Schedulability analysis of synchronization protocols based on overrun without payback for hierarchical scheduling frameworks revisited

Schedulability analysis of synchronization protocols based on overrun without payback for hierarchical scheduling frameworks revisited*

Reinder J. Bril, Uğur Keskin
Technische Universiteit Eindhoven (TU/e)
Den Dolech 2, 5612 AZ Eindhoven, The Netherlands
R.J.Bril@TUe.nl

Moris Behnam, Thomas Nolte
Mälardalen Real-Time Research Centre
P.O. Box 883, SE-721 23 Västerås,
Sweden

Abstract

In this paper, we revisit global as well as local schedulability analysis of synchronization protocols based on the stack resource policy (SRP) and overrun without payback for hierarchical scheduling frameworks based on fixed-priority pre-emptive scheduling (FPPS). We show that both the existing global and local schedulability analysis are pessimistic, present improved analysis, and illustrate the improvements by means of examples.

1 Introduction

1.1 Background

The Hierarchical Scheduling Framework (HSF) has been introduced to support hierarchical CPU sharing among applications under different scheduling services [2]. The HSF can be generally represented as a tree of nodes, where each node represents an application with its own scheduler for scheduling internal workloads (e.g. tasks), and resources are allocated from a parent node to its children nodes.

The HSF provides means for decomposing a complex system into well-defined parts called *subsystems*, which may share (so-called global) logical resources requiring mutual exclusive access. In essence, the HSF provides a mechanism for timing-predictable *composition* of course-grained subsystems. In the HSF a subsystem provides an introspective *interface* that precisely specifies the timing properties of the subsystem. This means that subsystems can be independently developed, analyzed and tested, and later assembled without introducing unwanted temporal interference. Temporal isolation between subsystems is provided through budgets which are allocated to subsystems.

As large extents of embedded systems are resource constrained, a tight analysis is instrumental in a successful deployment of HSF techniques in real applications. We therefore aim at reducing potential pessimism in existing schedulability analysis for HSFs. Looking further at existing industrial real-time systems, fixed priority pre-emptive scheduling (FPPS) is the de facto standard of task scheduling, hence we focus on an HSF with support for FPPS in the scheduling of tasks within a subsystem. Having such support will simplify migration to the HSF and integration of existing legacy applications into the HSF, avoiding a too big technology revolution for engineers.

Our current research efforts are directed towards the conception and realization of a two-level HSF that is based on (*i*) FPPS for both *global scheduling* of budgets (allocated to subsystems) and *local scheduling* of tasks (within a subsystem), (*ii*) the periodic resource model [2] for budgets, and (*iii*) the stack resource policy (SRP) [3] for both inter- and intra-subsystem resource sharing. For such an HSF, two mechanisms have been studied that prevent depletion of a budget during global resource access, i.e. *skipping* [4] and *overrun* [5]. Skipping prevents depletion by checking the remaining budget before granting resource access, and delaying access to a next budget period when the remaining budget is insufficient. Overrun prevents depletion by temporarily increasing the budget with a statically determined amount for the duration of that access. The overrun mechanism comes in two flavors, i.e. *with payback* and *without payback*, which determine whether or not the additional amount of budget has to be paid back during the next budget period.

*This paper rectifies and extends [1]; see Appendix A.

1.2 Contributions

We show that existing global and local schedulability analysis of synchronization protocols based on SRP and overrun without payback for two-level hierarchical scheduling based on FPPS is pessimistic. One of the causes of the pessimism in the global analysis is that during an overrun, as a resource is locked, not all higher priority subsystems are able to preempt. Taking this into account reduces the amount of interference considered due to higher priority subsystems. We present improved global and local analysis assuming that the deadline of a subsystem holds for the sum of its normal budget and its overrun budget. We illustrate the improvements by means of examples, and show that the improved global analysis is both uniform and sustainable. We briefly discuss further options for improvements.

1.3 Overview

This paper has the following structure. In Section 2 we present related work. A real-time scheduling model is the topic of Section 3. The existing global and local schedulability analysis is recapitulated in Section 4, and improved global and local analysis is presented in Sections 5 and 6, respectively. Options for further improvements are briefly sketched in Section 7. The paper is concluded in Section 8.

2 Related work

There has been a growing attention to hierarchical scheduling of real-time systems [6, 7, 8, 9, 2]. Deng and Liu [6] proposed a two-level HSF for open systems, where subsystems may be developed and validated independently. Kuo and Li [8] and Lipari and Baruah [9] presented schedulability analysis techniques for such a two-level framework with the FPPS global scheduler and the Earliest Deadline First (EDF) global scheduler, respectively. Shin and Lee [2] proposed the periodic resource model $\Gamma(\Pi,\Theta)$ to specify guaranteed periodic CPU allocations, where $\Pi \in \mathbb{R}^+$ is a period and $\Theta \in \mathbb{R}^+$ is a periodic allocation time ($0 < \Theta \leq \Pi$). Easwaran, Aland, and Lee [10] proposed the explicit deadline periodic (EDP) resource model $\Omega(\Pi,\Theta,\Delta)$ that extends the periodic resource model by explicitly distinguishing a relative deadline $\Delta \in \mathbb{R}^+$ for the allocation time Θ ($0 < \Theta \leq \Delta \leq \Pi$).

For synchronization protocols in HSFs, two mechanisms have been studied to prevent depletion of a budget during global resource access, i.e. *overrun* (*with payback* and *without payback*) and *skipping*. Overrun with payback was first introduced in the context of aperiodic servers in [11]. The mechanism was later (re-) used for a synchronization protocol in the context of two-level hierarchical scheduling in [12] and extended with overrun without payback. The analysis presented in [12] does not allow analysis of individual subsystems, however. Analysis supporting composability was first described in [13, 14]. The idea of skipping was first described in the skip protocol SP [15] used in a pfair-scheduling environment. In the context of HSFs, the SIRAP protocol [4] is based on skipping, and its associated analysis supports composability. A comparative evaluation of both depletion prevention mechanisms was presented in [16]. The results showed that the performance of these mechanisms is heavily depending on the system's parameters, such as the subsystem period, the worst case execution time inside a critical section, tasks period, and task set utilization.

3 Real-time scheduling model

We consider a two-level hierarchical FPPS model using the periodic resource model to specify guaranteed CPU allocations to tasks of subsystems and using a synchronization protocol for mutual exclusive resource access to global logical resources based on SRP[1] and overrun without payback.

3.1 System model

A system *Sys* contains a set $\mathcal{R}$ of M global logical resources $R_1, R_2, \ldots, R_M$, a set $\mathcal{S}$ of N subsystems $S_1, S_2, \ldots, S_N$, a set $\mathcal{B}$ of N budgets for which we assume a periodic resource model [2], and a single processor. Each subsystem S_s has a dedicated budget associated to it. In the remainder of this paper, we leave budgets implicit, i.e. the timing characteristics of budgets are taken care of in the description of subsystems. Subsystems are scheduled by means of FPPS and have fixed, unique priorities. For notational convenience, we assume that subsystems are given in order of decreasing priorities, i.e. S_1 has highest priority and S_N has lowest priority.

[1] The focus of this paper is on synchronization protocols for *global* logical resources. We therefore do not consider local logical resources.

3.2 Subsystem model

Each subsystem S_s contains a set $\mathcal{T}_s$ of n_s periodic tasks $\tau_1, \tau_2, \ldots, \tau_{n_s}$ with fixed, unique priorities, which are scheduled by means of FPPS. For notational convenience, we assume that tasks are given in order of decreasing priorities, i.e. τ_1 has highest priority and τ_{n_s} has lowest priority. The set $\mathcal{R}_s$ denotes the subset of M_s global resources accessed by subsystem S_s. The maximum time that a subsystem S_s executes while accessing resource $R_l \in \mathcal{R}$ is denoted by X_{sl}, where $X_{sl} \in \mathbb{R}^+ \cup \{0\}$ and $X_{sl} > 0 \Leftrightarrow R_l \in \mathcal{R}_s$. The timing characteristics of S_s are specified by means of a triple $< P_s, Q_s, \mathcal{X}_s >$, where $P_s \in \mathbb{R}^+$ denotes its (budget) period, $Q_s \in \mathbb{R}^+$ its (normal) budget, and $\mathcal{X}_s$ the set of maximum execution access times of S_s to global resources. The maximum value in $\mathcal{X}_s$ (or zero when $\mathcal{X}_s = \emptyset$) is denoted by X_s, i.e.

$$X_s = \max\{X_{sl} | R_l \in \mathcal{R}\}. \tag{1}$$

The overrun budget of S_s is equal to X_s and also denoted by X_s. Note that we assume the (relative) deadline $D_s \in \mathbb{R}^+$ of subsystem S_s to be equal to its period P_s, i.e. $D_s = P_s$. A release of (the budget of) a subsystem is also called a *job*.

3.3 Task model

The timing characteristics of a task $\tau_{si} \in \mathcal{T}_s$ are specified by means of a quartet $< T_{si}, C_{si}, D_{si}, \mathcal{C}_{si} >$, where $T_{si} \in \mathbb{R}^+$ denotes its minimum inter-arrival time, $C_{si} \in \mathbb{R}^+$ its worst-case computation time, $D_{si} \in \mathbb{R}^+$ its (relative) deadline, $\mathcal{C}_{si}$ a set of maximum execution times of τ_{si} to global resources, where $C_{si} \leq D_{si} \leq T_{si}$. The set $\mathcal{R}_{si}$ denotes the subset of $\mathcal{R}_s$ accessed by task τ_{si}. The maximum time that a task τ_{si} executes while accessing resource $R_l \in \mathcal{R}$ is denoted by c_{sil}, where $c_{sil} \in \mathbb{R}^+ \cup \{0\}$, $C_{si} \geq c_{sil}$, and $c_{sil} > 0 \Leftrightarrow R_l \in \mathcal{R}_{si}$.[2]

3.4 Resource model

The *CPU supply* refers to the amount of CPU allocation that a virtual processor can provide. The supply bound function $\mathtt{sbf}_\Omega(t)$ of the EDP resource model $\Omega(\Pi, \Theta, \Delta)$ that computes the minimum possible CPU supply for every interval length t is given by

$$\mathtt{sbf}_\Omega(t) = \begin{cases} t - (k+1)(\Pi - \Theta) + (\Pi - \Delta) & \text{if } t \in V^{(k)} \\ (k-1)\Theta & \text{otherwise,} \end{cases} \tag{2}$$

where $k = \max\left(\left\lceil \left(t - (\Delta - \Theta)\right)/\Pi \right\rceil, 1\right)$ and $V^{(k)}$ denotes an interval $[k\Pi + \Delta - 2\Theta, k\Pi + \Delta - \Theta]$.

The supply bound function $\mathtt{sbf}_\Gamma(t)$ of the periodic resource model $\Gamma(\Pi, \Theta)$ is a special case of (2), i.e. with $\Delta = \Pi$.

3.5 Synchronization protocol

Overrun without payback prevents depletion of a budget of a subsystem S_s during access to a global resource R_l by temporarily increasing the budget of S_s with X_{sl}, the maximum time that S_s executes while accessing R_l. To be able to use SRP in an HSF for synchronizing global resources, its associated ceiling terms needs to be extended.

3.5.1 Resource ceiling

With every global resource R_l, two types of resource ceilings are associated; an *external* resource ceiling RC_l for global scheduling and an *internal* resource ceiling rc_{sl} for local scheduling. According to SRP, these ceilings are defined as

$$RC_l = \min(N, \min\{s \mid X_{sl} > 0\}), \tag{3}$$

$$rc_{sl} = \min(n_s, \min\{i \mid c_{sil} > 0\}). \tag{4}$$

Note that we use the outermost min in (3) and (4) to define RC_l and rc_{sl} also in those situations where no subsystem uses R_l and no task of $\mathcal{T}_s$ uses R_l, respectively.

[2] In [12], it is required that $c_{sil} < C_{si}$ and $c_{sil} < Q_s$. Moreover, it is observed that c_{sil} will typically be much smaller than both C_{si} and Q_s.

3.5.2 System/subsystem ceiling

The system/subsystem ceilings are dynamic parameters that change during the execution. The system/subsystem ceiling is equal to the highest external/internal resource ceiling of a currently locked resource in the system/subsystem. Note that because resource ceilings correspond to priorities, the highest resource ceiling has the lowest value.

Under SRP, a task τ_{si} can only preempt the currently executing task τ_{sj} (even when accessing a global resource) if the priority of τ_{si} is greater (i.e. the index i is lower) than S_s its subsystem ceiling. A similar condition for preemption holds for subsystems.

3.5.3 Concluding remarks

The maximum time X_{sl} that S_s executes while accessing R_l can be reduced by assigning a value to rc_{sl} that is *smaller* than the value according to SRP. For HSRP [12], the internal resource ceiling is therefore set to the highest priority, i.e. $rc_{sl}^{\mathrm{HSRP}} = 1$. Decreasing rc_{sl} may cause a subsystem to become unfeasible for a given budget [17], however, because the tasks with a priority higher than the old ceiling and at most equal to the new ceiling may no longer be feasible.

The results in this paper apply for any internal resource ceiling rc'_{sl} where $rc_{sl} \geq rc'_{sl} \geq rc_{sl}^{\mathrm{HSRP}} = 1$.[3]

4 Recap of existing schedulability analysis

In this section, we briefly recapitulate the global schedulability analysis presented in [12] and the local schedulability analysis described in [16, 5]. Although the global schedulability analysis presented in [16, 5] looks different, it is based on the analysis described in [12] and therefore yields the same result.

For illustration purposes, we will use an example system Sys_1 containing two subsystems S_1 and S_2 sharing a global resource R_1. The characteristics of the subsystems are given in Table 1.

subsystem	P_s	$Q_s + X_s$
S_1	5	2
S_2	7	$Q_2 + X_2$

Table 1. Subsystem characteristics of Sys_1.

4.1 Global analysis

The worst-case response time WR_s of subsystem S_s is given by the smallest $x \in \mathbb{R}^+$ satisfying

$$x = B_s + (Q_s + X_s) + \sum_{t<s} \left\lceil \frac{x}{P_t} \right\rceil (Q_t + X_t), \tag{5}$$

where B_s is the maximum blocking time of S_s by lower priority subsystems, i.e.

$$B_s = \max(0, \max\{X_{tl} \mid t > s \wedge X_{tl} > 0 \wedge RC_l \leq s\}). \tag{6}$$

Note that we use the outermost max in (6) to define B_s also in those situations where the set of values of the innermost max is empty. To calculate WR_s, we can use an iterative procedure based on recurrence relationships, starting with a lower bound, e.g. $B_s + \sum_{t\leq s}(Q_t + X_t)$. The condition for global schedulability is given by

$$\forall_{1\leq s\leq N} WR_s \leq P_s. \tag{7}$$

We merely observe that the global analysis is similar to basic analysis for FPPS with resource sharing, where the period P_s of a subsystem S_s serves as deadline for the sum of the normal budget Q_s and the overrun budget X_s, and the interference of higher priority subsystems S_t is based on the sum $Q_t + X_t$. We will therefore use a superscript P to refer to this basic analysis for subsystems, e.g. WR_s^{P}.

[3] Because $rc_{sl}^{\mathrm{HSRP}} = 1$ for $R_l \in \mathcal{R}_s$, $X_{sl} = \max_i c_{sil}$. Hence, from $c_{sil} < Q_s$ we derive $X_s < Q_s$. Without the constraint on the internal resource ceiling, X_s may be larger than Q_s. For illustration purposes, we also allow $X_s > Q_s$ in this paper.

In the sequel, we are not only interested in the worst-case response time of a subsystem S_s for particular values of B_s, Q_s, and X_s, but in the value as a function of the sum of these three values. We will therefore use a functional notation when needed, e.g. $WR_s(B_s + Q_s + X_s)$.

The global feasibility area of the existing analysis is illustrated for our example system Sys_1 in Figure 1. Note that the y-axis is excluded, because we assume the capacity of subsystems to be positive, i.e. $Q_2 > 0$.

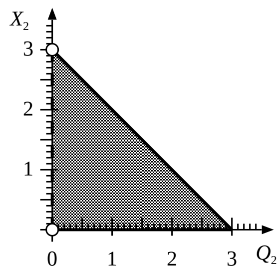

Figure 1. Global feasibility area assuming FPPS.

Figure 2 shows a timeline with a simultaneous activation of S_1 and S_2 for $Q_2 = 3.0$ and $X_2 = 0$, and a worst-case response time WR_2 of S_2 equal to 5.0. Note that even an infinitesimal increase of either Q_1 or Q_2 will make the system Sys_1 unschedulable.

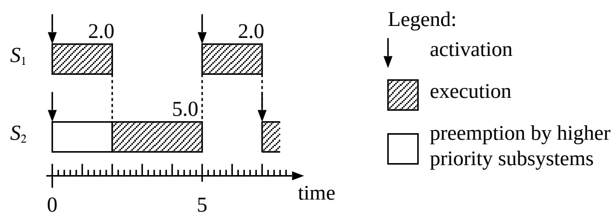

Figure 2. Timeline for $Q_2 = 3.0$ and $X_2 = 0$ under FPPS.

4.2 Local analysis

The existing condition for local schedulability of a subsystem S_s [5] is given by

$$\forall_{1 \leq i \leq n_s} \exists_{0 < t \leq D_{si}} b_{si} + C_{si} + \sum_{j<i} \left\lceil \frac{t}{T_{sj}} \right\rceil \cdot C_{sj} \leq \mathtt{sbf}_{\Gamma_s}(t), \tag{8}$$

where b_{si} is the maximum blocking time of τ_{si} by lower priority tasks, i.e.

$$b_{si} = \max(0, \max\{c_{sjl} \mid j > i \wedge c_{sjl} > 0 \wedge rc_{sl} \leq i\}), \tag{9}$$

and $\mathtt{sbf}_{\Gamma_s}(t)$ is the supply bound function of the periodic resource model $\Gamma_s(P_s, Q_s)$ for the subsystem S_s under consideration. Note that we use the outermost max in (9) to define b_{si} also in those situations where the set of values of the innermost max is empty.

The value for X_{sl} depends on the local scheduler and the synchronization protocol. The maximum time that subsystem S_s executes while task τ_{sil} accesses resource $R_l \in \mathcal{R}$ is denoted by X_{sil}, where $X_{sil} \in \mathbb{R}^+ \cup \{0\}$ and $X_{sil} > 0 \Leftrightarrow c_{sil} > 0$. For $c_{sil} > 0$, X_{sil} is given by [5]

$$X_{sil} = c_{sil} + \sum_{j<rc_{sl}} C_{sj}. \tag{10}$$

The value for X_{sl} is given by

$$X_{sl} = \max_{1 \leq i \leq n_s} X_{sil}. \tag{11}$$

5 Improved global analysis

As described in Section 4.1, the existing global schedulability analysis is based on FPPS, where the period P_s serves as deadline for the sum of the normal budget Q_s and overrun budget X_s.

5.1 Illustrating the improvement

In this section, we will present two steps that gradually improve the global analysis:

1. *Limited pre-emption of overrun budget* X_s;
2. *Blocking starts* before *the execution based on the overrun budget* X_s *starts*;

5.1.1 Limited pre-emption of overrun budget

Subsystem S_1 can not preempt S_2 during those intervals of time when S_2 is accessing resource R_1 in general, and when S_2 is executing based on its overrun budget X_2 in particular. This limited preempt-ability of subsystem S_2 gives rise to improved schedulability of system Sys_1, as illustrated in Figure 3. In this figure, it is assumed that X_2 can be executed without pre-emption. Note that $X_2 \leq 3.0$ and $Q_2 \leq 3.0$, because S_1 and S_2 will otherwise miss their deadline, respectively. Further note

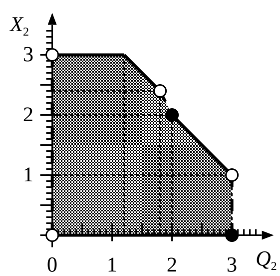

Figure 3. Global feasibility area assuming limited pre-emption of X_s.

that for $Q_2 = 1.2$ and $X_2 = 3.0$ the utilization of the system $U = \frac{Q_1+X_1}{P_1} + \frac{Q_2+X_2}{P_2} = 1$. Finally note that the feasibility area shown in Figure 3 would be identical when the global schedulability analysis would be based on fixed-priority scheduling with deferred pre-emption (FPDS) [18, 19], and each job of S_2 would consist of a sequence of two non-preemptable subjobs with computation times Q_2 and X_2, respectively.

We will briefly explain the anomalies in Figure 3 by means of timelines with a simultaneous release of S_1 and S_2 at time $t = 0$ and assuming that both S_1 and S_2 need their overrun budget for every activation.

Figure 4 shows a timeline with $Q_2 = 1.8$ and $X_2 = 2.4$. Note that the second job of S_2 misses its deadline at time $t = 14$, because the third job of S_1 is allowed to start at time $t = 10$. An infinitesimal decrease of either Q_2 or X_2 will allow the execution of X_2 of the second job to start just before $t = 10$ and will allow the second job to meet its deadline.

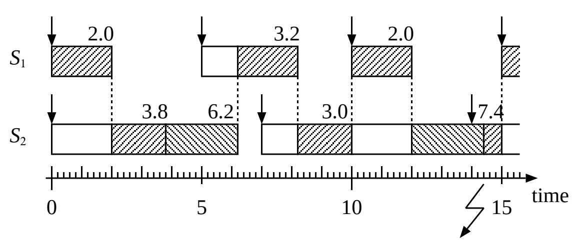

Figure 4. Timeline for $Q_2 = 1.8$ and $X_2 = 2.4$ under limited pre-emption of X_2 with a deadline miss at $t = 14$. The numbers to the top right corner of the boxes denote the response times (of the normal budget or the combination of normal and overrun budget) of the respective releases.

Figure 5 shows a timeline with $Q_2 = 2.0$ and $X_2 = 2.0$. In this case, the second job of S_2 meets its deadline, because the workload in the interval $[0,14)$ is equal to the length of that interval. Note that the configurations of S_2 represented by the line segment of the line $2Q_2 + X_2 = 6.0$ between the points $< 1.8, 2.4 >$ and $< 2.0, 2.0 >$ are not feasible.

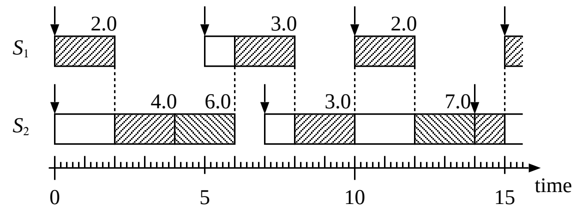

Figure 5. Timeline for $Q_2 = 2.0$ and $X_2 = 2.0$ under limited pre-emption of X_2.

Figure 6 shows a timeline with $Q_2 = 3.0$ and $X_2 = 1.0$. In this case, the first job of S_2 misses its deadline. Although an infinitesimal decrease of Q_2 will allow S_2 to meet its deadline, S_2 is only schedulable for $Q_2 = 3.0$ when $X_2 = 0$.

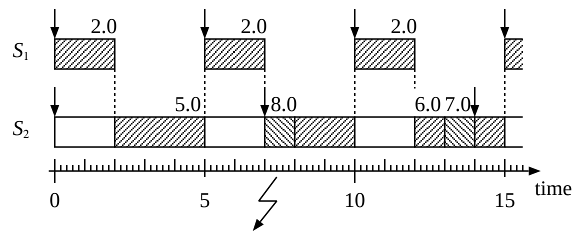

Figure 6. Timeline for $Q_2 = 3.0$ and $X_2 = 1.0$ under limited pre-emption of X_2 with a deadline miss at $t = 7$.

5.1.2 Blocking starts before overrun

Whenever S_2 uses its overrun budget X_2, it must lock R_1 already during the consumption of its normal budget Q_2, i.e. *before* it starts consuming its overrun budget X_2. Hence, the system ceiling is already set to the priority of S_1 before S_2 starts consuming X_2, preventing S_1 to preempt. The resulting improvement is illustrated in Figure 7. Note that the configurations of S_2 represented by the line segment of the line $2Q_2 + X_2 = 6.0$ starting at $< 1.8, 2.4 >$ till point $< 2.0, 2.0 >$ are now feasible. Similarly, the configurations of S_2 represented by $Q_2 = 3.0$ and $0 \leq X_2 \leq 1.0$ are feasible as well. We will briefly explain the

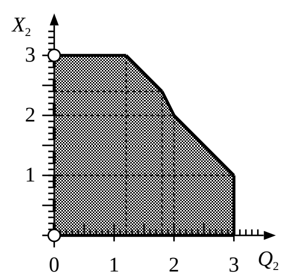

Figure 7. Global feasibility area assuming blocking starts before overrun.

differences between Figures 3 and 7 by means of timelines.

Figure 8 shows a timeline with $Q_2 = 1.8$ and $X_2 = 2.4$. Because the second job of S_2 locks R_1 just before the activation of S_1 at $t = 10$, S_2 is allowed to execute X_2 at $t = 10$. As a result, the second job of S_2 does not miss its deadline at time $t = 14$.

Figure 9 shows a timeline with $Q_2 = 3.0$ and $X_2 = 1.0$. Similar to the previous case, because the first job of S_2 locks R_1 just before the activation of S_1 at $t = 5$, S_2 is allowed to execute X_2 at $t = 5$. As a result, the first job of S_2 does not miss its deadline at time $t = 7$.

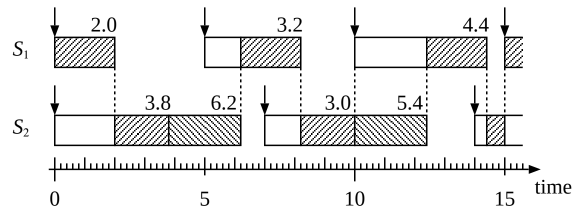

Figure 8. Timeline for $Q_2 = 1.8$ and $X_2 = 2.4$ assuming blocking starts before overrun.

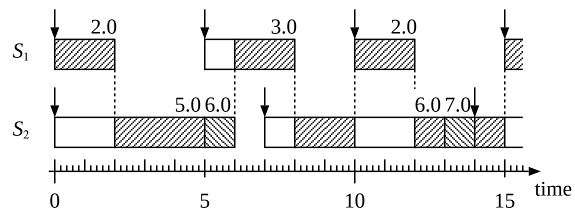

Figure 9. Timeline for $Q_2 = 3.0$ and $X_2 = 1.0$ assuming blocking starts before overrun.

5.2 Improving the global analysis

The improved global analysis is similar to the analysis for FPDS [18, 19] and FPPS with preemption thresholds [20] in the sense that we have to consider all jobs in a so-called level-s active period to determine the worst-case response time WR_s of subsystem S_s. Unlike the analysis described in [18, 19, 20], subsystems S_{s-1} till S_{RC_l} cannot preempt S_s at the finalization time of Q_s when S_s is accessing R_l, as illustrated in Figures 8 and 9 for the times $t = 10$ and $t = 5$, respectively.

In the remainder of this section, we first present the analysis for the special case where every subsystem accesses at most one global resource, i.e. $M_s \leq 1$, and subsequently present the general case.

5.2.1 Access to a single global resource

We first recapitulate the notion of a level-s active period. Next, we derive analysis for the worst-case finalization time WF^{Q}_{sk} of the normal budget Q_s of job ι_{sk} of subsystem S_s relative to the start of the constituting level-s active period. Finally, we derive analysis for the worst-case response time WR_s of S_s.

The worst-case length WL_s of a level-s active period with $s \leq N$ is given by the smallest $x \in \mathbb{R}^+$ that satisfies

$$x = B_s + \sum_{t \leq s} \left\lceil \frac{x}{P_t} \right\rceil (Q_t + X_t). \tag{12}$$

To calculate WL_s, we can use an iterative procedure based on recurrence relationships, starting with a lower bound, e.g. $B_s + \sum_{t \leq s}(Q_t + X_t)$. The maximum number wl_s of jobs of S_s in a level-s active period is given by

$$wl_s = \left\lceil \frac{WL_s}{P_s} \right\rceil. \tag{13}$$

For a job ι_{sk} of S_s with $0 \leq k < wl_s$, we split the interval from the start of the level-s active period to the finalization of job ι_{sk} in two sub-intervals: a first sub-interval including the execution of the normal budget Q_s by job ι_{sk} and a second sub-interval from the finalization of Q_s by ι_{sk} till the finalization of ι_{sk}, i.e. constituting the execution of the overrun budget X_s.

Let WF^{Q}_{sk} denote the worst-case finalization time of the normal budget Q_s of job ι_{sk} with $0 \leq k < wl_s$ relative to the start of the constituting level-s active period. To determine WF^{Q}_{sk}, we have to consider up to three suprema. First, the sequence of jobs ι_{s0} till ι_{sk} experience a blocking $B_s \geq 0$ by lower priority subsystems in the worst-case situation. Similar to FPDS [18, 19], the worst-case blocking is a supremum for $B_s > 0$ rather than a maximum. Second, the jobs ι_{s0} till $\iota_{s,k-1}$ need their overrun budget X_s to access global resources. Because the access to a global resource starts during the execution of the normal budget, the actual amount X of overrun budget used is a supremum rather than a maximum. Finally, the access to the global resource also starts "as late as possible" during the execution of job ι_{sk} in a worst-case situation, to maximize the interference of higher priority subsystems. This "as late as possible" also gives rise to a supremum rather than a maximum. The worst-case finalization time WF^{Q}_{sk} can therefore be described as

$$WF^{\mathrm{Q}}_{sk} = \lim_{Q \uparrow Q_s} \lim_{X \uparrow X_s} \lim_{B \uparrow B_s} WR^{\mathrm{P}}_s(B + k(Q_s + X) + Q),$$

where WR^{P}_s is the worst-case response time of a fictive subsystem S'_s with a period $P'_s = (k+1)T_s$, a normal budget $Q'_s = k(Q_s + X) + Q$, and a maximum blocking time B. Using the following equation from [19]

$$\lim_{x \uparrow C} WR^{\mathrm{P}}_i(x) = WR^{\mathrm{P}}_i(C) \tag{14}$$

we derive

$$WF^{\mathrm{Q}}_{sk} = WR^{\mathrm{P}}_s(B_s + (k+1)Q_s + kX_s). \tag{15}$$

Let job ι_{sk} of S_s access $R_l \in \mathcal{R}$. When ι_{sk} starts to consume its overrun budget X_s, the subsystems S_{s-1} till S_{RC_l} are already blocked, and only subsystems with a priority higher than RC_l can therefore still pre-empt X_s. To determine the worst-case response time WR_{sk} of job ι_{sk} of S_s, we now introduce a fictive subsystem S'_{RC_l}, i.e. a subsystem that can only be pre-empted by tasks with a priority higher than RC_l. The preemptions during WF^{Q}_{sk} by subsystems S_{s-1} till S_{RC_l} are treated as *additional* blocking of S'_{RC_l}. The worst-case interference of the subsystems S_{s-1} till S_{RC_l} in the interval of length WF^{Q}_{sk} is denoted by $WI^{s-1}_{RC_l,k}$ and given by

$$WI^{s-1}_{RC_l,k} = \sum_{s-1 \le t \le RC_l} \left\lceil \frac{WF^{\mathrm{Q}}_{sk}}{P_t} \right\rceil (Q_t + X_t). \tag{16}$$

The worst-case response time WR_{sk} of job ι_{sk} of subsystem S_s is now given by

$$WR_{sk} = WR^{\mathrm{P}}_{RC_l}(B'_{RC_l} + (k+1)(Q_s + X_s)) - kP_s, \tag{17}$$

where $WR^{\mathrm{P}}_{RC_l}$ represents the worst-case response time of a fictive subsystem S'_{RC_l} with a (budget) period P'_{RC_l} and a deadline equal to $(k+1)P_s$, a normal budget Q'_s equal to $(k+1)(Q_s + X_s) - X_s$, an overrun budget X'_s equal X_s, and a maximum blocking time B'_{RC_l} given by

$$B'_{RC_l} = B_s + WI^{s-1}_{RC_l,k}. \tag{18}$$

Finally, the worst-case response time WR_s of subsystem S_s is given by

$$WR_s = \max_{0 \le k < wl_s} WR_{sk}. \tag{19}$$

Example: Sys_1 **with** $Q_2 = 3.0$ **and** $X_2 = 1.0$**.**
We determine WR_2 using the analysis described above; see also Figure 9. Because S_2 is the lowest priority subsystem, $B_2 = 0$. We first determine wl_2 using (12) and (13), and find $WL_2 = 14$ and $wl_2 = \lceil WL_2/T_2 \rceil = \lceil 14/7 \rceil = 2$. Next we determine $WR_{2,0}$ and $WR_{2,1}$ using (15) till (18). Using (15), we find $WF^{\mathrm{Q}}_{2,0} = WR^{\mathrm{P}}_2(B_2 + Q_2) = WR^{\mathrm{P}}_2(3.0) = 5$. Because $RC_l = 1$, $WI^1_{1,0} = \lceil WF^{\mathrm{Q}}_{2,0}/P_1 \rceil (Q_1 + X_1) = \lceil 5/5 \rceil 2.0 = 2.0$. Using (18), we find $B'_1 = B_2 + WI^1_{1,0} = 2.0$. Using (17), we find $WR_{2,0} = WR^{\mathrm{P}}_1(B'_1 + (Q_2 + X_2)) = WR^{\mathrm{P}}_1(6) = 6$. Similarly, we find $WF^{\mathrm{Q}}_{2,1} = WR^{\mathrm{P}}_2(7.0) = 13$, $WI^1_{1,1} = \lceil WF^{\mathrm{Q}}_{2,1}/P_1 \rceil (Q_1 + X_1) = \lceil 13/5 \rceil 2.0 = 6.0$, $B'_1 = B_2 + WI^1_{1,1} = 6.0$, and $WR_{2,1} = WR^{\mathrm{P}}_1(B'_1 + 2(Q_2 + X_2)) - P_2 = WR^{\mathrm{P}}_1(14) - 7 = 7$. Finally, using (19) we find $WR_2 = \max(WR_{2,0}, WR_{2,1}) = \max(6, 7) = 7$.

5.2.2 Access to multiple global resources

When a subsystem uses multiple global resources, we have to slightly adapt our analysis. In particular, when the resource ceiling RC_{sl} of resource $R_l \in \mathcal{R}_s$ is *larger* than $RC_{sl'}$ of resource $R_{l'} \in \mathcal{R}_s$, i.e. *more* subsystems can pre-empt S_s during its access to R_l than to $R_{l'}$, and the maximum execution access time X_{sl} of S_s to R_l is *smaller* than $X_{sl'}$, the system may be schedulable for $R_{l'}$ but not for R_l. As an example consider a system containing 2 global resources R_1 and R_2 and 3 subsystems S_1, S_2, and S_3, where the subsystems have timing characteristics as given in Table 2. The schedulability of S_3 for $X_{3,1}$ follows immediately from the similarity of systems Sys_1 and Sys_2, and the feasibility area shown in Figure 7. Subsystem S_3 just meets its deadline at $t = 7$ for its overrun budget $X_{3,2} = 0.4$ under worst-case conditions, i.e. a simultaneous release of all three subsystems at time $t = 0$ and resources accesses by both S_1 and S_2 requiring the usage of their overrun budgets at every

subsystem	P_s	Q_s	$X_{s,1}$	$X_{s,2}$
S_1	5	1	0.6	0
S_2	5	0.2	0	0.2
S_3	7	3	1	0.4

Table 2. Subsystem characteristics of Sys_2.

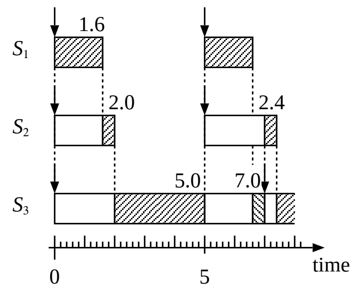

Figure 10. Subsystem S_3 just meets it deadline at $t = 7$ for $X_{3,2} = 0.4$.

activation; see Figure 10. Note that subsystem S_3 will miss its deadline at time $t = 7$ for an infinitesimal increase $\varepsilon > 0$ of $X_{3,2}$.

The easiest, but a pessimistic, way out would be to assume a *maximum* overrun budget and a *minimum* deferral of executions of subsystems with a priority higher than S_s, i.e. to use X_s and RC^s rather than R_l, where RC^s is defined as

$$RC^s = \max\{RC_l \mid R_l \in \mathcal{R}_s\}. \tag{20}$$

Note that such an analytical approach would classify Example 2 as unschedulable, however.

Alternatively, we can determine the worst-case response time for each job of S_s for individual global resources and subsequently take the maximum, i.e. we replace (17) by

$$WR_{skl} = WR^{\mathrm{P}}_{RC_l}(B'_{RC_l} + (k+1)Q_s + kX_s + X_{sl}) - kP_s \tag{21}$$

and

$$WR_{sk} = \max_l WR_{skl}. \tag{22}$$

Example: Sys_2**.**
We (only) determine $WR_{3,0}$ using the analysis described above; see also Figure 10. Because S_3 is the lowest priority subsystem, $B_3 = 0$, and $WF^{\mathrm{Q}}_{3,0} = WR^{\mathrm{P}}_3(B_3 + Q_3) = WR^{\mathrm{P}}_3(3.0) = 5.0$. We first determine $WR_{3,0,1}$. For R_1 and $RC_1 = 1$, we find $WI^2_{1,0} = \sum_{t=1}^{2}\lceil WF^{\mathrm{Q}}_{3,0}/T_t\rceil(Q_t + X_t) = 2.0$ and $B'_1 = B_3 + WI^2_{1,0} = 2.0$. Using (21), we find $WR_{3,0,1} = WR^{\mathrm{P}}_1(B'_1 + Q_3 + X_{3,1}) = WR^{\mathrm{P}}_1(6.0) = 6.0$. Next, we determine $WR_{3,0,2}$. For R_2 and $RC_2 = 2$, we find $WI^2_{2,0} = \sum_{t=2}^{2}\lceil WF^{\mathrm{Q}}_{3,0}/T_t\rceil(Q_t + X_t) = 0.4$ and $B'_2 = B_3 + WI^2_{2,0} = 0.4$. Using (21), we find $WR_{3,0,2} = WR^{\mathrm{P}}_2(B'_2 + Q_3 + X_{3,2}) = WR^{\mathrm{P}}_2(3.8) = 7.0$. Finally, using (22) we find $WR_{3,0} = \max(WR_{3,0,1}, WR_{3,0,2}) = \max(6.0, 7.0) = 7.0$.

5.2.3 Concluding remarks

In this section, we briefly discuss two aspects of the global analysis, i.e. the global analysis is *uniform* and *sustainable*.

The analysis for FPDS [18, 19] is not uniform for all tasks, i.e. the analysis for the lowest priority task differs from the analysis of the other tasks. This anomaly is caused by the fact that the lowest priority task cannot be blocked, i.e. its blocking time is zero, and the blocking time of all other tasks is a supremum rather than a maximum. Unlike the analysis for FPDS [18, 19], the global analysis presented in this section is uniform. This is an immediate consequence of the fact that blocking of a global resource R_l by a subsystem S_s is already done during the execution of the normal budget, i.e. *before* the execution based on the overrun budget starts. As a result, subsystems S_{s-1} till S_{RC_l} cannot preempt S_s at the finalization time of Q_s, irrespective of s.

As described in [21], a schedulability test is *sustainable* if any task system deemed schedulable by the test remains so if it behaves 'better' than mandated by its system specifications, i.e. sustainability requires that schedulability be preserved in situations in which it should be 'easier' to ensure schedulability. Given our scheduling model, we use the following definition for sustainability.

Definition 1 *A schedulability test for our real-time scheduling model for subsystems is* sustainable *if any system deemed schedulable by the schedulability test remains schedulable when the parameters of one or more individual subsystem[s] are changed in any, some, or all of the following ways: (*i*) decreased normal budgets; (*ii*) decreased overrun budgets, (*iii*) larger (budget) periods; and (*iv*) larger relative deadlines.*

With this definition, sustainability of our global schedulability test immediately follows from (7), i.e. $WR_s \leq P_s = D_s$ and the fact that

- the maximum number wl_s of jobs of subsystem S_s in a level-s active period, and
- the worst-case finalization time WF^{Q}_{sk} in (15), the worst-case interference $WI^{s-1}_{RC_l,k}$ in (16), and the worst-case response time WR_{skl} in (21)

are strictly non-increasing for decreasing normal budgets, decreasing overrun budgets, and increasing budget periods of subsystems.

6 Improved local analysis

Both the existing global schedulability analysis and the improved global schedulability analysis assume a deadline for a subsystem S_s equal to its period P_s for the sum of the normal budget Q_s and the overrun budget X_s. The existing local schedulability analysis for the tasks of S_s is exclusively based on Q_s, however. Hence, when a system is feasible from a global scheduling perspective, the latest finalization time of Q_s is guaranteed to be at least X_s before the next activation of S_s. Hence, we can use the supply bound function $\mathtt{sbf}_{\Omega_s}(t)$ of the EDP resource model $\Omega_s(P_s, Q_s, \Delta_s)$ for overrun without payback rather than $\mathtt{sbf}_{\Gamma_s}(t)$ of $\Gamma_s(P_s, Q_s)$ in (8), where $\Delta_s = P_s - X_s$. Because $X_s \geq 0$ for all subsystems (by definition), $\mathtt{sbf}_{\Gamma_s}(t) \leq \mathtt{sbf}_{\Omega_s}(t)$ for all subsystems. As a result, a subsystem may be schedulable according to the local analysis based on $\mathtt{sbf}_{\Omega_s}(t)$, but not be schedulable based on $\mathtt{sbf}_{\Gamma_s}(t)$.

Figure 11 shows an example of the supply bound functions $\mathtt{sbf}_{\Omega}(t)$ and $\mathtt{sbf}_{\Gamma}(t)$ for subsystem S_2 of system Sys_1 with $Q_2 = 1.8$ and $X_2 = 2.4$.

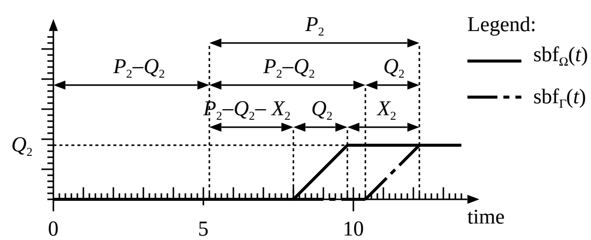

Figure 11. Supply bound functions $\mathtt{sbf}_{\Omega}(t)$ and $\mathtt{sbf}_{\Gamma}(t)$ for S_2 with $Q_2 = 1.8$ and $X_2 = 2.4$.

7 Discussion

In this section, we consider directions for further improvements.

7.1 Decreasing external resource ceilings

Figure 10 showed a timeline where subsystem S_3 just meets its deadline at $t = 7$ for $X_{3,2} = 0.4$. By *decreasing* the *external* resource ceiling RC_2 of resource R_2 from 2 to 1, subsystem S_1 can no longer pre-empt the execution of X_2. As a result, the resource holding time [17] of R_2 by S_2 is reduced from $Q_1 + X_{1,1} + X_{3,2} = 2.4$ to $X_{3,2} = 0.4$. For this particular example, it immediately follows from the similarity with system Sys_1 that we can even *increase* $X_{3,2}$ to 1.0 when we *decrease* RC_2 from

2 to 1 without making the system unschedulable. In general, decreasing a resource ceiling RC_s from u to v may *improve* the schedulability of subsystems S_w with $s \geq w \geq u$ and *worsen* the schedulability of subsystems S_w with $u > w \geq v$. Hence, given the improved global schedulability presented in Section 5, we may further improve the schedulability of a system by decreasing external resource ceilings of global resources. Note that this improvement is only possible because of the limited pre-emptability of the overrun budget on the one hand and the fact that the overrun budget is executed as last budget.

7.2 Further global analysis improvements

We briefly consider two further improvements of the global analysis, which we also illustrate by means of system Sys_1, i.e.

3. *The deadline P_s holds for Q_s only*;

4. *The remainder of X_s is discarded upon a replenishment*: because when the budget is replenished, X_s is no longer needed.

Because the deadline P_s only holds for Q_s, the improvement of the local schedulability analysis described in Section 6 does no longer apply for these two further improvements of the global analysis.

7.2.1 Deadline only for normal budget

The overrun budget X_s is needed if and only if the normal budget Q_s of a subsystem S_s becomes depleted whilst S_s holds a global resource. As soon as the normal budget is replenished, there is no need to use the overrun budget. Hence, the deadline of a subsystem S_s only holds for its normal budget. The resulting improvement is illustrated in Figure 12. Note that the for

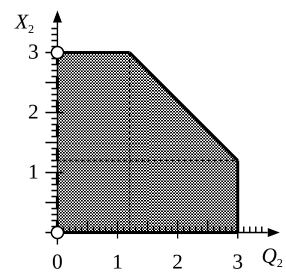

Figure 12. Feasibility area assuming the deadline only for the normal budget.

the line starting at $< 1.2, 3 >$ till point $< 3, 1.2 >$ the utilization of the system $U = \frac{Q_1+X_1}{P_1} + \frac{Q_2+X_2}{P_2} = 1$.

Figure 13 shows a timeline for $Q_2 = 3.0$ and $X_2 = 1.2$ with a simultaneous activation of S_1 and S_2 at $t = 0$. The figure illustrates that the worst-case response time of the normal budget Q_2 is equal to 6.6, and Q_2 is therefore always provided before the relative deadline $D_2 = 7.0$.

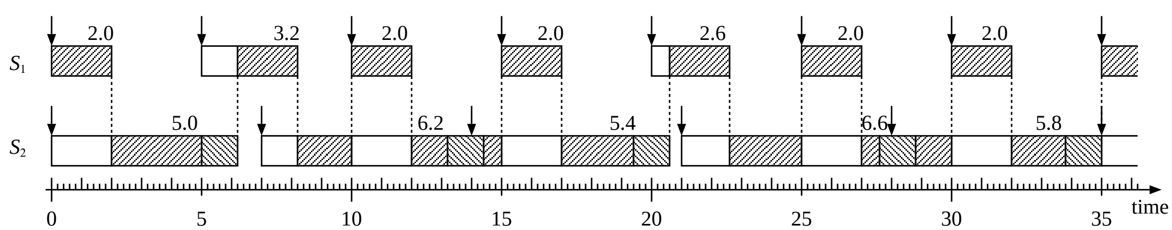

Figure 13. Timeline for $Q_2 = 3.0$ and $X_2 = 1.2$.

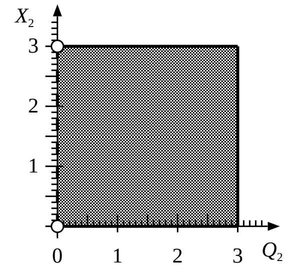

Figure 14. Feasibility area assuming overrun ends upon replenishment.

7.2.2 Overrun ends upon replenishment

The last improvement results from the observation that the remainder of the overrun budget X_s of a subsystem S_s can be *discarded* upon replenishment of its normal budget Q_s. As a result, the utilization U of the subsystems expressed as $\sum_{1 \leq s \leq N} \frac{Q_s + X_s}{P_S}$ can become *larger* than 1. The resulting improvement is illustrated in Figure 14.

Figure 15 shows a timeline for $Q_2 = 2.8$ and $X_2 = 3.0$ with a simultaneous activation of S_1 and S_2 at $t = 0$. The figure illustrates that 0.8 of the overrun budget X_2 is lost at times $t \in \{7, 21, 35\}$ and that 2.8 is lost at times $t \in \{14, 28\}$.

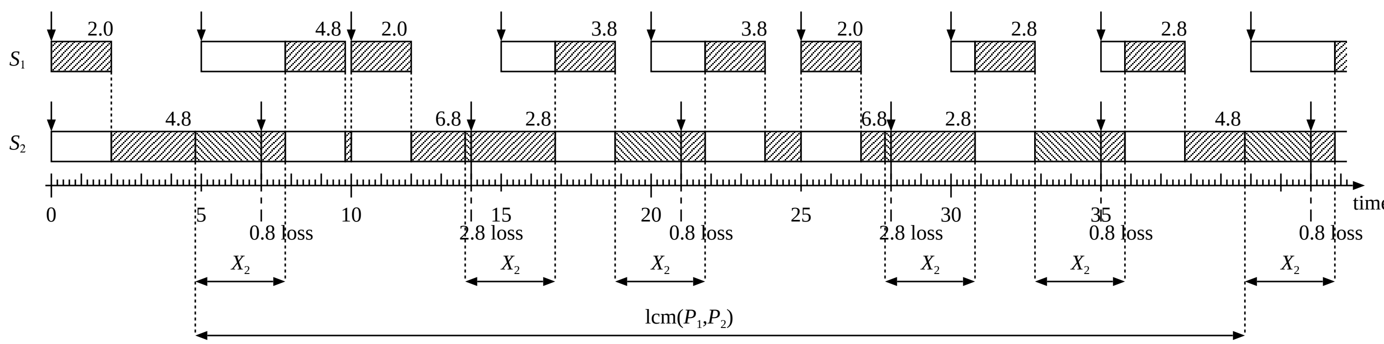

Figure 15. Timeline for $Q_2 = 2.8$ and $X_2 = 3.0$ when overrun ends upon replenishment.

8 Conclusion

We showed that existing global and local schedulability analysis of synchronization protocols based on SRP and overrun without payback for two-level hierarchical scheduling based on FPPS is pessimistic. We presented improved global and local analysis assuming that the deadline of a subsystem holds for the sum of its normal budget and its overrun budget. We illustrated the improvements by means of examples, and showed that the improved global analysis is both uniform and sustainable. Finally, we briefly discussed further options for improvements, i.e. (*i*) to *decrease* external resource ceilings and (*ii*) to assume that the deadline P_s only holds for Q_s and that X_s can be discarded upon a replenishment of the budget of S_s. For improvement (*ii*), the improved local analysis can not be applied, however.

The evaluation of the improvements through simulation, the consequences of decreasing resource ceilings, and the applicability of the improvements identified for the other flavor of the overrun mechanism, i.e. *with payback*, are left as topics of future work.

Acknowledgements

The work in this document is supported by the Swedish Foundation for Strategic Research (SSF), via the research programme PROGRESS. We thank Martijn M.H.P. van den Heuvel and Mike J. Holenderski from the TU/e for their comments on an earlier version of this document.

References

[1] R. Bril, U. Keskin, M. Behnam, and T. Nolte, “Schedulability analysis of synchronization protocols based on overrun without payback for hierarchical scheduling frameworks revisited,” in *Proc. 2nd Workshop on Compositional Theory and Technology for Real-Time Embedded Systems (CRTS)*, Dec. 2009.

[2] I. Shin and I. Lee, “Periodic resource model for compositional real-time guarantees,” in *Proc. 24th IEEE Real-Time Systems Symposium (RTSS)*, Dec. 2003, pp. 2–13.

[3] T. Baker, “Stack-based scheduling of realtime processes,” *Real-Time Systems*, vol. 3, no. 1, pp. 67–99, March 1991.

[4] M. Behnam, I. Shin, T. Nolte, and M. Nolin, “SIRAP: A synchronization protocol for hierarchical resource sharing in real-time open systems,” in *Proc. 7th ACM and IEEE Int. Conference on Embedded Software (EMSOFT)*, October 2007, pp. 279–288.

[5] M. Behnam, T. Nolte, and I. Shin, “Scheduling of semi-independent real-time components: Overrun methods and resource holding times,” in *Proc. 13th IEEE Int. Conference on Emerging Technologies and Factory Automation (ETFA)*, September 2008, pp. 575–582.

[6] Z. Deng and J.-S. Liu, “Scheduling real-time applications in open environment,” in *Proc. 18th IEEE Real-Time Systems Symposium (RTSS)*, Dec. 1997, pp. 308–319.

[7] X. Feng and A. Mok, “A model of hierarchical real-time virtual resources,” in *Proc. 23rd IEEE Real-Time Systems Symposium (RTSS)*, Dec. 2002, pp. 26–35.

[8] T.-W. Kuo and C.-H. Li, “A fixed-priority-driven open environment for real-time applications,” in *Proc. 20th IEEE Real-Time Systems Symposium (RTSS)*, Dec. 1999, pp. 256–267.

[9] G. Lipari and S. Baruah, “Efficient scheduling of real-time multi-task applications in dynamic systems,” in *Proc. 6th IEEE Real-Time Technology and Applications Symposium (RTAS)*, May-June 2000, pp. 166–175.

[10] A. Easwaran, M. Anand, and I. Lee, “Compositional analysis framework using EDP resource models,” in *Proc. 28th IEEE Real-Time Systems Symposium (RTSS)*, Dec. 2007, pp. 129–138.

[11] T. Ghazalie and T. Baker, “Aperiodic servers in a deadline scheduling environment,” *Real-Time Systems*, vol. 9, no. 1, pp. 31–67, July 1995.

[12] R. Davis and A. Burns, “Resource sharing in hierarchical fixed priority pre-emptive systems,” in *Proc. 27th IEEE Real-Time Systems Symposium (RTSS)*, Dec. 2006, pp. 257–267.

[13] M. Behnam, I. Shin, T. Nolte, and M. Nolin, “An overrun method to support composition of semi-independent real-time components,” in *Proc. Annual IEEE Int. Computer Software and Applications Conference (COMPSAC), Workshop on Component-Based Design of Resource-Constrained Systems (CoRCS)*, July 2008, pp. 1347–1352.

[14] I. Shin, M. Behnam, T. Nolte, and M. Nolin, “Synthesis of optimal interfaces for hierarchical scheduling with resources,” in *Proc. 29th IEEE Real-Time Systems Symposium (RTSS)*, Dec. 2008, pp. 209–220.

[15] P. Holman and J. Anderson, “Locking in pfair-scheduled multiprocessor systems.” in *Proc. 23rd IEEE Real-Time Systems Symposium (RTSS)*, Dec. 2002, pp. 149–158.

[16] M. Behnam, T. Nolte, M. Åsberg, and R. Bril, “Overrun and skipping in hierarchical scheduled real-time systems,” in *Proc. 15th IEEE Int. Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, August 2009, pp. 519–526.

[17] M. Bertogna, N. Fisher, and S. Baruah, “Static-priority scheduling and resource hold times,” in *Proc. 15th Int. Workshop on Parallel and Distributed Real-Time Systems*, March 2007, pp. 1–8.

[18] R. Bril, J. Lukkien, and W. Verhaegh, “Worst-case response time analysis of real-time tasks under fixed-priority scheduling with deferred preemption revisited,” in *Proc. 19th Euromicro Conference on Real-Time Systems (ECRTS)*, July 2007, pp. 269–279.

[19] ——, "Worst-case response time analysis of real-time tasks under fixed-priority scheduling with deferred preemption," *Real-Time Systems journal*, vol. 42, no. 1-3, pp. 63–119, August 2009.

[20] J. Regehr, "Scheduling tasks with mixed preemption relations for robustness to timing faults," in *Proc.* 23*rd* *IEEE Real-Time Systems Symposium (RTSS)*, Dec. 2002, pp. 315–326.

[21] A. Burns and S. Baruah, "Sustainability in real-time scheduling," *Journal of Computing Science and Engineering*, vol. 2, no. 1, pp. 74–97, March 2008.

A Rectifications and extensions

This document rectifies and extends [1]. Rectifications include various typos and omissions, such as

- the relation between c_{sil} and R_l in Section 3.3, i.e. we introduced a dedicated set $\mathcal{R}_{si}$ of global resources accessed by task τ_{si} and replaced $c_{sil} > 0 \Leftrightarrow R_l \in \mathcal{R}_s$ by $c_{sil} > 0 \Leftrightarrow R_l \in \mathcal{R}_{si}$;
- the relation between the internal resource ceilings rc_{sl}, rc'_{sl}, and rc_{sl}^{HSRP} in Section 3.5, i.e. we replaced '$\leq$' by '$\geq$';
- Figure 15, i.e. we included the access-time to the global resource and resolved the error with the timeline.

This document extends [1] with additional explanations, including

- the relation between highest and lowest system/subsystem ceiling on the one hand and the *value* of an external/internal resource ceiling on the other hand in Section 3.5;
- extended descriptions of the improved global analysis in Section 5.2.1;
- a discussion on the uniformity and the sustainability of the global analysis in Section 5.2.3;
- an example and a figure in Section 7.2.1.