

RESEARCH

Open Access

# ( $m,k$ )-firm pattern spinning to improve the GTS allocation of periodic messages in IEEE 802.15.4 networks

Tiago Semprebom<sup>1\*</sup>, Carlos Montez<sup>2</sup> and Francisco Vasques<sup>3</sup>

## Abstract

The IEEE 802.15.4 standard provides a flexible communication support for low-rate wireless personal area networks applications. When active, the beacon-enabled mode provides a real-time communication to the supported application by adopting a guaranteed time slot (GTS) mechanism. However, this mechanism permits only up to seven real-time communicating devices. One way to deal with this limitation is to share the communication opportunities among the periodic tasks, by skipping some of the task activations in a controlled way. One of the widely accepted periodic task models that allows skips in periodic activations is the ( $m,k$ )-firm model. Motivated by this problem, this paper proposes the use of a dynamic GTS scheduling approach based on the ( $m,k$ )-firm task model, to deal with the GTS starvation problem. The proposed scheduling approach is based on pre-defined spins of the originally defined ( $m,k$ )-firm pattern. The use of an exact schedulability analysis test ensures that for each admitted message stream, at least  $m$  messages will be transmitted within each window of  $k$  consecutive deadlines. The schedulability analysis may be executed in polynomial time and therefore can be used as an online admission test for GTS requests. The effectiveness of the approach has been assessed both by a set of simulations and an experimental evaluation.

## 1 Introduction

In the last few years, there has been a growth in the use of wireless technologies in application domains that require a trustworthy quality of service (QoS) [1-9]. It is expected that, in the near future, the availability of wireless network solutions will create a *de facto* standard for wireless communication in industrial environments. The IEEE 802.15.4 [10] is a strong contender to become the communication standard for low-rate wireless applications. Within this context, wireless sensor networks (WSNs) are targeting a number of applications ranging from military applications to modern healthcare.

This paper addresses WSNs applied to industrial communication scenarios, for example, the network control applications, where control data must be periodically transferred between sensors, controllers, and actuators in accordance with strict and well-defined timing constraints. More specifically, we investigate some of the real-time properties of the IEEE 802.15.4 standard. This

standard was not specifically developed for WSNs, but it may be appropriate for the purpose. Characteristics such as low energy consumption, low transmission rate, and low cost typically fit within the needs of WSNs.

Periodic real-time applications usually assume that task activations will be executed before their deadlines. For applications that exchange real-time messages, this implies that periodically sent messages need to reach their destinations before the related deadlines. However, there is a subset of real-time applications which do not require that all deadlines are met. Some relevant examples can be found in automatic control and multimedia domains [11,12]. Some works [9,13,14] investigate the effects of activation discards in the system performance and propose the use of novel techniques for periodic task discards, as long as those discards occur in a sparse fashion.

In an IEEE 802.15.4 network operating with beacon mode, for example, a periodic task can assume that its period is coincident with the period of the superframe and it may wish to use at least one slot per superframe. On the other hand, for another task running a report, every two or

\*Correspondence: tisemp@ifsc.edu.br

<sup>1</sup>Federal Institute of Santa Catarina (IFSC), São José 88130-310, Brazil  
Full list of author information is available at the end of the article

three consecutive superframes may be enough. This tolerance is application dependent. For example, it may be possible to apply interpolation or estimation techniques like Kalman filters [13,15] to compensate for not receiving some of the periodic messages.

The tolerance to activation discards can be expressed as a percentage of its maximum loss rate. Nevertheless, the specification of loss rate as a percentage depends on the window size within which the constraint is tested, which can be insufficient for some applications. For example, a task with a constraint of at most 30% of deadline misses would be allowed to miss up to 30 consecutive deadlines for each 100 activations without a failure.

In order to support QoS guarantees in IEEE 802.15.4 networks, there is the need to investigate the real-time characteristics of its physical (PHY) and media access control (MAC) layers [2,6,16]. Within this context, one of the most interesting features is the guaranteed time slot (GTS) mechanism. When the network is operating in a beacon-enabled mode, i.e., beacon frames are periodically transmitted for synchronizing the network devices, the IEEE 802.15.4 protocol allows the reservation of bandwidth for a specific subset of devices, providing a maximum service interval through the allocation of fixed GTSs.

The use of the GTS mechanism as proposed in the standard allows the allocation of just up to seven slots during the contention-free period (CFP) of each superframe, whose duration corresponds to 16 time slots. As each GTS node performs an explicit allocation by asking the personal area network (PAN) coordinator for a specific number of time slots; the number of nodes with allocated GTSs is limited to seven or less. This scarce resource is rapidly extinguished when considering networks with a large number of nodes. As a consequence, devices that are not able to have any allocated GTS should try to accomplish their transmissions during the contention access period (CAP), according to the restrictions imposed by this contention-based period.

However, the CAP is not able to provide any transmission guarantee due to the possibility of message collisions; therefore, its use is not adequate to support applications with timing constraints.

We propose a novel approach to deal with this limitation of IEEE 802.15.4 for real-time applications, where the GTS slots are allocated to nodes according to a (m,k)-firm scheduling strategy, guaranteeing that each node is able to transfer  $m$  messages within each window of  $k$  consecutive deadlines. Additionally, we also provide an exact schedulability test, where task constraints are expressed according to a (m,k)-firm model. The PAN coordinator runs an admission control algorithm based on the (m,k)-firm information and the amount of available resources, ensuring the (m,k)-firm constraints of the nodes. It is important to note that other works could also be used

within the same context [17-19]. However, the schedulability test proposed in [17] is just a sufficient test that has a high rejection ratio for tasks that could be otherwise scheduled, and the approaches proposed in [18,19] could only be used for IEEE 802.15.4 offline scenarios due to the high computational complexity of its schedulability tests.

The remainder of this paper is structured as follows. Section 2 provides an overview of the IEEE 802.15.4 medium access protocol and presents some of the most relevant related works that could be found in the literature. Section 3 presents a formal description of the problem. A schedulability test to provide deterministic (m,k)-firm guarantee is discussed in Section 4. The effectiveness of the proposed approach is highlighted in Section 5. Finally, conclusions are discussed in Section 6.

## 2 Background

### 2.1 IEEE 802.15.4 overview

The IEEE 802.15.4 standard [10] specifies the sub-layer medium access (MAC) and the physical layer for low-rate wireless personal area networks. The MAC protocol supports two operation modes to be selected by the coordinator:

1. *Non beacon-enabled mode*: in this operation mode, devices may simply send data according to a non-slotted carrier sense multiple access with collision avoidance (CSMA/CA) approach.
2. *Beacon-enabled mode*: in this operation mode, beacons are periodically sent by the PAN coordinator to identify its PAN and to synchronize nodes that are associated with it.

In this paper, we focus on the beacon-enabled mode, as this mode is specifically suited to provide timeliness guarantees for time-sensitive applications. When the coordinator selects the beacon-enabled mode, it forces the use of the superframe structure to manage communication between devices associated with the PAN. The superframe is held by a beacon interval (BI) and limited by two consecutive beacon frames, which includes an active period (divided into 16 equally sized time slots) and an inactive period (Figure 1). The coordinator interacts with the PAN devices during active periods, and those devices may enter into a saving energy mode (sleep mode) during inactive periods.

The BI and the superframe duration (SD) are determined by two parameters, the beacon order (BO) and the superframe order (SO), respectively, and are given by:

$$BI = aBaseSuperframeDuration \times 2^{BO} \quad (1)$$

$$SD = aBaseSuperframeDuration \times 2^{SO}, \quad (2)$$

for  $0 \leq SO \leq BO \leq 14$

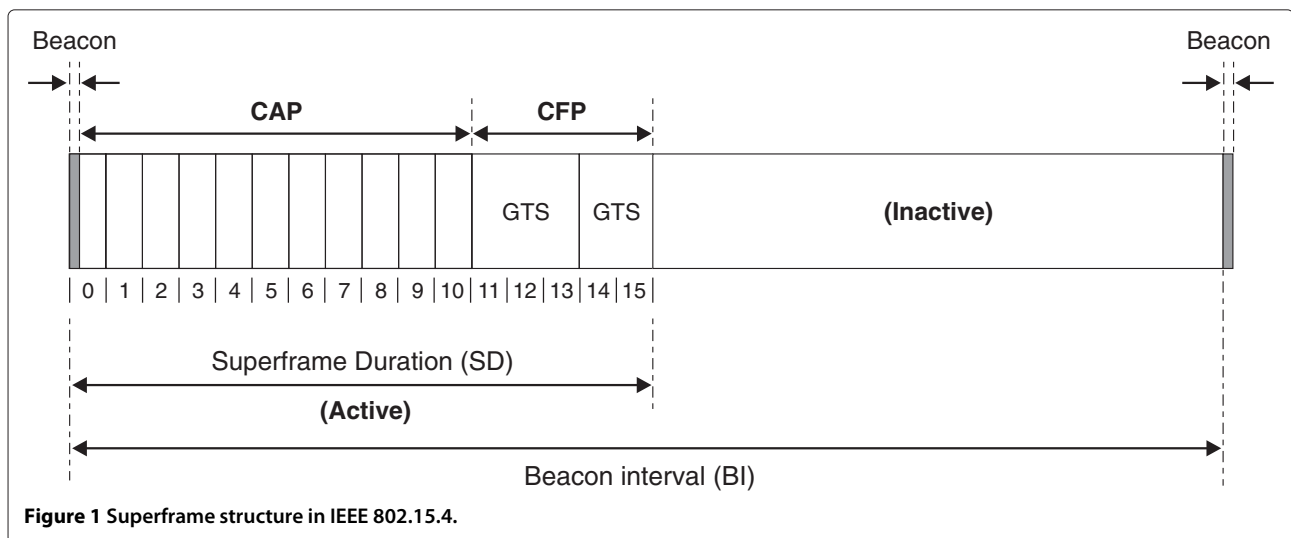


Figure 1 Superframe structure in IEEE 802.15.4.

In Equations 1 and 2,  $aBaseSuperframeDuration$  denotes the minimum duration of the superframe. The IEEE 802.15.4 standard sets this duration to 960 symbols (a symbol corresponds to 4 bits). This value corresponds to 15.36 ms, assuming a data rate of 250 kbps, in the 2.4-GHz ISM frequency band, which will be considered for the remainder of this paper.

The active portion of each superframe is composed of three parts: beacon, CAP, and CFP. The beacon shall be transmitted without contention periods (and without CSMA/CA) at the start of first slot. The CAP must begin immediately after the beacon and has a minimum length corresponding to nine slots. The CFPs, if any, follow immediately after the CAP and extend the end of the active portion of the superframe. All the allocated GTSs should be located within the CFP. Figure 1 illustrates a superframe structure with two allocated GTSs.

If the application exclusively uses the CAP, a device that wants to communicate must compete for the medium access using slotted CSMA/CA, which does not provide any timing guarantee. Thus, using just the CAP is not suitable to support real-time applications.

## 2.2 Related work

The GTS allocation mechanism in IEEE 802.15.4 networks allows a device to access the communication medium without contention periods. The PAN coordinator is responsible for the allocation and determines the length of the CFP in a superframe. The maximum number of GTSs that can be allocated by the PAN coordinator is seven, and by default, the slots are allocated on a first-in, first-out (FIFO) basis. However, recently, several works have been developed aiming to enhance this allocation mechanism [2-4,9].

The work presented in [2] improves the use of bandwidth allowing multiple devices to share the same set of GTSs through a round-robin scheduling algorithm. However, the underlying model assumes two important assumptions, which limits the applicability of the proposed solution: (1) the candidates sharing the same slot have similar arrival rates and (2) tasks do not use the previously reserved resources in all superframe periods.

In [3], the authors proposed a scheduling approach, where the CFP is always splitted into 16 mini slots. The basic idea is to exploit the fact that the GTS descriptor has a structure not fully utilized in the standard scheme and to propose a new mapping to represent nine new slots. The nodes in the PAN must be aware of this new mapping for the GTS allocation. Following the same ideas, in [4], the authors presented a more flexible approach and proposed a CFP divided into more than 16 mini time slots for periodic real-time message allocation. This proposal offers tighter delay bounds and improves the GTS utilization through an off-line bandwidth allocation algorithm. With both approaches [3,4], it is possible to extend the GTS allocation to support applications that require a slightly higher number of slots than the original IEEE 802.15.4 specification.

The problem of providing some guarantee for cycling tasks assuming environments where eventual discards in periodic activations may occur has been addressed by a considerable amount of work [1,9,17-26]. In general, these studies attempt to develop scheduling strategies that can offer some guarantee for tasks, considering their tolerance to periodic activation losses.

The work in [20] describes an algorithm for scheduling packets in real-time multimedia data streams called dynamic window-constrained scheduling (DWCS). This

algorithm considers the tolerance of these packets with respect to the deadline misses. The window constraint defines the number of packet deadlines that can be missed in a window of deadlines for a set of consecutive packets in a stream. The algorithm attempts to guarantee that no more than  $x$  out of a window of  $y$  deadlines are missed. A schedulability test was proposed to ensure the absence of failures. However, the presented algorithm is not exact, just sufficient. Moreover, the fact that DWCS uses earliest deadline first (EDF) as the first criterion for tie breaks takes the DWCS to behave like EDF when the system load is smaller or equal to 100%.

The work proposed in [21] presents a schedulability test for a time division multiple access (TDMA) scheduling with slot skipping (TDMA/SS). The authors assume that all message streams in the system are known, and that each node schedules messages in its output queue according to the deadline monotonic algorithm, in contrast to some other policy approaches that use FIFO queues of messages in each node. According to the proposal, when a node finishes its transmissions, it sends a completion message transmission notifying that there are no more messages to transmit, so a node that has real-time messages to be transmitted can immediately start their own transmissions. In a traditional TDMA protocol, a node that does not wish to transmit messages simply remains silent during the length of the reserved time slot.

The work entitled *Skip-over* [22] assumes a task model composed of periodic tasks that tolerate discards (called skips). These skips are characterized by a parameter  $s$ , which represents the tolerance of the task for deadline misses. The distance between two consecutive skips must be at least  $s$  periods. The work proposes a sufficient schedulability test where, in some cases, tasks are unnecessarily rejected. Similarly, in the mid-1990s, the work [23] introduced the notion of  $(m,k)$ -firm deadlines. According to this classical model, service guarantees are applied to tasks such as packet streams or periodic tasks<sup>a</sup>. The presented algorithm uses distance-based priority (DBP) as prioritization heuristic. The task that is closer to lose the  $m + 1$ th deadline in a window of  $k$  service requests will receive the highest priority of the system (this model tolerates up to  $k - m$  discards). The DBP does not offer any timing guarantee for the task execution. Tasks could have its  $(m,k)$ -firm constraints violated, occurring the so-called dynamic failure.

The problem of providing periodic task services guarantees using the  $(m,k)$ -firm task model has been addressed by many other research works. Some of them propose incremental improvements to the DBP in order to reduce the number of dynamic failures [25,26]; other works propose schedulability tests aiming to guarantee that the tasks accepted in the system will not have any  $(m,k)$ -firm constraints violation [1,9,17-19,24].

In [25], the authors assess some of the drawbacks of the DBP policy and propose some improvements. The fact that the DBP approach does not take into account other information than the distance to failure, underestimating the information about the met/miss history of other tasks is considered just a 'local strategy' and a weakness of this seminal approach. The authors propose the integrated DBP (IDBP) where, besides the distance to failure, a new concept named restoring distance is taken into account. The restoring distance of a task in a failure state is the minimum number of deadlines that need to be met in order for a task to leave this state. The IDBP shows better results than DBP; although, it is important to note that both DBP and IDBP do not provide schedulability test, therefore tasks may have its  $(m,k)$ -firm constraints violated. Another work aimed to improve the DBP is the GDPA [26]. The proposed scheme has been designed to provide an upper-bounded probability for the dynamic failures of tasks with  $(m,k)$ -firm deadlines when the system is underloaded. The proposed algorithm takes scheduling decisions based on DBP combined with EDF.

The work [18] presents an exact schedulability test for applications with  $(m,k)$ -firm constraints, comparing at each hyper-period the system state with the previous hyper-period states. It stops once the schedule is repeated. In other words, in the worst case condition, all the possible combinations of the  $(m,k)$ -firm pattern for all tasks in the set must be tested. The computational complexity of this test is not adequate for systems with dynamic load. Moreover, the work uses DBP as scheduling algorithm which is not optimal in the sense that DBP assignment is only based on the distance to failure of each individual stream under its own  $(m,k)$ -firm constraint. It does not consider other task parameters as period, deadline, or service times.

The work presented in [19] proposes a heuristic for executing  $(m,k)$ -firm tasks using a genetic algorithm. It also presents a sufficient test where by manipulating task periods, it generates a new task set with harmonic periods. The proposed solution is only suitable for off-line approaches, where the probabilistic genetic algorithm tries to find a partitioning between mandatory and optional feasible tasks, thus producing a possible execution schedule for the task set.

In [24], a task model with  $(n,m)$ -hard constraints is presented. The focus of the work is to improve the responsiveness of best effort tasks in the presence of tasks with  $(n,m)$ -hard constraints. An non-negligible disadvantage of this approach is that it requires a scheduling test (an off-line test) to ensure the schedulability of tasks with hard real-time constraints.

In [1] is a proposed sufficient schedulability analysis based on a classical test [27] for periodic and sporadic

real-time tasks. The work considers a non-preemptive scheduling based on DBP priority assignment algorithm. The approach was evaluated in an in-vehicle control system scenario. The major disadvantage of the proposed schedulability test is that it is only sufficient (pessimistic), and it rejects some task sets that are schedulable.

A dynamic GTS allocation model in IEEE 802.15.4 networks was introduced in [9]. The proposed approach ensures that at least  $m$  messages are allocated in GTSs for any window of  $k$  consecutive superframes. It adopts a deterministic schedulability test and an online prioritization heuristic algorithm called slotted DBP (distance-based priority) to attend requests. This model offers guarantees for applications with (m,k)-firm constraints with two important restrictions: all tasks must have both the same period and the same  $k$  value.

The approach proposed in this paper addresses the scheduling of real-time tasks with (m,k)-firm deadlines running over IEEE 802.15.4 devices. The main contributions of this paper are as follows: (1) the proposal of a new exact schedulability test, regarding specific (m,k)-firm classification patterns, that is able to take schedulability decisions for the mandatory tasks in polynomial time, (2) a new task classifier, based on [17], that accepts pattern spinning as a parameter for tasks with (m,k)-firm constraints, and (3) an experimental assessment of the proposed approach that considers typical environment overheads and physical constraints imposed by sensor devices.

### 3 System model

For a subset of real-time applications, (e.g., process control or multimedia), it may be admissible that some of its deadlines are missed. For this type of applications, it is required that a precise specification of how deadlines can be missed and how those misses must be distributed in time. The (m,k)-firm task model can be used to cope with these type of applications [23]. Using this model, a system can be designed to tolerate deadline misses, provided that the number of deadline misses is bounded and precisely parsed.

Consider a WSN supported real-time application, where tasks are distributed among sensor nodes. Consider also that communicating tasks generate message streams that must be supported by GTSs. Therefore, whenever more than seven GTS allocation requests arrive to the PAN coordinator, and each request has a (m,k)-firm specification, the PAN coordinator will be required to perform a (m,k)-firm allocation operation to accomplish it. Therefore, the problem addressed in this paper is how can the PAN coordinator allocate the maximum number of GTSs during the CFP, among  $N$  requesting sensor nodes, respecting the (m,k)-firm constraints of each of the

message streams. Formally, this problem can be stated as follows:

Consider a set  $\Gamma = \{\tau_1, \tau_2, \dots, \tau_n\}$  of  $N$  independent periodic communicating tasks. Each task  $\tau_i$  generates a message stream  $S_i$  characterized by a 5-tuple  $((m_i, k_i), C_i, P_i, D_i)$ , where  $C_i$  represents the worst case message duration,  $P_i$  represents the task (message stream) period,  $D_i$  denotes the job (message) deadline, and  $m_i$  and  $k_i$  denote the (m,k)-firm constraints of task  $\tau_i$  (message stream  $S_i$ ). In the case of the proposed model, a periodic task represents an application that runs in a network node, which generates a message stream. In addition, we make the following assumptions about tasks (message streams):

- A1.  $1 \leq m_i \leq k_i$ ,  $m_i$  and  $k_i \in \mathbf{Z}^+$ .
- A2.  $D_i = P_i$ : the deadline of task  $\tau_i$  (message streams  $S_i$ ) is equal to its period.
- A3.  $C_i = l_i \cdot ut$ ,  $l_i \in \mathbf{Z}^+$ : execution time (message duration) is a multiple of the slot length (ut).

The rationale for assumption A3 is that due to the preemptive nature of the proposed model, a message stream can transfer more than one message per period ( $C \geq 1$ ) but cannot guarantee that those messages will be contiguously transferred. For instance, a message stream can transmit two messages with duration shorter than one time slot but cannot transmit one message with the duration of two time slots. The duration of the time slots, as well as the BI value, can be flexibly adjusted through the BO and SO parameters controlled by the PAN coordinator.

In this work, it is considered an IEEE 802.15.4 cluster composed of  $N$  sensor nodes in the range of a node designated as the PAN coordinator. It is assumed that the PAN coordinator sets up the network determining the superframe structure through the BO and SO parameters [10]. Each sensor node  $i$  that needs an allocated GTS for a specific period of time sends a GTS request with its (m,k)-firm constraints to the PAN coordinator. An error-free channel was assumed establishing an upper bound on channel utilization. If an error-prone channel is considered [28], the maximum channel usage will be smaller. Nevertheless, this type of analysis is out of the scope of this paper.

In the remainder of this paper, the terms task and message stream (job and message) will be indistinctly used.

### 4 Proposed scheduling approach

In this section, we firstly introduce a classification algorithm to classify the task activations (jobs) in mandatory or optional. Then, we present an admission test to check the admissibility of the tasks according to the (m,k)-firm

guarantees, where the mandatory activations must be always schedulable during the CFP.

#### 4.1 Task classifier and (m,k)-firm pattern spins

In [17], the author presented a job classifier that dynamically determines whether jobs are mandatory (1) or optional (0). Jobs classified as mandatory will have their deadlines guaranteed. On the other hand, jobs classified as optional will receive lower priorities and are not guaranteed to meet their deadlines. The classification pattern of a task can be considered as a ring. Whenever a new task activation is done, a clockwise spin is performed at the ring (right spin), indicating whether the instance of the task is mandatory or optional. Figure 2 illustrates the classification pattern for a task (2,3)-firm.

Despite the simplicity of the presented algorithm, it has some limitations. The first job of each task is always classified as mandatory, which implies that the worst case execution time of any task will always occur during its first period. This means that the classifier is not able to consider any task discard in its first activation without failure. Therefore, the behavior of the classifier proposed in [17] inserts a significant pessimism in the schedulability analysis. This peculiarity can be viewed in Figure 3 where a simple task set composed of two tasks  $\tau_1$  and  $\tau_2$  with  $C_1 = C_2 = 1, P_1 = P_2 = 1, m_1 = m_2 = 1, k_1 = k_2 = 2$  cannot be scheduled according to [17] (Figure 3a). Nevertheless, a simple spin in the job classification pattern in  $\tau_2$  would generate a feasible schedule (Figure 3b).

According to [19], the optimal classification of jobs is a problem that cannot be treated in polynomial time (NP hard) for arbitrary service times, request periods, and  $k$  window values. In this work, we propose a variation of the task classifier introduced by [17], where spins in the original (m,k)-firm pattern are allowed before the task execution. In the proposed approach, a (m,k)-firm pattern may be changed by an anti-clockwise spin (this left spin operation is equivalent to a rotate-left instruction applied to the (m,k)-firm classification pattern). For example, according to [17], a task with (1,3)-firm constraints

produces a (m,k)-firm pattern '100', where the first job of the task is classified as mandatory and the two subsequent jobs are classified as optional. If additional left spins are admitted, new (m,k)-firm patterns can be considered, limited by  $(k - 1)$  spins. The new (m,k)-firm pattern alternatives for a task (1,3)-firm are one left spin produces a (m,k)-firm pattern = '001' and two left spins produces a (m,k)-firm pattern = '010'.

The jobs classification as mandatory or optional is based on the  $m_i$  and  $k_i$  values of each task. Thus, a job of task  $\tau_i$  activated at instant  $w.P_i$  is classified as mandatory if Equation 3 is satisfied and as optional otherwise.

$$w = \left\lfloor \left\lceil \frac{wm_i}{k_i} \right\rceil \times \frac{k_i}{m_i} \right\rfloor, \quad \text{for } w = \left\lfloor \frac{t}{P_i} + s_i \right\rfloor \quad (3)$$

The term  $t$  refers to the tested arrival time. The term  $s_i$  represents the spin value for task  $\tau_i$ . This value is equal to zero whenever the (m,k)-firm pattern spinning is not implemented.

#### 4.2 Schedulability analysis

There are several works in the literature that have adopted a dynamic priority assignment to deal with (m,k)-firm scheduling approaches [18,19,23,24]. Nevertheless, it often occurs that a fixed priority scheduling can be more attractive than a dynamic priority one, due to its implementation simplicity and lower overhead. This is particularly true in the case of wireless sensor networks due to the inherent limitations in processing power and energy consumption that must be dealt with when implementing applications upon sensor nodes. Therefore, in this work, we adopted a fixed priority scheduling algorithm for the (m,k)-firm task model.

The proposed schedulability test takes scheduling decisions online. Thus, whenever there is a request to consider a new task, it is necessary to perform a schedulability test (Algorithm 1), considering the execution of this new task together with the already accepted tasks. The new incoming task will be accepted only if it does not jeopardize the guarantees offered to the previously accepted tasks.

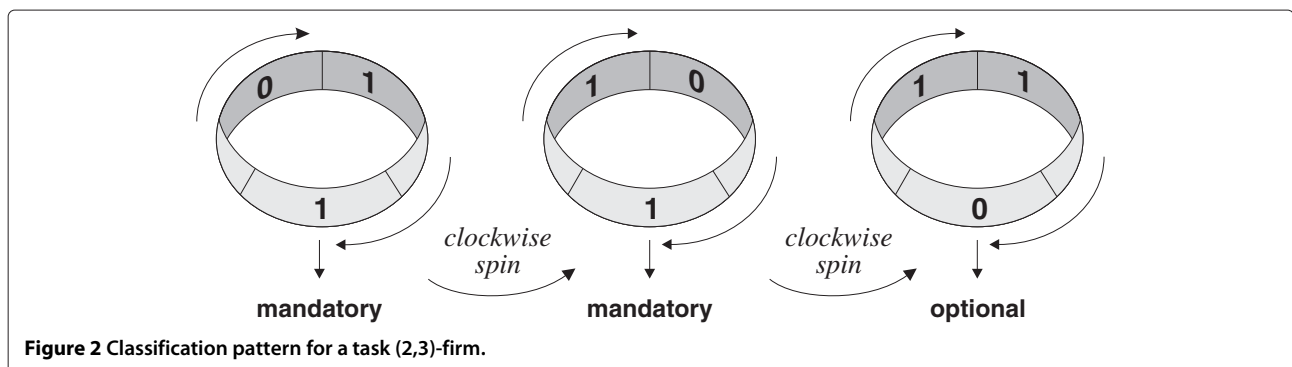
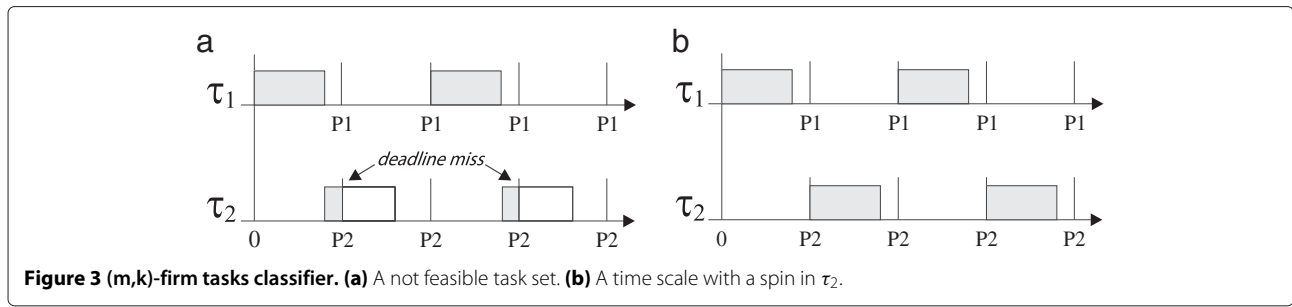


Figure 2 Classification pattern for a task (2,3)-firm.



The schedulability test proposed in this paper is based on the busy period concept [29,30]. Consider the release time of a task  $\tau_i$ . From that point, until the task is completely executed, the processor will be executing processes with priority  $i$  or higher. The processor is said to be executing a  $i$  - busy period. For a given task  $\tau_i$ , the busy period calculation is performed, considering the execution of tasks  $\tau_j$  ( $j < i$ ). The calculation starts at the arrival time of task  $\tau_i$  and is iteratively executed until the calculated busy period value converges or exceeds the deadline of the task. In other words, the iteration starts with  $W_i(0) = C_i$  and terminates when  $W_i(t + 1) = W_i(t)$ . If the iteration converges and  $W_i(t) \leq D_i$  for all mandatory activations that were tested, the task set is said to be feasible. Otherwise, if  $W_i(t) > D_i$ , the task set is unfeasible.

$$W_i(0) = C_i \quad (4)$$

$$W_i(t + 1) = (C_i - \Delta_i) + \sum_{j=1}^{i-1} \left\lceil \frac{W_i(t) + (\phi_i - \phi_j)}{P_j} \cdot \frac{m_j}{k_j} \right\rceil C_j \quad (5)$$

Equation 5 shows the busy period calculation, considering tasks with (m,k)-firm constraints, where:

$W_i$  : represents the evaluated  $i$ -busy period at time instant  $t$ .

$\Delta_i$  : represents the offset between  $\phi_i$  and  $\phi_j$ , that is, the difference between the arrival time of  $\tau_i$  and the arrival time of the  $\tau_j$ , that affect the tested period.

$\phi_i$  : represents the arrival time of task  $\tau_i$ .

$\phi_j$  : represents the arrival time of task  $\tau_j$ .

Equation 5 evaluates the  $i$ - busy period considering the execution of all mandatory activations of higher priority tasks tested during each time interval (line 21, Algorithm 1). Concerning the interference duration considered by the proposed schedulability test, it is necessary to start the iteration at the time instant where tasks with higher priority actually start to generate interference. This

starting time is calculated (lines 5 and 6, Algorithm 1) and can be viewed in Figure 4 as  $\Delta_i$ .

To obtain the  $i$ -busy period, we solve the equation iteratively. The iteration ends when  $W_i(t + 1) = W_i(t)$  or  $W_i(t) > D_i$ . This condition is referred as convergence of the iteration (line 7, Algorithm 1).

A straightforward implementation of the schedulability algorithm proposed in this paper takes  $O(s(N^2 \cdot \text{lcm}^2))$  time, where  $s$  represents the number of spins implemented in the activation pattern,  $N$  represents the number of the tasks in the task set, and  $\text{lcm}$  represents the least common multiple  $\text{lcm}[(k_i \cdot P_i), (k_j \cdot P_j)]$ .

---

#### Algorithm 1 Schedulability analysis.

---

```

1: procedure Scheduler
2: for  $i = 0 \rightarrow N$  do
3:   for  $j = 0 \rightarrow \text{task}[i].\text{arrival.length}$  do
4:      $W_i \leftarrow \text{task}[i].C$ 
5:      $\text{start\_time} \leftarrow \text{GetStartTime}$ 
6:      $\Delta_i \leftarrow \text{task}[i].\text{arrival}[j] - \text{start\_time}$ 
7:     while  $W_i \leq \text{task}[i].\text{period}$  and  $W_i$  don't converge do
8:        $W_i \leftarrow (\text{task}[i].C - \Delta_i) + \text{hpDemand}$ 
9:     end while
10:    if  $W_i > \text{task}[i].\text{period}$  or  $W_i$  don't converge then
11:      return unfeasible
12:    end if
13:  end for
14: end for
15: return feasible
16: end procedure

17: procedure hpDemand
18: for  $j = 0 \rightarrow j < i$  do
19:    $\text{activations} \leftarrow \left\lceil \frac{W_i + (\phi_i - \phi_j)}{\text{task}[i].\text{period}} \right\rceil$ 
20:    $\text{sum} \leftarrow (\text{activations} - \text{task}[j].\text{optional}) \times \text{task}[j].C$ 
21: end for
22: return sum
23: end procedure

```

---

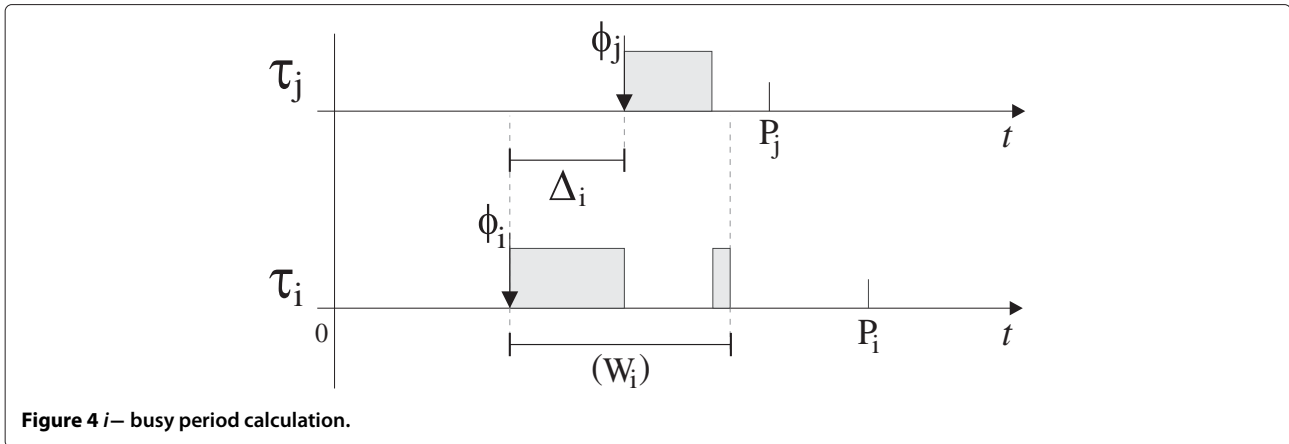


Figure 4 *i* – busy period calculation.

### 4.3 Proof of the schedulability test

This subsection aims to prove the schedulability test proposed in this paper. For the sake of convenience, the proof is divided in three parts, where the objective is to demonstrate that (1) if all instances of a task classified as mandatory have their deadlines met, then the  $(m,k)$ -firm constraints of the task are met, (2) for each time instant  $t$  tested, if  $t$  is the time instant of the arrival of a mandatory task, then Equation 5 is valid, and finally, (3) the schedule of the tasks is repeated after  $\text{lcm}[(kiPi), (kjPj)]$ .

In what concerns part (1), the structure of the proof is similar to the one presented in [17] considering the existence of  $s$  pattern spins in the  $(m,k)$ -firm specification. As the  $s$  constant inserted in Equation 3 does not change the structure of the proof in [17], it will not be repeated here.

The second part (2) of the proof is expressed by the following theorem:

**Theorem 1.** *For each tested time instant  $t$ , if  $t$  is the arrival time of a mandatory instance of  $\tau_i$ , then Equation 5 is valid.*

*Proof.* Considering Equation 5, any mandatory activation of  $\tau_j$  has higher priority than mandatory instances of  $\tau_i$ . The second term of Equation 5 represents the sum of all mandatory activations of  $\tau_j$  that have been activated until  $t$ .  $W_i(t)$  is the sum of the execution time of  $\tau_i$  added to the execution times of all mandatory activations of  $\tau_j$ , with priority higher than  $\tau_i$ , that occurs until  $t$ . So, if  $W_i(t + 1) = W_i(t)$  and  $W_i(t) \leq D_i$ , then the deadline of  $\tau_i$  for time instant  $t$  is accomplished.

In the  $(m,k)$ -spin algorithm-where it is only considered the spin of the classification pattern of the last task of the set-this theorem still holds because only the last task of the set (task  $\tau_i$ , lower priority) will be spun, which does not interfere with other tasks  $\tau_j$  with higher priority. Therefore, we must only verify if Equation 5 still holds if higher priorities tasks  $\tau_j$  are spun.

It is possible to verify this because concerning any time instant  $t$ , between the interval  $(kiPi)$  of a task  $\tau_i$  ( $j < i$ ), the number of mandatory activations of a task  $\tau_j$  already spun will never be greater than the number of mandatory activations obtained when the classification pattern has not been spun (0-spin). Figure 5 illustrates this scenario. The number of mandatory activations computed by the 0-spin pattern (top of figure), for any time  $t$  of  $\tau_j$  tested, will never be smaller than the number of mandatory activations computed until the same time instant  $t$ , in the other spin pattern situations (1-spin, 2-spin, 3-spin, or 4-spin pattern). Therefore, the  $(m,k)$ -spin algorithm is still valid. □

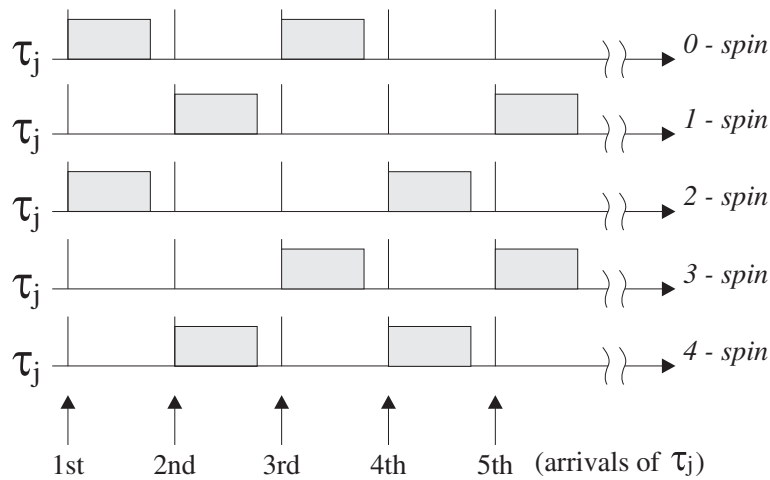
The third part of the schedulability test demonstration is illustrated in Theorem 2.

**Theorem 2.** *The system schedulability must be checked until  $\text{lcm}[(kiPi), (kjPj)]$ .*

*Proof.* In a set of real-time tasks, the critical instant of a priority level  $i$ , as proposed by [31], is the time instant when activations in level  $i$  are released together with all activations of higher priority. The task classifier presented by [17] force that all mandatory activations of  $\tau_i$  are released at the beginning of their periods, and the worst case for the system load occurs at  $t = 0$ . Therefore, as described in [32], in this case, it would be sufficient to check the schedulability only until the first deadline (first period) of each task. Nevertheless, the approach presented in this paper considers that task classification patterns can be spun in accordance with a well-defined classification pattern (bounded by  $k_i - 1$  spins). Therefore, the worst case generated by this task classifier does not necessarily occur at  $t = 0$ , and it is necessary to check other periodic instances of the task and not just the first one.

Therefore, one question that arises is when should we stop the checking routine? In [33], it is proved that for





**Figure 5** Possible arrival times of  $\tau_j$  with (2,5)-firm constraints.

the cases where all task activations are simultaneously released at  $t = 0$ , the worst case load will occur again in multiples of  $H_i = \text{lcm}(P_i \mid 1 \leq j \leq i)$ . In other words, the least common multiple of all values is from  $P_1$  to  $P_i$ . Assuming that the (m,k)-firm classification pattern of tasks are repeated at intervals with periodicity  $(k_i P_i)$ , similarly the worst case load is repeated in multiples of  $H_i = \text{lcm}(k_i P_i \mid 1 \leq j \leq i)$ . Thus, it is necessary and sufficient to verify the end of the busy period of  $\tau_i$  from the arrival time  $t$  to  $H_i$ .  $\square$

#### 4.4 Application example

The integration of packet discard techniques within industrial network applications—specifically, control applications—has already more than a decade. As pointed out in an important panel in the control area [34], in the past, control applications were centralized and based on synchronous systems. This panel concluded with the recommendation that the ‘theory and practice for control systems should be improved to applications that operate in a distributed, asynchronous, packet-based environments’.

Actually, this recommendation was already being addressed by some researches [12,17,35,36]. As aforementioned, [17] showed that control applications could deal with intentional message discards. In the same way, [35] and [36] proposed techniques for scheduling and control codesign of network control systems, i.e., the integration of message scheduling together with the design of the controller. These latter researches also proposed metrics for assessing the quality of control, which allows the design of techniques and strategies to adjust the packet sending rate, according to the desired control performance. This concept was highly relevant to work [12] that assessed the impact of packet

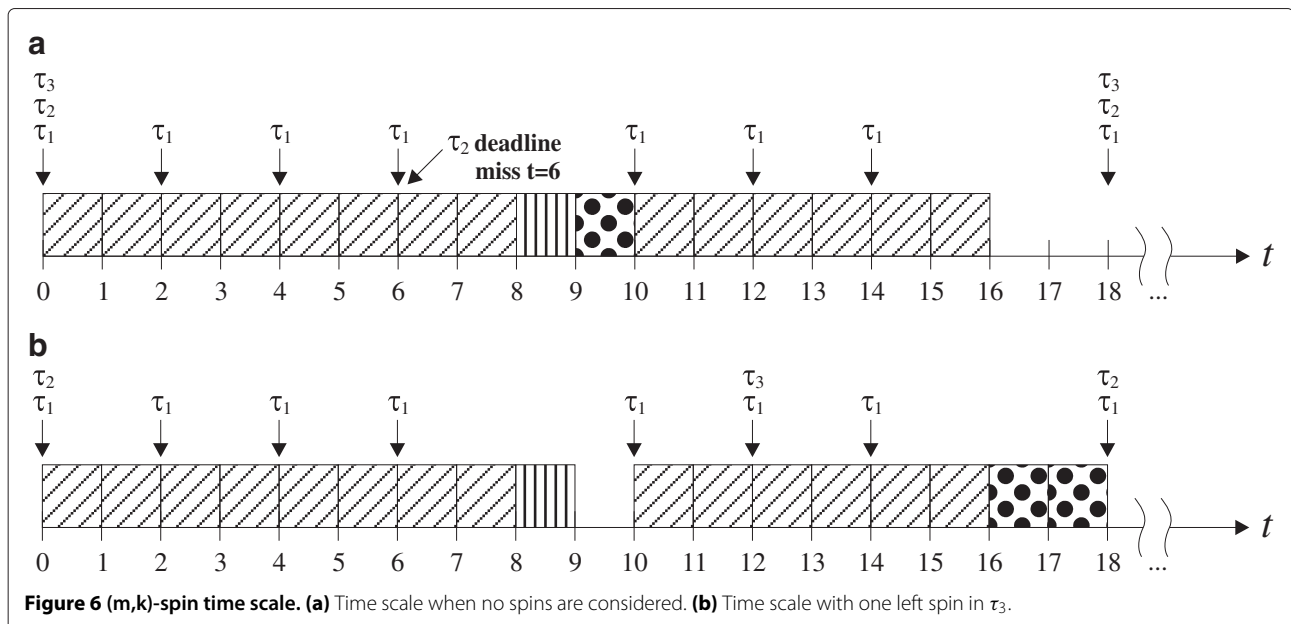
discards according to the (m,k)-firm policy in control applications.

Within the same context, this paper proposes an improvement for the GTS allocation scheme, adopting a (m,k)-firm model enhanced with spins in the (m,k)-firm pattern. It is important to note that a similar application example as in [17] could be used to illustrate the proposed approach. However, the example in [17] intentionally brings a set of tasks with harmonic periods (i.e., multiple periods), a fact which simplifies the schedulability test. In the sequence, we will present a more complex example that would be rejected by the test in [17].

As an example, consider a network control application composed of three periodic tasks with the following parameters:

$$\begin{aligned} \tau_1: C_1 = 2 \quad P_1 = 2 \quad m_1 = 7 \quad k_1 = 9 \quad s_1 = 0, \\ \tau_2: C_2 = 1 \quad P_2 = 9 \quad m_2 = 1 \quad k_2 = 2 \quad s_2 = 0, \\ \tau_3: C_3 = 2 \quad P_3 = 6 \quad m_3 = 1 \quad k_3 = 3 \quad s_3 = 1. \end{aligned}$$

According to Equation 3, task  $\tau_1$  has ‘111101110’ as (m,k)-firm pattern, which means that instances with arrival times 0,2,4,6,10,12,14,... are classified as mandatory, whereas instances with arrival time 8 and 16 are classified as optional. Task  $\tau_2$  has ‘10’ as (m,k)-firm pattern, implying that instances with arrival times 0,18,36,... are classified as mandatory and instances with arrival time 9,27,45,... are classified as optional. For task  $\tau_3$ , a (m,k)-firm spinning is necessary ( $s_3 = 1$ ), resulting in a ‘001’ (m,k)-firm pattern. Considering this left spin,  $\tau_3$  has activations with arrival times 12,30,48... classified as mandatory, whereas those with activation times 0,6,18,24,36,42... are classified as optional. Nevertheless, this task set could



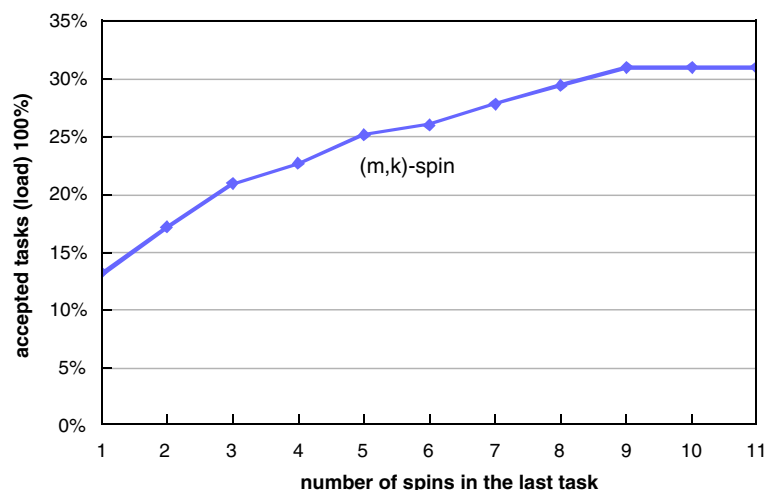
**Figure 6** (m,k)-spin time scale. (a) Time scale when no spins are considered. (b) Time scale with one left spin in  $\tau_3$ .

be considered to be schedulable if and only if a spin has been applied to  $\tau_3$ .

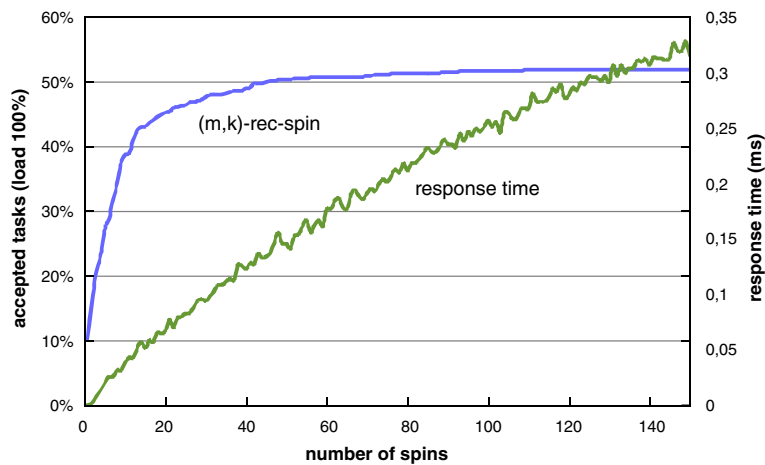
Figure 6 illustrates the time scale for this example. Figure 6a considers that none of the tasks in the set performs any spins on its (m,k)-firm classification pattern. Thus, task  $\tau_3$  has a deadline missed at  $t = 6$ . As a consequence, the task set is said to be not feasible. Otherwise, if a (m,k)-firm pattern spinning is performed, new time scales can be considered. Figure 6b shows the time scale when just one left spin in (m,k)-firm pattern for task  $\tau_3$  is considered. This spinning generates new arrival times for  $\tau_3$ , being now the task set time scale feasible.

## 5 Simulation and experimental results

The assessment of the proposed (m,k)-firm approach was conducted both by simulation and through an experimental setup using IEEE 802.15.4 sensor nodes. The target of this section is to present those results compared to the results obtained when applying the original (m,k)-firm approach [17] to the same setup. The software code developed for both the simulator and the experimental setup is available for download at [http://www.das.ufsc.br/~montez/\(m,k\)-spin/](http://www.das.ufsc.br/~montez/(m,k)-spin/). The proposed approach implemented in the MicaZ nodes is fully compatible with the IEEE 802.15.4 standard: all the (m,k)-firm constraints sent from the sensor nodes to the coordinator are encapsulated



**Figure 7** Percentage of tasks admitted by (m,k)-spin that were rejected by Ramanathan (with load = 100%).



**Figure 8** Percentage of tasks admitted by (m,k)-rec-spin that were rejected by Ramanathan (with load = 100%).

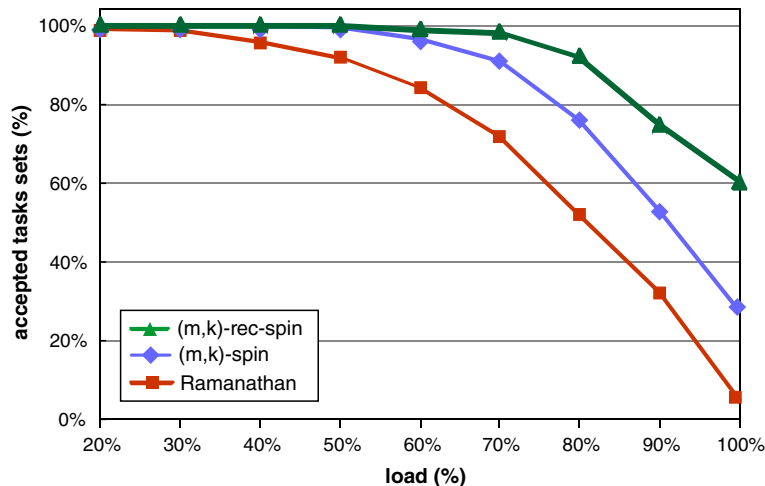
into the payload of the frames; the beacon message sent by the coordinator are not changed.

### 5.1 Simulation results

For the sake of convenience, we adopted the name *Ramanathan* for the algorithm presented in [17] and *(m,k)-spin* for the algorithm proposed in this paper (Algorithm 1). Concerning the *(m,k)-spin* approach, two possibilities were evaluated: (1) *(m,k)-spin*, the spinning operation of the (m,k)-firm pattern is only applied to the last task of the set and (2) *(m,k)-rec-spin*, the spinning operation of the (m,k)-firm pattern may be applied to any task of the set. The *(m,k)-rec-spin* is the recursive version of *(m,k)-spin* and its name is derived from the fact that the schedulability test makes an effort to spin the patterns of higher priority tasks when a task could not be scheduled,

even after making their  $(k_i - 1)$  spins in its (m,k)-firm pattern.

To assess the advantages of the proposed approach, we have built a simulator in C language. Simulations were performed considering a set of tasks (nodes), ranging from 2 to 10 starting at the same time instant. The periods of each task are randomly selected and distributed among 1 to 15. The deadline of each task is assumed to be equal to its period ( $D_i = P_i$ ). The  $m_i$  and  $k_i$  values are also randomly generated, the  $k_i$  value ranging from 2 to 10 and  $m_i$  ranging from 1 to  $k_i$ . The execution time of tasks is generated imposing an utilization factor of each point uniformly distributed in each interval. The utilization factor is divided in intervals of length 10%, ranging from 0.2 to 1.0. In order to reduce the statistical bias, each obtained result is related to 1,000 executions, which means that



**Figure 9** Percentage of accepted tasks, considering only harmonic task sets.

9,000 different task sets have been generated during the simulated interval, for each presented result.

In the literature, we can find approaches [19] where the classification of the task patterns as mandatory or optional is previously defined by a genetic algorithm. In this paper, we adopt a well-defined classification pattern with spins in (m,k)-firm patterns, thus avoiding the combinatorial explosion of classification patterns.

Figure 7 presents the advantages of using spins in (m,k)-firm patterns. The presented assessment refers to a simulation with 100% of load and a (m,k)-firm spinning applied only upon the last task of the task set. It can be observed that, in what concerns the number of accepted tasks, the attained improvement reaches 31% when admitted up to nine spins in the (m,k)-firm pattern. Obviously, beyond this value, the improvement is no longer relevant because the maximum number of admitted spins by one task is  $(k_i - 1)$ . It is interesting to note that even when making just one spin in the last task, there is already an improvement of about 14% in the number of accepted tasks.

Figure 8 illustrates the advantages of the (m,k)-rec-spin approach for the same conditions. In this scenario, all tasks within the task set can spin their (m,k)-firm patterns to improve the number of tasks admitted in the system. It can be observed that about 51% of the tasks were admitted when 150 spins were applied. Nevertheless, the response time to manage the spinning operations increases substantially when increasing the number of spins.

Figure 9 presents the results when varying the load for three specific approaches. The first is the original Ramanathan approach [17], the second considers  $(k_i - 1)$  spins just in the last task of the task set, and the third admits up to  $(k_i - 1)$  spins for all tasks in the task set. In this scenario, harmonic task sets were considered ( $k_i.P_i$  values are multiple of all tasks of lower priorities), resulting in a decreased response time for the task set.

Even for the case of lower system loads, the advantage of the two proposed approaches is highly relevant (Figure 9). It is interesting to note that with a load starting from 50%, the number of accepted task in Ramanathan is considerably smaller. It can be also observed that for the recursive (m,k)-rec-spin approach, there is an acceptance of about 60% for 100% of load. Even for the (m,k)-spin approach, where spins are admitted just for the last task, the improvement in the acceptance rate is above 30% considering 100% of load.

In Tables 1 and 2, columns named as *improvement* represent the improvement reached by the (m,k)-spin approach over the Ramanathan approach [17]. Table 1 illustrates the case when non-harmonic task sets are considered. It is possible to note that the benefits of (m,k)-spin

**Table 1 Non-harmonic task sets acceptance results**

Load	Ramanathan	(m,k)-spin	Improvement (%)
20%	998	999	0.0
30%	994	999	1.0
40%	989	998	1.0
50%	969	974	1.0
60%	890	924	4.0
70%	849	884	4.0
80%	665	734	10.0
90%	391	485	24.0
100%	91	179	97.0

approach over Ramanathan one are significant, specially when the system load exceeds 80%.

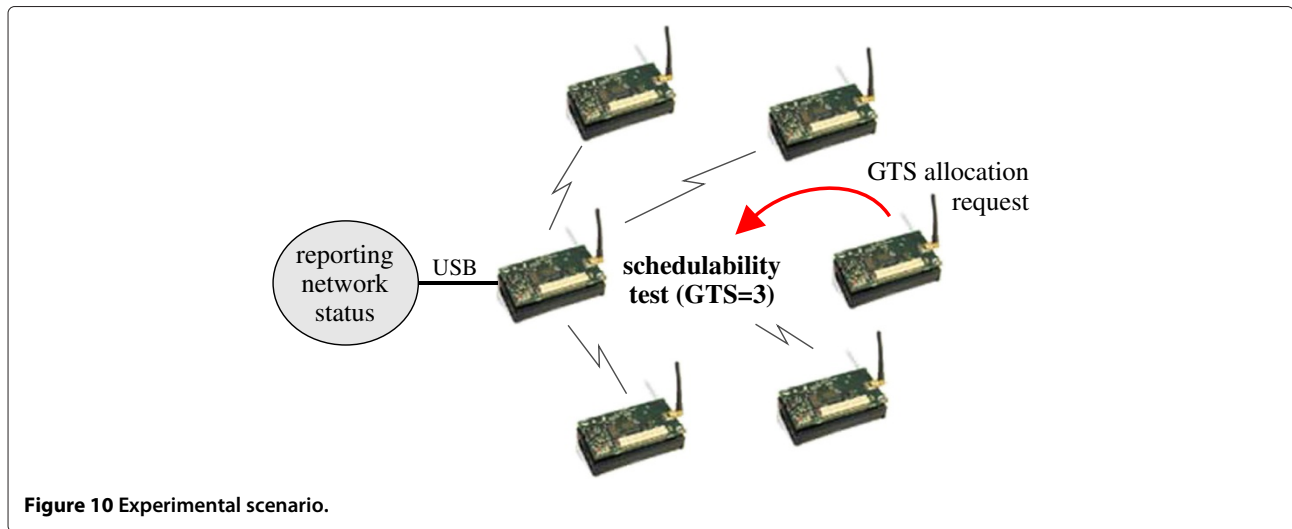
Table 2 considers the case when harmonic task sets are considered. It can be also noted that there is an extra improvement when compared with the results as shown in Table 1. It can be observed that even for lower load values, the benefits of the (m,k)-spin are worth mentioning; the improvements become more significant as the utilization factor values increase.

As a conclusion of the simulation assessment, it can be seen that even with just one spin on the (m,k)-firm classification pattern, the number of jobs accepted in the set may be significantly increased when compared to the original Ramanathan approach [17]. Besides, when general spins on the (m,k)-firm pattern are admitted for all the tasks in the set, the acceptance of the task set is even better. But, this general (m,k)-firm spinning implies the use of more computational resources due to the number of possible combinations of (m,k)-firm patterns that need to be tested.

Regarding the periodicity of the tasks, it is also indicated that harmonic task sets have obtained better results than non-harmonic ones, which is fully understandable as that the approach proposed in [17] does not consider spins on

**Table 2 Harmonic task sets acceptance results**

Load	Ramanathan	(m,k)-spin	Improvement (%)
20%	994	1000	1.0
30%	989	1000	1.0
40%	961	1000	4.0
50%	920	997	9.0
60%	841	968	16.0
70%	721	912	28.0
80%	520	760	46.0
90%	324	530	64.0
100%	48	276	462.0



the (m,k)-firm pattern. This fact becomes more evident with higher utilization factors, when the number of tasks in the set is higher.

### 5.2 Implementation upon IEEE 802.15.4 devices

This section presents the experimental assessment that has been performed to demonstrate the feasibility of implementing the proposed approach upon the sensor devices that are compatible with the IEEE 802.15.4 standard.

The experimental setup is composed of six MicaZ motes running OpenZB (<http://www.open-zb.net/>) operating system in a star topology. The MicaZ nodes are based on the Atmel Atmega 128L microcontroller (San Jose, CA, USA), which operates at 8 MHz with 128 kb of flash programming memory. There are other sensor nodes with better performance, as Sun SPOTs (Santa Clara, CA, USA) which operate at 180 MHz with 4 MB of flash memory. One of the nodes is the coordinator and five others were configured to send GTS allocation requests to the coordinator, containing (m,k)-firm constraints. For the sake of convenience, the number of available GTS considered in this scenario is three, thus there are five nodes competing for a limited number of slots. The superframe structure has been configured using the parameters  $BO = SO = 6$ , therefore, with BI approximately 1 s. To better evaluate the performance of the schedulability test, an experimental

assessment was performed 15 times for each scenario and the task parameters were resorted before starting a new experiment.

In the experimental assessment (Figure 10), the coordinator node performs the schedulability test (Algorithm 1) and reports the results to the nodes. According to the IEEE 802.15.4 standard, the PAN coordinator must take the schedulability decision within  $aGTSDescPersistenceTime$  (where  $aGTSDescPersistenceTime = 4$ ) superframes. Therefore, on receipt of the acknowledge message to the GTS request, the device continues to track beacons and wait for, at most,  $aGTSDescPersistenceTime$  superframes.

The obtained values for the average response time, worst case response time, and the standard deviation are presented in Table 3. It can be observed that even when non-harmonic task sets are considered, the response time (in average) is acceptable for devices compatible with the IEEE 802.15.4 standard. In the experimental assessment scenario, the response times were always under 0.3 s, which is negligible for most part of the applications.

When harmonic task sets with one spin are considered, the response time is significantly better due to the fact that the least common multiple (lcm) of the tasks are smaller; thus, the number of periods to be checked in the task set are significantly smaller.

**Table 3** Response times

	$(k_i - 1)$ spins		1 spin	
	Non-harmonic	Harmonic	Non-harmonic	Harmonic
Response time	143 ms	30 ms	80 ms	20 ms
Worst case	300 ms	60 ms	170 ms	30 ms
Standard deviation	100 ms	15 ms	40 ms	7 ms

## 6 Conclusions

Considering the flexibility of some real-time applications in what concerns tolerating discards and the reduced number of guaranteed time slots available in IEEE 802.15.4 networks, a logical way to follow is to explore the real-time constraints of these networks seeking to optimize the scarce resource utilization, such as the GTSs allocation. Unfortunately, the use of appropriate real-time mechanisms in wireless sensor networks has not been thoroughly investigated. Providing timing guarantees in wireless sensor networks is still an open challenge. This paper presents a new exact schedulability test for tasks with (m,k)-firm constraints. The proposed online solution that may be executed in polynomial time improves the GTS allocation in IEEE 802.15.4 networks when timing requirement is expressed using the (m,k)-firm model.

Simulation results show better usage of the GTS allocation when compared to the traditional approach [17] applied to IEEE 802.15.4 networks. Experimental results obtained using a setup of MicaZ devices point to the feasibility of implementing the proposed approach upon IEEE 802.15.4 sensor nodes.

Contrarily to the classifier proposed by [17] that always classifies as mandatory the first task activation, the proposed classification approach allows spins in the classification pattern. It was observed in the simulation experiments that even when only one spin is considered, the benefits in the schedulability of the system are considerable. It is also important to note that the time consumed by the schedulability test execution does not jeopardize the execution of the tasks already admitted into the system.

Other approaches that intend to extend the number of GTS, such as [3] or [4], could also benefit from our proposal because it enables the execution of more real-time applications in the network, sharing all existing GTSs. Additionally, the proposed approach maintains compatibility with commercial off-the-shelf devices. To our best knowledge, it does not exist in any other works offering QoS for applications that tolerate skips, more specifically in the GTS allocation context, as presented in this paper.

## Endnote

<sup>a</sup>In this work, an instantiation of a task is equivalent to the message transmission by a node.

## Competing interests

The authors declare that they have no competing interests.

## Acknowledgements

The authors acknowledge the support received from the national funding agencies (CNPq, CAPES, and FCT) and also from their institutions (IFSC, UFSC, IDMEC, and FEUP).

## Author details

<sup>1</sup>Federal Institute of Santa Catarina (IFSC), São José 88130-310, Brazil.

<sup>2</sup>Automation and Systems Department, Federal University of Santa Catarina

(UFSC), Florianópolis 88040-900, Brazil. <sup>3</sup>Faculty of Engineering, University of Porto (IDMEC - FEUP), Rua Dr. Roberto Frias, Porto 4200-465, Portugal.

Received: 5 June 2012 Accepted: 19 August 2013

Published: 4 September 2013

## References

1. J Li, Y Song, F Simonot-Lion, Providing real-time applications with graceful degradation of QoS and fault tolerance according to (m,k)-firm model. *IEEE Trans. Ind. Inf.* **2**(2), 112–119 (2006)
2. A Koubaa, M Alves, E Tovar, A Cunha, An implicit GTS allocation mechanism in IEEE 802.15.4 for time-sensitive wireless sensor networks: theory and practice. *Real-Time Syst. J.* **39**(1–3), 169–204 (2008)
3. L Cheng, AG Bourgeois, X Zhang, A new GTS allocation scheme for IEEE 802.15.4 networks with improved bandwidth utilization, in *International Symposium on Communications and Information Technologies*, (Sydney, 17–19 Oct 2007), pp. 1143–1148
4. J Chen, L Ferreira, E Tovar, An explicit GTS allocation algorithm for IEEE 802.15.4, in *16th IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, (Toulouse, 5–9 Sept 2011), pp. 1–8
5. LL Bello, E Toscano, Coexistence issues of multiple co-located IEEE 802.15.4/ZigBee networks running on adjacent radio channels in industrial environments. *IEEE Trans. Ind. Inf.* **5**(2), 157–167 (2009)
6. J Ruiz, JR Gallardo, D Makrakis, L Villasenor-Gonzalez, HT Mouftah, Cross-layer medium access control protocol with quality-of-service guarantees for wireless sensor networks. *EURASIP J. Wireless Commun. Netw.* **2011**(179), 1687–1499 (2011)
7. L Zhang, J Yu, X Deng, Modelling the guaranteed QoS for wireless sensor networks: a network calculus approach. *EURASIP J. Wireless Commun. Netw.* **2011**(82), 1687–1499 (2011)
8. E Toscano, LL Bello, Multichannel superframe scheduling for IEEE 802.15.4 industrial wireless sensor networks. *IEEE Trans. Ind. Inf.* **8**(2), 337–350 (2012)
9. T Semprebom, C Montez, R Moraes, F Vasques, P Portugal, Dynamic GTS scheduling of periodic skippable slots in IEEE 802.15.4 wireless sensor networks, in *Proceedings of the 8th IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems*, (Ansan, 20–22 May 2009). **8**(1), pp. 110–117
10. IEEE, *Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs)*, *IEEE Std 802.15.4-2006 (Revision of the IEEE Std 802.15.4-2003)* (IEEE, New Jersey, 2006)
11. G Bernat, A Burns, A Llamós, Weakly hard real-time systems. *IEEE Trans. Comput.* **50**(4), 308–321 (2001)
12. F Flavia, J Ning, S YeQiong, SL Francoise, Impact of a (m,k)-firm data dropouts policy on the quality of control, in *IEEE International Workshop on Factory Communication Systems*, (Torino, 28–30 Jun 2006), pp. 353–359
13. L Schenato, M Franceschetti, SS Sastry, Foundations of control and estimation over lossy networks. *Proc. IEEE.* **95**, 163–187 (2007)
14. A Willig, Recent and emerging topics in wireless industrial communications: a selection. *IEEE Trans. Ind. Inf.* **4**(2), 102–124 (2008)
15. EF Nakamura, AAF Loureiro, AC Frery, Information fusion for wireless sensor networks: methods, models, and classifications. *ACM Comput. Surv.* **39**(3), 1–55 (2007)
16. G Anastasi, M Conti, MD Francesco, A comprehensive analysis of the MAC unreliability problem in IEEE 802.15.4 wireless sensor networks. *IEEE Trans. Ind. Inf.* **7**, 52–65 (2011)
17. R Parameswaran, Overload management in real-time control applications using (m, k)-firm guarantee. *IEEE Trans. Parallel and Distributed Syst.* **10**(6), 549–559 (1999)
18. J Goossens, (m k)-firm constraints and DBP scheduling: impact of the initial k-sequence and exact schedulability test, in *16th International Conference on Real-Time and Network Systems*, (Rennes, 16–17 Oct 2008), pp. 61–65
19. G Quan, XS Hu, Enhanced fixed-priority scheduling with (m,k)-firm guarantee, in *21st IEEE Real-Time Systems Symposium*, (Orlando, 27), pp. 79–88
20. R West, Y Zhang, Dynamic window-constrained scheduling of real-time streams in media servers. *IEEE Trans. Comput.* **53**(6), 744–759 (2004)
21. B Andersson, N Pereira, E Tovar, Analysing TDMA with slot skipping. *IEEE Trans. Ind. Inf.* **4**(4), 225–236 (2008)

22. G Koren, D Shasha, Skip-over: algorithms and complexity for overloaded systems that allow skips, in *Proceedings of the 16th IEEE Real-Time Systems Symposium*, (Pisa, 5–7 Dec 1995), pp. 110–117
23. M Hamdaoui, P Ramanathan, A dynamic priority assignment technique for streams with (m,k)-firm deadlines. *IEEE Trans. Comput.* **44**(12), 1443–1451 (1995)
24. G Bernat, A Burns, Combining (n/m)-hard deadlines and dual-priority scheduling, in *Proceedings of the 18th IEEE Real-Time Systems Symposium*, (San Francisco, 5 Dec 1997), pp. 46–57
25. Z Wang, J Chen, Y Sun, An integrated DBP for streams with (m,k)-firm real-time Ggarantee. *J. Zhejiang Univ. Sci.* **5**(7), 816–826 (2004)
26. H Cho, Y Chung, D Park, Guaranteed dynamic priority assignment schemes for real-time tasks with (m, k)-firm deadlines. *ETRI J.* **32**(3), 422–429 (2010)
27. K Jeffay, DF Stanat, CU Martel, On non-preemptive scheduling of periodic and sporadic task, in *Proceedings of the 12th IEEE Real-Time Systems Symposium*, (San Antonio, 4–6 Dec 1991), pp. 129–139
28. M Jonsson, K Kunert, Towards reliable wireless industrial communication with real-time Guarantees. *IEEE Trans. Ind. Inf.* **5**(4), 429–442 (2009)
29. N Audsley, A Burns, M Richardson, K Tindell, AJ Wellings, Applying new scheduling theory to static priority pre-emptive scheduling. *Softw. Eng. J.* **8**(5), 284–292 (1993)
30. K Tindell, A Burns, AJ Wellings, An extendible approach for analysing fixed priority hard real-time tasks. *Real-Time Syst. J.* **6**(2), 133–151 (1993)
31. CL Liu, JW Layland, Scheduling algorithms for multiprogramming in hard-real-time environment. *J. Assoc. Comput. Machinery.* **20**, 46–61 (1973)
32. JP Lehoczky, Fixed priority scheduling of periodic task sets with arbitrary deadlines, in *Proceedings of the 11th IEEE Real-Time Systems Symposium*, (Lake Buena Vista, 5–7 Dec 1990), pp. 201–209
33. M Joseph, P Pandya, Finding response times in a real-time system. *Comput. J.* **29**(5), 390–395 (1986)
34. R Murray, K Astrom, S Boyd, R Brockett, G Stein, Future directions in control in an information-rich world. *Control Syst., IEEE.* **23**(2), 20–33 (2003)
35. I Bate, P Nightingale, A Cervin, Establishing timing requirements and control attributes for control loops in real-time systems, in *Proceedings of the 15th Euromicro Conference on Real-Time Systems*, (Porto, 2–4 Jul 2003), pp. 121–128
36. P Marti, J Yepez, M Velasco, R Villa, J Fuertes, Managing quality-of-control in network-based control systems by controller and message scheduling co-design. *Ind. Electron., IEEE Trans.* **51**(6), 1159–1167 (2004)

doi:10.1186/1687-1499-2013-222

**Cite this article as:** Semprebom et al.: (m,k)-firm pattern spinning to improve the GTS allocation of periodic messages in IEEE 802.15.4 networks. *EURASIP Journal on Wireless Communications and Networking* 2013 **2013**:222.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---