Quantifying the Effect of Rare Timing Events with Settling-Time and Overshoot

Pratyush Kumar and Lothar Thiele
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
E-mail: {pratyush.kumar, thiele}@tik.ee.ethz.ch

***Abstract*—For hard real-time systems, worst-case timing models are employed to validate whether timeliness properties, such as meeting deadlines, are always satisfied. We argue that such a deadline-interface should be generalised in view of two separate motivations: (a) applications can tolerate bounded non-satisfaction of timeliness properties due to inherent robustness or relaxed quality requirements, and (b) worst-case timing models do not expose the occurrence of certain rare yet predictable events. As a more expressive interface, we propose the Rare-Event with Settling-Time (REST) model wherein, during rare events nominal timing models can be violated up to a known bound. Such a violation may lead to non-satisfaction of the timeliness properties up to a certain bound. We characterise this bound in terms of (a) the longest interval when the deadlines are not met, which we call the settling-time, and (b) the maximum number of jobs that can miss deadlines during the settling-time called the overshoot. We propose two models of rare events, characterised on an interval domain. For a single stream of jobs, we provide methods to tightly compute the settling-time and overshoot. For multiple streams of jobs on a single processor, we show that amongst schedulers agnostic to the occurrence of the rare event, the EDF scheduler optimally minimises the settling-time. In contrast, RM is not optimal within the class of fixed priority schedulers.**

***Keywords*-settling-time; rare events; weakly-hard real-time systems;**

I. Introduction

For a hard real-time system (task) to function correctly it is considered imperative that every job (individual invocation of a task) meets its deadline. In other words, the interface of the timeliness properties of application requirements and resource guarantees is characterised by the deadline which is to be always met. Let us call this the deadline-interface. For many real-world systems, such a deadline-interface may not completely express either application requirements or resource guarantees.

Let us first discuss application requirements. For many engineering applications, such as feedback-control, a certain number of deadlines can be missed without compromising correct functioning. This is due to the inherent robustness of the control algorithms and the underlying physical systems they control. In other systems, such as multimedia playback, a specified number of deadlines may be missed without affecting the user's expectation of quality. For such applications, specifying correct functioning in terms of always meeting a given deadline does not tightly represent their requirement.

The timeliness properties guaranteed by a resource may also admit rare irregularities. For instance, consider a high-priority but infrequent interrupt service routine that reduces the resource guaranteed to another task. Alternatively, consider the dynamic segment of a FlexRay bus which is a non-preemptive resource. A periodic task, once in several periods, may miss its assigned mini-slot and thereby be served only in the next dynamic segment. Such irregularity implies that a resource may "guarantee" stricter timing properties which are only occasionally violated. For such systems the deadline-interface does not tightly specify what the resource can guarantee to the application.

The limited expressiveness of the deadline-interface can lead to an analysis gap. More specifically, we may conclude a system does not meet its timing properties though it can function correctly. Bridging such an analysis gap requires us to (a) predictably estimate the frequency and effect of the rare events characterised in terms of a formal timing model, and (b) for a given rare event model devise analysis techniques to compute metrics of relevance to the application.

With a similar motivation, weakly hard real-time systems were defined by Bernat and Burns [1] as systems where deadlines can be missed, only up to a bounded number and in specific patterns. They defined a generalisation, and variants thereof, of the (m, k)-firm interface proposed in [2]. In the (m, k)-firm model, at least m jobs out of any k consecutive jobs must meet their deadlines for correct functioning. For many practical settings, this interface seems to be adequate for tightly characterising the application requirements. For instance, a similar interface was shown to characterise asymptotic stability of control systems as switched systems with different feedback delays [3].

To compute (m, k)-firm or similar interfaces, we depend on the predictability of the rare events. For instance, in the case of rare high priority interrupt service routines, we need to bound how frequently such interrupts can arise and how long such routines would execute. The minimum requirement on any such model of the rare event is the minimum inter-arrival time under a sporadic occurrence pattern. Such models indeed are only bounds, in so far as we may not be able to predict tightly the patterns of the rare events. Once such a rare event model is in hand, we can then characterise the resource guarantee in terms of a

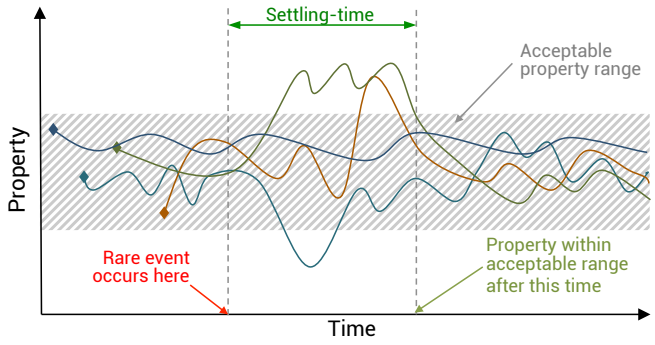

Figure 1. Representation of the occurrence and rejection of rare events

nominal (without rare events) model which can be deviated from as characterised by the rare event model. This exposed "two-part"-model (nominal and rare event models) can then enable the computation of timing properties of interest, such as the (m, k)-firm interface.

It is possible, though not necessary, that such rare events lead to deadline misses, perhaps for several subsequent jobs. To quantitatively characterise such behaviour we propose the metric of settling-time. It is defined as an upper bound on the time interval for which jobs may miss deadlines subsequent to the occurrence of a rare event. This can be considered analogous to a requirement in control systems, where the effects of a disturbance (in our case a rare event) need to be rejected (in our case by returning to timeliness) within the settling-time. In a similar vein, we define "overshoot" as the maximum damage incurred during the settling-time. We characterise such an overshoot by the maximum number of jobs that can miss their deadlines during the settling-time and the worst-case response time of such jobs.

We visualise the metrics in Figure 1. In the figure several trajectories of a system along a property of interest, for instance response time, are plotted. After a rare event occurs, the property can deviate from the accepted region. The duration of such deviation, maximised over all possible system trajectories, is defined as the settling-time. Indeed, the rare event could also occur at a different time instant and the settling-time metric must consider all possible valid occurrences of the rare event. The overshoot is characterised by the maximum deviation of the system property from the accepted region.

Computing such a settling-time is not straightforward. Relating the "magnitude" of the rare event to the settling-time can be non-trivial. For instance, on a fully utilised processor, even a small overrun of a job can lead to deadline misses that continue indefinitely into the future. On the other hand, in a system with spare capacity, rare events may lead to no deadline misses at all. The scheduling policy can also affect the results. The domino effect of EDF scheduler is an example [4]. Furthermore, in systems with variability, such as job arrival patterns with jitter or an irregular resource supply, it is important to identify the critical system trace for the occurrence of a rare event. This problem of variability can arise even in the characterisation of the occurrence of the rare event if it is not limited to a single instance. The settling-time must be a true upper-bound, independent of the time of occurrence of the rare-event and variability in temporal patterns of other tasks.

Main contributions: We define two models to characterise rare yet predictable events: demand overload, and supply shortage. The former captures rare events where the nominal task model is violated, for instance due to more arriving jobs or larger execution time. The latter captures rare events where the resource guarantee is violated, for instance interference from a high priority interrupt. Both models are modelled on curves as used in Network Calculus [5] and Real-Time Calculus [6].

For a single stream of jobs queueing in First-Come-First-Serve (FCFS) discipline, for either model of rare events, we provide a tight computation of the settling-time. In addition, we compute the maximum overshoot in terms of the maximum number of jobs that miss deadlines and their worst-case response times.

We then extend our analysis to multiple streams of jobs scheduled on a single processor. We present analytical results for fixed-priority systems where a rare event in a higher priority task can lead to deadline misses of a lower priority task. Similar such results can be presented for other scheduling policies. We show that within the class of schedulers which are agnostic to the occurrence of rare events or missing of deadlines, EDF optimally minimises the settling-time. This extends EDF's optimality under schedulability for the deadline-interface. This implies that a system designer need not plan differently to accomplish (a) optimal schedulability with no rare events, or (b) minimum settling-time with rare events, of whatever magnitude. However, an analogous result for rate-monotonic (RM) priority assignment within the class of fixed priority schedulers does not hold. In other words, the settling-time may be minimised by priority assignments different from rate-monotonic priority assignment.

Organisation: We position our work in relation to existing literature in Section II. We propose the two models of rare events in Section III. We present the computation of settling-time and overshoot for either model of rare event in Section IV. We then extend our results to consider multiple streams of jobs under scheduling policies in Section V. Finally we conclude in Section VI.

II. Related Work

In this section, we position our work in relation to other similar approaches from existing literature.

Stochastic approaches. Though we consider systems which are not traditionally hard real-time systems, we are only interested in deterministic guarantees. This invalidates

use of certain interfaces common in soft real-time systems which assume randomness and independence properties. For instance, statistical metrics such as average failure rate [7] or timing models which capture probabilistic variation of execution times [8] or arrival patterns [9] are not relevant here.

Overload and (m, k)-firm scheduling. In a system with non-zero settling-time, we have a condition of overload. Overload scheduling has been studied by several researchers primarily focussing on defining metrics such as value functions [10], [11] and scheduling techniques that maximise such metrics [12]. Similarly, several scheduling techniques have been proposed specifically for (m, k)-firm systems. A best-effort scheduling algorithm was proposed in the original paper [2]. Subsequently, guaranteed approaches such as Dynamic Windows-Constrained Scheduling (DWCS) [13], Skip-Over scheduling [14] and dual priority scheduling [15] have been proposed. In contrast, we propose a refined timing model of overload, namely occurrence of rare-events deviating from a nominal model, and quantify the response to such an event in terms of a specific metric namely settling-time. Unlike the referenced work, we focus on the tight analysis of the proposed models rather than proposing best-effort scheduling techniques. As a precursor to a later result, we show that amongst schedulers agnostic to arrival of rare events, the EDF scheduler optimally minimises the settling-time.

Sensitivity analysis. In our model, the rare event characterises a deviation from a nominal model. Study of such deviations is common in sensitivity analysis. Several works such as [16] and [17] have studied the feasibility region around a considered design point as a metric of sensitivity. In contrast, by allowing deadlines to be missed after rare events, we quantify the damage that a rare event imposes on a system.

Aperiodic scheduling. The rare event model introduces additional jobs in the system, which can be sporadic. Design and analysis of scheduling sporadic job patterns has been studied using fixed [18] and dynamic priority [19] servers. Such approaches can compute the worst-case response times of the nominal and additional jobs. However, we are additionally interested in the settling-time metric which is the longest time-interval for which deadlines are missed. Such a metric has not been analysed for aperiodic job scheduling.

Feedback computing In the context of real-time systems, settling-time metric was proposed in [20]. In this and related works, the authors investigate the use of feedback control in scheduling tasks, and use settling-time, amongst others, as a metric of quality of such a technique. Settling-time is defined as the time required to return to stability (which can be in terms of a given miss ratio). The authors compute settling-time using methods from control-theory after suitably approximating a real-time system. Instead, in our approach settling-time is explicitly defined in terms of deadlines and we tightly compute it for a given real-time system for commonly used scheduling principles such as EDF and RM.

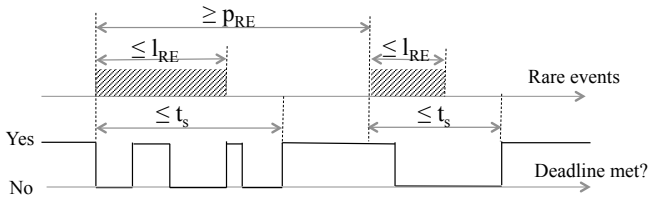

Figure 2. Representation of a REST model

III. Rare Events with Settling-Time (REST)

We start by describing the Rare Events with Settling-Time (REST) model. In the REST model, a system is characterised by a *nominal* timing model denoted as M. When the system operates exclusively in conformance to the nominal timing model, the jobs of the system are known to meet some desired *deadline* denoted D. The occurrence of a *rare event* deviates the system's timing behavior from M by up to a known bound characterised by the *rare event's model* denoted as M_{RE}. The rare event can last for an interval of time of length no more than l_{RE} time units. We consider a sporadic model of the rare event. The minimum time duration between the start of two consecutive rare events is given by p_{RE}. Rare events are not allowed to overlap, i.e., $p_{\text{RE}} > l_{\text{RE}}$. While estimating such parameters for some rare events can be practically difficult and inaccurate, they are necessary to safely bound the effect of the rare events in the meeting of deadlines.

The *settling-time*, denoted as $\mathtt{t_s}$, is the maximum length of time up to which jobs can miss the deadline D after the start of a rare event. The worst-case response time of jobs during the settling-time is denoted as $\widehat{D}$. The maximum number of jobs hat can miss their deadlines during the settling-time is denoted as $\widehat{N}$. A system with $\mathtt{t_s} = 0$ is said to be *unconditionally stable* with all jobs meeting the required deadline. On the other hand, a system with $\mathtt{t_s} \geq p_{\text{RE}}$ is said to be *unstable* with no job guaranteed to meet its deadline. In contrast, any system with $\mathtt{t_s} < p_{\text{RE}}$ is said to be stable. We represent this model in Figure 2.

Rare event model M_{RE} characterises how the timeliness properties of a system deviate from the nominal model. To model the discussed phenomena we propose two kinds of rare events (a) demand overflow, and (b) supply shortage. Before defining these rare events, we discuss some preliminaries in representing execution demand and supply in the interval domain.

A. Preliminaries - Representing in the interval domain

To characterise and subsequently analyse rare events we need to represent the timed properties of traces of execution

demand and supply. As such traces may exhibit variability in the time-domain, it is generally convenient and effective to represent them in the interval domain using abstract curves. This was first proposed in Network Calculus [5] and later adopted in Real-Time Calculus (RTC) [6]. We will briefly describe the main principles here.

Let us consider a stream of jobs being served by a resource in first-come-first-serve manner. We refer to such a stream of jobs as a task. The execution demand of the task is characterised by the *arrival function* R, where $R(t)$ denotes the cumulative execution demand of all jobs that have arrived up to and not including the time t. Execution demand can be expressed in different units such as number of processor cycles or the number of bits to be transmitted. If a task is said to be characterised by an *arrival curve* α, then any arrival function of that task must satisfy the property

$$R(t+\Delta) - R(t) \leq \alpha(\Delta), \quad \forall\ t,\ \Delta \geq 0. \tag{1}$$

In words, the arrival curve α upper-bounds execution demand of jobs that arrive within an interval of a given length.

Similarly, let the *service function* C characterise the supply of a resource, where $C(t)$ denotes the cumulative resource units provided by the resource up to and not including the time t. If a given resource is said to be characterised by a *service curve* β, then any service function of the resource must satisfy the following property

$$C(t+\Delta) - C(t) \geq \beta(\Delta), \quad \forall\ t,\ \Delta \geq 0. \tag{2}$$

In words, the service curve β lower-bounds the resource units guaranteed to a task being served by the resource. As an example, the full service of a resource that provides k resource units per time unit has a service curve $\beta(\Delta) = k{\cdot}\Delta$.

The arrival and service curves enable compact representation of a wide variety of arrival and service functions. Furthermore, computation of useful bounds such as maximum delay and backlog can be efficiently performed using a min-plus calculus on the abstract curves [21]. Finally, a well established compositional technique enables analysis of distributed systems using interface variables [22], [23].

Example 1. *Consider a periodic control task with period and relative deadline of 5 ms. Every 4th job has a worst-case execution demand of 2 ms, while for every other job it is 1 ms. The task is served by a TDMA resource of slot of length 2.5 ms reserved on a period of 5 ms. The arrival and service curves are shown in Figure 3(a). It can be shown that all the jobs meet the deadline of 5 ms using methods from Network Calculus [5].* □

B. Demand Overflow

The execution demand of a task may exceed the nominal model because of two reasons: jobs may execute for longer and/or additional jobs may arrive. These two cases can be jointly expressed with input arrival functions. We formally define such a rare event.

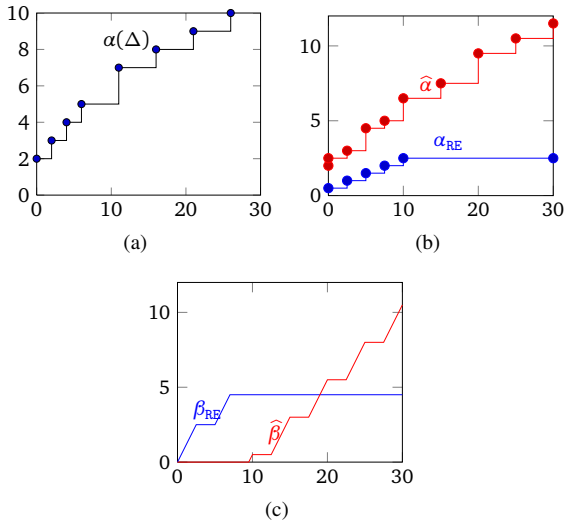

Figure 3. Arrival and service curves for Examples 1, 2 and 3. All axes are in the units of ms.

Definition 1 (Demand overflow)**.** *The rare event model of a demand overflow is given by an arrival curve denoted α_{RE} which upper-bounds the additional demand, over the nominal model, in any time interval of a given length.* □

The arrival curve of the demand overflow is interpreted as follows. Let R_{RE} denote the additional arrival function due to a demand overflow occurring at time 0. Then, all such arrival functions must satisfy the property

$$R_{\text{RE}}(v) - R_{\text{RE}}(u) \leq \alpha_{\text{RE}}(v-u), \quad \forall\ u,\ v\ \in [0, l_{\text{RE}}], \tag{3}$$

$$R_{\text{RE}}(v) = R_{\text{RE}}(l_{\text{RE}}), \quad \forall\ v \geq l_{\text{RE}}. \tag{4}$$

where l_{RE} is the maximum length of the occurrence of the rare event. The arrival function of the task with such a rare event, denoted as $\widehat{R}$, is given by the sum of the nominal arrival function, R, and the rare event's arrival curve, R_{RE},

$$\widehat{R} = R_{\text{RE}} + R. \tag{5}$$

In all subsequent notation, we explicitly qualify timing properties as being nominal or not. In the absence of any qualification, we refer to the properties that consider both the nominal and rare event models.

The arrival function $\widehat{R}$ can vary because (a) the time of occurrence of the rare event is not fixed, and (b) the nominal and rare-event arrival functions are themselves variable. To compactly characterise such variability we can again abstract such functions in the interval domain, as proposed in the following result.

Theorem 1. *The arrival curve of a task, $\widehat{\alpha}$, with a nominal arrival curve α and a demand overflow with an arrival curve*

$\alpha_{\texttt{RE}}$ *is tightly given as*

$$\widehat{\alpha} = \alpha + \alpha_{\texttt{RE}}. \tag{6}$$

□

Proof: Sketches of all proofs are presented in the appendix. ■

Note that in the above abstraction, we have included the effect of only one rare event. As we will show in later sections, such an abstraction is sufficient to tightly compute the settling-time for a stable system i.e., $\texttt{t}_\texttt{s} < p_{\texttt{RE}}$, with no overlapping rare events, i.e., $p_{\texttt{RE}} > l_{\texttt{RE}}$.

Example 2. *For Example 1, let the plant move into a different unstable dynamics at most once every 10 s for at most 10 ms. In this dynamics, the plant generates additional control tasks every 2.5 ms with worst-case execution demand of 0.5 ms each. Such a rare event can be modelled as a demand overflow with $l_{\texttt{RE}}$ = 10 ms, $p_{\texttt{RE}}$ = 10 s and $\alpha_{\texttt{RE}}$ as shown in Figure 3(b). Also shown is the arrival curve $\widehat{\alpha}$.* □

C. Supply Shortage

So far, we have discussed rare events when the resource demand of jobs is larger than nominally expected. Deviation from a nominal behavior can also be due to a reduced resource supply. One example is when the execution demand of a higher priority job is larger than a nominal model. This translates to a reduced resource supply for a lower priority job. As a different example consider a thermally-constrained system which is expected to maintain the temperature below a threshold. If occasionally the threshold is breached then the resource supply of a task is reduced due to enforced speed scaling. We refer to such rare events as supply shortages. We formally describe them here.

Definition 2 (Supply Shortage)**.** *The rare event model of a supply shortage is given by a service curve $\beta_{\texttt{RE}}$ which upper-bounds the reduction in supply of a resource, below a nominal model, in any time interval of a given length.* □

The service curve of a supply shortage is interpreted as follows. Let $C_{\texttt{RE}}(t)$ denote the supply function lost by a resource until time t, due to a supply shortage occurring at time 0. Then, we have

$$C_{\texttt{RE}}(v) - C_{\texttt{RE}}(u) \leq \beta_{\texttt{RE}}(v-u), \quad \forall\ u,\ v\ \in [0, l_{\texttt{RE}}]. \tag{7}$$

$$C_{\texttt{RE}}(v) = C_{\texttt{RE}}(l_{\texttt{RE}}), \quad \forall\ v \geq l_{\texttt{RE}}, \tag{8}$$

where $l_{\texttt{RE}}$ is the maximum length of occurrence of the rare event. Note that the service curve of the supply shortage is an *upper-bound* on the cumulative loss in a given interval of time. (In contrast, nominal service curve is a lower-bound on the cumulative supply in any given interval of time.) The supply function of the task, denoted as $\widehat{C}$, as a function of the nominal supply function C and the service curve of the supply shortage $C_{\texttt{RE}}$ is given as

$$\widehat{C} = \{C - C_{\texttt{RE}}\}^{\uparrow}, \tag{9}$$

where,

$$f^{\uparrow}(t) = \max_{\tau < t} f(\tau). \tag{10}$$

To compactly represent all such possible supply functions $\widehat{C}$, we abstract them into a curve in the interval domain as given in the following result.

Theorem 2. *The service curve of a resource, $\widehat{\beta}$, with a nominal service curve β and a supply shortage with a service curve $\beta_{\texttt{RE}}$ is tightly given as*

$$\widehat{\beta} = \{\beta - \beta_{\texttt{RE}}\}^{\uparrow}. \tag{11}$$

□

Example 3. *Consider the TDMA resource of Example 1. Let a low-level process interrupt the working of the TDMA cycle for at most 7 ms once every 20 s. Such a rare event can be modelled using a supply shortage with $l_{\texttt{RE}}$ = 7 ms, $p_{\texttt{RE}}$ = 20 s and $\beta_{\texttt{RE}}$ as given in Figure 3(c). Also shown is the total service curve $\widehat{\beta}$.* □

IV. Analysis of the REST model

In this section, we present the analysis of the REST model for a single task and resource. In particular, we compute the settling-time, bounds on the maximum number of jobs that miss the deadline and their worst-case response time.

A. Computation of Settling-Time

The settling-time must be computed for both the presented models of the rare events, namely demand overflow and supply shortage. To homogenise the presentation we always represent the arrival and supply curves of the task and resource, respectively, as $\widehat{\alpha}$ and $\widehat{\beta}$. In systems with no demand overflow rare event, we set $\widehat{\alpha}$ to the nominal value of α. Similarly, for systems with no supply shortage rare event, we set $\widehat{\beta}$ to the nominal value of β.

We begin by defining a function that will be used in subsequent results.

Definition 3 (Function TS)**.** *Function* TS *with three arguments α, β and D is defined as*

$$\texttt{TS}(\alpha, \beta, D) = \sup\{\Delta \geq 0 :\ \alpha(\Delta - D) > \beta(\Delta))\}. \tag{12}$$

□

In words, $\texttt{TS}(\alpha, \beta, D)$ is the smallest time after which α shifted to the right by D units is always less than β. This is similar to the use of demand bound function to analyse schedulability of EDF systems proposed in [24].

With the use of the above definition we can now present the result on computing the settling-time.

Theorem 3 (Settling-Time of a Single Task System)**.** *The settling-time of a task with arrival curve $\widehat{\alpha}$ and nominal deadline D, served by a resource with service curve $\widehat{\beta}$ is tightly given by*

$$\mathtt{t_s} = \mathtt{TS}(\widehat{\alpha}, \widehat{\beta}, D). \tag{13}$$

□

Note that, if the value of the settling-time computed as above is not smaller than $p_{\mathtt{RE}}$, we say that the system is unstable. Such a system is not characterised by any settling-time as all jobs can potentially miss their deadlines. If the value of computed settling-time is 0, we say that the system is unconditionally stable. For systems with settling-time in the interval $(0, l_{\mathtt{RE}} + \widehat{D})$, where $\widehat{D}$ is the worst-case response time of jobs during the settling-time, we set $\mathtt{t_s} = l_{\mathtt{RE}} + \widehat{D}$. In other words, for systems where at least one job can miss a deadline, the minimum value of $\mathtt{t_s}$ is set to the maximum length of the rare event plus the worst-case response time.

The above result is tight: there exists a valid trace of the system for which deadlines are indeed missed up to the settling-time after the occurrence of the rare event. Thus, independent of when the rare event occurs, its exact timing behavior, and for either model of the rare event, the above analytical result computes the tight settling-time.

B. Computation of Overshoot

In the REST model, during the settling-time, the timeliness properties are not satisfied. It may be useful for an application to quantify this non-timeliness. This is similar to the notion of an overshoot in a control system subsequent to a disturbance. We characterise such an overshoot in terms of two metrics: (a) the worst-case response time of jobs during the settling-time, and (b) the maximum number of jobs that miss their deadlines within the settling-time. In the subsequent results we compute these metrics.

Theorem 4 (WCRT during Settling-Time, $\widehat{D}$)**.** *The worst-case response time (WCRT) of a task in the REST model, during the settling-time, denoted as $\widehat{D}$, is tightly given by*

$$\widehat{D} = \mathtt{Del}(\widehat{\alpha}, \widehat{\beta}), \tag{14}$$

where, Del *is the maximum horizontal distance between two curves and is given as*

$$\mathtt{Del}(f, g) = \sup_{t \geq 0} \{\inf\{\Delta \geq 0 : \ f(t - \Delta) \leq (g)(t)\}\} \tag{15}$$

□

The above result is a direct application of response-time computations from Network Calculus [5]. They apply under the assumption that rare events do not overlap, i.e., $l_{\mathtt{RE}} < p_{\mathtt{RE}}$.

Theorem 5 (Maximum number of jobs missing deadlines, $\widehat{N}$)**.** *Consider a task with arrival curve $\widehat{\alpha}$ being served by a resource with service curve $\widehat{\beta}$. The maximum number of jobs that can miss deadlines during settling-time, denoted as $\widehat{N}$, is equal to the number of jobs that miss the deadline until time $\mathtt{t_s}$ for the specific arrival and service functions*

$$\widehat{R}(t) = \widehat{\alpha}(t), \tag{16}$$

$$\widehat{C}(t) = \widehat{\beta}(t). \tag{17}$$

□

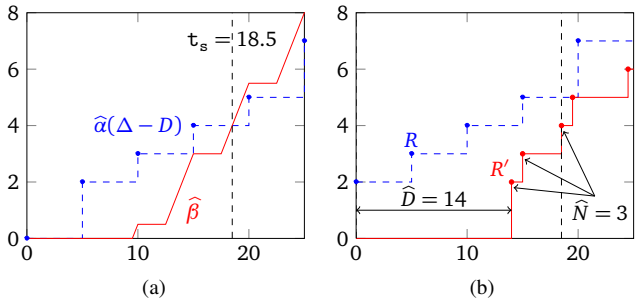

Figure 5. Analysis of rare event of Example 5. All times are in ms.

The above theorem provides the critical input of the system in terms of the arrival and service traces. By simulating these traces we compute $\widehat{N}$.

Since a rare event can only occur at most once every $p_{\mathtt{RE}}$ amount of time, we can know that at most $\widehat{N}$ jobs miss deadlines in specific intervals of length $p_{\mathtt{RE}}$. For systems where upper and lower bounds exist on how many jobs can arrive within different intervals of time, we can translate this representation to [1] and other similar models.

Example 4. *Consider the task and resource models of Example 1 and the demand overflow rare event model of Example 2. $\widehat{D}$ computed according to (14) is 5.5 > D. Thus, we have at least one missed deadline after the occurrence of a rare event.* $\mathtt{t_s}$ *computed according to (13) is 14.5. However, as this is smaller than $l_{\mathtt{RE}} + \widehat{D}$ (= 10+5.5=15.5), we set* $\mathtt{t_s}$ *to 15.5. Since this is less than $p_{\mathtt{RE}}$, we have a stable system. $\widehat{N}$ computed by simulating the critical input of Theorem 5 is found to be 1. Thus, at most 1 job can miss its deadline subsequent to a rare event.*

We illustrate the tightness of the results with a valid trace of the system shown in Figure 4. The arrival times of jobs are marked with upwards arrows with the execution time specified in brackets. The jobs that belong to the rare event are denoted with a subscript e. The rare event occurs in the time interval $[2.5, 12.5]$. Only 1 (= $\widehat{N}$) job, namely J_e^5, misses its deadline with response-time of 5.5 (= $\widehat{D}$), finishing at time 18. The settling-time for this trace is $18 - 2.5 = 15.5$, which equals the computed bound. □

Example 5. *Consider the task and resource models of Example 1 and the supply shortage rare event model of Example 3. The corresponding service curve of the resource*

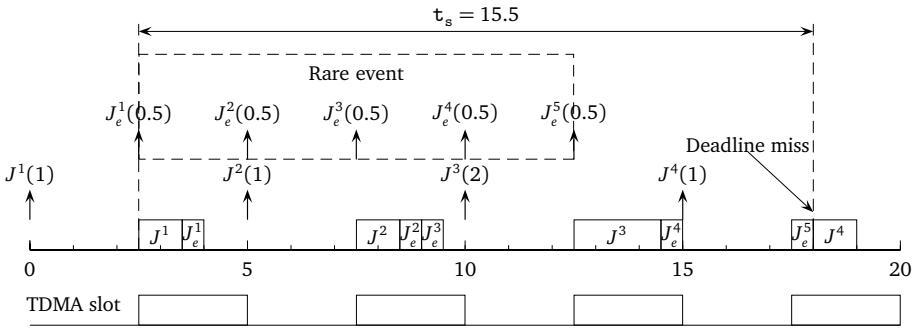

Figure 4. Critical trace for Example 4 with worst-case settling-time. Times are shown in ms.

$\widehat{\beta}$ was already computed in Example 3. We apply Theorem 3, to compute the settling-time. This is shown in Figure 5(a). The furthest point beyond which $\widehat{\alpha}$ shifted to the right by $D = 5$ is always below the computed $\widehat{\beta}$ is $\mathtt{t_s} = 18.5$. We then simulate the critical arrival and service functions of the system and compute the output. This arrival function is shown in Figure 5(b) as R. The service function is the same as $\widehat{\beta}$ shown in Figure 5(a). The output function is shown as R'. From this critical trace we can compute $\widehat{D}$ and $\widehat{N}$. The worst-case delay suffered by a job is $\widehat{D} = 14$. The maximum number of jobs that miss their deadline $\widehat{N} = 3$. □

V. Computing Settling-Time with Multiple Tasks

Most often multiple tasks share resources on which the timing behaviour of one task affects the other. Potentially, a rare event on one task can lead to deadline misses on other tasks. In this section, we study the analysis of REST systems with multiple tasks scheduled under (a) fixed priority and (b) earliest deadline first (EDF) policies. The focus in this work is not to design new scheduling techniques specific to the REST model, but rather to analyse commonly used schedulers under the settling-time metric. Note that in all subsequent discussion, we consider the uni-processor setting.

We restrict our analysis to rare events only on one task or resource. Analysis of multiple sources of rare events is a topic of future work. Nevertheless, we analyse the effect of the single source of rare event on all tasks of the system. For a system with multiple tasks, settling-time is the latest time after which no job of *any* task in the system misses its deadline relative to the start of the rare event.

We continue to use our earlier notation where arrival and service curves are denoted as $\widehat{\alpha}$ and $\widehat{\beta}$, respectively. In the absence of respective rare events, these curves are set equal to the respective nominal curves.

A. Fixed Priority Scheduler

We first consider the fixed priority scheduling policy. We assume each task has a distinct priority and the system is fully preemptive. In our notation, task T_i has a higher priority than task T_j iff $i < j$.

Theorem 6. *Consider a REST system with a resource with service curve $\widehat{\beta}$. Under a preemptible fixed-priority scheduling policy, the resource serves n tasks $T_1, T_2, \ldots T_n$, in decreasing order of priority. Task T_i has an arrival curve $\widehat{\alpha}_i$ and a relative deadline D_i. Then, the settling-time of the system is tightly given as*

$$\mathtt{t_s} = \max_{i \in 1,2,\ldots,n} (\mathtt{t_s})_i \tag{18}$$

where,

$$(\mathtt{t_s})_i = \mathtt{TS}\left(\widehat{\alpha}_i, \left(\widehat{\beta} - \sum_{j<i} \widehat{\alpha}_j\right)^{\uparrow}, D_i\right). \tag{19}$$

□

In the above result, we compute the settling-time for each task, denoted as $(\mathtt{t_s})_i$, similar to Theorem 3. The service curve used in (19) is given by the remaining service curve available from higher priority jobs [6]. Then, the maximum such settling-time, across tasks, is set as the system-wide settling-time. Note that the settling-time of a task does not depend on properties of other tasks of lower priority, thereby providing an inherent degree of isolation.

Example 6. *Consider three periodic tasks A, B, C, with periods and relative deadlines 3, 4 and 5, respectively and worst-case execution times of 1 each. The priority is A > B > C. Under this nominal model, the system is schedulable. Task B exhibits a demand overflow rare event where up to 3 additional jobs may arrive instantaneously, at most once every second. The settling-time of task C turns out to be longest. We illustrate this computation. Since, the resource is fully available, we have $\widehat{\beta} = \Delta$. Then, the available service for task C is given by $(\Delta - (\alpha_A + \alpha_B))^{\uparrow}$ and is as shown in Figure 6(a). Then by applying (19), we can compute the settling-time to be 12, as shown. We show the trace of the system where this settling-time is observed, in Figure 6(b). The extra jobs corresponding to the rare event are denoted with a subscript "e". The jobs which miss deadline are shown with a shade.* □

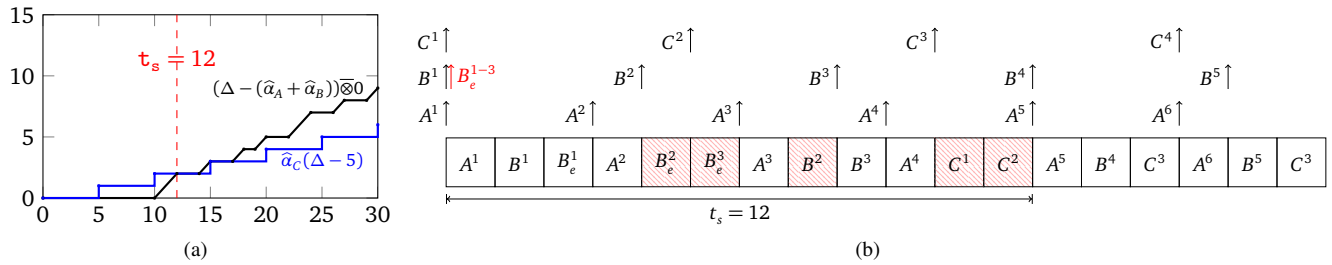

Figure 6. (a) Analytical computation of settling-time for Example 6. (b) A trace with the worst-case settling-time for the same example. All times are in ms.

B. Earliest Deadline First Scheduler

Now we consider the earliest deadline first (EDF) policy. Recall that in our task model, every task is assigned a relative deadline D shared by all jobs of that task. This deadline can be used to schedule the tasks under EDF. For such a system, the settling-time is given by the following result.

Theorem 7. *Consider a REST system with a resource of service curve $\widehat{\beta}$ serving under EDF n tasks $T_1, T_2, \ldots, T_n$. Task T_i has an arrival curve $\widehat{\alpha}_i$ and a relative deadline D_i. Then, the settling-time of the system is tightly given as*

$$\mathtt{t_s} = \mathtt{TS}\left(\sum_i \widehat{\alpha}_i(\Delta - D_i), \widehat{\beta}, 0\right). \tag{20}$$

□

In contrast to the fixed priority scheduler, for the EDF scheduler we do not compute task-specific settling-times. Instead, we compute a single system-wide settling-time that depends on parameters of all tasks. This lack of task-specific bounds can lead to the domino effect [4] where jobs from several tasks may miss deadlines.

Example 7. *Consider the task and rare event models of Example 6. If this system is scheduled under EDF, the settling-time computed by applying Theorem 7 is 7, as shown in Figure 7(a). The trace that leads to this worst-case settling-time is shown in Figure 7(b).* □

C. Settling-Time as a Metric

In real-time systems, traditionally, the question of interest has been: Is a given system schedulable? There is great interest in the design and analysis of scheduling policies which ensure schedulability. Indeed, a clear notion of *optimality under schedulability* exists: A scheduling policy is optimal under schedulability iff any system with given task model, resource model and deadline that is not schedulable under that policy is not schedulable under any other policy. The optimality of the EDF scheduling policy is well known [25].

For a system with rare events that can miss deadlines, a lower settling-time is desirable: the timeliness properties are not guaranteed for a shorter duration after the start of a rare event. Settling-time can be considered as a metric to compare scheduling policies in a REST system. To formally enable such comparison we define a notion of optimality.

Definition 4 (Optimality under the REST model)**.** *A scheduling policy is said to be optimal under the REST model, iff for any given nominal task model, resource model, deadline, and rare event model, the settling-time obtained with that scheduling policy cannot be lowered under any other scheduling policy.* □

A question of interest is the relation between optimality under schedulability and optimality under the REST model: are they conflicting or aligned properties? We characterise this relationship with the following result.

Theorem 8. *A scheduling policy that is optimal under the REST model is also optimal under schedulability.* □

The above result establishes that optimality under the REST model subsumes optimality under schedulability. This prompts the natural question whether there exists an optimal scheduling policy under the REST model.

In this work, we restrict our focus to the class of schedulers that are agnostic to the occurrence of rare events and the response time of the finished jobs. In other words, we do not assume any additional monitoring and subsequent reactionary steps on the part of the scheduler. For this class of schedulers the following result identifies an optimal scheduler.

Theorem 9 (Optimality of EDF under the REST model)**.** *EDF scheduler is optimal under the REST model within the class of schedulers with no run-time monitoring.* □

The above result establishes that EDF is the scheduling policy of choice to optimally minimise settling-time, in spite of the seemingly disappointing domino effect [4]. Thus, meeting schedulability constraints under nominal models and minimising system-wide settling-time subsequent to rare events, of whatsoever magnitude, are simultaneously solved by using an EDF scheduler.

In some systems it may be desirable to immunise certain tasks from potential rare events in other tasks, even at the cost of a longer system-wide settling-time. As already mentioned, this is possible in a fixed priority scheduler.

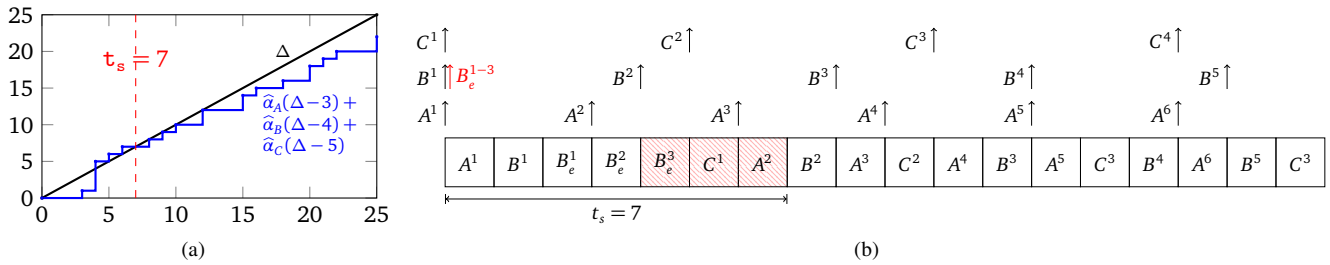

Figure 7. (a) Analytical computation of settling-time for Example 7. (b) A trace with the worst-case settling-time for the same example. All times are in ms.

Priority	$\mathtt{t_s}$	$(\mathtt{t_s})_A$	$(\mathtt{t_s})_B$	$(\mathtt{t_s})_C$
A > B > C	12	0	6	12
A > C > B	14	0	14	0
B > A > C	12	7	0	12
B > C > A	14	14	0	6
C > A > B	14	0	14	0
C > B > A	14	14	5	0

Table I
COMPUTED VALUES OF SETTLING-TIME FOR DIFFERENT PRIORITY ASSIGNMENTS FOR TASK-SET OF EXAMPLE 6

This is a notable advantage and hence it is interesting to investigate the effect of different priority assignments on the settling-time. We begin with an example.

Example 8. *For Example 6, settling-times for different priority assignments are shown in Table I. For each case, in the absence of any exceptions, all deadlines are met. The table also shows the task-level settling-times.* □

In the above example, for all priority assignments, under nominal timing models, the system is schedulable. However, in the presence of the rare event, the settling-times vary across priority assignments. The (joint) least value is obtained for the priority assignment in decreasing order of period. This prompts the question: Is the rate-monotonic (RM) priority assignment the optimal under the REST model among the class of fixed priority schedulers? We show with an example that this is not true.

Example 9. *Consider a periodic server with a period of 5 ms and budget of 3 ms. Let this resource have a supply shortage rare event where it is unavailable for up to 5 ms. Let this resource serve, under fixed-priority, two periodic tasks D and E with periods and deadlines 6 ms and 25 ms, respectively, and execution time of 2 each. The settling-time for priority $D > E$ is 23, while that for $E > D$ is 19.* □

In the above example, by assigning the less frequent task a higher priority, we can reduce the settling-time. This does not follow the rate-monotonic (RM) approach. In contrast to the optimality of EDF, we do not have a similar result for RM. This potentially is another contentious point in the comparison [26] of the two famed schedulers!

Theorem 10 (Non-optimality of RM)**.** *Rate-monotonic priority assignment is not optimal under the REST model within the class of fixed priority scheduling policies.* □

This confirms that optimality under the REST model is strictly stronger than optimality under schedulability.

VI. CONCLUSION

We proposed the Rare Events with Settling-Time (REST) model as a refinement of the deadline-interface in light of relaxed timeliness requirements from applications and the irregularities inherent in the timing models of resources. We defined rare events as deviating from a nominal timing model and characterised it as a sporadic event of known upper-bound on its magnitude. Our main contribution was the definition of settling-time as the longest time window after the rare event when the system needs to return to guaranteeing timeliness properties. We quantified the overshoot during the settling-time in terms of worst-case response time and maximum number of jobs missing deadlines. We provided tight and analytical solutions to compute settling-time and overshoot metrics for timing models that capture variability. In particular we showed that the timing models commonly used for analysis of real-time systems, suffice for tightly computing the settling-time. We analysed fixed priority and EDF scheduling policies in terms of minimising settling-time. We proved within the class of scheduling techniques that are agnostic to the occurrence of rare events or deadline misses, EDF optimally minimises the settling-time. On the other hand, interestingly, RM is not optimal within the class of fixed-priority schedulers.

One direction of future work is the analysis of systems with multiple sources of rare events, i.e., with several tasks and/or resources having rare events. Techniques have to be devised to aggregate the effects of the rare events. The other direction is to compute settling-time in distributed systems which are inherently cyclic. An interesting question is the characterisation of cycles over which the settling-time increases, i.e., inherently unstable systems.

ACKNOWLEDGEMENTS

This work was supported by EU FP7 projects EURETILE and PRO3D, under grant numbers 247846 and 249776, respectively. We thank the anonymous shepherd for help in improving the quality of the paper.

REFERENCES

[1] G. Bernat, A. Burns, and A. Llamosí, "Weakly hard real-time systems," *IEEE Trans. Computers*, vol. 50, no. 4, pp. 308–321, 2001.

[2] M. Hamdaoui and P. Ramanathan, "A dynamic priority assignement technique for streams with (m, k)-firm deadlines," *IEEE Trans. Computers*, vol. 44, no. 12, pp. 1443–1451, 1995.

[3] P. Kumar, D. Goswami, S. Chakraborty, A. Annaswamy, K. Lampka, and L. Thiele, "A hybrid approach to cyber-physical systems verification," in *Proc. of the of 49th Design Automation Conference (DAC)*, (San Francisco, USA), June 2012.

[4] C. Locke, *Best-effortDecisionMakingforReal-TimeScheduling*. PhD thesis, Carnegie-Mellon University, 1986.

[5] J.-Y. L. Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, vol. 2050 of *Lecture Notes in Computer Science*. Springer, 2001.

[6] L. Thiele, S. Chakraborty, and M. Naedele, "Real-time calculus for scheduling hard real-time systems," in *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, vol. 4, 2000.

[7] A. Burns, G. Bernat, and I. Broster, "A probabilistic framework for schedulability analysis," in *EMSOFT* (R. Alur and I. Lee, eds.), vol. 2855 of *Lecture Notes in Computer Science*, pp. 1–15, Springer, 2003.

[8] T.-S. Tia, Z. Deng, M. Shankar, M. F. Storch, J. Sun, L.-C. Wu, and J. W.-S. Liu, "Probabilistic performance guarantee for real-time tasks with varying computation times," in *IEEE Real Time Technology and Applications Symposium*, pp. 164–173, IEEE Computer Society, 1995.

[9] Y. Jiang, "A basic stochastic network calculus," in *SIGCOMM* (L. Rizzo, T. E. Anderson, and N. McKeown, eds.), pp. 123–134, ACM, 2006.

[10] A. Burns, D. Prasad, A. Bondavalli, F. D. Giandomenico, K. Ramamritham, J. A. Stankovic, and L. Strigini, "The meaning and role of value in scheduling flexible real-time systems," *Journal of Systems Architecture*, vol. 46, no. 4, pp. 305–325, 2000.

[11] G. C. Buttazzo, M. Spuri, and F. Sensini, "Value vs. deadline scheduling in overload conditions," in *IEEE Real-Time Systems Symposium*, pp. 90–99, 1995.

[12] S. K. Baruah and J. R. Haritsa, "Scheduling for overload in real-time systems," *IEEE Trans. Computers*, vol. 46, no. 9, pp. 1034–1039, 1997.

[13] R. West, Y. Zhang, K. Schwan, and C. Poellabauer, "Dynamic window-constrained scheduling of real-time streams in media servers," *IEEE Trans. Computers*, vol. 53, no. 6, pp. 744–759, 2004.

[14] G. Koren and D. Shasha, "An approach to handling overloaded systems that allow skips," in *IEEE Real-Time Systems Symposium*, pp. 110–119, 1995.

[15] G. Bernat and A. Burns, "Combining (/sub m//sup n/)-hard deadlines and dual priority scheduling," in *IEEE Real-Time Systems Symposium*, pp. 46–57, IEEE Computer Society, 1997.

[16] S. Punnekkat, R. I. Davis, and A. Burns, "Sensitivity analysis of real-time task sets," in *ASIAN* (R. K. Shyamasundar and K. Ueda, eds.), vol. 1345 of *Lecture Notes in Computer Science*, pp. 72–82, Springer, 1997.

[17] E. Bini, M. D. Natale, and G. C. Buttazzo, "Sensitivity analysis for fixed-priority real-time systems," *Real-Time Systems*, vol. 39, no. 1-3, pp. 5–30, 2008.

[18] J. P. Lehoczky and S. Ramos-Thuel, "An optimal algorithm for scheduling soft-aperiodic tasks in fixed-priority preemptive systems," in *IEEE Real-Time Systems Symposium*, pp. 110–123, IEEE Computer Society, 1992.

[19] M. Spuri and G. C. Buttazzo, "Scheduling aperiodic tasks in dynamic priority systems," *Real-Time Systems*, vol. 10, no. 2, pp. 179–210, 1996.

[20] C. Lu, J. A. Stankovic, S. H. Son, and G. Tao, "Feedback control real-time scheduling: Framework, modeling, and algorithms," *Real-Time Systems*, vol. 23, no. 1-2, pp. 85–126, 2002.

[21] V. Firoiu, J.-Y. Le Boudec, D. Towsley, and Z.-L. Zhang, "Theories and models for internet quality of service," *Proceedings of the IEEE*, vol. 90, pp. 1565 – 1591, sep 2002.

[22] L. Thiele, "Modular performance analysis of distributed embedded systems," in *FORMATS*, vol. 3829 of *Lecture Notes in Computer Science*, p. 1, Springer, 2005.

[23] E. Wandeler, *Modular Performance Analysis and Interface-Based Design for Embedded Real-Time Systems*. PhD thesis, ETH Zurich, 2006.

[24] S. K. Baruah, A. K. Mok, and L. E. Rosier, "Preemptively scheduling hard-real-time sporadic tasks on one processor," in *IEEE Real-Time Systems Symposium*, 1990.

[25] 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.

[26] G. C. Buttazzo, "Rate monotonic vs. edf: Judgment day," *Real-Time Systems*, vol. 29, no. 1, pp. 5–26, 2005.

APPENDIX

Proof of Theorem 1: Let the demand overflow rare event start at some time τ. Let R and R_{RE} denote the

arrival function due to the nominal and rare events models, respectively. Then, the total arrival curve $\widehat{R}$ is given as

$$\widehat{R} = R + R_{\mathtt{RE}} \tag{21}$$

Since R and $R_{\mathtt{RE}}$ satisfy the respective arrival curves α and $\alpha_{\mathtt{RE}}$, we have

$$R(v) - R(u) \leq \alpha(v-u), \quad \forall\ 0 \leq u \leq v. \tag{22}$$

$$R_{\mathtt{RE}}(v) - R_{\mathtt{RE}}(u) \leq \alpha_{\mathtt{RE}}(v-u), \quad \forall\ 0 \leq \tau \leq u \leq v \leq \tau + l_{\mathtt{RE}}. \tag{23}$$

To maximise the sum, $\widehat{R}$, we see that we need to set $\tau = 0$. We thus get,

$$\widehat{R}(v) - \widehat{R}(u) \leq \alpha(v-u) + \alpha_{\mathtt{RE}}(v-u),\ \forall\ 0 \leq u \leq v. \tag{24}$$

Hence, we can represent the above condition by stating that the arrival curve of the task is given by

$$\widehat{\alpha} = \alpha + \alpha_{\mathtt{RE}} \tag{25}$$

■

Proof of Theorem 2: Let the supply shortage rare event start at some time τ. Let C and $C_{\mathtt{RE}}$ denote the service functions due to the nominal and supply shortage rare events, respectively. Then the total service function $\widehat{C}$ is given as

$$\widehat{C} = \{C - C_{\mathtt{RE}}\}^{\uparrow}. \tag{26}$$

Since C and $C_{\mathtt{RE}}$ satisfy bounds in the interval domain given by β and $\beta_{\mathtt{RE}}$, respectively, we have

$$C(v) - C(u) \geq \beta(v-u), \quad \forall\ 0 \leq u \leq v, \tag{27}$$

$$C_{\mathtt{RE}}(v) - C_{\mathtt{RE}}(u) \leq \beta_{\mathtt{RE}}(v-u), \quad \forall\ 0 \leq \tau \leq u \leq v \leq \tau + l_{\mathtt{RE}}. \tag{28}$$

From the above conditions, it follows that

$$\widehat{C}(v) - \widehat{C}(u) \geq \{\beta - \beta_{\mathtt{RE}}\}^{\uparrow}(v-u),\ \forall\ 0 \leq u \leq v. \tag{29}$$

Hence, we can represent the above condition by stating that the resource provides a service curve $\widehat{\beta}$ given as

$$\widehat{\beta} = \{\beta - \beta_{\mathtt{RE}}\}^{\uparrow}. \tag{30}$$

■

Proof of Theorem 3: We consider both types of rare events, namely demand overflow and supply shortage rare events. As discussed earlier, these exceptions can be both represented by the arrival and service curves $\widehat{\alpha}$ and $\widehat{\beta}$, respectively.

We prove the result by contradiction. Let a rare event occur at some time τ. Let $\widehat{R}(t)$ and $\widehat{C}(t)$ denote any arrival and service functions, respectively. Let some job miss its deadline at some time $v > \tau + \mathtt{t_s}$, where $\mathtt{t_s}$ is computed as in (13). Let u be the latest time before v when the task queue was empty.

Case (a): $u > \tau$. After the occurrence of the rare event, we know that in any interval $[u, t]$ the execution demand is bounded by the standard (without rare events) arrival curve α and the execution supply of the resource is bounded by the standard service curve β. Since, we know that in the system without rare events every job meets its deadline, every job will continue to meet its deadline after u. Hence the assumption that the job finishing at $v > u$ misses its deadline is a contradiction.

Case (b): $u \leq \tau$. Since delay of job finishing at time v is larger than D and $[u, v]$ is a busy interval, we know that

$$\begin{aligned} \widehat{C}(v) - \widehat{C}(u) &< \widehat{R}(v-D) - \widehat{R}(u) \\ \Rightarrow \widehat{\beta}(v-u) &< \widehat{\alpha}(v-u-D). \end{aligned} \tag{31}$$

With $v - u > \mathtt{t_s}$, this contradicts the definition of $\mathtt{t_s}$ in (12).

Tightness We now show that the above bound is a tight bound by providing an example of valid arrival and service functions with the above bound as its settling-time. Consider a trace where the rare-event occurs at time 0, i.e., the exceptional job arrives at time 0. Let the trace of execution demand be given by

$$\widehat{R}(t) = \widehat{\alpha}(t). \tag{32}$$

Let the trace of the execution supply be given by

$$\widehat{C}(t) = \widehat{\beta}(t). \tag{33}$$

Consider the job finishing at time $\mathtt{t_s}$. A lower-bound on the delay suffered by this job denoted as δ is given by

$$\delta = \inf\{\Delta\ :\ C(\mathtt{t_s}) \geq R(\mathtt{t_s} - \Delta)\}. \tag{34}$$

From (12), we know that this value of δ is larger than D. Hence, the settling-time for this trace is not smaller than $\mathtt{t_s}$.

■

Proof of Theorem 4: From results in Network Calculus, the worst-case delay given arrival and service curves is given as $\mathtt{Del}(\alpha, \beta)$. We have shown in Theorems 1 and 2, that the effective arrival and service curves in the presence of rare events is given by $\widehat{\alpha}$ and $\widehat{\beta}$ as computed in (6) and (11), respectively. ■

Proof of Theorem 5: The theorem concerns the critical arrival and service functions that lead to the largest number of jobs missing the deadlines. Both of these functions are variable subject to interval bounds by the respective arrival and service curves. In the proof, we argue about the computation of the critical arrival function for a fixed service function equal to the service curve. A similar argument applies to the computation of the critical service curve.

Consider some arrival function $\widehat{R}$ and the service function $\widehat{C} = \widehat{\beta}$. Let more than $\widehat{N}$ jobs miss their deadlines for this arrival function. We use the following monotonicity principle to modify $\widehat{R}$: "An earlier arrival time of any of the jobs cannot lead to a smaller worst-case delay". This can be used to iteratively decrease the arrival time of the jobs such that

they conform to the arrival pattern. Since the delay of the jobs are decreasing during these modifications, there exists no arrival function $\widehat{R}$ with greater number of jobs missing deadlines than $\widehat{N}$. ■

Proof of Theorem 6: We will compute the settling-time of task T_a for some a. To do this, we will identify it as a single task system with a suitable service curve and then apply Theorem 3.

From Modular Performance Analysis (MPA) [23], the service curve available to a task T_a, denoted β_a, in a fixed priority setting, in terms of overall service curve β and the set of arrival curves $\{\alpha_i\}$ is

$$\beta_a(\beta, \{\alpha_i\}) = (\beta - \sum_{i<a} \alpha_i)^{\uparrow}. \tag{35}$$

Case (a): Supply shortage exception on the resource According to (11), the overall service is bounded by $\widehat{\beta} = \{\beta - \beta_{\mathtt{RE}}\}^{\uparrow}$. Then, it can be shown that

$$\beta_a(\{\beta - \beta_{\mathtt{RE}}\}^{\uparrow}, \{\alpha_i\}) = \{\beta_a(\beta, \{\alpha_i\}) - \beta_{\mathtt{RE}}\}^{\uparrow}. \tag{36}$$

In other words, the supply shortage on the resource is equivalent, in terms of bounding the settling-time, to the same supply shortage on the single-task system serving T_a. Hence, the settling-time result follows from Theorem 3.

Case (b): Demand overflow on task T_b with $b > a$ In this case, the settling-time of the task T_a is trivially 0, as the higher priority jobs are not affected by the timing properties of lower priority jobs.

Case (c): Demand overflow on task T_a This follows from Theorem 3, as the service curve available to task T_a is given as in (35), and the exceptional arrival curve of T_a is used in (19).

Case (d): Demand overflow on task T_b with $b < a$ In the case of a demand overflow on a higher priority task, the analysis is equivalent to considering that the demand overflow as an equivalent supply shortage on the resource serving task T_b.

Tightness The tightness of the result is shown by providing a valid trace, for which a job of task T_i misses its deadline up to the end of the settling-time window. In all the above cases the trace to consider is given as follows:

$$\begin{aligned} \widehat{R}_i(t) &= \widehat{\alpha}_i(t), \forall\ i, \\ \widehat{C}(t) &= \widehat{\beta}. \end{aligned} \tag{37}$$

■

Proof of Theorem 7: The proof is similar in form to the proof of Theorem 3. We first define the following notion of demand bound function under rare events [24].

$$\widehat{\mathtt{dbf}}(\Delta) = \sum_i \left(\widehat{\alpha}_i(\Delta - D_i)\right). \tag{38}$$

We prove the theorem by contradiction. We assume that a rare event occurs at time t^* and deadline is missed at some time $v > t^* + \mathtt{t_s}$. Let u be the latest time before v when there are no pending jobs across all tasks.

Case (a): Demand overflow If $u > t^*$, we know that there are no pending jobs at u and no rare events after u. Hence, according to the system model, there should be no deadline misses. If $u \leq t^*$, then we know that the maximum demand of all jobs having arrival times and deadlines within $[u, v]$ is bounded by $\widehat{\mathtt{dbf}}(v-u)$. If some task is to miss its deadline at v, then this demand bound is not served by the resource. But we know that $\widehat{\beta}(v-u) > \widehat{\mathtt{dbf}}(v-u)$ as $v - u > \mathtt{t_s}$. Hence, we have a contradiction.

Case (b): Supply shortage Define w as follows

$$w = t^* + \sup\{\Delta \mid (\beta - \beta_{\mathtt{RE}})^{\uparrow}(\Delta) < 0\}. \tag{39}$$

If $u < w$, then in the interval $[u, v]$ the service provided is not less than $\widehat{\beta}(v - t^*)$. The execution demand of all jobs with arrival times and deadlines within $[u, v]$ cannot be larger than $\widehat{\mathtt{dbf}}(v-u)$, which in turn is not larger than $\widehat{\beta}(v-t^*)$ as $v - u < v - t^* < \mathtt{t_s}$. If $u > w$, then in any interval after u the arrival and service curves are nominal values (without exceptions) and hence no job can miss its deadline. Thus, we have a contradiction.

Tightness The tightness of the result is shown by providing a valid trace, for which a job of task T_i misses its deadline up to the end of the settling-time window. In all the above cases the trace to consider is given as in (37). ■

Proof of Theorem 8: A scheduling policy that is optimal under REST model, is optimal for or any given nominal task model, resource model, deadline and rare event model. Trivially, if we set the rare event to have no execution time, minimisation of settling-time is equivalent to guaranteeing schedulability. ■

Proof of Theorem 9: Consider the trace of the system, as given in (37) with the worst-case settling-time under EDF. Let $(\mathtt{t_s})_{\mathtt{EDF}}$ denote this settling-time under EDF. The execution demand of all jobs arriving and having deadlines in the interval $[0, (\mathtt{t_s})_{\mathtt{EDF}}]$ is given by $\widehat{\mathtt{dbf}}((\mathtt{t_s})_{\mathtt{EDF}})$ which exceeds the available service in that interval, according to (20). Thus, no scheduling policy can have a lower setting-time than $(\mathtt{t_s})_{\mathtt{EDF}}$.

■

Proof of Theorem 10: This follows directly from Example 9. ■