Schedulability analysis of synchronization protocols based on overrun without payback for hierarchical scheduling frameworks revisited
Bril, R.J.; Keskin, U.; Behnam, M.; Nolte, T.

Published: 01/01/2010

Document Version
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
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 (FPSS). 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 (FPSS) is the de facto standard of task scheduling, hence we focus on an HSF with support for FPSS 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) FPSS 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 pre-empt. 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 $\Theta (0 < \Theta \leq \Delta \leq \Pi)$.

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

\[
X_s = \max \{ X_{si} | R_i \in R_s \}. \tag{1}
\]

The overrun budget of \( S \) 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 \( S \) 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 \( \tau_{si} \in T_s \) are specified by means of a quartet \( < T_{si}, C_{si}, D_{si}, C_{si} \geq D_{si} \geq T_{si} > \), 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, \( C_{si} \) a set of maximum execution access times of \( \tau_{si} \) to global resources, where \( C_{si} \leq D_{si} \leq T_{si} \). The set \( R_{si} \) denotes the subset of \( R_s \) accessed by task \( \tau_{si} \). The maximum time that a task \( \tau_{si} \) executes while accessing resource \( R_i \in R_{si} \) is denoted by \( c_{si} \), where \( c_{si} \in \mathbb{R}^+ \cup \{0\} \), \( C_{si} \geq c_{si} \), and \( c_{si} > 0 \iff R_i \in R_{si} \).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 \( \text{sbf}_\Omega(t) \) of the EDP resource model \( \Omega(\Pi, \Theta, \Delta) \) that computes the minimum possible CPU supply for every interval length \( t \) is given by

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

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

The supply bound function \( \text{sbf}_\Gamma(t) \) 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 \( S \) during access to a global resource \( R_i \) by temporarily increasing the budget of \( S \) with \( X_{si} \), the maximum time that \( S \) executes while accessing \( R_i \). 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 \( R_i \), two types of resource ceilings are associated: an external resource ceiling \( RC_i \) for global scheduling and an internal resource ceiling \( rc_{si} \) for local scheduling. According to SRP, these ceilings are defined as

\[
RC_i = \min(N, \min\{ s | X_{si} > 0 \}), \tag{3}
\]

\[
rc_{si} = \min(n_s, \min\{ i | c_{si} > 0 \}). \tag{4}
\]

Note that we use the outermost min in (3) and (4) to define \( RC_i \) and \( rc_{si} \), also in those situations where no subsystem uses \( R_i \) and no task of \( T_s \) uses \( R_i \), respectively.

²In [12], it is required that \( c_{si} < C_{si} \) and \( c_{si} < Q_s \). Moreover, it is observed that \( c_{si} \) 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 \( \tau_i \) can only preempt the currently executing task \( \tau_j \) (even when accessing a global resource) if the priority of \( \tau_i \) is greater (i.e. the index \( i \) is lower) than \( \tau_j \) 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. \( rc_{sl}^{HSRP} = 1 \). 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 \( rc_{sl} \geq rc_{sl}^{'} \geq rc_{sl}^{HSRP} = 1 \).

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.

<table>
<thead>
<tr>
<th>subsystem</th>
<th>( P_s )</th>
<th>( Q_s )</th>
<th>( X_s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_1 )</td>
<td>5</td>
<td>2</td>
<td>( Q_1 + X_1 )</td>
</tr>
<tr>
<td>( S_2 )</td>
<td>7</td>
<td>( Q_2 + X_2 )</td>
<td></td>
</tr>
</tbody>
</table>

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

\[
x = B_s + (Q_s + X_s) + \sum_{t < s} \left\lfloor \frac{x}{P_t} \right\rfloor (Q_t + X_t),
\]

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

\[
B_s = \max (0, \max \{X_{sl} \mid t > s \land X_{sl} > 0 \land RC_i \leq s \}).
\]

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

\[
\forall 1 \leq s \leq N \quad WR_s \leq P_s
\]

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 \( S_i \) 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. \( WR_s^P \).

Because \( rc_{sl}^{HSRP} = 1 \) for \( R_l \in R_s \), \( X_{sl} = \max c_{sl} \). Hence, from \( c_{sl} < Q_s \) 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 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.

### 4.2 Local analysis

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

$$\forall 1 \leq i \leq n_s \exists b_{si} \in \mathbb{R}_{\geq 0} \sum_{j<i} \left[ \frac{f}{T_{sj}} \right] \cdot C_{sj} \leq sbf_{\Gamma_s}(t),$$

(8)

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

$$b_{si} = \max(0, \max\{c_{sj} \mid j > i \land c_{sj} > 0 \land rc_{sl} \leq i\}),$$

(9)

and $sbf_{\Gamma_s}(t)$ 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 $X_{sli}$ depends on the local scheduler and the synchronization protocol. The maximum time that subsystem $S_s$ executes while task $\tau_{sli}$ accesses resource $R_l \in R_s$ is denoted by $X_{sli}$, where $X_{sli} \in \mathbb{R}^+ \cup \{0\}$ and $X_{sli} > 0 \Leftrightarrow c_{sli} > 0$. For $c_{sli} > 0$, $X_{sli}$ is given by [5]

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

(10)

The value for $X_{si}$ is given by

$$X_{si} = \max_{1 \leq i \leq n_s} X_{sli}.$$  

(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 $\text{Sys}_1$, as illustrated in Figure 3. In this figure, it is assumed 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 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 3. Global feasibility area assuming limited pre-emption of $X_s$.

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 $X_2$.](image)

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 $t = 7$.](image)

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 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 7. Global feasibility area assuming blocking starts before overrun.](image)
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 $W_{RS}$ of subsystem $S_s$. Unlike the analysis described in [18, 19, 20], subsystems $S_{s-1}$ till $S_{RC}$ cannot preempt $S_s$ at the finalization time of $Q_s$ when $S_s$ is accessing $R_s$, 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 $i_{sk}$ 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

$$x = B_s + \sum_{t \leq s} \left[ \frac{x}{P_t} \right] (Q_t + X_t).$$

(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

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

(13)

For a job $i_{sk}$ 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 $i_{sk}$ in two sub-intervals: a first sub-interval including the execution of the normal budget $Q_s$ by job $i_{sk}$ and a second sub-interval from the finalization of $Q_s$ by $i_{sk}$ till the finalization of $i_{sk}$, i.e. constituting the execution of the overrun budget $X_s$.

Let $WF_{Q_{sk}}^Q$ denote the worst-case finalization time of the normal budget $Q_s$ of job $i_{sk}$ with $0 \leq k < wl_s$ relative to the start of the constituting level-$s$ active period. To determine $WF_{Q_{sk}}^Q$, we have to consider up to three suprema. First, the sequence of jobs $t_{i0}$ till $i_{sk}$ 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 $t_{i0}$ till $i_{v,k-1}$ 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 $i_{sk}$ 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 $WF_{sk}^{Q}$ can therefore be described as

$$WF_{sk}^{Q} = \lim_{Q',k} \lim_{x,B} WR_{P}^{C}(B + k(Q_x + X) + Q),$$

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

$$\lim_{x,C} WR_{P}^{C}(x) = WR_{P}^{C}(C)$$

we derive

$$WF_{sk}^{Q} = WR_{P}^{C}(B_s + (k + 1)Q_s + kX_s).$$

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

$$WF_{RC_{i},k} = \sum_{s-1 \leq s \leq RC_{i}} \left[ WR_{sk}^{Q}(P_{s}) \right] (Q_s + X_s).$$

The worst-case response time $WF_{sk}$ of job $t_{sk}$ of subsystem $S_s$ is now given by

$$WF_{sk} = WR_{RC_{i}}^{P}(B'_{RC_{i}} + (k + 1)(Q_s + X_s)) - kP_s,$$

where $WR_{RC_{i}}^{P}$ represents the worst-case response time of a fictive subsystem $S'_{RC_{i}}$ with a (budget) period $P'_{RC_{i}}$ 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 $B'_{RC_{i}}$ given by

$$B'_{RC_{i}} = B_s + WF_{RC_{i},k}^{-1}.$$  

Finally, the worst-case response time $WF_{s}$ of subsystem $S_s$ is given by

$$WF_{s} = \max_{0 \leq s < w_{s}} WF_{sk}.$$  

### Example: $Sys_3$ with $Q_2 = 3.0$ and $X_2 = 1.0$.

We determine $WF_{s}$ using the analysis described above; see also Figure 9. Because $S_2$ is the lowest priority subsystem, $B_2 = 0$. We first determine $w_{l_2}$ using (12) and (13), and find $W_{l_2} = 14$ and $w_{l_2} = \lceil W_{l_2}/T_2 \rceil = \lceil 14/7 \rceil = 2$. Next we determine $WF_{2.0}$ and $WF_{2.1}$ using (15) till (18). Using (15), we find $WF_{2.0}^{Q} = WF_{2.0}^{P}(B_s + Q_2) = WF_{2.0}^{P}(3.0) = 5$. Because $RC_{i} = 1, W_{l_2}^{1.0} = \lceil WF_{2.0}^{Q}/P_{1} \rceil (Q_1 + X_1) = \lceil 5/5 \rceil 2.0 = 2.0$. Using (18), we find $B'_{1} = B_s + W_{l_2}^{1.0} = 2.0$. Using (17), we find $WF_{2.0} = WF_{2.0}^{P}(B'_{1} + (Q_2 + X_2)) = WF_{2.0}^{P}(6) = 6$. Similarly, we find $WF_{2.1}^{Q} = WF_{2.1}^{P}(7.0) = 13, W_{l_2}^{1.1} = \lceil WF_{2.1}^{Q}/P_{1} \rceil (Q_1 + X_1) = \lceil 13/5 \rceil 2.0 = 6.0, B'_{1} = B_s + W_{l_2}^{1.1} = 6.0, and WR_{2.1} = WR_{2.1}^{P}(B'_{1} + (2Q_2 + X_2)) - P_2 = WR_{2.1}^{P}(14) - 7 = 7$. Finally, using (19) we find $WF_{2} = \max(WF_{2.0}, WF_{2.1}) = \max(6,7) = 7$.

### 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 $RC_{sk}$ of resource $R_{i} \in R_{i}$ is larger than $RC_{sk}^{P}$ of resource $R_{P} \in R_{P}$, i.e. more subsystems can pre-empt $S_{s}$ during its access to $R_{i}$ than to $R_{P}$, and the maximum execution access time $X_{sk}$ of $S_{s}$ to $R_{i}$ is smaller than $X_{sk}^{P}$, the system may be schedulable for $R_{P}$ but not for $R_{i}$. 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 $S_1$ 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
activation; see Figure 10. Note that subsystem $S_3$ will miss its deadline at time $t = 7$ for an infinitesimal increase $\epsilon > 0$ 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 $S$, i.e. to use $X_t$ and $RC^*$ rather than $R_l$, where $RC^*$ is defined as

$$RC^* = \max \{ RC_t \mid R_l \in R_s \}. \quad (20)$$

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 $S$, for individual global resources and subsequently take the maximum, i.e. we replace (17) by

$$WR_{skl} = WR_{RC}^P(B_{RC}^* + (k+1)Q_s + kX_s + X_{sl}) - kP_s \quad (21)$$

and

$$WR_{sk} = \max_l WR_{skl}. \quad (22)$$

**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 $WR_{3,0}^Q = WR_{3}^P(B_3 + Q_3) = WR_{3}^P(3.0) = 5.0$. We first determine $WR_{3,0,1}$. For $R_1$ and $RC_1 = 1$, we find $W_{1,0}^T = \sum_{t=1}^{2} [WF_{3,0}^Q/T](Q_t + X_t) = 2.0$ and $B_1^* = B_3 + W_{1,0}^T = 2.0$. Using (21), we find $WR_{3,0,1} = WR_{3}^P(B_3^* + Q_3 + X_{3,1}) = WR_{3}^P(6.0) = 6.0$. Next, we determine $WR_{3,0,2}$. For $R_2$ and $RC_2 = 2$, we find $W_{2,0}^T = \sum_{t=2}^{2} [WF_{3,0}^Q/T](Q_t + X_t) = 0.4$ and $B_2^* = B_3 + W_{2,0}^T = 0.4$. Using (21), we find $WR_{3,0,2} = WR_{3}^P(B_2^* + Q_3 + X_{3,2}) = WR_{3}^P(3.8) = 7.0$. 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 $R_1$ by a subsystem $S_t$ 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 $S_{t-1}$ till $S_{RC}$ cannot preempt $S_t$ at the finalization time of $Q_s$, irrespective of $s$.  

<table>
<thead>
<tr>
<th>subsystem</th>
<th>$P_s$</th>
<th>$Q_s$</th>
<th>$X_{s,1}$</th>
<th>$X_{s,2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1$</td>
<td>5</td>
<td>1</td>
<td>0.6</td>
<td>0</td>
</tr>
<tr>
<td>$S_2$</td>
<td>5</td>
<td>0.2</td>
<td>0</td>
<td>0.2</td>
</tr>
<tr>
<td>$S_3$</td>
<td>7</td>
<td>3</td>
<td>1</td>
<td>0.4</td>
</tr>
</tbody>
</table>

**Table 2. Subsystem characteristics of $Sys_2$.**

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

TU/e, CS-Report 10-05, June 2010
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 subsystems 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 of jobs of subsystem $S_s$ in a level-$s$ active period, and
- the worst-case finalization time $WF_{sk}^Q$ in (15), the worst-case interference $WF_{RC,i,k}^{P_s}$ 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 $sbf_{Q_s}(t)$ of the EDP resource model $\Omega_s(P_s, Q_s, \Delta_s)$ for overrun without payback rather than $sbf_{\Gamma_s}(t)$ 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), $sbf_{\Gamma_s}(t) \leq sbf_{Q_s}(t)$ for all subsystems. As a result, a subsystem may be schedulable according to the local analysis based on $sbf_{Q_s}(t)$, but not be schedulable based on $sbf_{\Gamma_s}(t)$.

Figure 11 shows an example of the supply bound functions $sbf_{Q_s}(t)$ and $sbf_{\Gamma_s}(t)$ for subsystem $S_2$ of system $Sys_1$ with $Q_2 = 1.8$ and $X_2 = 2.4$.

![Figure 11. Supply bound functions $sbf_{Q_s}(t)$ and $sbf_{\Gamma_s}(t)$ for $S_2$ with $Q_2 = 1.8$ and $X_2 = 2.4$.](image)

**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.](image)

the line starting at $<1.2, 3>$ till point $<3, 1.2>$ the utilization of the system $U = \frac{Q_1 + X_1}{2} + \frac{Q_2 + X_2}{3} = 1$.

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 $X_2 = 1.2$.](image)
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 \( \sum_{1 \leq s \leq N} Q_s + X_s \) 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 14. Feasibility area assuming overrun ends upon replenishment.

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


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 $R_{al}$ of global resources accessed by task $\tau_{sil}$ and replaced $c_{sil} > 0 \iff R_l \in R_{al}$ by $c_{sil} > 0 \iff R_l \in R_{sil}$;
- the relation between the internal resource ceilings $r_{c_{sil}}$, $r'_{c_{sil}}$, and $r_{HSRP}^{HSRP}$ in Section 3.5, i.e. we replaced ‘≤’ by ‘≥’;
- 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.