A schedulability test for real-time dependant periodic task systems with latency constraints

Liliana Cucu-Grosjean *　　Yves Sorel †

1 Introduction

Context of the study Real-time systems are typically modeled as finite collections of simple repetitive tasks. When different instances of those tasks are separated by a known inter-arrival time, we deal with **periodic tasks**. These tasks may also have **precedence constraints**. Such tasks may need to be scheduled before a given delay calculated from the beginning of another task. The latter delays are called **latency constraints**[1].

Our contribution In this paper we provide a schedulability test for dependant periodic task systems scheduled using non-preemptive policies. We consider the case of one processor and the schedulability test is based on the periodicity of a feasible schedule [1]. A feasible schedule is periodic if it repeats from a time instant s with a period p. In the case of one processor, this property allows a real-time designer to check the deadlines only for instances of tasks within the time interval $[s, s+p]$.

2 Model

We consider a periodic task system τ, where a task $A \in \tau$ is given by its period T_A and its worst-case execution time C_A. In a feasible schedule two consecutive instances A_i and A_{i+1} are scheduled such that $s_{A_i} + T_A = s_{A_{i+1}}$, where s_{A_i} is the start time of A_i.

Precedence constraints of tasks are defined by a infinite directed graph $\mathcal{G} = (\tau, \mathcal{E})$ where $\mathcal{E} \subseteq \tau \times \tau$ the set of precedence constraints between tasks. For instance in Figure 1 $\tau = \{A, B, C, D\}$. In Figure 1 we present two repetitions of the graph pattern.

In order to simplify the presentation and because of space limitation, we consider here that the graph pattern contains at most one instance of each task.

We denote by $\mathcal{M}(A, B)$ the set of tasks E with at least one path from A to E and at least one path from E to B.

A latency constraint $L_{AB} \in \mathbb{N}^+$ is defined for a pair (A, B) if there is, at least, one path from A to B. In this case in a feasible schedule we have $s_B + C_B - s_A \leq L_{AB}$.

*liliana.cucu@loria.fr. INRIA Nancy-Grand Est, 615 rue du Jardin Botanique, 54600 Villers-les-Nancy, France.

†yves.sorel@inria.fr. INRIA Paris-Rocquencourt, BP 105 - Domaine de Voluceau, 78153 Le Chesnay, France.

[1]An interested reader may find details on the connection between deadlines and latency constraints in [2]

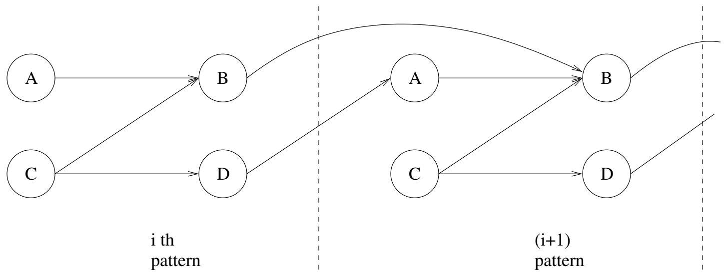

Figure 1: System of tasks defined by a infinite directed graph

3 Schedulability test

As said before, we propose a schedulability test based on the periodicity of a feasible schedule. In order to prove the periodicity we give an intermediary result (see Theorem 1) indicating that all tasks within a latency constraint must have the same period.

Theorem 1 *We consider a periodic task system τ with precedence and latency constraints. Let L_{AB} a latency constraint. If there is at least one task $E \in \mathcal{M}(A, B)$ such that $T_E \neq T_A$, then the system is not schedulable.*

Theorem 2 proves that a feasible schedule of a task system with all instances having constant execution times is periodic.

Theorem 2 *We consider a periodic task system τ with precedence and latency constraints. All instances of any task have a constant execution time equal to the worst-case execution time of the task. A feasible schedule of τ is periodic from the time instant $s = 0$ with a period p, where p is the least common multiple of $T_A, \forall A \in \tau$.*

Thanks to Theorem 2 we have a schedulability test given in Corollary 1.

Corollary 1 *We consider a periodic task system τ with precedence and latency constraints. In the case of one processor, in order to decide the schedulability of the system, one has to check the deadlines only for instances within the time interval $[s, s+p]$, with s and p defined by Theorem 2.*

References

[1] Leung, J. and Merrill, M. L. (1980). *A note on preemptive scheduling of periodic, real-time tasks.* Information Processing Letters, 11: 115-118

[2] Cucu L., Pernet N. and Sorel Y. (2008). *Periodic real-time scheduling: from deadline-based model to latency-based model.* Annals of Operations Research, 159(1): 41-51