Combining EDF and FP in Distributed Real-Time Systems: Schedulability Analysis and Optimization

Juan M. Rivas, J. Javier Gutiérrez, J. Carlos Palencia, and Michael González Harbour

Computers and Real-Time Group, Universidad de Cantabria, 39005-Santander, SPAIN
{rivasjm, gutierjj, palencij, mgh}@unican.es

Abstract

The increasing acceptance of the Earliest Deadline First (EDF) scheduling algorithm in industrial environments, together with the continued usage of Fixed Priority (FP) scheduling is leading to heterogeneous systems with different scheduling policies in the same distributed system. Schedulability analysis techniques usually consider the entire system as a whole (holistic approach), with only one preestablished scheduling policy for all the resources. In this work, composition mechanisms will be proposed that enable us to combine different FP and EDF response-time analysis techniques for checking the schedulability of heterogeneous systems. Additionally, priority and scheduling deadline assignment techniques will be combined into the a new algorithm called HOSPA (Heuristic Optimized Scheduling Parameters Assignment), for optimizing the assignment of priorities and scheduling deadlines to tasks and messages in heterogeneous distributed hard real-time systems.

1. Introduction

The usage of Earliest Deadline First (EDF) scheduling policy is starting to catch the attention of industrial environments, given its benefits in terms of increased resource usage. EDF is now present at different layers of a real-time application such as programming languages, operating systems, or even communication networks. So, it is available in real-time languages like Ada 2005 [26] or RTSJ [20], and in real-time operating systems such as SHaRK [21] or ERIKA [3]. It has been also implemented at the application level in OSEK/VDX embedded operating systems [1], and there are real-time networks using EDF for scheduling messages too; for instance in general purpose networks [2], or in the CAN Bus [17]. It is expected that the number of industrial applications using EDF will increase in the foreseeable future.

On the other hand, Fixed Priority (FP) scheduling continues to be the most popular on-line scheduling policy.

Depending on the particular requirements, for instance a mixture of hard real-time requirements together with requirements for different criticality levels, one or another scheduling policy may lead to better results. In a distributed system the scheduling problem can be better solved if different scheduling algorithms could be combined, so each processing resource would be scheduled with the optimum algorithm for the work it’s going to handle. There may also be other reasons to pick a particular scheduler for certain parts of the system, for instance we may have some legacy parts designed for FP, or an FP network such as a CAN bus, integrated with other processors where EDF is chosen because of its better usage of the available resources.

Existing response-time analysis and scheduling parameters assignment techniques that operate with the whole system are defined only for homogeneous systems, where every processor and network is scheduled with FP or EDF, so these techniques are not used for heterogeneous systems where FP and EDF schedulers could coexist.

Different abstractions have been proposed to make the performance analysis of distributed hard real-time systems. A comparison among some of these performance analysis techniques for distributed systems has been presented in [18]. That work distinguishes four different abstractions for formal performance analysis of distributed embedded systems in early design stages: holistic scheduling (MAST, Modeling and Analysis Suite for Real-Time Applications[4][12]), two compositional analysis approaches (SymTA/S [6] and MPA-RTC [24]), and a timed automata based analysis.

The compositional approaches combine local analysis and output event propagation models allowing the formal performance analysis of heterogeneous distributed systems which use different scheduling algorithms in their processors and communication networks, supporting the
combination and integration of different kinds of analysis techniques. SymTA/S [6][7] uses formal analysis techniques based on the busy window technique proposed by Lehoczky [9], and supports local analysis techniques for FP scheduling (preemptive and non-preemptive), TDMA, Round Robin, EDF, or CAN. On the other hand, MPA-RTC [24] does not rely on classical scheduling theory but it uses Real-Time Calculus [23] which extends the basic concepts of Network Calculus [8] to analyze the flow of event streams through a network of computation and communication resources. This technique can also be applied to FP scheduling, EDF and TDMA. The timed automata approach allows heterogeneous distributed systems but it is only applicable to small systems. The response-time analysis (RTA) used in MAST does not allow systems in which processing resources (processors and networks) use different scheduling policies (except for hierarchical scheduling in the same processing resource).

The objective of the present work is to extend the holistic scheduling abstraction implemented by the response time analysis techniques used in MAST with the ability of managing heterogeneous distributed systems composed of processing resources scheduled either by FP or EDF. The extension is performed at two levels:

- **Schedulability analysis**: adaptation of the response-time analysis techniques available for homogeneous distributed real-time systems scheduled by FP only or EDF only, to allow the analysis of heterogeneous systems.

- **Scheduling parameters assignment**: integration of current optimization techniques for the assignment of priorities and scheduling deadlines into a unique algorithm called HOSPA (Heuristic Optimized Scheduling Parameters Assignment). The new algorithm will allow the optimization of heterogeneous systems.

The motivation for this extension presented in this paper is to show that response-time analysis techniques and the associated optimization techniques are also amenable to composition techniques like real-time calculus and the techniques used in SymTA/S. In this way, it is possible to take advantage of optimization techniques based on RTA.

The paper is organized as follows. In Section 2 we provide a quick review of the model that we use for the distributed system. The current schedulability analysis and optimization techniques for distributed systems are shown in Section 3. Section 4 shows how analysis algorithms and scheduling-parameters-assignment algorithms for FP and EDF have been integrated to work together over the whole system. In Section 5, we discuss the implementation of the proposed techniques in MAST. Section 6 presents two examples that show how the proposed algorithms work, and also draws out the evaluation of the algorithms in terms of effectiveness. Finally, in Section 7 we give our conclusions.

### 2. The System Model

Our system model assumes a real-time distributed system with multiple processors (CPUs) and one or more communication networks. We will use either FP or EDF scheduling in each of the processors and the networks. The analysis of message traffic on the networks can be carried out using the same techniques that are used in the CPUs, so in our distributed system we treat messages and communication resources exactly as if they were tasks in processing resources, except for a blocking term that accounts for the non preemptability of message packets. Messages can only be preempted at packet boundaries. In our hard real-time system model we assume that tasks and messages are statically assigned to processors and communication networks.

We will consider a task model with periodic or sporadic (i.e., aperiodic with a specific minimum interarrival time) distributed end-to-end flows, following the terminology of the OMG standard MARTE [13]. Each end-to-end flow \( \Gamma_i \) is released by a periodic sequence of external events with period \( T_i \), and contains a set of \( m_i \) steps. A step is a task executing some code in a processor or a message sent through a communication network. Each periodic release of an end-to-end flow causes the execution of one instance of that end-to-end flow. Each step is released when the preceding one in its end-to-end flow finishes its execution.

Figure 1 shows an example of one of these end-to-end flows, with just three steps, each executing in a different processing resource (two CPUs and one network in this case). The arrival of the external event that releases the end-to-end flow is represented by a vertical arrow labeled \( e_i \), and has a period of \( T_i \). The horizontal arrows represent the release of the following step in the end-to-end flow. They represent a kind of precedence constraint because a...
step cannot be executed before the preceding step has been completed.

The \( j \)-th step of end-to-end flow \( \Gamma_j \), is identified as \( \tau_{ij} \). It has a worst-case execution time (or worst-case transmission time) of \( C_{ij} \). The non-preemptability of a network packet causing a delay for the transmission of the \( j \)-th step is modelled through a blocking time of \( B_{ij} \) which also accommodates blocking due to mutual exclusion synchronization.

The timing requirements that we consider are end-to-end deadlines, \( D_n \), that start at the end-to-end flow instance’s period, and must be met by the final step in the end-to-end flow. We allow deadlines to be larger than the periods, and thus at each time there may be several instances of the same end-to-end flow pending. Each step may also have an associated global deadline, \( D_{ij} \), which is relative to the start of the end-to-end flow instance’s period. The global deadline of the last step in the end-to-end flow coincides with the end-to-end deadline, \( D_{im} = D_i \).

For each step \( \tau_{ij} \) we define its response time as the difference between its completion time and the instant at which the period of the end-to-end flow instance started, \( t_{in} \) for the \( n \)-th instance. The worst-case response time (or an estimation of an upper bound on it) will be called \( R_{ij} \), and the best-case response time (or a lower bound estimation) will be called \( R^b_{ij} \).

We allow the external event that triggers an end-to-end flow to have a maximum release jitter \( J_{i1} \) in relation to \( t_{in} \). Other steps \( \tau_{ij} \) may also have an initial release jitter \( J_{ij} \). This implies that the release time of \( \tau_{ij} \), relative to the \( n \)-th instance of the end-to-end flow is within \([t_{in}, t_{in}+J_{ij}]\). Despite this jitter, global deadlines and response times always refer to the theoretical start of their respective instance’s period \( t_{in} \), not to the actual release of the end-to-end flow. We assume that \( J_{ij} \) may be larger than the period of its end-to-end flow, \( T_i \).

Each step \( \tau_{ij} \) may also have an associated initial offset, \( \Phi_{ij} \), which is the minimum release time for the step, relative to \( t_{in} \). Therefore, the actual release time is within \([t_{in}+\Phi_{ij}, t_{in}+\max(\Phi_{ij}, J_{ij})]\). We assume that \( \Phi_{ij} \) may be larger than the period of its end-to-end flow, \( T_i \).

We assume that the following scheduling algorithms may be used:

- FP scheduling, in which each step \( \tau_{ij} \) is assigned a priority, \( P_{ij} \).

- Global EDF scheduling, in which each step \( \tau_{ij} \) is assigned a global scheduling deadline \( SD_{ij} \) that is referenced to the arrival of the event that releases the end-to-end flow, possibly in a different processing resource. Global EDF schedulers require clock synchronization among all the processing resources involved.

- Local EDF scheduling, in which each step \( \tau_{ij} \) is assigned a local scheduling deadline \( SD_{ij} \) that is referenced to the release time of its associated step in its own processing resource, as happens with \( d_2 \) in Figure 1. Local-deadline schedulers just use the local clock of each processing resource and clock synchronization is not necessary.

3. Current Response-Time Analysis and Optimization Techniques

Different response time analysis (RTA) techniques have been proposed to analyze a system model like the one proposed here. For FP scheduling, Tindell and Clark presented the so-called holistic response-time analysis [25], that is based on the assumption that all steps of an end-to-end flow are independent of the others, except that the variability of the response time of each step can cause release jitter for the next step in the end-to-end flow. A version of this technique for EDF scheduling was developed by Spuri [22] for global-deadline schedulers. An extension of Spuri’s analysis technique for local-deadline schedulers has been proposed in [19]. These analyses are pessimistic because they can create initial conditions for the analysis that may not occur in practice.

Palencia and González provide improvements of these algorithms for FP scheduling [14][15] and EDF with global-deadline schedulers [16]. The FP technique was later improved by Makki-Turja [11]. These techniques exploit the interdependencies among steps of the same end-to-end flow by considering offsets for the release of each step. Offset-based analysis is still pessimistic, but offers much better results than the holistic analysis at the expense of more complexity.

Another problem for scheduling distributed real-time systems is finding an assignment of scheduling parameters that leads to a feasible scheduling. This problem is fully solved for single processor systems, but it has no known optimum solution for distributed systems other than an intractable brute-force mechanism.

Different heuristics that can provide acceptable solutions to the assignment of scheduling parameters in a reasonable time have been studied for fixed priorities and EDF. In this paper, we will use those based on iteratively applying RTA. The algorithm called HOPA (Heuristic Optimized Priority Assignment) [5] is shown to usually find better solutions than simulated annealing, in less time. An evolution of HOPA applicable to EDF is presented in [19]: HOSDA (Heuristic Optimized Scheduling Deadline Assignment) is useful for both global-deadline and local-deadline schedulers, and obtains better results than other methods, which are not able to optimize.
4. Analysis and Assignment of Scheduling Parameters for Heterogeneous EDF and FP Scheduling

In this section, we propose a composition approach to join the different response time analysis and the techniques for assigning scheduling parameters for heterogeneous distributed systems that combine FP and EDF schedulers.

One of the interesting properties of holistic and offset-based response time analysis is that they provide a natural way of composing analysis in different resources using different scheduling policies. The analysis in each resource is made independently, and therefore we can use whatever technique is appropriate. As a result of the analysis in one resource we get response times and then we can use them to calculate inherited offsets, \( \Phi'_{ij} \), and release jitters, \( J'_{ij} \), for the analysis in the other resources as we show below. In this way, we can combine techniques for FP and EDF. The timing effects of the activation of a step from a previous one are captured through these inherited offsets and release jitters.

For calculating response times of any step \( \tau_{ij} \), we just need to know the following parameters shown in Figure 2: the priority \( (P_{ij}) \) or scheduling deadline \( (SD_{ij} \) or \( Sd_{ij} \) for FP and EDF respectively, the inherited release jitter \( (J'_{ij}) \), and the inherited offset \( (\Phi'_{ij}) \) if we are using offset-based techniques.

![Figure 2. Input and output parameters for the analysis of a step](image)

To make this integration effective we just need to explain how to calculate inherited release jitters and offsets for any step \( \tau_{ij} \) from the response times of the previous step in its end-to-end flow. Suppose step \( \tau_{ij} \) shown in Figure 3, which is released at the finalization of step \( \tau_{ij-1} \) and activates, at its finalization, \( \tau_{ij+1} \).

![Figure 3. Portion of an end-to-end flow](image)

In addition to the initial jitter \( J_{ij} \) and offset \( \Phi_{ij} \), steps after the first are also affected by the variability of the response times of the preceding steps in the end-to-end flow. This implies that every step \( \tau_{ij} \) will have an inherited release jitter, \( J'_{ij} \) and an inherited offset \( \Phi'_{ij} \). The release time of the \( n \)-th instance of \( \tau_{ij} \), \( j \neq 1 \), depends on the response times of the preceding step and is within:

\[
release \in [t_{in} + max(R^b_{ij-1}, \Phi_{ij}), \\
\quad t_{in} + J_{ij} + max(R_{ij-1}, \Phi_{ij})]
\]

(1)

The inherited release jitter of step \( \tau_{ij} \), \( J'_{ij} \), is obtained as the difference between the worst and the best case release times of step \( \tau_{ij} \):

\[
J'_{ij} = J_{ij} + max(R_{ij-1}, \Phi_{ij}) - max(R^b_{ij-1}, \Phi_{ij})
\]

(2)

And the inherited offset of \( \tau_{ij} \), \( \Phi'_{ij} \), used by the offset-based analysis techniques, is obtained as the best-case release time:

\[
\Phi'_{ij} = max(R^b_{ij-1}, \Phi_{ij})
\]

(3)

The response-time analysis is applied iteratively to each step as shown in Figure 4. As in the homogeneous analysis techniques, the algorithm will end if, after iterating through every step in the system, no changes in the new response times are detected. If changes are detected, the iteration will start again, but this time with the newly calculated inherited offsets and release jitters.

As with the homogeneous algorithms, the monotonicity of the response times with respect to the release jitters ensures the convergence of the response time analysis.

In addition to the heterogeneous RTA, this paper proposes the integration of the algorithms for priority assignment, HOPA [5], and scheduling deadline assignment, HOSDA [19], into a single one called HOSPA (Heuristic Optimized Scheduling Parameters Assignment).

Both algorithms HOPA and HOSDA use knowledge of the factors that influence the timing behavior to find an optimized solution to the priority or scheduling deadline assignment in distributed systems. Like these algorithms, HOPA/HOSDA are based on the distribution of the global deadlines of each end-to-end flow among the different steps that compose it and the iteration over the results of RTA to redistribute these deadlines. Once each step is assigned what we will call a virtual deadline (VD), it is converted into a priority (FP), a local scheduling deadline (L-SD), or a global scheduling deadline (G-SD) depending on the scheduling policy used by the step.
As we can see in Figure 5, the algorithm starts with an initial virtual deadline assignment; for this purpose we normally use the proportional deadline assignment algorithm proposed in [10]. Then, the analysis of the whole system for each step is carried out using the heterogeneous RTA in order to obtain the worst-case response times. If a schedulable solution is found, or some other stopping condition [5][19] is reached, the algorithm ends; otherwise, a new virtual deadline assignment is performed and new scheduling parameters are calculated.

5. Implementation in MAST

In this section we present the implementation of the proposed analysis and scheduling parameters assignment techniques into the MAST suite of tools [12], which is based on a detailed model of the timing behavior of a real-time system [4] and is offered as free software. The implementation effort is relatively small as we are reusing previous implementations of the RTA algorithms that we have now combined. The MAST platform allows us to make the resulting implementation of the techniques available to the community.

For homogeneous systems, MAST lets users choose which schedulability analysis techniques to use, and this option is kept for heterogeneous systems. By default, MAST will use the most precise available technique for each scheduling policy present in the system. At this moment, MAST includes the schedulability techniques by Tindell and Clark [25], and Palencia and González [14][15] for FP, and the techniques by Spuri [22] for EDF with global-deadline schedulers, and by Rivas et al [19] for EDF with local-deadline schedulers. The offset-based technique by Palencia and González [16] for EDF with global-deadline schedulers will be added to MAST in the near future and it will be available for heterogeneous systems too.

Since the analysis techniques are potentially going to be called several times in the scheduling parameters assignment loop, and in order to avoid excessive execution times for this algorithms, a new factor has been added that will truncate the analysis iteration if any step reaches a response time greater than its deadline multiplied by this factor. Although the response times obtained in a truncated loop are not the correct ones, they are “good enough” for the next virtual deadline assignment, so a new scheduling parameters assignment could be tried in the heuristic optimization process.

For the assignment techniques, we have added into MAST the possibility of choosing the initial virtual deadline assignment according to the following three options:
• Proportional Deadline (PD) assignment, which distributes deadlines proportionally to the worst-case execution times of the steps in the end-to-end flow (as described in [10]). This is the default assignment.

• Normalized Proportional Deadline (NPD) assignment, which is similar to PD but normalizing it by the utilization of the processing resources (see [10]).

• User-defined assignment, which takes the values set by the user.

In addition, the implementation of the algorithms for assigning scheduling parameters supports the concept of preassignment. When a priority or a scheduling deadline is preassigned, it remains fixed and cannot be changed by the tools.

The heterogeneous RTA and associated HOSPA algorithm have been implemented in MAST. The following section shows some examples and the evaluation of the techniques using this implementation.

6. Examples and Evaluation

This section shows two specific examples: a simple system to contrast the results obtained when different schedulability analysis algorithms for FP and EDF are combined; a complex example, to show that the scheduling parameters assignment techniques work fine for heterogeneous FP and EDF scheduling.

6.1. A Simple Example Showing the Heterogeneous Schedulability Analysis

In order to better understand the presented combination of schedulability analysis techniques, we will illustrate it with a simple example based on the one shown in [14]. Consider the system that appears in Figure 6 consisting of two CPUs and a network executing four end-to-end flows. Γ1, Γ3 and Γ4 are periodic end-to-end flows composed of a single step, while Γ2 is a periodic end-to-end flow composed of five steps. The first step τ21 suspends itself to request service from τ23. Right before the suspension of the task executing it, τ21 transmits a message through the network (τ22), which activates in turn τ23. When τ23 completes its execution it sends a new message through the network (τ24) and then the suspended task is resumed to execute step τ25. It should be noticed that steps τ21 and τ25 are executed by the same task.

Both CPUs can use FP or EDF scheduling policies. The network is a packet-based network scheduled by an FP scheduling policy. The range of priorities for FP policies is in the interval [1, 3] (the higher the value the higher the priority). Preemption is allowed in the CPUs as in the network, although for this example, we assume that messages sent through the network are composed by a single non-preemptible packet. The timing parameters of all the end-to-end flows with their steps are shown in Table 1. Timing requirements have been imposed through global end-to-end deadlines, and they have been chosen to be proportional to the period and the number of processing resources traversed by the end-to-end flow. We will assume that the best-case execution time of each step (execution of code in CPUs and transmission time of messages) is zero.

Table 1. Timing parameters for the simple example

<table>
<thead>
<tr>
<th>Γi</th>
<th>Tij</th>
<th>τij</th>
<th>Cij</th>
<th>Dij</th>
</tr>
</thead>
<tbody>
<tr>
<td>Γ1</td>
<td>20</td>
<td>τ11</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>Γ2</td>
<td>150</td>
<td>τ21</td>
<td>20</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td>τ22</td>
<td>25</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td>τ23</td>
<td>15</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td>τ24</td>
<td>34</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td>τ25</td>
<td>30</td>
<td>750</td>
</tr>
<tr>
<td>Γ3</td>
<td>30</td>
<td>τ31</td>
<td>5</td>
<td>30</td>
</tr>
<tr>
<td>Γ4</td>
<td>200</td>
<td>τ41</td>
<td>100</td>
<td>200</td>
</tr>
</tbody>
</table>

We propose the following case studies in order to show how the different schedulability analysis techniques can be combined for analyzing heterogeneous systems:

• Case-study FP-Only: both CPUs use the FP scheduling policy.
• Case-study EDF-1: CPU-1 uses the EDF scheduling policy while CPU-2 uses FP.
• Case-study EDF-2: CPU-2 uses the EDF scheduling policy while CPU-1 uses FP.

Figure 6. Simple distributed system
• Case-study EDF-All: both CPU-1 and CPU-2 use the EDF scheduling policy.

The FP-Only case-study is introduced with the sole purpose to serve as a reference. The other three case studies allow checking how the combination of the schedulability analysis techniques for FP and EDF work together. The scheduling parameters for all the steps that we have used in the experiments are shown in Table 2. Scheduling deadlines for a local-deadline scheduler and for a global-deadline scheduler have been set according to a proportional assignment (truncated to an integer value). Priorities have also been assigned by following the PD algorithm.

Table 2. Scheduling parameters for the simple example

<table>
<thead>
<tr>
<th>τ_{ij}</th>
<th>FP-Only</th>
<th>EDF-1</th>
<th>EDF-2</th>
<th>EDF-All</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P_{ij}</td>
<td>P_{ij}</td>
<td>S_{d_{ij}}/SD_{ij}</td>
<td>P_{ij}</td>
</tr>
<tr>
<td>τ_{11}</td>
<td>3</td>
<td>-20</td>
<td>3</td>
<td>-3</td>
</tr>
<tr>
<td>τ_{21}</td>
<td>2</td>
<td>-120/20</td>
<td>2</td>
<td>-120/20</td>
</tr>
<tr>
<td>τ_{22}</td>
<td>3</td>
<td>3</td>
<td>-3</td>
<td>-3</td>
</tr>
<tr>
<td>τ_{23}</td>
<td>2</td>
<td>-90/362</td>
<td>-90/362</td>
<td></td>
</tr>
<tr>
<td>τ_{24}</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>τ_{25}</td>
<td>1</td>
<td>-181/750</td>
<td>-181/750</td>
<td></td>
</tr>
<tr>
<td>τ_{31}</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>30</td>
</tr>
<tr>
<td>τ_{41}</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>200</td>
</tr>
</tbody>
</table>

Table 3 shows the results of applying the MAST analysis tools to the proposed case studies. We have analyzed the FP-Only case-study by applying the offset-based (Off) [15], as well as the holistic (Hol) [25] techniques. For the other three case studies, we have used the schedulability analysis tools implemented in MAST that obtain the tightest response times, i.e., offset-based algorithm for FP [15] and holistic for EDF, both with their local (L) [19] and global (G) [22] variants. Worst-case response times are shown in order to allow the comparison among the different techniques.

The results show the same results for Γ_1 and Γ_3, and different results for Γ_2 and Γ_4 depending on the particular schedulers. The difference in the results is explained not only because of the different schedulers, but also the analysis techniques used. For instance, for fixed priorities the results are better because the offset-based analysis technique is used, while for EDF scheduling just the holistic analysis is used.

Table 3. Worst-case response times for the case studies

<table>
<thead>
<tr>
<th>τ_{ij}</th>
<th>FP-Only</th>
<th>EDF-1</th>
<th>EDF-2</th>
<th>EDF-All</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Off/Hol</td>
<td>L/G</td>
<td>L/G</td>
<td>L/G</td>
</tr>
<tr>
<td>τ_{11}</td>
<td>4/4</td>
<td>4/4</td>
<td>4/4</td>
<td>4/4</td>
</tr>
<tr>
<td>τ_{21}</td>
<td>28/28</td>
<td>28/28</td>
<td>28/28</td>
<td>28/28</td>
</tr>
<tr>
<td>τ_{22}</td>
<td>53/53</td>
<td>53/53</td>
<td>53/53</td>
<td>53/53</td>
</tr>
<tr>
<td>τ_{23}</td>
<td>73/73</td>
<td>73/73</td>
<td>83/193</td>
<td>83/193</td>
</tr>
<tr>
<td>τ_{24}</td>
<td>107/132</td>
<td>107/107</td>
<td>117/237</td>
<td>117/237</td>
</tr>
<tr>
<td>τ_{25}</td>
<td>145/198</td>
<td>173/173</td>
<td>183/275</td>
<td>183/303</td>
</tr>
<tr>
<td>τ_{31}</td>
<td>5/5</td>
<td>5/5</td>
<td>5/5</td>
<td>5/5</td>
</tr>
<tr>
<td>τ_{41}</td>
<td>160/160</td>
<td>160/160</td>
<td>160/120</td>
<td>160/120</td>
</tr>
</tbody>
</table>

In any case, the purpose of the test is not to show the merits of one scheduling technique or another, but to show that these techniques can be combined together with the composition rules defined in this paper, to analyze distributed systems with heterogeneous scheduling.

6.2. A Complex Example for the Assignment of Scheduling Parameters

In order to illustrate the performance of the HOSPA algorithm, we have chosen an example consisting of 5 processing resources (4 CPUs and 1 network) hosting 12 end-to-end flows composed of 30 steps (21 tasks executing in the CPUs and 9 messages sent across the network). As for the simple example, timing requirements have also been chosen proportionally to the period and the number of processing resources traversed by the end-to-end flow. Best-case execution times have been made equal to zero. The architecture of the example including the timing parameters (periods, worst-case execution times and deadlines) and the assignment of the steps to their corresponding processing resource are shown in Table 4.

It is assumed that the network splits large messages into packets; in this example the maximum packet transmission time has been set to 5 time units. Messages are preemptible at the packet boundary. The network is scheduled by an FP policy, with a range of priorities in the interval [1,10], while processors are scheduled by an EDF policy with a local-deadline or a global-deadline scheduler.

The results of applying HOSPA to this example are shown in Table 5. Scheduling parameters should be interpreted as scheduling deadlines for the steps executed in CPUs and as priorities (easy to distinguish because they take values between 1 and 9) for the network. Deadlines have been added to this table in order to see how far are the worst-case response times from them. We can notice that it
is easier to make the system schedulable using a global-deadline scheduler than a local-deadline one.

In order to better compare the results obtained, Table 6 presents the slacks for the system and for the individual processing resources when we apply HOSPA for the local and global-deadline schedulers and when we apply the
proportional deadline distribution (PD). The slack is defined as the percentage by which all the execution times of all the steps in the systems or in a processing resource may be increased while still keeping the system schedulable, if positive, and the percentage by which all the execution times of all the steps in the systems or in a processing resource have to be decreased to make the system schedulable, if negative.

Table 6. System and processing resource slacks (%)

<table>
<thead>
<tr>
<th>Kind of slack</th>
<th>Local PD</th>
<th>Local HOSPA</th>
<th>Global HOSPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>-3.13</td>
<td>0.78</td>
<td>9.38</td>
</tr>
<tr>
<td>CPU-1</td>
<td>-8.24</td>
<td>5.32</td>
<td>14.71</td>
</tr>
<tr>
<td>CPU-2</td>
<td>-98.44</td>
<td>14.20</td>
<td>14.20</td>
</tr>
<tr>
<td>CPU-3</td>
<td>-98.44</td>
<td>14.71</td>
<td>15.23</td>
</tr>
<tr>
<td>CPU-4</td>
<td>-6.23</td>
<td>3.21</td>
<td>14.71</td>
</tr>
<tr>
<td>Network</td>
<td>-6.91</td>
<td>3.62</td>
<td>12.70</td>
</tr>
</tbody>
</table>

We can see that HOSPA is able to optimize and find a schedulable assignment even when the simple PD assignment fails the schedulability test. The differences in slack are very significant. The Global EDF scheduler also gets much better slack results than Local EDF.

7. Conclusions

In this paper we propose a new mechanism for the integration of different response-time analysis techniques so that they can be applied to heterogeneous distributed systems with different scheduling policies in each resource. We showed that this technique was capable of calculating worst case response times for a simple example using FP and EDF schedulers.

We also adapted the HOPA/HOSDA algorithms into the new HOSPA technique for the assignment of scheduling parameters (priorities and both global and local scheduling deadlines) in a heterogeneous system. We showed its operation with a complex heterogeneous system. The results presented for this example show that the global EDF scheduler obtains higher slacks than local EDF which means that the global EDF system has a higher tolerance to possible errors in the estimation of worst-case response times, and more space for growth. In view of these results, we plan as a future work a more extensive evaluation of the heterogeneous techniques that we are presenting in this work, carrying out a study over a wider variety of case studies to try to determine in which cases we can leverage the benefits of heterogeneous scheduling.

In addition to integrating specific techniques for analyzing response times and optimization algorithms based on these techniques for heterogeneous systems, one of the main contributions of this paper is the formalization of the composition mechanisms, implemented through the concepts of inherited release jitters and inherited offsets, which will allow to accommodate new techniques for other schedulers in the future.

References


Table 6. System and processing resource slacks (%)


