\tau_i \in \tau'$ using similar approach used by [Bini and Baruah 2007]. Let $w_i(t)$ represents maximum amount of time that the processor executes $\tau_i$ in any interval length $t$, and $w_i^0(t)$ represents maximum amount of time that the processor executes $\tau_i$ in any interval length $t$ when $\tau_i$ is the only task in the system (similar to [Bini and Baruah 2007]). Then the worst case workload at time $t$ for task $\tau_i$ is given by the following equation:

$$W_i(t) = \sum_{j=1}^{i} w_j(t) + w_0(t) + w_{-1}(t).$$

$$\leq \sum_{j=1}^{i} w_j^0(t) + w_0^0(t) + w_{-1}^0(t). \quad (15)$$

\(^1\)We slightly abuse notation in this section by denoting response time for tasks in $\tau'$ as $R_i$. 

ACM Transactions on Embedded Computing Systems, Vol. 1, No. 1, Article 1, Publication date: January YYYY.
Observe that for task $\tau_{-1}$, the maximum amount of time processor executes this task $w_{-1}^0(t)$ equals $\min\{\Delta - \Theta, t\}$ at time $t$. Therefore, we get the upper bound $w_{-1}^0(t) \leq \Delta - \Theta$.

For $\tau_0$, since it has release offset of $\Delta - \Theta$, $w_0^0(t) = \min\{t - (\Delta - \Theta) - (p_0 - e_0), \frac{t - (\Delta - \Theta)}{p_0}\} e_0$. The following equation gives an upper bound by linear approximation of the step function corresponding to $\tau_0$’s RBF.

$$w_0^0(t) \leq u_0 t + e_0 (1 - u_0) - (\Delta - \Theta) u_0.$$  \hfill (16)

For all other tasks in $\tau$, $w_i^0$ can be upper bounded by the similar linear approximation of RBF.

$$w_i^0(t) \leq w_i t + e_i (1 - u_i).$$  \hfill (17)

Now, from the workload upper bounds $w_i^s$ for all tasks in $\tau'$, and using the steps similar to Theorem 2 of [Bini and Baruah 2007], we obtain response time upper bound $R_i^{ub}$.

$$R_i \leq \left( e_i + \sum_{j=0}^{i-1} e_j (1 - u_j) + (1 - u_0)(\Delta - \Theta) \right) \left( 1 - \sum_{j=0}^{i-1} u_j \right)^{-1}. $$  \hfill (18)

For any task $\tau_i \in \tau$, if the right hand side of Equation 18 is greater than $d_i$, we use Equation 14 and 18 to obtain efficient initial value for the iterative algorithm similar to [Davis et al. 2008].

$$R_i = \max\{(d_i + e_i)/2, d_i - R_{i-1}^{ub}, R_i^{lb}\}$$  \hfill (19)

The response time lower bound and upper bound heuristics obtained in this section can be used to obtain an iterative approximate test, in which case instead of using Equation 12, we use Equation 5 in conjunction with the approximation of Equation 11 for a given approximation parameter $k$.

5. DETERMINING MINIMUM CAPACITY USING TESTING SET

5.1. Exact Test

The following theorem states the exact schedulability condition for EDP resource $\Omega$, where task system is scheduled using fixed-priority scheduling algorithm [Shin and Lee 2003; 2004; Easwaran et al. 2007]. It says that for the task system $\tau$ to be schedulable with EDP resource, each task $\tau_i$ in $\tau$ must have a fixed point $t$ before its deadline at which the cumulative request bound function for $\tau_i$ is less than the supply provided to the system at that point.

**Theorem 1 (from [Easwaran et al. 2007]).** A sporadic task system $\tau$ is fixed-priority schedulable upon an EDP resource $\Omega = (\Pi, \Theta, \Delta)$, if and only if,

$$\left( \forall i, \exists t \in (0, d_i) : R_i(t) \leq \text{sbf}(\Omega, t) \right) \land \left( U(\tau) \leq \frac{\Theta}{\Pi} \right)$$  \hfill (20)

While the above approach leads to an exact solution to the MIB-RT problem, it requires potentially pseudo-polynomial-time to return a solution. In the next section, we present a polynomial-time approximate algorithm to obtain minimum capacity for EDP resource when the component-level scheduling algorithm is fixed-priority for the task system $\tau$. We consider fixed period ($\Pi$) and deadline ($\Delta$) for the EDP resource $\Omega$. At a high-level, our approach uses the testing set point similar to Easwaran et al. [Easwaran et al. 2007]; however, to obtain polynomial-time complexity and an approximation ratio, we must fundamentally reformulate the problem to carefully determine which testing set point need to be considered.

5.2. Approximate Test
ALGORITHM 1: Pseudo-code for determining minimum capacity for a periodic resource given $\Pi$, $\Delta$, and $\tau$ using fixed-priority scheduling algorithm. Note the algorithm is exact when $k$ equals $\infty$.  

FPMINIMUMCAPACITY($\Pi$, $\Delta$, $\tau$, $k$)
1 $\Theta_{\text{min}}^{\text{ini}} \leftarrow U(\tau) \cdot \Pi$
2 for each $\tau_i \in \tau$
3 $\Theta_{\text{ini}}^{\text{min}} \leftarrow \infty$
4 for each $(t_a, t_{a+1}) \in \mathcal{S}_i(\tau, k) \triangleright$ (In order)
5 $D_{t_a} \leftarrow R_i(t_a) + r_i(t_a)$
6 $D_{t_{a+1}} \leftarrow R_i(t_{a+1})$
7 $\alpha \leftarrow \sum_{(t, \ell) \in t \geq d_i + (k-1)p_i} u_i$
8 $\Theta_{[t_1]} \leftarrow \Phi_1(L_{t_a}^i, [t_1] + 1, \Pi, \Delta)$
9 $\Theta_{[t_2]} \leftarrow \Phi_2(L_{t_a}^i, [t_2] - 1, \Pi, \Delta)$
10 if $[t_2] \leq [t_1]$
11 $\Theta_{[t_1]} \leftarrow \Phi_3(L_{t_a}^i, [t_1], \Pi, \Delta)$
12 $\Theta_{[t_2]} \leftarrow \Phi_3(L_{t_a}^i, [t_2], \Pi, \Delta)$
13 else
14 $\Theta_{[t_1]}, \Theta_{[t_2]} \leftarrow \infty$
15 $\Theta_{\text{min}}^{\text{ini}} \leftarrow \min\{\Theta_{[t_1] + 1}, \Theta_{[t_2] - 1}, \Theta_{[t_1]}, \Theta_{[t_2]}\}$
16 $\Theta_{\text{min}} \leftarrow \min\{\Theta_{\text{min}}^{\text{ini}}, \Theta_{\text{ini}}^{\text{min}}\}$
17 end (of inner loop)
18 $\Theta_{\text{min}} \leftarrow \max\{\Theta_{\text{min}}, \Theta_{\text{ini}}^{\text{min}}\}$
19 end (of outer loop)
20 return $\Theta_{\text{min}}$

In Algorithm 1, we present the pseudocode for our algorithm, FPMINIMUMCAPACITY. Given task system $\tau$ and an EDP resource with $\Pi$ and $\Delta$ as input, the algorithm returns approximate minimum capacity to correctly schedule the task system with the resource. The approximation parameter of the algorithm is the input $k \in \mathbb{N}^+$ ($k = \lfloor \frac{1}{\epsilon} \rfloor$). For some fixed $k$ input, the algorithm returns the approximate minimum capacity; if $k$ is equal to $\infty$, it returns exact minimum capacity. If FPMINIMUMCAPACITY returns a value $\Theta_{\text{min}}^{\text{ini}}$ that does not exceed $\Delta$, then $\tau$ can be fixed-priority scheduled to meet all deadlines upon $\Omega = (\Pi, \Delta, \Theta_{\text{min}}^{\text{ini}})$. Note that the approximate capacity $\Theta_{\text{min}}^{\text{ini}}$ can be at most $(1 + \epsilon)$ times the exact capacity. If FPMINIMUMCAPACITY returns a capacity greater than $\Delta$, then our algorithm cannot guarantee that $\tau$ can be scheduled on any $\Omega$ with parameters $\Pi$ and $\Delta$. (Unless $k = \infty$, the algorithm is an approximation, and, thus, a returned capacity greater than $\Delta$ does not necessarily imply infeasibility of $\tau$).

In our proposed algorithm, the objective is to compute minimum capacity $\Theta_{\text{min}}^{\text{ini}}$ for a task system $\tau$ such that $\tau$ is fixed-priority schedulable under EDP resource model. For each task $\tau_i \in \tau$, we find minimum capacity $\Theta_{\text{ini}}^{\text{min}}$ such that there exists a fixed point $t \in (0, d_i]$ at which the supply bound function $\text{sbf}$ exceeds the cumulative request bound function $\hat{R}_i(t)$ (Theorem 1). To calculate $\Theta_{\text{ini}}^{\text{min}}$, we determine, for each consecutive pair of values $(t_a, t_{a+1})$ in the testing set $\mathcal{S}_i(\tau, k)$, the minimum capacity $\Theta_{\text{min}}^{\text{ini}}$ required to guarantee that the line segment $L_{t_a}^i$ is beneath $\text{sbf}((\Pi, \Theta_{\text{min}}^{\text{ini}}, \Delta), t)$ for some $t \in (t_a, t_{a+1})$. Since $L_{t_a}^i$ is equivalent to $\hat{R}_i$ for all $t \in (t_a, t_{a+1})$, this implies that there exist $a$ $t \in (t_a, t_{a+1})$ such that $\hat{R}_i(t) \leq \text{sbf}((\Pi, \Theta_{\text{min}}^{\text{ini}}, \Delta), t)$. To determine $\Theta_{\text{ini}}^{\text{min}}$, we take specific steps of the $\text{sbf}$ (denote a selected step by $\ell$) and determine the minimum $\Theta_{\ell}$ such that some point of

$^2$We would like to thank the authors of [van den Heuvel et al. 2012] for identifying and correcting a small bug in our algorithm. In the original algorithm we did not check the condition in Line 10, and calculated $\Theta_{[t_1]}$ and $\Theta_{[t_2]}$ for all cases including the case $[t_2] > [t_1]$, which corresponds to an undefined interval (See Lemma 9, 10 and 11 for more details).
the line segment is below the $\ell$-feasibility region with capacity $\Theta_T$. Each $\Theta_T$ for $(t_{a}, t_{a+1})$ is set in lines 8, 9, 11 and 12. The following functions are used to determine the values of $\Theta_T$.

$$\Phi_1(L_{t_{a}}, \ell, \Pi, \Delta) \defeq \frac{D_{t_{a+1}} - t_{a+1} + \Pi + \Delta}{\ell + 1},$$

$$\Phi_2(L_{t_{a}}, \ell, \Pi, \Delta) \defeq \frac{D_{t_{a}}}{\ell},$$

$$\Phi_3(L_{t_{a}}, \ell, \Pi, \Delta) \defeq \frac{D_{t_{a}} + a(\Pi + \Delta - t_{a})}{\ell + a}.$$  

(21)

We show that we only need to consider the integer values of $\ell$ given by the following equations.

$$\ell_1 \defeq \frac{(t_{a+1} - \Delta) + \sqrt{(t_{a+1} - \Delta)^2 + 4\Pi D_{t_{a+1}}}}{2\Pi},$$

$$\ell_2 \defeq \frac{(t_{a} - \Delta) + \sqrt{(t_{a} - \Delta)^2 + 4\Pi D_{t_{a}}}}{2\Pi}.$$  

(22)

(23)

That is, we consider $[\ell_1], [\ell_1] + 1, [\ell_2] - 1$ and $[\ell_2]$ to evaluate $\Theta_T$. The logic behind the choice of $\Phi$ functions and our definition of $\ell_1$ and $\ell_2$ will be more apparent in the proof of correctness section.

Since we are looking for only one point in $t \in (0, d_t]$ for task $\tau_i$ where $\hat{R}_t(t) \leq \text{sbf}(\Omega, t)$, we only need a single line segment of $\hat{R}_t(t)$ that intersects with $\text{sbf}(\Omega, t)$ and gives minimum capacity. Thus, we set $\Theta_{\text{min}}^i$ to be the minimum of all $\Theta_{\text{min}}^i$ values for each of the line segment of $\hat{R}_t$. Finally, we set $\Theta_{\text{min}}^\tau$ to be the maximum of all $\Theta_{\text{min}}^i$ values. This ensures that for each task $\tau_i \in \tau$, we find a $t \leq d_t$ such that $\hat{R}_t(t) \leq \text{sbf}(\Pi, \Theta_{\text{min}}^\tau, \Delta, t)$. Since $\hat{R}_t(t) \geq R_t(t)$ for all $t$, this implies Theorem 1; thus $\tau$ is fixed-priority schedulable upon EDP resource $\Omega = (\Pi, \Theta_{\text{min}}^\tau, \Delta)$.

§Algorithm Complexity. The complexity of $\text{FPMinimumCapacity}$ depends on the number of tasks $n$ in the task set $\tau$ and the cardinality of testing set $\hat{S}_i(\tau, k)$ for each task $\tau_i$. The outer loop of the algorithm (Lines 2 to 19) iterates for each task, thus $n$ times in total. The inner loop (Lines 4 to 17) scans every pair of testing set points in $\hat{S}_i(\tau, k)$ (in non-decreasing order) for task $\tau_i$, and this can take at most $1 + (i - 1)(k - 1)$ times for a single task. Using a “heap-of-heaps” described by Mok [Mok 1988], the time complexity to obtain an element of the testing set is $O(\log n)$. Setting $D_{t_{a}}, D_{t_{a+1}}$ and $\alpha$ (Lines 5, 6 and 7) is done in constant time on each iteration of the inner loop. Again, setting $\ell$ values and evaluating $\Theta$ values using these (Line 8 to 12) takes constant time. Therefore, the runtime complexity of $\text{FPMinimumCapacity}$ is $O(\log n \cdot \sum_{i=1}^{n} [\hat{S}_i(\tau, k)])$. If $k = \infty$, the complexity for exactly determining the minimum capacity is the same complexity as the test of Theorem 1 on a fixed $\Omega$, which may be pseudo-polynomial depending on the period of tasks. Otherwise, if $k$ is a fixed integer, the complexity is at most $O(\log n \cdot \sum_{i=1}^{n} (1 + (i - 1)(k - 1)))$ times, which is $O(\log^2 n)$.

§Algorithm Correctness. To prove the correctness of $\text{FPMinimumCapacity}$, we prove the following theorem which states that the value returned by the algorithm (i.e., $\Theta_{\text{min}}$) is at least the optimal minimum capacity value $\Theta^\tau(\Pi, \Delta, \tau)$. Furthermore, if the input $k$ equals $\infty$, then the returned capacity is optimal.

**Theorem 2.** For all $k \in \mathbb{N}^+ \cup \{\infty\}$, $\text{FPMinimumCapacity}$ returns $\Theta_{\text{min}} \geq \Theta^\tau(\Pi, \Delta, \tau)$. Furthermore, if $k = \infty$, $\Theta_{\text{min}} = \Theta^\tau(\Pi, \Delta, \tau)$.

We require some additional definitions similar to [Fisher and Dewan 2012] for notational convenience. The next definition quantifies the minimum capacity $\Theta(\leq \Delta)$ that is required for $\text{sbf}$ to exceed the line segment $L_{t_{a}}$ at some point $(t, D_t)$. We use the convention that $\inf$ returns $\infty$ on an empty set.
Theorem 2. For the “only if” direction, we must show if some point of the line segment is in the feasibility region for any given \( \ell \), \( D = (t, D_i) \). We prove this by contrapositive; that is, if any of the three conditions is violated, the line segment will not have a point in the \( \ell \)-feasibility region.

**Lemma 3.** For any two consecutive pair of values \((t_a, t_{a+1}) \in \tilde{S}_i(\tau, k)\), there exists \((t, D_i) \in \mathcal{L}_i^{t_a}\) such that \((t, D_i) \in F(\Pi, \Delta, \Theta)\) for some \( \ell \in \mathbb{N}^+ \), if and only if, the following conditions hold:

\[
\Theta \geq \Phi_1(\mathcal{L}_i^{t_a}, \ell, \Pi, \Delta) \quad (26a) \\
\wedge \Theta \geq \Phi_2(\mathcal{L}_i^{t_a}, \ell, \Pi, \Delta) \quad (26b) \\
\wedge \Theta \geq \Phi_3(\mathcal{L}_i^{t_a}, \ell, \Pi, \Delta) \quad (26c)
\]

**Proof:** For the “only if” direction, we must show if some point of the line segment is in the \( \ell \)-feasibility region for any given \( \ell \) then the three conditions of Equation (26) hold. We prove this by contrapositive; that is, if any of the three conditions is violated, the line segment will not be in \( F(\Pi, \Delta, \Theta) \) for that \( \ell \). We now consider the negation of the conditions of Equation (26). By negation, at least one of the Equations (26a), (26b), or (26c) must be violated. We show that if any of the conditions is violated, then for all \((t, D_i) \in \mathcal{L}_i^{t_a}, (t, D_i) \not\in F(\Pi, \Delta, \Theta)\).

**Case 1:** Equation (26a) is false. That is,

\[
\Theta < \frac{D_{t_{a+1}} - t_{a+1} + \Pi + \Delta}{\ell + 1} \Rightarrow \Theta < \frac{\bar{D}_{t_a} + \alpha(t_{a+1} - t_a)}{\ell + 1}.
\]

The last inequality follows from the fact that \( D_{t_{a+1}} = \bar{D}_{t_a} + \alpha(t_{a+1} - t_a) \). For any \((t, D_i) \in \mathcal{L}_i^{t_a}\), let \( x \) be \( t - t_a \) where \( 0 \leq x \leq t_{a+1} - t_a \); thus, \( t = t_a + x \) and \( D_t = \bar{D}_{t_a} + \alpha x \). Consider the expression

\[
(\bar{D}_{t_a} + \alpha x) - (t_a + x) + \Pi + \Delta = \frac{D_{t_{a+1}} - t_{a+1} + \Pi + \Delta}{\ell + 1} \leq \frac{(\bar{D}_{t_a} + \alpha x) - (t_a + x) + \Pi + \Delta}{\ell + 1} \leq \frac{D_{t_{a+1}} - t_{a+1} + \Pi + \Delta}{\ell + 1} \text{ for all } (t, D_i) \in \mathcal{L}_i^{t_a}. \]

This implies that the first condition of \( \ell \)-feasibility is violated for all \((t, D_i)\).

**Case 2:** Equation (26b) is false. That is, \( \Theta < \bar{D}_{t_a}/\ell \). Again, consider any \((t, D_i) \in \mathcal{L}_i^{t_a}\). Observe that \( D_t = \bar{D}_{t_a} + \alpha(t - t_a) \geq \bar{D}_{t_a} \), since \( t \geq t_a \) and \( \alpha \geq 0 \). Thus, \( \Theta < \bar{D}_{t_a}/\ell \) implies \( \Theta < D_t/\ell \) for all \((t, D_i)\); this implies that the second condition of \( F(\Pi, \Delta, \Theta) \) is violated.

**Case 3:** Equation (26c) is false. That is,

\[
\Theta < \frac{\bar{D}_{t_a} + \alpha(t + \Pi - t_a)}{\ell + \alpha}.
\]
Consider any \((t, D_t) \in L_t^i\). We consider two further subcases based on the value of \(t\), and show that \((t, D_t) \notin F_t(\Pi, \Delta, \Theta)\) for both subcases.

**Subcase 3a:** \(t < \frac{D_t - a t_a + t + \Delta - (t + 1) \Theta}{1 - \alpha}\).

By solving for \(\Theta\), we obtain

\[
\Theta < \frac{D_t - a t_a + t + \Delta}{1 - \alpha} \Rightarrow \Theta < \frac{D_t - t + \Delta}{t + 1}.
\]

The implication follows from \(D_t = D_t + \alpha (t - t_a)\). The above inequality implies that the first condition of \(\ell\)-feasibility is violated.

**Subcase 3b:** \(t \geq \frac{D_t - a t_a + t + \Delta - (t + 1) \Theta}{1 - \alpha}\).

Again, solving for \(\Theta\),

\[
\Theta \geq \frac{D_t - a t_a + t + \Delta}{1 - \alpha} \Rightarrow \Theta \geq \frac{D_t - t + \Delta}{t + 1}.
\]

Now consider the value of the first partial derivative of \(\Phi_3\) with respect to \(\alpha\); i.e., \(\frac{\partial \Phi_3}{\partial \alpha}\)

\[
\frac{\ell (\Delta - t_a)}{(\ell + \alpha)^2} - \frac{t - t_a}{(\ell + 1)^2}\]

Note the sign of the above partial derivative is independent of the value of \(\alpha\); therefore, either \(\frac{\partial \Phi_3}{\partial \alpha} \leq 0\), or \(\frac{\partial \Phi_3}{\partial \alpha} > 0\) for any \(0 \leq \alpha \leq 1\); in other words, the sign remains constant for all \(\alpha\). If \(\frac{\partial \Phi_3}{\partial \alpha} > 0\), then \(\Phi_3\) is maximized when \(\alpha\) is as large as possible (i.e., \(\alpha\) equals one). By Equation (27), this implies that \(\Theta < \frac{D_t - t + \Delta - t_a}{t + 1}\) which is impossible due to Equation (28). Thus, \(\frac{\partial \Phi_3}{\partial \alpha} \leq 0\) must be true. If the partial derivative is non-positive, then \(\Phi_3\) is maximized when \(\alpha\) is as small as possible (i.e., \(\alpha\) equals zero). By Equation (27), \(\Theta < \frac{D_t}{\ell}\) which violates the second condition of \(\ell\)-feasibility.

Thus, we have proved that if the line segment has a point in the \(\ell\)-feasibility region, then the conditions in Equation (26) hold.

For the “if” direction, we need to show, if the conditions hold then there exists a point on the line segment that is included in the \(\ell\)-feasibility region. Again, we can show by contrapositive similar to the “only if” direction that if the line segment is strictly above the \(\ell\)-feasibility region, at least one of the three conditions is violated (See [Dewan and Fisher 2012a] for a detail proof).

The following lemma formalizes the equivalence between the concept of a line segment \(L_t^i\) being included in some \(\ell\)-feasibility region and the concept of a cumulative request-bound function \(\bar{R}_s\) falling below a supply-bound function \(sbf\).

**Lemma 4.** For consecutive pair of values \((t_a, t_a + 1) \in S_i(\tau, k)\) and \((t, D_t) \in L_t^i\) such that \(t_a < t \leq t_a + 1\), the inequality \(\bar{R}_s(t) \leq sbf((\Pi, \Theta, \Delta), t)\) holds, if and only if, there exists \(\ell \in \mathbb{N}\) such that \((t, D_t) \in F_\ell(\Pi, \Delta, \Theta)\).

**Proof:** For the “if” direction, we must show that if the point \((t, D_t) \in L_t^i\) satisfies \((t, D_t) \in F_\ell(\Pi, \Delta, \Theta)\), then there is sufficient supply over an interval of length \(t\) to satisfy the execution of a job of \(\tau_i\) and the approximated execution times of all higher-priority tasks (formally, \(\bar{R}_s(t) \leq sbf((\Pi, \Theta, \Delta), t)\)). Observe that every point in \(F_\ell(\Pi, \Delta, \Theta)\) is below the \(sbf\) function (see Figure 2). Thus, if \((t, D_t) \in F_\ell(\Pi, \Delta, \Theta)\), then \(D_t \leq sbf((\Pi, \Theta, \Delta), t)\). Finally, Lemma 2 states that \(\bar{R}_s(t) \leq D_t\) implying the “if” direction.

For the “only if” direction, observe that \(L_t^i\) and \(\bar{R}_s(t)\) are equivalent for \(t \in (t_a, t_a + 1)\). Thus, we must show that if line segment \(L_t^i\) has point \((t, D_t)\) contained below the \(sbf\) function for \(\Omega\),
then there exists an $\ell \in \mathbb{N}^+$ such that $(t, D_t, \alpha) \in \mathcal{F}_t(\Pi, \Theta, \Delta)$. Consider $\ell = \left\lceil \frac{D_t}{\Theta} \right\rceil$. The second condition of $\ell$-feasibility (Equation (10)) is trivially satisfied for this $\ell$. It also must be true that $D_t > (\ell - 1)\Theta$. Thus, $(t, D_t)$ must be below the line defined by $y = x - ((\ell + 1)\Theta)$ (otherwise, $(t, D_t)$ would be above the SBF function at $t$). This last constraint is equivalent to the first condition of $\ell$-feasibility region. Therefore, for $\ell = \left\lceil \frac{D_t}{\Theta} \right\rceil$ we have satisfied the two conditions of Equation (10), implying that $(t, D_t) \in \mathcal{F}_t(\Pi, \Theta, \Delta)$. \hfill \square

In the above lemma, we did not include $t_a$ in the interval of time values where line segment inclusion in the $\ell$-feasibility region implies that the approximate request-bound function is below the supply-bound function. The exclusion of $t_a$ from the above lemma is due to the fact that $\tilde{R}_i$ is discontinuous at $t_a$. However, notice that $t_a$ is the right end point of the predecessor line segment immediately to the left of $L_i$.

Lemma 3 equates the concept of finding $t$ such that $\tilde{R}_i(t)$ is below the SBF for a given $\Theta$ and the concept of point $(t, D_t)$ of a line segment $L_i$ being contained in some $\ell$-feasibility region for $\Theta$. The next lemma uses Definitions 4 and 5 to show that if we can compute $\Theta^*(\Pi, \Delta, L_i^*_{\ell})$ for any $\ell \in \mathbb{N}^+$, then we can also compute $\Theta^*(\Pi, \Delta, L_i^*)$.

**Lemma 5.**

$$\Theta^*(\Pi, \Delta, L_i^*) = \inf_{\ell \geq 0} \{ \Theta^*(\Pi, \Delta, L_i^*_{\ell}) \}. \quad (29)$$

**Proof:** Let $\Theta_{RHS}$ denote the right-hand side of Equation (29). We show that both $\Theta_{RHS} \geq \Theta^*(\Pi, \Delta, L_i^*)$ and $\Theta_{RHS} \leq \Theta^*(\Pi, \Delta, L_i^*)$. First, we show $\Theta_{RHS} \geq \Theta^*(\Pi, \Delta, L_i^*)$. By definition of infimum, for any $\delta > 0$, there exists $\ell \in \mathbb{N}^+$ such that $\Theta_{RHS} + \delta \geq \Theta^*$ for this $\ell$. Therefore, for all $\delta > 0$, $\Theta_{RHS} + \delta$ must be in the set considered in the inf on the right-hand side of Equation (24) (Definition 4). Thus, $\Theta_{RHS} \geq \Theta^*(\Pi, \Delta, L_i^*)$.

Next, we show $\Theta_{RHS} \leq \Theta^*(\Pi, \Delta, L_i^*)$. By Definition 4 and application of Lemma 2, there exist $(t, D_t) \in L_i$ such that $\tilde{R}_i(t) \leq \text{SBF}((\Pi, \Delta, \Theta^*(\Pi, \Delta, L_i^*))$, $t)$. Lemma 4 implies that there exists $\ell \in \mathbb{N}^+$ such that $(t, D_t) \in \mathcal{F}_t(\Pi, \Theta^*(\Pi, \Delta, L_i^*))$. This implies $\Theta^*(\Pi, \Delta, L_i^*)$ is in the set considered in the right-hand side of Equation (25) (Definition 5), implying the inequality. \hfill \square

In the next few lemmas, we derive the values $\ell_1$ and $\ell_2$ (Equations (22) and (23)), and prove that we only need to evaluate the $\Phi$ functions at these $\ell$ values to obtain minimum capacity. Consider the three conditions given in Equation (26) of Lemma 3. There are three possible cases. We invoke the reader to verify that these cases are complete and mutually exclusive.

**Case I:** $$(\Phi_1(L_i^*, \ell, \Pi, \Delta) > \Phi_2(L_i^*, \ell, \Pi, \Delta)) \land (\Phi_1(L_i^*, \ell, \Pi, \Delta) > \Phi_3(L_i^*, \ell, \Pi, \Delta));$$
**Case II:** $$(\Phi_2(L_i^*, \ell, \Pi, \Delta) > \Phi_3(L_i^*, \ell, \Pi, \Delta)) \land (\Phi_2(L_i^*, \ell, \Pi, \Delta) > \Phi_1(L_i^*, \ell, \Pi, \Delta));$$
**Case III:** $$(\Phi_3(L_i^*, \ell, \Pi, \Delta) > \Phi_1(L_i^*, \ell, \Pi, \Delta)) \land (\Phi_3(L_i^*, \ell, \Pi, \Delta) > \Phi_2(L_i^*, \ell, \Pi, \Delta)).$$

For each of the above cases, we solve for the value of $\ell$ and obtain bounds for the value of $\ell$.

**Lemma 6.** For any $L_i$, and $\ell \in \mathbb{N}^+$, $\Pi$, $\Delta$, Case I holds, if and only if,

$$\ell \geq \frac{(t_{a+1} - \Delta) + \sqrt{(t_{a+1} - \Delta)^2 + 4\Pi D_{t_{a+1}}}}{2\Pi} + 1. \quad (30)$$

**Proof:** Let us consider the “only if” direction of the lemma; that is, Case I holds. From Case I, we have that both $\Phi_1(L_i^*, \ell, \Pi, \Delta) > \Phi_2(L_i^*, \ell, \Pi, \Delta)$ and $\Phi_1(L_i^*, \ell, \Pi, \Delta) > \Phi_3(L_i^*, \ell, \Pi, \Delta)$. For $\Phi_1(L_i^*, \ell, \Pi, \Delta) > \Phi_2(L_i^*, \ell, \Pi, \Delta)$, solving for $\ell$, ...
if direction follows by simply reversing the direction of each implication in the proof.

The bidirectional implication follows since Inequality (31) is a quadratic inequality with respect to \( \ell \), defining a convex parabola \( \Pi \ell^2 - (t_{a+1} + \alpha t_a - \Delta) \ell - \bar{D}_{t_a} \). The zeros are

\[
\ell = \frac{t_{a+1} - \alpha t_a - \Delta \pm \sqrt{(1-\alpha)t_{a+1} + \alpha t_a - \Delta}^2 + 4\Pi\bar{D}_{t_a}}{2\Pi}
\]

The bidirectional implication follows since Inequality (31) is a quadratic inequality with respect to \( \ell \), defining a convex parabola \( \Pi \ell^2 - ((1-\alpha)t_{a+1} + \alpha t_a - \Delta) \ell - D_{t_a} \). The zeros are

\[
\ell > \frac{t_{a+1} - \alpha t_a - \Delta \pm \sqrt{(1-\alpha)t_{a+1} + \alpha t_a - \Delta}^2 + 4\Pi D_{t_a}}{2\Pi}
\]

Since the square-root term in the numerator is always greater than the term preceding the \( \pm \), one root is positive and the other is negative. Inequality (31) implies that we are interested in values of \( \ell \in \mathbb{N}^+ \) such that the parabola strictly exceeds zero. Since the parabola is convex, all values of \( \ell \) strictly greater than the positive root satisfy this inequality.

For \( \Phi_1(L_{t_a}, \ell, \Pi, \Delta) > \Phi_3(L_{t_a}, \ell, \Pi, \Delta) \), solving for \( \ell \),

\[
\ell > \frac{(t_{a+1} - \alpha t_a - \Delta) \pm \sqrt{(1-\alpha)t_{a+1} + \alpha t_a - \Delta}^2 + 4\Pi D_{t_a+1}}{2\Pi}
\]

The bidirectional implication follows since Inequality (32) is a quadratic inequality with respect to \( \ell \), defining a convex parabola \( \Pi \ell^2 - ((t_{a+1} - \Delta) \ell - ((D_{t_a} + \alpha(t_{a+1} - t_a))) \). By similar reasoning done for Inequality (31), all values of \( \ell \) strictly greater than the positive root satisfy this inequality.

Combining Equations (31) and (32), we obtain

\[
\ell > \max \left\{ \frac{(1-\alpha)t_{a+1} + \alpha t_a - \Delta \pm \sqrt{(1-\alpha)t_{a+1} + \alpha t_a - \Delta}^2 + 4\Pi \bar{D}_{t_a}}{2\Pi} \right\}
\]

Observe that \( (1-\alpha)t_{a+1} + \alpha t_a - \Delta \) equals \( t_{a+1} - \Delta - \alpha(t_{a+1} - t_a) \) which is at most \( t_{a+1} - \Delta \), since \( t_{a+1} > t_a \) and \( 0 \leq \alpha \leq 1 \). Thus, we conclude that the second value of Equation (33) is the maximum of the two bounds obtained. The lemma follows by observing that \( \ell \) is an integer. The “if” direction follows by simply reversing the direction of each implication in the proof.

Similarly, we can prove the next two lemmas and bound the values of \( \ell \) for Case II and III. Due to space constraint the proofs are omitted and can be found in [Dewan and Fisher 2012a].

**Lemma 7.** For any \( L_{t_a} \) and \( \ell \in \mathbb{N}^+, \Pi, \Delta \), Case II holds, if and only if,

\[
\ell \leq \Delta + \sqrt{\frac{(t_a - \Delta)^2 + 4\Pi D_{t_a}}{2\Pi}} - 1.
\]

**Lemma 8.** For any \( L_{t_a} \) and \( \ell \in \mathbb{N}^+, \Pi, \Delta \), Case III holds, if and only if,

\[
\ell \leq \Delta + \sqrt{\frac{(t_{a+1} - \Delta)^2 + 4\Pi \bar{D}_{t_a+1}}{2\Pi}}.
\]

We now prove three lemmas and corollaries which show that for all \( \ell \in \mathbb{N}^+ \) not equal to the values \( \lfloor \ell_1 \rfloor, \lfloor \ell_1 \rfloor + 1, \lfloor \ell_2 \rfloor \) or \( \lfloor \ell_2 \rfloor - 1 \) will result in a larger minimum \( \Theta \). The first lemma, towards this goal, shows that if a point on the line segment is in an \( \ell^* \)-feasibility region and \( \ell^* \) is at least \( \lfloor \ell_1 \rfloor + 1 \), then the point is also in the \( \lfloor \ell_1 \rfloor + 1 \)-feasibility region.
Lemma 9. For any \( t_a, t_{a+1} \in \tilde{S}_i(\tau, k) \), \( t, D_i \) \( L^i_{t_a} \), \( \ell' \in \mathbb{N}^+ \), \( \Pi, \Delta \), and \( \Theta \), if \( \ell' \geq [\ell_1] + 1 \) and \( \Theta \leq \Delta \) then

\[
[(t, D_i) \in \mathcal{F}_r(\Pi, \Delta, \Theta)] \Rightarrow [(t, D_i) \in \mathcal{F}_{[\ell_1]+1}(\Pi, \Delta, \Theta)].
\]

Proof: By Lemma 6 and \( \ell' \geq [\ell_1] + 1 \), Case I must hold for all such \( \ell' \). Combining Case I and Lemma 3, we have that if \( (t, D_i) \in \mathcal{F}_r(\Pi, \Delta, \Theta) \), then

\[
\Theta \geq \Phi_1(L^i_{t_a}, \ell', \Pi, \Delta).
\]

Now consider the first partial derivative of \( \Phi_1 \) with respect to \( \ell' \); i.e.,

\[
\frac{\partial \Phi_1}{\partial \ell'} = -\frac{D_{t_{a+1}} + D_{t_{a+1}} - \ell(\ell + 1)\Theta}{(\ell + 1)^2} = \frac{[\ell_{a+1} - D_{t_{a+1}}] + [\Pi - \Delta]}{(\ell + 1)^2}.
\]

Since \( \Pi \geq \Delta \), the second term in the numerator is positive. Consider the first term, \( t_{a+1} - D_{t_{a+1}} \).

By \( (t, D_i) \in \mathcal{F}_r(\Pi, \Delta, \Theta) \) and the first condition of \( \ell' \)-feasibility,

\[
t \geq D_i + \ell \Pi + \Delta - (\ell + 1)\Theta
\]

\[
\Rightarrow t + (t_{a+1} - t) \geq D_i + \alpha(t_{a+1} - t) + \ell \Pi + \Delta - (\ell + 1)\Theta
\]

(since \( \alpha < 1 \))

\[
\Rightarrow t_{a+1} \geq D_{t_{a+1}} + \ell \Pi + \Delta - (\ell + 1)\Theta
\]

\[
\Rightarrow t_{a+1} \geq D_{t_{a+1}}.
\]

The second to last implication is due to \( D_i = \bar{D}_{t_a} + \alpha(t - t_a) \) and \( D_{t_{a+1}} = \bar{D}_{t_a} + \alpha(t_{a+1} - t_a) \). The last implication is due to \( \Theta \leq \Delta \). Therefore, the first term in the numerator of \( \frac{\partial \Phi_1}{\partial \ell'} \) is also positive. Thus, \( \frac{\partial \Phi_1}{\partial \ell'} \) is non-decreasing for all \( \ell' \). Thus, the \( \Phi_1 \) evaluated at \( [\ell_1] + 1 \) is a lower bound; i.e., for all \( \ell' \geq [\ell_1] + 1 \),

\[
\Phi_1(L^i_{t_a}, \ell', \Pi, \Delta, \Theta) \geq \Phi_1(L^i_{t_a}, [\ell_1] + 1, 1, \Pi, \Delta).
\]

The above inequality implies that \( \Theta \geq \Phi_1(L^i_{t_a}, [\ell_1] + 1, 1, \Pi, \Delta) \), satisfying Equation (26a) of Lemma 3. For \( [\ell_1] + 1 \), Case I holds, implying that Equations (26b) and (26c) must also hold. Thus, by Lemma 3, \( (t, D_i) \in \mathcal{F}_{[\ell_1]+1}(\Pi, \Delta, \Theta) \).

The next corollary follows from the above lemma and the definition of \( \Theta^*_l \) (Definition 5).

Corollary 1. For any \( t_a, t_{a+1} \in \tilde{S}_i(\tau, k) \), \( \ell' \in \mathbb{N}^+ \), \( \Pi \), and \( \Delta \), if \( \ell' \geq [\ell_1] + 1 \) then

\[
\Theta^*_l(P, \Delta, L^i_{t_a}) \geq \Theta^*_l([\ell_1]+1)(\Pi, \Delta, L^i_{t_a}).
\]

The next lemma shows that if a point on the line segment is in an \( \ell' \)-feasible region and \( \ell' \) is at most \( [\ell_2] - 1 \), then the point is also in the \([\ell_2] - 1 \)-feasible region. Again we omit the proof since it can be derived in a similar way of Lemma 10.

Lemma 10. For any \( t_a, t_{a+1} \in \tilde{S}_i(\tau, k) \), \( t, D_i \) \( L^i_{t_a} \), \( \ell' \in \mathbb{N}^+ \), \( \Pi, \Delta \), and \( \Theta \), if \( \ell' \leq [\ell_2] - 1 \) and \( \Theta \leq \Delta \) then

\[
[(t, D_i) \in \mathcal{F}_r(\Pi, \Delta, \Theta)] \Rightarrow [(t, D_i) \in \mathcal{F}_{[\ell_2]-1}(\Pi, \Delta, \Theta)] .
\]

The next corollary follows from the above lemma and the definition of \( \Theta^*_l \) (Definition 5).

Corollary 2. For any \( t_a, t_{a+1} \in \tilde{S}_i(\tau, k) \), \( \ell' \in \mathbb{N}^+ \), \( \Pi \), and \( \Delta \), if \( \ell' \leq [\ell_2] - 1 \) then

\[
\Theta^*_l(P, \Delta, L^i_{t_a}) \geq \Theta^*_l([\ell_2]-1)(\Pi, \Delta, L^i_{t_a}).
\]

Lemma 11. For any \( t_a, t_{a+1} \in \tilde{S}_i(\tau, k) \), \( t, D_i \) \( L^i_{t_a} \), \( \ell' \in \mathbb{N}^+ \), \( \Pi, \Delta \), and \( \Theta \), if \( \ell_2 \leq \ell' \leq [\ell_1] \) then

\[
[(t, D_i) \in \mathcal{F}_r(\Pi, \Delta, \Theta)] \Rightarrow [(t, D_i) \in \mathcal{F}_{[\ell_1]}(\Pi, \Delta, \Theta)] \lor [(t, D_i) \in \mathcal{F}_{[\ell_2]}(\Pi, \Delta, \Theta)] .
\]

ACM Transactions on Embedded Computing Systems, Vol. 1, No. 1, Article 1, Publication date: January YYYY.
The next corollary follows from the above lemma and the definition of $\Theta^*_t$ (Definition 5).

**Corollary 3.** For any $t_a, t_{a+1} \in \hat{S}_i(t, k)$, $\ell' \in \mathbb{N}^+$, $\Pi$, and $\Delta$, if $[\ell_2] \leq \ell' \leq [\ell_1]$ then

$$\Theta^*_t(\Pi, \Delta, L^i_{t_a}) \leq \min\{\Theta^*_t(\Pi, \Delta, L^i_{t_a}), \Theta^*_t(\Pi, \Delta, L^i_{t_a})\}.$$  

Combining Corollaries 1, 2, 3, and using Definitions 4 and 5, we obtain the following corollary.

**Corollary 4.**

$$\Theta^*(\Pi, \Delta, L^i_{t_a}) = \min_{\ell \in \{[\ell_1], [\ell_1] + 1, [\ell_2], [\ell_2] - 1\}} \{\Theta^*_t(\Pi, \Delta, L^i_{t_a})\}.$$  

**Proof:** By Lemma 5, we may determine $\Theta^*(\Pi, \Delta, L^i_{t_a})$ by evaluating $\Theta^*_t(\Pi, \Delta, L^i_{t_a})$ for all possible $\ell \in \mathbb{N}^+$. The corollary follows by applying Corollaries 1, 2, and 3, respectively, for the following regions of $\ell$: $[1, [\ell_2] - 1], [\ell_2], [\ell_1]$, and $[\ell_1 + 1, \infty)$. 

By the above corollary, we now know how to compute $\Theta^*(\cdot)$ efficiently from $\Theta^*_t(\cdot)$. The next lemma shows that we may use the $\Phi$ functions to efficiently compute $\Theta^*_t(\cdot)$.

**Lemma 12.** For any $L^i_{t_a}$ and $\ell \in \mathbb{N}^+$,

$$\Theta^*_t(\Pi, \Delta, L^i_{t_a}) = \begin{cases} \Phi_1(L^i_{t_a}, \ell, \Pi, \Delta), & \text{if } \ell \geq [\ell_1] + 1; \\ \Phi_2(L^i_{t_a}, \ell, \Pi, \Delta), & \text{if } \ell \leq [\ell_2] - 1; \\ \Phi_3(L^i_{t_a}, \ell, \Pi, \Delta), & \text{otherwise.} \end{cases} \quad (36)$$  

**Proof:** From Definition 5, $\Theta^*_t(\Pi, \Delta, L^i_{t_a})$ is the minimum $\Theta \leq \Delta$ such that there exists $(t, D) \in L^i_{t_a}$, $(t, D) \in F_\ell(\Pi, \Delta, \Theta)$. By Lemma 3, such a $\Theta$ is also the minimum value that satisfied all three conditions of Equation (26). Since each of the conditions is a lower bound on $\Theta$ (with equality permitted), $\Theta$ must satisfy equality of at least one of the three conditions of Equation (26) and must exceed or equal the other two conditions. Notice that, if $\ell \geq [\ell_1] + 1$, then by Lemma 6, $\Theta$ equals $\Phi_1(L^i_{t_a}, \ell, \Pi, \Delta)$. We can show an identical proof for intervals $(0, [\ell_2] - 1]$ and $[\ell_2], [\ell_1]$, by applying Lemmas 7 and 8, respectively. 

The final lemma that we prove before providing a proof for Theorem 2 shows that a choice of $\Theta$ based on the computation of $\Theta^*(\cdot)$ is a “safe” choice in the sense that all tasks in $\tau_i$ will complete by their deadline under an EDP resource $\Omega = (\Pi, \Theta, \Delta)$.

**Lemma 13.** For all $\tau_i \in \tau$, $\exists t \in (0, d_i]$ such that $\hat{R}_i(t) \leq \text{sbf}((\Pi, \Theta, \Delta), t)$ and $U(\tau) \leq \frac{2}{n}$, if and only if,

$$\Theta \geq \max \left( \frac{\max_{\tau_i \in \tau} \left\{ \min_{t_{a}, t_{a+1} \in \hat{S}_i(t, k)} \{\Theta^*(\Pi, \Delta, L^i_{t_a})\} \right\}}{\Pi} \right). \quad (37)$$  

**Proof:** We will prove this lemma by contrapositive. For the “if” direction, we must prove if either $U(\tau) > \frac{2}{n}$ or $\forall t \in (0, d_i] : \hat{R}_i(t) > \text{sbf}((\Pi, \Theta, \Delta), t)$, then the negation of the inequality of Equation (37) is true. If we consider $U(\tau) > \frac{2}{n}$, the inequality of Equation (37) is trivially violated due to the second expression in the outer max of Equation (37).

Now, consider the case when there exists a $\tau_i \in \tau$ such that $\hat{R}_i(t) > \text{sbf}((\Pi, \Theta, \Delta), t)$ for all $t$ in $(0, d_i]$. By Lemma 4, this implies for all $t \in \mathbb{N}^+$, $t_{a}, t_{a+1} \in \hat{S}_i(t, k)$, and $(t, D_t) \in L^i_{t_a}$, that $(t, D_t) \notin F_\ell(\Pi, \Delta, \Theta)$. By Definition 5, it must be for all $t \in \mathbb{N}^+$ that $\Theta < \Theta^*_t(\Pi, \Delta, L^i_{t_a})$. By Lemma 5, this implies that $\Theta < \Theta^*(\Pi, \Delta, L^i_{t_a})$ for any $t_{a}, t_{a+1} \in \hat{S}_i(t, k)$, which violates the inequality of Equation (37) due to the first term in the outer max. For the “only if” direction of the lemma, we will also consider the contrapositive. The contrapositive will follow by simply reversing the implications of the proof for the “if” direction. 

After proving the above conditions, we are ready to prove Theorem 2 which states that FPMINIMUMCAPACITY returns a valid value for finite $k$ and an exact value for $k = \infty$. 

ACM Transactions on Embedded Computing Systems, Vol. 1, No. 1, Article 1, Publication date: January YYYY.
\textbf{Proof of Theorem 2} We show that $\Theta_{\text{min}}$ returned from FP\textsc{MinimumCapacity} corresponds to the value on the right-hand side of Equation (37) of Lemma 13. The loop from Line 4 to 17 iterates through each consecutive pair of values $t_a$ and $t_{a+1}$ in $\hat{S}_i(\tau, k)$ to find optimal capacity for each line segment defined by the endpoints $(t_a, D_{t_a})$ and $(t_{a+1}, D_{t_{a+1}})$. It sets variables corresponding to $\hat{R}_i(t_a)$ and $\hat{R}_i(t_{a+1})$ in Lines 5 and 6 respectively. Then, in the next few lines it sets four different values to $\ell$ (based on $\ell_1$ and $\ell_2$) defined in Equations (22) and (23)) and evaluates $\Phi_1(\cdot)$ according to Lemma 12 to compute $\Theta^*_\ell(\cdot)$ for each of the four integer values of $\ell$. Therefore, $\Theta_{t_{a}}^\text{min}$, set in Line 15, equals $\Theta^*(\Pi, \Delta, L_i^*_{t_a})$ by Lemma 5. At the end of this loop it sets $\Theta_{i}^\text{min}$ to be the minimum of $\Theta_{t_{a}}^\text{min}$ and $\Theta_{t_{a+1}}^\text{min}$ (Line 16). Thus, once the inner loop is executed for all $t_a, t_{a+1} \in \hat{S}_i(\tau, k)$, $\Theta_i^\text{min}$ contains the minimum of all $\Theta_{t_{a}}^\text{min}$ values. The outer loop from Line 2 to Line 19 finds $\Theta_i^\text{min}$ for all task $\tau_i$ in $\tau$. Finally, in Line 18, $\Theta_i^\text{min}$ is set to the maximum of $U(\tau) \cdot \Pi$ and $\Theta_i^\text{min}$ for all $\tau_i \in \tau$.

By Lemma 13, $\hat{R}_i(t) \leq \text{sbf}((\Pi, \Theta_{\text{min}}, \Delta), t)$ for some $t \in (0, d_i]$ and $U(\tau) \leq \frac{1}{\Pi}$. By Lemma 1, $R_i(t) \leq \hat{R}_i(t)$. This implies $R_i(t) \leq \text{sbf}((\Pi, \Theta_{\text{min}}, \Delta), t)$ which is the schedulability condition given by Theorem 1. Therefore, $\tau$ will always meet all deadlines when scheduled by fixed-priority scheduling upon $\Omega = (\Pi, \Theta_{\text{min}}, \Delta)$. When $k = \infty$, $\hat{R}_i(t)$ equals $R_i(t)$ for all $t \geq 0$; in this case, $\Theta_{\text{min}}$ equals $\Theta^*(\Pi, \Delta, \tau)$ (i.e., $\Theta_{\text{min}}$ is exact capacity). \qed

\section{6. AN APPROXIMATION SCHEME}

In the previous section, we have shown that FP\textsc{MinimumCapacity} gives a valid answer when $k$ is finite and an exact answer when $k$ is infinite. In this section, we show that as $k$ increases, the guaranteed accuracy of FP\textsc{MinimumCapacity} increases along with its running time. Theorem 3 presents the tradeoff between accuracy and computational complexity, in terms of $k$.

\textsc{Theorem 3.} Given $\Pi, \Delta, \tau$, and $k \in \mathbb{N}^+$, the procedure FP\textsc{MinimumCapacity} returns $\Theta_{\text{min}}$ such that

$$\Theta^*(\Pi, \Delta, \tau) \leq \Theta_{\text{min}} \leq \left(\frac{k + 1}{k}\right) \cdot \Theta^*(\Pi, \Delta, \tau).$$

Furthermore, FP\textsc{MinimumCapacity} $(\Pi, \Delta, \tau, k)$ has time complexity $O(kn^2 \log n)$

The following corollary quantifying our FPTAS is immediately obtainable from Theorem 3, by substituting a value for $k$ dependent on the accuracy parameter $\epsilon$ ($k = \lceil \frac{1}{\epsilon} \rceil$).

\textsc{Corollary 5.} Given $\Pi, \Delta, \tau$, and $\epsilon > 0$, the procedure FP\textsc{MinimumCapacity} $(\Pi, \Delta, \tau, \lceil \frac{1}{\epsilon} \rceil)$ returns $\Theta_{\text{min}}$ such that

$$\Theta^*(\Pi, \Delta, \tau) \leq \Theta_{\text{min}} \leq (1 + \epsilon) \cdot \Theta^*(\Pi, \Delta, \tau).$$

Furthermore, FP\textsc{MinimumCapacity} $(\Pi, \Delta, \tau, \lceil \frac{1}{\epsilon} \rceil)$ has time complexity $O\left(\frac{n^2 \log n}{\epsilon}\right)$.

To prove Theorem 3, we need to prove two additional lemmas.

\textsc{Lemma 14.} Given $\Pi, \Delta$, and consecutive pair of values $t_a, t_{a+1} \in \hat{S}_i(\tau, k)$, the following is true for all $k, \ell(\in \mathbb{N}^+)$, and $\alpha(\in [0, 1])$,

$$\Theta^*_\ell(\Pi, \Delta, L_i^*) \leq \left(\frac{k + 1}{k}\right) \cdot \Theta^*_\ell\left(\Pi, \Delta, \left\langle \left(t_a, \frac{k \cdot D_{t_a}}{k + 1}\right), \left(t_{a+1}, \frac{k \cdot D_{t_{a+1}}}{k + 1}\right) \right\rangle, \frac{k \cdot \alpha}{k + 1}\right).$$

\textbf{Proof:} By Lemma 12, $\Theta^*_\ell(\Pi, \Delta, L_i^*)$ must be equal to one of $\Phi_1$, $\Phi_2$ or $\Phi_3$ according to the value of $\ell$. We show that for each of the three possibilities, Equation (38) must hold.

ACM Transactions on Embedded Computing Systems, Vol. 1, No. 1, Article 1, Publication date: January YYYY.
If $\Theta^*_\ell(\Pi, \Delta, L^i_{t_a})$ is equal to $\Phi_3(L^i_{t_a}, \ell, \Pi, \Delta)$ (i.e., $\frac{D_{t_{a+1}} - t_{a+1} + t\Pi + \Delta}{t_{a+1}}$), then $\ell \geq [\ell_1] + 1$ by Lemma 12. This implies by definition of $\ell_1$,

$$\ell \geq \left\lfloor \frac{(t_{a+1} - \Delta) + \sqrt{(t_{a+1} - \Delta)^2 + 4t\Pi D_{t_{a+1}}}}{2\Pi} \right\rfloor + 1$$

$$> \frac{(t_{a+1} - \Delta) + \sqrt{(t_{a+1} - \Delta)^2 + 4t\Pi D_{t_{a+1}}}}{2\Pi}$$

$$\geq \frac{2(t_{a+1} - \Delta)}{2\Pi} = \frac{t_{a+1} - \Delta}{\Pi}.$$  

Thus, $\ell \Pi + \Delta - t_{a+1} \geq 0$. By Lemma 12 and $\ell \geq [\ell_1] + 1$,

$$\Theta^*_\ell(\Pi, \Delta) \left( (t_{a}, k \cdot \frac{D_{t_{a+1}}}{k + 1}), (t_{a+1}, k \cdot \frac{D_{t_{a+1}}}{k + 1}), k \cdot \frac{\alpha}{k + 1} \right)$$

$$= \Phi_1 \left( (t_{a}, k \cdot \frac{D_{t_{a+1}}}{k + 1}), (t_{a+1}, k \cdot \frac{D_{t_{a+1}}}{k + 1}), k \cdot \frac{\alpha}{k + 1}, \ell, \Pi, \Delta \right)$$

$$= \frac{k \cdot D_{t_{a+1}} - t_{a+1} + \ell \Pi}{\ell_{a+1}} \cdot \Theta^*_\ell(\Pi, \Delta, L^i_{t_a}).$$

In this case, Equation (38) holds.

If $\Theta^*_\ell(\Pi, \Delta, L^i_{t_a})$ is equal to $\Phi_2(L^i_{t_a}, \ell, \Pi, \Delta)$ (i.e., $\frac{D_{t_{a+1}}}{t_{a+1}}$), then $\ell \leq [\ell_2] - 1$ by Lemma 12. Lemma 12 also implies

$$\Theta^*_\ell(\Pi, \Delta) \left( (t_{a}, k \cdot \frac{D_{t_{a+1}}}{k + 1}), (t_{a+1}, k \cdot \frac{D_{t_{a+1}}}{k + 1}), k \cdot \frac{\alpha}{k + 1} \right)$$

$$= \Phi_2 \left( (t_{a}, k \cdot \frac{D_{t_{a+1}}}{k + 1}), (t_{a+1}, k \cdot \frac{D_{t_{a+1}}}{k + 1}), k \cdot \frac{\alpha}{k + 1}, \ell, \Pi, \Delta \right)$$

$$\geq \frac{k \cdot D_{t_{a+1}}}{\ell_{a+1}} \cdot \Theta^*_\ell(\Pi, \Delta, L^i_{t_a}).$$

Finally, if $\Theta^*_\ell(\Pi, \Delta, L^i_{t_a})$ is equal to $\Phi_3(L^i_{t_a}, \ell, \Pi, \Delta)$ (i.e., $\frac{D_{t_{a+1}} + \alpha(t\Pi + t\Delta - t_{a+1})}{t_{a+1}}$), then $[\ell_2] \leq \ell \leq [\ell_1]$ by Lemma 12. Lemma 12 also implies that

$$\Theta^*_\ell(\Pi, \Delta) \left( (t_{a}, k \cdot \frac{D_{t_{a+1}}}{k + 1}), (t_{a+1}, k \cdot \frac{D_{t_{a+1}}}{k + 1}), k \cdot \frac{\alpha}{k + 1} \right)$$

$$= \Phi_3 \left( (t_{a}, k \cdot \frac{D_{t_{a+1}}}{k + 1}), (t_{a+1}, k \cdot \frac{D_{t_{a+1}}}{k + 1}), k \cdot \frac{\alpha}{k + 1}, \ell, \Pi, \Delta \right)$$

$$\geq \frac{k \cdot D_{t_{a+1}}}{\ell_{a+1}} \cdot \Theta^*_\ell(\Pi, \Delta, L^i_{t_a}).$$

$\square$

**Lemma 15.** Given $\Pi, \Delta, \tau_1 \in \tau$, and $k \in \mathbb{N}^+$, there exists consecutive pair of values $t_a, t_{a+1} \in \hat{S}_i(\tau, k)$ such that,

$$\Theta^*(\Pi, \Delta, \tau) \geq \Theta^*(\Pi, \Delta, \left( (t_{a}, k \cdot \frac{D_{t_{a+1}}}{k + 1}), (t_{a+1}, k \cdot \frac{D_{t_{a+1}}}{k + 1}), k \cdot \frac{\alpha}{k + 1} \right)).$$  

(39)
Proof: Let $\Theta_{\text{RHS}}$ denote the right-hand side of Equation (39). By definition of $\Theta^*(\Pi, \Delta, \tau)$ and Theorem 1, for all $t \in \tau$, there exist $t \in (0, d_i]$ such that
\[
R_i(t) \leq \text{sbf}((\Pi, \Theta^*(\Pi, \Delta, \tau), \Delta), t).
\]  
Now consider any pair of consecutive values $t_a, t_{a+1} \in \hat{S}_i(\tau, k)$. By Lemma 1, we have, for all $t \in (t_a, t_{a+1}]$,
\[
(k+1)^{t_i}(R_i(t)) = \left(\frac{k+1}{k}\right) \left(\frac{\sum_{j=1}^{i-1} \text{RBF}(\tau_j, t)}{\sum_{j=1}^{i-1} \text{RBF}(\tau_j, t)}\right) 
\geq \frac{e_i + \sum_{j=1}^{i-1} \text{RBF}(\tau_j, t)}{k}\frac{\sum_{j=1}^{i-1} \delta(\tau_i, t) \cdot \frac{k}{\alpha}} {k+1}
= \hat{R}_i(t)
\]  
Combining the inequalities of Equations (40) and (41) gives us, for all $t \in (t_a, t_{a+1}]$,
\[
\text{sbf}((\Pi, \Theta^*(\Pi, \Delta, \tau), \Delta), t) \geq \frac{k}{k+1} \cdot \hat{R}_i(t).
\]  
Lemma 4 and Equation (42) imply that there exists $t \in \mathbb{N}$ and $(t, D_t) \in \left\{(t, \frac{k \cdot D_t}{k+1}), (t_{a+1}, \frac{k \cdot D_{a+1}}{k+1}) \right\}$ such that $(t, D_t) \in \mathcal{F}_t(\Pi, \Theta^*(\Pi, \Delta, \tau), \Delta)$. The above expression and Definition 5 implies
\[
\Theta^*(\Pi, \Delta, \left\{(t, \frac{k \cdot D_t}{k+1}), (t_{a+1}, \frac{k \cdot D_{a+1}}{k+1}) \right\}) \leq \Theta^*(\Pi, \Delta, \tau).
\]  
The lemma follows from the expression above and Lemma 5. \hfill \Box

Corollary 6. Given $\Pi, \Delta, k \in \mathbb{N}^+$, and $\tau_i$, there exists consecutive pair of values $t_a, t_{a+1} \in \hat{S}_i(\tau, k)$,
\[
\left(\frac{k+1}{k}\right) \cdot \Theta^*(\Pi, \Delta, \tau) \geq \inf_{t \in \mathbb{N}^+} \left\{ \Theta^*(\Pi, \Delta, \tau) \right\}.
\]  
Now, we are ready to give the proof of Theorem 3.

Proof of Theorem 3 We already proved the first part in Theorem 2; now we must prove the second part of the inequality. From our algorithm, the value of $\Theta^\text{min}$ can be either equal to $\Pi \cdot U(\tau)$ or greater than this term. If $\Theta^\text{min} = \Pi \cdot U(\tau)$, Theorem 1 implies that $\Theta^*(\Pi, \Delta, \tau)$ must be at least $U(\tau) \cdot \Pi$. For this case, the second inequality follows, since $\frac{k+1}{k+1} \geq 1$ for all $k \in \mathbb{N}^+$. Now consider the case when $\Theta^\text{min} > \Pi \cdot U(\tau)$.

\[
\Theta^\text{min} = \max_{\tau_i \in \tau} \left\{ \min_{t_a, t_{a+1} \in \hat{S}_i} \left\{ \Theta^*(\Pi, \Delta, \tau) \right\} \right\}
\]  
according to Theorem 2 and Lemma 13. By Lemma 5, this is equivalent to
\[
\Theta^\text{min} = \max_{\tau_i \in \tau} \left\{ \min_{t_a, t_{a+1} \in \hat{S}_i} \left\{ \inf_{t \in \mathbb{N}^+} \left\{ \Theta^*(\Pi, \Delta, \tau) \right\} \right\} \right\}.
\]  
Applying Corollary 6, we find
\[
\epsilon^\text{min} \leq \max_{\tau_i \in \tau} \left\{ \left(\frac{k+1}{k}\right) \cdot \Theta^*(\Pi, \Delta, \tau) \right\}.
\]  
From this and the definition of $\Theta^*(\Pi, \Delta, \tau)$ the second inequality of this theorem follows. \hfill \Box
7. SIMULATIONS

In this section, we present simulation results and compare the performance of our proposed algorithms. We implemented six schedulability tests: exact test derived in Section 4 without any heuristics (i.e., iterative convergence to determine response time in Equation 12); exact test with heuristics (using response time lower bound and upper bound derived in Section 4.1 and 4.2); exact algorithm by [Easwaran et al. 2007]; our proposed approximate algorithm FP-MINIMUMCAPACITY; iterative convergence-based approximate test with heuristics and sufficient algorithm by [Shin and Lee 2008]. We denote these algorithms as BS-E, BS-E-h, MC-E, MC-A, BS-A-h and Suff respectively in the plots. The simulation parameters and value ranges are shown below:

1. The number of tasks in a task system \( \tau \) is \([4, 60]\) at 4-increments.
2. The system utilization \( U(\tau) \) is taken from the range \([0.1, 0.9]\) at 0.05-increments and individual task utilizations \( u_i \) are generated using UUniFast algorithm [Bini and Buttazzo 2004].
3. Each sporadic task \( \tau_i = (e_i, d_i, p_i) \) has a period \( p_i \) uniformly drawn from the interval \([10, 10000]\). The execution time \( e_i \) is set to \( u_i, p_i \). We assume \( d_i \leq p_i \) and is uniformly drawn from the interval \([\lceil e_i \rceil, p_i]\).
4. The component level scheduling algorithm is FP.
5. \( k \) is taken from the range \([1, 25]\) at 2-increments. \( \Pi \) is set in the range \([10, 10000]\); \( \Delta \) is equal to \( \Pi \).
6. Note that, we assume integral values for \( p_i, d_i, \Pi, \Delta \) and fractional values for \( e_i \) and \( \Theta \) for the ease of simulation, our results will still hold for non-integral parameters.
7. A 2.33 GHz Intel Core 2 Duo E6550 machine with 2.0GB RAM is used for simulations.
8. Each point in the plots represents mean of 1000 simulation runs with 95% confidence intervals.

For each simulation, for a specific task system size \( n \) and utilization \( U(\tau) \), we randomly generate taskset parameters \( u_i, p_i, e_i \) and \( d_i \) for each task \( \tau_i \). We execute three exact algorithms, two approximate algorithms, and the sufficient algorithm to determine exact, approximate and sufficient capacity, respectively. In [Dewan and Fisher 2012a], we compared the algorithms with a constrained deadline \( (p_i \leq d_i) \) sporadic task system scheduled by FF scheduler. We first compare the relative error\(^4\) of our proposed approximate algorithm (MC-A) with the sufficient algorithm (Suff), and iterative approximate algorithm (BS-A-h), with respect to the exact algorithm (MC-E)\(^5\).

In Figure 3, the relative error in the calculation of capacity for our algorithm is plotted as a function of task system utilization \( (n = 20; \Pi = 100; \Delta = \Pi; k = 3) \). For MC-A, the mean relative error is less than 5%, whereas for Suff it ranges from 40% to 85%. For the sufficient algorithm, relative error is very high due to the fact that the algorithm overestimates capacity. The relative error for our approximation algorithm does not vary much with the increase in system utilization, where as it decreases for the sufficient algorithm. A potential explanation for the this is that some of the functions of Suff for setting the capacity do not depend on the utilization, only the task and resource period parameters. Such functions will be constant over increasing utilization while the optimal capacity must increase as utilization increases. This results in a reduction in the relative error of these (non-utilization-dependent) functions. This observation continues to hold for the next two plots where we compare relative error by varying workload size (Figure 4) and resource period (Figure 5). We observe that for Suff, the relative error ranges from 25 – 95% and it increases with the increase of workload size and resource period. For MC-A, relative error for both these cases are below 5%, and it is independent of workload size and resource period. In Figure 6, we compare the relative error of the two approximation algorithms (MC-A and BS-A-h) by varying \( k \). For both the cases, we observe that the relative error is very low (below 1%) even for moderate value of \( k \) (\( \geq 5 \)). The relative error for BS-A-h is slightly higher than MC-A in all the above cases due to the fact that in the former case we have used a threshold of \( 10^{-6} \) while performing binary search of minimum

\(^3\)Note that we can obtain the exact capacity from FP-MINIMUMCAPACITY with \( k = \infty \).

\(^4\)Relative error is defined as follows: \( \frac{\Theta - \Theta^*}{\Theta^*} \) where \( \Theta^* \) is exact capacity and \( \Theta \) is either sufficient or approximate capacity.

\(^5\)Note that the relative error of BS-E and BS-E-h are equal to the threshold (equals \( 10^{-6} \) in the simulations) of the binary search used to determine minimum capacity.
capacity $\Theta$. Further, in BS-A-h, the approximation of the special tasks (which represent resource unavailability period) results slight overestimation of capacity by this algorithm.

Next, we compare the execution time (in ms) for all six algorithms varying system utilization, workload size and resource period. In Figure 7, execution time for these algorithms are plotted against system utilization. We observe that for the iterative convergence-based algorithms (BS-E, BS-E-h and BS-A-h), execution time decreases with increasing system utilization. This is due to...
the fact that the initial values of the response time for each task in task system \( \tau \) (see Equation 12) is higher at the initial steps of the iterative convergence algorithm. This results in fewer number of iterations for the response time to converge, and thus reduces overall execution time of the algorithm. The execution time for the other three algorithms (i.e., MC-E, MC-A, Suff) does not vary much with respect to utilization. While comparing the heuristic based iterative algorithms, we observe that the approximate algorithm BS-A-h performs worse than the exact algorithm BS-E-h. Since the approximate response time obtained by BS-A-h is higher than the exact response time obtained by BS-E-h, this results the former algorithm taking more iterations to converge. This is not trivial since at each iteration, the response time calculation for the approximate algorithm takes less time than the exact algorithm. We observe similar run-time performance of BS-A-h in the last two plots.

In Figure 8, we compare execution time for all algorithms varying workload size, and observe that as the number of tasks in the system increases, the execution time for all the algorithms except the constant-time sufficient algorithm increases. However, the execution time for MC-E grows at higher rate than BS-E-h, and crosses it at around \( n = 48 \). This is due to the fact that MC-E calculates minimum capacity for each point in the testing set, and the size of testing set grows pseudo-polynomially with the workload size (Equation 3), whereas the growth of the iterative algorithms (BS-E, BS-E-h and BS-A-h) are proportional to the workload size. Thus, we may conclude that the heuristic-based exact algorithm (BS-E-h) is more suitable when workload size is high. Finally, in Figure 9 execution time is compared against resource period. Here we observe that MC-E, MC-A and Suff do not vary much with resource period as expected. However, execution time for iterative exact algorithms decrease with increasing period, due to large initial response time for larger value of \( \Pi \).

Although for moderate task system size the execution time of MC-E is very competitive to MC-A (Figure 7), while determining the interface parameters using the capacity determination algorithm as a subroutine of the period selection algorithm [Fisher 2009], it significantly adds up to the time required to determine interface parameters. Therefore, the near-optimal capacity determination algorithm MC-A can be used with very low relative error \(< 1\% \) for \( k = 5 \). In this paper, we have considered constrain-deadline sporadic tasks in which case the execution time of the exact algorithm (MC-E) is proportional to \( d_{\text{max}} \) (i.e., maximum relative deadline among tasks). However, when task deadlines are arbitrary, this condition no longer holds and an approximate algorithm performs much better than the exponential time exact algorithm [Dewan and Fisher 2012b].

8. CONCLUSIONS AND FUTURE WORK

In this paper, we have addressed the minimization of interface bandwidth (MIB-RT) problem of explicit-deadline periodic (EDP) resource model in the context of compositional real-time systems with fixed-priority component level scheduling algorithms. In this setting, we have explored two approaches of fixed-priority scheduling: response time based analysis and testing set based analysis. For the former approach we derived an efficient bandwidth allocation scheme using dedicated uniprocessor schedulability test by modeling the resource unavailability period of the periodic resource model as special higher priority tasks. For the later approach, we have addressed the time complexity of the exact test [Easwaran et al. 2007] by devising a fully polynomial time approximation scheme (FPTAS). In this model, given fixed period and deadline of the EDP resource, for any sporadic task system our algorithm returns bandwidth that is at most a factor of \( (1 + \epsilon) \) greater than...
the optimal minimum bandwidth, for any $\epsilon > 0$. We showed that our algorithm has a polynomial time complexity in terms of the number of tasks in the task system $n$ and the approximation parameter $1/\epsilon$, whereas exact algorithms for MIB-RT problem on fixed-priority periodic resources may require pseudo polynomial time [Shin and Lee 2008; Easwaran et al. 2007] depending on the task parameters (i.e., task deadline or period) of the task system. We performed thorough simulations over synthetically generated tasks to compare both the response time based and the testing set point based approaches and identified which approach performs better in which scenario. We observed that our testing set based approximation algorithm improves performance over the sufficient algorithm [Shin and Lee 2008] by effectively reducing relative error. Further, the algorithm closely approximates the bandwidth from the exact algorithm regardless of the task parameters while maintaining polynomial time complexity.

Our result in this paper is for fixed-priority scheduling algorithms of constrained deadline sporadic task systems. Future direction from this paper may be to extend this work to more general task models such as fixed-priority task system with arbitrary deadlines, hybrid priority task systems, generalized multiframe task model etc. Furthermore, the approximation algorithms for MIB-RT on uniprocessor frameworks may be applicable to multiprocessor compositional frameworks (e.g., [Shin et al. 2008]) as well as to other compositional resource models.

Acknowledgments

This article is a revise and extended version of [Dewan and Fisher 2010] presented at the 16th Real-Time and Embedded Technology and Applications Symposium (RTAS). This research has been supported in part by a National Science Foundation CAREER Award (CNS-0953585) and a grant from Wayne State University’s Office of the Vice President for Research. We are sincerely thankful to anonymous reviewers for the detailed comments, corrections, and insightful directions.

REFERENCES


Bandwidth Allocation for Fixed-Priority-Scheduled Compositional Real-Time Systems


