Response time analysis for sporadic tasks in uniprocessor fixed-priority scheduling with starting and resuming delays

Hadrien Barral, Yasmina Abdeddaïm, Damien Masson, Joël Goossens

Université Gustave Eiffel

Sous la co-tutelle de :
CNRS
ÉCOLE DES PONTS PARISTECH
UNIVERSITÉ GUSTAVE EIFFEL

UNIVERSITÉ LIBRE DE BRUXELLES

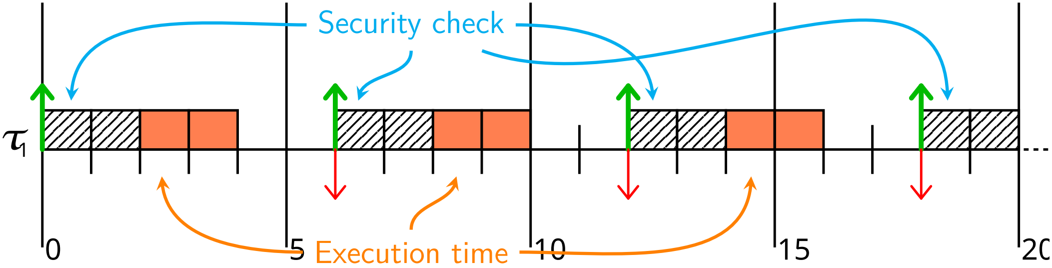
RTNS 2025-11-06

How can I know if this is schedulable?

Real-time + Security is not easy

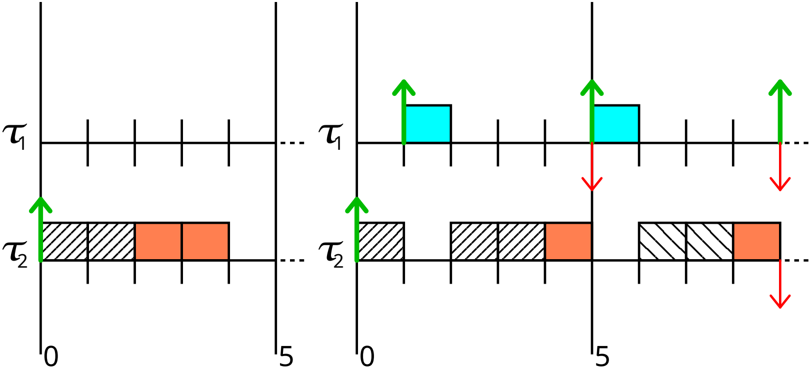
Let τ a real-time task with **strong security requirements**, e.g.:

- τ needs to check the state of the system before performing certain actions



How can I know if this is schedulable?

All of these requirements **take time**, and must be **re-done** everytime τ is prempted.



❌ We cannot simply add this time in the WCET, because we do not know **how**

Formal definition of the model

Let our real-time task set $\Sigma = \{\tau_1, \dots \tau_n\}$ be a set of independent **sporadic** real-time tasks executed using **preemptive fixed priority** algorithm in a **uniprocessor** platform, with **discrete time**.

Every real-time task is defined as a tuple $(C_i, \mathrm{SD}_i, \mathrm{RD}_i, T_i, D_i)$ with:

- C_i is the **WCET** of task τ_i,
- T_i is the **period** of the task,
- $D_i \leq T_i$ is the **deadline** of the task.
- SD_i is the worst-case **starting delay** of the task,
- RD_i is the worst-case **resuming delay** of the task,

Model based on Goossens & Masson, RTNS 2024

τ_1 is the highest priority task. Let $\mathrm{PD}_i = \max(\mathrm{SD}_i, \mathrm{RD}_i)$.

Finding the worst-case activation scenario is hard

Task set 1

τ_i	C_i	T_i	D_i	SD_i	RD_i
τ_1	1	6		1	
τ_2		12			
τ_3		12			

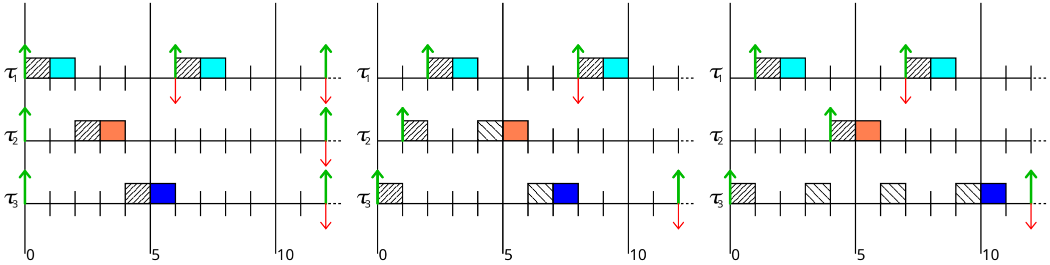


Figure: Three different activation scenarios for Task Set 1

Existing Response-time bound

Adding Instruction Cache Effect to Schedulability Analysis of Preemptive Real-Time System, RTTAS 1996 $\rightarrow$ **No proof given!**

Theorem 1

Let $\Sigma = \{\tau_1, \ldots, \tau_n\}$ *be a task set.*
The worst-case response time of task τ_i *is upper bounded by* $\mathcal{R}_i^{\mathrm{PD}}$ *where:*

$$\mathcal{R}_i^{\mathrm{PD}} = \min_{t>0} \Bigg\{ t \mid t = \mathrm{SD}_i + C_i + \sum_{k=1}^{i-1} \left\lceil \frac{t}{T_k} \right\rceil \left(\mathrm{SD}_k + C_k + \max_{\ell \in [k+1, i]} \mathrm{PD}_\ell \right) \Bigg\} \tag{1}$$

If no such minimum exists, we take $\mathcal{R}_i^{\mathrm{PD}} = \infty$.

A new response-time bound

Theorem 2

Let $\Sigma = \{\tau_1, \ldots, \tau_n\}$ *be a task set.*
The worst-case response time of task τ_i *is upper bounded by* $\mathcal{R}_i^{\mathrm{PD}}$ *where:*

$$\mathcal{R}_i^{\mathrm{PD}} = \min_{t>0} \Bigg\{ t \mid t = \mathrm{SD}_i + C_i + \sum_{k=1}^{i-1} \left\lceil \frac{\max\left(t - \mathrm{SD}_i, 0\right)}{T_k} \right\rceil \left(\mathrm{SD}_k + C_k + \max_{\ell \in [k+1, i]} \mathrm{PD}_\ell \right) \Bigg\} \tag{2}$$

If no such minimum exists, we take $\mathcal{R}_i^{\mathrm{PD}} = \infty$.

Schedulability criterion

Corollary 1

Let $\Sigma = \{\tau_1, \ldots, \tau_n\}$ be a task set. If $\forall \tau_i \in \Sigma, \mathcal{R}_i^{\mathrm{PD}} \leq D_i$, the task set is schedulable according to the fixed priority algorithm.

Proofs are provided in the paper for those interested. 🙂

Sources of response time bound pessimism (1/2)

The pessimism of our response time upper bound is due to the fact that our bound **separately** maximizes for a job:

- its computation time (execution time + preparation delays)
- the number of times the job can be preempted

TASK SET 4

τ_i	C_i	T_i	D_i	SD_i	RD_i
τ_1	1	6		1	
τ_2		7			
τ_3		11			

Our formula gives: $\mathcal{R}_3^{\mathrm{PD}} = 29$.

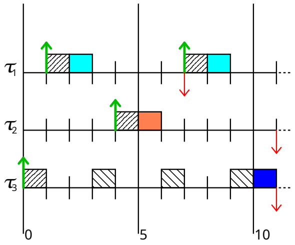
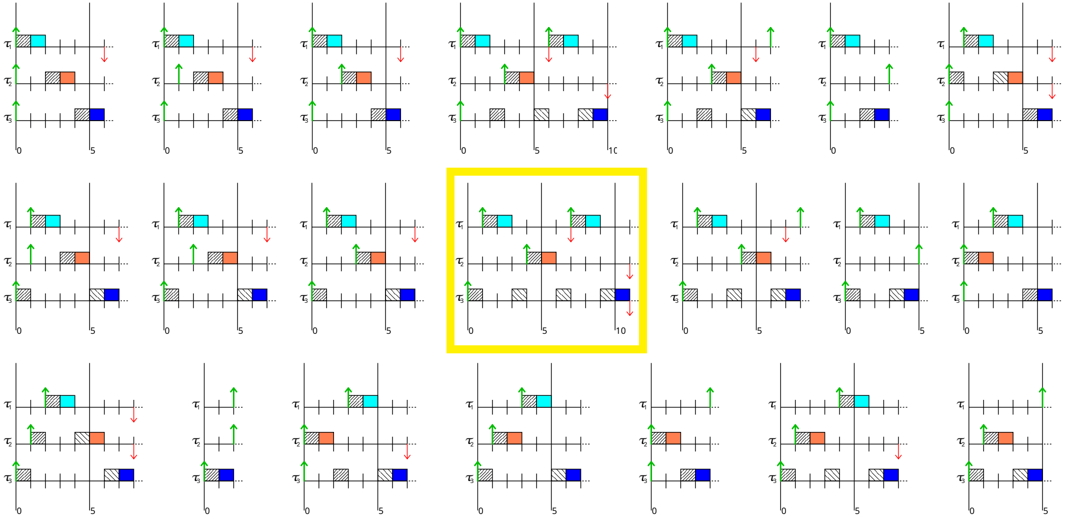


Figure: Worst possible response time for τ_3.

Sources of response time bound pessimism (2/2)

Figure: A selection of activation scenarios for Task Set 4, focused on τ_3's worst response time.



Application to Goossens & Masson, RTNS 2024 (1/2)

Robust schedulability tests for fixed job priorities: Addressing context switch costs with non-resumable delay, RTNS 2024

Differences with our model

1. Periodic tasks (with start offsets)
2. $\forall i, \mathrm{SD}_i \geq \mathrm{RD}_i$
3. Auhors give a simulation interval

Nice, our formula gives a easy-to-check schedulablity criterion!
If the criterion fails, fallback to simulation.

Application to Goossens & Masson, RTNS 2024 (2/2)

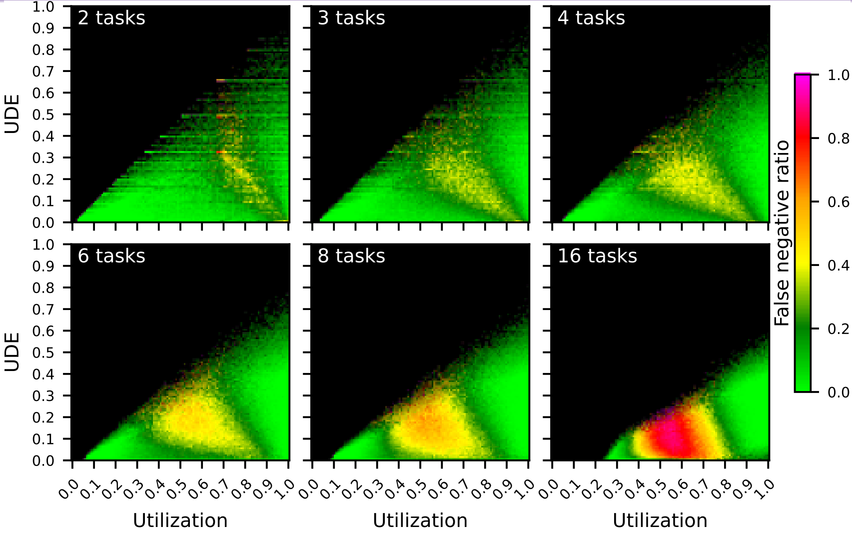


Figure: False-negative ratio for theorem 2 with Goossens & Masson's RTNS2024 model.

Tools are available!

https://github.com/LIGM-LRT/sporadic-with-start-resume-delays-response-time-analysis_rtns-2025

Tools include:

- Simulation of periodic + sporadic tasksets with start/resume delays
- Scripts to reproduce the paper experiments
- Scripts to reproduce the paper figures
- Improved *Draw Schedule* tools

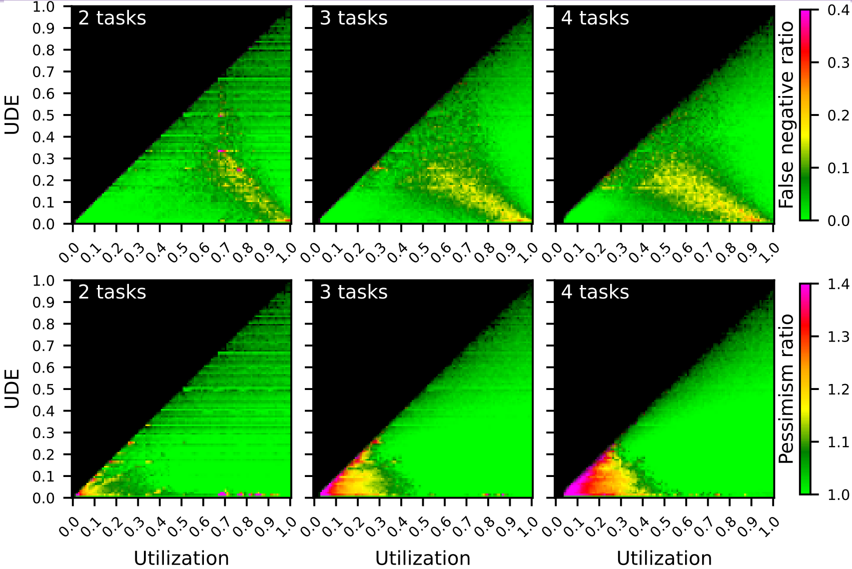
Future work

What's next?

1. Get a better understanding of when pessimism is high
2. Improve the response-time formula
3. Implement it in an usecase *(the original idea... got sidetracked)*
4. Optimal priority assigment

Time for questions! 😀

(Backup slide) Comparison with the real worst-case scenario



The worst-case scenario is task-dependent

TASK SET 2

<table>
<tr><th>τ_i</th><th>C_i</th><th>T_i</th><th>D_i</th><th>SD_i</th><th>RD_i</th></tr>
<tr><td>τ_1</td><td rowspan="3">1</td><td rowspan="3" colspan="2">99</td><td colspan="2">0</td></tr>
<tr><td>τ_2</td><td colspan="2">1</td></tr>
<tr><td>τ_3</td><td>1</td><td>2</td></tr>
</table>

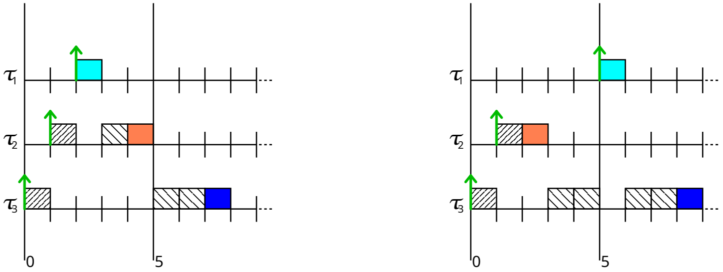


Figure: Left: worst possible response time for τ_2. Right: same for τ_3.